Symmetry and efficiency in complex FIR filters

DOI:
10.6100/IR652955

Document status and date:
Published: 01/01/2009

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Symmetry and Efficiency in Complex FIR Filters

Fons Bruekers

The work described in this thesis has been carried out
at the PHILIPS RESEARCH LABORATORIES Eindhoven,
the Netherlands, as part of the Philips Research Programme.

CIP-gegevens Koninklijke Bibliotheek, Den Haag
Bruekers, Fons
Symmetry and Efficiency in Complex FIR Filters
Proefschrift Technische Universiteit Eindhoven,
-Met literatuur opgave,
-Met samenvatting in het Nederlands.
ISBN 978-90-74445-87-0
Trefw.: digital filters, FIR filters, complex filters, linear-phase filters, multirate filters, polyphase filters, symmetry, efficiency, arithmetic complexity, digital signal processing.

About the cover:
A symmetric complex impulse response, surrounded by several of its projections.

Cover design:
Frans Schraven

Symmetry and Efficiency in Complex FIR Filters

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 23 september 2009 om 16.00 uur

door

Alphons Antonius Maria Lambertus Bruekers

geboren te Nederweert

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. R.J. Sluijter
en
prof.dr. Y.C. Lim

Copromotor:
dr.ir. A.W.M. van den Enden

Voor: Anke, Stéphanie, Charlotte,
Pap en Mam

Preface

Summary

Symmetry and Efficiency in Complex FIR Filters

The main contribution of this thesis is a series of novel methods for the design of symmetric and efficient complex FIR filters, including: i) the reduction over complex integer coefficients of generalized-Hermitian-symmetric filters to Hermitian-symmetric filters, ii) the introduction of alternative structures for complex filters, and iii) a general applicable recipe for the restoration of symmetry in multirate polyphase filter structures.

Chapter 1: Introduction In the field of Digital Signal Processing (DSP) filters play an important role. For instance, digital filters used in radio transmitters and receivers operating at a high sampling rate, form an interesting class. For these filters, efficiency is crucial. Application of filters with a different behaviour for positive and negative frequencies is beneficial in many cases such as in multirate systems. In such filters some coefficients will be complex. This thesis focuses on methods for improving the efficiency of symmetric filters. Finite Impulse Response (FIR) filters with a symmetric impulse response show a linear phase frequency response.

This introductory chapter gives the story behind the title of this thesis, and sketches the field of DSP in general and of digital-filter design in particular. Next, the value of complex filters is explained.

The inspiration for writing this thesis arises from the experiences with the development and use of the DESFIL software package for filter design, from which some background information will be presented. Many of the results presented in this thesis can be used in future versions of filter-design tools like DESFIL. Next, the background of the three main research questions that are treated in this thesis are explained. These questions are the following.

Is it relevant to design generalized-Hermitian-symmetric filters?

What structures implement generalized-Hermitian-symmetric filters?

Is it possible to restore the symmetry in polyphase filter structures?

Subsequently the outline of the thesis is presented. Finally, the notational aspects as they appear in this thesis will be introduced.

Chapter 2: Symmetric filters Because of their linear-phase property, symmetric filters form an interesting class of FIR filters. Moreover, symmetric FIR filters allow for an efficient implementation. Non-symmetric FIR filters are briefly addressed in Chapter 4 and Chapter 5. In this chapter the classical definition of Hermitian symmetry is extended to a more general definition that is also applicable to complex filters, generalized-Hermitian or (σ, μ)-symmetry, where σ is the *shape of symmetry* and μ the *center of symmetry*, with $|\sigma| = 1$, $\sigma \in \mathbb{C}$ and $\mu \in \mathbb{Z}/2$. The usefulness of this novel definition that allows for a unified treatment of even- and odd-length filters is shown extensively. Also a number of interesting properties that are used in the following chapters, are presented and derived. Special attention is also paid to symmetric filters with finite precision coefficients. For these filters, new theorems on reducing any (σ, μ)-symmetric FIR filter to a $(1, \mu)$- or (j, μ)-symmetric filter are presented. Based on these theorems, a procedure is designed that can be used to reduce such (σ, μ)-symmetric filters. An example showing the possible savings in arithmetic costs by applying the reduction procedure is discussed in detail.

Chapter 3: First- and second-order filters Examples of simple filters are the low-order FIR filters. For the first- and second-order FIR filters the possibilities to position their transmission zeros in the z-plane for a limited range of coefficient values, are studied. In addition it is shown that the newly defined (j, μ)-symmetric complex filters may be beneficial over the $(1, \mu)$-symmetric complex filters depending on the given specification.

Chapter 4: Transversal and complex structures The transversal filter structure is one of many possible structures for both symmetric and non-symmetric FIR filters. Important properties of this structure are: i) coefficients are identical to the samples of the impulse response, ii) the coefficients are invariant under the polyphase decomposition for multirate filters, and iii) pipelining can be incorporated in a trivial way. Moreover the transversal structure itself may also be part of a composed filter structure.
For the purpose of making filter structures more efficient in terms of costs, this chapter shows how (σ, μ)-symmetry can appear in the transversal structure and how it can be exploited. It gives an overview of known structures and structures inspired by the novel definition of symmetry. When two filters have inputs or outputs in common, interesting structures exist. Various alternatives to decompose complex filters or coefficients into their individual real and imaginary parts are discussed, and compared in detail. Also new structures for efficiently combining conjugate coefficients have been found and subsequently involved in a detailed comparison of computational costs of filters.

Chapter 5: Polyphase structures One of the most important concepts in multirate filtering is the polyphase decomposition and the closely related polyphase filter structure. This concept allows for efficient implementations of interpolating and decimating filters. However, application of this decomposition to linear-phase filters, in many cases destroys the symmetry that could have been exploited to reduce computational costs, as elaborated in the previous chapter.
Central to this chapter is the restoration of the symmetry in polyphase structures. A new theorem states that any real or complex multirate (σ, μ)-symmetric filter with integer or

rational interpolation or decimation factors can be constructed from symmetric filters in a polyphase structure. An example procedure for restoring the symmetry is presented and applied to several examples to show its value.

Chapter 6: Conclusions This final chapter provides the answers to the three main research questions treated in this thesis, and lists a number of possible interesting topics for future research.

Appendices A variety of appendices support the discussions and analyses in the main part of this thesis. First there is a collection of identities for multirate and complex systems including their proofs, followed by brief introductions into pipelining, analog polyphase filters and Euclid's algorithms. Next, interesting alternative constructions to implement multiplications with integer and complex-integer coefficients are discussed and many examples are presented. Finally the complex-base numbers and complex primes are briefly introduced.

History of this thesis

In 1986, I moved to research in the field of DSP in general and filter design in particular, where Ad van den Enden played an important role, first as my tutor and later as sparring partner. I have participated, as a core member or consultant, in many research projects where, in one way or another, digital filters were needed.
First ideas for filterbanks in audio coding resulted in for that time too costly or infeasible structures. Together with others, more efficient alternative structures have been derived and analyzed. As a by-product, new networks for perfect inversion and perfect reconstruction were developed: the ladder networks. In another project, new digital radio and television receiver structures were designed. An analog to digital converter placed close to the receiving antenna, operating at a very high sampling rate, produces heavily oversampled signals. Application of simple decimating filters with complex-valued coefficients enabled a significant reduction of implementation costs. Before these filters were accepted, lengthy and intensive discussions were needed.
Although literature at that time provided us with many filter design methods, it was key to have the relevant algorithms available ”under the keyboard”. In a step-by-step approach I have developed the DESFIL software package. Quantized coefficients could be designed for real or complex FIR filters, stand-alone or in cascade or parallel, mono- or multirate, in a way that an exhaustive search was still efficient. Many colleagues within Philips, active in the field of research, development and training, have used DESFIL and provided me with valuable feedback. Due to a shift of my interest towards new research topics like lossless coding and watermarking, I did not continue the extension of DESFIL with the many ideas and requests that had accumulated.
Except for the ladder networks [22], it was decided not to go for publications on filter design, but to write internal reports, e.g., [15] [21] [25] [28]. However, for a number of filter design and filter application ideas, we have applied for patents, and so far 8 of them have been granted as US patent, viz., [16] [19] [30] [118] [119] [121] [124] [134]. Also,

in several chapters of his PhD thesis [135], Ad van den Enden has described some of the results of our cooperation.

My thesis can be seen as a consolidation of the many unpublished and unimplemented ideas. Issues like symmetry, low-order filters, transversal and complex structures, polyphase structures, and coefficients are discussed in separate chapters of this thesis. While writing, some of the concepts were improved by means of a more formal approach, using lemmas and theorems. I believe that the material provided in this thesis, the examples, theorems and identities, is of value to those who want to extent their toolbox for the design and analysis of symmetric and efficient complex FIR filters.

Acknowledgement

It all started when, within a few months of each other, Fred Boekhorst, Jan Bergmans, Hans Peek and Ad van den Enden, more or less independently triggered me, and finally convinced me that writing a Ph.D. thesis would be a logical next step in my scientific career. I would like to thank them for this.

My work over more than 2 decades at Philips Research Laboratories, Eindhoven (the NatLab), served as a source of inspiration for the subjects treated in this thesis. I thank Hans Brandsma, Ben Waumans, Hans Peek, Theo Claasen, Peter van Otterloo, Rick Harwig, Fred Boekhorst, Carel-Jan van Driel, Jean-Paul Linnartz, Willem Jonker and Bart van Rijnsoever from the NatLab management, who offered me trust and opportunities to develop my scientific skills and later also to work on this thesis.

I am very grateful to Ad van den Enden, who introduced me to the field of DSP and filter design. Ad had the unflagging patience to answer all my questions, to discuss my many, often crazy, ideas, and to serve as sparring partner in discussions at a very high sound level, during many enjoyable years at the NatLab. It was also his thesis that motivated me to write one myself.

The DESFIL filter design software package was developed while I was working with Ad. During this development I received much support from many colleagues in research, development and the ICT department. In particular, Wim Verhaegh supported me in solving various optimisation problems and Ton Nillesen gave indispensable feedback from a user perspective. I am convinced that without their support, DESFIL would not have been as valuable and popular inside and outside Philips as it is today.

I thank Jan Bergmans, Paul Hovens, Johan van Valburg, Marc Arends, Pepijn Boer, Evert-Jan Pol and David McCulloch, who contributed in various ways to the solving of the many tasks that crossed my path when I was writing my thesis. Of course, I must thank many colleagues, too many to mention without forgetting at least some of them, for the various kinds of support they gave me while working on this thesis over the many years that it took.

I owe much gratitude to my former colleague, Ton Kalker, who taught me how to use mathematics to construct proofs. The many inspiring discussions resulted in theorems that, for a long time, were only conjectures. Without Ton this thesis would not have had the mathematical basis as it has now.

It was in a cosy restaurant where, by chance, I met my former colleague, Rob Sluijter, whom I have known for more than 30 years. He inquired after the progress of my thesis, and, during a couple of meetings that followed, it was Rob who convinced me that the thesis was ready to be finalized. It needs no explanation that I am very grateful for this.
Both Rob, as my first promoter, and Ad, as my co-promoter, have read and commented on draft versions of this thesis, and I have really enjoyed and greatly appreciated the many evening sessions that we had together, where all the ins and outs of the various topics were discussed in detail. This shows that serious matters and fun can be combined very well. Rob and Ad, thank you very much for all your effort!
I am also very grateful to my second promoter prof.dr. Yong Ching Lim, and to the other members of the promotion committee: prof.dr.ir. Jan Bergmans, prof.dr.ir. Marc Moonen, dr. Hennie ter Morsche and prof.dr.ir. Kees Slump, for reviewing my thesis and giving me valuable feedback.
Last but not least, I am especially grateful to my parents, my wife Anke, and my daughters, Stéphanie and Charlotte, for their encouragement and support of all my activities.

Contents

Glossary of acronyms, symbols and notations

Acronym	**Description**
AC	Average Costs
B&B	Branch and Bound
clk	system clock
CRT	Chinese Remainder Theorem
CSD	Canonical Signed Digit
dB	deciBell
FA	Full-Adder
FIR	Finite Impulse Response
$\gcd$	greatest common divisor
HA	Half-Adder
HS	Half-Subtractor
iff	if and only if
IFIR	Interpolated FIR, Interpolated Finite Impulse Response
IIR	Infinite Impulse Response
ILP	Integer Linear Programming
LP	Linear Programming
MILP	Mixed Integer Linear Programming
PPC	PolyPhase Component
rad	radian
ROC	Ratio Of Costs
SRD	Sampling Rate Decreaser
SRI	Sampling Rate Increaser

Symbol / Notation	**Description**
j	$\sqrt{-1}$
$\lceil x \rceil$	smallest integer at least x, ceil
$\lfloor x \rfloor$	largest integer at most x, floor
$a \leftarrow b$	b substitutes a, a is replaced by b
$a \vert b$	a is a divisor of b
$a\vert_b$	remainder from the integer division of a by b, Section 1.9.5 on page 14
$\gcd(a,b)$	greatest common divisor of the non-zero integers a and b
a^*	conjugate of a
$(.)^{\overline{*}}$	special conjugate, Definition 1.6 on page 14
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$\triangleq$	defined as
$\wedge$	logical and
$\vee$	logical or
or	logical or
xor	logical exclusive or
$\in$	element in
$\ni$	contains the element
$\cap$	intersection
$\cup$	union
$\subset$	strict or proper subset
$\subseteq$	subset
$A \backslash B$	set A minus set B
$\mathbb{N}$	set of natural numbers including 0
$\mathbb{N}^+$	set of natural numbers excluding 0
$\mathbb{Z}$	set of integers
$\mathbb{Z}/2^i$	set of scaled integers, Definition 1.1 on page 13
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
$\mathbb{C}_\mathbb{Z}$	set of complex integers, Definition 1.2 on page 14
$\mathbb{C}_{\mathbb{Z}/2^i}$	set of scaled complex integers, Definition 1.3 on page 14
$\mathbb{C}_\mathbb{Z}/\mathbb{C}_\mathbb{Z}$	set of complex rationals, Definition 1.4 on page 14
$\vert . \vert$	modulus of a complex quantity
$\mathcal{P}(.)$	phase of a complex quantity
$\Re$, $\Re(.)$	real part of a complex quantity
$\Im$, $\Im(.)$	imaginary part of a complex quantity
$\Vert . \Vert_p$	p-norm of a complex quantity (different from not L_p-norm), Section 1.9.6 on page 15
$\Vert . \Vert_1$, $\Vert . \Vert_\infty$	1-norm and ∞-norm according to Definition 1.10 on page 16

Symbol / Notation	**Description**
θ	relative frequency
z	complex discrete-time frequency
z^{-1}	discrete-time unit delay
W_D	twiddle factor, Lemma A.2 on page 113
$\mathscr{F}$	Fourier transform, Section A.6 on page 120
$h[n]$	impulse response or signal
$h_{\mathrm{r}}[n]$	real part of $h[n]$
$h_{\mathrm{i}}[n]$	imaginary part of $h[n]$
$H(z)$	system function or z-transform of $h[n]$
$\langle h[0], \cdots, h[i] \rangle$	system function or z-transform of $h[n]$, Section 1.9 on page 12
$H^{\overline{*}}(z)$	system function or z-transform of $h^*[n]$, Definition 1.6 on page 14
$H_{\mathrm{r}}(z)$	system function or z-transform of $h_{\mathrm{r}}[n]$
$H_{\mathrm{i}}(z)$	system function or z-transform of $h_{\mathrm{i}}[n]$
$H(\mathrm{e}^{\mathrm{j}\theta})$	frequency response or Fourier transform of $h[n]$
$H_{\mathrm{r}}(\mathrm{e}^{\mathrm{j}\theta})$	frequency response or Fourier transform of $h_{\mathrm{r}}[n]$
$H_{\mathrm{i}}(\mathrm{e}^{\mathrm{j}\theta})$	frequency response or Fourier transform of $h_{\mathrm{i}}[n]$
$H_{zp}(\theta)$	zero-phase frequency response or Fourier transform of $h[n]$
$H_{\mathrm{r},zp}(\theta)$	zero-phase frequency response or Fourier transform of $h_{\mathrm{r}}[n]$
$H_{\mathrm{i},zp}(\theta)$	zero-phase frequency response or Fourier transform of $h_{\mathrm{i}}[n]$
$\mathbf{H}(z)$	matrix or vector of system functions or z-transforms
$\downarrow D$	SRD with decimation factor D
$\uparrow I$	SRI with interpolation factor I
D	decimation factor of an SRD
I	interpolation factor of an SRI
$H_{R:r}(z)$	polyphase component $R:r$ of $H(z)$, Definition 5.1 on page 82
R	decomposition factor
r	decomposition index
$\mathrm{path}(r)$	r^{th} path of a multirate filter, Definition 5.3 on page 90
$\mathcal{R}, \mathcal{D}, \mathcal{I}$	index set, Definition 5.2 on page 84
$\mathcal{R}_0, \mathcal{D}_0, \mathcal{I}_0$	fundamental index set, Definition 5.2 on page 84
L	filter length
S	specification of a filter, Definition 2.8 on page 46
α, β	minimal factor, Definition 2.6 on page 38
σ	shape of symmetry, Definition 2.1 on page 19
μ	center of symmetry, Definition 2.1 on page 19
$\mathcal{M}(.)$	mirroring operator, Definition 2.2 on page 22
$\mathcal{Q}(.)$	quantization function, Definition 2.5 on page 33
Ξ	coefficient range, Section 1.9.6 on page 15

Symbol / Notation	Description
;	connector, Section 1.9.3 on page 13
$\mathcal{B}$	binary constructions for integer coefficients, Section E.1 on page 144
$\mathcal{C}$	CSD constructions for integer coefficients, Section E.1 on page 144
$\mathcal{AC}$, $\hat{\mathcal{A}}\mathcal{C}$	alternative constructions for integer coefficients, Section E.2 on page 146
$(\mathrm{X}, \mathcal{X})$	constructions for complex integer coefficients with global constructions X and local constructions $\mathcal{X}$, Section E.3 on page 147
$\mathcal{A}(\mathrm{X}, \mathcal{X})$, $\hat{\mathcal{A}}(\mathrm{X}, \mathcal{X})$	alternative constructions for complex integer coefficients, Section E.3 on page 147
$C_{\mathcal{X}}(a)$	minimal costs for coefficient a using constructions $\mathcal{X}$, Definition E.1 on page 145
$AC_{\mathcal{X}}(\Xi)$	average costs for coefficient range Ξ using constructions $\mathcal{X}$, Definition E.2 on page 145
ROC	ratio of costs, Section F.2 on page 167
$\Gamma_{\mathcal{X}}(a)$	minimal constructions for coefficient a using constructions $\mathcal{X}$, Definition E.1 on page 145
$\#$	number of $\cdots$
$\#add$	number of additions
$\#bit$	number of bits
$\#mul$	number of multiplications
$\#shift$	number of shifts
$\oplus$	adder, Section E.1 on page 144
$\ominus$	subtractor, Section E.1 on page 144
$\circledast$	cyclic convolution

List of Figures

List of Tables

Chapter 1

Introduction

This first chapter provides an introduction to the topic of this thesis, namely, symmetry and efficiency in complex FIR filters. After discussing the title in Section 1.1, a general overview of digital signal processing is given in Section 1.2. In Section 1.3, the focus is directed towards the design of digital filters in general, and, in Section 1.4, the relevance of complex filters is addressed in particular.
The most important sources of inspiration for the issues treated in this thesis are the development and the extensive use of the filter-design tool: DESFIL. Also, results from this thesis can be exploited in future versions of DESFIL. Therefore Section 1.5 presents a flavour of what can be done with DESFIL, and Section 1.6 explains how DESFIL is organized.
The research questions that are addressed in this thesis are formulated in Section 1.7 and the outline of the thesis is given in Section 1.8. Finally, Section 1.9 treats the special notations and definitions that are used throughout the thesis.

1.1 About the title

The first impression of the title may be a bit confusing, because of the words ”Efficiency” and ”Complex”, that may express opposite properties.
Basically this thesis is about the design and analysis of Finite Impulse Response (FIR) Filters that form an interesting component, or functionality, in the field of digital signal processing. The focus is mainly on the design and analysis of FIR filters that are somehow optimised to operate at high frequencies, have little dissipation or result in smaller chips. One way to express all this in the title, for instance, could have been by using the words ”Low Power” and ”Small”, giving for instance: ”Design and Analysis of Low Power Small FIR Filters”.
Throughout this thesis, many concepts are described that enable the design of low cost FIR filters, a prominent one of which is the use of complex-valued coefficients and signals. ”Complex valued” implies that the value has both a real and imaginary part, and, in signal processing terminology, this means that frequency responses are not necessarily

symmetric. Much attention will be paid to the design of FIR filters with complex-valued coefficients. A title also comprising this complex-valued work could for instance be: ”Design and Analysis of Low Power Small FIR Filters with Complex-Valued Coefficients and Signals”.
Another important concept is the symmetry in FIR filters. This symmetry gives the filter a linear-phase frequency response, and may be exploited to improve the efficiency in filters. To cover this also, the word ”Symmetry” could be added to the title.
Personally I prefer a more compact version of the title with the word ”Efficiency” representing ”Low Power” and ”Small”, and ”Complex” representing ”Complex-Valued Coefficients and Signals”, resulting in: ”Symmetry and Efficiency in Complex FIR Filters”. The apparent contradiction in terms may fascinate and hopefully invites to read this thesis.

1.2 Digital signal processing

About four decades ago digital signal processing came to life. At that time both the discrete-time signal processing theory and the digital technology were ready to be combined and used. Theory enabled the design and the analysis of discrete-time systems, whereas digital hardware in general, but digital computers in particular, made it possible to actually run experiments.
A clear advantage of digital signal processing is that, in principle, any specification can be met, if the effort is just large enough. By increasing the sampling frequency, the accuracy of the signal representation and the number of operations, basically any function can be realized since both the signal bandwidth and the signal-to-noise ratio are increased.
At any moment in time, there is an upper limit to the sampling frequency, the accuracy of the signal representation and the number of operations that can be used in a digital system. The limit is strongly determined by the field of application. For professional applications like military equipment, space exploration and the oil industry, price is hardly an issue, only performance counts. In the field of consumer lifestyle and healthcare applications, next to the signal processing performance also the price is important. This is one of the reasons why digital signal processing appeared much later in consumer applications than in professional applications.
A wide range of discrete-time signal processing algorithms is needed to make systems such as CD-players or MPEG-coders: correlation, transformation, filtering, error correction, channel coding, entropy coding, signal decomposition, ... To the user, these algorithms are mostly hidden and the user even does not care about these details as long as the ”black” box operates satisfactory. Only in exceptional cases are individual algorithms presented to the user. In general, a user is only interested in derived properties, such as: performance, price, size, weight and battery lifetime. In research, however, a lot of attention is paid to the improvement of the many digital signal-processing algorithms, where many alternatives and new concepts are studied.
In this thesis, the focus is on the design of the frequently used discrete-time or digital FIR filters, with an emphasis on the reduction of computational cost. This type of filter has already been used for some decades, and many methods for their design are known. Also, the chip technology, often used to implement digital filters, is developing at an enor-

mous pace. In 1965, Gordon Moore [93] made the observation, known as Moore's law, that since 1959, the complexity for minimum component costs has increased at a rate of roughly a factor two per year, and he believed that the rate would be constant for at least another 10 years. Since then the rate has decreased only a bit and today millions of components can be placed at a chip. Nevertheless, it is important to optimise the complexity of digital filters since they are applied in areas where the limits are met. No matter how high the system clock frequency may be, some filters have to operate close to that, as in communication transmitters and receivers. Also, the power dissipation of a chip is important. Clearly, the less energy is used in a system the better the environmental values are preserved, but functionality can be improved too. In mobile applications especially, where the energy has to come from a battery, a reduction of the dissipation is an increase in operation time. In the context of mass production, also referred to as high volume electronics, aspects like chip area are important too, since chip area is directly related to price.

In this thesis, it will be shown that alternative filter structures can result in a significant reduction of the number of operations, and how these filter structures can be designed. As a consequence, more functionality, or the same functionality on a smaller chip, or the same functionality with less dissipation, can be implemented in current state technology.

1.3 Digital-filter design

The field of digital-filter design is not new, and many of its aspects are described abundantly. In this section, a brief anthology is presented to sketch the field of filter design. In the following chapters, more references are made to relevant literature.

The area of filter design is extremely wide. Structures and specifications range from very basic tapped delay lines for a low-pass filter, to compositions of adaptive recursive structures for sound equalizers and multirate perfect-reconstruction filterbanks in signal coding schemes.

In the process of elaborating a filter design problem, a number of steps can be identified. An early step is the analysis of the problem at hand, and subsequent steps include the generation of (preferably many) alternative solutions to choose from. Also, the platform for implementing the final solution is important. Software or hardware implementations have their typical properties that may impose particular filter design constraints.

When analyzing a filter problem and searching for alternatives, it is a great advantage when many different strategies can be followed and evaluated. The more methods one knows the better the final solution can be. In literature many filter design techniques are described and some of them may apply to the filter problem at hand.

Even for the limited class of linear-phase FIR filters many methods are available. The most popular one is perhaps the method from Parks and McClellan [111] presented in 1972, and [89] that is based on the Remez algorithm. Quadratic minimization in [61] is a rather straightforward method while simulated annealing [117], genetic-algorithms [109], neural networks [7] or tabu search algorithms [71] can be used for exotic structures and specifications. Background information about these optimization algorithms can be found in [1].

Compositions of filters can be found in [55] and [69], for instance. Here, the concept of post-processing to improve filter performance is introduced and elaborated upon. In [98] Neuvo introduced the Interpolated FIR (IFIR) filters, and in [83], Lim introduced the concept of frequency masking. Another example is the control of derivatives in frequency responses, as described in [70].
In practice, it often appears to be a non-trivial step to move from a theoretical description of a filter design method to, for instance, a signal-flow graph and practical values for coefficients. Also, many papers describe a method to obtain unquantized values for coefficients, while quantized coefficients are needed. Therefore, the availability of computer tools to support the work is of great importance for a designer.
Today, filter design tools can be purchased as a stand-alone function or as part of a complete design environment. Also via the Internet a lot of filter design functionality is offered, often for free, but accompanied with many disclaimers. In these tools, exotic design methods are found, as well as the popular straightforward ones. In most cases, only unquantized coefficients are derived, and the, for many applications important, quantized coefficients are best obtained by rounding the unquantized values.....

1.4 Relevance of complex filtering

The attention in literature for complex filters is very limited compared to real filters. Complex filters, with linear and non-linear phase, can be designed using several algorithms, e.g., [31] and [81]. The practical value of complex filters in multirate systems was not recognized until 2001 [135]. In the remainder of this section it is explained how complex filters relate to the given specification, and how complex filters may contribute to the design of efficient multirate filters.

1.4.1 Specification

Fundamental for complex filters is their property to have different frequency responses for positive and negative frequencies: the frequency response is non-symmetric around relative frequency $\theta = 0$. Some complex filters may be designed by starting with a real filter as follows. In case the filter specification is non-symmetric around $\theta = 0$, but is symmetric around $\theta = \theta_c$, the desired impulse response $h[n]$ can be designed by first designing a real impulse response $g[n]$ satisfying a modified specification that is symmetric around $\theta = 0$, and subsequently modulating this impulse response like $h[n] = g[n]\mathrm{e}^{\mathrm{j}\theta_c n}$. If the desired filter should have quantized coefficients, the previous approach is generally unsuitable. In cases where a filter specification has no symmetry around any frequency, the modulation method cannot be applied either. These filters have to be designed directly as complex filters.

1.4.2 Multirate

In the front-end of digital communication receivers there is typically a signal band that is relatively narrow compared to the sampling frequency. For improving the efficiency of the signal processing, a first step is to reduce the sampling frequency, by applying a decimating filter. It is well known that, as a first-order approximation, it can be assumed that the filter length of a band-pass filter is inversely proportional to the transition bandwidth [56] [60] [70]. This implies that a filtering scheme in which the transition band can be wider, the resulting filter lengths will be shorter. Complex filters can successfully be used here. Selecting only part of the narrow-band real input signal, such as the positive frequencies only, allows the decimating filters to have a very wide transition band. One consequence of introducing complex filters is that subsequent processing is on complex signals, which, as such, will increase costs. However the savings in the first stages of the receivers are huge, and in many receiver structures, parts of the traditional processing is already complex.

1.5 DESFIL

About 15 years ago, our attention in signal-processing research was directed to the design of decimating filters needed in digital radio and television receivers [15] [16] [20] [24] [51] [124] [134]. Characteristic of this kind of application is that the sampling rate is very close to the maximum system-clock frequency, and that the filters are consequently implemented in dedicated hardware. Low power dissipation is crucial for mobile communication equipment especially, and low-power design of the complete system is essential. No appropriate tools were available at that time, to design suitable filters, so the tools had to be user-built. Step by step, a filter-design tool has been developed that is suited for the design of linear-phase multirate FIR filters of relatively short length, with optimally quantized (and possibly complex) coefficients, to obtain efficient solutions.
Many known methods have been be combined and extended with new insights. The tool is called DESFIL, which is short for Design and Evaluation Software for FILters, and consists of many programs with clearly defined tasks [28]. Next to the programs for designing a filter, some supporting programs are desired for displaying results and some elementary manipulations on filters. What initially was meant to be a tool supporting own research only, gradually became a tool used within many Philips research and development laboratories [105]. Today, DESFIL is still in use even outside Philips.
The inspiration for this thesis mainly originates from the development and usage of DESFIL. In addition, results presented in this thesis can be used in new versions of DESFIL. Therefore this section will briefly touch upon various aspects of DESFIL, and section 1.6 will discuss in more detail, the two-step approach that is used.

1.5.1 Alternative tools

As already mentioned in Section 1.3, much theory on filter design is available, but, if there is no operational tool that exploits a particular theory, it is still a long way to an effective filter solution. Of course, there was the very popular Parks-McClellan tool [111], based on the Remez-exchange algorithm, made freely available by the IEEE [89], ideal for very long linear-phase FIR filters, but not suited for complex coefficients, nor for the quantization of coefficients. MATLAB was already available, and supported, along with some filter design algorithms and data plotting software. However, also these did not, and today still do not, support the optimal quantization of possibly complex coefficients.

1.5.2 Possibilities for specifying filters

The specification of a filter-design tool generally comprises multiple aspects, like the specifications that a filter should satisfy, the user interface, and the output data format. Here the focus is on the possibility of specifying filters that can be designed using DESFIL. The design process will eventually result in the impulse response of a symmetric FIR filter that satisfies the specifications. These specifications basically consist of 2 interdependent parts viz.: the time and frequency domains. Next these parts of the specification will be described and discussed briefly. In addition, the option to design a filter in a cascade connection, and the possibility of determining an optimal scale factor are discussed.

Time domain

The dominant part of the specification of the time domain consists of the direct specification of the impulse response itself, namely: *length*, *real or complex valued* and *type of symmetry*. These three parameters determine the number of degrees of freedom in the design process, and the existence of structural transmission zeros, for instance.
In addition, some indirect specifications can be given. For designing cascades of multirate filters or Interpolated FIR (IFIR) filters [27] [98], *comb* filters are essential. Also M*-band* filters [133] have, like the comb filters, specific zero-valued coefficients which imply particular frequency-domain relations. A special part of the specification is the level of *coefficient quantization*: what is the size of the quantization step. Finally, for example, in video processing filters, it is important to control the *step response*.

Frequency domain

The *sampling frequency* is used in the many specifications of the frequency-domain, like the desired gains or attenuations in particular bands, as a reference only. A band may be a *passband* or a *stopband*, for instance. Opposite to these bands where the ideal gain is constant, the gain can be set to *vary linearly*. A special specification applies to the *Nyquist edge*. Such an edge in particular fits in the transition between a passband and a stopband, and is within given tolerances point symmetric. When a filter has multiple passbands the *sign of the gain* may be chosen per band, which can result in more efficient filters [23].

Cascades of filters

Often, a filter under construction will be used in a cascade connection with known filters. In such a case, the specifications can relate to the filter under construction as well as to the complete cascade of filters, depending on the application at hand. A typical case is the design of a cascade of decimating filters. As an example, first the filter $H(z)$ with decimation factor 2 is designed. To obtain a total decimation factor of 6, next the second filter $G(z)$ is designed with decimation factor 3. The used time domain specification relates to $G(z)$: e.g., length and symmetry. However, in the frequency domain, the specification relates to the cascade connection $H(z)G(z^2)$ and is such that this cascade connection is suited for the total decimation factor of 6. A similar approach can be used for interpolating filters. If the step response of such a cascade is relevant, this part of the time domain specification relates to the cascade and not only to the filter under construction $G(z)$.

Scaling

In many applications it is not important what the exact passband gain is. More important may be the ratio between the passband gain and the maximum stopband ripple. In that case not only $H(z)$ can define the optimal filter, but also the scaled filter $sH(z)$ with scale factor $s \in \mathbb{R}$. For filters with unquantized coefficients this is a trivial approach. However if the filter coefficients are being quantized, this scaling introduces freedom that leads to more efficient solutions in many cases.

1.5.3 Special versions

In general, all possible specifications can be combined. However, the possibility to independently provide time and frequency domain specifications may result in inconsistencies. As a consequence, the resulting filter cannot satisfy all specifications. Some inconsistencies are easily detected during input, whereas others are more difficult to find. It is up to the user to ensure that meaningfull input is provided.
A few users had special filter design constraints that could not easily be integrated with existing features. For these users, special versions of DESFIL have been devised, which provide the special features, but not all of the standard ones. The first case is the design of Variable Phase Delay (VPD) filters [29] [101] [102] [103] [104]. VPD filters typically consist of 3 (simultaneously designed) parallel branches, where each branch is a cascade connection of a known filter and the filter under construction. The second case [62] is the design of a real multirate filter. In general the polyphase components of this filter have a non-linear phase response. The additional requirement in this version is that the phase non-linearity of the polyphase components is controllable during design [28].

1.6 Two-phase approach: Design & Evaluate

Powerful general-purpose optimization tools in principle allow the design of a filter satisfying all requirements, like time and frequency-domain specifications, and efficiency. For common specifications, the design of linear-phase real and complex FIR filters with unquantized coefficients can be formulated as a set of constraints that are linear in the filter coefficients. A technique perfectly suited for this class of problems is Linear Programming, or LP (Section 1.6.1). The requirement to design a filter with quantized coefficients can be implemented using the Mixed Integer Linear Programming or MILP technique. By applying LP in combination with the Branch and Bound or B&B method (Section 1.6.2), MILP can design a filter with quantized coefficients that meets the specifications. Alternatively, MILP can generate *all* filters with quantized coefficients that meet the specifications.

In DESFIL the two-phase approach is as follows. First in the design phase the program DESIGN generates a list of all filters with quantized coefficients, using MILP, where each filter in the list meets the specifications.
In the second phase (evaluation), every filter in the list is inspected and analyzed in terms of costs, and the filter with the minimal costs is selected. Different definitions for costs may be used, depending on the application, and, as a consequence different optimal filters can result from the same list of filters. A typical case is that of a filter of transversal structure, and where the coefficients are represented as Canonical Signed Digit (CSD), for minimal arithmetic costs. The DESFIL program to evaluate all filters from the list in this way is called EVALT (EVALuate Transversal). An alternative approach is that the polynomials with integer coefficients, that describe the filters with quantized coefficients, are factorized over the integers. Unlike the factorization over the real numbers, this factorization over the integers is not always possible. In fact, for each filter from the list, all possible cascade connections of smaller transversal filters are evaluated in terms of costs. The DESFIL program to evaluate all filters from the list in this way is called EVALC (EVALuate Cascades).
This two-phase Design & Evaluate approach is special in the sense that no matter what filter is selected in the evaluation phase, it meets the specifications. In addition to evaluating the transversal filters or the cascades of transversal filters, alternative structures can be imposed and the related optimal filter can be selected. In the next part, some background information is presented on LP, B&B and CSDs.

1.6.1 Linear Programming (LP)

LP is the optimization of a linear cost function, subject to linear inequality constraints. Probably the first of many papers that describe the application of LP for the design of digital filters is from Cavin in 1969 [34]. Often, this LP is used to deal with frequency domain specifications, but it is one of a few methods that can incorporate time domain specifications as well, like the step response [106] [115].
Besides the linear-phase FIR filters, non-linear-phase FIR filters [76] [130], complex filters [31], or 2-D filters [32] [58] can be designed with LP techniques. In some cases, the design problem is transformed to enable filters of a higher order to be designed, or to improve the efficiency of the design algorithm itself. The disadvantage of such a transform may be that the B&B method to obtain quantized coefficients cannot be applied. Like FIR-filter designs that fit directly to LP techniques, IIR filters can be designed by applying LP iteratively [37] [38] [131].
In 1992, the year that the development of DESFIL started, the program METEOR was presented [128]. METEOR is, like DESIGN, LP based, but is not dealing with coefficient quantization or complex-valued coefficients. Much attention is directed to improving the arithmetic efficiency of the design method.

1.6.2 Branch and Bound (B&B)

In principle, the solutions obtained from an LP optimization are non-quantized. The B&B method can be used in conjunction with LP to generate integer or quantized solutions, if these exist. Since part of a solution can be integer and part non-integer, this approach is called Mixed Integer Linear Programming or MILP. The LP problem is split (Branched) into 2 new LP problems, each with an additional constraint (Bound) on the allowed values of one of the variables. This process is repeated until the specified variables are integer or until it is clear that no solution exists.

In 1979 Lim [84] shows how to design linear-phase filters with coefficients that are powers of 2, using B&B, and in 1980 Kodek [75] uses B&B to design FIR filters with quantized coefficients. The large computational complexity of this design method is mentioned, and many subsequent papers pay attention to the possible reduction of this complexity [68] [97] for 1-D and [35] for 2-D filters. As an example, [40] considers similarities between the several sub problems as produced by B&B, to reduce the number of LP constraints.

Since an LP problem results in 2 new LP problems, it has to be decided which to put on stack, and which to continue with. Depending on the application, the depth-first search or the breadth-first search strategy [125] can be used.

When the filter coefficients have to be powers of two, a special version of B&B is described in [126]. If the filter coefficients have to be represented as CSD with a limited number of additions or subtractions, special versions of B&B [2] [107] can be used. Here, the values are quantized without an intermediate conversion to the CSD notation. A complicating factor is that the maximum numbers of non-zero elements in the coefficients have to be specified a priori. A special application of the B&B method is found in the design of sparse or thinned filters. In these filters some coefficients are set to zero, so saving on multiplications, whereas the other coefficients are not quantized [127].

1.6.3 Canonical Signed Digits (CSDs)

In 1960, Reitwiesner [120] introduced the CSDs that require the minimal number of nonzeros (1 and -1) to represent an integer. These CSDs directly lead to an implementation of a coefficient with the minimal number of adders and subtractors. Especially for digital filters, the CSD representation of the coefficients can reduce the arithmetic complexity significantly.

The coefficients that can be realized as CSD with a limited number of additions or subtractions are distributed non-uniformly. In [77], filter specifications are adapted in such a way that the resulting infinite precision coefficients can be mapped onto the allowable CSD values, with minimal error. An other approach is to start with expensive CSDs and subsequently reduce the cost per coefficient while preserving the original specification as much as possible [57]. The approach followed in DESFIL, i.e., a designed filter meets the specification and subsequent steps do not violate the original specification, but are used to reduce the costs only, is not found.

1.7 Research questions

After several decades, the field of filter design is still very challenging. This thesis will focus on three main questions that result from topical research on designing symmetric and efficient complex FIR filters, as will be explained next.

The symmetric complex filters with possibly quantized coefficients as designed with DESFIL, for instance, are Hermitian-symmetric filters. Use of a real scale factor, $s \in \mathbb{R}$, may be beneficial in reducing implementation costs, as already supported by DESFIL (Section 1.5.2). In principle, the scale factor may be complex, $s \in \mathbb{C}$, resulting in generalized-Hermitian-symmetric filters.

- Is it relevant to design generalized-Hermitian-symmetric filters?

- What structures implement generalized-Hermitian-symmetric filters?

A popular application of FIR filters is as decimating or interpolating filter with integer or rational decimation or interpolation factors. The polyphase decomposition and the related polyphase structures are very powerful means to reduce the costs of such multirate filters. However, symmetry present in linear-phase filters may be destroyed by the polyphase decomposition and hence can no longer be exploited as a second means to reduce costs.

- Is it possible to restore the symmetry in polyphase filter structures?

These three questions will be treated extensively in the main part of this thesis.

1.8 Outline of this thesis

This thesis is organized around the three research questions as follows.

- Is it relevant to design generalized-Hermitian-symmetric filters?

This question is addressed in Chapter 2 and Chapter 3.
Chapter 2: **Symmetric filters**, extends the classical definition of Hermitian symmetry to a more general definition that is also applicable to complex filters, generalized-Hermitian or (σ, μ)-symmetry, where σ is the shape of symmetry and μ the center of symmetry, with $|\sigma| = 1$, $\sigma \in \mathbb{C}$ and $\mu \in \mathbb{Z}/2$. Next to the (σ, μ)-symmetry, the (σ, μ)-mirroring operator is defined. Both definitions enable a unified treatment of even- and odd-length real and complex filters. Among other interesting properties, the transformation of mirrored filters into symmetric filters is discussed extensively, since it serves as a basis for the restoration of symmetry in polyphase structures in Chapter 5. The focus in this chapter is on (σ, μ)-symmetric filters with finite precision coefficients. For these filters, new theorems and a procedure are presented on the reduction of (σ, μ)-symmetric FIR filters to $(1, \mu)$- or (j, μ)-symmetric filters. To show the possible savings in arithmetic costs by applying the reduction procedure, an example is discussed in detail.
Chapter 3: **First- and second-order filters**, shows that special instances of generalized-Hermitian symmetry, and specifically (j^k, μ)-symmetry, are interesting. Depending on the given specification, (j, μ)-symmetric complex filters may be beneficial over the $(1, \mu)$-symmetric complex filters.

- What structures implement generalized-Hermitian-symmetric filters?

This question is addressed in Chapter 4.
Chapter 4: **Transversal and complex structures**, shows how (σ, μ)-symmetry may appear in the transversal structure and how it can be exploited to realize structures that are more efficient in terms of computational costs. The fact that two filters have inputs or outputs in common, as typically occurs in the polyphase structure (Chapter 5), can be exploited too. Various alternatives to decompose complex filters or coefficients into their individual real and imaginary parts are discussed and compared in detail. Also, new structures for efficiently combining conjugate coefficients will be presented. Finally a detailed comparison of computational costs of transversal filters is presented.

- Is it possible to restore the symmetry in polyphase filter structures?

This question is addressed in Chapter 5.
Chapter 5: **Polyphase structures**, elaborates on the concept of the polyphase decomposition and the closely related polyphase structure, to obtain efficient implementations of interpolating and decimating filters with integer or rational interpolation or decimation factors. In particular, the restoration of symmetry in polyphase structures is discussed. A new theorem and a related procedure on the restoration of symmetry are presented in detail, including an example. Results from Chapter 2 and Chapter 4 are used in this chapter.

Chapter 6: **Conclusions**, presents the main results from this thesis, and will also list some interesting topics for future research.

To make this thesis to a great extent self-supporting, a variety of appendices is added to serve the discussions and analyses in the main part of this thesis.
Appendix A: **Some common identities**, presents a collection of identities for multirate and complex systems, including their proofs. It supports many of the previous chapters.
Appendix B: **Introduction to pipelining**, relates in particular to Chapter 4, that shows structures that deal differently with respect to pipelining.
Appendix C: **Introduction to analog polyphase filters**, in principle does not support any of the chapters. It is only because of the term *polyphase filters* that relates to Chapter 5, and the term *analog polyphase* that relates to complex filters.
Appendix D: **Introduction to Euclid's algorithm**, is needed in the proofs and procedure as discussed in Chapter 5 and Appendix A.
Appendix E: **Alternatives for coefficients**, presents alternatives for the Canonical Signed Digits (CSDs) that require few additions or subtractions. These alternative constructions apply to both the integers and the complex integers.
Appendix F: **Complex-base numbers: introduction and evaluation**, discusses a known alternative representation for the complex numbers. In addition, this alternative is evaluated with respect to implementation costs.
Appendix G: **Introduction to complex primes**, describes how to test whether a complex integer is prime or not. Also, a procedure for the factorization of a complex number in complex primes is shown. This appendix mainly supports Appendix E.

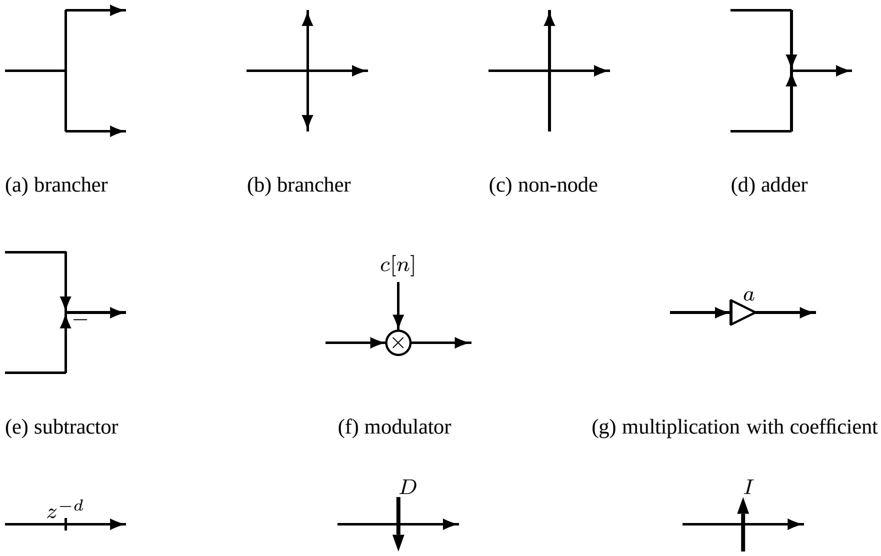
1.9 Notation

Throughout this thesis some special notational conventions and definitions are used. These will be explained here first.

1.9.1 System function

The expression $\langle h[0], \cdots, h[i]\rangle$ in case $i > 0$, or $\langle h[0]\rangle$ in case $i = 0$, denotes the causal system function or filter $H(z) = \sum_{n=0}^{i} h[n]z^{-n}$.

1.9.2 Schemes



(a) brancher (b) brancher (c) non-node (d) adder

(e) subtractor (f) modulator (g) multiplication with coefficient

(h) delay (i) Sampling Rate Decreaser (SRD) (j) Sampling Rate Increaser (SRI)

Figure 1.1: *Frequently used elements in the schemes.*

A number of elements is used frequently in this thesis to construct schemes. In a *brancher*, Figure 1.1(a) and Figure 1.1(b), the input signal is sent to multiple destinations, and in a *non-node*, Figure 1.1(c), two signals pass without any interaction. An *adder*, Figure 1.1(d), and *subtractor*, Figure 1.1(e), produce signals that are respectively the sum and difference of the input signals. The subtrahend of the subtractor is identified with the minus-sign. The complexity of a filter is often expressed in the number of adders or additions, where the subtractors or subtractions are counted as adders or additions. This is possible since the costs of an adder and a subtractor, or an addition and a subtraction, are practically equal. A *modulator*, Figure 1.1(f), multiplies an input signal together with

a carrier, $c[n]$, that can be real, e.g. $c[n] = \sin(\theta_c n)$, or complex, e.g. $c[n] = \mathrm{e}^{\mathrm{j}\theta_c n}$. When an input signal is multiplied by a constant, this constant is called the *coefficient*, Figure 1.1(g). In many cases a multiplication with a coefficient is called a coefficient for short. A signal can be *delayed* over d samples, Figure 1.1(h), where $d \in \mathbb{Z}$. For $d < 0$, the signal is advanced over $-d$ samples. The elements for changing the sampling rate, the *Sampling Rate Decreaser* or SRD, Figure 1.1(i), and the *Sampling Rate Increaser* or SRI, Figure 1.1(j), are described in Section A.1.
Besides these elements, some schemes contain rectangular boxes in which a particular function is indicated, like $H(z)$ for a complete filter in Figure 2.1. Inputs and outputs may assumed to be complex unless indicated otherwise, like in Figure 2.3.

1.9.3 Syntax

A traditional way to present signal processing systems is by means of drawings. Besides this method also an alternative description is used in this thesis depending on what is most convenient in a particular case. The semicolon (;) is used as *connector* and denotes that in $A; B$ the output of A is connected to the input of B. Although $A; B$ and $B; A$ are two different systems their behaviour may be identical, for instance when both A and B are single input, single output, linear and time-invariant. Note that if A and B can be described by the matrices $\mathbf{A}$ and $\mathbf{B}$ respectively, the behaviour of $A; B$ is equal to the behaviour of the matrix product $\mathbf{BA}$. The use of the semicolon as connector in this thesis, resembles its application in computer programming languages as separator or terminator.

Example 1.1. Consider the following examples:

- The decimating filter, filter $H(z)$ followed by the SRD with factor 3, can be described with $H(z); \downarrow 3$.
- The interpolating filter, filter $H(z)$ preceeded by the SRI with factor I, can be described with $\uparrow I; H(z)$.
- Using the first noble identity, Lemma A.3, gives the following equality: $H(z); \downarrow 2; G(z); \downarrow 3 = H(z); G(z^2); \downarrow 6$.

This syntax is easily extendable to matrices for describing multiple input and output schemes.
End of example

1.9.4 Sets of scalars

Next to the standard sets of scalars $\mathbb{N}$, $\mathbb{N}^+$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$, some special sets like the scaled integers, the complex integers, the scaled complex integers, and the complex rationals, are used abundantly in thesis. Their definitions follow here.

Definition 1.1 (Scaled integers)**.** *The set of scaled integers:*

$$\mathbb{Z}/2^i \triangleq \{a2^{-i} | a \in \mathbb{Z}\},$$

with $i \in \mathbb{Z}$.

Definition 1.2 (Complex integers)**.** *The set of complex integers, also called the Gaussian integers:*

$$\mathbb{C}_{\mathbb{Z}} \triangleq \{a_{\mathrm{r}} + \mathrm{j}a_{\mathrm{i}} | a_{\mathrm{r}}, a_{\mathrm{i}} \in \mathbb{Z}\}.$$

Definition 1.3 (Scaled complex integers)**.** *The set of scaled complex integers:*

$$\mathbb{C}_{\mathbb{Z}/2^i} \triangleq \{a_{\mathrm{r}} + \mathrm{j}a_{\mathrm{i}} | a_{\mathrm{r}}, a_{\mathrm{i}} \in \mathbb{Z}/2^i\},$$

with $i \in \mathbb{Z}$*.*

Definition 1.4 (Complex rationals)**.** *The set of quotients of complex integers:*

$$\mathbb{C}_{\mathbb{Z}}/\mathbb{C}_{\mathbb{Z}} \triangleq \{\tfrac{a}{b} | a, b \in \mathbb{C}_{\mathbb{Z}} \wedge b \neq 0\}.$$

The z-transform is a powerful way to describe finite-length filters in terms of their impulse response. To define the set from which the FIR filter coefficients are selected, the following notation is used.

Definition 1.5. *The set of all* z*-transforms* $H(z) = \sum_n h[n]z^{-n}$ *with the FIR filter coefficients* $h[n] \in \mathbb{C}$*, is denoted as* $\mathbb{C}(z)$*, so:*

$$H(z) \in \mathbb{C}(z).$$

Similarly for $\mathbb{R}(z)$*,* $\mathbb{Z}(z)$*,* $\mathbb{Z}/2^i(z)$*,* $\mathbb{C}_{\mathbb{Z}}(z)$ *and* $\mathbb{C}_{\mathbb{Z}/2^i}(z)$*.*

When all coefficients $h[n]$ are conjugated, the related z-transform is denoted as $H^{\overline{*}}(z)$.

Definition 1.6. *Let* $H(z) = \sum_n h[n]z^{-n}$ *with* $h[n] \in \mathbb{C}$*, then:*

$$H^{\overline{*}}(z) \triangleq \sum_n h^*[n]z^{-n}.$$

Note that this definition differs from conjugation of the z-transform denoted as $H^*(z)$ which is $(H(z))^*$.

1.9.5 Remainder of integer division

In some of the proofs and procedures, the remainder of an integer division is needed. In this section the notation is introduced, and also for the complex integers the remainder is defined. Any integer x can be written as $x = qM + r$, where q denotes the quotient, and r denotes the remainder of the integer division.

Definition 1.7. *Any* $x \in \mathbb{Z}$ *can be written as:*

$$x = qM + r,$$

where $x \in \mathbb{Z}$*,* $M \in \mathbb{N}^+$ *and* $r \in \mathbb{N}$ *with* $r < M$*. Now the remainder is:*

$$x|_M \triangleq r.$$

For the complex integers the remainder is determined for the real and imaginary parts individually.

Definition 1.8. *For $x \in \mathbb{C}_\mathbb{Z}$ and $M \in \mathbb{N}^+$, the remainder is denoted as $x|_M$ and applied to the individual parts, like:*

$$x|_M \triangleq x_\mathrm{r}|_M + \mathrm{j}(x_\mathrm{i}|_M).$$

If integer x is *even* it means that x is a multiple of 2. If a complex integer x is *even* it means that both x_r and x_i are a multiple of 2. In case of a complex-integer filter $H(z) \in \mathbb{C}_\mathbb{Z}(z)$ the remainder applies to the individual coefficients.

Definition 1.9. *Let $H(z) = \sum_n h[n]z^{-n}$ with $h[n] \in \mathbb{C}_\mathbb{Z}$, then:*

$$H|_M(z) \triangleq \sum_n h[n]|_M z^{-n}.$$

Some illustrative examples for the remainder of complex integers are presented next.

Example 1.2. For any $x \in \mathbb{C}_\mathbb{Z}$:

$$\begin{aligned} (3-\mathrm{j}7)|_2 &= 1+\mathrm{j} & \quad (3+\mathrm{j}7)|_5 &= (3-\mathrm{j}3)|_5 \\ (-x)|_2 &= x|_2 & (2x)|_2 &= 0 \\ x^*|_2 &= x|_2 & \frac{1}{\mathrm{j}}|_2 &= \mathrm{j} \end{aligned}$$

End of example

1.9.6 Norms

To describe two important parameters of filters, the range of allowed coefficient values and the maximum possible value of the output signal, two norms are used. The required definition differs from the traditionally one where the Lp-norm of the finite-length vector $\mathbf{a}$ is defined as:

$$\|\mathbf{a}\|_p \triangleq \left(\sum_n |\mathbf{a}_n|^p\right)^{\frac{1}{p}},$$

with the interesting cases:

$$\|\mathbf{a}\|_1 = \sum_n |\mathbf{a}_n| \text{ and } \|\mathbf{a}\|_\infty = \max_n |\mathbf{a}_n|.$$

In case $\mathbf{a}_n \in \mathbb{C}$, this definition considers the modulus of $\mathbf{a}_n$. However, in this thesis, the norms are defined over the individual real and imaginary parts of the coefficients. This can be interpreted as if a complex polynomial is represented as a $2 \times m$ matrix of real numbers, and subsequently considering the traditional Lp-norm of that real matrix. To distinguish from the Lp-norm, that is not used in this thesis, this alternative norm is called the p-norm and is defined next.

Definition 1.10 (p-norm)**.** *The p-norm of $A(z) \in \mathbb{C}(z)$ is:*

$$\|A(z)\|_p \triangleq \left(\sum_n |a_\mathrm{r}[n]|^p + |a_\mathrm{i}[n]|^p \right)^{\frac{1}{p}}.$$

Note that this definition satisfies all standard conditions for norms over the real vector space. Two interesting cases are the 1-norm of $A(z) \in \mathbb{C}(z)$:

$$\|A(z)\|_1 = \sum_n |a_\mathrm{r}[n]| + |a_\mathrm{i}[n]|,$$

and the ∞-norm of $A(z) \in \mathbb{C}(z)$:

$$\|A(z)\|_\infty = \max_n \left(|a_\mathrm{r}[n]|, |a_\mathrm{i}[n]| \right).$$

In some of the analyses the ∞-norms of the filters, and hence the allowed coefficient values, are limited to a given value. This value, the *coefficient range*, Ξ, will be used such that $\|H(z)\|_\infty \leq \Xi$. Also, in Appendix A, a number of interesting inequalities regarding the norms are presented.

Chapter 2

Symmetric filters

Symmetric filters are very popular, because their phase response is linear, which is beneficial in many applications. Design methods for these filters have been available since the early days of digital signal processing [111] [115]. Many alternative design methods, e.g., [128] [139], and alternative structures, e.g., [85], have been under development ever since.

In case a filter should be implemented in dedicated hardware, for instance when the sampling frequency is high, the quantization of the filter coefficients is an efficient way to reduce the costs of the implementation. Many methods can be found in the literature [36] [86] [122].

Application of complex-valued coefficients, instead of real-valued coefficients, enable frequency responses to be different for positive and negative frequencies. Design methods for such filters can be found in, e.g., [33] [49] [99]. In [135] it is described how complex filters can be applied to efficient multirate filtering. For causal filters with filter length L, $0 \leq n < L$, and real-valued coefficients, the symmetric case $h[n] = h[L-1-n]$, e.g., $\langle a, b, c, b, a \rangle$, or the anti-symmetric case $h[n] = -h[L-1-n]$, e.g., $\langle a, b, c, -c, -b, -a \rangle$, guarantee a linear-phase frequency response. Commonly for linear-phase filters with complex-valued coefficients, the real part is taken symmetric and the imaginary part is taken anti-symmetric, i.e., $h[n] = h^*[L-1-n]$, e.g., $\langle a, b, c, c^*, b^*, a^* \rangle$. This type of symmetry is referred to as *conjugate symmetry* [64] or *Hermitian symmetry* [133]. Also in non-filter applications these definitions for symmetry are used [96].

The linear-phase frequency response of the symmetric or anti-symmetric real filters, or the Hermitian-symmetric complex filters, relies on simple relations between coefficients. Coefficients are either equal or opposite in the real case, or conjugate (equal real part and opposite imaginary part) in the complex case. To express this special property, the linear phase is said to be simple-structurally guaranteed. These simple relations reduce implementation costs of a linear-phase filter for two reasons: i) the simple relations are invariant under coefficient quantization, and ii) two equal or opposite coefficients may be combined to a single coefficient using the distributive property, therebye saving in the number of multiplications.

In [64] [65] [123] [133] and [135] it is recognized that if filter $H(z)$ has a linear phase, also the filter $aH(z)$, with $a \in \mathbb{C}$, has a linear phase. In [133] the related symmetry is called *generalized-Hermitian symmetry*. For arbitrary values of a, the real and imaginary part of $aH(z)$ are neither symmetric nor anti-symmetric, and therefore the linear-phase property is not simple-structurally guaranteed. When the coefficients are quantized the phase response will be non-linear, and it is difficult to combine coefficients to reduce cost. However, for special values of a, the resulting filter $aH(z)$ will simple-structurally guarantee the linear-phase response.

Central in this chapter are three theorems about *reducing* generalized-Hermitian-symmetric filters to Hermitian-symmetric filters. This chapter starts with a definition for symmetry that is discussed in great detail in Section 2.1. This (σ, μ)-symmetry, with *shape of symmetry* σ, $\sigma \in \mathbb{C}$, $|\sigma| = 1$, and the *center of symmetry* μ, $\mu \in \mathbb{Z}/2$, is applicable to filters with real-valued and complex-valued coefficients, and treats even- and odd-length filters in a unified manner. In close relation to symmetry, the concept of mirroring is introduced in Section 2.2, and for two mutually mirrored filters with a common input or output it is shown in Section 2.3 how they can be replaced by symmetric filters. This result is important for the restoration of symmetry in polyphase structures that will be treated in Chapter 5. Section 2.4 describes the consequences in the frequency domain when a filter is symmetric. The traditional list of 4 types of symmetric real filters is extended in Section 2.5 with 5 types of (σ, μ)-symmetric complex filters. The new type 5 through 8 filters exhibit simple relations between the coefficients, whereas the type 9 filter does not. It is shown how the coefficients may be quantized to preserve symmetry in Section 2.6, and how the shape of symmetry may be changed in Section 2.7. In Section 2.8 the main theorem is presented about the possibility of reducing type 9 filters to type 5, 6, 7 or 8 filters, even in case the individual real and imaginary parts of the filter coefficients are integer. Application of this reduction theorem is illustrated extensively in Section 2.9 by means of a typical example, also showing the possibility to save in arithmetic costs. Section 2.10 will show that the type 6, 7, 8 and 9 integer filters, can be designed by first designing a type 5 filter. Finally, in Section 2.11, it is shown that the structural transmission zeros known from the real symmetric filters do not exist in complex symmetric filters.

2.1 Symmetry

To support the discussion about many types of symmetric filters, where the filters may be non-causal and complex, filter symmetry is defined in Definition 2.1. In this definition neither the filter length nor the causality of the impulse response are relevant. These aspects are covered by the *center of symmetry* μ, with $\mu \in \mathbb{Z}/2$. Also, in complex filters there are more possibilities than equal or opposite, symmetric or anti-symmetric respectively. This aspect is covered by *shape of symmetry* σ, with $\sigma \in \mathbb{C}$ and $|\sigma| = 1$.

The center of symmetry, $\mu \in \mathbb{Z}/2$, is either an integer value, $\mu \in \mathbb{Z}$, or an integer value plus a half, $\mu \in \mathbb{Z} + \frac{1}{2}$, so distinguishing implicitly between the sets of odd- and even-length filters respectively. In the causal case where the length equals L and the indices n range from 0 through $L-1$, the center of symmetry is: $\mu = \frac{L-1}{2}$. The (σ, μ)-symmetry for complex filters $H(z) \in \mathbb{C}(z)$ is defined next.

Definition 2.1 (Symmetry). *Filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric iff:*

$$H(z) = \sigma z^{-2\mu} \overline{H^*}(z^{-1}),$$

with $\sigma \in \mathbb{C}$, $|\sigma| = 1$ and $\mu \in \mathbb{Z}/2$.

Throughout this thesis it may be assumed that $\sigma \in \mathbb{C}$, $|\sigma| = 1$ and $\mu \in \mathbb{Z}/2$, unless stated otherwise. The definition for (σ, μ)-symmetry is suitable in the proofs of the various lemmas that will follow, whereas the formulation in the next lemma, in terms of impulse response h, is more intuitive.

Lemma 2.1. *Filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric iff:*

$$h[n] = \sigma h^*[2\mu - n] \text{ for all } n.$$

Proof. For the first part of the proof the impulse response is transformed into the z-domain as follows:

$$\begin{aligned} H(z) &= \sum_n h[n] z^{-n} \\ &= \sum_n \sigma h^*[2\mu - n] z^{-n} \\ &= \sigma \sum_m h^*[m] z^{m-2\mu} \\ &= \sigma z^{-2\mu} \sum_m h^*[m] (z^{-1})^{-m} \\ &= \sigma z^{-2\mu} \overline{H^*}(z^{-1}), \end{aligned}$$

with $m = 2\mu - n$. For the second part of the proof the inverse of the transform is used, along the same line but in the reverse order. □

Only in case $\mu \in \mathbb{Z}$ (odd-length filter) the central coefficient $h[\mu]$ exists and satisfies a special condition.

Lemma 2.2. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric with $\mu \in \mathbb{Z}$, then $h[\mu] = c\sqrt{\sigma}$ with $c \in \mathbb{R}$.*

Proof. By Lemma 2.1, $h[\mu] = \sigma h^*[2\mu - \mu] = \sigma h^*[\mu]$, which implies $\mathcal{P}(h[\mu]) = \mathcal{P}(\sigma) - \mathcal{P}(h[\mu])$, or $\mathcal{P}(h[\mu]) = \frac{\mathcal{P}(\sigma)}{2}$. Therefore $h[\mu] = c\sqrt{\sigma}$ with $c \in \mathbb{R}$. □

From the definition for symmetry a number of interesting lemmas can be derived. These are presented in Lemma 2.3 through Lemma 2.6. First, the cascade connection, or product, of two symmetric filters is also symmetric.

Lemma 2.3. *If filter $G(z) \in \mathbb{C}(z)$ is (σ_G, μ_G)-symmetric and filter $H(z) \in \mathbb{C}(z)$ is (σ_H, μ_H)-symmetric, then the cascade connection:*

$$G(z)H(z) \text{ is } (\sigma_G \sigma_H, \mu_G + \mu_H)\text{-symmetric}.$$

Proof. Define $F(z) = G(z)H(z)$, then:

$$\begin{aligned} F(z) &= \sigma_G z^{-2\mu_G} G^{\overline{*}}(z^{-1}) \sigma_H z^{-2\mu_H} H^{\overline{*}}(z^{-1}) \\ &= \sigma_G \sigma_H z^{-2\mu_G} z^{-2\mu_H} G^{\overline{*}}(z^{-1}) H^{\overline{*}}(z^{-1}) \\ &= \sigma_G \sigma_H z^{-2(\mu_G+\mu_H)} F^{\overline{*}}(z^{-1}). \end{aligned}$$ □

The parallel connection, or sum, of two filters that have identical shapes and centers of symmetry, is also symmetric.

Lemma 2.4. *If the filters $G(z), H(z) \in \mathbb{C}(z)$ are (σ, μ)-symmetric, then the parallel connection:*

$$G(z) + H(z) \textit{ is } (\sigma, \mu)\textit{-symmetric}.$$

Proof. Define $F(z) = G(z) + H(z)$, so:

$$\begin{aligned} F(z) &= \sigma z^{-2\mu} G^{\overline{*}}(z^{-1}) + \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}) \\ &= \sigma z^{-2\mu} \big(G^{\overline{*}}(z^{-1}) + H^{\overline{*}}(z^{-1}) \big) \\ &= \sigma z^{-2\mu} F^{\overline{*}}(z^{-1}). \end{aligned}$$ □

Given the symmetric filter $H(z)$, then the scaled version $aH(z)$ and also the delayed version $z^{-n}H(z)$ are symmetric. The frequency response of a filter $H(z)$ can be shifted over θ_c by modulating its coefficients $h[n]$ with the complex sequence $\mathrm{e}^{\mathrm{j}\theta_c n}$, so obtaining the coefficients $h[n]\mathrm{e}^{\mathrm{j}\theta_c n}$. If the filter $H(z)$ is symmetric, the resulting modulated filter is also symmetric. These properties are presented in the following lemma.

Lemma 2.5. *If complex filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric, then for $a \in \mathbb{C}$ and $n \in \mathbb{Z}$, the filter:*

$$a z^{-n} H(z\mathrm{e}^{-\mathrm{j}\theta_c}) \textit{ is } \Big(\sigma \frac{a}{a^*} \mathrm{e}^{2\mathrm{j}\theta_c \mu}, (\mu + n)\Big)\textit{-symmetric}.$$

Proof. Repeated application of Definition 2.1, and some calculus gives:

$$\begin{aligned} G(z) &= a z^{-n} H(z\mathrm{e}^{-\mathrm{j}\theta_c}) \\ &= a z^{-n} \sigma (z\mathrm{e}^{-\mathrm{j}\theta_c})^{-2\mu} H^{\overline{*}}\big(z^{-1}(\mathrm{e}^{-\mathrm{j}\theta_c})^*\big) \\ &= \frac{a}{a^*} z^{-2n} \sigma z^{-2\mu} \mathrm{e}^{2\mathrm{j}\theta_c\mu} a^* z^n H^{\overline{*}}(z^{-1}\mathrm{e}^{\mathrm{j}\theta_c}) \\ &= \sigma \frac{a}{a^*} \mathrm{e}^{2\mathrm{j}\theta_c\mu} z^{-2(\mu+n)} G^{\overline{*}}(z^{-1}). \end{aligned}$$ □

Conjugation of the coefficients of a symmetric filter results in a symmetric filter.

Lemma 2.6. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric, then conjugation of the coefficients gives:*

$$H^{\overline{*}}(z) \textit{ is } (\sigma^{-1}, \mu)\textit{-symmetric}.$$

Proof. The Definition 2.1 for symmetry is used twice:

$$\begin{aligned}
H(z) &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}), \\
F(z) &= H^{\overline{*}}(z) \\
&= \sigma^{*} z^{-2\mu} H(z^{-1}) \\
&= \sigma^{-1} z^{-2\mu} F^{\overline{*}}(z^{-1}).
\end{aligned}$$

□

For a list of simple filters, Table 2.1, the shape of symmetry σ, and center of symmetry μ, are given assuming $a, b \in \mathbb{C}$, $n \in \mathbb{Z}$ and $|b| = 1$.

$H(z)$	σ	μ
± 1	1	0
$\pm \mathrm{j}$	-1	0
az^{-n}	$\frac{a}{a^*}$	n
$1 \pm bz^{-n}$	$\pm b$	$\frac{n}{2}$
$a \pm a^* z^{-n}$	± 1	$\frac{n}{2}$
$a \pm \mathrm{j} a^* z^{-n}$	$\pm \mathrm{j}$	$\frac{n}{2}$
$a \pm az^{-n}$	$\pm \frac{a}{a^*}$	$\frac{n}{2}$
$a \pm \mathrm{j} az^{-n}$	$\pm \mathrm{j} \frac{a}{a^*}$	$\frac{n}{2}$

Table 2.1: *Some typical symmetric filters.*

In the remainder of this chapter a special instance of Lemma 2.5 is used frequently, viz: the scaled version of a (σ, μ)-symmetric filter can obtain any shape of symmetry depending on the scale factor. Alternatively, a given (σ, μ)-symmetric filter can be factorized into a scale factor and another symmetric filter. The following theorem states that every (σ, μ)-symmetric complex filter is reducible to a $(1, \mu)$-symmetric complex filter, where the filter coefficients are elements of $\mathbb{C}$.

Theorem 2.1 (Reduction over $\mathbb{C}$)**.** *Let $H(z) \in \mathbb{C}(z)$ be a complex (σ, μ)-symmetric filter, then there is a $(1, \mu)$-symmetric complex filter $G(z) \in \mathbb{C}(z)$ and a complex number $a \in \mathbb{C}$, such that $H(z) = aG(z)$. This is called, $H(z)$ is reducible to a type $(1, \mu)$ filter over $\mathbb{C}$.*

Proof. Let $a = \sqrt{\sigma}$ and define $G(z) = \frac{H(z)}{a}$. Application of Lemma 2.5 gives the required result. □

A direct consequence of this theorem is that every (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$, can be designed by first designing a $(1, \mu)$-symmetric complex filter $G(z) \in \mathbb{C}(z)$, and secondly multiplying all coefficients of $G(z)$ with the scale factor $a = \sqrt{\sigma}$. Directly designing the (σ, μ)-symmetric filter does not provide additional solutions. In Section 2.8, Theorem 2.2 shows the surprising result that, also for complex filters with integer coefficients, it is in principle sufficient to design a $(1, \mu)$-symmetric complex filter.

2.2 Mirroring

A real $(1, 0)$-symmetric filter is trivially invariant under reversal of the coefficient order. In case of a real $(-1, 0)$-symmetric filter the same is true except for a minus-sign. In Definition 2.1 the symmetry for complex filters, $H(z) \in \mathbb{C}(z)$, is formulated in terms of the center of symmetry, μ, and the shape of symmetry, σ. Based on the same parameters, the order-reversing or "mirroring"-operator $\mathcal{M}$ is defined.

Definition 2.2 (Mirroring)**.** *For the filter $H(z) \in \mathbb{C}(z)$, the (σ, μ)-mirroring operator $\mathcal{M}_{\sigma,\mu}$ is defined as:*

$$\mathcal{M}_{\sigma,\mu}\big(H(z)\big) \triangleq \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}),$$

with $\sigma \in \mathbb{C}$, $|\sigma| = 1$ and $\mu \in \mathbb{Z}/2$.

The two definitions for symmetry and mirroring are strongly related, see the following lemma.

Lemma 2.7. *Filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric iff:*

$$H(z) = \mathcal{M}_{\sigma,\mu}\big(H(z)\big).$$

Proof. The proof follows directly from Definition 2.1 and Definition 2.2. □

More in general, any mirroring of a symmetric filter results in another symmetric filter.

Lemma 2.8. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric, then the (σ_0, μ_0)-mirrored version:*

$$\mathcal{M}_{\sigma_0,\mu_0}\big(H(z)\big) \textit{ is } \Big(\frac{\sigma_0^2}{\sigma}, 2\mu_0 - \mu\Big)\textit{-symmetric}.$$

Proof. First Definition 2.2 for mirroring, and Definition 2.1 for symmetry are used. Defining $F(z) = \mathcal{M}_{\sigma_0,\mu_0}(H(z))$ gives:

$$\begin{aligned} F(z) &= \sigma_0 z^{-2\mu_0} H^{\overline{*}}(z^{-1}), \\ H(z) &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}), \\ F(z) &= \frac{\sigma_0}{\sigma}\frac{z^{-2\mu_0}}{z^{-2\mu}} H(z), \end{aligned}$$

where $\frac{\sigma_0}{\sigma}\frac{z^{-2\mu_0}}{z^{-2\mu}}$ has $((\frac{\sigma_0}{\sigma})^2, 2(\mu_0 - \mu))$-symmetry. Application of Lemma 2.3 now concludes the proof. □

Application of two general mirroring operations in succession on a filter, is equivalent to a scaling and delay of the original filter.

Lemma 2.9. *For any filter $H(z) \in \mathbb{C}(z)$:*

$$\mathcal{M}_{\sigma_1,\mu_1}\Big(\mathcal{M}_{\sigma_0,\mu_0}\big(H(z)\big)\Big) = \sigma z^{-2\mu} H(z),$$

with $\sigma = \frac{\sigma_1}{\sigma_0}$ and $\mu = \mu_1 - \mu_0$.

Proof. Definition 2.2 is used repeatedly.

$$\begin{aligned}
\mathcal{M}_{\sigma_1,\mu_1}\big(\mathcal{M}_{\sigma_0,\mu_0}\big(H(z)\big)\big) &= \mathcal{M}_{\sigma_1,\mu_1}\big(\sigma_0 z^{-2\mu_0} H^{\overline{*}}(z^{-1})\big) \\
&= \sigma_1 z^{-2\mu_1} \left.\Big(\big(\sigma_0 z^{-2\mu_0} H^{\overline{*}}(z^{-1})\big)^{\overline{*}}\Big)\right|_{z\leftarrow z^{-1}} \\
&= \sigma_1 z^{-2\mu_1} \sigma_0^* z^{2\mu_0} H(z) \\
&= \frac{\sigma_1}{\sigma_0} z^{-2(\mu_1-\mu_0)} H(z). \qquad \square
\end{aligned}$$

The mirroring operation is an involution, i.e., applying two identical mirror operations in succession on a filter, results in the original filter.

Lemma 2.10. *For any filter* $H(z) \in \mathbb{C}(z)$*:*

$$\mathcal{M}_{\sigma,\mu}\big(\mathcal{M}_{\sigma,\mu}\big(H(z)\big)\big) = H(z).$$

Proof. Follows directly from Lemma 2.9. □

Given $z^{-n_0}\mathcal{M}_{\sigma,\mu}(H(z))$, the delayed mirrored version of a filter $H(z)$, can be expressed in mirroring the same filter with a different center of symmetry. Also, mirroring a delayed filter $z^{-n_1}H(z)$, $\mathcal{M}_{\sigma,\mu}(z^{-n_1}H(z))$, can be expressed in mirroring the non-delayed filter with a different center of symmetry. Both properties are described in the following lemma.

Lemma 2.11. *For any filter* $H(z) \in \mathbb{C}(z)$*, mirror operation* $\mathcal{M}_{\sigma,\mu}$ *and* $n_0, n_1 \in \mathbb{Z}$*:*

$$z^{-n_0}\mathcal{M}_{\sigma,\mu}\big(z^{-n_1}H(z)\big) = \mathcal{M}_{\sigma,\mu+\frac{n_0-n_1}{2}}\big(H(z)\big).$$

Proof. By Definition 2.2:

$$\begin{aligned}
z^{-n_0}\mathcal{M}_{\sigma,\mu}\big(z^{-n_1}H(z)\big) &= z^{-n_0}\sigma z^{-2\mu} \left.\Big(\big(z^{-n_1}H(z)\big)^{\overline{*}}\Big)\right|_{z\leftarrow z^{-1}} \\
&= z^{-n_0}\sigma z^{-2\mu} z^{n_1} H^{\overline{*}}(z^{-1}) \\
&= \sigma z^{-2(\mu+\frac{n_0-n_1}{2})} H^{\overline{*}}(z^{-1}) \\
&= \mathcal{M}_{\sigma,\mu+\frac{n_0-n_1}{2}}\big(H(z)\big). \qquad \square
\end{aligned}$$

Similarly, $a\mathcal{M}_{\sigma,\mu}(H(z))$, the scaled mirrored version of a filter $H(z)$, can be expressed in mirroring the same filter with a different shape of symmetry. Also, $\mathcal{M}_{\sigma,\mu}(bH(z))$, the mirrored scaled filter $bH(z)$, can be expressed in mirroring the non-scaled filter with a different shape of symmetry. Both properties are described in the following lemma.

Lemma 2.12. *For any filter* $H(z) \in \mathbb{C}(z)$*, mirror operation* $\mathcal{M}_{\sigma,\mu}$ *and* $a, b \in \mathbb{C}$ *with* $|ab| = 1$*:*

$$a\mathcal{M}_{\sigma,\mu}\big(bH(z)\big) = \mathcal{M}_{ab^*\sigma,\mu}\big(H(z)\big).$$

Proof. By Definition 2.2:

$$\begin{aligned} a\mathcal{M}_{\sigma,\mu}\big(bH(z)\big) &= a\sigma z^{-2\mu} \left(\big(bH(z)\big)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\ &= ab^*\sigma z^{-2\mu} H^{\overline{*}}(z^{-1}) \\ &= \mathcal{M}_{ab^*\sigma,\mu}\big(H(z)\big). \end{aligned}$$ □

Related to Lemma 2.12 is the scaling of a mirrored version of filter $H(z)$, or the mirroring of a scaled filter $H(z)$, without affecting the shape of symmetry of the mirroring operator.

Lemma 2.13. *For any filter $H(z) \in \mathbb{C}(z)$, mirror operation $\mathcal{M}_{\sigma,\mu}$ and $a, b \in \mathbb{C}$:*

$$a\mathcal{M}_{\sigma,\mu}\big(bH(z)\big) = b^*\mathcal{M}_{\sigma,\mu}\big(a^*H(z)\big).$$

Proof. By Definition 2.2:

$$\begin{aligned} a\mathcal{M}_{\sigma,\mu}\big(bH(z)\big) &= a\sigma z^{-2\mu} \left(\big(bH(z)\big)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\ &= b^*\sigma z^{-2\mu} \left(\big(a^*H(z)\big)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\ &= b^*\mathcal{M}_{\sigma,\mu}\big(a^*H(z)\big). \end{aligned}$$ □

Mirroring a cascade connection, or product, of the two filters $G(z)$ and $H(z)$, is identical to the cascade connection of individually mirrored versions of both filters. In this process there is freedom in distributing the shape of symmetry and the center of symmetry over both filters.

Lemma 2.14. *For any pair of filters $G(z), H(z) \in \mathbb{C}(z)$:*

$$\mathcal{M}_{\sigma_G\sigma_H,\mu_G+\mu_H}\big(G(z)H(z)\big) = \mathcal{M}_{\sigma_G,\mu_G}\big(G(z)\big)\mathcal{M}_{\sigma_H,\mu_H}\big(H(z)\big).$$

Proof. By Definition 2.2:

$$\begin{aligned} \mathcal{M}_{\sigma_G\sigma_H,\mu_G+\mu_H}\big(G(z)H(z)\big) &= \sigma_G\sigma_H z^{-2(\mu_G+\mu_H)} \left(\big(G(z)H(z)\big)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\ &= \sigma_G\sigma_H z^{-2(\mu_G+\mu_H)} G^{\overline{*}}(z^{-1})H^{\overline{*}}(z^{-1}) \\ &= \sigma_G z^{-2\mu_G} G^{\overline{*}}(z^{-1})\sigma_H z^{-2\mu_H} H^{\overline{*}}(z^{-1}) \\ &= \mathcal{M}_{\sigma_G,\mu_G}\big(G(z)\big)\mathcal{M}_{\sigma_H,\mu_H}\big(H(z)\big). \end{aligned}$$ □

Mirroring a parallel connection, or sum, of the two filters $G(z)$ and $H(z)$, is identical to the parallel connection of the individually mirrored versions of both filters. Opposite to the cascade connection, now both filters are mirrored with identical parameters.

Lemma 2.15. *For any pair of filters $G(z), H(z) \in \mathbb{C}(z)$:*

$$\mathcal{M}_{\sigma,\mu}\big(G(z) + H(z)\big) = \mathcal{M}_{\sigma,\mu}\big(G(z)\big) + \mathcal{M}_{\sigma,\mu}\big(H(z)\big).$$

Proof. By Definition 2.2:

$$\begin{aligned}
\mathcal{M}_{\sigma,\mu}\big(G(z)+H(z)\big) &= \sigma z^{-2\mu}\ \left(\left(G(z)+H(z)\right)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\
&= \sigma z^{-2\mu}\big(G^{\overline{*}}(z^{-1})+H^{\overline{*}}(z^{-1})\big) \\
&= \sigma z^{-2\mu}G^{\overline{*}}(z^{-1})+\sigma z^{-2\mu}H^{\overline{*}}(z^{-1}) \\
&= \mathcal{M}_{\sigma,\mu}\big(G(z)\big)+\mathcal{M}_{\sigma,\mu}\big(H(z)\big).
\end{aligned}$$

□

The cascade connection of a filter and its mirrored version is symmetric.

Lemma 2.16. *For any filter $H(z) \in \mathbb{C}(z)$ the cascade connection:*

$$H(z)\mathcal{M}_{\sigma,\mu}\big(H(z)\big) \text{ is } (\sigma^2, 2\mu)\text{-symmetric}.$$

Proof. After applying Lemma 2.7 the following equality has to hold.

$$\begin{aligned}
H(z)\mathcal{M}_{\sigma,\mu}\big(H(z)\big) &= \mathcal{M}_{\sigma^2,2\mu}\big(H(z)\mathcal{M}_{\sigma,\mu}\big(H(z)\big)\big), \\
&\Leftrightarrow \\
\sigma z^{-2\mu}H(z)H^{\overline{*}}(z^{-1}) &= \sigma^2 z^{-4\mu}\ \left(\left(\sigma z^{-2\mu}H(z)H^{\overline{*}}(z)\right)^{\overline{*}}\right)\Big|_{z\leftarrow z^{-1}} \\
&= \sigma^2 z^{-4\mu}\sigma^* z^{2\mu}H(z)H^{\overline{*}}(z),
\end{aligned}$$

which is true since $\sigma^* = \frac{1}{\sigma}$. □

In Table 2.2 a list of typical mirroring operations with shape of symmetry, σ, and center of symmetry, μ, are given for $a, b \in \mathbb{C}$ and $n \in \mathbb{Z}$.

Example	
$\mathcal{M}_{1,0}(1)$	$= 1$
$\mathcal{M}_{1,0}(a)$	$= a^*$
$\mathcal{M}_{\mathrm{j},0}(1)$	$= \mathrm{j}$
$\mathcal{M}_{\mathrm{j},0}(a)$	$= \mathrm{j}a^*$
$\mathcal{M}_{1,n}(z^{-n})$	$= z^{-n}$
$\mathcal{M}_{\sigma,\frac{n}{2}}(1)$	$= \sigma z^{-n}$
$\mathcal{M}_{\sigma,0}(a+bz^{-n})$	$= \sigma b^* z^n + \sigma a^*$
$\mathcal{M}_{1,\frac{n}{2}}(a+bz^{-n})$	$= b^* + a^* z^{-n}$

Table 2.2: *Some typical mirroring operations.*

2.3 Mirrored and symmetric pairs of filters

For any filter and its mirrored version, with a common input or output, it is possible to obtain an alternative structure with two symmetric filters, each with an opposite shape of symmetry. This is called the mirrored-pair identity. The inverse procedure is referred to as the symmetric-pair identity. The mirrored-pair identity will be used in combination with complex filters or polyphase structures. In Figure 2.1 and Figure 2.2 the schemes corresponding to these possibilities are presented.

2.3.1 Mirrored-pairs

In the next two lemmas it is shown how mirroring can lead to symmetry.

Lemma 2.17. *For any filter $H(z) \in \mathbb{C}(z)$:*

$$H(z) + \mathcal{M}_{\sigma,\mu}\big(H(z)\big) \text{ is } (\sigma,\mu)\text{-symmetric}.$$

Proof. $G(z)$ is defined as $G(z) = H(z) + \mathcal{M}_{\sigma,\mu}\big(H(z)\big)$, so:

$$\begin{aligned}\mathcal{M}_{\sigma,\mu}\big(G(z)\big) &= \mathcal{M}_{\sigma,\mu}\big(H(z) + \mathcal{M}_{\sigma,\mu}\big(H(z)\big)\big)\\ &= \mathcal{M}_{\sigma,\mu}\big(H(z)\big) + \mathcal{M}_{\sigma,\mu}\big(\mathcal{M}_{\sigma,\mu}\big(H(z)\big)\big)\\ &= \mathcal{M}_{\sigma,\mu}\big(H(z)\big) + H(z)\\ &= G(z).\end{aligned}$$

In this proof Lemma 2.15 and Lemma 2.10 are used. □

Two mutually mirrored filters that have a common input or output, see Figure 2.1, can be replaced by two symmetric filters with opposite shapes of symmetry, and a combination network.

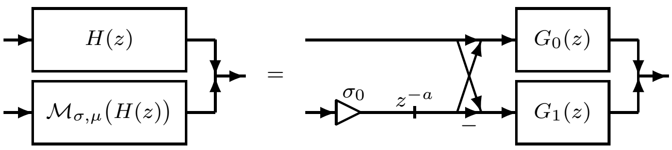


Figure 2.1: *Mirrored-pair identity in case of a common output.*

Lemma 2.18 (Mirrored-pair identity)**.** *Assume filter $H(z) \in \mathbb{C}(z)$ and $a \in \mathbb{Z}$, then for a common input:*

$$\begin{bmatrix} H(z) \\ \mathcal{M}_{\sigma,\mu}\big(H(z)\big)\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a}\end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1\end{bmatrix}\begin{bmatrix} G_0(z) \\ G_1(z)\end{bmatrix},$$

and for a common output:

$$\begin{bmatrix} H(z) & \mathcal{M}_{\sigma,\mu}\big(H(z)\big)\end{bmatrix} = \begin{bmatrix} G_0(z) & G_1(z)\end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1\end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a}\end{bmatrix},$$

with:

$$G_0(z) = \tfrac{1}{2}\big(H(z) + \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}(H(z))\big) \textit{ is } \Big(\frac{\sigma}{\sigma_0}, \mu - \frac{a}{2}\Big)\textit{-symmetric},$$
$$G_1(z) = \tfrac{1}{2}\big(H(z) - \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}(H(z))\big) \textit{ is } \Big(-\frac{\sigma}{\sigma_0}, \mu - \frac{a}{2}\Big)\textit{-symmetric}.$$

Proof. The two filters with a common input can be represented by the vector:

$$\begin{aligned}
\begin{bmatrix} H(z) \\ \mathcal{M}_{\sigma,\mu}\big(H(z)\big) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} H(z) \\ \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \end{bmatrix} \\
&= \tfrac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} H(z) + \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \\ H(z) - \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix}.
\end{aligned}$$

Lemma 2.11, Lemma 2.12 and Lemma 2.17 were used. The second matrix equation from the lemma can be obtained by transposing the first equation. □

2.3.2 Symmetric-pairs

In the next two lemmas it is shown how symmetry can lead to mirroring.

Lemma 2.19. *For the pair of filters $G_0(z), G_1(z) \in \mathbb{C}(z)$, with $G_0(z)$ is (σ,μ)-symmetric and $G_1(z)$ is $(-\sigma,\mu)$-symmetric:*

$$G_0(z) + G_1(z) = \mathcal{M}_{\sigma,\mu}\big(G_0(z) - G_1(z)\big).$$

Proof. By Lemma 2.7, Lemma 2.12 and Lemma 2.15:

$$\begin{aligned}
G_0(z) + G_1(z) &= \mathcal{M}_{\sigma,\mu}\big(G_0(z)\big) + \mathcal{M}_{-\sigma,\mu}\big(G_1(z)\big) \\
&= \mathcal{M}_{\sigma,\mu}\big(G_0(z)\big) + \mathcal{M}_{\sigma,\mu}\big(-G_1(z)\big) \\
&= \mathcal{M}_{\sigma,\mu}\big(G_0(z) - G_1(z)\big).
\end{aligned}$$ □

Two symmetric filters that have opposite shapes of symmetry and a common input or output, see Figure 2.2, can be replaced by two mutually mirrored filters and a combination network.

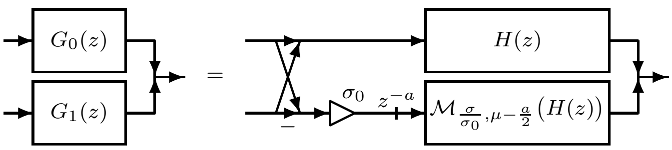


Figure 2.2: *Symmetric-pair identity in case of a common output.*

Lemma 2.20 (Symmetric-pair identity)**.** *Assume filter $G_0(z) \in \mathbb{C}(z)$ is (σ,μ)-symmetric, filter $G_1(z) \in \mathbb{C}(z)$ is $(-\sigma,\mu)$-symmetric and $a \in \mathbb{Z}$, then for a common input:*

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} H(z) \\ \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \end{bmatrix},$$

and for a common output:

$$\begin{bmatrix} G_0(z) & G_1(z) \end{bmatrix} = \begin{bmatrix} H(z) & \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

with:

$$\begin{aligned} H(z) &= \tfrac{1}{2}\big(G_0(z) + G_1(z)\big), \\ \mathcal{M}_{\sigma,\mu}\big(H(z)\big) &= \tfrac{1}{2}\big(G_0(z) - G_1(z)\big). \end{aligned}$$

Proof. The two filters with a common input can be represented with the vector:

$$\begin{aligned} \begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} &= \tfrac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} G_0(z) + G_1(z) \\ G_0(z) - G_1(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} H(z) \\ \mathcal{M}_{\sigma,\mu}\big(H(z)\big) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sigma_0 z^{-a} \end{bmatrix} \begin{bmatrix} H(z) \\ \mathcal{M}_{\frac{\sigma}{\sigma_0},\mu-\frac{a}{2}}\big(H(z)\big) \end{bmatrix}. \end{aligned}$$

Lemma 2.11, Lemma 2.12 and Lemma 2.19 were used. The second matrix equation from the lemma can be obtained by transposing the first equation. □

2.4 Frequency domain

The effect of a filter being symmetric, on the frequency response of that filter, is illustrated with a number of lemmas. The linear-phase conditions for real FIR filters are well understood and described many times in literature, for instance [116] [133] and [136]. In case of complex FIR filters more can be said about the conditions for obtaining a linear phase, [64] [65] [133] [135].
Strictly speaking, linear phase means that the phase response has the form $\mathcal{P}(H(\mathrm{e}^{\mathrm{j}\theta})) = a\theta$. It is common practice that any phase response of the form $\mathcal{P}(H(\mathrm{e}^{\mathrm{j}\theta})) = a\theta + b$ is called linear ($a, b \in \mathbb{R}$). In [64] this type of phase response is called generalized-linear or affine.
In this section it will be shown that (σ,μ)-symmetry implies linear phase. First it is shown that (σ,μ)-symmetry is equivalent to a special frequency domain relation.

Lemma 2.21. *Filter $H(z) \in \mathbb{C}(z)$ is (σ,μ)-symmetric iff:*

$$H(\mathrm{e}^{\mathrm{j}\theta}) = \sigma \mathrm{e}^{-\mathrm{j}2\mu\theta} \overline{H^{*}}(\mathrm{e}^{-\mathrm{j}\theta}).$$

Proof. The proof from left to right is obtained by substituting $z \leftarrow \mathrm{e}^{\mathrm{j}\theta}$, so:

$$\begin{aligned} H(\mathrm{e}^{\mathrm{j}\theta}) &= H(z)\big|_{z\leftarrow \mathrm{e}^{\mathrm{j}\theta}} \\ &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1})\big|_{z\leftarrow \mathrm{e}^{\mathrm{j}\theta}} \\ &= \sigma \mathrm{e}^{-\mathrm{j}2\mu\theta} H^{\overline{*}}(\mathrm{e}^{-\mathrm{j}\theta}), \end{aligned}$$

and the proof from right to left is based on the fact that a finite-length filter is determined completely by the Fourier transform on the unit circle. □

Now the frequency response of a (σ, μ)-symmetric filter can be written as a real-valued function multiplied by a complex quantity that only affects the phase.

Lemma 2.22. *The filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric iff:*

$$H(\mathrm{e}^{\mathrm{j}\theta}) = \sqrt{\sigma}\mathrm{e}^{-\mathrm{j}\mu\theta} H_{\mathrm{zp}}(\theta),$$

with zero-phase frequency response $H_{\mathrm{zp}}(\theta) \in \mathbb{R}$ of the form:

$$H_{\mathrm{zp}}(\theta) = \sum_n \frac{h[n]}{\sqrt{\sigma}} \mathrm{e}^{-\mathrm{j}(n-\mu)\theta},$$

and the period of $H_{\mathrm{zp}}(\theta)$ is 2π in case $\mu \in \mathbb{Z}$ or 4π in case $\mu \in \mathbb{Z} + \frac{1}{2}$.

Proof. Define the zero-phase frequency response $H_{\mathrm{zp}}(\theta)$, or for the reverse of the proof $H(\mathrm{e}^{\mathrm{j}\theta})$, through the following equality:

$$H_{\mathrm{zp}}(\theta) = \frac{1}{\sqrt{\sigma}} \mathrm{e}^{\mathrm{j}\mu\theta} H(\mathrm{e}^{\mathrm{j}\theta}).$$

By Lemma 2.21:

$$\begin{gathered} H(\mathrm{e}^{\mathrm{j}\theta}) = \sigma \mathrm{e}^{-\mathrm{j}2\mu\theta} H^{\overline{*}}(\mathrm{e}^{-\mathrm{j}\theta}) \\ \Leftrightarrow \\ \frac{1}{\sqrt{\sigma}} \mathrm{e}^{\mathrm{j}\mu\theta} H(\mathrm{e}^{\mathrm{j}\theta}) = \left(\frac{1}{\sqrt{\sigma}} \mathrm{e}^{\mathrm{j}\mu\theta} H(\mathrm{e}^{\mathrm{j}\theta})\right)^* \\ \Leftrightarrow \\ H_{\mathrm{zp}}(\theta) \in \mathbb{R}. \end{gathered}$$

The form of $H_{\mathrm{zp}}(\theta)$ follows from its definition, and:

$$H(\mathrm{e}^{\mathrm{j}\theta}) = H(z)\big|_{z\leftarrow \mathrm{e}^{\mathrm{j}\theta}} = \sum_n h[n] z^{-n}\big|_{z\leftarrow \mathrm{e}^{\mathrm{j}\theta}}.$$

Since the period of $H(\mathrm{e}^{\mathrm{j}\theta})$ is 2π and $\mu \in \mathbb{Z}/2$, the period of $H_{\mathrm{zp}}(\theta)$ is 2π or 4π. □

The next lemma shows that the phase of a (σ, μ)-symmetric filter is linear.

Lemma 2.23. *For any (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$ the phase response is:*

$$\mathcal{P}\big(H(e^{j\theta})\big) = \frac{\mathcal{P}(\sigma)}{2} - \mu\theta + \mathcal{P}\big(H_{zp}(\theta)\big).$$

Proof. The proof follows directly by taking the phase of the equation in Lemma 2.22. □

Finally, the effect on the frequency response, by mirroring filter $H(z)$ is given in the following lemma.

Lemma 2.24. *For any filter $H(z) \in \mathbb{C}(z)$:*

$$\big|\mathcal{M}_{\sigma,\mu}\big(H(e^{j\theta})\big)\big| = \big|H(e^{j\theta})\big|,$$

and:

$$\mathcal{P}\big(\mathcal{M}_{\sigma,\mu}\big(H(e^{j\theta})\big)\big) = \mathcal{P}(\sigma) - 2\mu\theta - \mathcal{P}\big(H(e^{j\theta})\big).$$

The phase $\mathcal{P}$ is considered modulo 2π.

Proof. Define $G(z) = \mathcal{M}_{\sigma,\mu}(H(z))$, then:

$$\begin{aligned}
G(z) &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}),\\
G(e^{j\theta}) &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1})\big|_{z \leftarrow e^{j\theta}}\\
&= \sigma e^{-j2\mu\theta} H^{\overline{*}}(e^{-j\theta})\\
&= \sigma e^{-j2\mu\theta} H^{*}(e^{j\theta}),\\
\big|G(e^{j\theta})\big| &= \big|H(e^{j\theta})\big|,\\
\mathcal{P}\big(G(e^{j\theta})\big) &= \mathcal{P}(\sigma) - 2\mu\theta - \mathcal{P}\big(H(e^{j\theta})\big).
\end{aligned}$$

□

2.5 Types of symmetry

In the first part of this chapter the shape of symmetry was allowed to have any value provided that $\sigma \in \mathbb{C}$ and $|\sigma| = 1$. For real filters not every value of σ is allowed, and for complex filters, special values of σ imply special simple relations between the coefficients.

2.5.1 Real filters

For real (σ, μ)-symmetric filters $H(z) \in \mathbb{R}(z)$ the possible values for the shape of symmetry are limited to two possibilities.

Lemma 2.25. *If filter $H(z) \in \mathbb{R}(z)$ is (σ, μ)-symmetric, then $\sigma \in \{-1, 1\}$.*

Proof. By Definition 2.1 for symmetry and the fact that:

$$H(z) \in \mathbb{R}(z) \Rightarrow H^{\overline{*}}(z^{-1}) \in \mathbb{R}(z),$$

it follows that $\sigma \in \mathbb{R}$, and since $|\sigma| = 1$ this results in $\sigma \in \{-1, 1\}$. □

The real symmetric filters are special instances of the (σ,μ)-symmetric filters. In [116] [133] and [136], the labels type 1 and 2 are used to identify the symmetric filters, and the labels type 3 and 4 are used to identify the anti-symmetric filters.

Definition 2.3. *Let the filter $H(z) \in \mathbb{R}(z)$ be (σ,μ)-symmetric and the coefficients $h[n]$ be defined for $0 \leq n < L$. The 4 traditional types are:*

type	L	symmetry	σ	μ
1	odd	symmetric	1	$\mathbb{Z}$
2	even	symmetric	1	$\mathbb{Z}+\frac{1}{2}$
3	odd	anti-symmetric	-1	$\mathbb{Z}$
4	even	anti-symmetric	-1	$\mathbb{Z}+\frac{1}{2}$

$$\textit{with } \mu = \frac{L-1}{2}.$$

The frequency responses and zero-phase responses of the 4 possible types are given next.

Lemma 2.26. *The filters type 1, 2, 3 and 4 have the following frequency responses and zero-phase responses:*

$$\text{type 1: } H(\mathrm{e}^{\mathrm{j}\theta}) = \mathrm{e}^{-\mathrm{j}\mu\theta} H_{zp}(\theta) \textit{ with } H_{zp}(\theta) = h[\mu] + 2\sum_{n<\mu} h[n]\cos(\theta(\mu-n)),$$
$$\text{type 2: } H(\mathrm{e}^{\mathrm{j}\theta}) = \mathrm{e}^{-\mathrm{j}\mu\theta} H_{zp}(\theta) \textit{ with } H_{zp}(\theta) = 2\sum_{n<\mu} h[n]\cos(\theta(\mu-n)),$$
$$\text{type 3: } H(\mathrm{e}^{\mathrm{j}\theta}) = \mathrm{j}\mathrm{e}^{-\mathrm{j}\mu\theta} H_{zp}(\theta) \textit{ with } H_{zp}(\theta) = 2\sum_{n<\mu} h[n]\sin(\theta(\mu-n)),$$
$$\text{type 4: } H(\mathrm{e}^{\mathrm{j}\theta}) = \mathrm{j}\mathrm{e}^{-\mathrm{j}\mu\theta} H_{zp}(\theta) \textit{ with } H_{zp}(\theta) = 2\sum_{n<\mu} h[n]\sin(\theta(\mu-n)).$$

Proof. The proof is well-known and straightforward: substituting $z \leftarrow \mathrm{e}^{\mathrm{j}\theta}$ in $H(z)$, use Lemma 2.1 and Lemma 2.22, Definition 2.3, and finally use one of Euler's identities $\mathrm{e}^{\mathrm{j}x} + \mathrm{e}^{-\mathrm{j}x} = 2\cos(x)$ or $\mathrm{e}^{\mathrm{j}x} - \mathrm{e}^{-\mathrm{j}x} = 2\mathrm{j}\sin(x)$. □

2.5.2 Complex filters

For $H(z) \in \mathbb{R}(z)$ only two possible values for the shape of symmetry exist, viz. 1 and -1, see Lemma 2.25. For $H(z) \in \mathbb{C}(z)$, however, any σ with $|\sigma| = 1$ is possible. The values of $\sigma \in \mathbb{C}_{\mathbb{Z}}$, i.e., $\sigma \in \{1, -1, \mathrm{j}, -\mathrm{j}\}$, receive special attention since the relations between the filter coefficients are simple.

In addition to the 4 types of linear-phase real filters, five additional types of linear-phase complex filters are defined. These new types relate to special values for the shape of symmetry, σ, as defined in Definition 2.4. The center of symmetry, μ, is not used here. Note that the complex filters include the real filters and that type 1 and 2 are covered by type 5, and that type 3 and 4 are covered by type 6.

Definition 2.4. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric and $\mu \in \mathbb{Z}/2$, then:*

type	σ
5	1
6	-1
7	j
8	$-$j
9	$\notin \{1, -1, \mathrm{j}, -\mathrm{j}\}$

The simple relations between the coefficients, that relate to the special values of σ as defined before, are specified in Lemma 2.27.

Lemma 2.27. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric, then the type 5, 6, 7 and 8 filters have the following properties:*

type	$H_\mathrm{r}(z)$	$H_\mathrm{i}(z)$
5	$(1, \mu)$-*symm.*	$(-1, \mu)$-*symm.*
6	$(-1, \mu)$-*symm.*	$(1, \mu)$-*symm.*
7	$\mathcal{M}_{1,\mu}\big(H_\mathrm{i}(z)\big)$	$\mathcal{M}_{1,\mu}\big(H_\mathrm{r}(z)\big)$
8	$\mathcal{M}_{-1,\mu}\big(H_\mathrm{i}(z)\big)$	$\mathcal{M}_{-1,\mu}\big(H_\mathrm{r}(z)\big)$

with $H(z) = H_\mathrm{r}(z) + \mathrm{j}H_\mathrm{i}(z)$ and $H_\mathrm{r}(z), H_\mathrm{i}(z) \in \mathbb{R}(z)$.

Proof. Let $\sigma = \sigma_\mathrm{r} + \mathrm{j}\sigma_\mathrm{i}$, then:

$$\begin{aligned} H(z) &= \sigma z^{-2\mu} H^{\overline{*}}(z^{-1}), \\ H_\mathrm{r}(z) + \mathrm{j}H_\mathrm{i}(z) &= z^{-2\mu}(\sigma_\mathrm{r} + \mathrm{j}\sigma_\mathrm{i})\big(H_\mathrm{r}(z^{-1}) - \mathrm{j}H_\mathrm{i}(z^{-1})\big), \\ H_\mathrm{r}(z) &= z^{-2\mu}\big(\sigma_\mathrm{r} H_\mathrm{r}(z^{-1}) + \sigma_\mathrm{i} H_\mathrm{i}(z^{-1})\big), \\ H_\mathrm{i}(z) &= z^{-2\mu}\big(\sigma_\mathrm{i} H_\mathrm{r}(z^{-1}) - \sigma_\mathrm{r} H_\mathrm{i}(z^{-1})\big). \end{aligned}$$

In the following table, the properties of the filters $H_\mathrm{r}(z)$ and $H_\mathrm{i}(z)$ are given for special values of σ, see Definition 2.4.

type	σ	σ_r	σ_i	$H_\mathrm{r}(z)$	$H_\mathrm{i}(z)$
5	1	1	0	$z^{-2\mu}H_\mathrm{r}(z^{-1})$	$-z^{-2\mu}H_\mathrm{i}(z^{-1})$
6	-1	-1	0	$-z^{-2\mu}H_\mathrm{r}(z^{-1})$	$z^{-2\mu}H_\mathrm{i}(z^{-1})$
7	j	0	1	$z^{-2\mu}H_\mathrm{i}(z^{-1})$	$z^{-2\mu}H_\mathrm{r}(z^{-1})$
8	$-$j	0	-1	$-z^{-2\mu}H_\mathrm{i}(z^{-1})$	$-z^{-2\mu}H_\mathrm{r}(z^{-1})$

The proof is concluded by using Definition 2.1 and Definition 2.2. ☐

Complex coefficients consist of individual real and imaginary parts, the parts. For a type 9 filter the L complex coefficients in general require $2L$ different parts. For type 5, 6, 7 and 8 filters the L complex coefficients in general require only L different parts, because most values are used twice. In Chapter 4 it is discussed extensively how multiplications may be shared in case of symmetry.

For type 5 and 6 filters the individual real and imaginary parts of the filters are symmetric too, see Lemma 2.27. As a consequence the zero-phase response of the type 5 and 6 filters can be expressed in the zero-phase responses of the individual real and imaginary parts of the filters.

Lemma 2.28. *If the filter $H(z) \in \mathbb{C}(z)$ is $(\pm 1, \mu)$-symmetric, then $H_{zp}(\theta) = H_{\mathrm{r},zp}(\theta) \mp H_{\mathrm{i},zp}(\theta)$.*

Proof. Since $H(z) = H_\mathrm{r}(z) + \mathrm{j}H_\mathrm{i}(z)$, filter $H_\mathrm{r}(z)$ is $(\pm 1, \mu)$-symmetric and $H_\mathrm{i}(z)$ is $(\mp 1, \mu)$-symmetric, see Lemma 2.27. Now:

$$\begin{aligned} H(\mathrm{e}^{\mathrm{j}\theta}) &= H_\mathrm{r}(\mathrm{e}^{\mathrm{j}\theta}) + \mathrm{j}H_\mathrm{i}(\mathrm{e}^{\mathrm{j}\theta}), \\ \sqrt{\pm 1}\mathrm{e}^{-\mathrm{j}\mu\theta}H_{zp}(\theta) &= \sqrt{\pm 1}\mathrm{e}^{-\mathrm{j}\mu\theta}H_{\mathrm{r},zp}(\theta) + \mathrm{j}\sqrt{\mp 1}\mathrm{e}^{-\mathrm{j}\mu\theta}H_{\mathrm{i},zp}(\theta), \\ H_{zp}(\theta) &= H_{\mathrm{r},zp}(\theta) \mp H_{\mathrm{i},zp}(\theta). \end{aligned}$$ □

Up to here, the (σ, μ)-symmetry and its equivalence to the linear-phase property of filters, are discussed in detail for filters with complex coefficients, i.e., $H(z) \in \mathbb{C}(z)$. In the remaining part of this chapter the focus will be on (σ, μ)-symmetric filters with complex-integer coefficients, i.e., $H(z) \in \mathbb{C}_\mathbb{Z}(z)$. It will be shown how symmetry of these filters can be preserved under quantization of the coefficients. Also, conversion of the shape of symmetry while the coefficients remain complex integer is discussed. Special focus is on the theorem stating that any complex-integer type 9 filter can be composed of a filter of one of the other types and a complex-integer scale factor, while the filter coefficients remain complex integer.

2.6 Quantizing coefficients

The quantization of filter coefficients in general will affect all kinds of filter properties, like the magnitude and phase response. Simple relations between coefficients, like equal and opposite, as in type 5, 6, 7 and 8, are invariant under quantization, and therefore symmetry and linear phase are preserved.
In general, the quantization of the coefficients of a type 9 filter results in a non-symmetric filter. The relations between the coefficients, $h[n] = \sigma h^*[2\mu - n]$ for all n, see Lemma 2.1, will be lost. In this section it is shown how the coefficients of a type 9 filter may be quantized in a way that preserves the symmetry and hence the linear phase.
First, the quantization $\mathcal{Q}$ of a filter $H(z)$ is defined as the quantization of the individual real and imaginary parts of the coefficients.

Definition 2.5. *For the filter $H(z) \in \mathbb{C}(z)$ the quantization function $\mathcal{Q}_i(H(z))$ is:*

$$\mathcal{Q}_i\big(H(z)\big) \triangleq \sum_n \big(\mathcal{Q}_i\big(h_\mathrm{r}[n]\big) + \mathrm{j}\mathcal{Q}_i\big(h_\mathrm{i}[n]\big)\big)z^{-n},$$

with $Q_i(x)$, for $x \in \mathbb{R}$ and $i \in \mathbb{Z}$, is a function from $\mathbb{R}$ to $\mathbb{Z}/2^i$.

The value of i in $\mathcal{Q}_i(H(z))$ should be sufficiently large to have an appropriate approximation of the filter $H(z)$.
For the type 5, 6, 7 and 8 symmetric filters, the symmetry may be preserved under quantization when satisfying a mild condition only. Note that other filter properties like the magnitude response, may be affected severely.

Lemma 2.29. *For any quantization function $\mathcal{Q}_i(x)$ from $\mathbb{R}$ to $\mathbb{Z}/2^i$ with $x \in \mathbb{R}$ and $i \in \mathbb{Z}$:*

- *the symmetry of a type 7 filter is preserved,*
- *the symmetry of a type 5, 6 and 8 filter is preserved if the quantization itself is anti-symmetric, i.e., $\mathcal{Q}_i(x) = -\mathcal{Q}_i(-x)$.*

Proof. By Lemma 2.27, the real and imaginary parts of all pairs of coefficients of type 5, 6 and 8 filters, $(h[n], h[2\mu - n])$, are either equal or opposite. The condition $\mathcal{Q}_i(x) = -\mathcal{Q}_i(-x)$ for any $x \in \mathbb{R}$ is sufficient to preserve these relations. Since for type 7 filters the parts are all equal, the extra condition is not required. □

In [136] a few popular quantization characteristics are presented, from which it follows that the property $\mathcal{Q}_i(x) = -\mathcal{Q}_i(-x)$ is not satisfied in some cases.
For type 9 filters quantization is possible at the cost of an approximation of the shape of symmetry. The next lemma describes the alternative quantization procedure. Basically, the type 9 filter $H(z)$ is first reduced over $\mathbb{C}$ by the complex scale factor a, such that the resulting filter is of type 5, 6, 7 or 8. Second, both the resulting filter and the scale factor are quantized. According to Lemma 2.29 the symmetry of any of these filters is invariant under quantization. Finally, the quantized filter is multiplied by the quantized version of scale factor a.

Lemma 2.30. *If filter $H(z) \in \mathbb{C}(z)$ is (σ, μ)-symmetric, then a filter $H'(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ that is (σ', μ)-symmetric can be constructed as:*

$$H'(z) = \mathcal{Q}_i\Big(\frac{H(z)}{a}\Big)\mathcal{Q}_0(a),$$

for $a \in \{b\sqrt{\sigma}, b\sqrt{-\sigma}, b\sqrt{\frac{\sigma}{\mathrm{j}}}, b\sqrt{-\frac{\sigma}{\mathrm{j}}}\}$, $b \in \mathbb{R}$ and $\mathcal{Q}_i(x) = -\mathcal{Q}_i(-x)$. The new shape of symmetry σ' is respectively:

$$\frac{\mathcal{Q}_0(a)}{\mathcal{Q}_0^*(a)}, -\frac{\mathcal{Q}_0(a)}{\mathcal{Q}_0^*(a)}, \mathrm{j}\frac{\mathcal{Q}_0(a)}{\mathcal{Q}_0^*(a)} \text{ and } -\mathrm{j}\frac{\mathcal{Q}_0(a)}{\mathcal{Q}_0^*(a)}.$$

In case $a = b\sqrt{\frac{\sigma}{\mathrm{j}}}$, the extra requirement of anti-symmetry for the quantizer is not needed.

Proof. By Lemma 2.5 and Lemma 2.29 it can be verified that filter $G(z) = \mathcal{Q}_i\big(\frac{H(z)}{a}\big)$ is type 5, 6, 7 or 8 respectively. Again, by Lemma 2.5 it can be verified that the filter $H'(z) = G(z)\mathcal{Q}_0(a)$ is (σ', μ)-symmetric. □

2.7 Shape of symmetry conversion

A number of transformations of a symmetric filter $H(z) \in \mathbb{C}(z)$, result in the symmetric filter $G(z) \in \mathbb{C}(z)$ with a different shape of symmetry. Many of these transformations are introduced earlier as a lemma. In general, any shape of symmetry can be transformed into any other shape of symmetry, by applying one or more transformations. As a consequence the type of filter may change.

Changing the shape of symmetry may be a way to reduce the costs of a filter, since special relations between coefficients can be exploited. A transformation in general may affect the frequency response of the filter, and it depends on the context whether such transformation is allowed or not.
In case the quantization of coefficients should be unaffected by a transformation, i.e., if both filters $H(z), G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$, not every shape of symmetry, and hence not every transformation, is possible. Also, when filter $G(z)$ should be of type 5, 6, 7 or 8, in case filter $H(z)$ is of type 5, 6, 7 or 8, the possibilities for transformations are limited.
In this section some transformations are discussed in more detail. In Section 2.8, type 9 filters with quantized coefficients will be discussed extensively.

2.7.1 Complex scaling

From Lemma 2.5 it follows directly that the shape of symmetry of any (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$ can be changed by multiplication with the scale factor $a \in \mathbb{C}$. The filter $G(z) = aH(z)$ is $(\sigma_0\sigma, \mu)$-symmetric, with $\sigma_0 = \frac{a}{a^*}$, and its frequency response is similar: change of gain by $|a|$ and an offset in the phase by $\mathcal{P}(a)$. An interesting set of values for a is $a \in \mathbb{C}_{\mathbb{Z}}$, because quantization is not affected.
Conversion between the type 5, 6, 7 and 8 filters requires $\sigma, \sigma_0 \in \{1, -1, \mathrm{j}, -\mathrm{j}\}$, that can be made, e.g., by $a \in \{1, \mathrm{j}, 1+\mathrm{j}, 1-\mathrm{j}\}$. Note that $a = 1+\mathrm{j}$ and $a = 1-\mathrm{j}$, both introduce a gain of $\sqrt{2}$.

2.7.2 Complex modulation

Another approach to change the shape of symmetry of a filter is by complex modulation. This modulation has as primary effect that the frequency response of the filter will be shifted, see Section A.6. This shift can be anticipated for in the filter design process. As secondary effect the shape of symmetry of the filter may change. Now let the coefficients of filter $H(z) \in \mathbb{C}(z)$ that is (σ, μ)-symmetric, be modulated by the complex carrier $c[n] = \mathrm{e}^{\mathrm{j}\theta_c n}$, then the resulting filter $G(z) = H(z\mathrm{e}^{-\mathrm{j}\theta_c})$ is $(\sigma_0\sigma, \mu)$-symmetric with $\sigma_0 = \mathrm{e}^{\mathrm{j}2\theta_c\mu}$, see Lemma 2.5. The new shape of symmetry is affected by the frequency of the complex carrier and the original center of symmetry. Interesting instances of the carrier $c[n]$ are $c[n] = \mathrm{e}^{\mathrm{j}\theta_c n} = \mathrm{j}^{kn}$ with $k \in \mathbb{Z}$ because quantization of the coefficients is not affected.
To convert between the type 5, 6, 7 and 8 filters, here too it is necessary that $\sigma, \sigma_0 \in \{1, -1, \mathrm{j}, -\mathrm{j}\}$. If $\mathrm{e}^{\mathrm{j}\theta_c} = \mathrm{j}^k$ and $\mu \in \mathbb{Z}/2$, a suited frequency implies $(2\mu k)|_4 = m|_4$ for $\sigma_0 = \mathrm{j}^m$ with $m \in \mathbb{Z}$. The values $\sigma_0 = \pm\mathrm{j}$ can only be obtained for $\mu \in \mathbb{Z} + \frac{1}{2}$.
Compared to complex scaling, the complex modulation can realize $\sigma_0 = \pm\mathrm{j}$ without a gain of $\sqrt{2}$, in case $\mu \in \mathbb{Z} + \frac{1}{2}$.

2.7.3 Conjugation

Conjugation of the coefficients of the (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$, leaves the quantization unaffected, $G(z) = \overline{H^*}(z)$. Filter $G(z)$ is $(\frac{1}{\sigma}, \mu)$-symmetric, see Lemma 2.6.

So, under conjugation type 5 and 6 filters remain type 5 and 6 filters respectively, whereas the type 7 and 8 filters swap roles.
The effect on the frequency response is $G(\mathrm{e}^{\mathrm{j}\theta}) = H^{\overline{*}}(\mathrm{e}^{\mathrm{j}\theta}) = H^*(\mathrm{e}^{-\mathrm{j}\theta})$, showing that the role of positive and negative frequencies are interchanged and that the phase is opposite.

2.8 Reduction of (σ, μ)-symmetric filters over $\mathbb{C}_\mathbb{Z}$

Theorem 2.1 states that all symmetric complex filters essentially have shape of symmetry equal to 1. This is an important result as it allows efficient design of symmetric filters. In many cases, however, the coefficients of the filter $H(z)$ are highly restricted. For example, the coefficients of $H(z)$ might be complex integers. Within this restricted class of filters it is not obvious at all that a similar result holds. In particular, when $H(z) \in \mathbb{C}_\mathbb{Z}(z)$, by choosing $a = \sqrt{\sigma}$ as in the proof of Theorem 2.1, then $G(z)$ does not necessarily have complex-integer coefficients. Consider the following example.

Example 2.1. Given the $(\mathrm{j}, \frac{1}{2})$-symmetric filter $H(z) = \mathrm{j} + z^{-1}$. Reducing $H(z)$ to the type $(1, \frac{1}{2})$ filter $G(z) = c + c^* z^{-1}$, would mean the existence of complex integers c and λ such that $\mathrm{j} = \lambda c$ and $1 = \lambda c^*$. Using some simple calculus, it is easy to show that such λ and c cannot exist and therefore that $H(z)$ cannot be reduced to a type $(1, \frac{1}{2})$ filter. However, note that:

$$2H(z) = (1 + \mathrm{j})\big((1 + \mathrm{j}) + (1 - \mathrm{j})z^{-1}\big),$$

and therefore that $2H(z)$ can be reduced to a type $(1, \frac{1}{2})$ complex filter over $\mathbb{C}_\mathbb{Z}$.
End of example

In the previous section it has been shown how the shape of symmetry of a type 5, 6, 7 or 8 filter with quantized coefficients can be changed, while the type of symmetry remains type 5, 6, 7 or 8, and the coefficients are still quantized.
As a surprising result, it is proven in the remainder of this section that the concept of reduction can be generalized to arbitrary symmetric complex-integer filters. In particular, it is proven that an arbitrary symmetric filter $H(z) \in \mathbb{C}_\mathbb{Z}(z)$ can be reduced over $\mathbb{C}_\mathbb{Z}$ to a filter $G(z)$ with shape of symmetry equal to 1 or j.
In case $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$, the reduction of filter $2^i H(z) \in \mathbb{C}_\mathbb{Z}(z)$ can be considered instead. As a post-processing step, the scale factor λ and the filter $G(z)$ can be divided by 2^{i-j} and 2^j to restore the original scaling.

Theorem 2.2 (Reduction over $\mathbb{C}_\mathbb{Z}$)**.** *Let $H(z) \in \mathbb{C}_\mathbb{Z}(z)$ be a complex (σ, μ)-symmetric filter, then there is a $(1, \mu)$ or a (j, μ)-symmetric complex filter $G(z) \in \mathbb{C}_\mathbb{Z}(z)$ and a complex number $a \in \mathbb{C}_\mathbb{Z}$, such that $H(z) = aG(z)$. Moreover, if $H(z) \in \mathbb{C}_{2\mathbb{Z}}(z)$, then $G(z)$ may be chosen to have shape of symmetry equal to 1 or j.*

Proof. The proof of this theorem is presented in Section 2.8.1. □

From this theorem it follows directly that the design of (σ, μ)-symmetric filters $H(z) \in \mathbb{C}_\mathbb{Z}(z)$, does not give an additional degree of freedom, compared to the design of $(1, \mu)$-

or (j,μ)-symmetric filters in $\mathbb{C}_{\mathbb{Z}}(z)$ and a scale factor in $\mathbb{C}_{\mathbb{Z}}$. It is also clear that reduction of a filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ to $(1,\mu)$-symmetry, in the worst case requires the use of the scaled complex integers $\mathbb{C}_{\mathbb{Z}/2}$.
Alternatively, every (σ,μ)-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, can be designed by first designing a $(1,\mu)$-symmetric complex filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2}(z)$, and second multiplying all coefficients of $G(z)$ with an appropriate complex-integer scale factor. Directly designing the (σ,μ)-symmetric filter does not provide additional solutions.
In Section 2.8.2, a possible procedure for reducing complex filters is presented and in Section 2.8.3 the possibilities for reduction are elaborated. To show the application of the reduction procedure, an example is presented in Section 2.9.

2.8.1 Proof of reduction theorem

To reduce the filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ over $\mathbb{C}_{\mathbb{Z}}$, the scale factor $\lambda \in \mathbb{C}_{\mathbb{Z}}$ should be a complex-integer divisor of each coefficient of $H(z)$. Such a scale factor is called a divisor of the filter. To reduce the (σ,μ)-symmetric filter to a $(1,\mu)$-symmetric filter, the shape of symmetry of λ should be σ. Therefore $\sigma = \frac{\lambda}{\lambda^*}$. Note that a scale factor is a rudimental filter of length 1, e.g., see Table 2.1 line 3 for $n = 0$. Similarly to reduce the (σ,μ)-symmetric filter to a (j,μ)-symmetric filter, the shape of symmetry of λ should be $\frac{\sigma}{\mathrm{j}}$. Therefore $\frac{\sigma}{\mathrm{j}} = \frac{\lambda}{\lambda^*}$. First a kind of minimal form for the complex-integer scale factor λ, the *minimal factor*, is defined and its existence and uniqueness are proven in Lemma 2.31.
To illustrate the concept of the minimal factor consider the following. According to Lemma 2.1, for any (σ,μ)-symmetric filter $H(z) \in \mathbb{C}(z)$ holds: $h[n] = \sigma h^*[2\mu - n]$ for all n, or $\sigma = \frac{h[n]}{h^*[2\mu-n]}$. So, σ is an arbitrary complex value $\sigma \in \mathbb{C}$ with $|\sigma| = 1$. Now it is straightforward to determine a scale factor $\lambda \in \mathbb{C}$ such that, e.g., $\frac{\lambda}{\lambda^*} = \sigma$ or $\frac{\lambda}{\lambda^*} = \frac{\sigma}{\mathrm{j}}$, viz., $\lambda = c\sqrt{\sigma}$ or $\lambda = c\sqrt{-\mathrm{j}\sigma}$ respectively, with $c \in \mathbb{R}\backslash\{0\}$. See Example 2.2.

Example 2.2. Consider the $(\sigma,0)$-symmetric filter $H(z) \in \mathbb{C}(z)$ with, for a specific n: $h[n] = 1 + \mathrm{j}2$ and $h[-n] = \sqrt{5}$. For the shape of symmetry $\sigma = \frac{1}{5}\sqrt{5}(1 + \mathrm{j}2)$ the solutions for $\frac{\lambda}{\lambda^*} = \sigma$ and $\frac{\lambda}{\lambda^*} = \frac{\sigma}{\mathrm{j}}$ can be derived, viz., $\lambda = c\sqrt{\sigma} = c'\sqrt{1 + \mathrm{j}2}$ and $\lambda = c\sqrt{-\mathrm{j}\sigma} = c'\sqrt{2 - \mathrm{j}}$ respectively, with $c, c' \in \mathbb{R}\backslash\{0\}$.
End of example

In case $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, the existence of a scale factor $\lambda \in \mathbb{C}_{\mathbb{Z}}$ such that, e.g., $\frac{\lambda}{\lambda^*} = \sigma$ or $\frac{\lambda}{\lambda^*} = \frac{\sigma}{\mathrm{j}}$, is not obvious. However, Example 2.3 shows that such λ can exist.

Example 2.3. Consider the $(\sigma,0)$-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ with, for a specific n: $h[n] = -57 - \mathrm{j}41$ and $h[-n] = 21 - \mathrm{j}67$. For the shape of symmetry $\sigma = \frac{1}{5}(-4 + \mathrm{j}3)$ it can be checked that for $\frac{\lambda}{\lambda^*} = \sigma$ and $\frac{\lambda}{\lambda^*} = \frac{\sigma}{\mathrm{j}}$, the following scale factors are solutions, $\lambda = c(1 + \mathrm{j}3)$ and $\lambda = c(2 + \mathrm{j})$ respectively, with $c \in \mathbb{Z}\backslash\{0\}$.
End of example

The minimal factor used in this proof is such $\lambda \in \mathbb{C}_{\mathbb{Z}}$ with the additional constraints that $\gcd(\lambda_{\mathrm{r}}, \lambda_{\mathrm{i}}) = 1$ and $\lambda_{\mathrm{r}} \geq 0$. The minimal factor to reduce filter $H(z)$ to a $(1,\mu)$-symmetric filter is called a minimal factor of the first kind, and the minimal factor to reduce filter $H(z)$ to a (j, μ)-symmetric filter is called a minimal factor of the second kind. In Lemma 2.32, Lemma 2.33 and Lemma 2.34 it will be shown that the minimal factor of the first kind may be a divisor of filter $H(z)$. Similarly in Lemma 2.35, Lemma 2.36 and Lemma 2.37 it will be shown that the minimal factor of the second kind may be a divisor of filter $H(z)$. In Lemma 2.38 it is shown that at least one of these minimal factors is a divisor. Finally the proof of Theorem 2.2 is presented.

Definition 2.6 (Minimal factor)**.** *For every complex rational $\nu \in \mathbb{C}_{\mathbb{Z}}/\mathbb{C}_{\mathbb{Z}}$ with $|\nu| = 1$, a* minimal factor $\alpha = \alpha_{\mathrm{r}} + \mathrm{j}\alpha_{\mathrm{i}}$ *with $\alpha \in \mathbb{C}_{\mathbb{Z}}\backslash\{0\}$, is defined: $\alpha = \mathrm{j}$ if $\nu = -1$, and $\nu = \frac{\alpha}{\alpha^*}$ with $\gcd(\alpha_{\mathrm{r}}, \alpha_{\mathrm{i}}) = 1$, $\alpha_{\mathrm{r}} \geq 0$ if $\nu \neq -1$.*

Next an important property of the minimal factor will be shown.

Lemma 2.31. *For every $\nu \in \mathbb{C}_{\mathbb{Z}}/\mathbb{C}_{\mathbb{Z}}$ with $|\nu| = 1$, a minimal factor exists and is unique.*

Proof. The lemma is proven in a number of steps:

1. First it is shown that ν can be written as $\nu = \frac{h}{h^*}$ for some complex integer h. To prove this use that ν is a complex rational and write ν as $\nu = \frac{f}{g^*}$ with $f, g \in \mathbb{C}_{\mathbb{Z}}$. As $|\nu| = 1$, it follows that $|f| = |g|$. If $f + g \neq 0$ it can be verified for $h = f + g$ that $\nu = \frac{h}{h^*}$. In case $f + g = 0$ it can be verified for $h = \mathrm{j}f$ that $\nu = \frac{h}{h^*}$.

2. Now taking α as $\alpha = \frac{h}{\gcd(h_{\mathrm{r}}, h_{\mathrm{i}})}$, in case $\alpha_{\mathrm{r}} < 0$ substitute $\alpha \leftarrow -\alpha$, and in case $\alpha_{\mathrm{r}} = 0$ taking $\alpha = \mathrm{j}$, proves the existence of a minimal factor.

3. Take $\lambda \in \mathbb{C}_{\mathbb{Z}}$ such that $\nu = \frac{\lambda}{\lambda^*}$, then $\alpha\lambda^* = \alpha^*\lambda$ implies $\alpha\lambda^* \in \mathbb{Z}$. As a consequence the imaginary part $\Im(\alpha\lambda^*) = \alpha_{\mathrm{i}}\lambda_{\mathrm{r}} - \lambda_{\mathrm{i}}\alpha_{\mathrm{r}} = 0$. By definition α_{r} and α_{i} have no common factor, so there exists a unique $k \in \mathbb{Z}$ for every λ such that $\lambda_{\mathrm{r}} = k\alpha_{\mathrm{r}}$ and $\lambda_{\mathrm{i}} = k\alpha_{\mathrm{i}}$, which proves the uniqueness of the minimal factor. □

In the following definition the roles of the minimal factors of both kinds are presented. In the remainder of this thesis we will use α and β as minimal factor of the first kind and minimal factor of the second kind respectively. Note that for both kinds of minimal factors holds: $\alpha, \beta \in \mathbb{C}_{\mathbb{Z}}$. At this point no special properties of $G(z)$ are claimed, i.e., $G(z) \in \mathbb{C}(z)$. At the end of the proof it will be clear under which conditions $G(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ will hold.

Definition 2.7 (Minimal factors of first and second kind)**.** *Let $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ be any (σ, μ)-symmetric complex filter: for the minimal factor of the first kind, α, holds $H(z) = \alpha G(z)$ with $G(z)$ being $(1, \mu)$-symmetric, and for the minimal factor of the second kind, β, holds $H(z) = \beta G(z)$ with $G(z)$ being (j, μ)-symmetric.*

Now it will be shown which conditions on the minimal factors α and β make them a divisor of filter $H(z)$. In the following three lemmas it is shown that if $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ is odd, then α is a divisor of $H(z)$.

Lemma 2.32. *Let α be the minimal factor of $\sigma = \frac{f}{g^*}$ with $f, g \in \mathbb{C}_{\mathbb{Z}}$. If $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ is odd, then $f + g$ is even.*

Proof. This proof considers the remainder in case of an integer division by 2:

$$\frac{f}{g^*} = \frac{\alpha}{\alpha^*},$$
$$(\alpha^* f - \alpha g^*)|_2 = 0,$$
$$(\alpha^*|_2 f + \alpha|_2 g^*)|_2 = 0.$$

The assumption that $\alpha_\mathrm{r} + \alpha_\mathrm{i}$ is odd, implies two cases:

- α_r is odd and α_i is even, gives $(f + g^*)|_2 = (f + g)|_2 = 0$, and
- α_r is even and α_i is odd, gives $(\mathrm{j}f + \mathrm{j}g^*)|_2 = 0$ which implies $(f + g)|_2 = 0$. □

The next lemma shows that if $f + g$ is even, α is a divisor of both f and g.

Lemma 2.33. *Let α be the minimal factor of $\sigma = \frac{f}{g^*}$ with $f, g \in \mathbb{C}_\mathbb{Z}$ and $f + g$ is even, then there is an $h \in \mathbb{C}_\mathbb{Z}$ such that $f = \alpha h$ and $g = \alpha h^*$.*

Proof. Define $h_0 = \frac{f+g}{2}$ and $h_1 = -\mathrm{j}\frac{f-g}{2}$, then by assumption $h_\ell \in \mathbb{C}_\mathbb{Z}$, with $\ell \in \{0, 1\}$. Now there is a unique $k_\ell \in \mathbb{Z}$ such that $h_\ell = k_\ell \alpha$. This follows from Lemma 2.31 when taking $\sigma = \frac{h_\ell}{h_\ell^*}$ with $h_\ell \neq 0$. If $h_\ell = 0$ take $k_\ell = 0$. Now defining $h = k_0 + \mathrm{j}k_1$ gives $\alpha h = k_0\alpha + \mathrm{j}k_1\alpha = \frac{f+g}{2} + \frac{f-g}{2} = f$ and $\alpha h^* = k_0\alpha - \mathrm{j}k_1\alpha = \frac{f+g}{2} - \frac{f-g}{2} = g$. □

On basis of the two previous lemmas it is shown that α is a divisor of the filter $H(z)$.

Lemma 2.34. *Let $H(z) \in \mathbb{C}_\mathbb{Z}(z)$ be a (σ, μ)-symmetric filter, and let α be the minimal factor of σ. If $\alpha_\mathrm{r} + \alpha_\mathrm{i}$ is odd, there is a $(1, \mu)$-symmetric filter $G(z) \in \mathbb{C}_\mathbb{Z}(z)$ such that $H(z) = \alpha G(z)$.*

Proof. Consider the pair of coefficients (f, g) of $H(z)$ with $f = h[n]$ and $g = h[2\mu - n]$ for index n. Type (σ, μ)-symmetry implies that $\sigma = \frac{f}{g^*}$ and by Lemma 2.32 and Lemma 2.33 that there is a $p \in \mathbb{C}_\mathbb{Z}$ such that $f = \alpha p$ and $g = \alpha p^*$. This is applicable to all pairs (f, g), and therefore $H(z)$ is divisible by α and by application of Lemma 2.5 $G(z)$ is of type $(1, \mu)$. □

Similarly, in the following three lemmas it is shown that if $\beta_\mathrm{r} + \beta_\mathrm{i}$ is odd, then minimal factor β is a divisor of $H(z)$.

Lemma 2.35. *Let β be the minimal factor of $\frac{\sigma}{\mathrm{j}} = \frac{f}{(-\mathrm{j}g)^*}$ with $f, g \in \mathbb{C}_\mathbb{Z}$. If $\beta_\mathrm{r} + \beta_\mathrm{i}$ is odd, then $f - \mathrm{j}g$ is even.*

Proof. For this proof use the proof of Lemma 2.32 and substitute $g \leftarrow -\mathrm{j}g$. □

The next lemma shows that if $f - \mathrm{j}g$ is even, β is a divisor of both f and g.

Lemma 2.36. *Let β be the minimal factor of $\frac{\sigma}{\mathrm{j}} = \frac{f}{(-\mathrm{j}g)^*}$ with $f, g \in \mathbb{C}_\mathbb{Z}$ and $f - \mathrm{j}g$ is even, then there is an $h \in \mathbb{C}_\mathbb{Z}$ such that $f = \beta h$ and $g = \mathrm{j}\beta h^*$.*

Proof. For this proof use the proof of Lemma 2.33 and substitute $g \leftarrow -\mathrm{j}g$. □

On basis of the two previous lemmas it is shown that β is a divisor of the filter $H(z)$.

Lemma 2.37. *Let $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ be a (σ, μ)-symmetric filter, and let β be the minimal factor of $\frac{\sigma}{\mathrm{j}}$. If $\beta_{\mathrm{r}} + \beta_{\mathrm{i}}$ is odd, there is a (j, μ)-symmetric filter $G(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ such that $H(z) = \beta G(z)$.*

Proof. Consider the pair of coefficients (f, g) of $H(z)$ with $f = h[n]$ and $g = h[2\mu - n]$ for index n. Type (σ, μ) implies that $\sigma = \frac{f}{g^*}$ and by Lemma 2.35 and Lemma 2.36 that there is a $p \in \mathbb{C}_{\mathbb{Z}}$ such that $f = \beta p$ and $g = \beta \mathrm{j} p^*$. This is applicable to all pairs (f, g), and therefore $H(z)$ is divisible by β and by application of Lemma 2.5 $G(z)$ is of type (j, μ). □

The next lemma shows that at least one minimal factor, α or β, is a divisor of filter $H(z)$. In Example 2.5 it will be shown that a minimal factor that does not satisfy the parity condition can be a divisor.

Lemma 2.38. *If α is the minimal factor of σ, and β is the minimal factor of $\frac{\sigma}{\mathrm{j}}$, then $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ is odd or $\beta_{\mathrm{r}} + \beta_{\mathrm{i}}$ is odd.*

Proof. It is proven that if $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ is even, the alternative $\beta_{\mathrm{r}} + \beta_{\mathrm{i}}$ is odd. From the minimal factor, Definition 2.6, it follows that, even parity of $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ implies that both α_{r} and α_{i} are odd. From the definition of both minimal factors it follows $\frac{\beta}{\beta^*} = \frac{1}{\mathrm{j}} \frac{\alpha}{\alpha^*} = \frac{1-\mathrm{j}}{1+\mathrm{j}} \frac{\alpha}{\alpha^*}$. Consider $\beta' = (1 - \mathrm{j})\alpha$. It is easy to verify that both β'_{r} and β'_{i} are even, so $\beta'' = \frac{1-\mathrm{j}}{2}\alpha \in \mathbb{C}_{\mathbb{Z}}$. Also $\gcd(\beta''_{\mathrm{r}}, \beta''_{\mathrm{i}}) = 1$ because $\alpha = (1 + \mathrm{j})\beta''$ and a common factor of β'' is a common factor of α. Therefore $\beta'' = \beta$ is the minimal factor. Using $\beta = \frac{1-\mathrm{j}}{2}\alpha$ directly gives: $\beta_{\mathrm{r}} + \beta_{\mathrm{i}} = \alpha_{\mathrm{i}}$ is odd. □

The proof of Theorem 2.2 on page 36 now is as follows.

Proof. By Lemma 2.38, either $\alpha_{\mathrm{r}} + \alpha_{\mathrm{i}}$ is odd or $\beta_{\mathrm{r}} + \beta_{\mathrm{i}}$ is odd. Therefore all possible values of $\sigma \in \mathbb{C}_{\mathbb{Z}}/\mathbb{C}_{\mathbb{Z}}$ are covered by Lemma 2.34 and Lemma 2.37, and filter $G(z)$ is either of $(1, \mu)$-symmetric for $\lambda = \alpha$, or of (j, μ)-symmetric for $\lambda = \beta$.
Part 2 of the proof is: dividing (j, μ)-symmetric filter $G(z) \in \mathbb{C}(z)$ by $(1 + \mathrm{j})$ gives a $(1, \mu)$-symmetric filter. Reducing $G(z)$ over $\mathbb{C}_{\mathbb{Z}}$ is trivially possible if $G(z) \in \mathbb{C}_{2\mathbb{Z}}(z)$, what is implied by assuming $H(z) \in \mathbb{C}_{2\mathbb{Z}}(z)$. □

The proof of the theorem is constructive and a procedure for filter reduction along the lines of this proof will be discussed in the following section.

2.8.2 Reduction procedure

First the candidate filter for reduction is verified to be (σ, μ)-symmetric (Step 1). From this the minimal factors of the first and second kind, α and β, are determined (Step 2 and Step 3 respectively). The conditions of the minimal factors are verified and if found to be satisfied, the reduction is performed (Step 4 or Step 5).

Step 1: Check that $H(z)$ is (σ, μ)-symmetric by determining $\sigma_n = \frac{h[n]}{h^*[2\mu-n]} = \sigma$ for all n, and choose $p, q \in \mathbb{C}_{\mathbb{Z}}$ such that $\sigma = \frac{p}{q^*}$.

Step 2: Determine the minimal factor of the first kind, α:
if $p + q \neq 0$ take $h = p + q$, otherwise take $h = \mathrm{j}p$. Now $\alpha = \frac{h}{\gcd(h_\mathrm{r}, h_\mathrm{i})}$.
In case $\alpha_\mathrm{r} < 0$ substitute $\alpha \leftarrow -\alpha$, and in case $\alpha_\mathrm{r} = 0$ take $\alpha = \mathrm{j}$.

Step 3: Determine the minimal factor of the second kind, β:
if $p - \mathrm{j}q \neq 0$ take $h = p - \mathrm{j}q$, otherwise take $h = -\mathrm{j}p$. Now $\beta = \frac{h}{\gcd(h_\mathrm{r}, h_\mathrm{i})}$.
In case $\beta_\mathrm{r} < 0$ substitute $\beta \leftarrow -\beta$, and in case $\beta_\mathrm{r} = 0$ take $\beta = \mathrm{j}$.

Step 4: If $\alpha_\mathrm{r} + \alpha_\mathrm{i}$ is odd then $G(z) = \frac{H(z)}{\alpha}$ is $(1, \mu)$-symmetric (type 5).

Step 5: If $\beta_\mathrm{r} + \beta_\mathrm{i}$ is odd then $G(z) = \frac{H(z)}{\beta}$ is (j, μ)-symmetric (type 7).

Although Theorem 2.2 is about reducing filters $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, it is trivially applicable to filters $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$.

2.8.3 Reduction possibilities

Theorem 2.2 states that reduction of a (σ, μ)-symmetric filter to a $(1, \mu)$- or (j, μ)-symmetric filter, is always possible. This is equivalent to the statement that at least one of the minimal factors, α or β, is a divisor of $H(z)$. It is remarkable to see that properties of the filters $H_\mathrm{r}|_2(z)$ and $H_\mathrm{i}|_2(z)$, completely determine the reducibility of filter $H(z)$ to $(1, \mu)$-symmetry, to (j, μ)-symmetry or to both types of symmetry.

Theorem 2.3 (Reduction possibilities)**.** *Let $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ have (σ, μ)-symmetry:*

- *if $H_\mathrm{r}|_2(z)$ and $H_\mathrm{i}|_2(z)$ both have $(1, \mu)$-symmetry, then $H(z)$ can be reduced with a minimal factor of the first kind,*

- *if $H_\mathrm{r}|_2(z) = z^{-2\mu} H_\mathrm{i}|_2(z^{-1})$, then $H(z)$ can be reduced with a minimal factor of the second kind.*

Proof. If $(h[n] + h[2\mu - n])|_2 = 0$ holds for all n, then according to Lemma 2.33 the minimal factor of the first kind is a divisor of $H(z)$. This condition is equivalent to: $h[n]|_2 = h[2\mu - n]|_2$ holds for all n, which is equivalent to $H_\mathrm{r}|_2(z)$ and $H_\mathrm{i}|_2(z)$ both have $(1, \mu)$-symmetry. Similarly, if $(h[n] - \mathrm{j}h[2\mu - n])|_2 = 0$ holds for all n, then according to Lemma 2.36 the minimal factor of the second kind is a divisor of $H(z)$. This condition is equivalent to: $h[n]|_2 = \mathrm{j}h^*[2\mu - n]|_2$ holds for all n, which is equivalent to $H_\mathrm{r}|_2(z) = z^{-2\mu} H_\mathrm{i}|_2(z^{-1})$. □

2.8.4 Norms after reduction

In case a filter is reduced over the complex integers, the norms of the resulting filter and its output signal may have become smaller. As a consequence the required number of bits to represent the filter coefficients and the output signal may be less.

Theorem 2.4 (Norms after reduction)**.** *If the (σ, μ)-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ is reduced over $\mathbb{C}_{\mathbb{Z}}$ to $H(z) = aG(z)$, and the input signal is given by $X(z) \in \mathbb{C}(z)$, then:*

- *the maximum absolute coefficient value of filter $G(z)$ is at most the maximum absolute coefficient value of filter $H(z)$, i.e. $\|G(z)\|_\infty \leq \|H(z)\|_\infty$,*
- *the maximum absolute value of the output signal $X(z)G(z)$ is at most the maximum absolute value of the output signal $X(z)H(z)$, i.e. $\|G(z)\|_1 \leq \|H(z)\|_1$.*

Proof. Since filter $G(z)$ can be written as $G(z) = \frac{1}{a}H(z)$, Lemma A.14 and Lemma A.15 apply. By Lemma A.16, the 1-norm of $\frac{1}{a}$ is at most 1. The excluded case in Lemma A.16, $a = 0$, relates to an unpractical case and therefore needs no consideration. □

2.9 Examples

In this section two typical symmetric filters are discussed in more detail. Example 1 elaborates on the simplest possible non-trivial filter, and Example 2 discusses in detail the reduction of a (σ, μ)-symmetric filter.

2.9.1 Example 1

Example 2.4. By means of two filters of length 2, it is shown that a type 7 filter may be simpler than a type 5 filter, while their frequency responses are similar.
A very simple $(1, \frac{1}{2})$-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ is $H(z) = (1 - \mathrm{j}) + (1 + \mathrm{j})z^{-1}$, and another very simple, perhaps the simplest possible non-trivial, $(\mathrm{j}, \frac{1}{2})$-symmetric filter $G(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ is $G(z) = 1 + \mathrm{j}z^{-1}$. For real-valued input signals, filter $H(z)$ requires a single delay element and two real additions, whereas filter $G(z)$ requires a single delay element and no addition, see Figure 2.3(a) and Figure 2.3(b) respectively.

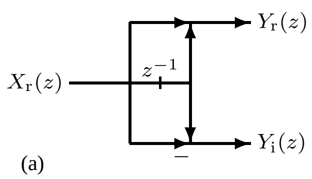


Figure 2.3: *Example 1: $(1, \frac{1}{2})$-symmetric filter $H(z)$ (a) and $(\mathrm{j}, \frac{1}{2})$-symmetric filter $G(z)$ (b).*

It follows directly that $H(z) = (1-\mathrm{j})G(z)$ and that their frequency responses are similar. Both filters have their zero at $z_0 = -\mathrm{j}$. In Figure 2.4 the amplitude- and phase responses of both filters, $H(z)$ (dashed line) and $G(z)$ (solid line), are shown. The magnitude responses differ, $|1 - \mathrm{j}| = \sqrt{2} \approx 3$ dB, and from the phase response plots, the phase shift $\mathcal{P}(1 - \mathrm{j}) = -\frac{\pi}{4}$ is clearly visible.

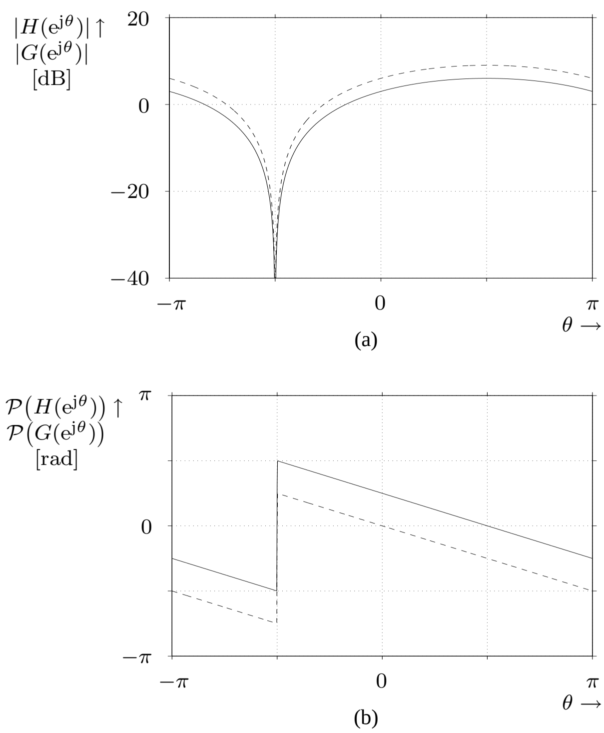


Figure 2.4: *Example 1: Frequency responses of filters $H(z)$ (dashed line) and $G(z)$ (solid line).*

Whether filter $G(z)$ may be used depends on its application: are the differences in gain and phase acceptable or not.
End of example

2.9.2 Example 2

Example 2.5. In this example the reduction of a (σ, μ)-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ to a $(1, \mu)$- or a (j, μ)-symmetric filter is illustrated according to the procedure described in Section 2.8. The arithmetic costs of the original and alternative filters are compared and it is found that the alternative filters require significantly less additions than the original filter.

Filter specification Consider the following filter specification. The non-causal complex filter $H(z)$ should have $(\frac{-4+\mathrm{j}3}{5}, 0)$-symmetry, a passband gain of 0 dB, a passband ripple of 1 dB and a stopband gain of -50 dB. The passband ranges from $\theta = -0.2\pi$ through $\theta = 0.4\pi$ and the stopband ranges from $\theta = 0.6\pi$ through $\theta = 1.6\pi$. The length $L = 17$ filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^{13}}(z)$, with the scaled coefficients listed in Table 2.3, meets the specification.

n	$h'_{\mathrm{r}}[n]$	$h'_{\mathrm{i}}[n]$	$g'_{\beta,\mathrm{r}}[n]$	$g'_{\beta,\mathrm{i}}[n]$	$g'_{\alpha,\mathrm{r}}[n]$	$g'_{\alpha,\mathrm{i}}[n]$
-8	21	-67	-5	-31	-18	-13
-7	82	-114	10	-62	-26	-36
-6	28	-16	8	-12	-2	-10
-5	-315	115	-103	109	3	106
-4	-648	16	-256	136	-60	196
-3	-312	-86	-142	28	-57	85
-2	704	512	384	64	224	-160
-1	1377	1871	925	473	699	-226
0	944	2832	944	944	944	0
1	21	2323	473	925	699	226
2	-256	832	64	384	224	160
3	198	-256	28	-142	-57	-85
4	528	-376	136	-256	-60	-196
5	321	-97	109	-103	3	-106
6	-32	4	-12	8	-2	10
7	-134	-42	-62	10	-26	36
8	-57	-41	-31	-5	-18	13

Table 2.3: *Example 2: Coefficients of the filters* $H'(z)$, $G'_\beta(z)$, $G'_\alpha(z)$.

The magnitude- and phase responses of filter $H(z)$ are presented in Figure 2.5. Clearly the magnitude response is non-symmetric around $\theta = 0$, implying the complex filter coefficients. The phase response toggles between the values $\frac{\mathcal{P}(\sigma)}{2}$ and $\frac{\mathcal{P}(\sigma)}{2} - \pi$ which is consistent with Lemma 2.23 for $\sigma = \frac{-4+\mathrm{j}3}{5}$ and $\mu = 0$.

Filter reduction Because Theorem 2.2 applies to filters with complex-integer coefficients only, a scaled version of the filter $H(z)$ will be used: $H'(z) = 2^{13}H(z) \in \mathbb{C}_\mathbb{Z}(z)$. After completion of the reduction process the original scaling can be restored trivially. The 17 coefficients $h'[n] = h'_\mathrm{r}[n] + \mathrm{j}h'_\mathrm{i}[n] \in \mathbb{C}_\mathbb{Z}$ of filter $H'(z)$ are listed in Table 2.3 column 2 and 3. It can be verified (Step 1) that the shape of symmetry is $\sigma = \frac{-4+\mathrm{j}3}{5}$, the minimal factor of the first kind $\alpha = 1 + \mathrm{j}3$ (Step 2), with $\alpha_\mathrm{r} + \alpha_\mathrm{i}$ is even, and that minimal factor of the second kind $\beta = 2 + \mathrm{j}$ (Step 3), with $\beta_\mathrm{r} + \beta_\mathrm{i}$ is odd. According to Step 5 of the reduction procedure, the filter $G'_\beta(z) = \frac{H'(z)}{\beta} \in \mathbb{C}_\mathbb{Z}(z)$ has $(\mathrm{j}, 0)$-symmetry, which also follows from Theorem 2.3 since $H'_\mathrm{r}|_2(z) = H'_\mathrm{i}|_2(z^{-1})$. The coefficients $g'_\beta[n] = g'_{\beta,\mathrm{r}}[n] + \mathrm{j}g'_{\beta,\mathrm{i}}[n] \in \mathbb{C}_\mathbb{Z}$ of filter $G'_\beta(z)$ are listed in Table 2.3 column 4 and 5.
The procedure does not indicate that reduction to $(1, 0)$-symmetry is also possible (Step 4). However it is straightforward to check that the filters $H'_\mathrm{r}|_2(z)$ and $H'_\mathrm{i}|_2(z)$ both have $(1, 0)$-symmetry, so that also minimal factor of the first kind α is a divisor of filter $H'(z)$, see Theorem 2.3. The coefficients of filter $G'_\alpha(z) = \frac{H'(z)}{\alpha} \in \mathbb{C}_\mathbb{Z}(z)$ are listed in Table 2.3 column 6 and 7.

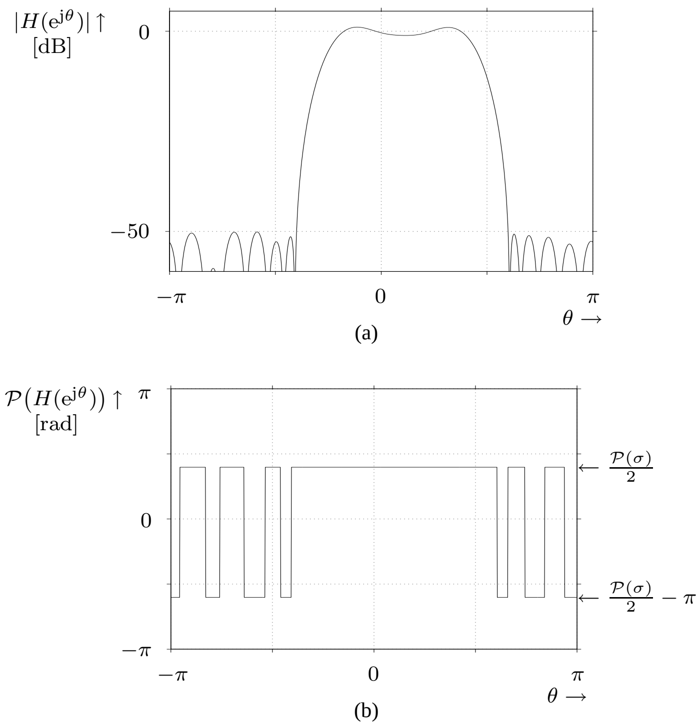


Figure 2.5: *Example 2: Frequency response of filter $H(z)$.*

Cost comparison In general it depends on the filter application whether the filter is best implemented on a general purpose processor or in dedicated hardware. Here it is assumed that the costs of the individual multiplications for the real and imaginary parts of the complex coefficients should be minimized, by using shift-and-add operations only, i.e., the Canonical Signed Digits or CSDs, see [86] [120]. In Appendix E alternatives for the CSDs are discussed in detail.

It can be checked that the 17 complex-integer coefficients of the filters $H'(z)$, $G'_\beta(z)$ and $G'_\alpha(z)$ require 63, 50 and 56 integer additions respectively (not exploiting symmetry). The additions that are part of the transversal filter structure are not counted for. The symmetry in the filters $G'_\beta(z)$ and $G'_\alpha(z)$ can be used to reduce the required number of additions for these sets of coefficients to 25 and 29 respectively. Note that in case the input to the filters is complex, every coefficient has to be used twice. The comparison of costs is not affected by this. The number of additions required for the scale factors β and α, are 2 and 4 respectively.

Besides the application of the reduction procedure, this example also shows that $(1, \mu)$-and (j, μ)-symmetric filters can save in the arithmetic complexity compared to (σ, μ)-symmetric filters, even when symmetry is not exploited. It is also clear that (j, μ)-symmetric filters may be more efficient than $(1, \mu)$-symmetric filters, and therefore are worthwhile to consider when designing filters.

Norms Inspection of Table 2.3 gives:

$$
\begin{aligned}
\|H'(z)\|_1 &= \|H'_\mathrm{r}(z)\|_1 + \|H'_\mathrm{i}(z)\|_1 &&= 5978 + 9600 &&= 15578,\\
\|G'_\beta(z)\|_1 &= \|G'_{\beta,\mathrm{r}}(z)\|_1 + \|G'_{\beta,\mathrm{i}}(z)\|_1 &&= 3692 + 3692 &&= 7384,\\
\|G'_\alpha(z)\|_1 &= \|G'_{\alpha,\mathrm{r}}(z)\|_1 + \|G'_{\alpha,\mathrm{i}}(z)\|_1 &&= 3122 + 1664 &&= 4786,
\end{aligned}
$$

$$
\begin{aligned}
\|H'(z)\|_\infty &= \max\left(\|H'_\mathrm{r}(z)\|_\infty, \|H'_\mathrm{i}(z)\|_\infty\right) &&= \max(1377, 2832) &&= 2832,\\
\|G'_\beta(z)\|_\infty &= \max\left(\|G'_{\beta,\mathrm{r}}(z)\|_\infty, \|G'_{\beta,\mathrm{i}}(z)\|_\infty\right) &&= \max(944, 944) &&= 944,\\
\|G'_\alpha(z)\|_\infty &= \max\left(\|G'_{\alpha,\mathrm{r}}(z)\|_\infty, \|G'_{\alpha,\mathrm{i}}(z)\|_\infty\right) &&= \max(944, 226) &&= 944.
\end{aligned}
$$

It can be checked that $\left\|\frac{1}{\alpha}\right\|_1 = 0.4$ and $\left\|\frac{1}{\beta}\right\|_1 = 0.6$ and that the results are consistent with Theorem 2.4, and also with Lemma A.14 and Lemma A.15. The number of bits needed for a 2-complement representation of the coefficients of the filters $H'(z)$, G'_β and G'_α are 13, 11 and 11 respectively.
End of example

2.10 Filter design

According to [64] only type 5 filters need to be designed, since any other type can be obtained by proper scaling of a type 5 filter, i.e., $(1, \mu)$-symmetric, see also Theorem 2.1. This is stated for symmetric filters with unquantized coefficients, $H(z) \in \mathbb{C}(z)$. In this section it will be shown by using results of the previous sections, that the same is true for symmetric filters with quantized coefficients, $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$.

Definition 2.8 (Filter design specification)**.** *S is the set of all allowed magnitude responses of a filter that has to be designed.*

Next, it is shown how a type 6, 7, 8 or 9 filter $H(z)$, with quantized coefficients and $|H(z)| \in S$, can be designed.

type 6: Design filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ to be type 6, i.e., $(-1, \mu)$-symmetric. First design type 5 filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ with $|G(z)| \in S$, and then take $H(z) = \mathrm{j}G(z)$.

type 7: Design filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ to be type 7, i.e., (j, μ)-symmetric. First design type 5 filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ with $|G(z)| \in \frac{S}{\sqrt{2}}$, and then take $H(z) = (1+\mathrm{j})G(z)$.

type 8: Design filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ to be type 8, i.e., $(-\mathrm{j}, \mu)$-symmetric. First design type 5 filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ with $|G(z)| \in \frac{S}{\sqrt{2}}$, and then take $H(z) = (1-\mathrm{j})G(z)$.

type 9: Design filter $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ to be type 9, i.e., (σ, μ)-symmetric. According to Theorem 2.2, any filter $H(z)$ can be obtained from either a type 5 or a type 7 filter. And a type 7 filter can be obtained from a type 5 filter.

The first possibility is to design a type 5 filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ for use with the minimal factor $\alpha \in \mathbb{C}_\mathbb{Z}$, as $|G(z)| \in \frac{S}{|\alpha|}$. Finally take $H(z) = \alpha G(z)$.

The second possibility is to design a type 5 filter $G(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$ for conversion to a type 7 filter and minimal factor $\beta \in \mathbb{C}_{\mathbb{Z}}$, as $|G(z)| \in \frac{S}{\sqrt{2}|\beta|}$. Finally take $H(z) = (1+\mathrm{j})\beta G(z)$.

A consequence of scaling a type 5 complex-integer filter with a scale factor like $(1+\mathrm{j})$, $(1-\mathrm{j})$, α or $(1+\mathrm{j})\beta$, is that some of the resulting coefficients may have less resolution than allowed. This may be prevented by designing the original type 5 complex-integer filter with increased resolution, then perform the scaling, and finally check the resulting resolution.

2.11 Structural-transmission-zero identities

For real symmetric filters it is well-known that, except for type 1, all other types have transmission zeros, independent of the actual value of the coefficients, they are simple-structural. In this section it is shown that complex symmetric filters have no simple-structural transmission zeros.

Definition 2.9. *z_0 is a* transmission zero *of filter $H(z) \in \mathbb{C}(z)$ iff $H(z_0) = 0$ and $|z_0| = 1$.*

Definition 2.10. *z_0 is a* simple-structural zero *of filter $H(z) \in \mathbb{C}(z)$ iff $H(z_0) = 0$ and z_0 depends only on simple relations (equal or opposite) between the coefficients of $H(z)$.*

2.11.1 Real filters

In this section the presence of simple-structural transmission zeros and the possibility to factorize any type 2, 3 or 4 filter in a cascade connection of a type 1 filter and a filter representing the simple-structural transmission zeros only, will be shown by using Definition 2.1 for symmetry and Definition 2.3 for the types of filters.

Lemma 2.39. *If filter $H(z)$ is a type 1 filter, then filter $H(z)$ has no simple-structural transmission zero.*

Proof. From Lemma 2.26 it is known that $H_{zp}(\theta) = h[\mu] + 2\sum_{n<\mu} h[n]\cos(\theta(\mu - n))$. Since $\mu \in \mathbb{Z}$ the term $h[\mu]$ exists. Let $h[\mu] > 2\sum_{n<\mu} |h[n]|$ then $H(z)$ has no transmission zeros and therefore $H(z)$ has no simple-structural transmission zeros. □

Lemma 2.40. *If filter $H(z)$ is a type 2 filter, then:*

$$H(z) = (1 + z^{-1})G(z) \text{ and filter } G(z) \text{ is a type } 1 \text{ filter}.$$

If filter $H(z)$ is a type 4 filter, then:

$$H(z) = (1 - z^{-1})G(z) \text{ and filter } G(z) \text{ is a type } 1 \text{ filter}.$$

Proof. Filter $H(z)$ is $(\pm1,\mu)$-symmetric with $\mu \in \mathbb{Z}+\frac{1}{2}$ and $H(z) \in \mathbb{R}(z)$. Defining $\mu = k+\frac{1}{2}$ with $k \in \mathbb{Z}$ and using the Definition 2.1 for symmetry, gives $H(z) = \pm z^{-2k-1}H(z^{-1})$. For $z = \mp1$ this results in $H(\mp1) = -H(\mp1)$ implying $H(\mp1) = 0$. Now $H(z)$ can be factorized as $H(z) = (1 \pm z^{-1})G(z)$, with filter $(1 \pm z^{-1})$ is $(\pm1, \frac{1}{2})$-symmetric, filter $G(z)$ is $(1,\mu_G)$-symmetric, $\mu_G = \mu - \frac{1}{2} \in \mathbb{Z}$, see Lemma 2.3, and $G(z) \in \mathbb{R}(z)$. From this it follows that $G(z)$ is a type 1 filter. □

Lemma 2.41. *If filter $H(z)$ is a type 3 filter, then:*

$$H(z) = (1+z^{-1})(1-z^{-1})G(z) = (1-z^{-2})G(z) \text{ and filter } G(z) \text{ is a type 1 filter.}$$

Proof. Filter $H(z)$ is $(-1,\mu)$-symmetric with $\mu \in \mathbb{Z}$ and $H(z) \in \mathbb{R}(z)$. Using the Definition 2.1 for symmetry, gives $H(z) = -z^{-2\mu}H(z^{-1})$. For $z = -1$ this results in $H(-1) = -H(-1)$ implying $H(-1) = 0$ and for $z = 1$ this results in $H(1) = -H(1)$ implying $H(1) = 0$. Now $H(z)$ can be factorized as: $H(z) = (1+z^{-1})(1-z^{-1})G(z)$, with filter $(1+z^{-1})$ is $(1, \frac{1}{2})$-symmetric, filter $(1-z^{-1})$ is $(-1, \frac{1}{2})$-symmetric, filter $G(z)$ is $(1,\mu_G)$-symmetric, $\mu_G = \mu - 1 \in \mathbb{Z}$, see Lemma 2.3, and $G(z) \in \mathbb{R}(z)$. From this it follows that $G(z)$ is a type 1 filter. □

In Figure 2.6 the corresponding cascade connections of filters are shown. The coefficients of filter $G_i(z)$ differ from the coefficients of filter $H_i(z)$ what can be beneficial when cost have to be minimized.

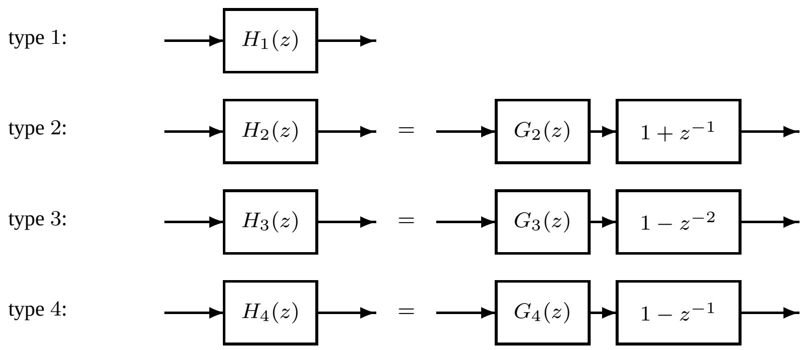


Figure 2.6: *Possible factorization in case of type 1, 2, 3 and 4 real linear-phase filters, applying the structural-transmission-zero identities.*

2.11.2 Complex filters

It is shown that symmetric complex filters have no simple-structural zeros, although the constituting real filters may have.

Lemma 2.42. *Let filter $H(z) \in \mathbb{C}(z)$ be a (σ,μ)-symmetric filter then $H(z)$ has no structural transmission zero.*

Proof. Consider filter $G(z) = \frac{H(z)}{\sqrt{\sigma}}$ that is $(1, \mu)$-symmetric. Both $G(z)$ and $H(z)$ have the same zeros since $|\sqrt{\sigma}| = 1$. Therefore proving that $G(z)$ has no structural transmission zeros is sufficient to proof that $H(z)$ has no structural transmission zeros. By Lemma 2.27, $G(z) = G_\mathrm{r}(z) + \mathrm{j}G_\mathrm{i}(z)$ with $G_\mathrm{r}(z)$ is $(1, \mu)$-symmetric and $G_\mathrm{i}(z)$ is $(-1, \mu)$-symmetric. In case $\mu \in \mathbb{Z}$, the filters $G_\mathrm{r}(z)$ and $G_\mathrm{i}(z)$ are type 1 and type 3 respectively. Type 1 filters have no structural transmission zeros, Lemma 2.39, and therefore filter $G(z)$ has no structural transmission zeros. In case $\mu \in \mathbb{Z} + \frac{1}{2}$, the filters $G_\mathrm{r}(z)$ and $G_\mathrm{i}(z)$ are type 2 and type 4 respectively. Type 2 filters have a structural transmission zero at $z = -1$ and type 4 filters have a structural transmission zero at $z = 1$, Lemma 2.40. Both zeros are different and therefore filter $G(z)$ has no structural transmission zeros. ☐

Of course a complex symmetric filter can have transmission zeros but these are fully determined by the values of the coefficients and not by possible shape of symmetry or center of symmetry.

2.12 Conclusion

A generalized definition of symmetry for FIR filters, with real or complex-valued coefficients, has been presented and discussed abundantly. This (σ, μ)-symmetry, with shape of symmetry σ, $\sigma \in \mathbb{C}$ and $|\sigma| = 1$, and center of symmetry μ, $\mu \in \mathbb{Z}/2$, is applicable to the 4 well-known types of linear-phase filters with real-valued coefficients, and treats even- and odd-length filters in a unified manner. With respect to the filters with complex-valued coefficients, 5 new types have been defined. The type 5, 6, 7 and 8 symmetric filters, with $\sigma = 1$, $\sigma = -1$, $\sigma = \mathrm{j}$ and $\sigma = -\mathrm{j}$ respectively, exhibit simple relations between the individual real and imaginary parts of the complex coefficients, such that the symmetry and therefore the linear-phase property are simple structurally guaranteed. These simple relations are invariant under quantization and as a consequence implementations can be made simpler. In Chapter 4, the transversal structures that exploit symmetry are addressed in detail.
In close relation to symmetry the concept of mirroring is introduced. It is shown that two mutually mirrored filters with a common input or output can be replaced by two symmetric filters and a combination network, and vice versa. This is an important property that is key in the process of restoring symmetry in polyphase structures as treated in Chapter 5. Furthermore a new theorem is presented that states that any type 9 symmetric filter with complex-integer-valued coefficients, can be reduced over the complex integers into a type 5, 6, 7 or 8 symmetric filter. Based on this property it is also shown that any type 6, 7, 8 or 9 symmetric filter, with complex-integer-valued coefficients, can be designed by first designing a type 5 filter with complex-integer-valued coefficients, and an appropriate complex-integer scale factor. Finally it is shown that complex (σ, μ)-symmetric filters do not have simple-structural transmission-zeros, unlike the real symmetric filters.

Chapter 3

First- and second-order filters

Complex filters in general, $H(z) \in \mathbb{C}(z)$, have the possibility to position their zeros independently at any place in the z-plane. In case the filter coefficients are quantized like $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, and limited in magnitude like $\|H(z)\|_\infty \leq \Xi$, the possible zero-locations are limited too.
This chapter discusses the possibilities of first- and second-order complex filters, to place their zeros in the z-plane, under the restrictions as mentioned before, in more detail. In addition it is shown that the newly defined (j, μ)-symmetric complex filters, may be beneficial over the $(1, \mu)$-symmetric complex filters depending on the given specification. The first- and second-order filters are treated in Section 3.1 and Section 3.2 respectively.

3.1 First order

In general, the system function of a first-order complex FIR filter $H(z) \in \mathbb{C}(z)$ is given by:

$$H(z) = h[0] + h[1]z^{-1}.$$

The location of the single zero can be anywhere in the z-plane, and is given by: $z_0 = -\frac{h[1]}{h[0]}$. For $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$, and a given *coefficient range* Ξ, so $\|H(z)\|_\infty \leq \Xi$, the possible zero-locations, z_0, are limited. In Figure 3.1 the possible locations are shown for the filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ and a set of coefficient ranges, viz., $\Xi \in \{1, 2, 4, 8\}$. Only the zero-locations in the first quadrant of the z-plane and not outside the unit circle are shown, since the other are similar, see Lemma 3.1.

Lemma 3.1. *If z_0 is a zero from the filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, then: z_0^* is a zero from filter $H^{\overline{*}}(z)$, $-z_0$ is a zero from filter $H(-z)$ and z_0^{-1} is a zero from filter $H(z^{-1})$, with $\|H(z)\|_\infty = \|H^{\overline{*}}(z)\|_\infty = \|H(-z)\|_\infty = \|H(z^{-1})\|_\infty$ and $H^{\overline{*}}(z)$, $H(-z)$, $H(z^{-1}) \in \mathbb{C}_{\mathbb{Z}}(z)$.*

Proof. The effect of the various operations on the zero-locations can be easily verified. The operations also leave the ∞-norm invariant, and map the coefficients from $\mathbb{C}_{\mathbb{Z}}$ into $\mathbb{C}_{\mathbb{Z}}$. $\square$

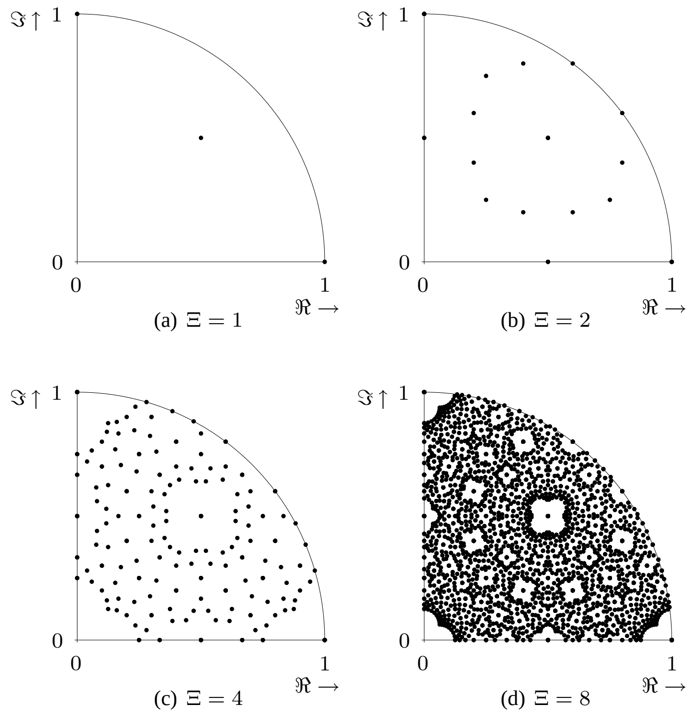


Figure 3.1: *Possible zero-locations of first-order filter* $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ *with* $\|H(z)\|_\infty \leq \Xi$ *for* $\Xi \in \{1, 2, 4, 8\}$.

From these examples it follows that if the values of the individual real and imaginary parts of the coefficients are limited to $h_{\mathrm{r}}[0], h_{\mathrm{i}}[0], h_{\mathrm{r}}[1], h_{\mathrm{i}}[1] \in \{0, -1, 1, -\mathrm{j}, \mathrm{j}\}$, i.e., $\Xi = 1$, the zeros are positioned at $\theta = 0$ and $\frac{\pi}{2}$ on the unit circle or at $\theta = \frac{\pi}{4}$ inside the unit circle. For larger values of Ξ more zero-locations are found, but at a relatively large distance from zero-locations found for small values of Ξ. This behaviour is similar for zeros z_0 both inside and on the unit circle.
Of particular interest are the zeros located on the unit circle, since these introduce a large attenuation in their vicinity and therefore are of value when designing low cost filters. First, a lemma about the symmetry of first-order filters with their zero on the unit circle is presented.

Lemma 3.2. *The filter* $H(z) = h[0] + h[1]z^{-1}$ *with* $H(z) \in \mathbb{C}(z)$ *has its zero,* z_0*, on the unit circle iff that filter* $H(z)$ *is* $(\sigma, \frac{1}{2})$*-symmetric, with* $\sigma = \frac{h[1]}{h^*[0]}$ *and* $|\sigma| = 1$.

Proof. The proof follows directly from the fact that for a zero to be on the unit circle holds: $|z_0| = 1$ or $|h[0]| = |h[1]|$, and from the definition for symmetry. □

The position of that single zero, z_0, on the unit circle, i.e., the angle or relative frequency θ_0, is given in the next lemma.

Lemma 3.3. *The* $(\sigma, \frac{1}{2})$*-symmetric filter* $H(z) = h[0] + \sigma h^*[0]z^{-1}$ *has its zero at:*

$$\theta_0 = \mathcal{P}(\sigma) + 2\arctan\Big(\frac{h_\mathrm{r}[0]}{h_\mathrm{i}[0]}\Big).$$

Proof. For the zero $z_0 = -\frac{\sigma h^*[0]}{h[0]}$ the relative frequency is:

$$\begin{aligned}\theta_0 &= \mathcal{P}(z_0),\\ &= \pi + \mathcal{P}(\sigma h^*[0]) - \mathcal{P}(h[0])\\ &= \pi + \mathcal{P}(\sigma) - 2\arctan\Big(\frac{h_\mathrm{i}[0]}{h_\mathrm{r}[0]}\Big)\\ &= \mathcal{P}(\sigma) + 2\arctan\Big(\frac{h_\mathrm{r}[0]}{h_\mathrm{i}[0]}\Big).\end{aligned}$$ □

For the type 5 filter $H(z) \in \mathbb{C}_\mathbb{Z}(z)$, $H(z) = h[0] + h^*[0]z^{-1}$, the possible relative frequencies θ_0 of the zero z_0 are determined for a range of coefficient ranges Ξ, such that $\|H(z)\|_\infty \leq \Xi$, see Figure 3.2(a). Only the relative frequencies in the first quadrant, $0 \leq \theta_0 \leq \frac{\pi}{2}$, are shown, since the other are similar, see Lemma 3.1.
It is trivial that if a particular θ_0 is possible for a particular coefficient range Ξ_0, the same θ_0 is also possible for any coefficient range $\Xi \geq \Xi_0$. This explains the vertical lines in the plot. From this plot it can in principle be derived what the minimal ∞-norm of a filter is, if a zero should be located at a particular relative frequency.
From this plot it is also clear that the value of Ξ has to be increased significantly to obtain a relative frequency close to one obtained for a small value of Ξ. Note that an increased range or resolution generally implies an increased cost of the implementation.
In the plot of Figure 3.2(a), clear envelopes are recognized, some of these are presented with dashed lines. Envelopes that relate to $h[0] \in \{\mathrm{j}, 1{+}\mathrm{j}, 1{+}\mathrm{j}2, 1{+}\mathrm{j}3, 2{+}\mathrm{j}3, 1{+}\mathrm{j}4, 3{+}\mathrm{j}4\}$ are plotted. The plot in Figure 3.2(a), clearly shows that $\theta_0 = \frac{\pi}{2}$ can be approached better for a particular maximum coefficient range than $\theta_0 = 0$. An expression for these envelopes is given in Lemma 3.5, but first an extra lemma is introduced.

Lemma 3.4. *For the given ratio* $\frac{y_0}{x_0}$*, the error of the approximation ratio* $\frac{y}{x} \neq \frac{y_0}{x_0}$*, is bounded by:*

$$\Big|\frac{y}{x} - \frac{y_0}{x_0}\Big| \geq \frac{1}{\Xi x_0},$$

assuming that: $y_0, y \in \mathbb{Z}$*,* $x_0, x \in \mathbb{N}^+$ *and* $x_0, |y_0|, x, |y| \leq \Xi$.

Proof. Since $x_0 y - x y_0 \in \mathbb{Z}\backslash\{0\}$, for the error holds: $\big|\frac{y}{x} - \frac{y_0}{x_0}\big| = \frac{|x_0 y - x y_0|}{x x_0} \geq \frac{1}{x x_0} \geq \frac{1}{\Xi x_0}$. In case $\gcd(x_0, y_0) = 1$ there exist x and y such that $|x_0 y - x y_0| = 1$ (Euclid). □

The envelopes are described in the next lemma.

Lemma 3.5. *The envelopes for the type* 5 *filter around* $\theta_0 = 2\arctan\big(\frac{h_\mathrm{r}[0]}{h_\mathrm{i}[0]}\big)$ *with* $\gcd(h_\mathrm{r}[0], h_\mathrm{i}[0]) = 1$*, are:*

$$\Xi(\theta) = \big|h_\mathrm{i}[0]\tan(\tfrac{\theta}{2}) - h_\mathrm{r}[0]\big|^{-1}.$$

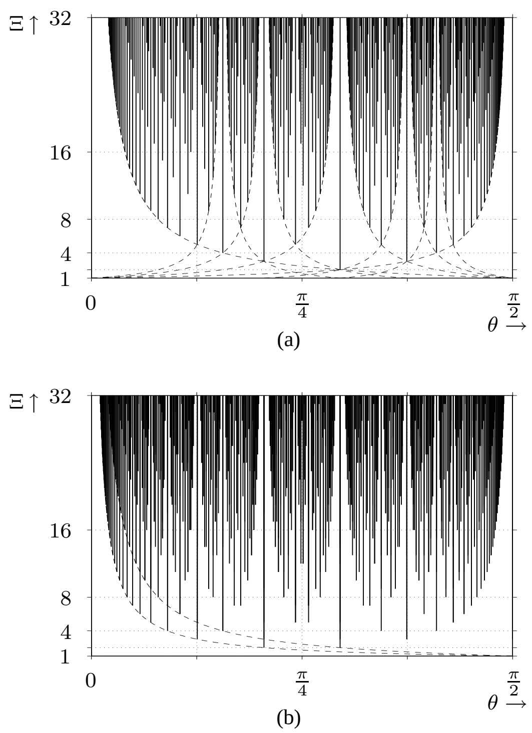


Figure 3.2: *Possible relative frequencies of the zeros on the unit circle of first-order complex $H(z)$, shown for type 5 filters in (a) and type 5, 6, 7 and 8 filters in (b).*

Proof. Since the $\arctan$-function is monotonic, and based on Lemma 3.3 and Lemma 3.4, the two bounds can be described with:

$$\theta = 2\arctan\Big(\frac{h_{\rm r}[0]}{h_{\rm i}[0]} \pm \frac{1}{h_{\rm i}[0]\Xi}\Big),$$

what is equivalent to the expression in the lemma using $\Xi \geq 0$. □

It is easy to see that the plots for type 5 and 6 filters are identical, use the substitution $H(z) \leftarrow {\rm j}H(z)$. Similarly, the plots for the type 7 and 8 filters are identical and can be obtained from the plot for type 5 by mirroring around $\theta = \frac{\pi}{4}$.
The zeros on the unit circle that can be realized with a particular value of Ξ, with either a type 5, 6, 7 or 8 filter, are depicted in Figure 3.2(b). The envelopes for $\theta_0 = 0$ in case of type 5 (6) and type 7 (8) filters are plotted too. This plot now clearly shows that for zero locations close to $\theta_0 = 0$, type 7 filters can deal with a smaller coefficient range than type 5 filters can. This is another example that shows the relevance of type 7 (and 8) filters. From Theorem 2.2 and Theorem 2.4, it follows that the required coefficient range for a type 9 filter is at least the coefficient range of a type 5, 6, 7 or a type 8 filter. As a consequence the type 9 filters are not relevant to consider here.

The amplitude responses of all first-order type 5 filters are identical except for a scale factor and a frequency shift, where the frequency shift is determined by the location of the zero. For a given application it may be necessary to realize a particular attenuation at a particular frequency. As illustrated in Figure 3.1 and Figure 3.2, the possibilities to locate zeros on the unit circle strongly depend on the allowed coefficient range. As a consequence, the maximum attenuation that can be realized at a particular frequency depends on the allowed coefficient range. This all is illustrated for a type 5 filter in Figure 3.3. Consider the complex-integer, first-order $(1,\mu)$-symmetric filter i, with coefficient range Ξ, i.e., $H_{\Xi,i}(\mathrm{e}^{\mathrm{j}\theta}) \in \mathbb{C}_{\mathbb{Z}}(z)$ and $\|H_{\Xi,i}(z)\|_\infty \leq \Xi$. First all filters are normalized to have a maximum gain equal to 1 (0 dB), to enable comparison:

$$\overline{H}_{\Xi,i}(\mathrm{e}^{\mathrm{j}\theta}) = \frac{H_{\Xi,i}(\mathrm{e}^{\mathrm{j}\theta})}{\max_\theta \left|H_{\Xi,i}(\mathrm{e}^{\mathrm{j}\theta})\right|}.$$

Finally, the minimal possible gain, or the maximal possible attenuation, over all filters i is determined as:

$$\widehat{H}_{\Xi}(\mathrm{e}^{\mathrm{j}\theta}) = \min_i \left|\overline{H}_{\Xi,i}(\mathrm{e}^{\mathrm{j}\theta})\right|.$$

Now, for a set of coefficient ranges the minimal possible gains are depicted in Figure 3.3. From this all it is immediately clear that for a desired attenuation the required coefficient range strongly depends on the frequency for which that attenuation has to be realized.

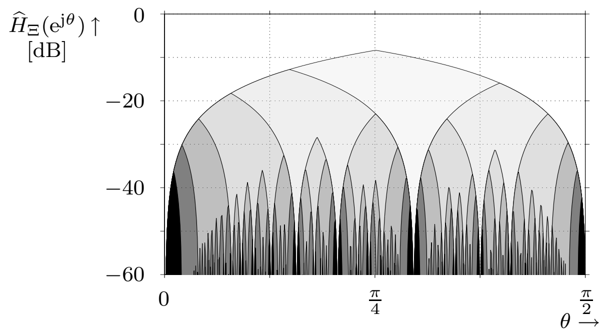


Figure 3.3: *Minimal possible gain for the first-order type 5 filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$ with $\Xi \in \{1, 2, 4, 8, 16, 32\}$. A darker area relates to a larger Ξ.*

3.2 Second order

Similarly as for the first-order case, second-order FIR filters with both zeros on the unit circle are considered. For a given maximum coefficient range, the possibilities to locate zeros on the unit circle are evaluated. In general the system function of a second-order complex FIR filter $H(z) \in \mathbb{C}(z)$ is given by:

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2}.$$

The location of the zero-pairs can be anywhere in the z-plane, and is given by:

$$(z_0, z_1) = \frac{-h[1] \pm \sqrt{h^2[1] - 4h[0]h[2]}}{2h[0]}.$$

For $H(z) \in \mathbb{C}_{\mathbb{Z}/2^i}(z)$, and a given Ξ for the coefficient range, the possible zero-locations, (z_0, z_1), are limited.
By the fact that a first-order filter with its zero located on the unit circle is $(\sigma_0, \frac{1}{2})$-symmetric, see Lemma 3.2 and Lemma 2.3, any second-order FIR filter with both zeros on the unit circle is $(\sigma, 1)$-symmetric.
The type 6 and 8 filters can be obtained by multiplying type 5 and 7 filters respectively, with the factor j, that does not change the possible zero-pairs nor the coefficient range. Similarly as for the first-order filters, by Theorem 2.2 and Theorem 2.4 the coefficient range of type 9 filters is at least the coefficient range of the type 5, 6, 7 or a type 8 filters. As a consequence the type 9 filters are not relevant in this case.

Lemma 3.6. *Let $H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} \in \mathbb{C}(z)$ be $(\sigma, 1)$-symmetric and (θ_0, θ_1) be the relative frequencies of the zero-pair (z_0, z_1) on the unit circle, then:*

$$\theta_0 + \theta_1 = \arctan\Big(\frac{\sigma_\mathrm{i}}{\sigma_\mathrm{r}}\Big) - 2\arctan\Big(\frac{h_\mathrm{i}[0]}{h_\mathrm{r}[0]}\Big),$$
$$\theta_0 - \theta_1 = 2\arccos\Big(\frac{h[1]}{2\sqrt{\sigma}|h[0]|}\Big) \text{ note that } \frac{h[1]}{\sqrt{\sigma}} \in \mathbb{R}.$$

Proof. Since $H(z)$ is $(\sigma, 1)$-symmetric introduce $a = h[0]$ and $b = \frac{h[1]}{\sqrt{\sigma}}$ with $a \in \mathbb{C}$ and $b \in \mathbb{R}$, see Lemma 2.1 and Lemma 2.2, giving: $H(z) = a + b\sqrt{\sigma}z^{-1} + a^*\sigma z^{-2}$. Now the general expression for (z_0, z_1) can be rewritten as:

$$(z_0, z_1) = \frac{\sqrt{\sigma}}{2a}\Big(-b \pm \mathrm{j}\sqrt{4|a|^2 - b^2}\Big),$$

with $4|a|^2 \geq b^2$ placing both zeros on the unit circle. The related relative frequencies are:

$$(\theta_0, \theta_1) = \frac{1}{2}\arctan\Big(\frac{\sigma_\mathrm{i}}{\sigma_\mathrm{r}}\Big) - \arctan\Big(\frac{a_\mathrm{i}}{a_\mathrm{r}}\Big) \mp \arctan\Big(\frac{\sqrt{4|a|^2 - b^2}}{b}\Big).$$

Using the identity $\cos(x) = \frac{1}{\sqrt{1+\tan^2(x)}}$, and the expressions for a and b, concludes the proof. □

In Figure 3.4 the possible pairs of frequencies where the zeros can be located on the unit circle, (θ_0, θ_1), are shown for $H(z) \in \mathbb{C}_\mathbb{Z}(z)$ and $\Xi \in \{1, 2, 4\}$. The left column relates to type 5 and 6 filters, and the right column to type 7 and 8 filters. From this plots it can be observed that for the same Ξ-value, type 5 and 7 filters can have different pairs (θ_0, θ_1).

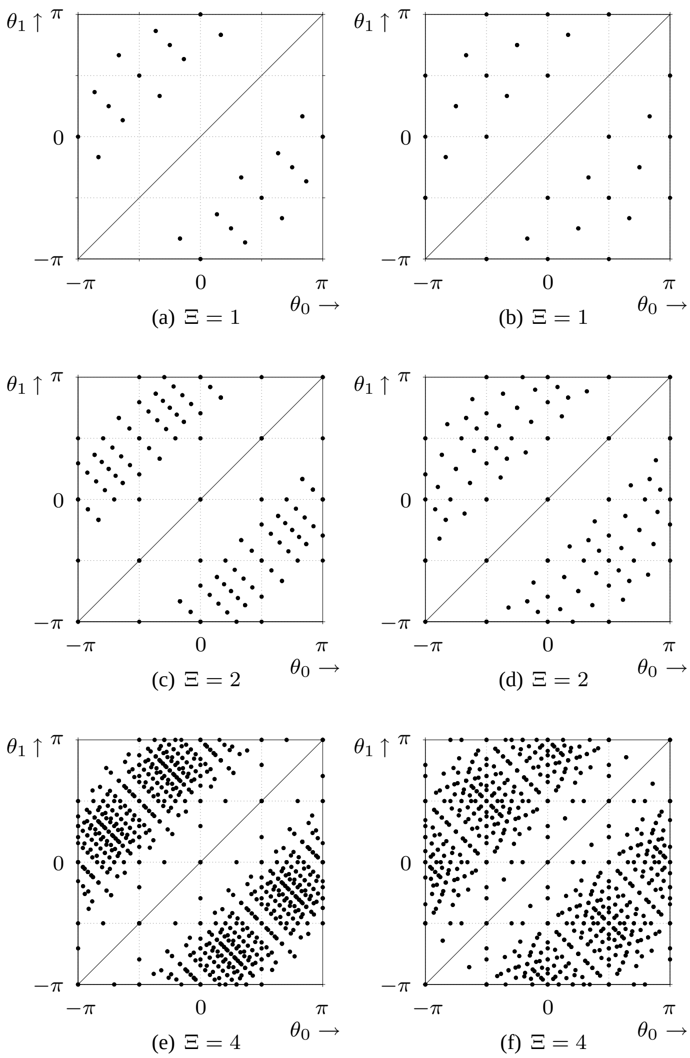


Figure 3.4: *Possible zero locations* (θ_0, θ_1) *on the unit circle for a second-order filter with complex-integer coefficients, type* 5 *and* 6 *(left), and type* 7 *and* 8 *(right).*

Also, the patterns for type 5 and type 7 are not mirrored versions as in the first-order case. The values of $\theta_0 + \theta_1$, however, are shifted over $\frac{\pi}{4}$ (evaluate Lemma 3.6 for $\sigma = 1$ and $\sigma = \mathrm{j}$).

From comparing Figure 3.2 and Figure 3.4 it is clear that for the same Ξ-value, a second-order filter has more possibilities to place zeros on the unit circle than a cascade of two first-order filters.

To make filters with wider stopbands the zeros should be close, i.e., small values for $|\theta_0 - \theta_1|$. Except for a limited number of double zeros, $\theta_0 = \theta_1$, the minimum distance between zeros, $|\theta_0 - \theta_1|$, is relatively large. In Figure 3.5 these minimum distances are plotted for a range of Ξ.

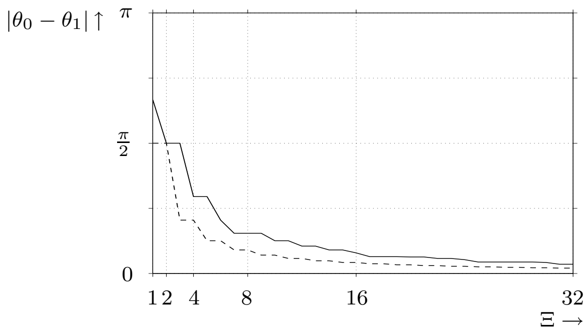


Figure 3.5: *Minimum distance between zeros on the unit circle for a second-order filter, except double zeros, for type* 5 *and* 6 *(solid line), and type* 7 *and* 8 *(dashed line) filters.*

From the plot it follows that for type 5 and 6 filters the coefficient range should be about twice the coefficient range for type 7 and 8 filters, when two zeros should be placed at the same distance.

3.3 Conclusion

In this chapter the possibilities of first-order and second-order symmetric filters with quantized coefficients, to position their zeros on the unit circle, have been elaborated. It has been shown that for a given coefficient range, the type 7 and 8 filters can place their zeros at different locations than the type 5 and 6 filters can. Also for the second-order filters, the type 7 and 8 filters can place their zeros closer to each other than the type 5 and 6 filters can.

Chapter 4

Transversal and complex structures

Among the various FIR filter structures like the lattices and ladders, the transversal structure is popular for many reasons, e.g.:

- The coefficients are identical to the elements of the impulse response, which is an important issue in the filter design process, and also can simply guarantee the linear-phase property, see Chapter 2.
- The coefficients are invariant under the polyphase decomposition for multirate filters, see Chapter 5. The subfilters or polyphase components (PPCs) resulting from this conversion, again can be transversal filters.
- In case of high speed applications, i.e., a high sampling rate, the transversal filter is very suitable, since the method of pipelining can be incorporated in a straightforward way, see Appendix B.

Of course, the transversal filter can also be used as a building block in composed filter structures, e.g., cascades of filters [43], Interpolated FIR (IFIR) structures [98] and frequency masking structures [83].
For the purpose of making filter structures more efficient in terms of costs, this chapter shows how (σ, μ)-symmetry can appear in the transversal structure, and how it can be exploited. This is an overview of known structures and structures inspired by the novel definition for symmetry. The fact that two filters have inputs or outputs in common can be exploited too, and various alternatives to decompose complex filters or coefficients into their individual real and imaginary parts, are discussed and compared in detail. Also new structures for efficiently combining conjugate coefficients have been found and subsequently involved in a detailed comparison of computational costs of filters.
In this chapter, first the two well-known transversal structures are presented in Section 4.1. In these structures both the signals and the filter components, i.e., the delay elements, adders and multipliers, are considered to be complex. Based on these two structures some

special structures are treated in case a filter is complex or real (σ, μ)-symmetric, or two filters have a common input or output, see Section 4.2, Section 4.3 and Section 4.4 respectively. Although the filters and their input and output signals may be assumed to be complex, for an actual implementation of such filter structure somehow a translation has to be made to real components. In the remainder of this chapter two possible approaches are supported. The first approach is presented in Section 4.5, where the complex filters and signals are decomposed into their individual real and imaginary parts. The resulting subfilters (their constituting components) and their signals are now assumed to be real. Also some alternative decompositions are discussed. Some alternative structures for (σ, μ)-symmetric filters are discussed in Section 4.6. The second approach is that the complex components of the filters are decomposed into their individual real and imaginary parts. The resulting subcomponents and their signals are now assumed to be real. Complex additions, subtractions and delays are trivially constructed of 2 real additions, subtractions and delays of the real and imaginary signal parts respectively. For the complex coefficients, the notion is used that the complex scale factors form a subset from the complex filters, and hence the alternative structures for complex filters, as discussed before, can be used for the complex scale factors too. This notion is also used in Appendix E to find low cost constructions for coefficients. In Section 4.7, new efficient structures for complex conjugated coefficients with a common input or output are presented for use with symmetric filters. The several methods to implement a complex, possibly symmetric, transversal filter structure as discussed in this chapter, are compared in Section 4.8 with respect to the required number of real multiplications and real additions.
In this chapter many filter structures are presented by means of an example scheme that by its nature is just one of the many possibilities. However from these examples it is clear how the structure can be adapted to the problem at hand.

4.1 Transversal structures

The z-domain description or system function of a FIR filter $H(z) \in \mathbb{C}(z)$, is a polynomial in z^{-1}, like:

$$H(z) = \sum_n h[n] z^{-n},$$

with $h[n] \in \mathbb{C}$ for all n. For the filter $H(z)$ with all its coefficients outside the range $0 \leq n < L$ equal to zero, the polynomial can be rewritten as:

$$\begin{aligned} H(z) &= \sum_{n=0}^{L-1} h[n] z^{-n} \\ &= h[0] + z^{-1}\left(h[1] + z^{-1}\left(h[2] + z^{-1}\left(\cdots + z^{-1} h[L-1]\right)\right)\right), \end{aligned}$$

which is an efficient form for evaluating polynomials according to Horner's scheme [138]. From this scheme it is known that it requires the minimal number of multiplications with z^{-1}, which corresponds to a minimal number of delay elements. Two different *transversal* filter structures can be derived. In the first structure the output signal of the transversal filter can be seen as the weighted sum of delayed versions of the input signal. In the second structure the output signal can be seen as the delayed sum of weighted versions of the input signal. The weighting factors are the filter coefficients. For a further explanation consider the following example.

Example 4.1. Assume the filter $H(z) \in \mathbb{C}(z)$ to be:

$$\begin{aligned} H(z) &= \langle a, b, c, d, e \rangle \\ &= a + z^{-1}(b + z^{-1}(c + z^{-1}(d + z^{-1}e))). \end{aligned}$$

It is easy to check that the structures from Figure 4.1(a) and Figure 4.1(b) implement filter $H(z)$. Note that the signals as well as the components (delays, adders and multipliers) may be complex.

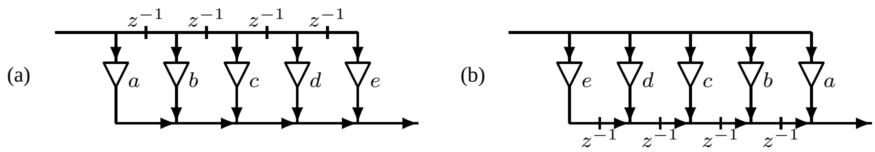


Figure 4.1: *Two alternative transversal filter structures: tapped delay line (a) and adding delay line (b).*

End of example

The delay line as used in Figure 4.1(a) is called a *tapped delay line*, whereas the delay line as used in Figure 4.1(b) is called an *adding delay line*. An advantage in case of a hardware implementation of the adding delay line, is the so called *free pipelining*. See Appendix B for a brief introduction to the concept of pipelining.
In general, the complex transversal filter, with a tapped or adding delay line, requires per input sample L complex multiplications of complex data with a complex coefficient, $L-1$ complex additions and $L-1$ complex delay elements. In case of symmetry, combining coefficients may reduce arithmetic costs, which will be discussed in the next two sections.

4.2 Complex (σ, μ)-symmetric transversal

Any FIR filter $H(z) \in \mathbb{C}(z)$, can be implemented by means of the transversal structures as shown in Figure 4.1. In case of a (σ, μ)-symmetric filter however, some special constructions are possible. In this section these constructions are discussed and explained by means of some examples.
From Lemma 2.5 it follows that any (σ, μ)-symmetric filter $H(z)$ can be factorized in the factor $\sqrt{\sigma}$ and a $(1, \mu)$-symmetric filter. In the following example this is illustrated for an $L = 5$ filter.

Example 4.2. By Lemma 2.5 any (σ, μ)-symmetric filter $H(z)$ can be factorized as:

$$H(z) = \sqrt{\sigma}G(z) \text{ and filter } G(z) \text{ is } (1, \mu)\text{-symmetric},$$

with $H(z), G(z) \in \mathbb{C}(z)$. As an example, take $H(z) = \langle p, q, r, s, t\rangle$ with $p, q, r, s, t \in \mathbb{C}$. Now $H(z) = \sqrt{\sigma}\langle a, b, c, b^*, a^*\rangle$ with $a, b \in \mathbb{C}$, $c \in \mathbb{R}$ and:

$$p = \sqrt{\sigma}a,\ q = \sqrt{\sigma}b,\ r = \sqrt{\sigma}c,\ s = \sqrt{\sigma}b^*,\ t = \sqrt{\sigma}a^*.$$

In Figure 4.2 this factorization is illustrated using the transversal structure with a tapped delay line. Like in Example 4.1, an adding delay line can be used to obtain free pipelining (not shown).

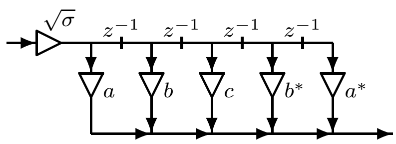


Figure 4.2: *Factorization of the (σ, μ)-symmetric filter $H(z)$ in scale factor $\sqrt{\sigma}$ and $(1, \mu)$-symmetric filter $G(z)$.*

End of example

Lemma 2.1 about the impulse response of a (σ, μ)-symmetric filter $H(z)$, inspires an alternative transversal-like filter structure. In this structure, that is illustrated in Example 4.3, the coefficients appear in conjugated pairs and the shape of symmetry, σ, is incorporated in the delay line, or alternatively in the taps.

Example 4.3. For the (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$, with $H(z) = \langle a, b, \sqrt{\sigma}c, \sigma b^*, \sigma a^*\rangle$, where $a, b \in \mathbb{C}$ and $c \in \mathbb{R}$, Figure 4.3(a) shows the transversal-like structure with the extra multipliers in the delay line, and Figure 4.3(b) shows the transversal-like structure with the extra multipliers in the taps. Both schemes have a tapped delay line. Like in Example 4.1 an adding delay line can be used to obtain free pipelining (not shown).

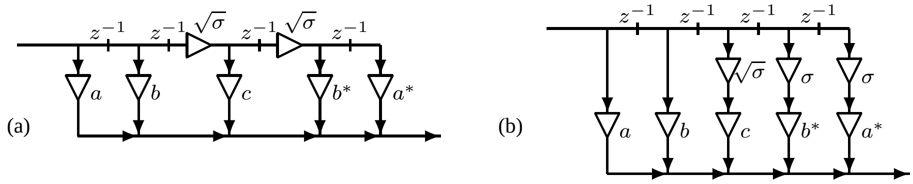


Figure 4.3: *Transversal-like structures for (σ, μ)-symmetric filters.*

End of example

The structures for other values of L, can easily be derived from these examples. For even-length filters, $\mu \in \mathbb{Z}$, the central multiplier, c, and one of its neighboring delay elements, are absent. The extra multipliers, σ or $\sqrt{\sigma}$, make this structure different from a traditional transversal structure.

In case two coefficients in the transversal structure, both the tapped and adding delay line, are equal or opposite, the two related multipliers can be combined into a single multiplier, on basis of the distributive property, so reducing arithmetic costs, e.g., [116].

From the schemes in Figure 4.2 and Figure 4.3 it follows that in general coefficients appear in conjugated pairs. As will be explained in Section 4.7, it is also possible to combine these conjugated multiplications into a single structure that has costs comparable to the costs of a single complex multiplier. In Figure 4.4 the new symbols for the combined conjugated pairs of coefficients, are introduced. The $*$ denotes the terminal that is associated with the conjugated version of the coefficient.

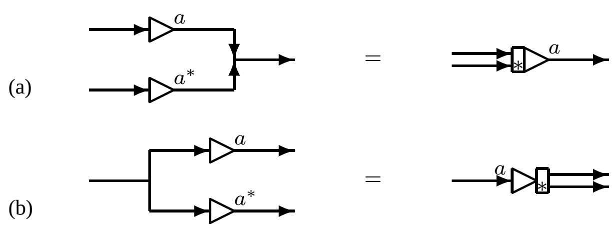


Figure 4.4: *Conjugated pairs of coefficients with common outputs (a) and common inputs (b).*

Example 4.4. The same filter as in Example 4.3 is used, i.e., the (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$ with $H(z) = \langle a, b, \sqrt{\sigma}c, \sigma b^*, \sigma a^* \rangle$, where $a, b \in \mathbb{C}$ and $c \in \mathbb{R}$. The two pairs of conjugated coefficients (a, a^*) and (b, b^*) are combined. The resulting scheme is presented for a tapped delay line Figure 4.5(a), and for an adding delay line Figure 4.5(b). Note the special component for multiplication with two combined conjugated coefficients.

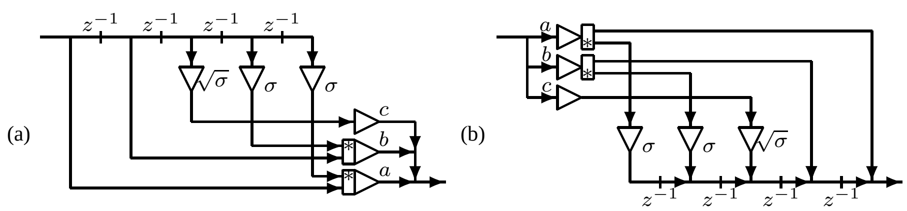


Figure 4.5: *Transversal-like structures for (σ, μ)-symmetric filters with combined conjugated coefficient.*

End of example

4.3 Real (σ, μ)-symmetric transversal

For the real (σ, μ)-symmetric filters $H(z) \in \mathbb{R}(z)$, Lemma 2.25 gives that the shape of symmetry is limited to $\sigma \in \{-1, 1\}$. Also, by Lemma 2.1 the relation $h[n] = \sigma h[2\mu - n]$ holds. This allows for an efficient implementation by means of a transversal structure where multiplications of signals with coefficients are combined. The savings can be up to half the number of multiplications. For a length L, (σ, μ)-symmetric filter $H(z) \in \mathbb{R}(z)$, and input signal $X(z) \in \mathbb{R}(z)$, the number of real multiplications is approximately $\frac{L}{2}$. In Example 4.5 this concept is illustrated for two (σ, μ)-symmetric real filters.

Example 4.5. Consider the two $L = 5$ filters $H(z), G(z) \in \mathbb{R}(z)$ that are $(-1, 2)$ and $(1, 2)$-symmetric respectively. So, with $a, b, c \in \mathbb{R}$:

$$H(z) = \langle a, b, 0, -b, -a \rangle,$$
$$G(z) = \langle a, b, c, b, a \rangle.$$

In Figure 4.6(a) the scheme related to $H(z)$ is shown with a tapped delay line, and in Figure 4.6(b) the scheme for $G(z)$ is shown with an adding delay line. It is easy to verify that these structures implement the filters $H(z)$ and $G(z)$ respectively.

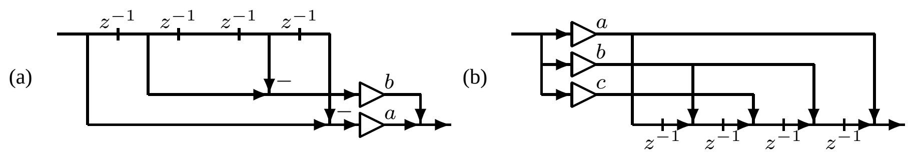


Figure 4.6: *Transversal structures with combined coefficients.*

End of example

4.4 Multiple input or output transversal structures

In many cases, in which the same signal is fed into N transversal filters, or where the outputs of N transversal filters are added, schemes can be implemented more efficiently than taking N of the systems from Figure 4.1. These configurations occur as subsystems in complex filters, for instance, see Section 4.5 where the individual real and imaginary parts of the complex filter may have a common input or output. Also in multirate filters, see Chapter 5, subfilters in the polyphase structure may have common inputs or outputs.

In this section it is illustrated by means of some typical examples, how transversal filters may be combined. From these examples the structures for different filter lengths can be derived easily. Choosing tapped or adding delay lines can have consequences, and also issues like the possible sharing of multiplications or delay lines and free pipelining, are addressed.

4.4.1 General filters

Two filters $H(z), G(z) \in \mathbb{C}(z)$ with a common input or output, can be described in vector notation as:

$$\begin{bmatrix} H(z) \\ G(z) \end{bmatrix} \text{ or } \begin{bmatrix} H(z) & G(z) \end{bmatrix} \text{ respectively.}$$

For a common input, and both filters implemented with a tapped delay line, the two delay lines can be combined. With an adding delay line such combination is not possible. In case of a common output, and both filters implemented with a tapped delay line, the two delay lines cannot be combined. With an adding delay line however, such combination is possible and the pipelining is free. In Example 4.6 this is illustrated. From the example it directly follows how more than 2 filters can be combined.

Example 4.6. To illustrate the alternatives for combining two filters $H(z), G(z) \in \mathbb{C}(z)$ with a common input or output, these filters are chosen as:

$$H(z) = \langle a, b, c, d, e \rangle,$$
$$G(z) = \langle p, q, r, s, t \rangle.$$

In Figure 4.7(a), for a common input, two tapped delay lines are combined and in Figure 4.7(b) the adding delay lines cannot be combined.

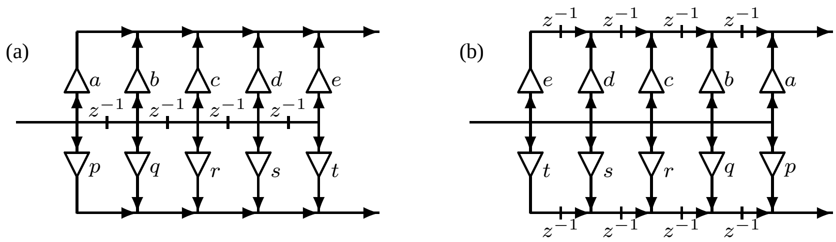


Figure 4.7: *Single input, dual output transversal structures.*

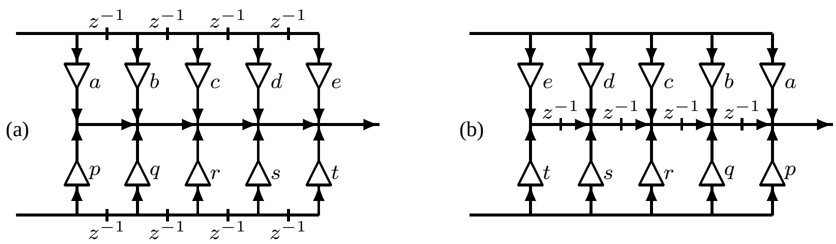


Figure 4.8: *Dual input, single output transversal structures.*

In Figure 4.8(a), for a common output, two tapped delay lines cannot be combined and in Figure 4.8(b) the adding delay lines, with free pipelining, are combined.
End of example

4.4.2 Complex (σ, μ)-symmetric filters

Similar as the general filters, also symmetric filters with possibly different shapes and centers of symmetry, can be combined to share delay lines or to obtain free pipelining. Several instances of the schemes from Figure 4.5, where the symmetry is exploited, can be combined in a similar way as the general filters in Figure 4.7 and Figure 4.8.

Example 4.7. To illustrate the alternatives for combining the $(\sigma_H, 2)$-symmetric filter $H(z) \in \mathbb{C}(z)$ and the $(\sigma_G, 2)$-symmetric filter $G(z) \in \mathbb{C}(z)$, with a common input or output, these filters are chosen as:

$$H(z) = \langle a, b, \sqrt{\sigma_H} c, \sigma_H b^*, \sigma_H a^* \rangle,$$
$$G(z) = \langle p, q, \sqrt{\sigma_G} r, \sigma_G q^*, \sigma_G p^* \rangle.$$

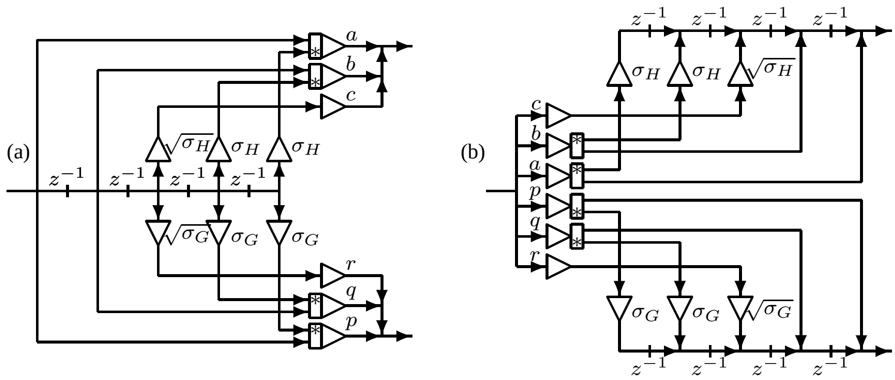


Figure 4.9: *Single input, dual output symmetric transversal structures.*

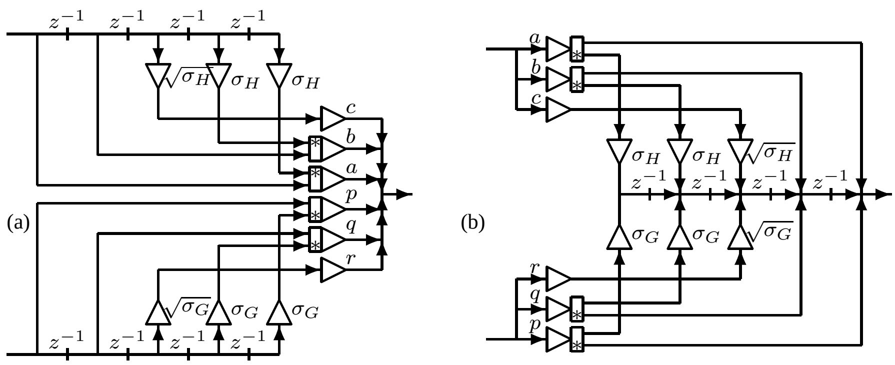


Figure 4.10: *Dual input, single output symmetric transversal structures.*

In Figure 4.9(a) and Figure 4.10(b) the delay lines are combined, and if $\sigma_H = \sigma_G$ also the multiplications with the shapes of symmetry may be combined. Note that in Figure 4.9(b) and Figure 4.10(a), except the input and output, nothing can be combined.
End of example

4.4.3 Complex mutually (σ, μ)-mirrored filters

Another special situation occurs when two mutually (σ, μ)-mirrored filters have a common input or common output. This may result from the polyphase decomposition of a (σ, μ)-symmetric filter, see Chapter 5, e.g., Example 5.3. Depending on the type of delay line and the inputs or outputs being common, the pairs of multiplications with conjugated coefficients of the filters $H(z), G(z) \in \mathbb{C}(z)$ may be combined. In the scheme with common input and a tapped delay line, and in the scheme with common output and an adding delay line, it is not possible to combine multiplications of both filters. These pairs of multiplications with conjugated coefficients have no common input nor a common output.

Example 4.8. For illustrating the possibilities for sharing multiplications of two mutually mirrored filters, two filters are chosen as:

$$\begin{aligned} H(z) &= \langle a, b, c, d, e \rangle &&= \mathcal{M}_{\sigma,2}\big(G(z)\big), \\ G(z) &= \langle \sigma e^*, \sigma d^*, \sigma c^*, \sigma b^*, \sigma a^* \rangle &&= \mathcal{M}_{\sigma,2}\big(H(z)\big). \end{aligned}$$

In Figure 4.11 and Figure 4.12 the possible structures are shown.

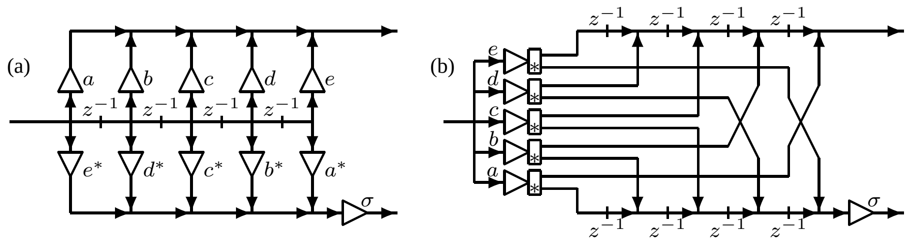


Figure 4.11: *Single input, dual output mutually mirrored transversal structures.*

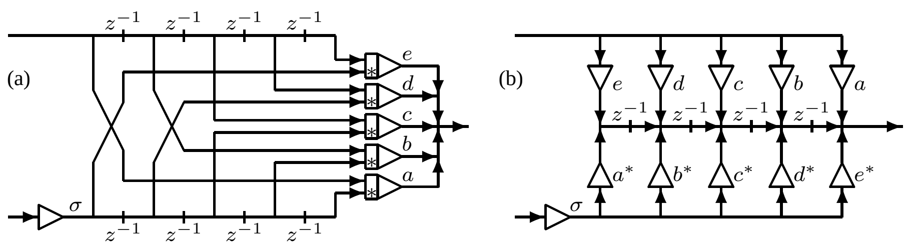


Figure 4.12: *Dual input, single output mutually mirrored transversal structures.*

End of example

In Lemma 2.18 it is shown how two mutually mirrored filters with a common input or output can be converted into two symmetric filters and a combination network. These symmetric filters may be combined according to one of the schemes in Figure 4.9 or Figure 4.10.

4.5 Complex filter and coefficient structures

In the preceding sections, several types of complex components (delays, adder, subtractors and multipliers) were used. In this section the complex filter will be decomposed into its individual real and imaginary parts first, so requiring real components only. In principle it is also possible to represent a complex quantity in its polar form. This polar form, however, requires costly transformations in case of addition, making this form unattractive for use in transversal filters. Besides the basic structure, also some interesting alternatives are discussed.
Since complex scale factors are special instances of complex filters with $L = 1$, the complex filter structures as discussed in this section, can also be used to construct complex scale factors in Section 4.7 and Appendix E.

4.5.1 Basic structure

The multiplication of two complex quantities can be expressed in its individual real and imaginary parts. So for any input signal $X(z) \in \mathbb{C}(z)$, and filter $H(z) \in \mathbb{C}(z)$, the output signal $Y(z) \in \mathbb{C}(z)$ is:

$$\begin{aligned} Y(z) &= X(z)H(z), \\ Y_\mathrm{r}(z) + \mathrm{j}Y_\mathrm{i}(z) &= \big(X_\mathrm{r}(z) + \mathrm{j}X_\mathrm{i}(z)\big)\big(H_\mathrm{r}(z) + \mathrm{j}H_\mathrm{i}(z)\big) \\ &= \big(X_\mathrm{r}(z)H_\mathrm{r}(z) - X_\mathrm{i}(z)H_\mathrm{i}(z)\big) + \mathrm{j}\big(X_\mathrm{r}(z)H_\mathrm{i}(z) + X_\mathrm{i}(z)H_\mathrm{r}(z)\big). \end{aligned}$$

The related matrix equation is:

$$\mathbf{Y}(z) = \mathbf{H}(z)\mathbf{X}(z),$$

with:

$$\mathbf{Y}(z) = \begin{bmatrix} Y_\mathrm{r}(z) \\ Y_\mathrm{i}(z) \end{bmatrix}, \quad \mathbf{H}(z) = \begin{bmatrix} H_\mathrm{r}(z) & -H_\mathrm{i}(z) \\ H_\mathrm{i}(z) & H_\mathrm{r}(z) \end{bmatrix} \text{ and } \mathbf{X}(z) = \begin{bmatrix} X_\mathrm{r}(z) \\ X_\mathrm{i}(z) \end{bmatrix}. \tag{4.1}$$

This matrix equation shows how a complex filter can be constructed from its individual real and complex parts. The resulting basic structure, Structure A, is presented in Figure 4.13. Note that the transposed matrix, $\mathbf{H}^T(z)$, is the matrix for $H^{\overline{*}}(z)$.
If the input signal is real or imaginary, Structure A from Figure 4.13, boils down to the structures from Figure 4.14(a) and Figure 4.14(b) respectively. Similarly, Figure 4.15(a) and Figure 4.15(b) show the structures in case only the real or only the imaginary part of the output signal are used. In the sequel only the full structure, like Figure 4.13, will be used. The schemes for special input or output signals can then be obtained easily. Also the case where the filter is real or imaginary, the corresponding structure can be derived easily.
In general, for a length L filter $H(z) \in \mathbb{C}(z)$, and input signal $X(z) \in \mathbb{C}(z)$, Structure A requires $4L$ real multiplications and $4L-2$ real additions per input sample. In the remaining part of this section some alternative structures are considered that in general require less multiplications per input sample.

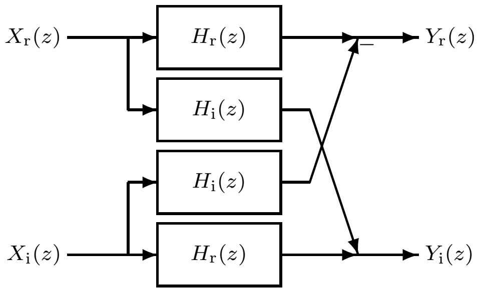


Figure 4.13: *Basic structure, Structure* A *(Equation 4.1).*

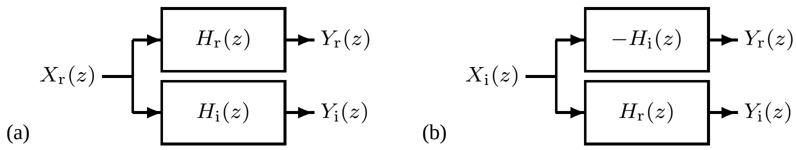


Figure 4.14: *Structure* A *for real (a) or imaginary (b) input signals.*

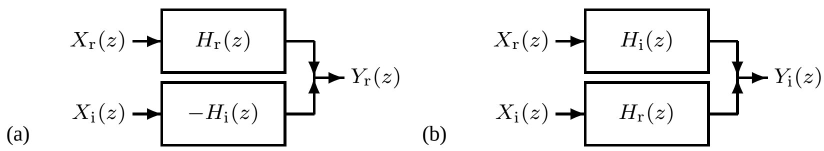


Figure 4.15: *Structure* A *for real (a) or imaginary (b) output signals.*

4.5.2 Alternative structures

The reason to look for alternative structures is the possible reduction of arithmetic costs. In general it may be assumed that the costs of a multiplication are significantly higher than the costs of an addition. In case multiplications are implemented by shift-and-add operations, different coefficients may have different costs as will be discussed extensively in Appendix E. Therefore it may be beneficial to have alternative sets of coefficients to choose from.

The matrix $\mathbf{H}(z)$, describing the complex filter operation in its individual real and imaginary parts, Equation 4.1, can be factorized in many ways. In this section a number of factorizations are collected that are found in literature. Each factorization gives a diagonal matrix containing three subfilters, and two combination matrices. The four factorizations differ in the sense that each alternative contains a different set of subfilters. From these alternatives many trivial variants can be derived by moving the minus-sign around (not shown). The four factorizations are:

$$\mathbf{H}(z) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} H_\mathrm{r}(z)+H_\mathrm{i}(z) & 0 & 0 \\ 0 & H_\mathrm{i}(z) & 0 \\ 0 & 0 & H_\mathrm{r}(z)-H_\mathrm{i}(z) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{4.2}$$

$$= \begin{bmatrix} 0 & 1 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} H_\mathrm{r}(z)-H_\mathrm{i}(z) & 0 & 0 \\ 0 & H_\mathrm{r}(z) & 0 \\ 0 & 0 & H_\mathrm{r}(z)+H_\mathrm{i}(z) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{4.3}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} H_\mathrm{r}(z) & 0 & 0 \\ 0 & H_\mathrm{r}(z)-H_\mathrm{i}(z) & 0 \\ 0 & 0 & H_\mathrm{i}(z) \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \tag{4.4}$$

$$= \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} H_\mathrm{r}(z) & 0 & 0 \\ 0 & H_\mathrm{r}(z)+H_\mathrm{i}(z) & 0 \\ 0 & 0 & -H_\mathrm{i}(z) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{4.5}$$

The first factorization, Equation 4.2, relates to Structure B in Figure 4.16 and is also found in, e.g., [110] [135] for scale factors instead of filters. The factorizations in Equation 4.3 ([5] Equation 9a), Equation 4.4 and Equation 4.5 ([5] Equation 9b and [114] Equation 5.4.2 for scale factors), relate to Structure C, Structure D and Structure E in Figure 4.17, Figure 4.18 and Figure 4.19 respectively. With respect to Equation 4.4 no reference is found.

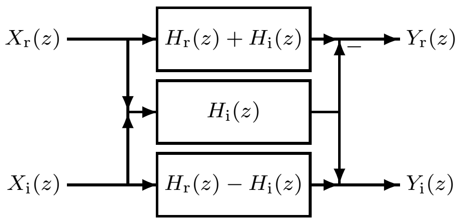


Figure 4.16: *Alternative structure, Structure* B *(Equation 4.2).*

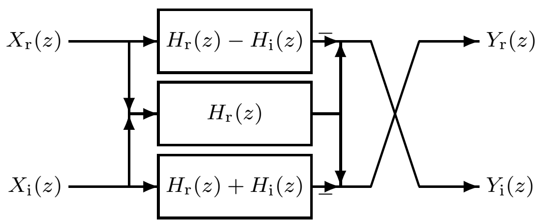


Figure 4.17: *Alternative structure, Structure* C *(Equation 4.3).*

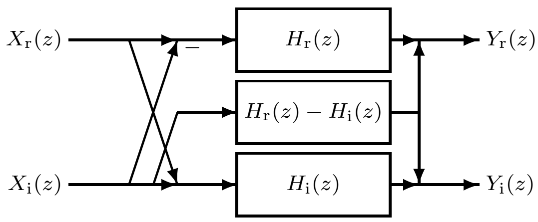


Figure 4.18: *Alternative structure, Structure* D *(Equation 4.4).*

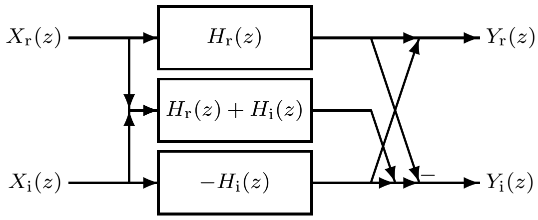


Figure 4.19: *Alternative structure, Structure* E *(Equation 4.5).*

In general, for a length L filter $H(z) \in \mathbb{C}(z)$, and input signal $X(z) \in \mathbb{C}(z)$, Structure B through Structure E, require $3L$ real multiplications per input sample. This is less than for Structure A in Figure 4.13, where $4L$ real multiplications per input sample are required. Similarly, Structure B and Structure C require $3L$ real additions per input sample, whereas Structure D and Structure E require $3L + 1$ real additions per input sample. For $L > 2$ and $L > 3$ respectively, this too is less than the $4L - 2$ real additions per input sample in the basic structure, Structure A.

Which of the structures are best chosen depends on many aspects. For instance the difference between Structure B and Structure C may be significant in case of differences in lengths, L_r of filter $H_r(z)$ and L_i of filter $H_i(z)$. It may be beneficial to select the shortest filter for the central branch, i.e., Structure B if $L_r > L_i$ and Structure C if $L_r < L_i$. Also, in case of finite precision coefficients, where multiplications can be implemented as shift-and-add operations, one scheme may be more efficient to implement than the other.

4.6 Complex symmetric filter structures

In this section the basic structure, Structure A, and the alternative structures, Structure B through Structure E, are considered in case the complex filter has symmetry type 5, 6, 7 or 8. The subfilters in all cases are real filters and satisfy the properties as described in Lemma 2.27.

For ease of comparison, it is assumed that a length L real symmetric filter requires only $\frac{L}{2}$ real multiplications per input sample, using the schemes as shown in Figure 4.5 and

Figure 4.6. The facts that L may be odd, or that a central coefficient may be zero, are ignored.

4.6.1 Basic structure for type 5 and 6 filters

By Lemma 2.27, the individual real and imaginary parts of a type 5 and 6 filter are symmetric filters. Therefore, each subfilter of the length L filter $H(z) \in \mathbb{C}(z)$ and input signal $X(z) \in \mathbb{C}(z)$, requires $\frac{L}{2}$ real multiplications per input sample. The total filter now requires $2L$ real multiplications per input sample. The number of additions has not changed and is $4L - 2$.
Although complex filters have no structural transmission zeros, see Lemma 2.42, the individual real and imaginary parts may have these zeros, Lemma 2.40 and Lemma 2.41. Depending on the filter length and the type, structural transmission zeros can be identified and used to split the subfilter. In Figure 4.20 the resulting structure is shown for an odd-length type 5 filter with $H_{\mathrm{i}}(z) = H'_{\mathrm{i}}(z)(1 - z^{-2})$. In Figure 4.21 the resulting structure is shown for an even-length type 5 filter with $H_{\mathrm{r}}(z) = H'_{\mathrm{r}}(z)(1 + z^{-1})$ and $H_{\mathrm{i}}(z) = H'_{\mathrm{i}}(z)(1 - z^{-1})$. Note that all subfilters in Figure 4.20 and Figure 4.21 are type 1 filters. A similar approach is possible for type 6 filters (not shown).

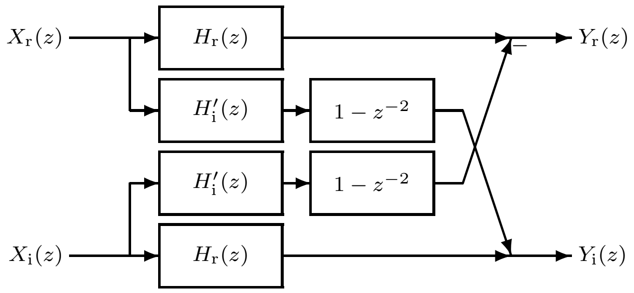


Figure 4.20: *Structural transmission zeros in an odd-length type 5 filter.*

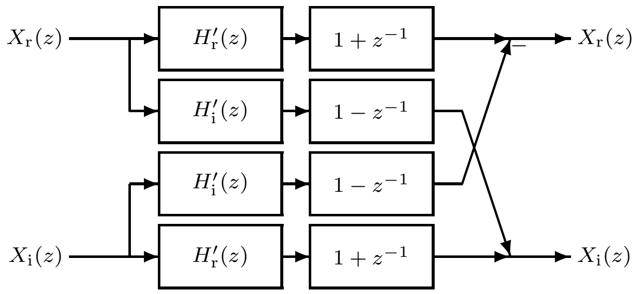


Figure 4.21: *Structural transmission zeros in an even-length type 5 filter.*

4.6.2 Basic structure for type 7 and 8 filters

By Lemma 2.27 the individual real and imaginary parts of a type 7 and 8 filter are mutually mirrored. The subfilters in general cannot assumed to be symmetric. As a consequence, each subfilter of the length L filter $H(z) \in \mathbb{C}(z)$ and input signal $X(z) \in \mathbb{C}(z)$, requires L real multiplications per input sample. The total filter now requires $4L$ real multiplications per input sample.
Similarly as shown in Figure 4.11(b) and Figure 4.12(a), it is possible to combine the four subfilters in pairs, so reducing the required number of real multiplications per input sample to $2L$. Also, from Figure 4.13 it is clear that the mutually mirrored subfilters have common inputs or common outputs. By Lemma 2.18 all four mutually mirrored subfilters can be converted into four symmetric subfilters. Also now, the total filter requires $2L$ real multiplications per input sample. The number of real additions has increased to $4L + 2$.
For the type 7 filter $H(z)$, the subfilters $A(z)$ and $B(z)$, in Figure 4.22, are $(1, \mu)$ and $(-1, \mu)$-symmetric respectively. Both filters are:

$$\begin{aligned} A(z) &= \tfrac{1}{2}\big(H_{\mathrm{r}}(z) + H_{\mathrm{i}}(z)\big) = \tfrac{1}{2}\big(H_{\mathrm{r}}(z) + \mathcal{M}_{1,\mu}\big(H_{\mathrm{r}}(z)\big)\big), \\ B(z) &= \tfrac{1}{2}\big(H_{\mathrm{r}}(z) - H_{\mathrm{i}}(z)\big) = \tfrac{1}{2}\big(H_{\mathrm{r}}(z) - \mathcal{M}_{1,\mu}\big(H_{\mathrm{r}}(z)\big)\big). \end{aligned}$$

For the type 8 filter $H(z)$, the subfilters $A(z)$ and $B(z)$, in Figure 4.22, are $(-1, \mu)$ and $(1, \mu)$-symmetric respectively. Both filters are:

$$\begin{aligned} A(z) &= \tfrac{1}{2}\big(H_{\mathrm{r}}(z) + H_{\mathrm{i}}(z)\big) = \tfrac{1}{2}\big(H_{\mathrm{r}}(z) + \mathcal{M}_{-1,\mu}\big(H_{\mathrm{r}}(z)\big)\big), \\ B(z) &= \tfrac{1}{2}\big(H_{\mathrm{r}}(z) - H_{\mathrm{i}}(z)\big) = \tfrac{1}{2}\big(H_{\mathrm{r}}(z) - \mathcal{M}_{-1,\mu}\big(H_{\mathrm{r}}(z)\big)\big). \end{aligned}$$

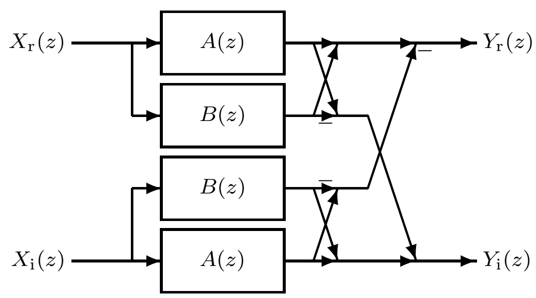


Figure 4.22: *Alternative structure, Structure* F*, restoring the symmetry of a type* 7 *or* 8 *filter.*

To obtain the structure of Figure 4.22, Structure F, Lemma 2.18 was applied to the filter pairs with a common input. The same is possible for the filter pairs with a common output. The preference for the one or other may depend on the application. In a similar way as for the type 5 and 6 filters, it is possible to exploit structural transmission zeros in the scheme of Figure 4.22 (not shown).

4.6.3 Alternative structures for type 5, 6, 7 and 8 filters

The symmetry of the subfilters $H_{\mathrm{r}}(z)$ and $H_{\mathrm{i}}(z)$, in case of type 5 and 6 filter $H(z)$, can also be found in one or two branches of the alternative structures, Structure B through Structure E. By Lemma 2.19, the real filters in the upper and lower branches of Structure B and Structure C are mutually mirrored. The total filter now requires $\frac{5}{2}L$ real multiplications per input sample for Structure B and Structure C, and $2L$ real multiplications per input sample for Structure D and Structure E.
In case of type 7 and 8 filters $H(z)$, the $H_{\mathrm{r}}(z)$ and $H_{\mathrm{i}}(z)$ are mutually mirrored. By Lemma 2.17, the real filters in the upper and lower branches of Structure B and Structure C and in the central branches of Structure D and Structure E, are symmetric. The total filter now requires $2L$ real multiplications per input sample for Structure B and Structure C, and $\frac{5}{2}L$ real multiplications per input sample for Structure D and Structure E.
Similarly as for the filter structures in Section 4.5 it may be possible to apply Lemma 2.40 and Lemma 2.41 about structural transmission zeros, for the symmetric subfilters.

4.7 Conjugated pairs

As found in Section 4.2, coefficients often appear in conjugated pairs. In Figure 4.4 the new symbols are introduced already. In this section it is shown how such pairs of coefficients can be combined efficiently. A pair of conjugated coefficients, (a, a^*), with a common output is described by:

$$Y(z) = aX_0(z) + a^* X_1(z),$$

and a pair of conjugated coefficients, (a, a^*), with a common input is described by:

$$Y_0(z) = aX(z) \text{ and } Y_1(z) = a^* X(z).$$

In matrix notation this respectively gives:

$$\mathbf{Y}(z) = \begin{bmatrix} \mathbf{A} & \mathbf{A}^T \end{bmatrix} \begin{bmatrix} \mathbf{X}_0(z) \\ \mathbf{X}_1(z) \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{Y}_0(z) \\ \mathbf{Y}_1(z) \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^T \end{bmatrix} \mathbf{X}(z),$$

with:

$$\mathbf{A} = \begin{bmatrix} a_{\mathrm{r}} & -a_{\mathrm{i}} \\ a_{\mathrm{i}} & a_{\mathrm{r}} \end{bmatrix}.$$

Note that the transposed matrix, $\mathbf{A}^T$, is the matrix for a^*.
Similarly as for the alternative complex filter structures in Section 4.5, the approach is to factorize the matrices $\begin{bmatrix} \mathbf{A} & \mathbf{A}^T \end{bmatrix}$ and $\begin{bmatrix} \mathbf{A} \\ \mathbf{A}^T \end{bmatrix}$. Although many factorizations are possible here too, only the results as found in Section 4.5 and Section 4.6 are used. It appears that the factorization based on Structure A, and the factorization based on Structure B and Structure C, give different sets of scale factors, whereas the factorization based on Structure D and Structure E, gives the same set of scale factors as the factorization based on Structure B and Structure C, at the cost of more additions. On basis of Structure F, a

factorization with yet another set of scale factors is found, however, at the cost of many more additions. Perhaps a more efficient factorization is possible.
The four schemes for the conjugated coefficients with a common output are shown in Figure 4.23 through Figure 4.26. The four schemes for the conjugated coefficients with a common input are shown in Figure 4.27 through Figure 4.30.

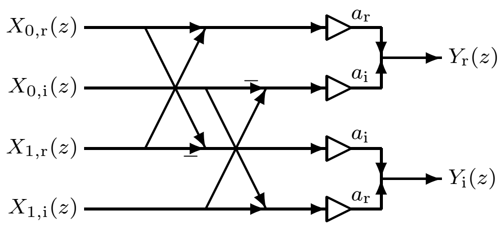


Figure 4.23: *Conjugated pair with common output, based on Structure* A.

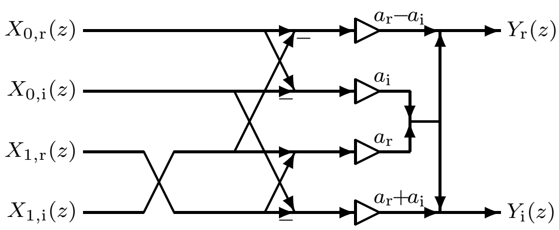


Figure 4.24: *Conjugated pair with common output, based on Structure* B *and Structure* C.

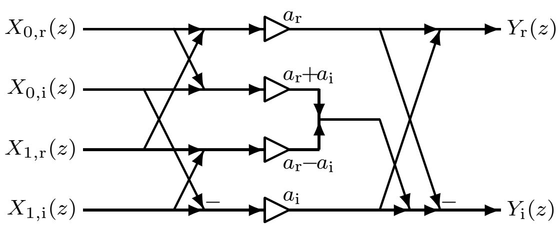


Figure 4.25: *Conjugated pair with common output, based on Structure* D *and Structure* E.

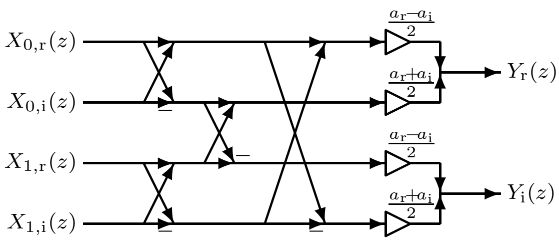


Figure 4.26: *Conjugated pair with common output, based on Structure* F.

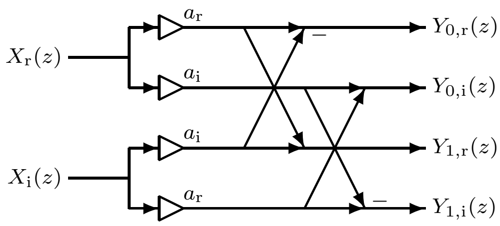


Figure 4.27: *Conjugated pair with common input, based on Structure* A.

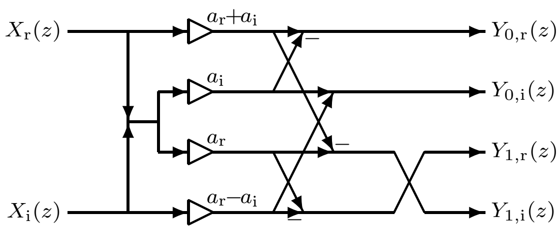


Figure 4.28: *Conjugated pair with common input, based on Structure* B *and Structure* C.

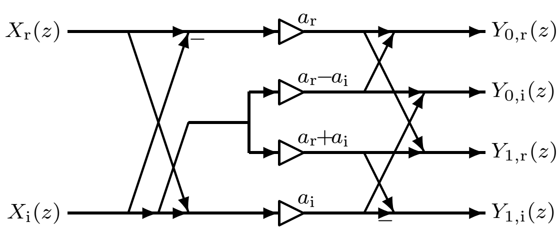


Figure 4.29: *Conjugated pair with common input, based on Structure* D *and Structure* E.

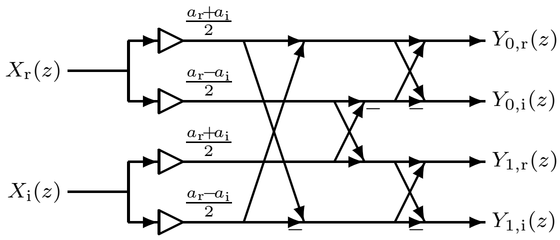


Figure 4.30: *Conjugated pair with common input, based on Structure* F.

4.8 Résumé

In the previous sections of this chapter, a series of alternatives for constructing possibly complex transversal filters are presented. Many of these alternatives have to be combined with others to obtain a final structure.

Section 4.1, Section 4.2 and Section 4.3 describe transversal structures with possibly complex components, and available symmetry is exploited for reducing arithmetic complexity. In Section 4.4 multiple transversal filter structures with common inputs or outputs are efficiently combined, also exploiting symmetry when possible.

Section 4.5 and Section 4.6 describe how a complex filter can be decomposed into real subfilters. These subfilters can be constructed from one of the transversal structures with real components. In total six alternative structures, Structure A through Structure F, are discussed. These alternative structures are also applicable to complex coefficients such that complex coefficients can be decomposed into real subcoefficients. In Section 4.7 two conjugated coefficients with a common input or output are efficiently combined again using the alternative structures found earlier.

Now the arithmetic complexity of the transversal filter structures in terms of the number of real multiplications per input sample $\#mul$, and the number of real additions per input sample $\#add$, for length L filters are collected. The facts that L may be odd, or that a central coefficient may be zero, are ignored.

First, the alternatives Structure A through Structure F are used to decompose the complex filter into real subfilters with real components. The arithmetic costs are listed for: ”any type” (the non-symmetric and the type 9 symmetric filters), type 5 or 6, and type 7 or 8, symmetric complex filters. The results are listed in the upper part of Table 4.1. Secondly, the alternatives Structure A through Structure F are used to decompose the complex coefficients into real subcoefficients. The results are listed in the lower part of Table 4.1.

In each case a possible composition of the filter is indicated with a note (a through t). For each note in Table 4.1, an example scheme is given in Table 4.2. Both, a global structure and a local structure are indicated. The local structure is to be used within the global structure.

filter structure	any type $\#mul$	any type $\#add$	any type note	type 5 or 6 $\#mul$	type 5 or 6 $\#add$	type 5 or 6 note	type 7 or 8 $\#mul$	type 7 or 8 $\#add$	type 7 or 8 note
A	$4L$	$4L-2$	a	$2L$	$4L-2$	e	$2L$	$4L-2$	i
B,C	$3L$	$3L$	b	$\frac{5}{2}L$	$3L$	f	$2L$	$3L$	j
D,E	$3L$	$3L+1$	c	$2L$	$3L+1$	g	$\frac{5}{2}L$	$3L+1$	k
F	$4L$	$4L+2$	d	$2L$	$4L+2$	h	$2L$	$4L+2$	l

coefficient structure	any type $\#mul$	any type $\#add$	any type note	type 5 or 6 $\#mul$	type 5 or 6 $\#add$	type 5 or 6 note	type 7 or 8 $\#mul$	type 7 or 8 $\#add$	type 7 or 8 note
A	$4L$	$4L-2$	m	$2L$	$4L-2$	q	$2L$	$4L-2$	q
B,C	$3L$	$5L-2$	n	$2L$	$\frac{9}{2}L-2$	r	$2L$	$\frac{9}{2}L-2$	r
D,E	$3L$	$6L-2$	o	$2L$	$5L-2$	s	$2L$	$5L-2$	s
F	$4L$	$8L-2$	p	$2L$	$6L-2$	t	$2L$	$6L-2$	t

Table 4.1: *Number of real multiplications and additions for filters of length L.*

note	example schemes: global	example schemes: local
a	Fig. 4.13	$2\times$ Fig. 4.7 or $2\times$ Fig. 4.8
b	Fig. 4.16 or 4.17	$3\times$ Fig. 4.1
c	Fig. 4.18 or 4.19	$3\times$ Fig. 4.1
d	Fig. 4.22	$2\times$ Fig. 4.7 or $2\times$ Fig. 4.8
e	Fig. 4.13	$2\times$ Fig. 4.9 or $2\times$ Fig. 4.10
f	Fig. 4.16 or 4.17	$2\times$ Fig. 4.1 and $1\times$ Fig. 4.5
g	Fig. 4.18 or 4.19	$1\times$ Fig. 4.1 and $2\times$ Fig. 4.5
h	Fig. 4.22	$2\times$ Fig. 4.11(b) or $2\times$ Fig. 4.12(a)
i	Fig. 4.13	$2\times$ Fig. 4.11(b) or $2\times$ Fig. 4.12(a)
j	Fig. 4.16 or 4.17	$1\times$ Fig. 4.1 and $2\times$ Fig. 4.5
k	Fig. 4.18 or 4.19	$2\times$ Fig. 4.1 and $1\times$ Fig. 4.5
l	Fig. 4.22	$2\times$ Fig. 4.9 or $2\times$ Fig. 4.10
m	Fig. 4.1	$L\times$ Fig. 4.13
n	Fig. 4.1	$L\times$ Fig. 4.16 or $L\times$ 4.17
o	Fig. 4.1	$L\times$ Fig. 4.18 or $L\times$ 4.19
p	Fig. 4.1	$L\times$ Fig. 4.22
q	Fig. 4.5	$\frac{L}{2}\times$ Fig. 4.23 or $\frac{L}{2}\times$ Fig. 4.27
r	Fig. 4.5	$\frac{L}{2}\times$ Fig. 4.24 or $\frac{L}{2}\times$ Fig. 4.28
s	Fig. 4.5	$\frac{L}{2}\times$ Fig. 4.25 or $\frac{L}{2}\times$ Fig. 4.29
t	Fig. 4.5	$\frac{L}{2}\times$ Fig. 4.26 or $\frac{L}{2}\times$ Fig. 4.30

Table 4.2: *Example schemes related to Table 4.1.*

Some observations that can be made from the results in Table 4.1 are:

- For ”any type” of filter, the least number of real multiplications is $3L$, and for type 5, 6, 7 and 8 the least number of real multiplications is $2L$. These numbers are valid for both the alternative filter and the alternative coefficient structures.
- In case of the global structures Structure B or Structure C, the minimum number of real additions is found to be $3L$.
- In case of the alternative coefficient structures, the number of additions exceeds the number of additions in case of the alternative filter structures, except for Structure A where the numbers are equal.
- In case the real multiplications are implemented using shift-and-add operations, see Appendix E, some alternative structures may be beneficial, since their coefficients may require less real additions. The alternative filter structures Structure B through Structure E are good alternatives.
- It is to be expected that the decomposition of a complex filter into real subfilters with real operations, will result in lower costs than the decomposition of complex operations into real operations.

4.9 Conclusion

The transversal filter structure is discussed in detail for (σ, μ)-symmetric real and complex filters. By incorporating the shape of symmetry in the delay line, or by using it as a separate factor, any (σ, μ)-symmetric filter can be implemented with a transversal structure, where all coefficients appear in conjugated pairs. In case of a real symmetric filter, it is trivial to combine coefficients to reduce the computational costs. For complex symmetric filters it is shown how pairs of conjugated coefficients can be combined in a new structure, to reduce computational costs. It is also shown how transversal filters with common inputs or common outputs can be combined efficiently. The choice of a transversal filter with a tapped delay line or an adding delay line however, appears to be key. Opposite to using complex components for implementing complex filters, a complex filter can be decomposed in its individual real and imaginary parts. A series of alternatives is discussed that are also of great value in the efficient construction of complex coefficients in Appendix E.
From the analysis of many alternative structures, it can be concluded that in general the decomposition of a complex filter into real subfilters with real operations, will result in lower costs than the decomposition of the individual complex operations into real operations.

Chapter 5

Polyphase structures

One of the most important concepts to save in computational costs in multirate filtering is the polyphase decomposition, and the closely related polyphase structure, as introduced by Bellanger in 1976 [4] and elaborated in, e.g., [43] [132] [133] [135]. This concept allows for efficient implementations of multirate filters both in hardware and software. Linear-phase filters exhibit symmetry that may be exploited fruitfully to reduce computational costs by a factor of 2 see, e.g., [116] and Chapter 4 of this thesis. Application of the polyphase decomposition to symmetric filters however, often destroys the available symmetry and hence the reduction of costs by a factor of 2 is no longer possible. Therefore it is desired to have, in case the original filter is symmetric, an alternative polyphase structure that is composed of symmetric filters only.

A first approach to restore the symmetry in a polyphase structure is published in 1996 [95] and is based on techniques that were introduced in [94]. The method from [95] is suited for real multirate filters with integer factors. A claim without any proof is made that the algorithm can be applied to multirate filters with rational factors too.

Central in this chapter is the restoration of symmetry in polyphase structures. A new theorem states that the polyphase structure of any real or complex multirate (σ, μ)-symmetric filter, with integer or rational multirate factors, can be constructed using symmetric filters only. A unified approach results in a general applicable algorithm to devise the polyphase structure that contains symmetric filters only. In addition, an extra degree of freedom in restoring the symmetry is identified that may lead to additional savings in computational costs. The main results of this chapter are filed for a patent in 2002 and the applications are published in 2004 [17] and 2006 [18].

In the time frame between the patent-application publications and the publication of this thesis, an alternative method to restore symmetry for real filters with integer multirate factors has been published [141] [142]. The method is extended in [13] to deal with rational multirate factors too. Next to restoring symmetry also the possible savings in memory are considered.

Browsing the literature for the term ”polyphase”, results in many hits in the field of continuous time filtering. There it is related to what would be called complex filters in this thesis. A brief introduction into the field of these continuous time complex or polyphase filters is given in Appendix C.

The outline of this chapter is as follows. After formally describing the polyphase decomposition and some other valuable definitions and identities in Section 5.1, the known basic polyphase structures for integer multirate factors are treated in Section 5.2, e.g., Figure 5.2 and Figure 5.3. In case of a rational multirate factor the polyphase structure can be obtained by applying the polyphase decomposition twice. The resulting nested polyphase structures, e.g., Figure 5.4(b) or Figure 5.4(c) are treated in Section 5.3. The nested polyphase structure depends on the order in which both polyphase decompositions are applied. It is known how a nested polyphase structure can be transformed to a structure that is independent of the order in which the polyphase decompositions are applied, e.g., Figure 5.5(b) or Figure 5.5(c). Section 5.4 discusses in detail this unified polyphase structure that will serve as a basis for the restoration of the symmetry. Section 5.5 describes the relations between the various polyphase components (PPCs) of a symmetric filter, that are key in the restoration of symmetry. The new theorem on restoring symmetry, its proof and a procedure to restore symmetry, are discussed in Section 5.6. Section 5.7 presents some typical examples to show the application of the procedure. Finally, in Section 5.8, some remarks are made about structural zeros in PPCs. The conclusions are found in Section 5.9.

5.1 Polyphase decomposition and identity

Related to the polyphase structure that will be discussed in detail in this chapter, is the polyphase decomposition. The notation, some definitions and identities, are presented in this section. The polyphase decomposition decomposes a filter $H(z)$ into a number of polyphase components (PPCs). The polyphase component is defined first.

Definition 5.1 (Polyphase component)**.** *Polyphase component $R : r$, with* decomposition factor $R \in \mathbb{N}^+$ *and* decomposition index $r \in \mathbb{Z}$, *of filter $H(z) \in \mathbb{C}(z)$, is defined as:*

$$H_{R:r}(z) \triangleq \sum_n h[Rn + r] z^{-n}.$$

A trivial instance of this definition is $H(z) = H_{1:0}(z)$.

The well-known polyphase decomposition, decomposes any filter into R PPCs, for a given decomposition factor R, as follows.

Lemma 5.1 (Polyphase decomposition)**.** *Any filter $H(z) \in \mathbb{C}(z)$ can be decomposed in R PPCs as:*

$$H(z) = \sum_{r=0}^{R-1} z^{-r} H_{R:r}(z^R).$$

Proof. In the definition of the z-transform $H(z)$ of an impulse response $h[n]$, the variable

n is substituted, $n \leftarrow Rn + r$, resulting in:

$$\begin{aligned} H(z) &= \sum_n h[n] z^{-n} \\ &= \sum_n \sum_{r=0}^{R-1} h[Rn+r] z^{-(Rn+r)} \\ &= \sum_{r=0}^{R-1} z^{-r} \sum_n h[Rn+r] z^{-Rn} \\ &= \sum_{r=0}^{R-1} z^{-r} H_{R:r}(z^R). \end{aligned} \qquad \square$$

Basically, a PPC is a filter from which again a PPC can be selected. In that case the equivalent decomposition factor is the product of the individual decomposition factors, and the equivalent decomposition index depends on the first decomposition factor and both indices. To generalize further additional delays can be included, see Lemma 5.2. From this it is also clear that a PPC with a non-prime decomposition factor can be considered as a repeated selection of the polyphase components with decomposition factors being the factors of the original decomposition factor.

Lemma 5.2 (Polyphase identity)**.** *For any filter $H(z) \in \mathbb{C}(z)$ and delay z^a with $a \in \mathbb{Z}$ holds:*

$$\left(z^a H_{R:r}(z)\right)_{P:p}(z) = H_{PR:r+R(a+p)}(z).$$

Proof. Repeated application of Definition 5.1 gives:

$$\begin{aligned} \left(z^a H_{R:r}(z)\right)_{P:p}(z) &= \left(z^a \sum_i h[Ri+r] z^{-i}\right)_{P:p}(z) \\ &= \left(\sum_j h[Rj+r+Ra] z^{-j}\right)_{P:p}(z) \\ &= \sum_j h[R(Pj+p)+r+Ra] z^{-j} \\ &= H_{PR:r+R(a+p)}(z). \end{aligned} \qquad \square$$

From Lemma 5.2 it follows directly that $H_{R:r}(z)$ and $H_{R:r+aR}(z)$ basically refer to the same PPC, except for a delay, since $r|_R = (r+aR)|_R$. This implies that the polyphase decomposition is not unique. To deal with this freedom in a formal way, the index set $\mathcal{R}$ and the fundamental index set $\mathcal{R}_0$ are defined. The fundamental index set $\mathcal{R}_0$ contains the integers ranging from 0 through $R-1$. Index set $\mathcal{R}$ contains exactly R elements such that $\mathcal{R}|_R = \mathcal{R}_0$. Note that, opposite to the fundamental index set, the index set $\mathcal{R}$ is not unique.

Definition 5.2. *For any $R \in \mathbb{N}^+$ the fundamental index set $\mathcal{R}_0$ and the index set $\mathcal{R}$ are:*

$$\mathcal{R}_0 \triangleq \{r | r \in \mathbb{N}, r < R\},$$
$$\mathcal{R} \triangleq \{r + a_r R | r \in \mathcal{R}_0, a_r \in \mathbb{Z}\}.$$

Similarly, the fundamental index sets $\mathcal{D}_0$ and $\mathcal{I}_0$, and the index sets $\mathcal{D}$ and $\mathcal{I}$ are defined. Now the polyphase decomposition can be formulated more generally.

Lemma 5.3 (Generalized polyphase decomposition)**.** *Any filter $H(z) \in \mathbb{C}(z)$ can be decomposed in R PPCs as:*

$$H(z) = \sum_{r \in \mathcal{R}} z^{-r} H_{R:r}(z^R).$$

Proof. By Lemma 5.1, Lemma 5.2, Definition 5.2 and taking $a_r \in \mathbb{Z}$:

$$\begin{aligned} H(z) &= \sum_{r \in \mathcal{R}_0} z^{-r} H_{R:r}(z^R) \\ &= \sum_{r \in \mathcal{R}_0} z^{-(r+a_r R)} H_{R:r+a_r R}(z^R) \\ &= \sum_{r \in \mathcal{R}} z^{-r} H_{R:r}(z^R). \end{aligned}$$

□

This generalized polyphase decomposition will be used mainly in proofs in the sequel of this chapter. In many schemes index set $\mathcal{R} = \mathcal{R}_0$ and hence Lemma 5.1 applies. An alternative description for a PPC is given in the next lemma, where $H_{R:r}(z)$ is interpreted as the z-transform of the R-fold decimated signal $h[n]$, started on moment $n = r$.

Lemma 5.4. *The $R : r$ PPC of filter $H(z) \in \mathbb{C}(z)$ equals:*

$$H_{R:r}(z) = \frac{1}{R} \sum_{i \in \mathcal{R}_0} H(z^{\frac{1}{R}} W_R^i) z^{\frac{r}{R}} W_R^{ir}.$$

Proof. From Definition 5.1 of the PPC, and Definition A.2 of the SRD, it follows that $H_{R:r}(z) = z^r H(z); \downarrow D$. Application of Lemma A.2 about downsampling concludes the proof. □

5.2 Basic polyphase structure

The concept of decimation with an integer decimation factor D is straightforward. After appropriate filtering to avoid aliasing, only 1 out of every D samples is left, the others are discarded by the sampling rate decreaser (SRD). For interpolation with an integer interpolation factor I, between every 2 successive input samples, $I - 1$ zeros are inserted by the sampling rate increaser (SRI), prior to appropriate filtering to suppress image frequency components.

In order to obtain computational efficient structures, the popular *polyphase structure* is used often, see [133]. The term *basic polyphase structure* is used in this thesis to distinguish from the *nested polyphase structure*, that is used in case of non-integer multirate factors, and will be discussed in the next section.
The basic polyphase structure is obtained, by first applying a polyphase decomposition with a decomposition factor equal to the decimation or interpolation factor, and second a noble identity, see Section A.3. Lemma 5.1 can be rewritten using Horner's scheme as:

$$H(z) = H_{R:0}(z^R) + z^{-1}\big(H_{R:1}(z^R) + z^{-1}\big(\cdots + z^{-1}H_{R:R-1}(z^R)\big)\big).$$

This relates to the two alternative structures as shown in Figure 5.1, that traditionally [133] are called the first and second polyphase structure respectively. Like with the transversal structures in Figure 4.1, the terms tapped and adding delay lines are applicable.

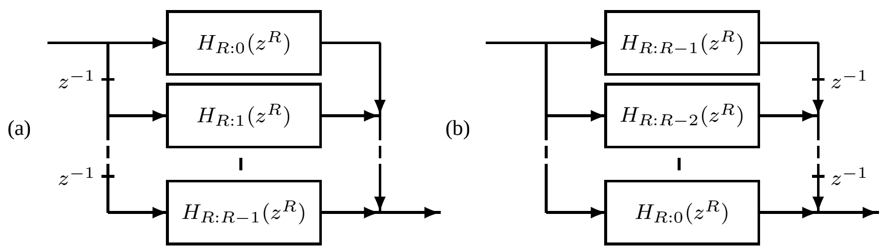


Figure 5.1: *First (a) and second (b) polyphase structure.*

In case of a decimating filter the first polyphase structure is used, since the SRD following the filter can be moved towards the left after applying the first noble identity, see Figure 5.2.

Lemma 5.5. *Any decimating filter $H(z) \in \mathbb{C}(z)$ with decimation factor $D \in \mathbb{N}^+$ and index set $\mathcal{D}$ can be rewritten as:*

$$H(z); \downarrow D = \sum_{d\in\mathcal{D}} \big(z^{-d}; \downarrow D; H_{D:d}(z)\big).$$

Proof. Application of the polyphase decomposition, Lemma 5.3, and the first noble identity (Lemma A.3) gives:

$$\begin{aligned} H(z); \downarrow D &= \sum_{d\in\mathcal{D}} \big(z^{-d}; H_{D:d}(z^D)\big); \downarrow D \\ &= \sum_{d\in\mathcal{D}} \big(z^{-d}; H_{D:d}(z^D); \downarrow D\big) \\ &= \sum_{d\in\mathcal{D}} \big(z^{-d}; \downarrow D; H_{D:d}(z)\big). \end{aligned}$$ □

In case $\mathcal{D} = \mathcal{D}_0$, the related Horner's scheme is:

$$H(z) = \big(\downarrow D; H_{D:0}(z)\big) \\ + \big(z^{-1}; \big(\big(\downarrow D; H_{D:1}(z)\big) + \big(z^{-1}; \big(\cdots + \big(z^{-1}; \downarrow D; H_{D:D-1}(z)\big)\big)\big)\big)\big).$$

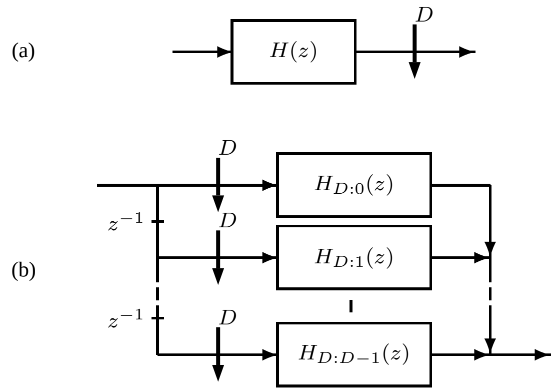


Figure 5.2: *First polyphase structure for an efficient decimating filter.*

Similarly, in case of an interpolating filter the second polyphase structure is selected, since the SRI preceding the filter can be moved towards the right after applying the second noble identity, see Figure 5.3.

Lemma 5.6. *Any interpolating filter $H(z) \in \mathbb{C}(z)$ with interpolation factor $I \in \mathbb{N}^+$ and index set $\mathcal{I}$ can be rewritten as:*

$$\uparrow I; H(z) = \sum_{i \in \mathcal{I}} \big(H_{I:i}(z); \uparrow I; z^{-i}\big).$$

Proof. Application of the polyphase decomposition, Lemma 5.3, and the second noble identity (Lemma A.4) gives:

$$\begin{aligned} \uparrow I; H(z) &= \uparrow I; \sum_{i \in \mathcal{I}} \big(H_{I:i}(z^I); z^{-i}\big) \\ &= \sum_{i \in \mathcal{I}} \big(\uparrow I; H_{I:i}(z^I); z^{-i}\big) \\ &= \sum_{i \in \mathcal{I}} \big(H_{I:i}(z); \uparrow I; z^{-i}\big). \end{aligned}$$ □

In case $\mathcal{I} = \mathcal{I}_0$, the related Horner's scheme is:

$$H(z) = \big(\big(\big(\big(\big(H_{I:I-1}(z); \uparrow I; z^{-1}\big) + \cdots\big); z^{-1}\big) + \big(H_{I:1}(z); \uparrow I\big)\big); z^{-1}\big) \\ + \big(H_{I:0}(z); \uparrow I\big).$$

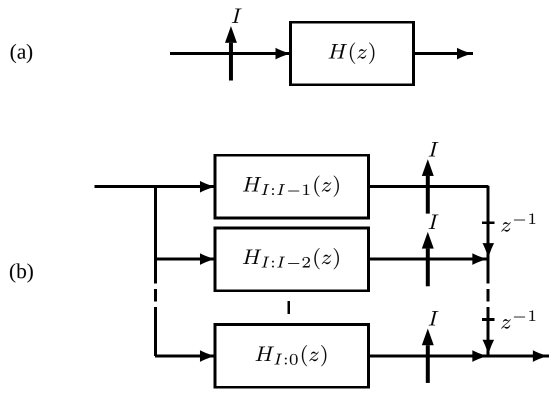


Figure 5.3: *Second polyphase structure for an efficient interpolating filter.*

The left part of Figure 5.2(b), the tapped delay line and the SRDs, will be referred to as the demultiplexer. The right part of Figure 5.3(b), the adding delay line and the SRIs, will be referred to as the multiplexer. In other work the functionality of these networks is often seen as a rotating switch or commutator, see for instance [43] and [135].
Since the SRDs are at the left most or the SRIs are at the right most position, the sampling rate inside the PPCs is minimal.

5.3 Nested polyphase structure

In Section 5.2 about the basic polyphase structure, the polyphase decomposition was applied to a multirate filter with an integer factor. For a rational factor, it is known how to apply the polyphase decomposition repeatedly [133] [135], and in this thesis the resulting structure is called the *nested polyphase structure*. In this section the multirate filter with the rational decimation factor $\frac{D}{I}$ (or interpolation factor $\frac{I}{D}$) is discussed in more detail. To obtain a nested structure both types of polyphase decompositions are applied, one after the other. Also the split-delay identity plays an essential role, see Lemma A.7.
Lemma 5.7 shows two alternative formulations for $\uparrow I; H(z); \downarrow D$. The first formulation results from, first applying the first polyphase decomposition, followed by the second polyphase decomposition. The second formulation results from applying both polyphase decompositions in the reversed order.
In many lemmas it is a requirement that the interpolation factor I and the decimation factor D have no common factor, or their greatest common divisor $\gcd(I, D) = 1$. This property is equivalent to the equality $pD + qI = 1$ with $p, q \in \mathbb{Z}$. See Appendix D for more details and a procedure to determine the values of p and q by means of the so called extended Euclid's algorithm. For a decimating filter with integer factor D the other parameters are: $I = 1$, $p = 0$ and $q = 1$. For an interpolating filter with integer factor I the other parameters are: $D = 1$, $p = 1$ and $q = 0$.

Lemma 5.7 (Nested polyphase structure)**.** *Any multirate filter $H(z) \in \mathbb{C}(z)$ with interpolating factor I and decimating factor D and $pD + qI = 1$ with $p, q \in \mathbb{Z}$, can be reformulated as:*

$$\uparrow I; H(z); \downarrow D = \sum_{d\in\mathcal{D}} \left(z^{-dq}; \downarrow D; \sum_{i\in\mathcal{I}} \left(\left(z^{-dp} H_{D:d}(z) \right)_{I:i}(z); \uparrow I; z^{-i} \right) \right),$$

or:

$$\uparrow I; H(z); \downarrow D = \sum_{d\in\mathcal{I}} \left(\sum_{d\in\mathcal{D}} \left(z^{-d}; \downarrow D; \left(H_{I:i}(z) z^{-iq} \right)_{D:d}(z) \right); \uparrow I; z^{-ip} \right).$$

Proof. The polyphase structures from Lemma 5.5 and Lemma 5.6 are used together with the split-delay identity, Lemma A.7, both noble identities, Lemma A.3 and Lemma A.4 and the first prime identity, Lemma A.5, giving:

$$\begin{aligned}
\uparrow I; H(z); \downarrow D =& \uparrow I; \sum_{d\in\mathcal{D}} \left(z^{-d}; \downarrow D; H_{D:d}(z) \right) \\
=& \uparrow I; \sum_{d\in\mathcal{D}} \left(z^{-dqI} z^{-dpD}; \downarrow D; H_{D:d}(z) \right) \\
=& \sum_{d\in\mathcal{D}} \left(\uparrow I; z^{-dqI} z^{-dpD}; \downarrow D; H_{D:d}(z) \right) \\
=& \sum_{d\in\mathcal{D}} \left(z^{-dq}; \downarrow D; \uparrow I; \left(z^{-dp} H_{D:d}(z) \right) \right) \\
=& \sum_{d\in\mathcal{D}} \left(z^{-dq}; \downarrow D; \sum_{i\in\mathcal{I}} \left(\left(z^{-dp} H_{D:d}(z) \right)_{I:i}(z); \uparrow I; z^{-i} \right) \right),
\end{aligned}$$

or:

$$\begin{aligned}
\uparrow I; H(z); \downarrow D =& \sum_{i\in\mathcal{I}} \left(H_{I:i}(z); \uparrow I; z^{-i} \right); \downarrow D \\
=& \sum_{i\in\mathcal{I}} \left(H_{I:i}(z); \uparrow I; z^{-iqI} z^{-ipD} \right); \downarrow D \\
=& \sum_{i\in\mathcal{I}} \left(H_{I:i}(z); \uparrow I; z^{-iqI} z^{-ipD}; \downarrow D \right) \\
=& \sum_{i\in\mathcal{I}} \left(\left(H_{I:i}(z) z^{-iq} \right); \downarrow D; \uparrow I; z^{-ip} \right) \\
=& \sum_{i\in\mathcal{I}} \left(\sum_{d\in\mathcal{D}} \left(z^{-d}; \downarrow D; \left(H_{I:i}(z) z^{-iq} \right)_{D:d}(z) \right); \uparrow I; z^{-ip} \right). \qquad \square
\end{aligned}$$

In the following example both types of nested polyphase structures are determined for $I = 2$ and $D = 3$.

Example 5.1. For interpolation factor $I = 2$ and decimation factor $D = 3$, $p = 1$ and $q = -1$, the nested polyphase structures related to both formulations from Lemma 5.7 are depicted in Figure 5.4(b) and Figure 5.4(c).

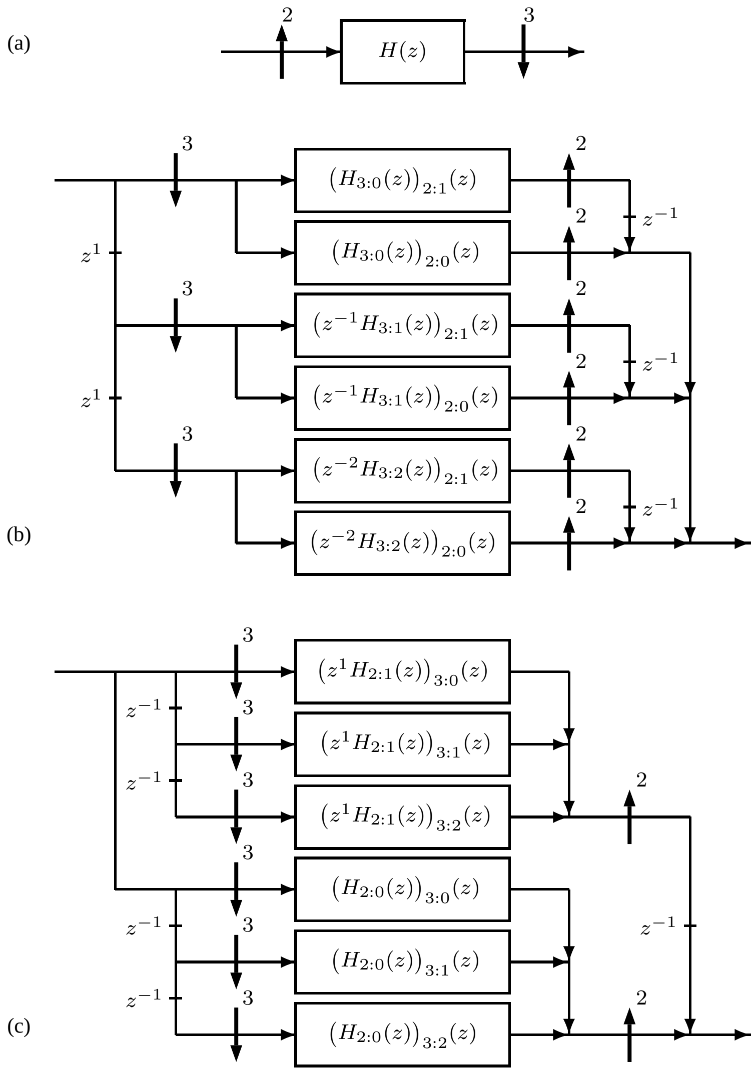


Figure 5.4: *Examples of nested polyphase structures in case of a rational decimation factor ($\frac{3}{2}$), $I = 2$ and $D = 3$.*

The respective expressions, with $\mathcal{D} = \{0, 1, 2\}$ and $\mathcal{I} = \{0, 1\}$, are:

$$\uparrow 2; H(z); \downarrow 3 = \sum_{d=0}^{2} \left(z^d; \downarrow 3; \sum_{i=0}^{1} \left(\left(z^{-d} H_{3:d}(z) \right)_{2:i}(z); \uparrow 2; z^{-i} \right) \right),$$

or:

$$\uparrow 2; H(z); \downarrow 3 = \sum_{i=0}^{1} \left(\sum_{d=0}^{2} \left(z^{-d}; \downarrow 3; \left(H_{2:i}(z) z^i \right)_{3:d}(z) \right); \uparrow 2; z^{-i} \right),$$

Application of the polyphase identity, Lemma 5.2, shows that all the subfilters are, possible delayed, PPCs with decomposition factor $ID = 6$ of $H(z)$.

r			Figure 5.4(b)		Figure 5.4(c)
0	$H_{6:0}(z)$	=	$\left(H_{3:0}(z)\right)_{2:0}(z)$	=	$\left(H_{2:0}(z)\right)_{3:0}(z)$
1	$H_{6:1}(z)$	=	$\left(z^{-1}H_{3:1}(z)\right)_{2:1}(z)$	=	$z^{-1}\left(z^{1}H_{2:1}(z)\right)_{3:2}(z)$
2	$H_{6:2}(z)$	=	$z\left(z^{-2}H_{3:2}(z)\right)_{2:0}(z)$	=	$\left(H_{2:0}(z)\right)_{3:1}(z)$
3	$H_{6:3}(z)$	=	$\left(H_{3:0}(z)\right)_{2:1}(z)$	=	$\left(z^{1}H_{2:1}(z)\right)_{3:0}(z)$
4	$H_{6:4}(z)$	=	$z\left(z^{-1}H_{3:1}(z)\right)_{2:0}(z)$	=	$\left(H_{2:0}(z)\right)_{3:2}(z)$
5	$H_{6:5}(z)$	=	$z\left(z^{-2}H_{3:2}(z)\right)_{2:1}(z)$	=	$\left(z^{1}H_{2:1}(z)\right)_{3:1}(z)$

Basically, the schemes differ only in the structure of the multiplex and demultiplex networks. These networks can be made more efficient by combining some of the adders and delay elements (not shown). Other examples can be found in for instance [133] and [135].
End of example

5.4 Unified polyphase structure

In the nested polyphase structure there exists an asymmetry in the structure. The order in which both polyphase decompositions are applied, affects the resulting structure. In this section the *unified polyphase structure* is presented where the asymmetry is absent. In [135] a structure similar to the unified polyphase structure in, e.g., Figure 5.5(c) is shown. In the reference, equal delays and the SRDs at the input, are combined. For each of the PPCs in Figure 5.4(b) and Figure 5.4(c), there is a route through both the demultiplexer and the multiplexer. The unified polyphase structure shows each possible route from input to output as a parallel branch or path. For the following discussion, and also for later when symmetry will be restored, the path of a multirate filter is defined first.

Definition 5.3 (Path)**.** *The* $\mathrm{path}(r)$ *of a multirate filter* $H(z) \in \mathbb{C}(z)$*, with interpolation factor* I*, decimation factor* D*,* $pD + qI = 1$ *with* $p, q \in \mathbb{Z}$*, and* $R = ID$*, is defined as:*

$$\mathrm{path}(r) \triangleq z^{-rq}; \downarrow D; H_{R:r}(z); \uparrow I; z^{-rp}.$$

Next it will be shown that on basis of this definition any multirate filter can be formulated as a sum of paths.

Lemma 5.8 (Unified polyphase structure)**.** *Any multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I, decimation factor D, $pD + qI = 1$ with $p, q \in \mathbb{Z}$, and $R = ID$, can be reformulated as:*

$$\uparrow I; H(z); \downarrow D = \sum_{r \in \mathcal{R}} \mathrm{path}(r).$$

Proof. The polyphase decomposition, Lemma 5.3, the split delay identity, Lemma A.7, both noble identities, Lemma A.3 and Lemma A.4, the first prime identity, Lemma A.5 and Definition 5.3 for the path, are used:

$$\begin{aligned}
\uparrow I; H(z); \downarrow D &= \uparrow I; \sum_{r \in \mathcal{R}} \big(z^{-r} H_{R:r}(z^R)\big); \downarrow D \\
&= \uparrow I; \sum_{r \in \mathcal{R}} \big(z^{-rqI} H_{R:r}(z^R) z^{-rpD}\big); \downarrow D \\
&= \sum_{r \in \mathcal{R}} \big(z^{-rq}; \uparrow I; H_{R:r}(z^R); \downarrow D; z^{-rp}\big) \\
&= \sum_{r \in \mathcal{R}} \big(z^{-rq}; \downarrow D; H_{R:r}(z); \uparrow I; z^{-rp}\big) \\
&= \sum_{r \in \mathcal{R}} \mathrm{path}(r).
\end{aligned}$$

□

The unified polyphase structure is clarified in the next example.

Example 5.2. Continuation of Example 5.1 with Lemma 5.8 gives:

$$\uparrow 2; H(z); \downarrow 3 = \sum_{r=0}^{5} \big(z^r; \downarrow 3; H_{6:r}(z); \uparrow 2; z^{-r}\big),$$

and the related scheme is shown in Figure 5.5(b). Recall that $I = 2$, $D = 3$, $p = 1$ and $q = -1$. Now, the delays in front of the SRDs and behind the SRIs can be normalized by means of the noble identities, such that $d \in \mathcal{D}_0$ and $i \in \mathcal{I}_0$. In this process delay elements moved towards the PPCs may cancel. In Figure 5.5(c) the result is shown. Note that only for PPC $H_{6:1}(z)$ the delay z^1 remains and that all other PPCs have no additional delay.
End of example

In Lemma 5.9 it will be shown that for $r \in \mathcal{R}_0$ the remaining delays can be limited to z^0 and z^1, and in Lemma 5.10 it will be shown that both values appear in every unified polyphase structure with normalized delays.

Lemma 5.9. *Any* $\mathrm{path}(r)$ *with* $r \in \mathcal{R}_0$ *can be rewritten in the normalized form as:*

$$\mathrm{path}(r) = z^{-(rq)|_D}; \downarrow D; z^{-x_r} H_{R:r}(z); \uparrow I; z^{-(rp)|_I},$$

with $x_r = \lfloor \frac{rq}{D} \rfloor + \lfloor \frac{rp}{I} \rfloor$ *and* $x_r \in \{-1, 0\}$.

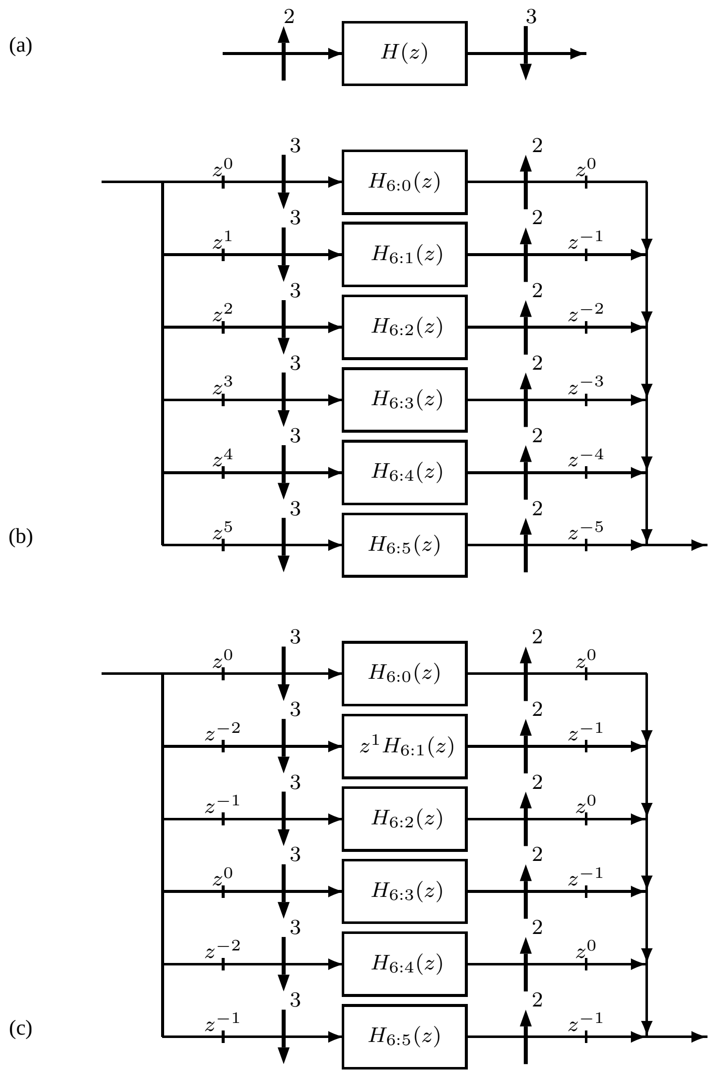


Figure 5.5: *Examples of unified polyphase structures in case of a rational decimation factor ($\frac{3}{2}$), $I = 2$ and $D = 3$.*

Proof. By the noble identities, Lemma A.3 and Lemma A.4, it follows that:

$$\begin{aligned}\mathrm{path}(r) &= z^{-rq}; \downarrow D; H_{R:r}(z); \uparrow I; z^{-rp} \\ &= z^{-(rq)|_D}; \downarrow D; z^{-x_r} H_{R:r}(z); \uparrow I; z^{-(rp)|_I},\end{aligned}$$

and is easy to check that $x_r = \lfloor \frac{rq}{D} \rfloor + \lfloor \frac{rp}{I} \rfloor$. Using that $a - 1 < \lfloor a \rfloor \leq a$ for $a \in \mathbb{R}$, gives $\frac{rq}{D} + \frac{rp}{I} - 2 < x_r \leq \frac{rq}{D} + \frac{rp}{I}$. Since $pD + qI = 1$ and $ID = R$, this results in $\frac{r}{R} - 2 < x_r \leq \frac{r}{R}$. By $r \in \mathcal{R}_0$ and $x_r \in \mathbb{Z}$ it follows that $x_r \in \{-1, 0\}$. ☐

Lemma 5.10. *For $I \geq 2$, $D \geq 2$, $pD + qI = 1$ and $r \in \mathcal{R}_0$:*

- *there exists an r such that $x_r = 0$,*
- *there exists an r such that $x_r = -1$.*

Proof. It will be proven that $x_0 = 0$ and $x_1 = -1$. Since $I \geq 2$ and $D \geq 2$, the values of $r = 0$ and $r = 1$ exist. Evaluation of the expression for x_r of Lemma 5.9 in case $r = 0$, gives $x_0 = 2\lfloor 0 \rfloor = 0$. Evaluation of the expression for x_r in case $r = 1$, gives $x_1 = \lfloor \frac{q}{D} \rfloor + \lfloor \frac{p}{I} \rfloor$. Since $pD+qI = 1$, the expression for x_1 is $x_1 = \lfloor \frac{1}{ID} - \frac{q}{D} \rfloor + \lfloor \frac{q}{D} \rfloor$ and write $q = kD + l$ with $k, l \in \mathbb{Z}$ and $1 \leq l < D$. Now $x_1 = \lfloor \frac{1}{ID} - k - \frac{l}{D} \rfloor + \lfloor k + \frac{l}{D} \rfloor = \lfloor \frac{1}{ID} - \frac{l}{D} \rfloor + \lfloor \frac{l}{D} \rfloor$. From $\frac{1}{ID} < \frac{l}{D} < 1$ it follows that $x_1 = -1$. □

The unified polyphase structure with its paths in the normalized, form will always contain the non-causal delay z^1 in combination with some of the PPCs. In case causality is relevant, this non-causal delay can be removed by increasing the overall delay of the multirate filter. Assuming all PPCs being causal, adding the delay z^{-D} in front of the system or z^{-I} behind the system will, after application of the noble identities, result in a causal system.

5.5 Relations between PPCs

In the previous discussions on the polyphase decomposition and structures, it was not relevant whether the filter itself was symmetric or not. The remainder of this chapter typically deals with the polyphase decomposition and structures of symmetric filters. In general, the PPCs of a (σ, μ)-symmetric filter are not symmetric as will be illustrated in Example 5.3. In this section it will be shown that a PPC of a symmetric filter is either symmetric or a second PPC exists that is a mirrored version of the first PPC. This property together with the mirrored-pair identity, see Lemma 2.18, form the basis for the restoration of symmetry as will be discussed in the next section.

Example 5.3. As an example consider the $(\sigma, 3)$-symmetric filter $H(z)$:

$$H(z) = \langle a, b, c, \sqrt{\sigma} d, \sigma c^*, \sigma b^*, \sigma a^* \rangle.$$

The two PPCs $H_{2:0}(z)$ and $H_{2:1}(z)$ have $(\sigma, \frac{3}{2})$ and $(\sigma, 1)$-symmetry respectively:

$$H_{2:0}(z) = \langle a, c, \sigma c^*, \sigma a^* \rangle,$$
$$H_{2:1}(z) = \langle b, \sqrt{\sigma} d, \sigma b^* \rangle.$$

A second example is the $(\sigma, \frac{5}{2})$-symmetric filter $G(z)$:

$$G(z) = \langle a, b, c, \sigma c^*, \sigma b^*, \sigma a^* \rangle.$$

The two PPCs $G_{2:0}(z)$ and $G_{2:1}(z)$ have no symmetry but are mutually mirrored versions:

$$G_{2:0}(z) = \langle a, c, \sigma b^* \rangle \quad = \mathcal{M}_{\sigma,1}\big(G_{2:1}(z)\big),$$
$$G_{2:1}(z) = \langle b, \sigma c^*, \sigma a^* \rangle = \mathcal{M}_{\sigma,1}\big(G_{2:0}(z)\big).$$

End of example

In Lemma 5.11 it will be shown that for any $R : r$ PPC of the (σ, μ)-symmetric filter $H(z)$, there exists an $R : s$ PPC that is mutually mirrored with respect to the center of symmetry μ_0. Different values of μ_0 result in different decomposition indices s that, however, refer to the same PPC, see Lemma 5.2.

Lemma 5.11. *For any (σ, μ)-symmetric filter $H(z) \in \mathbb{C}(z)$:*

$$\mathcal{M}_{\sigma,\mu_0}\big(H_{R:r}(z)\big) = H_{R:s}(z),$$

with $s = 2\mu - 2R\mu_0 - r$ and $s \in \mathbb{Z}$.

Proof. Applying Definition 5.1 and Definition 2.2 gives:

$$\begin{aligned}
\mathcal{M}_{\sigma,\mu_0}\big(H_{R:r}(z)\big) &= \mathcal{M}_{\sigma,\mu_0}\left(\sum_n h[nR + r]z^{-n}\right) \\
&= z^{-2\mu_0}\sigma \sum_n h^*[nR + r]z^{n} \\
&= z^{-2\mu_0} \sum_n h[2\mu - nR - r]z^{n} \\
&= z^{-2\mu_0} \sum_n h[nR + 2\mu - r]z^{-n} \\
&= z^{-2\mu_0} H_{R:(2\mu - r)}(z) \\
&= H_{R:(2\mu - 2R\mu_0 - r)}(z) \\
&= H_{R:s}(z) \text{ with } s = 2\mu - 2R\mu_0 - r.
\end{aligned}$$

From $\mu, \mu_0 \in \mathbb{Z}/2$, $R \in \mathbb{N}^+$ and $r \in \mathbb{Z}$, it follows that $s \in \mathbb{Z}$. For a given value of r, every μ_0 gives a value for s that refers to the same PPC except for a delay. This is because of the equality $s|_R = (2\mu - 2R\mu_0 - r)|_R$, that does not depend on μ_0. ☐

From Example 5.3 it is clear that symmetric PPCs not necessarily exist. However, in case a center of symmetry $\mu_0 \in \mathbb{Z}/2$ exists such that the decomposition indices $r = s$, the related $R : r$ PPC is symmetric.

5.6 Restoration of symmetry

This section forms the central part of the chapter. Here it will be shown that if a PPC in a unified polyphase structure of a symmetric filter is non-symmetric, this PPC can be combined with other non-symmetric PPCs, such that the filters in the resulting structure all are symmetric. This is formulated in the following theorem.

Theorem 5.1 (Symmetry restoration)**.** *Any (σ, μ)-symmetric multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I and decimation factor D, with $\gcd(I, D) = 1$, can be constructed from $R = ID$ symmetric filters in a polyphase structure.*

In Section 5.6.1 the proof of this theorem is presented, and in Section 5.6.2 a possible procedure for restoring the symmetry is given. In Section 5.7 the theorem is applied to some typical examples.

5.6.1 Proof of symmetry restoration theorem

For the proof of the theorem some lemmas are needed that will be presented first. The proofs of these lemmas can be used later as a recipe for the construction of the symmetric polyphase structure.
The assumption in the theorem that $\gcd(I, D) = 1$, implies that there exist $p, q \in \mathbb{Z}$ such that $pD + qI = 1$ (Euclid). This is used in the following series of lemmas. Also the *Chinese Remainder Theorem (CRT)* will be used. The CRT states that if $\gcd(I, D) = 1$, the pair $(r|_D, r|_I)$ has a one-to-one relation to $r|_R$ with $R = ID$ and $0 \le r < R$.

Lemma 5.12. *If for the (σ, μ)-symmetric multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I, decimation factor D, $pD + qI = 1$ and $s = 2\mu - 2R\mu_0 - r$, the relation $(r|_D = s|_D) \wedge (r|_I = s|_I)$ holds, then $\mathrm{path}(r)$ comprises a symmetric filter.*

Proof. By assumption and the CRT, $r|_R = s|_R$ so indices r and s refer to the same PPC. If $r = s$, Lemma 5.11 and Lemma 2.7 give that $H_{R:r}(z)$ is (σ, μ_0)-symmetric. To make $r = s$ let $\mu_0 = \frac{2\mu-2r}{2R}$. Since $r|_R = s|_R$ the relation $(2\mu)|_R = (2r)|_R$ holds, and makes $\mu_0 \in \mathbb{Z}/2$. See Figure 5.6(a) for the related structure. □

Lemma 5.13. *If for the (σ, μ)-symmetric multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I, decimation factor D, $pD + qI = 1$ and $s = 2\mu - 2R\mu_0 - r$, the relation $(r|_D \neq s|_D) \wedge (r|_I = s|_I)$ holds, then the 2 parallel paths, $\mathrm{path}(r)$ and $\mathrm{path}(s)$, can be constructed using 2 symmetric filters.*

Proof. By assumption and the CRT, indices r and s refer to different PPCs since $r|_D \neq s|_D$, and, by Lemma 5.11, $\mathcal{M}_{\sigma,\mu_0}(H_{R:r}(z)) = H_{R:s}(z)$. The symmetric construction is now as follows:

$$\begin{aligned}
&\mathrm{path}(r) + \mathrm{path}(s) = \\
&= \begin{bmatrix} \mathrm{path}(r) \\ \mathrm{path}(s) \end{bmatrix}; \begin{bmatrix} 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} z^{-rq}; \downarrow D; H_{R:r}(z); \uparrow I; z^{-rp} \\ z^{-sq}; \downarrow D; H_{R:s}(z); \uparrow I; z^{-sp} \end{bmatrix}; \begin{bmatrix} 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} z^{-rq}; \downarrow D; \quad H_{R:r}(z) \\ z^{-sq}; \downarrow D; z^{-x}; H_{R:s}(z) \end{bmatrix}; \begin{bmatrix} 1 & 1 \end{bmatrix}; \uparrow I; z^{-rp} \\
&= \begin{bmatrix} z^{-rq}; \downarrow D \\ z^{-sq}; \downarrow D; z^{-x} \end{bmatrix}; \mathbf{H}(z); \uparrow I; z^{-rp},
\end{aligned}$$

and:

$$\mathbf{H}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-a} \end{bmatrix}; \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \begin{bmatrix} E_0(z) & E_1(z) \end{bmatrix},$$

with the arbitrary delay z^{-a} and the delay z^{-x}, with $x = \frac{(s-r)p}{I}$, to compensate for the difference between delay z^{-rp} and delay z^{-sp}. From $s|_I = r|_I$ it follows that $x \in \mathbb{Z}$.

The 2 new filters are:

$$E_0(z) = \tfrac{1}{2}\big(H_{R:r}(z) + z^a H_{R:s}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{R:r}(z) - z^a H_{R:s}(z)\big).$$

By assumption and application of Lemma 2.11, $z^a H_{R:s}(z) = \mathcal{M}_{\sigma,\mu_0-\frac{a}{2}}(H_{R:r}(z))$. Now by Lemma 2.17 the filter $E_0(z)$ is $(\sigma, \mu_0 - \frac{a}{2})$-symmetric and the filter $E_1(z)$ is $(-\sigma, \mu_0 - \frac{a}{2})$-symmetric. See Figure 5.6(b) for the related structure. □

Lemma 5.14. *If for the (σ,μ)-symmetric multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I, decimation factor D, $pD + qI = 1$ and $s = 2\mu - 2R\mu_0 - r$, the relation $(r|_D = s|_D) \wedge (r|_I \neq s|_I)$ holds, then the 2 parallel paths, $\mathrm{path}(r)$ and $\mathrm{path}(s)$, can be constructed using 2 symmetric filters.*

Proof. By assumption and the CRT, indices r and s refer to different PPCs since $r|_I \neq s|_I$, and, by Lemma 5.11, $\mathcal{M}_{\sigma,\mu_0}(H_{R:r}(z)) = H_{R:s}(z)$. The symmetric construction is now as follows:

$$\begin{aligned}
&\mathrm{path}(r) + \mathrm{path}(s) = \\
&= \begin{bmatrix}\mathrm{path}(r)\\ \mathrm{path}(s)\end{bmatrix}; \begin{bmatrix}1 & 1\end{bmatrix} \\
&= \begin{bmatrix} z^{-rq}; \downarrow D; H_{R:r}(z); \uparrow I; z^{-rp} \\ z^{-sq}; \downarrow D; H_{R:s}(z); \uparrow I; z^{-sp}\end{bmatrix}; \begin{bmatrix}1 & 1\end{bmatrix} \\
&= z^{-rq}; \downarrow D; \begin{bmatrix} H_{R:r}(z); \quad\quad \uparrow I; z^{-rp} \\ H_{R:s}(z); z^{-x}; \uparrow I; z^{-sp}\end{bmatrix}; \begin{bmatrix}1 & 1\end{bmatrix} \\
&= z^{-rq}; \downarrow D; \mathbf{H}(z); \begin{bmatrix}\uparrow I; z^{-rp} & z^{-x}; \uparrow I; z^{-sp}\end{bmatrix},
\end{aligned}$$

and:

$$\mathbf{H}(z) = \begin{bmatrix}E_0(z)\\ E_1(z)\end{bmatrix}; \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}; \begin{bmatrix}1 & 0\\ 0 & z^{-b}\end{bmatrix},$$

with the arbitrary delay z^{-b} and the delay z^{-x}, with $x = \frac{(s-r)q}{D}$, to compensate for the difference between delay z^{-rq} and delay z^{-sq}. From $s|_D = r|_D$ it follows that $x \in \mathbb{Z}$. The 2 new filters are:

$$E_0(z) = \tfrac{1}{2}\big(H_{R:r}(z) + z^b H_{R:s}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{R:r}(z) - z^b H_{R:s}(z)\big).$$

By assumption and application of Lemma 2.11, $z^b H_{R:s}(z) = \mathcal{M}_{\sigma,\mu_0-\frac{b}{2}}(H_{R:r}(z))$. Now by Lemma 2.17, the filter $E_0(z)$ is $(\sigma, \mu_0 - \frac{b}{2})$-symmetric and the filter $E_1(z)$ is $(-\sigma, \mu_0 - \frac{b}{2})$-symmetric. See Figure 5.6(c) for the related structure. □

Lemma 5.15. *If for the (σ,μ)-symmetric multirate filter $H(z) \in \mathbb{C}(z)$, with interpolation factor I, decimation factor D, $pD+qI=1$ and $s_0 = 2\mu - 2R\mu_0 - r_0$, the relation $(r_0|_D \neq s_0|_D) \wedge (r_0|_I \neq s_0|_I)$ holds, then for $r_1 = s_0pD + r_0qI$ and $s_1 = r_0pD + s_0qI$ the 4 parallel paths, $\mathrm{path}(r_0)$, $\mathrm{path}(s_0)$, $\mathrm{path}(r_1)$ and $\mathrm{path}(s_1)$, can be constructed using 4 symmetric filters.*

Proof. It will be shown that $\mathrm{path}(r_1)$ and $\mathrm{path}(s_1)$ differ from $\mathrm{path}(r_0)$ and $\mathrm{path}(s_0)$, and that their filters are also mutually mirrored with center of symmetry μ_0. This will be followed by the symmetric construction.
From the definitions of r_1 and s_1 follows that $r_1|_D = (r_0qI)|_D = (r_0(1-pD))|_D = r_0|_D$ and similarly $s_1|_D = s_0|_D$, $r_1|_I = s_0|_I$ and $s_1|_I = r_0|_I$. From the relation between r_0 and s_0, it now follows that $r_0|_R$, $s_0|_R$, $r_1|_R$ and $s_1|_R$ are different, so r_0, s_0, r_1 and s_1 refer to different paths. Also the values of $r_1|_R$ and $s_1|_R$ are fully determined by $r_0|_R$. Furthermore $(r_1|_D \neq s_1|_D) \wedge (r_1|_I \neq s_1|_I)$ so it is not possible to use Lemma 5.13 or Lemma 5.14 for $\mathrm{path}(r_1)$ and $\mathrm{path}(s_1)$.
By Lemma 5.11, $\mathcal{M}_{\sigma,\mu_0}(H_{R:r_0}(z)) = H_{R:s_0}(z)$. Using the assumptions and definitions it is easy to verify that $s_1 = 2\mu - 2R\mu_0 - r_1$. Observe that $r_1 + s_1 = s_0pD + r_0qI + r_0pD + s_0qI = s_0 + r_0 = 2\mu - 2R\mu_0$. As a consequence $\mathcal{M}_{\sigma,\mu_0}(H_{R:r_1}(z)) = H_{R:s_1}(z)$. The symmetric construction is now as follows:

$$
\begin{aligned}
&\mathrm{path}(r_0) + \mathrm{path}(s_0) + \mathrm{path}(r_1) + \mathrm{path}(s_1) = \\
&= \begin{bmatrix} \mathrm{path}(r_0) \\ \mathrm{path}(s_0) \\ \mathrm{path}(r_1) \\ \mathrm{path}(s_1) \end{bmatrix}; \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} z^{-r_0q}; \downarrow D; H_{R:r_0}(z); \uparrow I; z^{-r_0p} \\ z^{-s_0q}; \downarrow D; H_{R:s_0}(z); \uparrow I; z^{-s_0p} \\ z^{-r_1q}; \downarrow D; H_{R:r_1}(z); \uparrow I; z^{-r_1p} \\ z^{-s_1q}; \downarrow D; H_{R:s_1}(z); \uparrow I; z^{-s_1p} \end{bmatrix}; \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} z^{-r_0q}; \downarrow D; & H_{R:r_0}(z); \uparrow I; z^{-r_0p} \\ z^{-s_0q}; \downarrow D; & H_{R:s_0}(z); \uparrow I; z^{-s_0p} \\ z^{-r_0q}; \downarrow D; z^{-x_0}; & H_{R:r_1}(z); \uparrow I; z^{-s_0p} \\ z^{-s_0q}; \downarrow D; z^{-x_1}; & H_{R:s_1}(z); \uparrow I; z^{-r_0p} \end{bmatrix}; \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} z^{-r_0q}; \downarrow D \\ z^{-s_0q}; \downarrow D \end{bmatrix}; \mathbf{H}(z); \begin{bmatrix} \uparrow I; z^{-r_0p} & \uparrow I; z^{-s_0p} \end{bmatrix},
\end{aligned}
$$

and:

$$
\mathbf{H}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-a} \end{bmatrix}; \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \begin{bmatrix} E_0(z) & E_2(z) \\ E_1(z) & E_3(z) \end{bmatrix}; \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \begin{bmatrix} 1 & 0 \\ 0 & z^{-b} \end{bmatrix},
$$

with the arbitrary delays z^{-a} and z^{-b}. The delays z^{-x_0} and z^{-x_1} are to compensate for the differences between the delays in front of the SRDs and behind the SRIs. For delay z^{-x_0} it is found that: $x_0 = \frac{(r_1-r_0)q}{D} + \frac{(r_1-s_0)p}{I} = \frac{(s_0pD+r_0(qI-1))q}{D} + \frac{(s_0(pD-1)+r_0qI)p}{I} =$

$(s_0 - r_0)pq + (r_0 - s_0)pq = 0$. Similarly it is found that: $x_1 = \frac{(s_1 - s_0)q}{D} + \frac{(s_1 - r_0)p}{I} = 0$. The 4 new filters are:

$$
\begin{aligned}
E_0(z) &= \tfrac{1}{4}\big(H_{R:r_0}(z) + z^a H_{R:s_1}(z) + z^b H_{R:r_1}(z) + z^{a+b} H_{R:s_0}(z)\big),\\
E_1(z) &= \tfrac{1}{4}\big(H_{R:r_0}(z) + z^a H_{R:s_1}(z) - z^b H_{R:r_1}(z) - z^{a+b} H_{R:s_0}(z)\big),\\
E_2(z) &= \tfrac{1}{4}\big(H_{R:r_0}(z) - z^a H_{R:s_1}(z) + z^b H_{R:r_1}(z) - z^{a+b} H_{R:s_0}(z)\big),\\
E_3(z) &= \tfrac{1}{4}\big(H_{R:r_0}(z) - z^a H_{R:s_1}(z) - z^b H_{R:r_1}(z) + z^{a+b} H_{R:s_0}(z)\big).
\end{aligned}
$$

By assumption and application of Lemma 2.11, $z^a H_{R:s_1}(z) = \mathcal{M}_{\sigma,\mu_0 - \frac{a+b}{2}}\left(z^b H_{R:r_1}(z)\right)$ and $z^{a+b} H_{R:s_0}(z) = \mathcal{M}_{\sigma,\mu_0 - \frac{a+b}{2}}\left(H_{R:r_0}(z)\right)$. Now by Lemma 2.17 the filters $E_0(z)$ and $E_3(z)$ are $(\sigma, \mu_0 - \frac{a+b}{2})$-symmetric and the filters $E_1(z)$ and $E_2(z)$ are $(-\sigma, \mu_0 - \frac{a+b}{2})$-symmetric. See Figure 5.6(d) for the related structure. □

The proof of Theorem 5.1 on page 94 now is as follows.

Proof. The assumption $\gcd(I, D) = 1$ implies that there exist $p, q \in \mathbb{Z}$ such that $pD + qI = 1$ (Euclid).
From Lemma 5.8 and Definition 5.2 it follows that exactly $R = ID$ paths, $\mathrm{path}(r)$ with $r \in \mathcal{R}$, have to be considered. For any $r \in \mathcal{R}$, there is an index $s = 2\mu - 2R\mu_0 - r$ such that exactly one of the following four rules applies (CRT):

Rule a: $(r|_D = s|_D) \wedge (r|_I = s|_I)$ apply Lemma 5.12,

Rule b: $(r|_D \neq s|_D) \wedge (r|_I = s|_I)$ apply Lemma 5.13,

Rule c: $(r|_D = s|_D) \wedge (r|_I \neq s|_I)$ apply Lemma 5.14,

Rule d: $(r|_D \neq s|_D) \wedge (r|_I \neq s|_I)$ apply Lemma 5.15.

Rule a applies to a single path, whereas Rule b and Rule c both apply to 2 paths, and Rule d applies to 4 paths. Lemma 5.12 shows that for index r the filter is symmetric, Lemma 5.13 and Lemma 5.14 show how 2 new symmetric filters are constructed, and Lemma 5.15 shows how 4 new symmetric filters are constructed. □

Theorem 5.1 is applicable only if D and I are coprime. In case D and I have a common factor the second prime identity, Lemma A.6, can be used to delete this factor. As a consequence however, the resulting filter may be non-symmetric even if the original filter was. Note that the savings by restoring symmetry are at most a factor of 2, whereas the savings by application of the second prime identity are at least a factor of 2.

5.6.2 Procedure for restoring symmetry

The proof of Theorem 5.1 is constructive. A procedure for restoring the symmetry in the polyphase structure of a symmetric filter could be along the lines as inspired by this proof. As a consequence the following part resembles the proof as presented before, in a great extent.

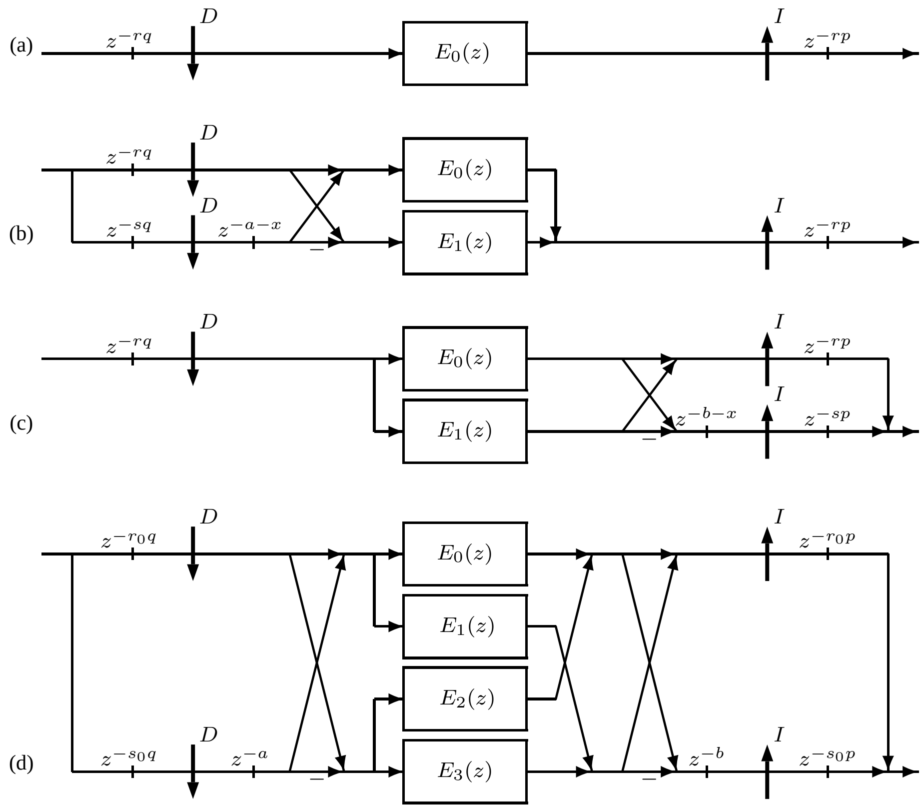


Figure 5.6: *The four possible constructions to restore symmetry.*

First, the multirate-filter specification is verified in Step 1 to determine if the procedure is applicable, and some parameters are derived in Step 2. Step 3, Step 4 and Step 5 are used repeatedly until all filters in the polyphase structure are symmetric. In Step 5 one of the rules as given in the proof of the theorem on the preceding page, Rule a, Rule b, Rule c or Rule d, will be applied. The value of the parameters a and b can be chosen to optimize the resulting filters $E_i(z)$.

Step 1: Evaluate the multirate filter specification.

1. Is filter $H(z)$ (σ, μ)-symmetric?
2. Are I and D coprime, i.e. $\gcd(I, D) = 1$?
3. If both conditions hold, then continue with Step 2, otherwise this procedure is not applicable.

Step 2: Determine the values $p, q \in \mathbb{Z}$ in $pD + qI = 1$ and let $R = ID$.

Step 3: The following steps have to be repeated until every index $r \in \mathcal{R}$ is considered either directly by selection, or implied by Step 4 and/or Step 5-Rule d.

Step 4: Determine index $s = 2\mu - 2R\mu_0 - r$ for any $\mu_0 \in \mathbb{Z}/2$.

Step 5: Determine $r|_D$, $r|_I$, $s|_D$and $s|_I$ and select the applicable rule.

Rule a: If $(r|_D = s|_D) \wedge (r|_I = s|_I)$, then:

$$\mathrm{path}(r) = z^{-rq}; \downarrow D; E_0(z); \uparrow I; z^{-rp}.$$

The new filter:

$$E_0(z) = \tfrac{1}{2}\big(H_{R:r}(z) + H_{R:s}(z)\big),$$

is (σ, μ_0)-symmetric. For $\mu_0 = \frac{\mu - r}{R}$: $r = s$ so $E_0(z) = H_{R:r}(z)$. The related structure is found in Figure 5.6(a).

Rule b: If $(r|_D \neq s|_D) \wedge (r|_I = s|_I)$, then with the arbitrary delay z^{-a} and the delay z^{-x}, with $x = \frac{(s-r)p}{I}$, the two new filters are:

$$E_0(z) = \tfrac{1}{2}\big(H_{R:r}(z) + z^a H_{R:s}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{R:r}(z) - z^a H_{R:s}(z)\big).$$

Filter $E_0(z)$ is $(\sigma, \mu_0 - \frac{a}{2})$-symmetric and filter $E_1(z)$ is $(-\sigma, \mu_0 - \frac{a}{2})$-symmetric. The related structure is found in Figure 5.6(b).

Rule c: If $(r|_D = s|_D) \wedge (r|_I \neq s|_I)$, then with the arbitrary delay z^{-b} and the delay z^{-x}, with $x = \frac{(s-r)q}{D}$, the two new filters are:

$$E_0(z) = \tfrac{1}{2}\big(H_{R:r}(z) + z^b H_{R:s}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{R:r}(z) - z^b H_{R:s}(z)\big).$$

Filter $E_0(z)$ is $(\sigma, \mu_0 - \frac{b}{2})$-symmetric and filter $E_1(z)$ is $(-\sigma, \mu_0 - \frac{b}{2})$-symmetric. The related structure is found in Figure 5.6(c).

Rule d: If $(r|_D \neq s|_D) \wedge (r|_I \neq s|_I)$, then let indices $r_0 = r$, $s_0 = s$, $r_1 = s_0 pD + r_0 qI$ and $s_1 = r_0 pD + s_0 qI$, and with the arbitrary delays z^{-a} and z^{-b}, the four new filters are:

$$E_0(z) = \tfrac{1}{4}\big(H_{R:r_0}(z) + z^a H_{R:s_1}(z) + z^b H_{R:r_1}(z) + z^{a+b} H_{R:s_0}(z)\big),$$
$$E_1(z) = \tfrac{1}{4}\big(H_{R:r_0}(z) + z^a H_{R:s_1}(z) - z^b H_{R:r_1}(z) - z^{a+b} H_{R:s_0}(z)\big),$$
$$E_2(z) = \tfrac{1}{4}\big(H_{R:r_0}(z) - z^a H_{R:s_1}(z) + z^b H_{R:r_1}(z) - z^{a+b} H_{R:s_0}(z)\big),$$
$$E_3(z) = \tfrac{1}{4}\big(H_{R:r_0}(z) - z^a H_{R:s_1}(z) - z^b H_{R:r_1}(z) + z^{a+b} H_{R:s_0}(z)\big).$$

Filters $E_0(z)$ and $E_3(z)$ are $(\sigma, \mu_0 - \frac{a+b}{2})$-symmetric and filters $E_1(z)$ and $E_2(z)$ are $(-\sigma, \mu_0 - \frac{a+b}{2})$-symmetric. The related structure is found in Figure 5.6(d).

Step 6: Return to Step 3.

5.7 Examples

In this section three examples are presented to illustrate the symmetry-restoration procedure as described in Section 5.6.2. In Example 5.4 the procedure is used for a symmetric multirate filter with the rational decimation factor $\frac{3}{2}$. Example 5.5 describes the procedure for a decimating filter with $D = 7$. Finally, Example 5.6 describes the procedure for a decimating filter with $D = 2$. In the latter case the quantized filter coefficients are known and the actual costs, i.e., the number of additions, are determined. Also the role of parameters a and b is shown.
In [8] a possible implementation in MATLAB of the symmetry restoration procedure for real filters only is presented. Based on this tool additional examples are elaborated that show the possibilities of the procedure.

Example 5.4. The general method to restore symmetry in a unified polyphase structure is illustrated for the $(\sigma, \frac{7}{2})$-symmetric filter $H(z) \in \mathbb{C}(z)$, with interpolation factor $I = 2$ and decimation factor $D = 3$. The resulting scheme is presented in Figure 5.7(b).

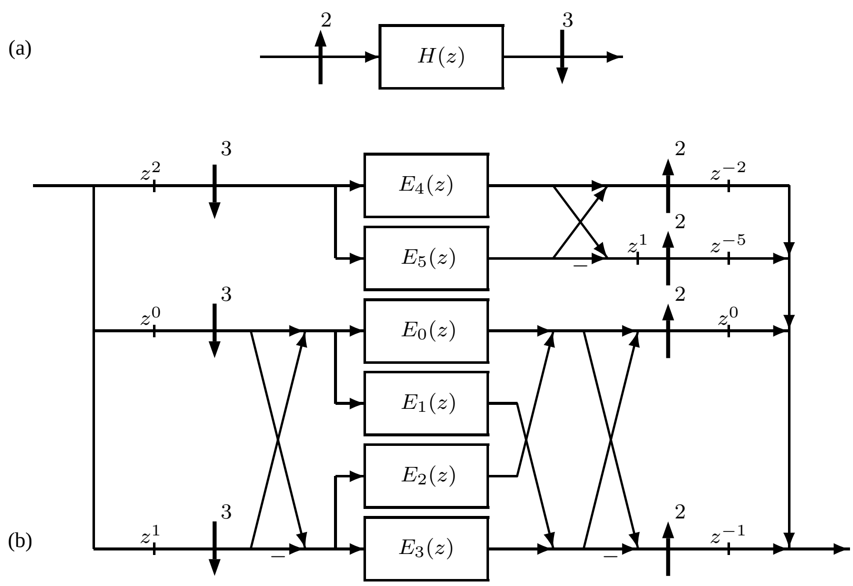


Figure 5.7: *Example of a polyphase structure with restored symmetry, in case of a rational decimation factor ($\frac{3}{2}$), $I = 2$, $D = 3$ and $\mu = \frac{7}{2}$.*

Step 1: By assumption the filter $H(z)$ is symmetric with $\mu = \frac{7}{2}$ and $\gcd(I, D) = \gcd(2, 3) = 1$.

Step 2: $I = 2$, $D = 3$, $R = ID = 6$, the values for p and q are $p = 1$ and $q = -1$.

Step 3: Select $r_0 = 0$.

Step 4: For $\mu_0 = \frac{1}{2}$: $s_0 = 2\mu - 2R\mu_0 - r_0 = 1$.

Step 5: Since $(0|_3 \neq 1|_3) \wedge (0|_2 \neq 1|_2)$ Step 5-Rule d is selected. Now $r_1 = s_0pD + r_0qI = 3$, $s_1 = r_0pD + s_0qI = -2$, $-r_0q = 0$, $-s_0q = 1$, $-r_0p = 0$ and $-s_0p = -1$. The values of a and b can be chosen freely, e.g., $a = b = 0$. So:

$$
\begin{aligned}
E_0(z) &= \tfrac{1}{4}\big(H_{6:0}(z) + H_{6:-2}(z) + H_{6:3}(z) + H_{6:1}(z)\big),\\
E_1(z) &= \tfrac{1}{4}\big(H_{6:0}(z) + H_{6:-2}(z) - H_{6:3}(z) - H_{6:1}(z)\big),\\
E_2(z) &= \tfrac{1}{4}\big(H_{6:0}(z) - H_{6:-2}(z) + H_{6:3}(z) - H_{6:1}(z)\big),\\
E_3(z) &= \tfrac{1}{4}\big(H_{6:0}(z) - H_{6:-2}(z) - H_{6:3}(z) + H_{6:1}(z)\big).
\end{aligned}
$$

Filters $E_0(z)$ and $E_3(z)$ are $(\sigma, \frac{1}{2})$-symmetric and filters $E_1(z)$ and $E_2(z)$ are $(-\sigma, \frac{1}{2})$-symmetric.

Step 6: Until now, only 4 of the 6 paths are considered $\{-2, 0, 1, 3\} \subset \mathcal{R}$, so return to Step 3.

Step 3: Select $r_0 = 2$.

Step 4: For $\mu_0 = 0$: $s = 2\mu - 2R\mu_0 - r = 5$.

Step 5: Since $(2|_3 = 5|_3) \wedge (2|_2 \neq 5|_2)$ Step 5-Rule c is selected. Now $-rq = 2$, $-rp = -2$, $-sp = -5$ and $-x = -\frac{(s-r)q}{D} = 1$. The value of b can be chosen freely, e.g., $b = 0$. So:

$$
\begin{aligned}
E_4(z) &= \tfrac{1}{2}\big(H_{6:2}(z) + H_{6:5}(z)\big),\\
E_5(z) &= \tfrac{1}{2}\big(H_{6:2}(z) - H_{6:5}(z)\big).
\end{aligned}
$$

Filter $E_4(z)$ is $(\sigma, 0)$-symmetric and filter $E_5(z)$ is $(-\sigma, 0)$-symmetric.

Step 6: Now, all 6 paths are considered: $\{-2, 0, 1, 2, 3, 5\} = \mathcal{R}$, so the symmetry-restoration procedure terminates.

Note that by selecting non-zero values for the a and b parameters alternative solutions can be obtained. Adding delays and applying noble identities, the delay before the SRD and after the SRI can be modified or made causal.
End of example

Example 5.5. The general method for restoring the symmetry in a unified polyphase structure is also applicable to basic polyphase structures. For a decimating filter the factor I is set to $I = 1$, and for an interpolating filter the factor D is set to $D = 1$. In this example the $(\sigma, 5)$-symmetric decimating filter $H(z) \in \mathbb{C}(z)$ with decimation factor $D = 7$ is discussed. The resulting structure is shown in Figure 5.8.

Step 1: By assumption the filter $H(z)$ is symmetric with $\mu = 5$ and $\gcd(I, D) = \gcd(1, 7) = 1$.

Step 2: $I = 1$, $D = 7$, $R = ID = 7$, the values for p and q are $p = 0$ and $q = 1$.

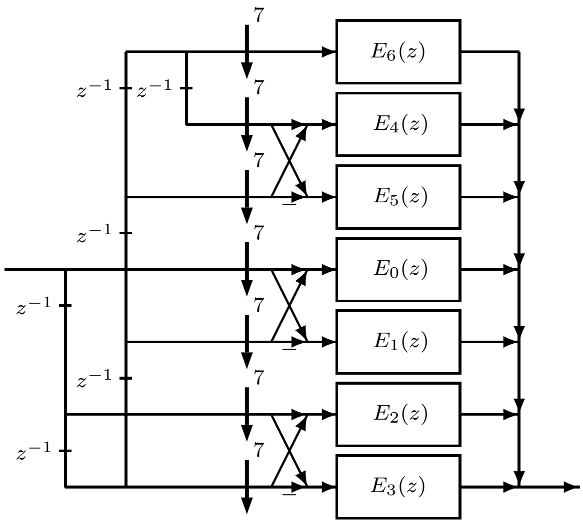


Figure 5.8: *Example of a polyphase structure with restored symmetry, in case of an integer decimation factor $D = 7$ and $\mu = 5$.*

Step 3: Select $r = 0$.

Step 4: For $\mu_0 = \frac{1}{2}$: $s = 2\mu - 2R\mu_0 - r = 3$.

Step 5: Since $(0|_7 \neq 3|_7) \wedge (0|_1 = 3|_1)$ Step 5-Rule b is selected. Now $-rq = 0$, $-rp = 0$, $-sq = -3$ and $-x = -\frac{(s-r)p}{I} = 0$. The value of a can be chosen freely, e.g., $a = 0$. So:

$$E_0(z) = \tfrac{1}{2}\big(H_{7:0}(z) + H_{7:3}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{7:0}(z) - H_{7:3}(z)\big).$$

Filter $E_0(z)$ is $(\sigma, \frac{1}{2})$-symmetric and filter $E_1(z)$ is $(-\sigma, \frac{1}{2})$-symmetric.

Step 6: Until now, only 2 of the 7 paths are considered $\{0, 3\} \subset \mathcal{R}$, so return to Step 3.

Step 3: Select $r = 1$.

Step 4: For $\mu_0 = \frac{1}{2}$: $s = 2\mu - 2R\mu_0 - r = 2$.

Step 5: Since $(1|_7 \neq 2|_7) \wedge (1|_1 = 2|_1)$ Step 5-Rule b is selected. Now $-rq = -1$, $-rp = 0$, $-sq = -2$ and $-x = -\frac{(s-r)p}{I} = 0$. The value of a can be chosen freely, e.g., $a = 0$. So:

$$E_2(z) = \tfrac{1}{2}\big(H_{7:1}(z) + H_{7:2}(z)\big),$$
$$E_3(z) = \tfrac{1}{2}\big(H_{7:1}(z) - H_{7:2}(z)\big).$$

Filter $E_2(z)$ is $(\sigma, \frac{1}{2})$-symmetric and filter $E_3(z)$ is $(-\sigma, \frac{1}{2})$-symmetric.

Step 6: Until now, only 4 of the 7 paths are considered $\{0, 1, 2, 3\} \subset \mathcal{R}$, so return to Step 3.

Step 3: Select $r = 6$.

Step 4: For $\mu_0 = 0$: $s = 2\mu - 2R\mu_0 - r = 4$.

Step 5: Since $(6|_7 \neq 4|_7) \wedge (6|_1 = 4|_1)$ Step 5-Rule b is selected. Now $-rq = -6$, $-rp = 0$, $-sq = -4$ and $-x = -\frac{(s-r)p}{I} = 0$. The value of a can be chosen freely, e.g., $a = 0$. So:

$$E_4(z) = \tfrac{1}{2}\big(H_{7:6}(z) + H_{7:4}(z)\big),$$
$$E_5(z) = \tfrac{1}{2}\big(H_{7:6}(z) - H_{7:4}(z)\big).$$

Filter $E_4(z)$ is $(\sigma, 0)$-symmetric and filter $E_5(z)$ is $(-\sigma, 0)$-symmetric.

Step 6: Until now, only 4 of the 7 paths are considered $\{0, 1, 2, 3, 4, 6\} \subset \mathcal{R}$, so return to Step 3.

Step 3: Select $r = 5$.

Step 4: For $\mu_0 = 0$: $s = 2\mu - 2R\mu_0 - r = 5$.

Step 5: Since $(5|_7 = 5|_7) \wedge (5|_1 = 5|_1)$ Step 5-Rule a is selected. Now $-rq = -5$ and $-rp = 0$. So filter $E_6(z) = H_{7:5}(z)$ is $(\sigma, 0)$-symmetric.

Step 6: Now, all 7 paths are considered: $\{0, 1, 2, 3, 4, 5, 6\} = \mathcal{R}$, so the symmetry-restoration procedure terminates.

In the structure, see Figure 5.8, the delays at the input are combined efficiently in a tapped delay line. Since for all indices r and s the relation $r|_I = s|_I$ holds, either Step 5-Rule a or Step 5-Rule b of the procedure apply. Similarly, in case of an interpolating filter, i.e. $D = 1$, either Step 5-Rule a or Step 5-Rule c of the procedure apply.
End of example

Example 5.6. Also this example illustrates restoration of symmetry in the polyphase structure of decimating filter $H(z)$ now with decimation factor $D = 2$. The focus now is on varying parameter a (and/or b in the general recipe), that can be beneficial to further reduce the number of additions in case the multiplications are implemented using shift-and-add operations, CSD [120] and Appendix E.
Consider the $(1, \frac{11}{2})$-symmetric real filter $H(z) \in \mathbb{Z}/2^{14}(z)$, with a passband gain of 0 dB, a passband ripple of 1 dB and a stopband gain of -80 dB. The passband ranges from $\theta = 0$ through $\theta = 0.3\pi$ and the stopband ranges from $\theta = 0.7\pi$ through $\theta = \pi$. The filter length is $L = 12$. For reasons of clarity a scaled version of the filter $H(z)$ will be used: $H'(z) = 2^{14}H(z) \in \mathbb{Z}(z)$. The 12 coefficients of filter $H'(z)$ are listed in Table 5.1 column 2 and 3 for each of the PPCs, $H'(z) = H'_{2:0}(z^2) + z^{-1}H'_{2:1}(z^2)$. The p_i refer to the multipliers in Figure 5.9.
Direct use of the basic polyphase structure, as in Figure 5.2, results in the scheme of Figure 5.9, where in total 11 additions, 6 delay elements and 12 multiplications are required.

					$a=0$				$a=1$			
	$H'_{2:0}(z)$		$H'_{2:1}(z)$		$2E'_0(z)$		$2E'_1(z)$		$2E'_0(z)$		$2E'_1(z)$	
n	$h'_{2:0}[n]$	#	$h'_{2:1}[n]$	#	$2e'_0[n]$	#	$2e'_1[n]$	#	$2e'_0[n]$	#	$2e'_1[n]$	#
-1									$-544 = q_0$	1	$544 = q_4$	1
0	$-109 = p_0$	3	$-544 = p_5$	1	-653	4	435	4	$-10 = q_1$	1	$-208 = q_5$	2
1	$-935 = p_1$	4	$99 = p_4$	3	-836	3	-1034	2	$5408 = q_2$	3	$-7278 = q_6$	4
2	$3230 = p_2$	4	$6343 = p_3$	5	9573	6	-3113	4	$6460 = q_3$	4	0	
3	$6343 = p_3$	5	$3230 = p_2$	4	9573		3113		5408		7278	
4	$99 = p_4$	3	$-935 = p_1$	4	-836		1034		-10		208	
5	$-544 = p_5$	1	$-109 = p_0$	3	-653		-435		-544		-544	
		20		20		13		10		9		7

Table 5.1: *Filter coefficients and their required number of additions for the schemes in Figure 5.9 and Figure 5.10.*

The equal multiplications in both PPCs cannot be combined. In case the multiplications are implemented using shift-and-add operations, CSD, the 12 multiplications can be replaced by $20+20 = 40$ additions (the column with header $\#$ gives the number of additions and subtractions for the coefficient as CSD), such that in total $11 + 20 + 20 = 51$ additions and 6 delay elements are required. If symmetry could have been exploited (not in this scheme), the number of multiplications would have been 6 or the number of additions would have been $11 + 20 = 31$.

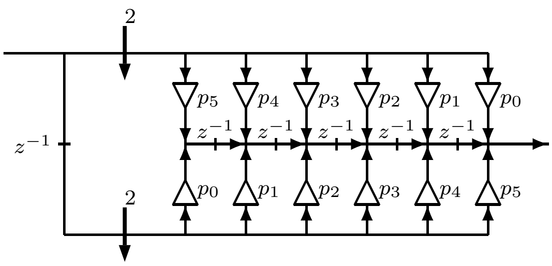


Figure 5.9: *Basic polyphase structure for $H(z)$; $\downarrow 2$, requires 51 additions. The values for p_i and the required number of additions are listed in Table 5.1.*

Next, the symmetry will be restored according to the procedure as described in Section 5.6.2.

Step 1: By assumption the filter $H(z)$ is symmetric with $\sigma = 1$, $\mu = \frac{11}{2}$ and $\gcd(I, D) = \gcd(1, 2) = 1$.

Step 2: $I = 1$, $D = 2$, $R = ID = 2$, the values for p and q are $p = 0$ and $q = 1$.

Step 3: Select $r = 0$.

Step 4: For $\mu_0 = \frac{5}{2}$: $s = 2\mu - 2R\mu_0 - r = 1$.

Step 5: Since $(0|_2 \neq 1|_2) \wedge (0|_1 = 1|_1)$ Step 5-Rule b is selected. Now $-rq = 0$, $-rp = 0$, $-sq = -1$ and $-x = -\frac{(s-r)p}{I} = 0$. Different values of a will be considered later. So:

$$E_0(z) = \tfrac{1}{2}\big(H_{2:0}(z) + z^a H_{2:1}(z)\big),$$
$$E_1(z) = \tfrac{1}{2}\big(H_{2:0}(z) - z^a H_{2:1}(z)\big).$$

Filter $E_0(z)$ is $(1, \frac{5}{2} - \frac{a}{2})$-symmetric and filter $E_1(z)$ is $(-1, \frac{5}{2} - \frac{a}{2})$-symmetric.

Step 6: Now, both paths are considered: $\{0,1\} = \mathcal{R}$, so the symmetry-restoration procedure terminates.

Next, the number of additions required to implement the coefficients of both filters, using CSDs and exploiting symmetry, is evaluated for different values of a. It is found that for $a = 1$ this number is optimal. In Table 5.1 the coefficients of filter $2E_0(z)$ and $2E_1(z)$ are listed for the cases $a = 0$ and $a = 1$. Introduction of the factor 2 is to avoid non-integer coefficients but has no consequences with respect to the number of additions. For $a = 0$, the coefficients require $13+10 = 23$ additions, whereas for $a = 1$, the coefficients require only $9+7 = 16$ additions. Now, in total $12+16+2 = 30$ additions and 8 delay elements are required. The q_i in Table 5.1 refer to the multipliers in Figure 5.10. The equations and Table 5.1 allow filters to be non-causal for $a = 1$. The scheme of Figure 5.10 is made causal by introducing an additional delay at the output. As a consequence the scheme in Figure 5.10 implements $H(z); \downarrow 2; z^{-1}$.

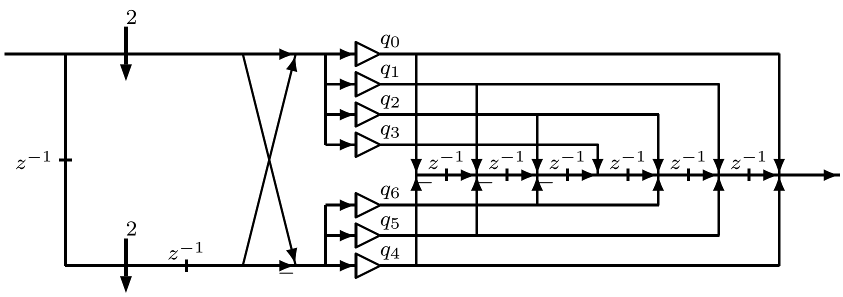


Figure 5.10: *Polyphase structure with restored symmetry, for $H(z); \downarrow 2; z^{-1}$, requires 30 additions. The values for q_i and the required number of additions are listed in Table 5.1.*

End of example

5.8 Structural zeros in PPCs

Real linear-phase (σ, μ)-symmetric filters can have structural zeros depending on the values of σ and μ, whereas complex symmetric filters never have, see Section 2.11. Since the individual real and imaginary parts of a complex filter are real filters, they can have structural zeros.
Application of Theorem 5.1 to restore the symmetry in a unified polyphase structure, can result in many subfilters with structural zeros even when the original filter has none or just a few structural zeros. It depends on the structure in which all the subfilters are connected whether structural zeros can be shared or not.

5.9 Conclusion

Central in this chapter is the restoration of symmetry in polyphase structures of symmetric multirate filters. A new theorem states that the polyphase structure of any real or complex multirate (σ, μ)-symmetric filter, with integer or rational interpolation or decimation factors, can be constructed from symmetric filters only. A unified approach results in a general applicable algorithm to devise the structure that contains symmetric filters only. In addition an extra degree of freedom in restoring the symmetry is identified.
To tool up for the treatment of the new theorem, a few important definitions, identities and known structures, the basic polyphase structures for integer multirate factors, and the nested polyphase structures for rational multirate factors, are treated in detail first. The unified polyphase structure for rational multirate factors serves as basis for the restoration of symmetry. By means of some typical examples it is shown how the new procedure can be used to restore the symmetry and what savings can be achieved.
It is subject to future research to integrate the techniques described in [94] and [95] with the new method described in this chapter, to further reduce the complexity of multirate filters.

Chapter 6

Conclusions

In this final chapter of the thesis, the research questions are answered in Section 6.1, and some suggestions for future work are made in Section 6.2.

6.1 Answers to the research questions

The three main questions posed in Section 1.7 will be answered in this section, by combining results from the various chapters in this thesis. More details can be found in the conclusions of the individual chapters.

- Is it relevant to design generalized-Hermitian-symmetric filters?

Any generalized-Hermitian-symmetric filter with complex coefficients, $H(z) \in \mathbb{C}(z)$, can be obtained from a Hermitian-symmetric filter with complex coefficients, $G(z) \in \mathbb{C}(z)$, in a trivial way, by taking $H(z) = aG(z)$ for a special value of $a \in \mathbb{C}$. Theorem 2.1 describes this so called reduction of symmetric filters over $\mathbb{C}$. Therefore it is not relevant to have a special filter design tool for directly designing the generalized-Hermitian-symmetric filter $H(z) \in \mathbb{C}(z)$.
With respect to the generalized-Hermitian-symmetric filter with complex-integer coefficients, $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, the answer is more surprising. According to Theorem 2.2, any generalized-Hermitian-symmetric filter with complex-integer coefficients, $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, can be obtained from a Hermitian-symmetric filter with scaled complex-integer coefficients, $G(z) \in \mathbb{C}_{\mathbb{Z}/2}(z)$, by taking $H(z) = aG(z)$ for a special value of $a \in \mathbb{C}_{\mathbb{Z}}$. This is called the reduction of symmetric filters over $\mathbb{C}_{\mathbb{Z}}$. As a consequence, it is also not relevant to have a special filter design tool for directly designing the generalized-Hermitian-symmetric filter $H(z) \in \mathbb{C}_{\mathbb{Z}}(z)$. Starting with a Hermitian-symmetric filter $G(z) \in \mathbb{C}_{\mathbb{Z}}(z)$, and a special value for a, viz., $a = 1$, j, $1 + \text{j}$ or $1 - \text{j}$, the resulting generalized-Hermitian-symmetric filter $H(z) = aG(z)$ shows clear forms of symmetry. It was shown in Chapter 3, for first- and second-order filters, that, depending on the specifications, these special generalized-Hermitian-symmetric filters may be beneficial.

- What structures implement generalized-Hermitian-symmetric filters?

The transversal structure is very suited to implement any generalized-Hermitian-symmetric filter. This is elaborated in Chapter 4, where it is also shown how the symmetry can be accomplished with a special scale factor and coefficients that appear in conjugated pairs. A few efficient structures are presented for these conjugated pairs. Alternatively, several structures are shown for the decomposition of FIR filters into individual real and imaginary parts.

- Is it possible to restore the symmetry in polyphase filter structures?

Yes, according to Theorem 5.1, any generalized-Hermitian-symmetric multirate filter with integer or rational interpolation or decimation factors can be constructed from generalized-Hermitian-symmetric filters in a polyphase structure. The several subfilters in a polyphase structure exhibit relations that can be exploited to obtain generalized-Hermitian-symmetric filters according to the methods described in Chapter 2. The structures that may be applied to implement these filters are discussed in Chapter 4.

6.2 Suggestions for future work

The results presented in this thesis inspire a few topics that may be worthwhile considering in future work.
The DESFIL software package can be extended in several ways: i) design a generalized-Hermitian-symmetric filter with complex-integer coefficients, by designing a Hermitian-symmetric filter and an appropriate scale factor, and ii) restore the symmetry in a polyphase structure and select the optimal configuration.
The theorems about the reduction of generalized-Hermitian-symmetric filters over $\mathbb{C}_{\mathbb{Z}}$ can be extended to multi-dimensional filters. Also the restoration of symmetry in polyphase structures may be studied for multi-dimensional filters.
When studying alternatives for integer and complex-integer coefficients (Appendix E), the exhaustive search had to be limited. It would be beneficial to know close bounds for the search.

Appendix A

Some common identities

A number of identities like the noble identities and the prime identities, including their proofs known from e.g. [133] and [135], are summarized in this appendix. In addition, the split-delay identity and some identities related to complex modulators in combination with SRDs and SRIs are presented. Also some properties of the norms that are defined in Section 1.9.6 are listed. Finally, some examples are presented to demonstrate a possible application of the identities.

A.1 Increasing and decreasing the sampling rate

The building blocks for changing the sampling rate, the Sampling Rate Increaser (SRI) and the Sampling Rate Decreaser (SRD), are commonly described in the time domain, for instance [43]. For the SRI, the output signal $y[n] = x[\frac{n}{I}]$ if n is a multiple of I, and $y[n] = 0$ for all other n, where I is called the interpolation factor. For the SRD the output signal $y[n] = x[nD]$, where D is called the decimation factor. Both factors I and D are positive integers, i.e., $I, D \in \mathbb{N}^+$.

A.1.1 Increasing the sampling rate

For the SRI in Figure A.1, the relation between the output and the input in time domain is defined next.

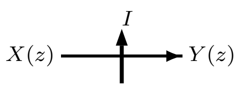


Figure A.1: *Sampling Rate Increaser (SRI).*

Definition A.1. *For a given input signal $x[n]$ and SRI $\uparrow I$ with $I \in \mathbb{N}^+$, the output signal $y[n]$ is defined as:*

$$y[n] \triangleq \begin{cases} x[\frac{n}{I}], & n = kI \text{ with } k \in \mathbb{Z}, \\ 0, & \text{otherwise}. \end{cases}$$

For the z-domain the next lemma describes the behaviour of the SRI.

Lemma A.1. *Let signal $Y(z)$ be the upsampled version of signal $X(z)$, with interpolation factor $I \in \mathbb{N}^+$, then:*

$$Y(z) = X(z^I).$$

Proof. By the SRI the indices of the input samples, $x[n]$, are basically multiplied by the factor I. All other samples become zero. Using the definition of the z-transform gives:

$$\begin{aligned} X(z) &= \sum_n x[n]z^{-n}, \\ Y(z) &= \sum_n y[n]z^{-n} \\ &= \sum_n x[n]z^{-nI} \\ &= \sum_n x[n](z^I)^{-n} \\ &= X(z^I). \end{aligned}$$

□

A.1.2 Decreasing the sampling rate

For the SRD in Figure A.2, the relation between the output and the input in time domain is defined next.

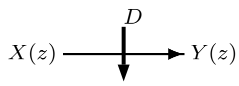


Figure A.2: *Sampling Rate Decreaser (SRD).*

Definition A.2. *For a given input signal $x[n]$ and SRD $\downarrow D$ with $D \in \mathbb{N}^+$, the output signal $y[n]$ is defined as:*

$$y[n] \triangleq x[nD].$$

For the z-domain the next lemma describes the behaviour of the SRD.

Lemma A.2. *Let signal $Y(z)$ be the downsampled version of signal $X(z)$, with decimation factor $D \in \mathbb{N}^+$, then:*

$$Y(z) = \frac{1}{D} \sum_{i=0}^{D-1} X(z^{\frac{1}{D}} W_D^i),$$

where W_D is the D^{th} root of unity, or twiddle factor, defined as $W_D = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{D}}$.

Proof. First an auxiliary signal $v[n]$ is described that has the same sampling frequency as $x[n]$, see Figure A.3. The signal $v[n] = x[n]$ for all n that are multiples of D, for all other n, $v[n] = 0$. This can be expressed using the well known orthogonality property:

$$\frac{1}{D} \sum_{i=0}^{D-1} W_D^{-in} = \begin{cases} 1, & n = kD, \\ 0, & \text{otherwise}, \end{cases}$$

so:

$$v[n] = x[n] \frac{1}{D} \sum_{i=0}^{D-1} W_D^{-in}.$$

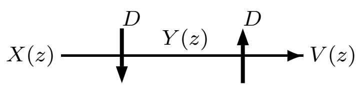


Figure A.3: *Scheme with an SRD and an SRI to support the proof.*

The z-transform of $v[n]$ is:

$$\begin{aligned} V(z) &= \sum_n v[n] z^{-n} \\ &= \frac{1}{D} \sum_n x[n] \sum_{i=0}^{D-1} W_D^{-in} z^{-n} \\ &= \frac{1}{D} \sum_{i=0}^{D-1} \sum_n x[n] W_D^{-in} z^{-n} \\ &= \frac{1}{D} \sum_{i=0}^{D-1} \sum_n x[n] (W_D^i z)^{-n} \\ &= \frac{1}{D} \sum_{i=0}^{D-1} X(z W_D^i). \end{aligned}$$

Using the SRI z-domain description, Lemma A.1, gives $Y(z^D) = V(z)$. Since $v[n] = 0$ for indices not being multiple of D and substituting: $z^D \leftarrow z$ or $z \leftarrow z^{\frac{1}{D}}$, concludes the proof. ☐

A.2 Some identities of SRDs and SRIs

In multirate structures with SRDs or SRIs, it is often possible to move around or manipulate with these elements according to special rules. The noble identities, the prime identities, the split-delay identity and some of the modulator identities are discussed extensively in the following sections of this appendix. Some, rather trivial, identities that follow directly from Definition A.2 are presented in Figure A.4. These examples are all based on SRDs only. Similar examples for SRIs can be obtained by transposing the SRD schemes [41] [135].

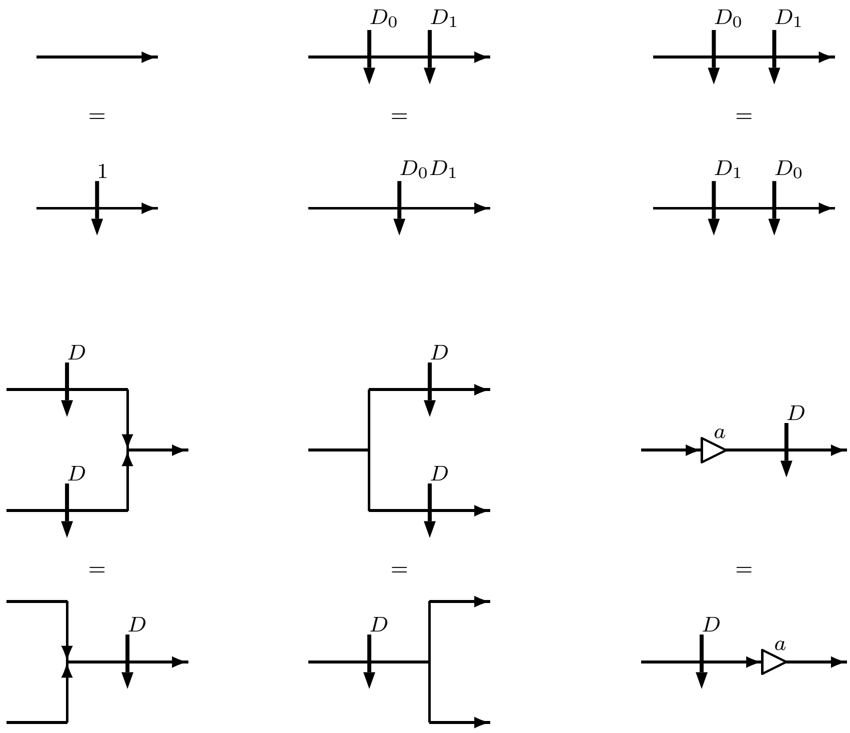


Figure A.4: *Some trivial identities of SRDs.*

The respective equalities are: $1 =\downarrow 1$, $\downarrow D_0; \downarrow D_1 =\downarrow D_0 D_1$, $\downarrow D_0; \downarrow D_1 =\downarrow D_1; \downarrow D_0$, $\left[\downarrow D \;\; \downarrow D\right] = \left[1 \;\; 1\right]; \downarrow D$, $\left[\begin{smallmatrix}\downarrow D\\ \downarrow D\end{smallmatrix}\right] =\downarrow D; \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ and $a; \downarrow D =\downarrow D; a$.

The equalities for the SRIs are: $1 =\uparrow 1$, $\uparrow I_0; \uparrow I_1 =\uparrow I_0 I_1$, $\uparrow I_0; \uparrow I_1 =\uparrow I_1; \uparrow I_0$, $\left[\uparrow I \;\; \uparrow I\right] = \left[1 \;\; 1\right]; \uparrow I$, $\left[\begin{smallmatrix}\uparrow I\\ \uparrow I\end{smallmatrix}\right] =\uparrow I; \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ and $a; \uparrow I =\uparrow I; a$.

A.3 Noble identities

The first and second noble identities describe the combination of a filter and an SRD or SRI respectively and the change of their order.

A.3.1 First noble identity

An SRD with decimation factor D followed by a filter with system function $H(z)$, is identical to a filter $G(z) = H(z^D)$ that is followed by the same SRD, see Figure A.5. In the filter with system function $H(z^D)$ each delay element of the original filter, is replaced by D delay elements. The new filter is also referred to as a comb filter with comb factor D.

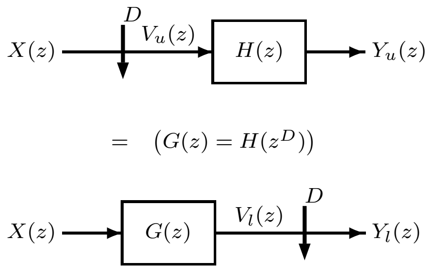


Figure A.5: *First noble identity.*

Lemma A.3. *For any filter $H(z) \in \mathbb{C}(z)$ and decimation factor $D \in \mathbb{N}^+$:*

$$\downarrow D; H(z) = G(z); \downarrow D \Leftrightarrow G(z) = H(z^D).$$

Proof. By Lemma A.2, the upper part of Figure A.5 gives:

$$\begin{aligned} V_u(z) &= \frac{1}{D} \sum_{i=0}^{D-1} X(z^{\frac{1}{D}} W_D^i), \\ Y_u(z) &= H(z) V_u(z) \\ &= \frac{1}{D} \sum_{i=0}^{D-1} H(z) X(z^{\frac{1}{D}} W_D^i), \end{aligned}$$

and the lower part of Figure A.5 gives:

$$\begin{aligned} V_l(z) &= G(z) X(z), \\ Y_l(z) &= \frac{1}{D} \sum_{i=0}^{D-1} V_l(z^{\frac{1}{D}} W_D^i) \\ &= \frac{1}{D} \sum_{i=0}^{D-1} G(z^{\frac{1}{D}} W_D^i) X(z^{\frac{1}{D}} W_D^i). \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical if for all i holds: $G(z^{\frac{1}{D}} W_D^i) = H(z)$, which is true iff $G(z) = H(z^D)$. □

A.3.2 Second noble identity

An SRI with interpolation factor I preceded by a filter with system function $H(z)$, is identical to a filter $G(z) = H(z^I)$ that is preceded by the same SRI, see Figure A.6. In this case each delay element of the original filter $H(z)$ is replaced by I delay elements. The new filter is also referred to as a comb filter with comb factor I.

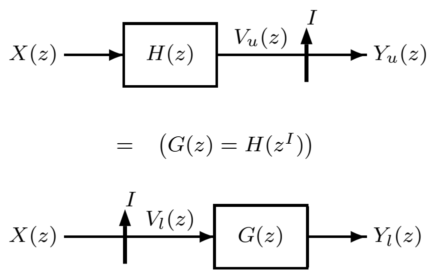


Figure A.6: *Second noble identity.*

Lemma A.4. *For any filter $H(z) \in \mathbb{C}(z)$ and interpolation factor $I \in \mathbb{N}^+$:*

$$H(z); \uparrow I = \uparrow I; G(z) \Leftrightarrow G(z) = H(z^I).$$

Proof. By Lemma A.1, the upper circuit given in Figure A.6 gives:

$$\begin{aligned} V_u(z) &= H(z)X(z), \\ Y_u(z) &= V_u(z^I) \\ &= H(z^I)X(z^I), \end{aligned}$$

and the lower part of Figure A.6 gives:

$$\begin{aligned} V_l(z) &= X(z^I), \\ Y_l(z) &= G(z)V_l(z) \\ &= G(z)X(z^I). \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical iff $G(z) = H(z^I)$. □

A.4 Prime identities

Two prime identities are described next. The first prime identity concerns a combination of an SRD and an SRI, and the second prime identity concerns a filter with an SRD and SRI.

A.4.1 First prime identity

An SRD with decimation factor D and an SRI with interpolation factor I in cascade, may be interchanged without affecting the overall behaviour, if their factors D and I are coprime or $\gcd(I, D) = 1$, see Figure A.7.

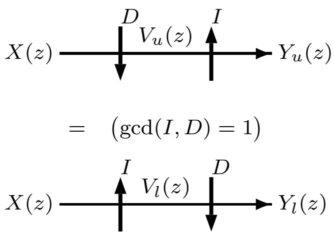


Figure A.7: *First prime identity.*

Lemma A.5. *For any decimation factor $D \in \mathbb{N}^+$ and interpolation factor $I \in \mathbb{N}^+$:*

$$\downarrow D; \uparrow I = \uparrow I; \downarrow D \Leftrightarrow \gcd(I, D) = 1.$$

Proof. The upper circuit of Figure A.7 gives:

$$\begin{aligned}
V_u(z) &= \frac{1}{D} \sum_{i=0}^{D-1} X(z^{\frac{1}{D}} W_D^i),\\
Y_u(z) &= V_t(z^I)\\
&= \frac{1}{D} \sum_{i=0}^{D-1} X(z^{\frac{I}{D}} W_D^i),
\end{aligned}$$

and the lower circuit of Figure A.7 gives:

$$\begin{aligned}
V_l(z) &= X(z^I),\\
Y_l(z) &= \frac{1}{D} \sum_{j=0}^{D-1} V_l(z^{\frac{1}{D}} W_D^j)\\
&= \frac{1}{D} \sum_{j=0}^{D-1} X(z^{\frac{I}{D}} W_D^{jI}).
\end{aligned}$$

The difference between the expressions only occurs in the exponents of the twiddle factors W_D^i and W_D^{jI}. Both expressions are equivalent iff for all $i:\ 0 \leq i < D$, there exists a j such that $i|_D = (jI)|_D$. This is true iff I and D are coprime, as will be shown next. The fact that I and D are coprime implies that there exist $p, q \in \mathbb{Z}$ such that $pD + qI = 1$. Considering the remainder after division by D gives $(qI)|_D = 1|_D$. As a consequence, $(iqI)|_D = i|_D$ or $((iq)|_D I)|_D = i|_D$, for all i. Defining $j = (iq)|_D$ gives $(jI)|_D = i|_D$ for all i.

If I and D have common factor k, define $I = kI'$ and $D = kD'$. Now $i|_D = (jI)|_D$ can be rewritten as $i|_D = k(jI' - mD')$ which implies that $i|_D$ is a multiple of k. Only for $k = 1$ it is possible to let $i|_D$ take any value, which means that I and D have to be coprime. □

A.4.2 Second prime identity

For a decimating or an interpolating filter that changes the sampling rate with a rational factor, the structure of Figure A.8 is used often. If the interpolation and decimation factors are not coprime, a common factor R may be removed from both factors and the filter in between has to be modified. The impulse response of the new filter is the decimated version (decimation factor R) of the original impulse response. In other words the z-transform of the new filter is equal to the $R:0$ polyphase component $H_{R:0}(z)$ of the original filter. Note that for $I = D$ the total system is monorate and time-invariant. A possible side effect of this identity is that the resulting filter may be non-symmetric even if the original filter is symmetric.

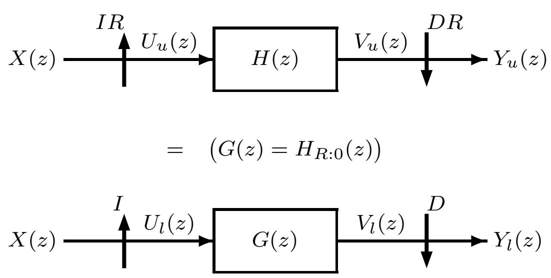


Figure A.8: *Second prime identity.*

Lemma A.6. *For any filter $H(z) \in \mathbb{C}(z)$, decimation factor $D \in \mathbb{N}^+$ and interpolation factor $I \in \mathbb{N}^+$:*

$$\uparrow IR; H(z); \downarrow DR = \uparrow I; G(z); \downarrow D \Leftrightarrow G(z) = H_{R:0}(z).$$

Proof. The upper circuit in Figure A.8 gives:

$$\begin{aligned}
U_u(z) &= X(z^{IR}),\\
V_u(z) &= H(z)U_u(z)\\
&= H(z)X(z^{IR}),\\
Y_u(z) &= \frac{1}{DR}\sum_{i=0}^{DR-1} V_u(z^{\frac{1}{DR}}W_{DR}^{i})\\
&= \frac{1}{DR}\sum_{i=0}^{DR-1} H(z^{\frac{1}{DR}}W_{DR}^{i})X(z^{\frac{I}{D}}W_{DR}^{iIR})\\
&= \frac{1}{DR}\sum_{i=0}^{DR-1} H(z^{\frac{1}{DR}}W_{DR}^{i})X(z^{\frac{I}{D}}W_{D}^{iI}).
\end{aligned}$$

The output signal of the lower circuit in Figure A.8 can be derived by Lemma A.1, Lemma A.2 and Lemma 5.4 and substituting $G(z) \leftarrow H_{R:0}(z)$:

$$\begin{aligned}
U_l(z) &= X(z^{I}),\\
V_l(z) &= G(z)U_l(z)\\
&= H_{R:0}(z)X(z^{I}),\\
Y_l(z) &= \frac{1}{D}\sum_{i=0}^{D-1} V_l(z^{\frac{1}{D}}W_{D}^{i})\\
&= \frac{1}{D}\sum_{i=0}^{D-1} H_{R:0}(z^{\frac{1}{D}}W_{D}^{i})X(z^{\frac{I}{D}}W_{D}^{iI})\\
&= \frac{1}{D}\sum_{i=0}^{D-1} \frac{1}{R}\sum_{j=0}^{R-1} H(z^{\frac{1}{DR}}W_{D}^{\frac{i}{R}}W_{R}^{j})X(z^{\frac{I}{D}}W_{D}^{iI})\\
&= \frac{1}{DR}\sum_{i=0}^{D-1}\sum_{j=0}^{R-1} H(z^{\frac{1}{DR}}W_{DR}^{i}W_{DR}^{jD})X(z^{\frac{I}{D}}W_{D}^{iI}).
\end{aligned}$$

Finally defining $k = i + jD$ gives:

$$Y_b(z) = \frac{1}{DR}\sum_{k=0}^{DR-1} H(z^{\frac{1}{DR}}W_{DR}^{k})X(z^{\frac{I}{D}}W_{D}^{kI}).$$

The equations are identical, so both systems are identical. □

A.5 Split-delay identity

If the interpolation and decimation factors of an SRI and SRD respectively are coprime, a delay between these SRI and SRD can be split, and the resulting parts can be placed outside the SRI and SRD, also the order of the SRI and SRD can be reversed. This is illustrated in Figure A.9. Note that the reversed process, combining both delays into one between the SRI and SRD, is trivial.

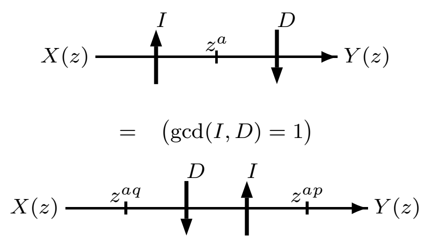


Figure A.9: *Split-delay identity.*

Lemma A.7. *For any interpolation factor $I \in \mathbb{N}^+$, delay z^a, $a \in \mathbb{Z}$, and decimation factor $D \in \mathbb{N}^+$, with $\gcd(I, D) = 1$, holds:*

$$\uparrow I; z^a; \downarrow D = z^{aq}; \downarrow D; \uparrow I; z^{ap},$$

with $pD + qI = 1$ and $p, q \in \mathbb{Z}$.

Proof. The proof consists of the following three steps.

1. The assumption $\gcd(I, D) = 1$ implies that $p, q \in \mathbb{Z}$ exist such that $pD + qI = 1$, see extended Euclid's algorithm Appendix D. Therefore $a = apD + aqI$ and $\uparrow I; z^a; \downarrow D = \uparrow I; z^{aqI}; z^{apD}; \downarrow D$.

2. Application of the first and second noble identities, Lemma A.3 and Lemma A.4 gives $\uparrow I; z^a; \downarrow D = z^{aq}; \uparrow I; \downarrow D; z^{ap}$.

3. Since $\gcd(I, D) = 1$ the first prime identity, Lemma A.5, gives $\uparrow I; z^a; \downarrow D = z^{aq}; \downarrow D; \uparrow I; z^{ap}$. □

A.6 Complex modulation identities

Modulation can be described in the time-domain as the multiplication on sample by sample basis, of a signal $x[n]$ and a carrier $c[n]$. Both the signal and the carrier may be real or complex, $x[n], c[n] \in \mathbb{C}[n]$. The effect of modulation can be described in the frequency domain as shifting the spectrum of the signal to other frequencies. The carrier can be real $c[n] = \cos(\theta_c n + \phi_c)$, or complex $c[n] = \mathrm{e}^{\mathrm{j}(\theta_c n + \phi_c)}$, with relative frequency θ_c and the phase ϕ_c. The fundamental difference between modulation with a real or complex carrier, can best be illustrated by comparing the Fourier transforms of the output signals in both cases. If:

$$x[n] \stackrel{\mathscr{F}}{\longleftrightarrow} X(\mathrm{e}^{\mathrm{j}\theta}),$$

then:

$$c[n]x[n] \stackrel{\mathscr{F}}{\longleftrightarrow} C(\mathrm{e}^{\mathrm{j}\theta}) \circledast X(\mathrm{e}^{\mathrm{j}\theta}),$$

where $\circledast$ is the cyclic convolution. For $c[n] \in \mathbb{R}$ this gives:

$$\cos(\theta_c n + \phi_c)x[n] \stackrel{\mathscr{F}}{\longleftrightarrow} \tfrac{1}{2}\big(\mathrm{e}^{\mathrm{j}\phi_c}X(\mathrm{e}^{\mathrm{j}(\theta-\theta_c)}) + \mathrm{e}^{-\mathrm{j}\phi_c}X(\mathrm{e}^{\mathrm{j}(\theta+\theta_c)})\big),$$

and for $c[n] \in \mathbb{C}$ this gives:

$$\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}x[n] \stackrel{\mathscr{F}}{\longleftrightarrow} \mathrm{e}^{\mathrm{j}\phi_c}X(\mathrm{e}^{\mathrm{j}(\theta-\theta_c)}).$$

In case of the real carrier $c[n] = \cos(\theta_c n+\phi_c)$, the resulting signal spectrum is composed of two shifted versions of the original spectrum. Both shifted versions may interfere depending on the signal bandwidth and carrier frequency. Such interference in general cannot be reversed in subsequent processing steps. In case of the complex carrier $c[n] = \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$, the resulting signal spectrum is only a single shifted versions of the original spectrum, no interference will occur. This last property makes the complex modulator valuable in many signal processing systems.

In this section a z-domain description of the modulator is given and also a number of complex modulation properties are described.

A.6.1 Complex modulation

The z-domain description of a signal that is modulated with the complex carrier $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$, is given in the following lemma.

Lemma A.8. *Let $x[n]$ be the input signal of a modulator with the complex carrier $c[n] = \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$, then the z-transform of the output signal $y[n] = c[n]x[n]$ is:*

$$Y(z) = \mathrm{e}^{\mathrm{j}\phi_c}X(z\mathrm{e}^{-\mathrm{j}\theta_c}),$$

where $X(z) \in \mathbb{C}(z)$ is the z-transform of $x[n]$.

Proof. Straightforward application of the z-transform gives:

$$\begin{aligned} Y(z) &= \sum_n x[n]c[n]z^{-n} \\ &= \sum_n x[n]\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}z^{-n} \\ &= \mathrm{e}^{\mathrm{j}\phi_c}\sum_n x[n]\mathrm{e}^{\mathrm{j}\theta_c n}z^{-n} \\ &= \mathrm{e}^{\mathrm{j}\phi_c}\sum_n x[n](z\mathrm{e}^{-\mathrm{j}\theta_c})^{-n} \\ &= \mathrm{e}^{\mathrm{j}\phi_c}X(z\mathrm{e}^{-\mathrm{j}\theta_c}). \end{aligned}$$ □

The substitution $z \leftarrow z\mathrm{e}^{-\mathrm{j}\theta_c}$, can be interpreted as a counter clockwise rotation in the z-plane. The symbol from Figure A.10 will be used to depict the modulation functionality in schemes. In equations the following notation will be used:

$$Y(z) = X(z); \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}.$$

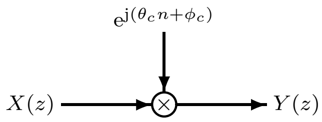


Figure A.10: *Complex modulator with carrier* $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$.

A.6.2 Swapping a complex modulator and a filter

The overall behaviour of a cascade connection of a modulator and a filter is invariant under swapping the modulator and the filter, if the filter is replaced by a modulated version, see Figure A.11.

Lemma A.9. *For any filter* $H(z) \in \mathbb{C}(z)$ *and complex carrier* $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$*:*

$$\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}; H(z) = G(z); \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)} \Leftrightarrow G(z) = H(z\mathrm{e}^{\mathrm{j}\theta_c}).$$

Proof. The upper part of Figure A.11 gives:

$$\begin{aligned} V_u(z) &= \mathrm{e}^{\mathrm{j}\phi_c}X(z\mathrm{e}^{-\mathrm{j}\theta_c}), \\ Y_u(z) &= H(z)V_u(z) \\ &= \mathrm{e}^{\mathrm{j}\phi_c}H(z)X(z\mathrm{e}^{-\mathrm{j}\theta_c}), \end{aligned}$$

and the lower part of Figure A.11 gives:

$$\begin{aligned} V_l(z) &= G(z)X(z), \\ Y_l(z) &= \mathrm{e}^{\mathrm{j}\phi_c} V_l(z\mathrm{e}^{-\mathrm{j}\theta_c}) \\ &= \mathrm{e}^{\mathrm{j}\phi_c} G(z\mathrm{e}^{-\mathrm{j}\theta_c}) X(z\mathrm{e}^{-\mathrm{j}\theta_c}). \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical iff $H(z) = G(z\mathrm{e}^{-\mathrm{j}\theta_c})$, or equivalently $G(z) = H(z\mathrm{e}^{\mathrm{j}\theta_c})$. □

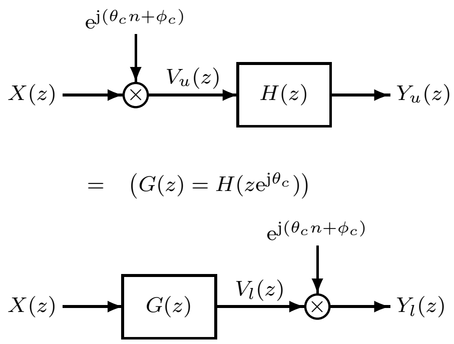


Figure A.11: *Swapping a complex modulator and a filter.*

A.6.3 Swapping a complex modulator and a delay

The overall behaviour of a cascade connection of a modulator and a delay is invariant under swapping the modulator and the delay, if the phase of the modulator is modified properly, see Figure A.12. Swapping a modulator and a delay is just a special case of swapping a modulator and a filter, Lemma A.9 and therefore the corresponding relations can be used too.

Lemma A.10. *For any delay z^{-N} with $N \in \mathbb{Z}$ and complex carrier $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$:*

$$\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}; z^{-N} = z^{-N}; \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c')} \Leftrightarrow \phi_c' = \phi_c - \theta_c N.$$

Proof. The upper part of Figure A.12 gives:

$$\begin{aligned} V_u(z) &= \mathrm{e}^{\mathrm{j}\phi_c} X(z\mathrm{e}^{-\mathrm{j}\theta_c}), \\ Y_u(z) &= z^{-N} V_u(z) \\ &= z^{-N} \mathrm{e}^{\mathrm{j}\phi_c} X(z\mathrm{e}^{-\mathrm{j}\theta_c}), \end{aligned}$$

and the lower part of Figure A.12 gives:

$$\begin{aligned} V_l(z) &= z^{-N}X(z), \\ Y_l(z) &= \mathrm{e}^{\mathrm{j}\phi'_c}V_l(z\mathrm{e}^{-\mathrm{j}\theta_c}) \\ &= \mathrm{e}^{\mathrm{j}\phi'_c}(z\mathrm{e}^{-\mathrm{j}\theta_c})^{-N}X(z\mathrm{e}^{-\mathrm{j}\theta_c}) \\ &= z^{-N}\mathrm{e}^{\mathrm{j}(\phi'_c+\theta_c N)}X(z\mathrm{e}^{-\mathrm{j}\theta_c}). \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical iff $\phi_c = \phi'_c + \theta_c N$, or equivalently $\phi'_c = \phi_c - \theta_c N$. □

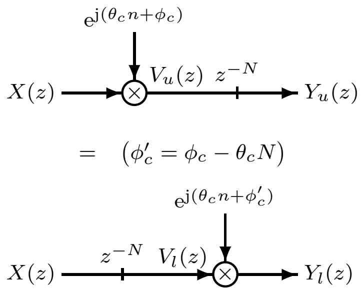


Figure A.12: *Swapping a complex modulator and a delay.*

A.6.4 Swapping a complex modulator and an SRD

If a modulator is moved from a position behind an SRD to a position before the SRD, see Figure A.13, the relative frequency of the modulator has to be adapted. The phase does not change. Due to decimation with a factor D, D shifted versions of the spectrum are added, see Lemma A.2. This allows the modulator in front of the SRD to have any of the D different frequencies, viz. $\theta'_c = \frac{\theta_c}{D} + d\frac{2\pi}{D}$, to give identical results.

Lemma A.11. *For any decimation factor D with $D \in \mathbb{N}^+$ and complex carrier $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$:*

$$\downarrow D; \mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)} = \mathrm{e}^{\mathrm{j}(\theta'_c n+\phi_c)}; \downarrow D \Leftrightarrow \theta'_c = \frac{\theta_c}{D} + d\frac{2\pi}{D},$$

with $d \in \mathbb{Z}$.

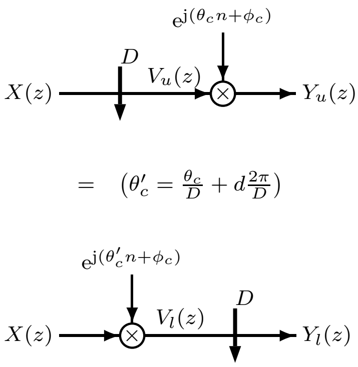


Figure A.13: *Swapping a complex modulator and an SRD.*

Proof. The upper part of Figure A.13 gives:

$$\begin{aligned} V_u(z) &= \frac{1}{D}\sum_{d=0}^{D-1} X(z^{\frac{1}{D}} W_D^d), \\ Y_u(z) &= e^{j\phi_c} V_u(z e^{-j\theta_c}) \\ &= e^{j\phi_c} V_u(z e^{-j(\theta_c + d2\pi)}) \\ &= \frac{1}{D} e^{j\phi_c} \sum_{d=0}^{D-1} X(z^{\frac{1}{D}} e^{-j(\frac{\theta_c}{D} + d\frac{2\pi}{D})} W_D^d). \end{aligned}$$

and the lower part of Figure A.13 gives:

$$\begin{aligned} V_l(z) &= e^{j\phi_c} X(z e^{-j\theta'_c}), \\ Y_l(z) &= \frac{1}{D}\sum_{d=0}^{D-1} V_l(z^{\frac{1}{D}} W_D^d) \\ &= \frac{1}{D} e^{j\phi_c} \sum_{d=0}^{D-1} X(z^{\frac{1}{D}} e^{-j\theta'_c} W_D^d), \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical iff $\theta'_c = \frac{\theta_c}{D} + d\frac{2\pi}{D}$. Since the spectra are 2π-periodic all integer values for d are allowed: $d \in \mathbb{Z}$. □

A.6.5 Swapping a complex modulator and an SRI

If a modulator is moved from a position before an SRI to a position behind the SRI, see Figure A.14, the relative frequency of the modulator has to be adapted. Due to interpolation with a factor I, the SRI introduces a periodic spectrum. This allows the modulator behind the SRI to have any of the I different frequencies, viz. $\theta_c' = \frac{\theta_c}{I} + i\frac{2\pi}{I}$, to give the same results.

Lemma A.12. *For any interpolation factor I with $I \in \mathbb{N}^+$ and complex carrier $\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}$:*

$$\mathrm{e}^{\mathrm{j}(\theta_c n+\phi_c)}; \uparrow I = \uparrow I; \mathrm{e}^{\mathrm{j}(\theta_c' n+\phi_c)} \Leftrightarrow \theta_c' = \frac{\theta_c}{I} + i\frac{2\pi}{I},$$

with $i \in \mathbb{Z}$.

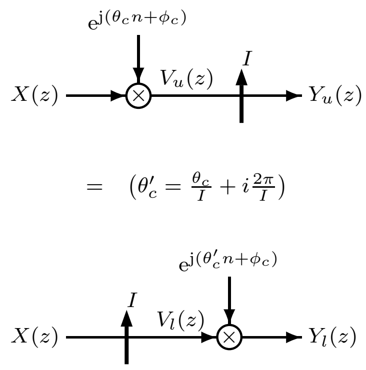


Figure A.14: *Swapping a complex modulator and an SRI.*

Proof. The upper part of Figure A.14 gives:

$$\begin{aligned} V_u(z) &= \mathrm{e}^{\mathrm{j}\phi_c} X(z\mathrm{e}^{-\mathrm{j}\theta_c}), \\ Y_u(z) &= V_u(z^I) \\ &= \mathrm{e}^{\mathrm{j}\phi_c} X(z^I \mathrm{e}^{-\mathrm{j}\theta_c}) \\ &= \mathrm{e}^{\mathrm{j}\phi_c} X(z^I \mathrm{e}^{-\mathrm{j}(\theta_c + i2\pi)}). \end{aligned}$$

and the lower part of Figure A.14 gives:

$$\begin{aligned} V_l(z) &= X(z^I), \\ Y_l(z) &= \mathrm{e}^{\mathrm{j}\phi_c} V_l(z\mathrm{e}^{-\mathrm{j}\theta_c'}) \\ &= \mathrm{e}^{\mathrm{j}\phi_c} X(z^I \mathrm{e}^{-\mathrm{j}\theta_c' I}), \end{aligned}$$

The expressions for $Y_u(z)$ and $Y_l(z)$ are identical iff $\theta_c' I = \theta_c + i2\pi$, or equivalently $\theta_c' = \frac{\theta_c}{I} + i\frac{2\pi}{I}$. Since the spectra are 2π-periodic all integer values for i are allowed: $i \in \mathbb{Z}$. $\square$

A.7 Norms

Based on Definition 1.10 about norms, a number of inequalities is discussed in this section. The sum or difference of values with a limited ∞-norm have a limited ∞-norm too.

Lemma A.13. *Let $a, b \in \mathbb{C}$, then:*

$$\|a \pm b\|_\infty \leq \|a\|_\infty + \|b\|_\infty.$$

Proof. This result follows directly from Definition 1.10, for the p-norm. □

From Definition 1.10 it follows that $\|aH(z)\|_p = \|a\|_1 \|H(z)\|_p$ for $a \in \mathbb{R}$. In case of $a \in \mathbb{C}$ the following inequalities hold for $p = 1$ and $p = \infty$.

Lemma A.14. *Let $a \in \mathbb{C}$ and $H(z) \in \mathbb{C}(z)$, then:*

$$\|aH(z)\|_1 \leq \|a\|_1 \|H(z)\|_1.$$

Proof. From the norm definition and the triangle inequality it follows:

$$\begin{aligned}
\|aH(z)\|_1 &= \|\left(a_\mathrm{r} H_\mathrm{r}(z) - a_\mathrm{i} H_\mathrm{i}(z)\right) + \mathrm{j}\left(a_\mathrm{i} H_\mathrm{r}(z) + a_\mathrm{r} H_\mathrm{i}(z)\right)\|_1 \\
&= \|a_\mathrm{r} H_\mathrm{r}(z) - a_\mathrm{i} H_\mathrm{i}(z)\|_1 + \|a_\mathrm{i} H_\mathrm{r}(z) + a_\mathrm{r} H_\mathrm{i}(z)\|_1 \\
&\leq \|a_\mathrm{r} H_\mathrm{r}(z)\|_1 + \|a_\mathrm{i} H_\mathrm{i}(z)\|_1 + \|a_\mathrm{i} H_\mathrm{r}(z)\|_1 + \|a_\mathrm{r} H_\mathrm{i}(z)\|_1 \\
&\leq |a_\mathrm{r}|\|H_\mathrm{r}(z)\|_1 + |a_\mathrm{i}|\|H_\mathrm{i}(z)\|_1 + |a_\mathrm{i}|\|H_\mathrm{r}(z)\|_1 + |a_\mathrm{r}|\|H_\mathrm{i}(z)\|_1 \\
&\leq \left(|a_\mathrm{r}| + |a_\mathrm{i}|\right)\left(\|H_\mathrm{r}(z)\|_1 + \|H_\mathrm{i}(z)\|_1\right) \\
&\leq \|a\|_1 \|H(z)\|_1.
\end{aligned}$$

□

Lemma A.15. *Let $a \in \mathbb{C}$ and $H(z) \in \mathbb{C}(z)$, then:*

$$\|aH(z)\|_\infty \leq \|a\|_1 \|H(z)\|_\infty.$$

Proof. From the norm definition and the triangle inequality it follows:

$$\begin{aligned}
\|aH(z)\|_\infty &= \|\left(a_\mathrm{r} H_\mathrm{r}(z) - a_\mathrm{i} H_\mathrm{i}(z)\right) + \mathrm{j}\left(a_\mathrm{i} H_\mathrm{r}(z) + a_\mathrm{r} H_\mathrm{i}(z)\right)\|_\infty \\
&= \max\left(\|a_\mathrm{r} H_\mathrm{r}(z) - a_\mathrm{i} H_\mathrm{i}(z)\|_\infty, \|a_\mathrm{i} H_\mathrm{r}(z) + a_\mathrm{r} H_\mathrm{i}(z)\|_\infty\right) \\
&\leq \max\left(\|a_\mathrm{r} H_\mathrm{r}(z)\|_\infty + \|a_\mathrm{i} H_\mathrm{i}(z)\|_\infty, \|a_\mathrm{i} H_\mathrm{r}(z)\|_\infty + \|a_\mathrm{r} H_\mathrm{i}(z)\|_\infty\right) \\
&\leq \max\left(|a_\mathrm{r}|\|H_\mathrm{r}(z)\|_\infty + |a_\mathrm{i}|\|H_\mathrm{i}(z)\|_\infty, |a_\mathrm{i}|\|H_\mathrm{r}(z)\|_\infty + |a_\mathrm{r}|\|H_\mathrm{i}(z)\|_\infty\right) \\
&\leq \left(|a_\mathrm{r}| + |a_\mathrm{i}|\right)\max\left(\|H_\mathrm{r}(z)\|_\infty, \|H_\mathrm{i}(z)\|_\infty\right) \\
&\leq \|a\|_1 \|H(z)\|_\infty.
\end{aligned}$$

□

For the special case $a \in \mathbb{C}_\mathbb{Z}$ the following lemma is applicable.

Lemma A.16. $a \in \mathbb{C}_{\mathbb{Z}} \wedge a \neq 0 \Rightarrow \left\|\frac{1}{a}\right\|_1 \leq 1$.

Proof. Rewriting part of the lemma gives:

$$\left\|\frac{1}{a}\right\|_1 = \left\|\frac{a^*}{|a|^2}\right\|_1 = \frac{\|a\|_1}{|a|^2}.$$

To proof now is $\|a\|_1 \leq |a|^2$ or $|a_\mathrm{r}|+|a_\mathrm{i}| \leq a_\mathrm{r}^2+a_\mathrm{i}^2$, for $a \in \mathbb{C}_{\mathbb{Z}} \wedge a \neq 0$. The implication $|x| \geq 1 \Rightarrow x^2 \geq 1$, for $x \in \mathbb{R}$, proofs the lemma for $|a_\mathrm{r}| \geq 1 \wedge |a_\mathrm{i}| \geq 1$. Remains to proof the case $|a_\mathrm{r}| < 1 \vee |a_\mathrm{i}| < 1$. This latter case implies $a_\mathrm{r} = 0 \wedge |a_\mathrm{i}| \geq 1$ or $|a_\mathrm{r}| \geq 1 \wedge a_\mathrm{i} = 0$ which concludes the proof. □

Next, the maximum output values of complex filters with complex input signals are derived, using the maximum output values of real filters with real input signals. From [63] it is known that the maximum value of the output signal for a given real input signal is limited by the 1-norm of the real filter.

Lemma A.17. *For filter $H(z) \in \mathbb{R}(z)$ and input signal $X(z) \in \mathbb{R}(z)$:*

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

Proof. Let $Y(z) = X(z)H(z)$ and its elements be expressed by means of the convolution sum. For all n:

$$y[n] = \sum_i x[i]h[n-i],$$
$$|y[n]| \leq \sum_i |x[i]||h[n-i]|.$$

Let $|x[i]| = \|X(z)\|_\infty$ for all i, then:

$$\|Y(z)\|_\infty \leq \|X(z)\|_\infty \sum_i |h[n-i]|,$$
$$\|Y(z)\|_\infty \leq \|X(z)\|_\infty \|H(z)\|_1.$$

It can be checked that for $x[n] = c\,\mathrm{sign}(h[-n])$ with $c \in \mathbb{R}$, the bound is reached. □

The same inequality is found in case both the filter and the input signal are complex as is shown in the next lemma.

Lemma A.18. *For filter $H(z) \in \mathbb{C}(z)$ and input signal $X(z) \in \mathbb{C}(z)$:*

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

Proof. For a complex system, by Lemma A.17:

$$\begin{aligned}
Y(z) &= X(z)H(z),\\
Y_\mathrm{r}(z) &= X_\mathrm{r}(z)H_\mathrm{r}(z) - X_\mathrm{i}(z)H_\mathrm{i}(z),\\
Y_\mathrm{i}(z) &= X_\mathrm{i}(z)H_\mathrm{r}(z) + X_\mathrm{r}(z)H_\mathrm{i}(z),\\
\|Y_\mathrm{r}(z)\|_\infty &\leq \|X_\mathrm{r}(z)\|_\infty \|H_\mathrm{r}(z)\|_1 + \|X_\mathrm{i}(z)\|_\infty \|H_\mathrm{i}(z)\|_1,\\
\|Y_\mathrm{i}(z)\|_\infty &\leq \|X_\mathrm{i}(z)\|_\infty \|H_\mathrm{r}(z)\|_1 + \|X_\mathrm{r}(z)\|_\infty \|H_\mathrm{i}(z)\|_1.
\end{aligned}$$

Since $(\|X_\mathrm{r}(z)\|_\infty = \|X_\mathrm{i}(z)\|_\infty) \Rightarrow (\|Y_\mathrm{r}(z)\|_\infty = \|Y_\mathrm{i}(z)\|_\infty)$ the proof is complete. □

From this lemma it follows that if for a given ∞-norm of the input signal the ∞-norm of the output signal should be minimal, the 1-norm of the filter should be minimal. It can be checked that for $x[n] = c(\mathrm{sign}(h_\mathrm{r}[-n]) - \mathrm{j}\,\mathrm{sign}(h_\mathrm{i}[-n]))$ with $c \in \mathbb{R}$, the bound is reached. Also for the input signal $\mathrm{j}x[n]$ the bound is reached.

A.8 Examples

In the following examples some of the identities that have been presented in the previous sections, will be used to derive alternative structures.

Example A.1. If a decimating filter with a non-prime decimation factor has to be realized, it is often favorable to design a cascade of multiple decimating filters each with a smaller decimation factor. In order to analyze the overall behaviour of such a cascade, it may be necessary to determine the equivalent filter that in fact is realized. The top of Figure A.15 shows two decimating filters in cascade. If the first noble identity is applied, as shown in Figure A.5, to the SRD with decimation factor D_0 and to filter $H_1(z)$, the circuit at the bottom of Figure A.15 results.

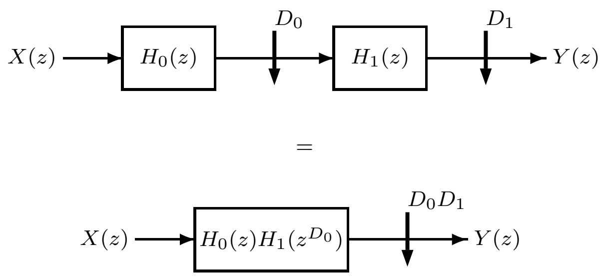


Figure A.15: *First noble identity applied to a cascade of two decimating filters.*

From this example it is clear that for any cascade of decimating filters, the equivalent filter and the decimation factor can be derived by repeatedly applying the first noble identity. Basically, the same procedure is applicable for a cascade of interpolating filters. There the second noble identity has to be used.
End of example

Example A.2. In digital receivers, for instance [59], the structure as is given in Figure A.16 is often used. Behind the decimating filter $H_0(z)$ with decimation factor D, a complex modulator shifts the spectrum in one direction. At the end, the filter $H_1(z)$ takes care for extra selectivity. To study the overall behaviour of this scheme, two alternative structures can be used, one with the modulator at the output (middle scheme) and one with the modulator at the input.
End of example

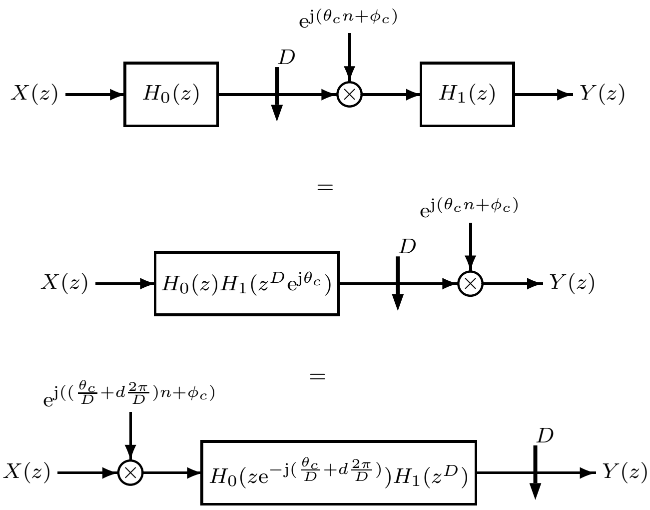


Figure A.16: *Swapping a complex modulator with a filter and an SRD in a receiver-like structure.*

Example A.3. Consider the top circuit in Figure A.17 with a cascade of a decimating and an interpolating filter. If $\gcd(D, I) = 1$, the first prime identity of Figure A.7 and both noble identities can be applied, so resulting in the circuit at the bottom of Figure A.17.

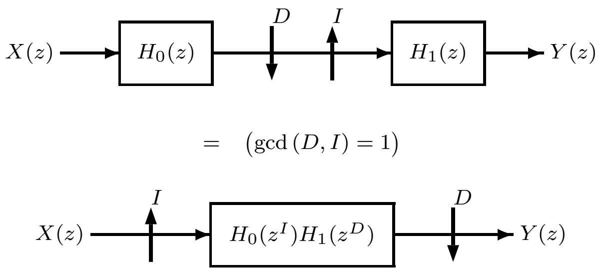


Figure A.17: *First prime identity and both noble identities applied to a cascade of a decimating and an interpolating filter.*

End of example

Appendix B

Introduction to pipelining

In this appendix only a brief introduction to the concept of pipelining is given, to explain its relevance in case of high speed hardware implementations of signal processing algorithms. In for instance [90], more details can be found.
In a discrete-time signal processing system, the signal samples basically move from delay element to delay element, in the pace of the sampling rate. In between two delay elements the sample values may change due to operations like additions and multiplications, see Figure B.1(a). In hardware terminology this is reformulated as: binary values stored in registers pass through a combinatorial network towards the next register, see Figure B.1(b).

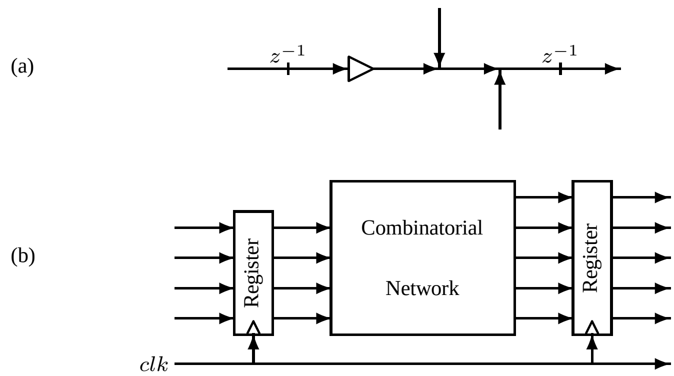


Figure B.1: *Signal processing and hardware structures.*

Because the system clock, clk, may only trigger a register to copy the input values when all these values are stable, the maximum value of the system clock frequency, f_{clk}, is limited by the delay, t_i, of the combinatorial network in between two successive registers.

For a complete system, the system clock frequency, f_{clk}, is limited by the worst case delay, $\max(t_i)$, of all combinatorial networks:

$$f_{clk} < \frac{1}{\max(t_i)}.$$

From this it is clear that one way to increase the system clock frequency, and hence the sampling rate, is to diminish the worst case delay of the combinatorial networks. If somehow a combinatorial network can be partitioned and a register can be interposed, the system clock frequency may be increased, or the speed requirements of the combinatorial networks may be relaxed.
In Figure B.2(a), the transversal structure with a *tapped delay line*, see Figure 4.1(a), is extended with one extra delay element or register, at the input and one at the output. This is to make explicit that data is received from and transferred to other system parts. These delay elements do not belong to the filter itself. From this figure it is clear that the critical path, as indicated with the thick arrow, is formed by the multiplier with coefficient a and 4 successive adders.

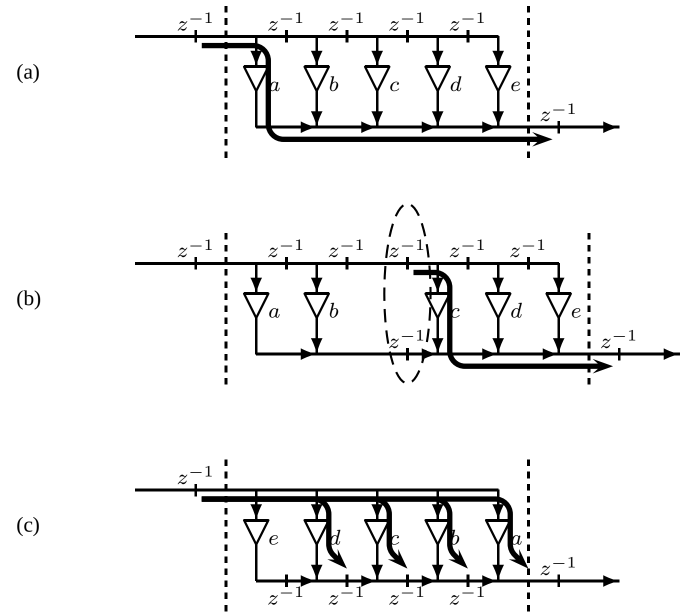


Figure B.2: *Transversal structure: examples of the critical paths and pipelining.*

The system from Figure B.2(a) can be partitioned such that the new critical path is formed by the multiplier with coefficient c and only 2 successive adders, as indicated with the thick arrow. This is at the cost of two additional delay elements, see inside the ellipse in

Figure B.2(b). By interposing the delay elements also the functionality has changed. The newly obtained filter $h'[n]$ or $H'(z)$ is:

$$h'[n] = \langle 0, a, b, c, d, e \rangle = \langle 0, h[n] \rangle,$$
$$H'(z) = z^{-1}H(z).$$

It strongly depends on the application of the filter whether an additional delay matters. If this filter is part of a control loop such a delay changes the overall system behaviour. In case this filter is part of a system with parallel branches, the delays in these branches have to be increased too, which may be a costly affair.
For all these reasons the structure from Figure B.2(c), the transversal filter with the *adding delay line* is preferred, see Figure 4.1(b). Without additional delay elements and without increased system delay, the critical paths all consist of a single multiplier and a single adder only, as indicated with the thick arrows. This is what is called *free pipelining*.
In this example, the concept of pipelining is applied on the level of multiplications and additions. The same concept can in general be applied on many different levels in a system. Basically, pipelining is a method to obtain a high degree of parallelism or concurrency in a system.

Appendix C

Introduction to analog polyphase filters

It is perhaps a bit strange to have an introduction to a special type of analog filters in a document that is focused on digital filters. There are two reasons for doing so. First of all it is the fact that the analog, or continuous time, polyphase filters realize non-symmetric frequency responses that relate to complex impulse responses, and it is of course interesting to see how such responses can be realized in the analog domain. The second reason is the name itself. Scanning the literature for polyphase filters, while implicitly assuming to search for the efficient structure to realize multirate digital filters, gives many hits on analog polyphase filters.

In this appendix only a brief introduction is given, just to show how in principle analog filters with different responses for positive and negative frequencies, can be realized. More information about this type of filters, that were introduced by Gingell in 1973 [52], can be found in for instance [50] and [129].

C.1 Example

To demonstrate the behaviour of an analog polyphase filter, a simple example will be analyzed. In Figure C.1 the electrical scheme is given of a 1-section 4-phase network that basically consists of resistors R [1] and capacitors C only. The buffers at the input and output are not fundamental for the filtering operation.

The relations between the input voltages: $X_{\mathrm{r},+}$, $X_{\mathrm{i},+}$, $X_{\mathrm{r},-}$, $X_{\mathrm{i},-}$, and the output volt-

[1] Traditionally, a resistor is denoted as R and a capacitor is denoted as C and have no relation with the decomposition factor and costs respectively, that are used in the major part of this thesis.

ages: $Y_{\mathrm{r},+}$, $Y_{\mathrm{i},+}$, $Y_{\mathrm{r},-}$, $Y_{\mathrm{i},-}$ are:

$$\begin{aligned}
Y_{\mathrm{r},+}(p) &= X_{\mathrm{r},+}(p) + \big(X_{\mathrm{i},-}(p) - X_{\mathrm{r},+}(p)\big)T(p) \\
&= X_{\mathrm{r},+}(p)\big(1 - T(p)\big) + X_{\mathrm{i},-}(p)T(p), \\
Y_{\mathrm{i},+}(p) &= X_{\mathrm{i},+}(p)\big(1 - T(p)\big) + X_{\mathrm{r},+}(p)T(p), \\
Y_{\mathrm{r},-}(p) &= X_{\mathrm{r},-}(p)\big(1 - T(p)\big) + X_{\mathrm{i},+}(p)T(p), \\
Y_{\mathrm{i},-}(p) &= X_{\mathrm{i},-}(p)\big(1 - T(p)\big) + X_{\mathrm{r},-}(p)T(p),
\end{aligned}$$

where p is the complex frequency [2] $p = \sigma + \mathrm{j}\omega$ and $T(p)$ is:

$$\begin{aligned}
T(p) &= \frac{R}{R + \frac{1}{pC}} \\
&= \frac{p\tau}{1 + p\tau} \quad \text{with} \quad \tau = RC.
\end{aligned}$$

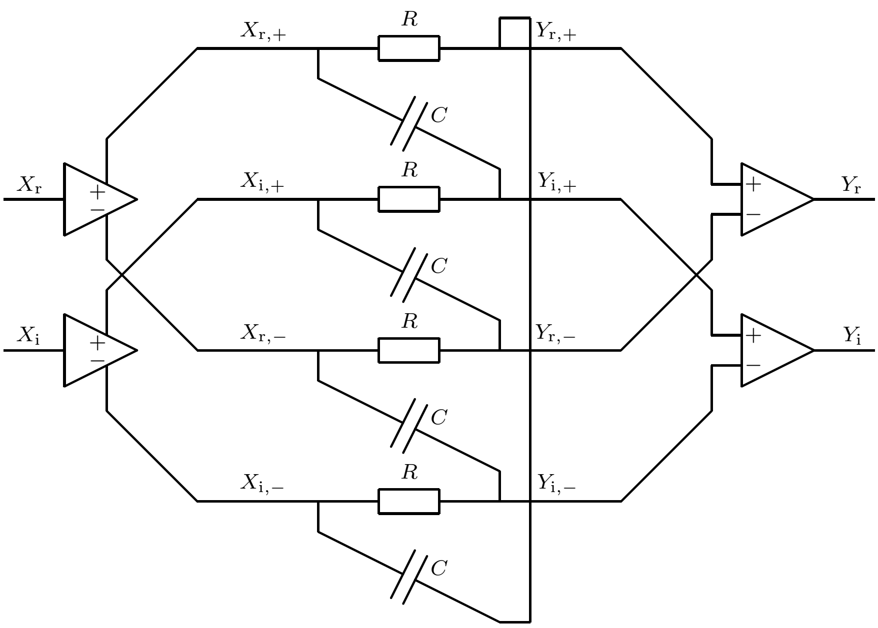


Figure C.1: *Example of a 1-section 4-phase RC polyphase network.*

[2]Traditionally, the real part of the complex frequency p is denoted as σ and has no relation with the shape of symmetry σ that is used in the major part of this thesis.

The two input buffers give: $X_{\mathrm{r},+}(p) = \frac{1}{2}X_{\mathrm{r}}(p)$ and $X_{\mathrm{r},-}(p) = -\frac{1}{2}X_{\mathrm{r}}(p)$, also:

$$X_{\mathrm{r}}(p) = X_{\mathrm{r},+}(p) - X_{\mathrm{r},-}(p).$$

Similar definitions for $X_{\mathrm{i}}(p)$, $Y_{\mathrm{r}}(p)$ and $Y_{\mathrm{i}}(p)$ result in:

$$\begin{aligned} Y_{\mathrm{r}}(p) &= X_{\mathrm{r}}(p)\big(1 - T(p)\big) - X_{\mathrm{i}}(p)T(p), \\ Y_{\mathrm{i}}(p) &= X_{\mathrm{i}}(p)\big(1 - T(p)\big) + X_{\mathrm{r}}(p)T(p). \end{aligned}$$

A definition of the complex input and output signals as:

$$X(p) = X_{\mathrm{r}}(p) + \mathrm{j}X_{\mathrm{i}}(p),$$

together with a similar definitions for $Y(p)$, results in:

$$Y(p) = X(p)\big(1 + (\mathrm{j} - 1)T(p)\big),$$

or:

$$\begin{aligned} H(p) &= \frac{Y(p)}{X(p)} \\ &= 1 + (\mathrm{j} - 1)T(p) \\ &= \frac{1 + \mathrm{j}p\tau}{1 + p\tau}. \end{aligned}$$

The zero and pole are located at $p = \mathrm{j}\frac{1}{\tau}$ and $p = -\frac{1}{\tau}$ respectively. The frequency response of the scheme can be obtained by substituting $p \leftarrow \mathrm{j}\omega$:

$$H(\mathrm{j}\omega) = \frac{1 - \omega\tau}{1 + \mathrm{j}\omega\tau}.$$

The magnitude and phase of the frequency response, for $\tau = \frac{1}{2}$, are presented in Figure C.2. Both responses are clearly non-symmetric with respect to $\omega = 0$, the notch and phase change are both at $\omega_0 = \frac{1}{\tau} = 2$.

For $\omega = 0$, the ideal capacitor C behaves as a break and therefore only the resistors in the phases are present. As a result the magnitude equals 0 dB and the phase equals 0 radians. For $|\omega| \to \infty$ the ideal capacitor C behaves as a short-circuit and as a consequence the magnitude and phase approximate 0 dB and $\frac{\pi}{2}$ radians respectively.

If for instance both the input signal X_{i} and the output signal Y_{i} are negated, by simply interchanging the $+$ and $-$ terminals of the input and output buffers respectively, the system function realized is conjugated. The zero is then moved from $\omega = 2$ to $\omega = -2$.

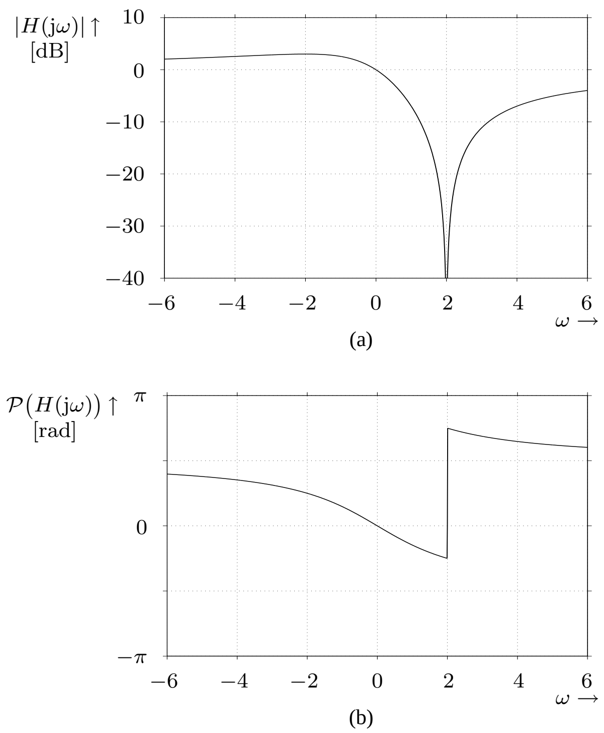


Figure C.2: *Magnitude and phase of the frequency response corresponding to the polyphase filter in Figure C.1 for* $\tau = \frac{1}{2}$.

C.2 Remark

In the previous example only a single passive section, with one R and one C per phase, is used. If more sections, with possible other values for R and C, other interconnects or active elements, are added, the calculus is a bit more laborious but the idea remains the same. It was also assumed that all components, all R's, all C's and all buffers are ideal: no parasites and no tolerances with respect to the actual value. In case of digital systems this is something that can be guaranteed trivially, whereas in analog systems this is practically impossible. Nevertheless analog polyphase filters are fruitfully applied in many systems, like for instance the GSM receiver [92], and they offer an additional degree of freedom to the system designer.

Appendix D

Introduction to Euclid's algorithms

In the process of manipulating multirate structures, for instance in Chapter 5 and Appendix A, it is often a prerequisite that two integer multirate factors have no common factor, in other words, they have to be relatively prime or coprime. An other important property as for instance is used in the split-delay identity, Lemma A.7, is that any integer number can be written as the integer-weighted sum of two coprimes. Euclid's algorithms can be used to determine both the common factor and the weights.

More information about these algorithms can be found in for instance [91]. The reasons to pay attention to these algorithms is that the presented descriptions often are hard to comprehend, although the basis of these algorithms is simple.

In this appendix the algorithms are introduced mainly on basis of an example and a description in a Pascal-like pseudo language. The so called Euclid's algorithm is used to determine the common factor and the so called extended Euclid's algorithm also provides the weights.

D.1 Euclid's algorithm

If two integers have no common factor they are said to be coprime. This is generally formulated as: the greatest common divisor of A and B equals 1, or $\gcd(A, B) = 1$. Euclid's algorithm can be used to determine the greatest common divisor of two natural numbers. In this section first some basic properties are presented followed by an example and a program description.

An important rule is that, if $A \geq B \geq 0$, then $\gcd(A, B) = \gcd(A - B, B)$, or formulated alternatively $\gcd(A, B) = \gcd(A|_B, B)$. Also important is the trivial property $\gcd(A, 0) = A$. For the special situation $A = B = 0$, in principle any number is a common factor, however the algorithm is designed to produce $\gcd(0, 0) = 0$.

Example D.1. For $A = 387$ and $B = 109$ the common factor or $\gcd$ is:

$$\begin{aligned}\gcd(387,109) &= \gcd(3\cdot 109+60,109) &&= \gcd(60,109)\\ &= \gcd(60,1\cdot 60+49) &&= \gcd(60,49)\\ &= \gcd(1\cdot 49+11,49) &&= \gcd(11,49)\\ &= \gcd(11,4\cdot 11+5) &&= \gcd(11,5)\\ &= \gcd(2\cdot 5+1,5) &&= \gcd(1,5)\\ &= \gcd(1,5\cdot 1+0) &&= \gcd(1,0)\\ &= 1.\end{aligned}$$

From this result it can be concluded that the numbers 387 and 109 are coprime.
End of example

In the following description of the algorithm, basically the same recipe is used as in the example. The procedure *swap*(x,y) is used to interchange, or swap, the values of the variables x and y.

```
proc euclid(A,B,gcd) ≡
   if A ≥ B ≥ 0
   then
      d0 := A; d1 := B;
      while d1 > 0
      do
         d0 := d0 mod d1; swap(d0,d1);
      od
      gcd := d0;
   else
      output('ERROR: inconsistent input');
      exit;
   fi
endproc
```

with

```
proc swap(x,y) ≡
   t := x; x := y; y := t;
endproc
```

D.2 Extended Euclid's algorithm

The extended Euclid's algorithm can be used to determine the integer weights i and j in:

$$\gcd(A,B) = i\cdot A + j\cdot B. \tag{D.1}$$

Equation D.1 is also known as Bézout's identity. The algorithm will first be illustrated by means of an example and second with a program description.

Example D.2. In the Example D.1, it was found that for $A = 387$ and $B = 109$ the $\gcd(A, B) = 1$. Now determine the integers i and j such that:

$$1 = i \cdot 387 + j \cdot 109.$$

The process is initiated with two trivial equations, line 0 and line 1 in Table D.2, where the original values are expressed in themselves. Now, the value $q = 387 \operatorname{div} 109 = 3$ is calculated, and line 2 becomes line 0 minus $q = 3$ times line 1. As a result the value $d = 60$ is formulated as an integer weighted sum of 387 and 109. Next, the value $q = 109 \operatorname{div} 60 = 1$ is calculated, and line 3 becomes line 1 minus $q = 1$ times line 2. As a result the new value $d = 49$ is formulated as an integer weighted sum of 387 and 109. This procedure is repeated as long as $d \geq 1$ (line 6). The equation in every next line is a linear combination of the two previous equations. Basically, the same steps as in Euclid's algorithm to derive the common factor are followed, and consequently the values in column d are identical to the values obtained earlier.

line	q	d	$=$	$i \cdot A$	$+$	$j \cdot B$
0		387	$=$	$1 \cdot 387$	$+$	$0 \cdot 109$
1		109	$=$	$0 \cdot 387$	$+$	$1 \cdot 109$
2	3	60	$=$	$1 \cdot 387$	$+$	$-3 \cdot 109$
3	1	49	$=$	$-1 \cdot 387$	$+$	$4 \cdot 109$
4	1	11	$=$	$2 \cdot 387$	$+$	$-7 \cdot 109$
5	4	5	$=$	$-9 \cdot 387$	$+$	$32 \cdot 109$
6	1	1	$=$	$20 \cdot 387$	$+$	$-71 \cdot 109$
7	5	0	$=$	$-109 \cdot 387$	$+$	$387 \cdot 109$

(D.2)

The equation in line 6 expresses the desired result.
End of example

Equation D.1 can trivially be extended to:

$$K \cdot \gcd(A, B) = K \cdot i \cdot A + K \cdot j \cdot B.$$

So, for all natural numbers A and B, with $\gcd(A, B) = 1$, it is always possible to determine the integers i and j to express the integer K as:

$$K = K \cdot i \cdot A + K \cdot j \cdot B.$$

In the following description of the algorithm, basically the same steps are taken as in the previous example. Note that the Euclid's algorithm as described earlier can be obtained from the extended version by deleting some parts. Observe that:
$q := d_0$ *div* d_1; $d_0 := d_0 - q * d_1$;
is equal to:
$d_0 := d_0$ *mod* d_1;.

```
proc extended_euclid(A,B,gcd,i,j) ≡
    if A ≥ B ≥ 0
    then
        d0 := A; i0 := 1; j0 := 0;    { line 0 in D.2 }
        d1 := B; i1 := 0; j1 := 1;    { line 1 in D.2 }
        while d1 > 0
        do
            q := d0 div d1;
            d0 := d0 − q ∗ d1; swap(d0,d1);
            i0 := i0 − q ∗ i1; swap(i0,i1);
            j0 := j0 − q ∗ j1; swap(j0,j1);    { lines 2..7 in D.2 }
        od
        gcd := d0; i := i0; j := j0;    { copy line 6 in D.2 }
    else
        output('ERROR: inconsistent input');
        exit;
    fi
endproc
```

Appendix E

Alternatives for coefficients

An operation that occurs in almost every DSP algorithm is the multiplication of a signal sample with a constant or coefficient. In principle a general purpose multiplier can be used for such multiplication. In many applications, however, such a multiplier may be a bottle-neck in view of speed and costs.
An interesting alternative, especially when both the sample and coefficient are integers or scaled integers, is the *hard-wired* multiplier. The *binary* representation of an integer directly inspires the *shift-and-add* construction for such multiplier. For any $a \in \mathbb{N}$ the unique binary number representation is the series of bits $a_i \in \{0, 1\}$ such that:

$$a = \sum_{i \geq 0} a_i 2^i.$$

In many cases the costs of the shift operations can be neglected compared to the costs of the adders. From this it follows that the costs of a multiplication are related to the number of ones in the binary representation of the coefficient. More precisely, the required number of adders is one less than the number of ones.
It is now a challenge to find alternative constructions that have lower costs. Already in 1951 Booth [10] introduced the so called *recoding* of binary numbers. Booth suggested to replace sequences of binary ones like, e.g., $[00111]_2 \leftarrow [0100\overline{1}]_2$ where $\overline{1}$ stands for -1. Since the costs of an adder and a subtractor are practically equal, this recoding clearly can give a cost reduction if the sequence of 1's is sufficiently long.
The introduction of the signed-digit, as $\overline{1}$ is, makes number representations non-unique or redundant. In 1960 Reitwiesner [120] presented a special definition for binary signed-digit numbers, and proved their uniqueness and minimality of costs. For these reasons this type of number representation is often referred to as the Canonical Signed Digits (CSDs). For any $a \in \mathbb{Z}$, the unique CSD number representation is the series of trits $a_i \in \{-1, 0, 1\}$ such that:

$$a = \sum_{i \geq 0} a_i 2^i \wedge a_i a_{i+1} = 0.$$

In the same period of time the property of signed-digits was also studied by others, e.g., [3], and also special number systems for complex coefficients were introduced [74] [112]. In the following period a much effort was, and today still is, directed to the design of efficient algorithms for designing single coefficients. Bernstein [6] presented his algorithm that exploited some kind of factorization in 1986, and much later this algorithm was still under discussion [14]. Based on [6] many improvements/extensions were presented [26] [45] [47] [78] [79]. Instead of optimizing a single coefficient also blocks of coefficients, i.e., coefficients with a common input or output, are optimized [11] [42] [46].
An important approach for cost reduction is the search for a repetition of patterns or subexpressions intra [45] [78], and inter coefficients [54] [87] [140]. Possible lower bounds for the costs are studied in [80] where the focus is on coefficients that consist of multiple-hundreds of bits.
The work in this appendix is *not* about efficient or alternative search methods or programs like in most references. After defining minimal costs and the related minimal constructions in Section E.1, alternative constructions for integers are presented in Section E.2. Here cascade connections of CSD-based constructions are used to obtain constructions of lower costs, also sums and differences of such cascade connections are used. The results for the integers together with the structures discussed in Section 4.5 and Section 4.6, are used to construct complex integers in Section E.3. The possibilities to use scaled coefficients for both the integers and complex integers are elaborated in Section E.4. The remarkable property that the scaling of coefficients with a factor of 2 may result in lower costs is also illustrated. Finally, in Section E.5 a number of examples are presented. The results from this appendix can, and some already are, be used in the DESFIL software package, see Section 1.5.
Appendix F provides next to a brief historical overview, also a comparison of implementation costs of the standard base-2 and the complex-base system from [112]. Appendix G gives a brief introduction to complex primes to support the analysis of complex integers and their factorization.

E.1 Costs, minimal costs and minimal constructions

In many papers it is assumed that adders and subtractors have equal costs, e.g., [44], and that bit-shifts are for free. An adder consists typically of a single half-adder and multiple full-adders, and a subtractor consists typically of full-adders and inverters for the inputs of the subtrahend. Since the costs of adders and subtractors are proportional to the number of bits to represent the data, in principle costs can be reduced when the maximum shift is limited [108]. In many papers, however, the effect of the number of bits on the costs is neglected. A cascade connection is the connection of outputs from one circuit to the inputs of next circuits. An output driver should be capable of driving a single or multiple inputs. In [9] examples of alternatives for coefficients are presented that require less additions but more equivalent gates. A synthesis tool does not necessarily transform a minimal number of adders into a minimal chip area. In this appendix the costs of the adder and the subtractor are defined as 1, and the costs for the cascade connector are defined as 0.
Since both the binary and CSD number representations are unique, the constructions ac-

cording to these representations are unique, and hence the costs of these constructions are minimal. The constructions based on the binary and CSD representations will be referred to as $\mathcal{B}$ and $\mathcal{C}$ respectively. The possible alternatives for the constructions that will be considered in the remainder of this appendix are not necessarily unique. For obvious reasons the focus is on alternatives for coefficients, a, with minimal costs in case particular constructions, $\mathcal{X}$, may be used. In addition, mulitiple alternatives for a coefficient can have minimal costs.

Definition E.1. *For any $a \in \mathbb{Z}$ or $a \in \mathbb{C}_{\mathbb{Z}}$ constructed from constructions in $\mathcal{X}$:*

- *the* minimal costs *are denoted as: $C_{\mathcal{X}}(a)$, and*
- *the* minimal constructions *are denoted as: $\Gamma_{\mathcal{X}}(a)$.*

Using constructions $\mathcal{X}$, the minimal constructions for coefficient a, $\Gamma_{\mathcal{X}}(a)$, form a set of alternatives that all have the same minimal costs, $C_{\mathcal{X}}(a)$. For the binary and CSD constructions the minimal constructions contain exactly 1 element. In the remainder of this appendix many minimal constructions are considered from which only a single element will be mentioned. Also, the special symbols $\oplus$ and $\ominus$ are used for the adder and subtractor respectively. This is to emphasize that these are counted in the costs, and to distinguish from the ordinary $+$ and $-$ symbols as used in the notation for the complex integers.

Example E.1. The unique construction of coefficient value 7, using the binary constructions $\mathcal{B}$, is:

$$\Gamma_{\mathcal{B}}(7) \ni 2^2 \oplus 2^1 \oplus 2^0 \text{ and } C_{\mathcal{B}}(7) = 2.$$

This can be read as: Using the binary constructions $\mathcal{B}$ to construct coefficient 7, the minimal constructions contain at least $2^2 \oplus 2^1 \oplus 2^0$ and the minimal costs are 2. Alternatively, the unique construction of coefficient value 7, using the CSD constructions $\mathcal{C}$, is:

$$\Gamma_{\mathcal{C}}(7) \ni 2^3 \ominus 2^0 \text{ and } C_{\mathcal{C}}(7) = 1.$$

This can be read as: Using the CSD constructions $\mathcal{C}$ to construct coefficient 7, the minimal constructions contain at least $2^3 \ominus 2^0$ and the minimal costs are 1.
End of example

To enable the comparison of costs of arbitrary constructions for coefficients, the Averaged Costs (AC) are defined for both integer and complex integer coefficients.

Definition E.2. *For all coefficients $a \in \mathbb{Z}$ or $a \in \mathbb{C}_{\mathbb{Z}}$ that are constructed using constructions $\mathcal{X}$, the Averaged Costs (AC) are defined as:*

$$AC_{\mathcal{X}}(\Xi) \triangleq \frac{1}{N} \sum_{0<\|a\|_\infty \leq \Xi} C_{\mathcal{X}}(a),$$

with $N = \mathrm{N}_a :: 0 < \|a\|_\infty \leq \Xi$.

For $a \in \mathbb{Z}$ and $a \in \mathbb{C}_{\mathbb{Z}}$, the value of N is respectively $N = 2a$ and $N = 4a(a+1)$. The averaged costs can be interpreted as follows. For a uniformly distributed set of coefficients a with $\|a\|_\infty \leq \Xi$, $AC_{\mathcal{X}}(\Xi)$ expresses the average costs, if constructions from $\mathcal{X}$ are used.

E.2 Alternative constructions for integers

The standard constructions $\mathcal{B}$ and $\mathcal{C}$ all are sums or differences of powers of 2. Alternatively, it is possible to save on costs by putting standard constructions in a cascade connection, as is illustrated in Example E.2.

Example E.2. An implementation of coefficient value 45, using the binary constructions $\mathcal{B}$, gives:

$$\Gamma_{\mathcal{B}}(45) \ni 2^5 \oplus 2^3 \oplus 2^2 \oplus 2^0 \text{ and } C_{\mathcal{B}}(45) = 3.$$

Usage of the CSD constructions $\mathcal{C}$ does not reduce the minimal costs:

$$\Gamma_{\mathcal{C}}(45) \ni 2^6 \ominus 2^4 \ominus 2^2 \oplus 2^0 \text{ and } C_{\mathcal{C}}(45) = 3.$$

Now consider the following two alternative constructions for 45 that both have costs of 2:

$$(2^3 \oplus 2^0)(2^2 \oplus 2^0),$$
$$(2^4 \ominus 2^0)(2^2 \ominus 2^0).$$

These constructions are the cascade connections of the CSD constructions for the factors 9 and 5, and the factors 15 and 3, respectively.
End of example

In the remainder of this section new constructions are defined and evaluated that exploit the possibilities of cascading, adding and subtracting of constructions. The idea is that for a given coefficient, the CSD construction may have higher costs than a cascade connection of CSD constructions for the factors of the coefficient. Also, for a given coefficient such cascade connection may have higher costs than the adding or subtracting of cascades. The recipe of alternately cascading and adding or subtracting constructions, can be repeated until no progress in cost reduction can be achieved. The resulting constructions are denoted with $\mathcal{AC}$.
Since minimal constructions can be designed off-line on a super computer, the design process may be an exhaustive search. Of course, a more efficient design process is always welcome, but this is not the object of this appendix.
Creation of minimal constructions that consist of cascade connections, is based on evaluating all possible factorizations of the given integer value. The prime factorization gives a finite number of prime factors and the subsequent list of all possible combinations of prime factors is finite too. An algorithm to efficiently generate all different combinations is obtained from [137].
Creation of minimal constructions that consist of additions or subtractions is based on evaluating all possible partitions of the given integer value. For each integer an infinite number of partitions exists, e.g., $5 = 4+1 = 3+2 = 6-4+3 = 1000-990-5 = \cdots$. So, for practical reasons the range of the exhaustive search has to be limited. As a result however there is no guarantee that the designed construction set is optimal. To make this restriction explicit, a hat is added to the symbol, i.e., $\hat{\mathcal{AC}}$. In the sequel the limit is set to $|a| \leq 2^{16}$ which, most importantly, covers the most interesting coefficient values and also

results in a practical computer load. In [113] an optimization program was used to search for $|a| \leq 2^{20}$. For the range $2^{15} \leq |a| \leq 2^{16}$ in total 81 values were found with lower costs. In all cases intermediate values exceeding 2^{16} were used.

Example E.3. The smallest value for which a cascade connection of CSDs gives lower costs than the CSD itself, is 45. The smallest value for which an addition or subtraction of cascade connections of CSDs gives lower costs than a cascade connection of CSDs, is 173. The smallest value for which again cacading gives an improvement is 10995. The smallest value for which again adding or subtracting gives an improvement is 58107.

a	$\Gamma_{\hat{\mathcal{AC}}}(a) \ni$
45	$(2^3 \oplus 2^0)(2^2 \oplus 2^0)$
173	$2^3 \oplus (2^5 \oplus 2^0)(2^2 \oplus 2^0)$
10995	$(2^4 \ominus 2^0)((2^8 \ominus 2^0)(2^2 \ominus 2^0) \ominus 2^5)$
58107	$2^2((2^8 \ominus 2^0)(2^6 \oplus 2^0) \ominus 2^{11}) \ominus 2^0$

End of example

In Figure E.1 several ACs are given to show the relevance of the alternative constructions. Observe that for popular values of Ξ, i.e., $\Xi < 2^{10}$, the constructions $\hat{\mathcal{AC}}$ contribute in the reduction of costs.

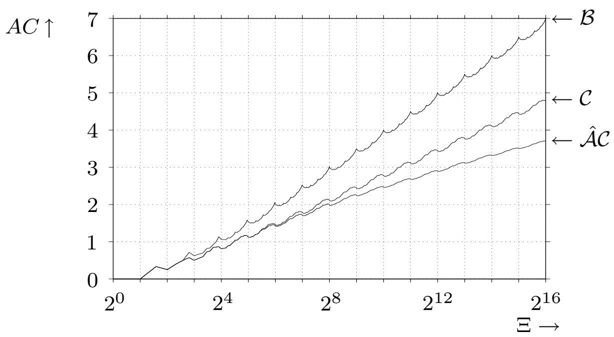


Figure E.1: *Averaged Costs for the standard constructions $\mathcal{B}$ and $\mathcal{C}$, and alternative constructions $\hat{\mathcal{AC}}$.*

For a limited set of values, $\Xi = 2^{10}$, examples of minimal constructions that have lower costs than CSD constructions, are listed in Table E.5. An example to illustrate the possible effect of alternative constructions for integer coefficients is presented in Section E.5.

E.3 Alternative constructions for complex integers

In the previous sections the constructions were designed to implement a multiplication of integer data with an integer coefficient. Now, the focus is on constructions to implement multiplications of complex-integer data with a complex-integer coefficient.

The constructions for complex-integer coefficients are made in terms of constructions for integer coefficients, and Structure A through Structure F for complex filters as presented in Section 4.5 and Section 4.6 (Figure 4.13, Figure 4.16 through Figure 4.19 and Figure 4.22). In these sections it is shown how a complex filter can be decomposed into its individual real and imaginary parts so requiring real subfilters only. By taking $H(z) = a$ with $a \in \mathbb{C}_{\mathbb{Z}}$ the individual real and imaginary parts become: $H_{\mathrm{r}}(z) = a_{\mathrm{r}}$ and $H_{\mathrm{i}}(z) = a_{\mathrm{i}}$ with $a_{\mathrm{r}}, a_{\mathrm{i}} \in \mathbb{Z}$. The respective subfilters are now integer coefficients only.
Structure A through Structure F are used here to define the *global constructions* of the complex-integer coefficient, and the constructions defined earlier for integer coefficients, are used for the *local constructions*. The fact that Structure F implements the coefficient value $2a$ instead of a can be ignored since a scaling with a factor 2 will not increase costs. In [48] it is shown that it is impossible to realize a denormalized lattice structure, i.e., $\left[\begin{smallmatrix} 1 & -q \\ q & 1 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} p & -1 \\ 1 & p \end{smallmatrix}\right]$ with one multiplication only. This is basically the same problem as realizing a complex integer a with either $a_{\mathrm{r}} = 1$ or $a_{\mathrm{i}} = 1$. So at least 2 multiplications are required which is consistent with the results here. Simplifying Structure A through Structure F by taking $H_{\mathrm{r}} = 1$, gives that Structure A, Structure C, Structure D and Structure E require 2 multiplications only. Similarly, taking $H_{\mathrm{i}} = 1$ gives that Structure A, Structure B, Structure D and Structure E require 2 multiplications only. Note that denormalizing over the integers in general is not possible.
Like for the integer coefficients the complex-integer constructions are defined for the complex-integer coefficients. For each of the global constructions the construction set deals with integer construction set $\mathcal{X}$ for the integer coefficients.

Definition E.3. *The complex-integer constructions with global constructions* X *and local integer constructions* $\mathcal{X}$*, are denoted as* $(\mathrm{X}, \mathcal{X})$.

First it will be shown that each of the global constructions A through E contribute in minimizing costs. Table E.1 lists 5 example values $a \in \mathbb{C}_{\mathbb{Z}}$ each with a different global construction to realize the minimal costs. All cases make use of the local constructions $\hat{\mathcal{A}}\mathcal{C}$.

	$C_{(.,\hat{\mathcal{A}}\mathcal{C})}(a)$					
a	A	B	C	D	E	F
$1 + \mathrm{j}2$	$\underline{2}$	4	4	4	5	8
$3 + \mathrm{j}$	4	$\underline{3}$	4	5	5	6
$1 + \mathrm{j}3$	4	4	$\underline{3}$	5	5	6
$408 + \mathrm{j}279$	10	10	10	$\underline{9}$	12	16
$279 - \mathrm{j}408$	10	10	10	12	$\underline{9}$	16

Table E.1: *Examples showing relevance of global constructions* A *through* E.

For all 5 example values global construction F has the highest costs. This is mainly caused by the 6 adders and subtractors that are required in Structure F regardless of the coefficient. It can be proven that there is no coefficient value for which global construction F has minimal costs. Therefore Structure F will not be considered anymore.
Similarly as for the integer coefficients, the ACs for several complex constructions are shown in Figure E.2. The top line $(\mathrm{A}, \mathcal{B})$ shows the AC for global construction A and

the binary constructions $\mathcal{B}$ as local constructions. Replacing constructions $\mathcal{B}$ by $\mathcal{C}$ or $\hat{\mathcal{A}}\mathcal{C}$, gives an improvement consistent with the results from Figure E.1, lines $(\mathrm{A}, \mathcal{C})$ and $(\mathrm{A}, \hat{\mathcal{A}}\mathcal{C})$ respectively. Minimization of the costs over the set of global constructions, i.e., $\mathrm{X} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, in combination with local constructions $\mathcal{C}$ and $\hat{\mathcal{A}}\mathcal{C}$, again gives further improvements, lines $(\mathrm{X}, \mathcal{C})$ and $(\mathrm{X}, \hat{\mathcal{A}}\mathcal{C})$ respectively, see Figure E.3.

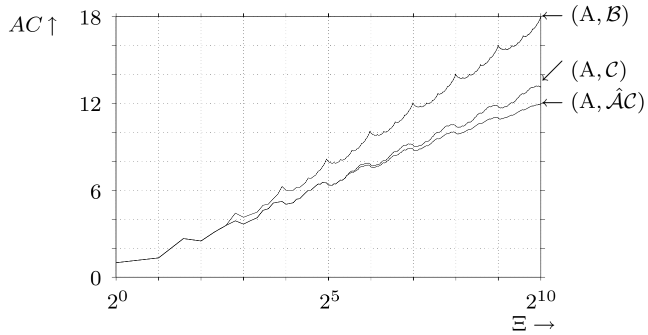


Figure E.2: *Averaged Costs for the complex construction sets:* $(\mathrm{A}, \mathcal{B})$, $(\mathrm{A}, \mathcal{C})$ *and* $(\mathrm{A}, \hat{\mathcal{A}}\mathcal{C})$.

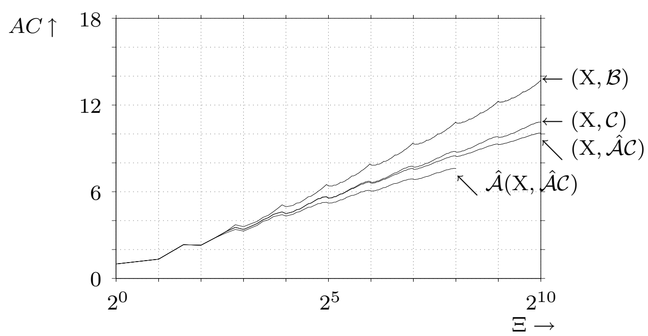


Figure E.3: *Averaged Costs for the complex construction sets:* $(\mathrm{X}, \mathcal{B})$, $(\mathrm{X}, \mathcal{C})$, $(\mathrm{X}, \hat{\mathcal{A}}\mathcal{C})$ *and* $\hat{\mathcal{A}}(\mathrm{X}, \hat{\mathcal{A}}\mathcal{C})$.

Similarly as the integers, the complex integers may be constructed using cascades, additions and subtractions. The resulting constructions are denoted with $\mathcal{A}(\mathrm{X}, \hat{\mathcal{A}}\mathcal{C})$. Creation of minimal constructions for complex-integers that consist of cascade connections is based on evaluating all possible factorizations of the given complex-integer value. The search process is identical to the process for integer values. The required *complex prime factorization* is described briefly in Appendix G. An algorithm to efficiently generate all different combinations is obtained from [137].

Creation of minimal constructions for complex-integer that consist of additions and subtractions is based on evaluating all possible partitions of the given complex-integer value like for the integer values. Here too the range of the exhaustive search has to be limited.

In the sequel the limit is set to $\|a\|_\infty \leq 2^8$ which, most importantly, covers the most interesting coefficient values and also results in a practical computer load. As a result however there is no guarantee that the designed constructions are optimal. To make this restriction explicit, a hat is added to the symbol, i.e., $\hat{\mathcal{A}}(\mathrm{X}, \hat{\mathcal{AC}})$.

Example E.4. In a similar way as in Example E.3 for the integers, the complex integers with the smallest norm for which an extra cascading or addition/subtraction is relevant, are shown. The alternative complex constructions are described in terms of complex integers between square brackets with the subscript referring to an optimal global construction for that complex integer, i.e., one of the structures Structure A through Structure E. As local construction for the integers the constructions $\hat{\mathcal{AC}}$ are used.
Starting with a cascade connection:

a	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{AC}})}(a) \ni$
$6 + \mathrm{j}3$	$[3]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$13 + \mathrm{j}10$	$[\mathrm{j}5]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A} \oplus [8]_\mathrm{A}$
$83 + \mathrm{j}54$	$[2+\mathrm{j}]_\mathrm{A}([64]_\mathrm{A} \ominus [5]_\mathrm{A}[4-\mathrm{j}]_\mathrm{A})$
$226 + \mathrm{j}123$	$[2]_\mathrm{A}([2+\mathrm{j}]_\mathrm{A}[64-\mathrm{j}]_\mathrm{A} \ominus [16]_\mathrm{A}) \ominus [\mathrm{j}]_\mathrm{A}$

Starting with an addition or subtraction:

a	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{AC}})}(a) \ni$
$6 + \mathrm{j}3$	$[\mathrm{j}]_\mathrm{A} \oplus [6+\mathrm{j}2]_\mathrm{B}$
$41 + \mathrm{j}38$	$([1]_\mathrm{A} \ominus [8-\mathrm{j}24]_\mathrm{C})[1-\mathrm{j}2]_\mathrm{A}$
$163 + \mathrm{j}120$	$[5]_\mathrm{A}([8+\mathrm{j}24]_\mathrm{C} \ominus [1]_\mathrm{A}) \oplus [128]_\mathrm{A}$

End of example

For a limited set of values, $\Xi = 2^5$, examples of minimal constructions that have lower costs than $(\mathrm{X}, \hat{\mathcal{AC}})$ constructions, are listed in Table E.6.

E.4 Scaled coefficients

For many applications, filters in particular, it is often irrelevant whether the filter $H(z)$ or the scaled filter $sH(z)$ is implemented. If values for scale factor s exist such that the coefficients of filter $sH(z)$ have lower costs, it is worthwhile to consider the alternatives. The general question is: do pairs (a, s) exist with $a, s \in \mathbb{Z}\backslash\{0\}$ or $a, s \in \mathbb{C}_\mathbb{Z}\backslash\{0\}$, for which: $C_\mathcal{X}(sa) < C_\mathcal{X}(a)$? This will be discussed in detail for the integers and the complex integers in the next sections.

E.4.1 Scaling integers

Inspection of $C_{\hat{\mathcal{AC}}}(a)$ teaches that $C_{\hat{\mathcal{AC}}}(sa) < C_{\hat{\mathcal{AC}}}(a)$ holds for many pairs (a, s) with $a \in \mathbb{Z}$ and $s \in \mathbb{N}^+$. In Figure E.4 the dots represent the pairs (a, s) for which $C_{\hat{\mathcal{AC}}}(sa) < C_{\hat{\mathcal{AC}}}(a)$ with $1 \leq a \leq 2^{10}$ and $1 \leq s \leq 2^7$. The right-top part of the figure is empty since

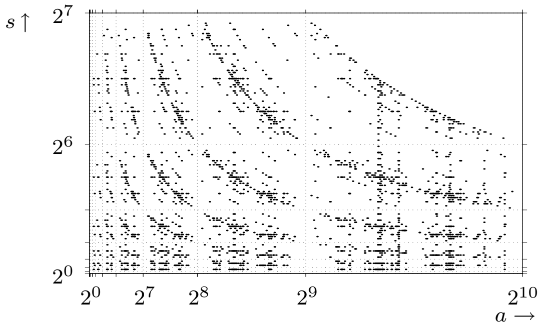


Figure E.4: *Pairs* (a, s) *for which* $C_{\hat{A}C}(sa) < C_{\hat{A}C}(a)$.

the cost function $C_{\hat{A}C}(a)$ is only available for $1 \le a \le 2^{16}$. As a consequence the values of s are limited by the hyperbole $s = \frac{2^{16}}{a}$.
A remarkable result is that in the range $0 \le 2a \le 2^{16}$, four values of a are found for which $C_{\hat{A}C}(2a) < C_{\hat{A}C}(a)$ implying that a multiplication with a factor of 2 may save costs. In Table E.2 these four values and their constructions are listed. This property for $a = 26827$ has been mentioned in [82]. Each implementation for $2a$ consists of a sum or difference with 2^0. In all cases the costs reduce from 5 to 4.

a	$\Gamma_{\hat{A}C}(a) \ni$	$\Gamma_{\hat{A}C}(2a) \ni$
26827	$(2^7 \oplus 2^4 \ominus 2^2 \ominus 2^0)(2^8 \ominus 2^6 \oplus 2^0)$	$(2^4 \ominus 2^0)(2^9 \ominus 2^0)(2^3 \ominus 2^0) \ominus 2^0$
26933	$2^2 \oplus (2^3 \ominus 2^0)(2^{12} \ominus 2^8 \oplus 2^3 \ominus 2^0)$	$(2^9 \oplus 2^0)(2^4 \ominus 2^0)(2^3 \ominus 2^0) \oplus 2^0$
28003	$(2^2 \ominus 2^0) \oplus (2^{12} \ominus 2^7 \oplus 2^5)(2^3 \ominus 2^0)$	$(2^6 \ominus 2^0)(2^3 \ominus 2^0)(2^7 \ominus 2^0) \ominus 2^0$
29347	$(2^2 \ominus 2^0) \oplus (2^8 \ominus 2^5)(2^7 \oplus 2^2 \ominus 2^0)$	$(2^7 \oplus 2^0)(2^6 \oplus 2^0)(2^3 \ominus 2^0) \ominus 2^0$

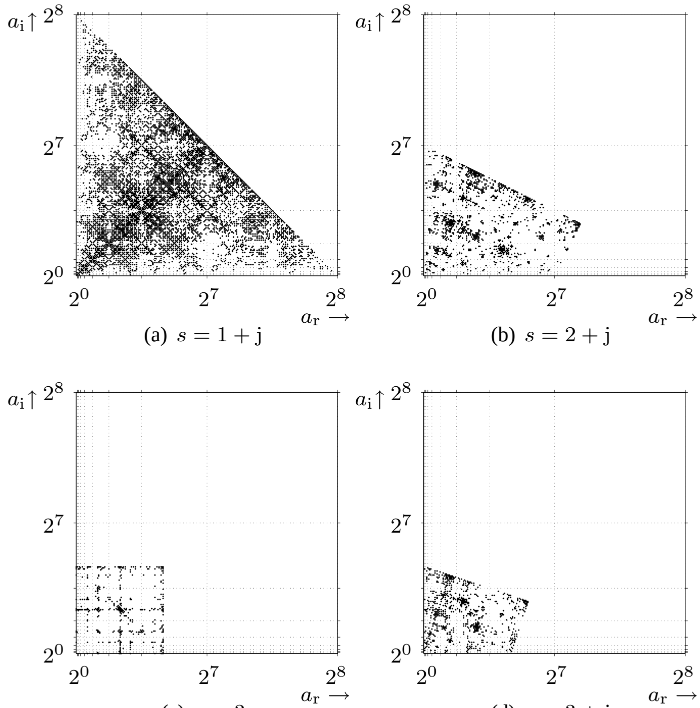
Table E.2: *Examples of* $a \in \mathbb{Z}$ *for which* $C_{\hat{A}C}(2a) < C_{\hat{A}C}(a)$.

In earlier sections it is shown that more complicated constructions require larger coefficients for being relevant. So it may be expected that for values a much larger than discussed here, in more cases $C_{\hat{A}C}(2a) < C_{\hat{A}C}(a)$ will hold. In [79] integers of *several hundreds of bits*, are evaluated without considering the even numbers. So particular there the savings may become significant.

E.4.2 Scaling complex integers

Also for the complex integers, many pairs (a, s) with $a, s \in \mathbb{C}_{\mathbb{Z}}$ exist for which $C_{\hat{A}(\mathrm{X},\hat{A}C)}(sa) < C_{\hat{A}(\mathrm{X},\hat{A}C)}(a)$ holds, with $\mathrm{X} = \{\mathrm{A, B, C, D, E}\}$.
For the integers it is easy to plot the results for many values of scale factor s. Since the complex integers have an additional degree of freedom, every value of s results in a separate plot. In Figure E.5 for some typical values of s, i.e., $s = 1 + \mathrm{j}$, $s = 2 + \mathrm{j}$, $s = 3$ and $s = 3 + \mathrm{j}$, those values of a are indicated with a black dot for which holds:

$C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(sa) < C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$ and $\|sa\|_\infty \leq 256$. This causes all a to lie within a rotated square. Only the first quadrant is shown since the other quadrant are rotated versions of the first.



(a) $s = 1 + \mathrm{j}$ (b) $s = 2 + \mathrm{j}$

(c) $s = 3$ (d) $s = 3 + \mathrm{j}$

Figure E.5: *For some typical values of s, the complex integers $a = a_\mathrm{r} + \mathrm{j}a_\mathrm{i}$ for which $C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(sa) < C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$ and $\mathrm{X} = \{\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D},\mathrm{E}\}$, are ploted.*

Similar as for the integers, there is the remarkable result that values of $a \in \mathbb{C}_\mathbb{Z}$ are found for which $C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(2a) < C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$. For $\|a\|_\infty < 2^8$ in total 16 solutions exist. Two of these are listed in Table E.3 and their costs reduce from 9 to 8. Each implementation for $2a$ consists of a sum or difference with 2^0 or j. The others are the conjugates and associates $\mathrm{j}^k a$ and $\mathrm{j}^k a^*$.

a	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a) \ni$	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(2a) \ni$
$105 + \mathrm{j}53$	$[105 + \mathrm{j}53]_\mathrm{C}$	$[105]_\mathrm{A}[2 + \mathrm{j}]_\mathrm{A} \oplus [\mathrm{j}]_\mathrm{A}$
$115 + \mathrm{j}87$	$[5 + \mathrm{j}7]_\mathrm{B}[16 - \mathrm{j}5]_\mathrm{A}$	$[2 + \mathrm{j}]_\mathrm{A}[8 + \mathrm{j}]_\mathrm{A}[16 + \mathrm{j}]_\mathrm{A} \ominus [\mathrm{j}]_\mathrm{A}$

Table E.3: *Examples of $a \in \mathbb{C}_\mathbb{Z}$ for which $C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(2a) < C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$.*

E.5 Examples

In this appendix many possibilities for costs reduction of integer and complex-integer multiplications are described. In this section a few typical examples will be discussed in more detail. Even in recent work only CSD-like methods are applied to reduce the number of adders and subtractors. Consider the following example.

Example E.5. In [39], where efficient transforms are designed, an implementation of the integer value 31183 is proposed at the cost of 5 adders. This equals the costs according the CSD constructions, i.e., $C_{\mathcal{C}}(31183) = 5$. On basis of constructions $\hat{\mathcal{A}}\mathcal{C}$ however, the results are:

$$\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(31183) \ni (2^9 \ominus 2^6) \oplus (2^4 \ominus 2^0)(2^{11} \oplus 2^0),$$
$$C_{\hat{\mathcal{A}}\mathcal{C}}(31183) = 4,$$

which gives a saving of 1.
End of example

The possibilities of alternative constructions for coefficients, as discussed in Section E.2, and scaled coefficients, as discussed in Section E.4, are illustrated next using a transversal filter structure with integer coefficients.

Example E.6. The property that the minimal costs of an integer or a complex integer after scaling, may be less than the costs of the original value, inspires an alternative for the transversal filter structure. Consider the $(1, 5)$-symmetric real filter $H(z) \in \mathbb{Z}/2^{14}(z)$ with a passband gain of 0 dB, a passband ripple of 1 dB and a stopband gain of -60 dB. The passband ranges from $\theta = 0$ through $\theta = 0.2\pi$ and the stopband ranges from $\theta = 0.6\pi$ through $\theta = \pi$. The filter length is $L = 11$.
For reasons of clarity a scaled version of the filter $H(z)$ will be used $H'(z) = 2^{14}H(z) \in \mathbb{Z}(z)$. The 6 different coefficients $h'[n]$ of the $(1, 5)$-symmetric filter $H'(z)$ are listed in column 2 of Table E.4. For the CSD constructions $\mathcal{C}$, the costs of the coefficients $h'[n]$ and $3h'[n]$ are shown in columns 3 and 4. Column 5 gives the optimal scale factor s. Similarly for the alternative construction set $\hat{\mathcal{A}}\mathcal{C}$, the data is presented in the columns 6 through 8.
It can be observed that for the central coefficient, $h'[5]$, the use of the scaling is beneficial for both constructions. The savings, 3 and 2 additions respectively, are more than the cost of the additional scaling with $s = 3$, $C_{\mathcal{C}}(3) = C_{\hat{\mathcal{A}}\mathcal{C}}(3) = 1$. Also, the application of alternative construction set $\hat{\mathcal{A}}\mathcal{C}$ is beneficial for coefficient $h'[2]$. This results in the following total costs for the coefficients and scale factor:

	$\mathcal{X} = \mathcal{C}$	$\mathcal{X} = \hat{\mathcal{A}}\mathcal{C}$
$s = 1$	15	13
$s = 3$	13	12

A structure that can be used to implement this filter is shown in Figure E.6. This concept can be generalized to a set of different factors s_i, at the cost of a less-regular filter structure. The alternative transversal structure, see Figure E.6 for an example, has some

resemblance to the filter structure as obtained in [87] where common subexpressions are exploited.
End of example

n		$h'[n]$	$C_{\mathcal{C}}(h'[n])$	$C_{\mathcal{C}}(3h'[n])$	s	$C_{\hat{\mathcal{A}}\mathcal{C}}(h'[n])$	$C_{\hat{\mathcal{A}}\mathcal{C}}(3h'[n])$	s
0	10	−124	1	3	1	1	2	1
1	9	−252	1	3	1	1	2	1
2	8	279	3	4	1	2	3	1
3	7	2065	2	5	1	2	3	1
4	6	4424	3	4	1	3	3	1 or 3
5		5548	5	2	3	4	2	3

$$\begin{aligned}
\Gamma_{\mathcal{C}}(279) &\ni 2^8 \oplus 2^5 \ominus 2^3 \ominus 2^0,\\
\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(279) &\ni (2^3 \oplus 2^0)(2^5 \ominus 2^0),\\
\Gamma_{\mathcal{C}}(5548) &\ni 2^{13} \ominus 2^{11} \ominus 2^9 \ominus 2^6 \ominus 2^4 \ominus 2^2,\\
\Gamma_{\mathcal{C}}(3 \cdot 5548) &\ni 2^{14} \oplus 2^8 \oplus 2^2,\\
\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(5548) &\ni (2^6 \oplus 2^4 \ominus 2^2)(2^6 \oplus 2^3 \oplus 2^0),\\
\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(3 \cdot 5548) &\ni 2^{14} \oplus 2^8 \oplus 2^2.
\end{aligned}$$

Table E.4: *Coefficients of the filters $H'(z)$ and $3H'(z)$, and their costs.*

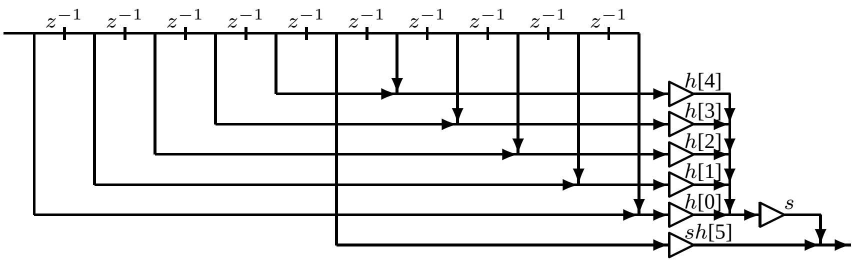


Figure E.6: *Transversal structure with shared and scaled multiplications.*

A limited list of alternative constructions that have lower costs than the standard constructions are presented in the following two examples. Compared to, e.g., the well-known CSDs the alternative constructions are not unique. Nevertheless only a single construction will be listed per coefficient.

Example E.7. For a limited set of integer values $1 \leq a \leq 2^{10}$, an element of the alternative constructions $\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(a)$, is presented in Table E.5. Only the odd values of a are listed since the even values can be obtained by considering $2^i a$. The smallest even value a that can be implemented at lower costs than the related odd value, $\frac{a}{2}$, is $a = 53654$, see

Section E.4 and Table E.2. An alternative is only presented if its costs are less than the costs of a CSD construction, so only if $C_{\hat{\mathcal{A}}\mathcal{C}}(a) < C_{\mathcal{C}}(a)$. Values for which the savings in costs of implementation are more than 1 compared to CSDs, are labeled with $\rightarrow$.
End of example

Example E.8. For a limited set of complex integers a for which $\|a\|_\infty < 2^5$, alternative constructions that have lower costs than $(\mathrm{X}, \hat{\mathcal{A}}\mathcal{C})$ constructions, are listed in Table E.6. Only the non-even values of a are listed since the even values can be obtained by considering $2^i a$. For the even value a that can be implemented at lower costs than the related odd value, $\frac{a}{2}$, holds: $\|a\|_\infty = 105$, see Section E.4 and Table E.3. An alternative is only presented if its costs are less than the minimal costs of global constructions A through E, so only if $C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a) < C_{(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$. Values for which the savings in costs of implementation are more than 2, are labeled with $\rightarrow$. Only values a for which $a_\mathrm{r} > 0 \wedge 0 \leq a_\mathrm{i} \leq a_\mathrm{r}$ are presented. The other values can be obtained by negation, multiplication by j^k and conjugation.
End of example

a	$C_{\mathcal{C}}(a)$	$C_{\hat{\mathcal{A}}\mathcal{C}}(a)$	$\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(a) \ni$
45	3	2	$(2^3 \oplus 2^0)(2^2 \oplus 2^0)$
51	3	2	$(2^2 \ominus 2^0)(2^4 \oplus 2^0)$
75	3	2	$(2^4 \ominus 2^0)(2^2 \oplus 2^0)$
85	3	2	$(2^2 \oplus 2^0)(2^4 \oplus 2^0)$
93	3	2	$(2^2 \ominus 2^0)(2^5 \ominus 2^0)$
99	3	2	$(2^5 \oplus 2^0)(2^2 \ominus 2^0)$
105	3	2	$(2^4 \ominus 2^0)(2^3 \ominus 2^0)$
153	3	2	$(2^3 \oplus 2^0)(2^4 \oplus 2^0)$
155	3	2	$(2^2 \oplus 2^0)(2^5 \ominus 2^0)$
165	3	2	$(2^5 \oplus 2^0)(2^2 \oplus 2^0)$
171	4	3	$(2^3 \oplus 2^0)(2^4 \oplus 2^2 \ominus 2^0)$
173	4	3	$2^3 \oplus (2^5 \oplus 2^0)(2^2 \oplus 2^0)$
179	4	3	$(2^2 \ominus 2^0)(2^4 \oplus 2^0) \oplus 2^7$
181	4	3	$2^0 \oplus (2^5 \oplus 2^2)(2^2 \oplus 2^0)$
189	3	2	$(2^6 \ominus 2^0)(2^2 \ominus 2^0)$
195	3	2	$(2^2 \ominus 2^0)(2^6 \oplus 2^0)$
203	4	3	$(2^3 \ominus 2^0)(2^5 \ominus 2^2 \oplus 2^0)$
205	4	3	$(2^2 \oplus 2^0)(2^5 \oplus 2^3 \oplus 2^0)$
211	4	3	$2^0 \oplus (2^5 \ominus 2^1)(2^3 \ominus 2^0)$
213	4	3	$(2^2 \ominus 2^0)(2^6 \oplus 2^3 \ominus 2^0)$
217	3	2	$(2^3 \ominus 2^0)(2^5 \ominus 2^0)$
231	3	2	$(2^5 \oplus 2^0)(2^3 \ominus 2^0)$
279	3	2	$(2^3 \oplus 2^0)(2^5 \ominus 2^0)$
297	3	2	$(2^3 \oplus 2^0)(2^5 \oplus 2^0)$
299	4	3	$2^1 \oplus (2^3 \oplus 2^0)(2^5 \oplus 2^0)$
301	4	3	$2^0 \oplus (2^6 \ominus 2^2)(2^2 \oplus 2^0)$
307	4	3	$2^0 \oplus (2^4 \oplus 2^1)(2^4 \oplus 2^0)$
309	4	3	$(2^3 \oplus 2^1)(2^5 \ominus 2^0) \ominus 2^0$
315	3	2	$(2^6 \ominus 2^0)(2^2 \oplus 2^0)$
325	3	2	$(2^6 \oplus 2^0)(2^2 \oplus 2^0)$
331	4	3	$2^0 \oplus (2^6 \oplus 2^1)(2^2 \oplus 2^0)$
333	4	3	$(2^3 \oplus 2^0)(2^5 \oplus 2^2 \oplus 2^0)$
339	4	3	$(2^2 \ominus 2^0)(2^7 \ominus 2^4 \oplus 2^0)$
341	4	3	$(2^4 \ominus 2^2 \ominus 2^0)(2^5 \ominus 2^0)$
343	4	3	$(2^6 \ominus 2^4 \oplus 2^0)(2^3 \ominus 2^0)$
345	4	3	$(2^4 \ominus 2^0)(2^5 \ominus 2^3 \ominus 2^0)$
347	4	3	$2^5 \oplus (2^6 \ominus 2^0)(2^2 \oplus 2^0)$
349	4	3	$(2^2 \ominus 2^0)(2^5 \ominus 2^0) \oplus 2^8$
355	4	3	$(2^2 \oplus 2^0)(2^6 \oplus 2^3 \ominus 2^0)$
357	4	3	$(2^4 \oplus 2^2 \oplus 2^0)(2^4 \oplus 2^0)$
359	4	3	$2^7 \oplus (2^5 \oplus 2^0)(2^3 \ominus 2^0)$
361	4	3	$2^0 \oplus (2^6 \oplus 2^3)(2^2 \oplus 2^0)$

continued on next page

Table E.5: *Examples of alternative constructions for integer coefficients.*

continued from previous page

a	$C_{\mathcal{C}}(a)$	$C_{\hat{\mathcal{A}}\mathcal{C}}(a)$	$\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(a) \ni$
363	4	3	$(2^5 \oplus 2^0)(2^4 \ominus 2^2 \ominus 2^0)$
365	4	3	$(2^2 \oplus 2^0)(2^6 \oplus 2^3 \oplus 2^0)$
371	4	3	$(2^4 \ominus 2^2)(2^5 \ominus 2^0) \ominus 2^0$
373	4	3	$2^0 \oplus (2^4 \ominus 2^2)(2^5 \ominus 2^0)$
381	3	2	$(2^2 \ominus 2^0)(2^7 \ominus 2^0)$
387	3	2	$(2^7 \oplus 2^0)(2^2 \ominus 2^0)$
395	4	3	$(2^2 \oplus 2^0)(2^6 \oplus 2^4 \ominus 2^0)$
397	4	3	$2^0 \oplus (2^7 \oplus 2^2)(2^2 \ominus 2^0)$
403	4	3	$(2^4 \ominus 2^2 \oplus 2^0)(2^5 \ominus 2^0)$
405	4	3	$(2^6 \oplus 2^4 \oplus 2^0)(2^2 \oplus 2^0)$
407	4	3	$2^7 \oplus (2^3 \oplus 2^0)(2^5 \ominus 2^0)$
409	4	3	$2^0 \oplus (2^5 \ominus 2^3)(2^4 \oplus 2^0)$
411	4	3	$(2^2 \ominus 2^0)(2^7 \oplus 2^3 \oplus 2^0)$
413	4	3	$(2^3 \ominus 2^0)(2^6 \ominus 2^2 \ominus 2^0)$
419	4	3	$2^5 \oplus (2^7 \oplus 2^0)(2^2 \ominus 2^0)$
421	4	3	$2^0 \oplus (2^6 \ominus 2^2)(2^3 \ominus 2^0)$
423	4	3	$(2^3 \oplus 2^0)(2^6 \ominus 2^4 \ominus 2^0)$
425	4	3	$(2^5 \ominus 2^3 \oplus 2^0)(2^4 \oplus 2^0)$
427	4	3	$(2^3 \ominus 2^0)(2^6 \ominus 2^2 \oplus 2^0)$
429	4	3	$(2^5 \oplus 2^0)(2^4 \ominus 2^2 \oplus 2^0)$
435	4	3	$(2^4 \ominus 2^0)(2^5 \ominus 2^2 \oplus 2^0)$
437	4	3	$(2^6 \ominus 2^0)(2^3 \ominus 2^0) \ominus 2^2$
441	3	2	$(2^6 \ominus 2^0)(2^3 \ominus 2^0)$
455	3	2	$(2^6 \oplus 2^0)(2^3 \ominus 2^0)$
459	4	3	$(2^5 \ominus 2^2 \ominus 2^0)(2^4 \oplus 2^0)$
461	4	3	$(2^6 \oplus 2^1)(2^3 \ominus 2^0) \ominus 2^0$
465	3	2	$(2^4 \ominus 2^0)(2^5 \ominus 2^0)$
467	4	3	$2^1 \oplus (2^4 \ominus 2^0)(2^5 \ominus 2^0)$
469	4	3	$(2^3 \ominus 2^0)(2^6 \oplus 2^2 \ominus 2^0)$
555	4	3	$(2^4 \ominus 2^0)(2^5 \oplus 2^2 \oplus 2^0)$
557	4	3	$(2^3 \oplus 2^0)(2^2 \oplus 2^0) \oplus 2^9$
561	3	2	$(2^5 \oplus 2^0)(2^4 \oplus 2^0)$
563	4	3	$2^1 \oplus (2^5 \oplus 2^0)(2^4 \oplus 2^0)$
565	4	3	$(2^2 \oplus 2^0)(2^7 \ominus 2^4 \oplus 2^0)$
567	3	2	$(2^6 \ominus 2^0)(2^3 \oplus 2^0)$
585	3	2	$(2^3 \oplus 2^0)(2^6 \oplus 2^0)$
587	4	3	$2^1 \oplus (2^3 \oplus 2^0)(2^6 \oplus 2^0)$
589	4	3	$(2^4 \oplus 2^2 \ominus 2^0)(2^5 \ominus 2^0)$
595	4	3	$(2^5 \oplus 2^2 \ominus 2^0)(2^4 \oplus 2^0)$
597	4	3	$(2^2 \oplus 2^0)(2^4 \oplus 2^0) \oplus 2^9$
599	4	3	$2^5 \oplus (2^6 \ominus 2^0)(2^3 \oplus 2^0)$

continued on next page

Table E.5: *Examples of alternative constructions for integer coefficients.*

continued from previous page

	a	$C_{\mathcal{C}}(a)$	$C_{\hat{A}\mathcal{C}}(a)$	$\Gamma_{\hat{A}\mathcal{C}}(a) \ni$
	601	4	3	$2^0 \oplus (2^7 \ominus 2^3)(2^2 \oplus 2^0)$
	603	4	3	$(2^3 \oplus 2^0)(2^6 \oplus 2^2 \ominus 2^0)$
	605	4	3	$(2^2 \oplus 2^0)(2^7 \ominus 2^3 \oplus 2^0)$
	611	4	3	$(2^5 \oplus 2^0)(2^2 \ominus 2^0) \oplus 2^9$
	613	4	3	$2^0 \oplus (2^5 \oplus 2^2)(2^4 \oplus 2^0)$
	615	4	3	$(2^4 \ominus 2^0)(2^5 \oplus 2^3 \oplus 2^0)$
	617	4	3	$2^5 \oplus (2^3 \oplus 2^0)(2^6 \oplus 2^0)$
	619	4	3	$(2^4 \oplus 2^2)(2^5 \ominus 2^0) \ominus 2^0$
	621	4	3	$(2^3 \oplus 2^0)(2^6 \oplus 2^2 \oplus 2^0)$
	627	4	3	$(2^5 \oplus 2^0)(2^4 \oplus 2^2 \ominus 2^0)$
	629	4	3	$(2^4 \oplus 2^0)(2^5 \oplus 2^2 \oplus 2^0)$
	635	3	2	$(2^2 \oplus 2^0)(2^7 \ominus 2^0)$
	645	3	2	$(2^7 \oplus 2^0)(2^2 \oplus 2^0)$
	651	4	3	$(2^4 \oplus 2^2 \oplus 2^0)(2^5 \ominus 2^0)$
	653	4	3	$2^3 \oplus (2^7 \oplus 2^0)(2^2 \oplus 2^0)$
	659	4	3	$(2^7 \oplus 2^2)(2^2 \oplus 2^0) \ominus 2^0$
	661	4	3	$2^0 \oplus (2^7 \oplus 2^2)(2^2 \oplus 2^0)$
	663	4	3	$(2^5 \oplus 2^3 \ominus 2^0)(2^4 \oplus 2^0)$
	665	4	3	$(2^7 \ominus 2^5 \ominus 2^0)(2^3 \ominus 2^0)$
	667	4	3	$2^5 \oplus (2^2 \oplus 2^0)(2^7 \ominus 2^0)$
	669	4	3	$(2^2 \ominus 2^0)(2^8 \ominus 2^5 \ominus 2^0)$
	675	4	3	$(2^7 \oplus 2^3 \ominus 2^0)(2^2 \oplus 2^0)$
	677	4	3	$2^5 \oplus (2^7 \oplus 2^0)(2^2 \oplus 2^0)$
	679	4	3	$(2^3 \ominus 2^0)(2^7 \ominus 2^5 \oplus 2^0)$
	681	4	3	$2^0 \oplus (2^5 \oplus 2^3)(2^4 \oplus 2^0)$
	683	5	4	$2^0 \oplus (2^5 \ominus 2^3 \ominus 2^1)(2^5 \ominus 2^0)$
$\rightarrow$	685	5	3	$(2^2 \oplus 2^0)(2^7 \oplus 2^3 \oplus 2^0)$
	689	4	3	$2^7 \oplus (2^5 \oplus 2^0)(2^4 \oplus 2^0)$
	691	5	4	$2^0 \oplus (2^5 \ominus 2^1)(2^5 \ominus 2^3 \ominus 2^0)$
$\rightarrow$	693	5	3	$(2^6 \ominus 2^0)(2^4 \ominus 2^2 \ominus 2^0)$
	695	4	3	$2^7 \oplus (2^6 \ominus 2^0)(2^3 \oplus 2^0)$
	697	4	3	$(2^4 \oplus 2^0)(2^5 \oplus 2^3 \oplus 2^0)$
	699	4	3	$2^6 \oplus (2^2 \oplus 2^0)(2^7 \ominus 2^0)$
	701	4	3	$(2^6 \ominus 2^0)(2^2 \ominus 2^0) \oplus 2^9$
	707	4	3	$(2^2 \ominus 2^0)(2^6 \oplus 2^0) \oplus 2^9$
	709	4	3	$2^6 \oplus (2^7 \oplus 2^0)(2^2 \oplus 2^0)$
	711	4	3	$(2^3 \oplus 2^0)(2^6 \oplus 2^4 \ominus 2^0)$
	713	4	3	$(2^5 \ominus 2^3 \ominus 2^0)(2^5 \ominus 2^0)$
$\rightarrow$	715	5	3	$(2^6 \oplus 2^0)(2^4 \ominus 2^2 \ominus 2^0)$
$\rightarrow$	717	5	3	$(2^2 \ominus 2^0)(2^8 \ominus 2^4 \ominus 2^0)$
	719	4	3	$(2^7 \oplus 2^4)(2^2 \oplus 2^0) \ominus 2^0$

continued on next page

Table E.5: *Examples of alternative constructions for integer coefficients.*

continued from previous page

	a	$C_{\mathcal{C}}(a)$	$C_{\hat{\mathcal{A}}\mathcal{C}}(a)$	$\Gamma_{\hat{\mathcal{A}}\mathcal{C}}(a) \ni$
	721	4	3	$2^0 \oplus (2^7 \oplus 2^4)(2^2 \oplus 2^0)$
$\rightarrow$	723	5	3	$(2^2 \ominus 2^0)(2^8 \ominus 2^4 \oplus 2^0)$
$\rightarrow$	725	5	3	$(2^7 \oplus 2^4 \oplus 2^0)(2^2 \oplus 2^0)$
	727	4	3	$2^{10} \ominus (2^3 \oplus 2^0)(2^5 \oplus 2^0)$
	729	4	3	$(2^6 \oplus 2^4 \oplus 2^0)(2^3 \oplus 2^0)$
	733	4	3	$(2^8 \ominus 2^0)(2^2 \ominus 2^0) \ominus 2^5$
	739	4	3	$(2^2 \ominus 2^0)(2^8 \oplus 2^0) \ominus 2^5$
	741	4	3	$(2^2 \ominus 2^0)(2^8 \ominus 2^3 \ominus 2^0)$
	743	4	3	$(2^5 \oplus 2^0)(2^3 \ominus 2^0) \oplus 2^9$
	745	4	3	$2^0 \oplus (2^5 \ominus 2^3)(2^5 \ominus 2^0)$
	747	4	3	$(2^8 \ominus 2^3 \oplus 2^0)(2^2 \ominus 2^0)$
	749	4	3	$(2^8 \ominus 2^0)(2^2 \ominus 2^0) \ominus 2^4$
	755	4	3	$(2^8 \ominus 2^2)(2^2 \ominus 2^0) \ominus 2^0$
	757	4	3	$2^0 \oplus (2^8 \ominus 2^2)(2^2 \ominus 2^0)$
	765	3	2	$(2^8 \ominus 2^0)(2^2 \ominus 2^0)$
	771	3	2	$(2^2 \ominus 2^0)(2^8 \oplus 2^0)$
	779	4	3	$2^3 \oplus (2^2 \ominus 2^0)(2^8 \oplus 2^0)$
	781	4	3	$2^0 \oplus (2^4 \ominus 2^2)(2^6 \oplus 2^0)$
	787	4	3	$2^4 \oplus (2^2 \ominus 2^0)(2^8 \oplus 2^0)$
	789	4	3	$(2^2 \ominus 2^0)(2^8 \oplus 2^3 \ominus 2^0)$
	791	4	3	$(2^3 \ominus 2^0)(2^7 \ominus 2^4 \oplus 2^0)$
	793	4	3	$2^0 \oplus (2^8 \oplus 2^3)(2^2 \ominus 2^0)$
	795	4	3	$(2^7 \oplus 2^5 \ominus 2^0)(2^2 \oplus 2^0)$
	797	4	3	$2^5 \oplus (2^8 \ominus 2^0)(2^2 \ominus 2^0)$
	803	4	3	$2^5 \oplus (2^2 \ominus 2^0)(2^8 \oplus 2^0)$
	805	4	3	$(2^2 \oplus 2^0)(2^7 \oplus 2^5 \oplus 2^0)$
	807	4	3	$2^{10} \ominus (2^3 \ominus 2^0)(2^5 \ominus 2^0)$
	809	4	3	$(2^3 \oplus 2^0)(2^5 \oplus 2^0) \oplus 2^9$
	811	5	4	$2^0 \oplus (2^7 \oplus 2^5 \oplus 2^1)(2^2 \oplus 2^0)$
$\rightarrow$	813	5	3	$(2^2 \ominus 2^0)(2^8 \oplus 2^4 \ominus 2^0)$
	815	4	3	$(2^6 \ominus 2^4)(2^4 \oplus 2^0) \ominus 2^0$
	817	4	3	$2^0 \oplus (2^6 \ominus 2^4)(2^4 \oplus 2^0)$
$\rightarrow$	819	5	3	$(2^6 \ominus 2^0)(2^4 \ominus 2^2 \oplus 2^0)$
	821	5	4	$2^0 \oplus (2^4 \oplus 2^2)(2^5 \oplus 2^3 \oplus 2^0)$
	823	4	3	$2^8 \oplus (2^6 \ominus 2^0)(2^3 \oplus 2^0)$
	825	4	3	$(2^4 \ominus 2^0)(2^6 \ominus 2^3 \ominus 2^0)$
	827	4	3	$(2^6 \ominus 2^0)(2^2 \oplus 2^0) \oplus 2^9$
	829	4	3	$2^6 \oplus (2^8 \ominus 2^0)(2^2 \ominus 2^0)$
	835	4	3	$2^6 \oplus (2^2 \ominus 2^0)(2^8 \oplus 2^0)$
	837	4	3	$(2^5 \ominus 2^2 \ominus 2^0)(2^5 \ominus 2^0)$
	839	4	3	$(2^7 \ominus 2^3)(2^3 \ominus 2^0) \ominus 2^0$

continued on next page

Table E.5: *Examples of alternative constructions for integer coefficients.*

continued from previous page

	a	$C_{\mathcal{C}}(a)$	$C_{\hat{A}\mathcal{C}}(a)$	$\Gamma_{\hat{A}\mathcal{C}}(a) \ni$
	841	4	3	$2^0 \oplus (2^7 \ominus 2^3)(2^3 \ominus 2^0)$
	843	5	4	$(2^2 \ominus 2^0)(2^8 \oplus 2^5 \ominus 2^3 \oplus 2^0)$
$\rightarrow$	845	5	3	$(2^6 \oplus 2^0)(2^4 \ominus 2^2 \oplus 2^0)$
	847	4	3	$(2^3 \ominus 2^0)(2^7 \ominus 2^3 \oplus 2^0)$
	849	4	3	$(2^6 \oplus 2^0)(2^4 \oplus 2^0) \ominus 2^8$
	851	5	4	$(2^5 \ominus 2^3 \ominus 2^0)(2^5 \oplus 2^2 \oplus 2^0)$
	853	5	4	$2^0 \oplus (2^4 \ominus 2^2)(2^6 \oplus 2^3 \ominus 2^0)$
	855	4	3	$(2^3 \oplus 2^0)(2^7 \ominus 2^5 \ominus 2^0)$
	857	4	3	$(2^3 \ominus 2^0)(2^7 \ominus 2^0) \ominus 2^5$
	859	4	3	$2^{10} \ominus (2^5 \oplus 2^0)(2^2 \oplus 2^0)$
	861	4	3	$(2^7 \ominus 2^2 \ominus 2^0)(2^3 \ominus 2^0)$
	867	4	3	$(2^2 \ominus 2^0)(2^8 \oplus 2^5 \oplus 2^0)$
	869	4	3	$2^0 \oplus (2^5 \ominus 2^2)(2^5 \ominus 2^0)$
	871	4	3	$(2^7 \oplus 2^0)(2^3 \ominus 2^0) \ominus 2^5$
	873	4	3	$(2^3 \oplus 2^0)(2^7 \ominus 2^5 \oplus 2^0)$
	875	4	3	$(2^7 \ominus 2^2 \oplus 2^0)(2^3 \ominus 2^0)$
	883	4	3	$2^0 \oplus (2^7 \ominus 2^1)(2^3 \ominus 2^0)$
	885	4	3	$(2^4 \ominus 2^0)(2^6 \ominus 2^2 \ominus 2^0)$
	889	3	2	$(2^3 \ominus 2^0)(2^7 \ominus 2^0)$
	903	3	2	$(2^7 \oplus 2^0)(2^3 \ominus 2^0)$
	907	4	3	$2^2 \oplus (2^7 \oplus 2^0)(2^3 \ominus 2^0)$
	909	4	3	$(2^7 \oplus 2^1)(2^3 \ominus 2^0) \ominus 2^0$
	915	4	3	$(2^4 \ominus 2^0)(2^6 \ominus 2^2 \oplus 2^0)$
	917	4	3	$(2^3 \ominus 2^0)(2^7 \oplus 2^2 \ominus 2^0)$
	919	4	3	$2^4 \oplus (2^7 \oplus 2^0)(2^3 \ominus 2^0)$
	921	4	3	$2^5 \oplus (2^3 \ominus 2^0)(2^7 \ominus 2^0)$
	923	4	3	$(2^7 \oplus 2^2)(2^3 \ominus 2^0) \ominus 2^0$
	925	4	3	$2^0 \oplus (2^7 \oplus 2^2)(2^3 \ominus 2^0)$
	931	4	3	$(2^7 \oplus 2^2 \oplus 2^0)(2^3 \ominus 2^0)$
	935	4	3	$(2^6 \ominus 2^3 \ominus 2^0)(2^4 \oplus 2^0)$
	937	4	3	$(2^6 \ominus 2^0)(2^4 \ominus 2^0) \ominus 2^3$
	939	4	3	$2^{10} \ominus (2^2 \oplus 2^0)(2^4 \oplus 2^0)$
	941	4	3	$(2^6 \ominus 2^0)(2^4 \ominus 2^0) \ominus 2^2$
	945	3	2	$(2^6 \ominus 2^0)(2^4 \ominus 2^0)$
	947	4	3	$2^1 \oplus (2^6 \ominus 2^0)(2^4 \ominus 2^0)$
	949	4	3	$2^2 \oplus (2^6 \ominus 2^0)(2^4 \ominus 2^0)$
	971	4	3	$(2^4 \ominus 2^0)(2^6 \oplus 2^0) \ominus 2^2$
	973	4	3	$(2^4 \ominus 2^0)(2^6 \oplus 2^0) \ominus 2^1$
	975	3	2	$(2^4 \ominus 2^0)(2^6 \oplus 2^0)$
	979	4	3	$2^2 \oplus (2^4 \ominus 2^0)(2^6 \oplus 2^0)$

Table E.5: *Examples of alternative constructions for integer coefficients.*

a	$C_{(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a) \ni$
$[6+\mathrm{j}3]_\mathrm{A}$	6	4	$[3]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[7+\mathrm{j}6]_\mathrm{A}$	6	4	$[2+\mathrm{j}]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A}$
$[9+\mathrm{j}3]_\mathrm{A}$	6	5	$[1+\mathrm{j}3]_\mathrm{C} \oplus [8]_\mathrm{A}$
$[10+\mathrm{j}5]_\mathrm{A}$	6	4	$[5]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[12+\mathrm{j}3]_\mathrm{A}$	6	4	$[3]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A}$
$[12+\mathrm{j}5]_\mathrm{A}$	6	5	$[\mathrm{j}]_\mathrm{A} \oplus [12+\mathrm{j}4]_\mathrm{B}$
$[13+\mathrm{j}6]_\mathrm{B}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[4+\mathrm{j}5]_\mathrm{A}$
$[13+\mathrm{j}10]_\mathrm{B}$	7	6	$[\mathrm{j}5]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A} \oplus [8]_\mathrm{A}$
$[14+\mathrm{j}7]_\mathrm{A}$	6	4	$[7]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[14+\mathrm{j}11]_\mathrm{C}$	7	6	$[7]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [\mathrm{j}4]_\mathrm{A}$
$[15+\mathrm{j}3]_\mathrm{A}$	6	5	$[-1+\mathrm{j}3]_\mathrm{C} \oplus [16]_\mathrm{A}$
$[15+\mathrm{j}5]_\mathrm{A}$	6	5	$[5]_\mathrm{A}[3+\mathrm{j}]_\mathrm{B}$
$[15+\mathrm{j}10]_\mathrm{A}$	6	4	$[2+\mathrm{j}]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A}$
$[17+\mathrm{j}3]_\mathrm{A}$	6	5	$[1+\mathrm{j}3]_\mathrm{C} \oplus [16]_\mathrm{A}$
$[17+\mathrm{j}6]_\mathrm{A}$	6	4	$[8-\mathrm{j}]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[18+\mathrm{j}9]_\mathrm{A}$	6	4	$[9]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[18+\mathrm{j}11]_\mathrm{C}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[5+\mathrm{j}8]_\mathrm{A}$
$[19+\mathrm{j}6]_\mathrm{A}$	8	6	$[3]_\mathrm{A}[1+\mathrm{j}2]_\mathrm{A} \oplus [16]_\mathrm{A}$
$[19+\mathrm{j}7]_\mathrm{B}$	7	6	$[9-\mathrm{j}]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[19+\mathrm{j}10]_\mathrm{B}$	7	6	$[2+\mathrm{j}]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A} \oplus [4]_\mathrm{A}$
$[20+\mathrm{j}5]_\mathrm{A}$	6	4	$[5]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A}$
$[21+\mathrm{j}6]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[7+\mathrm{j}2]_\mathrm{A}$
$[21+\mathrm{j}7]_\mathrm{B}$	6	5	$[7]_\mathrm{A}[3+\mathrm{j}]_\mathrm{B}$
$[21+\mathrm{j}10]_\mathrm{B}$	7	6	$[\mathrm{j}5]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A} \oplus [16]_\mathrm{A}$
$[21+\mathrm{j}14]_\mathrm{B}$	7	6	$[7]_\mathrm{A}[3+\mathrm{j}2]_\mathrm{A}$
$[21+\mathrm{j}18]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A}$
$[22+\mathrm{j}3]_\mathrm{A}$	8	6	$[4-\mathrm{j}]_\mathrm{A}[5+\mathrm{j}2]_\mathrm{A}$
$[22+\mathrm{j}5]_\mathrm{B}$	7	6	$[-5]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A} \oplus [32]_\mathrm{A}$
$[22+\mathrm{j}7]_\mathrm{B}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[1+\mathrm{j}4]_\mathrm{A} \oplus [16]_\mathrm{A}$
$[22+\mathrm{j}9]_\mathrm{B}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[7+\mathrm{j}8]_\mathrm{A}$
$[22+\mathrm{j}11]_\mathrm{B}$	8	6	$[11]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
$[22+\mathrm{j}13]_\mathrm{B}$	8	7	$[-\mathrm{j}]_\mathrm{A} \oplus [22+\mathrm{j}14]_\mathrm{B}$
$[22+\mathrm{j}15]_\mathrm{B}$	7	6	$[2+\mathrm{j}]_\mathrm{A}[-1+\mathrm{j}8]_\mathrm{A} \oplus [32]_\mathrm{A}$
$[22+\mathrm{j}17]_\mathrm{B}$	7	6	$[-1+\mathrm{j}8]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A} \oplus [16]_\mathrm{A}$
$[22+\mathrm{j}19]_\mathrm{B}$	8	6	$[8+\mathrm{j}]_\mathrm{A}[3+\mathrm{j}2]_\mathrm{A}$
$[22+\mathrm{j}21]_\mathrm{B}$	8	7	$[13+\mathrm{j}4]_\mathrm{B}[2+\mathrm{j}]_\mathrm{A}$
$[23+\mathrm{j}3]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A} \oplus [-1]_\mathrm{A}$
$[23+\mathrm{j}6]_\mathrm{B}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[8+\mathrm{j}7]_\mathrm{A}$
$[23+\mathrm{j}10]_\mathrm{B}$	7	6	$[4+\mathrm{j}]_\mathrm{A}[6+\mathrm{j}]_\mathrm{A}$
$[23+\mathrm{j}11]_\mathrm{B}$	7	5	$[3+\mathrm{j}]_\mathrm{B}[8+\mathrm{j}]_\mathrm{A}$
$[23+\mathrm{j}12]_\mathrm{A}$	8	6	$[12]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [-1]_\mathrm{A}$
$[23+\mathrm{j}14]_\mathrm{B}$	7	6	$[12+\mathrm{j}]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$

continued on next page

Table E.6: *Examples of alternative constructions for complex-integer coefficients.*

continued from previous page

	a	$C_{(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a) \ni$
	$[23+\mathrm{j}18]_\mathrm{B}$	7	6	$[9]_\mathrm{A}[-1+\mathrm{j}2]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[24+\mathrm{j}3]_\mathrm{A}$	6	4	$[3]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A}$
	$[24+\mathrm{j}7]_\mathrm{A}$	6	5	$[-\mathrm{j}]_\mathrm{A} \oplus [24+\mathrm{j}8]_\mathrm{B}$
	$[24+\mathrm{j}9]_\mathrm{A}$	6	5	$[\mathrm{j}]_\mathrm{A} \oplus [24+\mathrm{j}8]_\mathrm{B}$
	$[24+\mathrm{j}11]_\mathrm{A}$	8	6	$[4-\mathrm{j}]_\mathrm{A}[5+\mathrm{j}4]_\mathrm{A}$
	$[24+\mathrm{j}13]_\mathrm{A}$	8	6	$[3]_\mathrm{A}[8-\mathrm{j}]_\mathrm{A} \oplus [\mathrm{j}16]_\mathrm{A}$
	$[24+\mathrm{j}19]_\mathrm{A}$	8	6	$[3]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A} \oplus [\mathrm{j}16]_\mathrm{A}$
	$[24+\mathrm{j}21]_\mathrm{C}$	7	6	$[3]_\mathrm{A}[8+\mathrm{j}7]_\mathrm{A}$
	$[25+\mathrm{j}3]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[8+\mathrm{j}]_\mathrm{A} \oplus [1]_\mathrm{A}$
	$[25+\mathrm{j}5]_\mathrm{B}$	6	5	$[8-\mathrm{j}]_\mathrm{A}[3+\mathrm{j}]_\mathrm{B}$
	$[25+\mathrm{j}6]_\mathrm{B}$	7	6	$[2-\mathrm{j}]_\mathrm{A}[-4+\mathrm{j}]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[25+\mathrm{j}10]_\mathrm{B}$	7	6	$[5]_\mathrm{A}[5+\mathrm{j}2]_\mathrm{A}$
	$[25+\mathrm{j}12]_\mathrm{A}$	8	6	$[12]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [1]_\mathrm{A}$
	$[25+\mathrm{j}13]_\mathrm{B}$	8	7	$[\mathrm{j}4]_\mathrm{A} \oplus [25+\mathrm{j}9]_\mathrm{B}$
	$[25+\mathrm{j}14]_\mathrm{A}$	8	6	$[7]_\mathrm{A}[-1+\mathrm{j}2]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[25+\mathrm{j}18]_\mathrm{A}$	8	6	$[9]_\mathrm{A}[1+\mathrm{j}2]_\mathrm{A} \oplus [16]_\mathrm{A}$
	$[25+\mathrm{j}19]_\mathrm{B}$	8	7	$[8]_\mathrm{A} \oplus [17+\mathrm{j}19]_\mathrm{C}$
	$[25+\mathrm{j}20]_\mathrm{B}$	7	6	$[5]_\mathrm{A}[5+\mathrm{j}4]_\mathrm{A}$
	$[26+\mathrm{j}3]_\mathrm{A}$	8	6	$[3]_\mathrm{A}[-2+\mathrm{j}]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[26+\mathrm{j}5]_\mathrm{B}$	7	6	$[5]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [16]_\mathrm{A}$
	$[26+\mathrm{j}7]_\mathrm{B}$	7	6	$[9+\mathrm{j}8]_\mathrm{A}[2-\mathrm{j}]_\mathrm{A}$
	$[26+\mathrm{j}9]_\mathrm{B}$	7	6	$[9]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [8]_\mathrm{A}$
	$[26+\mathrm{j}11]_\mathrm{B}$	8	7	$[\mathrm{j}]_\mathrm{A} \oplus [26+\mathrm{j}10]_\mathrm{B}$
$\rightarrow$	$[26+\mathrm{j}13]_\mathrm{B}$	9	6	$[13]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A}$
	$[26+\mathrm{j}15]_\mathrm{A}$	8	6	$[4+\mathrm{j}]_\mathrm{A}[7+\mathrm{j}2]_\mathrm{A}$
	$[26+\mathrm{j}17]_\mathrm{A}$	8	6	$[1+\mathrm{j}8]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[26+\mathrm{j}23]_\mathrm{B}$	8	6	$[2+\mathrm{j}]_\mathrm{A}[15+\mathrm{j}4]_\mathrm{A}$
	$[27+\mathrm{j}6]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[9+\mathrm{j}2]_\mathrm{A}$
	$[27+\mathrm{j}9]_\mathrm{B}$	6	5	$[9]_\mathrm{A}[3+\mathrm{j}]_\mathrm{B}$
	$[27+\mathrm{j}10]_\mathrm{B}$	7	6	$[\mathrm{j}5]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[27+\mathrm{j}11]_\mathrm{B}$	7	6	$[-1+\mathrm{j}7]_\mathrm{A}[1-\mathrm{j}4]_\mathrm{A}$
	$[27+\mathrm{j}12]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[9+\mathrm{j}4]_\mathrm{A}$
	$[27+\mathrm{j}14]_\mathrm{A}$	8	6	$[14]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [-1]_\mathrm{A}$
	$[27+\mathrm{j}18]_\mathrm{B}$	7	6	$[9]_\mathrm{A}[3+\mathrm{j}2]_\mathrm{A}$
	$[27+\mathrm{j}20]_\mathrm{B}$	7	6	$[5]_\mathrm{A}[-1+\mathrm{j}4]_\mathrm{A} \oplus [32]_\mathrm{A}$
	$[27+\mathrm{j}21]_\mathrm{B}$	7	6	$[3+\mathrm{j}3]_\mathrm{B}[8-\mathrm{j}]_\mathrm{A}$
	$[27+\mathrm{j}24]_\mathrm{B}$	7	6	$[3]_\mathrm{A}[9+\mathrm{j}8]_\mathrm{A}$
	$[27+\mathrm{j}26]_\mathrm{B}$	8	6	$[2+\mathrm{j}]_\mathrm{A}[16+\mathrm{j}5]_\mathrm{A}$
	$[28+\mathrm{j}7]_\mathrm{A}$	6	4	$[7]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A}$
	$[28+\mathrm{j}11]_\mathrm{C}$	7	6	$[7]_\mathrm{A}[4+\mathrm{j}]_\mathrm{A} \oplus [\mathrm{j}4]_\mathrm{A}$
	$[28+\mathrm{j}13]_\mathrm{C}$	7	6	$[14]_\mathrm{A}[2+\mathrm{j}]_\mathrm{A} \oplus [-\mathrm{j}]_\mathrm{A}$

continued on next page

Table E.6: *Examples of alternative constructions for complex-integer coefficients.*

continued from previous page

a	$C_{(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$C_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a)$	$\Gamma_{\hat{\mathcal{A}}(\mathrm{X},\hat{\mathcal{A}}\mathcal{C})}(a) \ni$
$[28+\mathrm{j}19]_{\mathrm{C}}$	7	6	$[2+\mathrm{j}]_{\mathrm{A}}[15+\mathrm{j}2]_{\mathrm{A}}$
$[28+\mathrm{j}21]_{\mathrm{C}}$	7	6	$[\mathrm{j}7]_{\mathrm{A}}[3-\mathrm{j}4]_{\mathrm{A}}$
$[28+\mathrm{j}23]_{\mathrm{C}}$	7	6	$[7]_{\mathrm{A}}[4+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}16]_{\mathrm{A}}$
$[28+\mathrm{j}25]_{\mathrm{A}}$	8	6	$[7]_{\mathrm{A}}[4-\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}32]_{\mathrm{A}}$
$[29+\mathrm{j}6]_{\mathrm{A}}$	8	6	$[3]_{\mathrm{A}}[-1+\mathrm{j}2]_{\mathrm{A}} \oplus [32]_{\mathrm{A}}$
$[29+\mathrm{j}7]_{\mathrm{B}}$	7	6	$[7]_{\mathrm{A}}[4+\mathrm{j}]_{\mathrm{A}} \oplus [1]_{\mathrm{A}}$
$[29+\mathrm{j}10]_{\mathrm{A}}$	8	7	$[6+\mathrm{j}2]_{\mathrm{B}}[5]_{\mathrm{A}} \oplus [-1]_{\mathrm{A}}$
$[29+\mathrm{j}12]_{\mathrm{B}}$	7	6	$[2+\mathrm{j}]_{\mathrm{A}}[14-\mathrm{j}]_{\mathrm{A}}$
$[29+\mathrm{j}14]_{\mathrm{A}}$	8	6	$[14]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [1]_{\mathrm{A}}$
$[29+\mathrm{j}15]_{\mathrm{B}}$	7	6	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [-1]_{\mathrm{A}}$
$[29+\mathrm{j}17]_{\mathrm{B}}$	7	6	$[15+\mathrm{j}]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}}$
$[29+\mathrm{j}18]_{\mathrm{A}}$	8	6	$[2+\mathrm{j}]_{\mathrm{A}}[16+\mathrm{j}]_{\mathrm{A}} \oplus [-2]_{\mathrm{A}}$
$[29+\mathrm{j}20]_{\mathrm{B}}$	7	6	$[4+\mathrm{j}]_{\mathrm{A}}[8+\mathrm{j}3]_{\mathrm{A}}$
$[29+\mathrm{j}22]_{\mathrm{B}}$	8	6	$[16+\mathrm{j}3]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}}$
$[29+\mathrm{j}23]_{\mathrm{B}}$	8	7	$[-\mathrm{j}8]_{\mathrm{A}} \oplus [29+\mathrm{j}31]_{\mathrm{B}}$
$[29+\mathrm{j}24]_{\mathrm{A}}$	8	6	$[3]_{\mathrm{A}}[-1+\mathrm{j}8]_{\mathrm{A}} \oplus [32]_{\mathrm{A}}$
$[29+\mathrm{j}26]_{\mathrm{B}}$	8	7	$[-1]_{\mathrm{A}} \oplus [30+\mathrm{j}26]_{\mathrm{C}}$
$[30+\mathrm{j}11]_{\mathrm{A}}$	8	6	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [-\mathrm{j}4]_{\mathrm{A}}$
$[30+\mathrm{j}13]_{\mathrm{A}}$	8	6	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [-\mathrm{j}2]_{\mathrm{A}}$
$[30+\mathrm{j}15]_{\mathrm{A}}$	6	4	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}}$
$[30+\mathrm{j}19]_{\mathrm{A}}$	8	6	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}4]_{\mathrm{A}}$
$[30+\mathrm{j}21]_{\mathrm{C}}$	7	6	$[4+\mathrm{j}2]_{\mathrm{A}}[8+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}]_{\mathrm{A}}$
$[30+\mathrm{j}23]_{\mathrm{A}}$	8	6	$[15]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}8]_{\mathrm{A}}$
$[30+\mathrm{j}25]_{\mathrm{C}}$	7	6	$[7+\mathrm{j}16]_{\mathrm{A}}[2-\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}3]_{\mathrm{A}}$	6	5	$[-1+\mathrm{j}3]_{\mathrm{C}} \oplus [32]_{\mathrm{A}}$
$[31+\mathrm{j}11]_{\mathrm{C}}$	7	6	$[8+\mathrm{j}]_{\mathrm{A}}[4+\mathrm{j}]_{\mathrm{A}} \oplus [-\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}12]_{\mathrm{A}}$	6	4	$[8+\mathrm{j}]_{\mathrm{A}}[4+\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}13]_{\mathrm{C}}$	7	6	$[15-\mathrm{j}]_{\mathrm{A}}[2+\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}18]_{\mathrm{A}}$	6	4	$[2+\mathrm{j}]_{\mathrm{A}}[16+\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}19]_{\mathrm{C}}$	7	6	$[2+\mathrm{j}]_{\mathrm{A}}[16+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}]_{\mathrm{A}}$
$[31+\mathrm{j}22]_{\mathrm{A}}$	8	6	$[1+\mathrm{j}2]_{\mathrm{A}}[15-\mathrm{j}8]_{\mathrm{A}}$
$[31+\mathrm{j}26]_{\mathrm{C}}$	7	6	$[2+\mathrm{j}]_{\mathrm{A}}[16+\mathrm{j}]_{\mathrm{A}} \oplus [\mathrm{j}8]_{\mathrm{A}}$

Table E.6: *Examples of alternative constructions for complex-integer coefficients.*

Appendix F

Complex-base numbers: introduction and evaluation

Traditionally complex numbers are represented either with their real and imaginary parts or with their modulus and argument. In the literature some references to work are found that use a complex base for representing complex numbers. Out of curiosity these papers were studied and it was found that the claims that were made are questionable.

In 1960 Knuth [74] proposed to use a single-component positional numbering system for representing complex numbers. This allows for complex-arithmetic operations on single data elements, instead of a series of real arithmetic operations on the individual real and imaginary parts, or moduli and arguments. Knuth proposed the unique or non-redundant representation of any $a \in \mathbb{C}$ in a quaternary system, as:

$$a = \sum_i a_i p^i \quad \text{with base} \quad p = \mathrm{j}2 \quad \text{and} \quad a_i \in \{0, 1, 2, 3\}.$$

In the, for digital systems preferred, binary system with $a_i \in \{0, 1\}$, base $p = \mathrm{j}\sqrt{2}$ was proposed. However, the values $a \in \mathbb{C}_{\mathbb{Z}/2^k}$ cannot be represented with a finite number of bits. Somewhat later Penney [112] proposed the non-redundant alternative base $p = -1+\mathrm{j}$, with $a_i \in \{0, 1\}$, and the equally suitable $p = -1-\mathrm{j}$. Nielsen provides a thorough analysis of many, possibly exotic, number systems in [100].

Today, among others, Jamil [66] [67] and Khmelnik [72] [73] are developing hardware for arithmetic units to fully exploit the possibilities of the complex-base number systems. In [73] Khmelnik claims to achieve 5 to 10 times speedup, in terms of cycles, of complex number processing. No claims are made with respect to hardware costs.

In this appendix the use of the base-p number system for representing complex integers, with $p = -1 + \mathrm{j}$, is described and compared to the traditional base-2 system, as is generally used in this thesis. In particular the hardware costs of the addition and subtraction are evaluated. Also the multiplication form shift-and-add, is considered briefly.

F.1 Base $p = -1 + \mathrm{j}$: representation

In Table F.1 and Table F.2 some typical complex and scaled complex integers respectively are listed, both in base-10 and base-p. The shaded, wild-formed, area in Figure F.1 shows the set of complex integers that can be represented with at most 10 bits in base-p. The rectangular shows the set in case base-2 (2's-complement) is used with 5 bits for the real part and 5 bits for the imaginary part. Due to the wild form of the set in base-p, it is easy to see that numbers with a smaller modulus may require more bits than numbers with a larger modulus. From Figure F.1 it is also clear that for a particular complex integer, the base-p representation may require more bits than the base-2 representation does.

base-10	base-p	base-10	base-p
4	111010000	$\mathrm{j}4$	1110000
3	1101	$\mathrm{j}3$	1110111
2	1100	$\mathrm{j}2$	1110100
1	1	j	11
0	0		
-1	11101	$-\mathrm{j}$	111
-2	11100	$-\mathrm{j}2$	100
-3	10001	$-\mathrm{j}3$	110011
-4	10000	$-\mathrm{j}4$	110000
$1+\mathrm{j}$	1110	$-1+\mathrm{j}$	10
$1-\mathrm{j}$	111010	$-1-\mathrm{j}$	110

Table F.1: *Examples of base-p complex integers, with $p = -1 + \mathrm{j}$.*

base-10	base-p	base-10	base-p
$\frac{1}{8}$	0.000011	$\mathrm{j}\frac{1}{8}$	0.011101
$\frac{1}{4}$	1.1101	$\mathrm{j}\frac{1}{4}$	0.0111
$\frac{1}{2}$	1.11	$\mathrm{j}\frac{1}{2}$	0.01
$-\frac{1}{2}$	0.11	$-\mathrm{j}\frac{1}{2}$	111.01
$-\frac{1}{4}$	0.0001	$-\mathrm{j}\frac{1}{4}$	0.0011
$-\frac{1}{8}$	0.000111	$-\mathrm{j}\frac{1}{8}$	0.000001

Table F.2: *Examples of base-p scaled complex integers, with $p = -1 + \mathrm{j}$.*

For the values $a \in \mathbb{C}_{\mathbb{Z}}$ with $\|a\|_\infty \leq \Xi$ and $2^0 \leq \Xi \leq 2^{10}$, the required number of bits in base-p (top stair case) and in base-2 (lower stair case) are shown in Figure F.2. The difference of both numbers is shown in the bottom curve. From this it can be concluded that base-p representation requires 2 to 4 bits more than the base-2 representation.

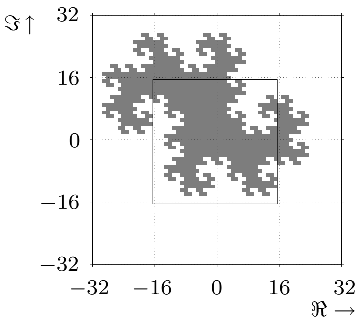


Figure F.1: *Set of 10 bit complex integers in base-$p = -1 + \mathrm{j}$ and base-2 (square).*

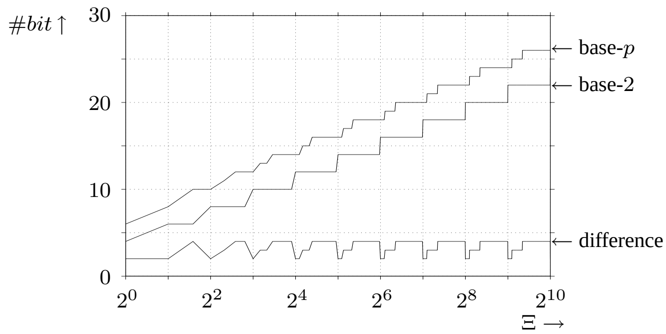


Figure F.2: *Required number of bits, and the difference, to represent any $a \in \mathbb{C}_\mathbb{Z}$ with $\|a\|_\infty \leq \Xi$ in base-$p = -1 + \mathrm{j}$ and base-2.*

F.2 Base $p = -1 + \mathrm{j}$: addition and subtraction

Next some example circuits that perform additions and subtractions are discussed. In the sequel the relatively small value $\Xi = 3$ is chosen to obtain practically sized figures. Lemma A.13 (for $\|a\|_\infty, \|b\|_\infty \leq \Xi$ holds $\|a \pm b\|_\infty \leq 2\Xi$) and Figure F.2 show that for a base-2 complex adder 6 inputs and 8 outputs are required. Similarly for a base-p adder 10 inputs and 12 outputs are required.
The traditional schemes for base-2 adders use half- and full-adder components. The so called half-adder, adds two 1-bit inputs a_i and b_i and generates the data output d_i and the carry output $c_{i,out}$ that are defined as:

$$d_i = a_i \operatorname{xor} b_i,$$
$$c_{i,out} = a_i b_i.$$

The so called full-adder, adds like the half-adder, two 1-bit inputs a_i and b_i and also the 1-bit carry $c_{i,in}$. Basically the full-adder adds three 1-bit signals. The data output d_i and the carry output $c_{i,out}$ are now defined as:

$$d_i = a_i \,\mathrm{xor}\, b_i \,\mathrm{xor}\, c_{i,in},$$
$$c_{i,out} = a_i b_i \,\mathrm{or}\, a_i c_{i,in} \,\mathrm{or}\, b_i c_{i,in}.$$

In Figure F.3 a possible well-known structure of a base-2 (2's-complement) complex adder for $\Xi = 3$ is given. Per signal part, a series of full-adders and a single half-adder is used. In total 2 extra output bits are generated.

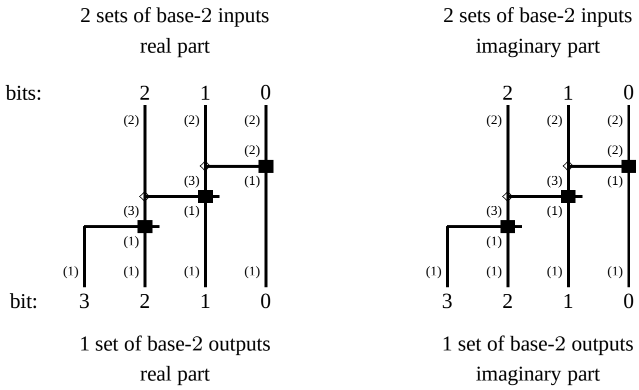


Figure F.3: *Example structure of a base-2 complex adder for $\Xi = 3$.*

To simplify the schemes in this appendix a special notation is used. Bit-lines with equal weights are grouped and only the number of bit-lines is given between parenthesis. A diamond is used to indicate that a carry output and bit-lines are grouped. For the full- and half-adder a black square is used. The full-adder is distinguished by the symbolic carry input.
For a base-2 (2's-complement) complex subtractor the structure is similar, the two half-adders are replaced by full-adders, and the b-inputs are inverted (not shown).
Adding or subtracting on a bit-by-bit bases gives 4 cases to be considered. Table F.3 shows that the bit-addition for $a_i = b_i = 1$ and the bit-subtraction for $a_i = 0$ and $b_i = 1$ produce a carry out. Note that unlike the base-2 (2's-complement) addition, the carry is not to be combined with the next higher bit, $i + 1$. Furthermore the addition produces 2 and the subtraction produces 3 carries. Due to this, the complexity of a base-p addition and subtraction is likely to be larger than for base-2.
In Figure F.4 a possible structure of a base-p adder for $\Xi = 3$ is given. A series of full-adders and half-adders is used. Here too, 2 extra output bits are generated.
Similar to the half-adder the half-subtractor is defined. The so called half-subtractor for subtracting the 1-bit input b_i from the 1-bit input a_i, generates the data output d_i and the carry output $c_{i,out}$, that are defined as:

$$d_i = a_i \,\mathrm{xor}\, b_i,$$
$$c_{i,out} = \overline{a_i} b_i.$$

		$a_i + b_i$		$a_i - b_i$	
a_i	b_i	base-10	base-p	base-10	base-p
0	0	0	0	0	0
0	1	1	1	-1	11101
1	0	1	1	1	1
1	1	2	1100	0	0

Table F.3: *Base-p bit addition and subtraction.*

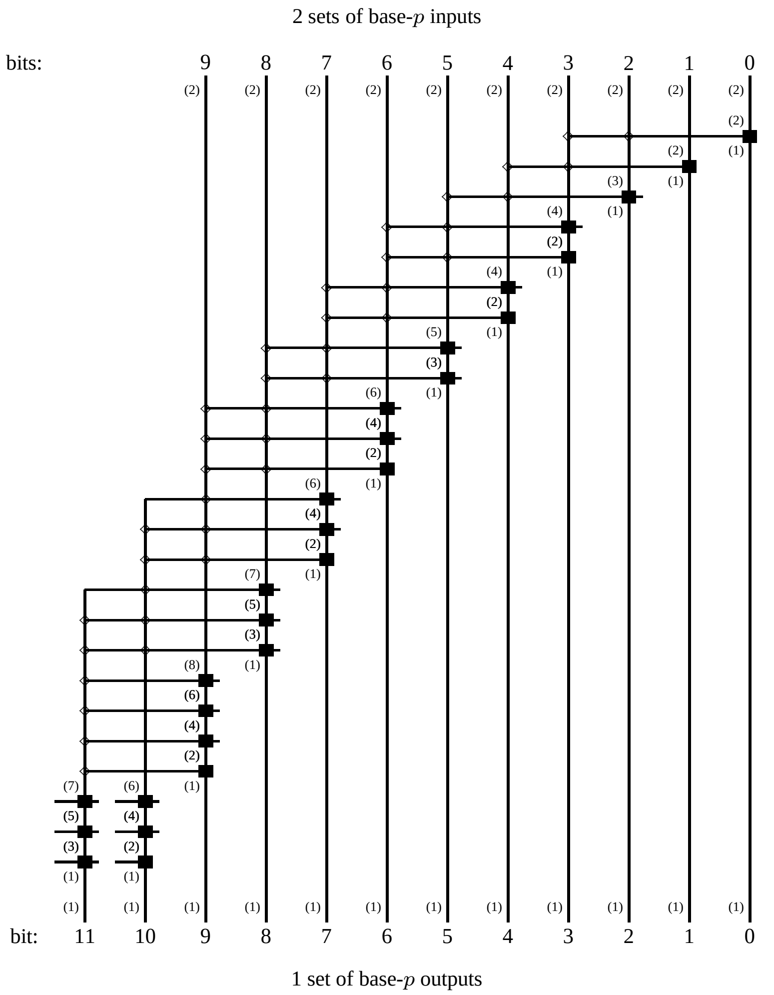


Figure F.4: *Example structure of a base-p adder for $\Xi = 3$.*

The half-subtractor is represented by a black square and a minus-sign. In Figure F.6 a possible structure of a base-p subtractor for $\Xi = 3$ is given. A series of half-subtractors, full-adders and half-adders is used. Here too 2 extra output bits are generated. The half- and full-adders in the two highest bit-lines can in principle be simplified, since the carry-outputs, $c_{i,out}$, are not used.
To enable cost comparison of base-2 and base-p additions and subtractions, the following relation between costs of half-adders, HA, full-adders, FA, and half-subtractors, HS, based on the earlier definitions, is assumed:

$$C(HA) = C(HS) = \frac{1}{2}C(FA).$$

In Figure F.5 the ratio of costs, ROC, for $2^0 \leq \Xi \leq 2^{10}$ is shown for both the adder and the subtractor. The ratio is defined as the quotient of the costs for a base-p implementation and the costs for a base-2 implementation. Clearly, the base-p adders and subtractors have higher costs than the base-2 versions. In the example schemes the value $\Xi = 3$ is used, however, typical values are in the range $2^8 \leq \Xi \leq 2^{10}$. Observe that for these Ξ-values the ROC for the adder is about 6 and for the subtractor about 11.

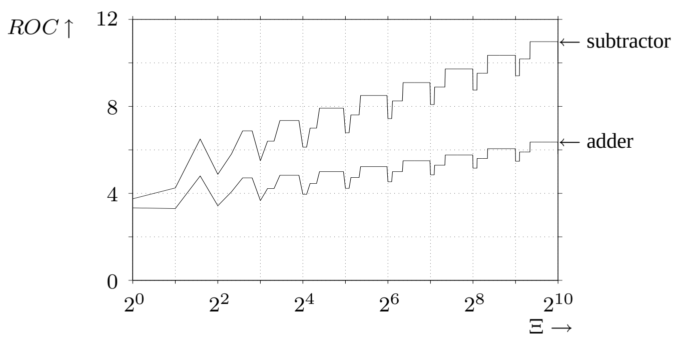


Figure F.5: *Ratio of costs, ROC: base-p versus base-2.*

F.3 Base $p = -1 + \mathrm{j}$: shift-and-add multiplication

Like for base-2, see Appendix E, the shift-and-add procedure can be used to realize the multiplication of complex-integer data and a complex-integer coefficient in base-p. In base-p only a single series of bit-shifts and additions is needed for complex-integer data and coefficients, whereas in base-2 a few series of bit-shifts and integer additions are required for the individual real and imaginary parts.

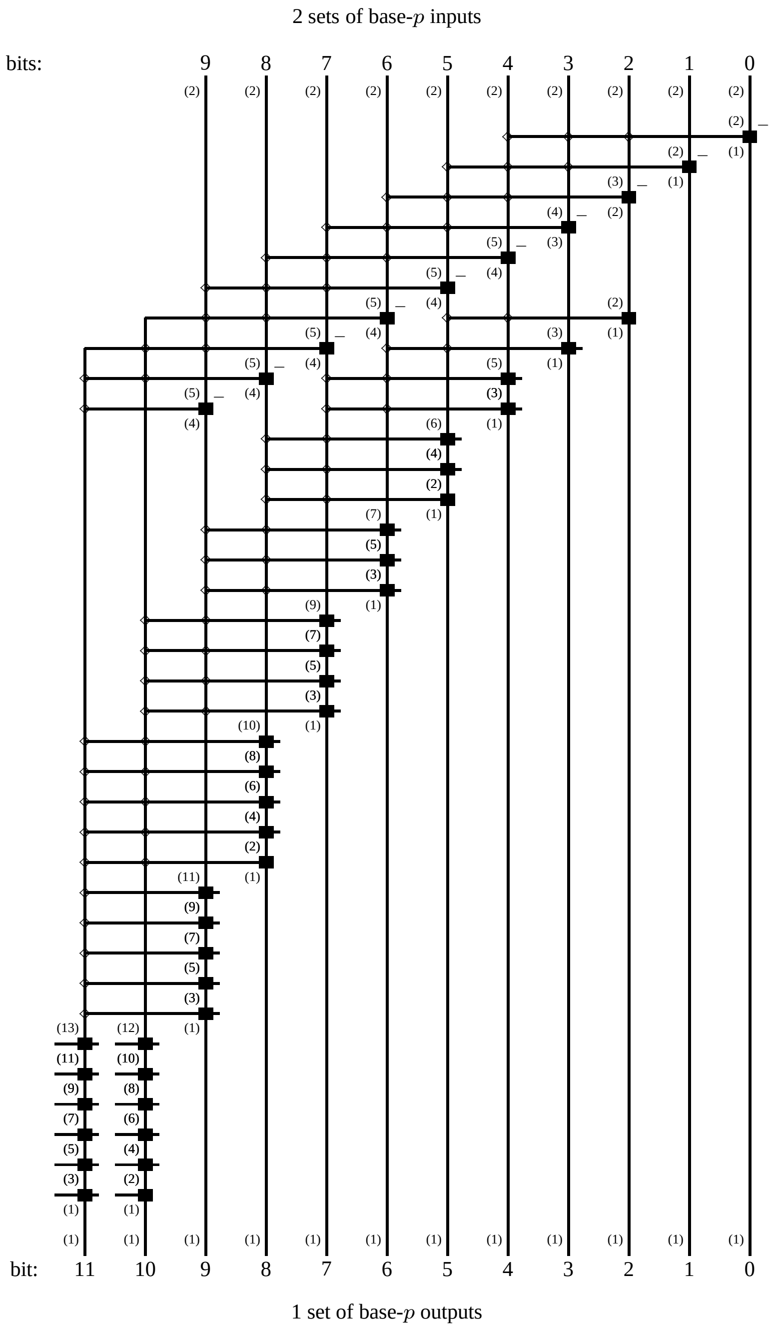


Figure F.6: *Example structure of a base-p subtractor for $\Xi = 3$.*

Next, base-2 and base-p are compared in case of multiplication by the factors 2 and p.

- Multiplication in base-2 by the factor 2 (i.e. $[0010]_2$):
  1 shift of both the real and imaginary parts over 1 position, and no additions or subtractions are required.
- Multiplication in base-2 by the factor p (i.e. $-[0001]_2 + \mathrm{j}[0001]_2$):
  no shifts are needed, and 2 additions or subtractions of individual real and imaginary parts are required in case Structure A is used. These 2 additions or subtractions compare to 1 base-2 complex addition or subtraction.
- Multiplication in base-p by the factor 2 (i.e. $[1100]_p$):
  2 shifts are needed, one over 2 and one over 3 positions, and 1 base-p addition is needed.
- Multiplication in base-p by the factor p (i.e. $[0010]_p$):
  1 shift over 1 position, and no additions are required.

The previous results are sumarized in Table F.4 where $\#shift$ and $\#add$ are the numbers of shifts and add or subtract operations respectively. Since the costs of a shift can be neglected compared to the costs of an addition, the table basically shows symmetrical entries. However, note that the base-p addition that is required for the factor 2 has about 6 times higher costs than the base-2 addition that is required for the factor p.

factor	base-2		base-p	
	$\#shift$	$\#add$	$\#shift$	$\#add$
2	1	0	2	1
p	0	1	1	0

Table F.4: *Comparison of base-2 and base-p shift-and-add multiplication.*

F.4 Conclusion

In terms of speedup [73] claims a factor 5 to 10. However, the hardware-cost performance of the complex-base number operations, as found in this appendix, is disappointing. Although no exhaustive search has been performed for low cost base-p adder and subtractor structures, it is unlikely that the ratio of costs for adders and subtractors can be reduced from values of 6 and 11 respectively to values below 1. As a consequence the costs of the shift-and-add method for base-p multiplications in general are significantly higher than for base-2. Note that in base-p the costs of a subtraction are twice the cost of an addition. Other issues like the conversion between bases will also affect the ratio of costs negatively.

Appendix G

Introduction to complex primes

Natural numbers can be factorized into primes in a unique way. This idea is applied in Appendix E to derive alternative constructions for the integer-valued coefficients. In the same appendix a similar approach is applied to complex-integer-valued coefficients that can be factorized into complex primes in a unique way.
This appendix gives a brief introduction into the field of complex primes. Searching for information about complex primes and the related factorization of complex integers, gives either a very sloppy or a very mathematical expose, e.g., [53]. Therefore this appendix presents some explanation and some examples mainly based on [88].
In Section G.1 more will be said about prime factorization and complex primes. Section G.2 and Section G.3 present methods to test complex primes and to obtain complex prime factorizations. Finally two examples are presented in Section G.4.

G.1 Primes and complex primes

A prime number p is a natural number $p \in \mathbb{N}$ and $p \geq 2$, which is only divisible by 1 or itself but not by any other natural number. According to Gauss, any non-zero natural number $x \in \mathbb{N}^+$ can be factorized uniquely into prime numbers p_i like:

$$x = \prod_i p_i^{e_i} \text{ with } e_i \in \mathbb{N}.$$

Unique factorization of the negative integers is possible also, if the extra unit factor -1 is introduced. Now any non-zero integer $x \in \mathbb{Z}\backslash\{0\}$ can be factorized uniquely into prime numbers like:

$$x = (-1)^k \prod_i p_i^{e_i} \text{ with } e_i \in \mathbb{N} \text{ and } k \in \{0, 1\}.$$

According to Gauss the definition of primes can be extended to complex primes, also called the Gaussian primes, $P \in \mathbb{C}_\mathbb{Z}$ and the unique factorization of a complex integer into complex primes. Now 4 units need to be defined, viz.: $1, \mathrm{j}, -1, -\mathrm{j}$, such that non-zero complex integer $x \in \mathbb{C}_{\mathbb{Z}} \backslash \{0\}$ can be factorized uniquely into complex prime numbers like:

$$x = \mathrm{j}^k \prod_i P_i^{e_i} \text{ with } e_i \in \mathbb{N} \text{ and } k \in \{0, 1, 2, 3\}.$$

Also, if $P \in \mathbb{C}_{\mathbb{Z}}$ is a complex prime the values $\mathrm{j}^k P$, the associates, and their conjugates $\mathrm{j}^k P^*$ are complex primes too. So if a single complex prime is found, 7 more complex primes are obtained by considering the associates and the conjugates. Except for $P = 1 + \mathrm{j}$ where the associates and the conjugates coincide, only 3 additional complex primes can be obtained.

To distinguish between the integers and complex integers explicitly in this appendix, the integers will be referred to as the natural integers. Similarly to distinguish here between the primes and the complex primes explicitly, the primes will be referred to as the natural primes.

A number which is a natural prime not necessarily is a complex prime. As an example consider the natural primes: $2 = (1 + \mathrm{j})(1 - \mathrm{j})$ and $5 = (2 + \mathrm{j})(2 - \mathrm{j})$.

Natural primes can for instance be found by the computation demanding method often referred to as the Sieve of Eratosthenes. Starting with number 2, i.e., the smallest natural prime, all multiples of 2 are discarded since they are no prime trivially. The next largest number not discarded yet, i.e., 3, is selected and now all multiples of 3 are discarded since they are no prime either. This process of discarding multiples is repeated until sufficient primes are obtained.

Basically, the complex primes can be obtained in a similar way, however, it is possible to use the natural primes to obtain the complex primes in an efficient way. The same is true for the complex factorization. The factorization of a natural number into natural primes can be used to obtain the factorization of a complex integer into complex primes. In [12] a computer program to calculate complex primes is presented.

G.2 Test for complex primes

Given a non-zero complex integer $x \in \mathbb{C}_{\mathbb{Z}} \backslash \{0\}$, the question is how to check if x is a complex prime or not. Let $x = x_\mathrm{r} + \mathrm{j}x_\mathrm{i}$, then 2 cases can be distinguished, viz. x is real or imaginary, i.e., $x \in \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$, or x has non-zero real and non-zero imaginary parts, i.e., $x \notin \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$.

Case 1: If $x \in \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$, or equivalently $x_\mathrm{r} x_\mathrm{i} = 0$, the natural number $|x|$ should be a natural prime p with $p|_4 = 3$.

Case 2: If $x \notin \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$, or equivalently $x_\mathrm{r} x_\mathrm{i} \neq 0$, the natural number xx^* should be a natural prime p with $p|_4 = 1$ or $p = 2$.

G.3 Factorization in complex primes

Given a non-zero complex integer $x \in \mathbb{C}_{\mathbb{Z}} \backslash \{0\}$, the question is how to derive the complex factorization of x. The factorization process consists of the following four steps.

Let $x = x_\mathrm{r} + \mathrm{j}x_\mathrm{i}$, then:

Step 1: Remove the common factor λ from x_r and x_i, so $y = \frac{x}{\lambda}$ with $\lambda = \gcd(x_\mathrm{r}, x_\mathrm{i})$. Next λ is factorized in Step 2, and y is factorized in Step 3. The factorization of $x = \lambda y$, is the product of factorizations of λ and y in Step 4.

Step 2: The common factor λ is a natural number that can be factorized uniquely in natural primes as $\lambda = \prod_i p_i^{e_i}$. For each of the natural primes p_i holds either $p_i = 2$, $p_i|_4 = 1$ or $p_i|_4 = 3$. For $p_i = 2$ the complex factorization is $p_i = P_i P_i^*$ with $P = (1+\mathrm{j})$. For $p_i|_4 = 1$ the unique complex factorization is $p_i = P_i P_i^*$, and for $p_i|_4 = 3$ a complex factorization is not possible.

Step 3: The natural number yy^* can also be factorized uniquely up to unity in natural primes as $yy^* = \prod_i p_i^{e_i}$. These p_i all satisfy $p_i|_4 = 1$. Each p_i can be factorized as $p_i = P_i P_i^*$. It has to be tested whether $P_i|y$ or $P_i^*|y$. Note that the value of e_i does not depend on this test. Finally the unit value can be determined by dividing y by all found factors.

Step 4: Finally, combine the factorizations from Step 2 and Step 3 to obtain the factorization of x.

G.4 Examples

Two examples will now illustrate the test for complex primes and if possible the factorization in complex primes.

Example G.1. Consider $x = 5 - \mathrm{j}2$.

Test: Since $x \notin \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$, Case 2 of the test has to be used. The value $xx^* = 29$ is a natural prime with $29|_4 = 1$.

As a consequence $5 - \mathrm{j}2$ is a complex prime and factorization is not possible.
End of example

Example G.2. Consider $x = 36 - \mathrm{j}2$.

Test: Since $x \notin \mathbb{Z} \cup \mathrm{j}\mathbb{Z}$, Case 2 of the test has to be used. The value $xx^* = 1300$ is not a natural prime. As a consequence $36 - \mathrm{j}2$ is not a complex prime and hence can be factorized.

Factorization: According to the 4 steps:

Step 1: The common factor $\lambda = \gcd(36, -2) = 2$ and consequently $y = \frac{x}{\lambda} = 18 - \mathrm{j}$.

Step 2: Since λ is a natural prime there is only one factor to consider, viz. $p = 2$. The factorization of λ now is: $\lambda = (1+\mathrm{j})(1-\mathrm{j})$.

Step 3: For $y = 18 - \mathrm{j}$ the value $yy^* = 325$ that can be factorized in natural primes as $325 = 5^2 \cdot 13^1$, with $5 = (2+\mathrm{j})(2-\mathrm{j})$ and $13 = (3+\mathrm{j}2)(3-\mathrm{j}2)$. It is easy to verify that: $(2+\mathrm{j})|y$ and $(3+\mathrm{j}2)|y$, such that up to a unity the factorization is known: $y = \mathrm{j}^k(2+\mathrm{j})^2(3+\mathrm{j}2)^1$ with $k = 3$.

Step 4: The complex prime factorization of x now is:

$$36 - \mathrm{j}2 = -\mathrm{j}(1+\mathrm{j})(1-\mathrm{j})(2+\mathrm{j})^2(3+\mathrm{j}2).$$

End of example

Bibliography

[1] E. Aarts and J.K. Lenstra (editors). *Local search in combinatorial optimization*. John Wiley & Sons Ltd., 1997.

[2] F. Ashrafzadeh, B. Nowrouzian, and A.T.G. Fuller. A novel modified branch-and-bound technique for discrete optimization over canonical signed-digit number space. In *ISCAS '98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, volume 5, pages 391–394, 1998.

[3] A. Avizienis. Signed-digit number representation for fast parallel arithmetic. *IRE Transactions on Electronic Computers*, 10:389–400, Sep 1961.

[4] M.G. Bellanger, G. Bonnerot, and M. Coudreuse. Digital filterering by polyphase network: Application to sample-rate alteration and filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24(2):109–114, Apr 1976.

[5] J. Benesty and P. Duhamel. Fast constant modulus adaptive algorithm. *IEE Proceedings F Radar and Signal Processing*, 138(4):379–387, August 1991.

[6] R. Bernstein. Multiplication by integer constants. *Software - Practice and Experience*, 16(7):641–652, Jul 1986.

[7] D. Bhattacharya and A. Antoniou. Design of equiripple FIR filters using a feedback neural network. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(4):527–531, Apr 1998.

[8] P. Boer. Optimisation of filters structures: restoration of symmetry in polyphase filters. *Internal Communication*, Feb 2002.

[9] P. Boer. Synthesis of alternatives for coefficients. *Private Communication*, May 2007.

[10] A. D. Booth. A signed binary multiplication technique. *Quart. Journ. Mech. and Applied Math.*, IV(2):236–240, 1951.

[11] N. Boullis and A. Tisserand. Some optimizations of hardware multiplication by constant matrices. In *Proceedings of the* 16^{th} *IEEE Symposium on Computer Arithmetic (ARITH'03)*. IEEE, 2003.

[12] P. Bratley. Algorithm 401: An improved algorithm to produce complex primes. *Communications of the ACM*, 13(11):693, Nov 1970.

[13] R. Bregović, T. Saramäki, Y.J. Yu, and Y.C. Lim. An efficient implementation of linear-phase FIR filters for a rational sampling rate conversion. In *Proceedings. ISCAS 2006 Kos Greece. The 2006 IEEE International Symposium on Circuits and Systems*. IEEE, May 2006.

[14] P. Briggs and T. Harvey. Multiplication by integer constants. Research Report TR03-425, Rice University, Jul 1994.

[15] A.A.M.L. Bruekers. Digital filters for TVIF, a study. Technical Report TN 1992/120, Philips Research Laboratories, Eindhoven, The Netherlands, 1992.

[16] A.A.M.L Bruekers and G.C.M. Gielis. Receiver with quadrature decimating stage, method of processing digital signals. *United States Patent*, US 5,784,414. Filed: Sep 5, 1995, Granted: Jul 21, 1998.

[17] A.A.M.L. Bruekers and A.A.C.M. Kalker. Multirate filter as well as display system and mobile telephone comprising said multirate filter. *International Patent Application Publication*, WO 2004/054103. Filed: Dec 6, 2002, Published: Jun 24, 2004.

[18] A.A.M.L. Bruekers and A.A.C.M. Kalker. Multirate filter as well as display system and mobile telephone comprising said multirate filter. *United States Patent Application Publication*, US 2006/0056553. Priority: Dec 6, 2002, Filed: Dec 4, 2003, Published: Mar 16, 2006.

[19] A.A.M.L Bruekers and A.J. Rijnberg. Filter devices and methods. *United States Patent*, US 6,792,444. Filed: Nov 30, 2001, Granted: Sep 14, 2004.

[20] A.A.M.L. Bruekers and A.W.M. van den Enden. Efficient structures for increasing and decreasing the sampling rate. Technical Report TN 1990/292, Philips Research Laboratories, Eindhoven, The Netherlands, 1990.

[21] A.A.M.L. Bruekers and A.W.M. van den Enden. Filter design for the digital channel divider (DCD). Technical Report TN 1992/191, Philips Research Laboratories, Eindhoven, The Netherlands, 1992.

[22] A.A.M.L. Bruekers and A.W.M. van den Enden. New networks for perfect inversion and perfect reconstruction. *Selected areas in communications, IEEE journal on*, 10(1):130–137, Jan 1992.

[23] A.A.M.L. Bruekers and A.W.M. van den Enden. On the design of linear phase FIR filters: a bug in the design method as used in e.g. the Parks and McClellan program. Technical Report TN 1992/330, Philips Research Laboratories, Eindhoven, The Netherlands, 1992.

[24] A.A.M.L. Bruekers and A.W.M. van den Enden. Some identities of time varying, mono- and multirate systems. Technical Report TN 1992/119, Philips Research Laboratories, Eindhoven, The Netherlands, 1992.

[25] A.A.M.L. Bruekers and A.W.M. van den Enden. Fast algorithms for the discrete cosine, discrete sine and discrete W-transform. Technical Report TN 1994/258, Philips Research Laboratories, Eindhoven, The Netherlands, 1994.

[26] A.A.M.L. Bruekers and A.W.M. van den Enden. Cascade and parallel CSD notation and numbers. Technical Report TN 1996/090, Philips Research Laboratories, Eindhoven, The Netherlands, 1996.

[27] A.A.M.L. Bruekers and A.W.M. van den Enden. Interpolated FIR filters and frequency masking filters. Technical Report TN 1996/091, Philips Research Laboratories, Eindhoven, The Netherlands, 1996.

[28] A.A.M.L. Bruekers and A.W.M. van den Enden. Principles used in the FIR filter design programs DESIGN, DESIGN_VPD, PHADES, EVALT and EVALC. Technical Report TN 1996/016, Philips Research Laboratories, Eindhoven, The Netherlands, 1996.

[29] A.A.M.L. Bruekers and A.W.M. van den Enden. Variable phase delay (VPD) filters. Technical Report TN 1996/015, Philips Research Laboratories, Eindhoven, The Netherlands, 1996.

[30] A.A.M.L Bruekers and R.J. van der Vleuten. Prediction on data in a transmission system. *United States Patent*, US 6,285,301. Filed: Mar 16, 1999, Granted: Sep 4,2001.

[31] D. Burnside and T.W. Parks. Accelerated design of FIR filters in the complex domain. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 81–84. ICASSP, 1993.

[32] D. Burnside and T.W. Parks. Optimal minimax two-dimensional FIR design using a multiple simplex exchange. In *ISCAS '94. IEEE International Symposium on Circuits and Systems*, volume 2, pages 593–596, 1994.

[33] D. Burnside and T.W. Parks. Optimal design of FIR filters with the complex Chebyshev error criteria. *IEEE Transactions on Signal Processing*, 43(3):605–616, Mar 1995.

[34] R.K. Cavin, C.H. Ray, and V.T. Rhyne. The design of optimal convolutional filters via linear programming. *IEEE Transactions on Geoscience Electronics*, GE-7(3):142–145, Jul 1969.

[35] C. Charalambous and M.F. Abusharkh. An efficient method for designing optimal 2-D FIR digital filters with finite wordlength. In *IEEE International Symposium on Circuits and Systems*, volume 1, pages 464–467, 1991.

[36] Chao-Liang Chen and Jr Willson, A.N. A trellis search algorithm for the design of FIR filters with signed-powers-of-two coefficients. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(1):29–39, Jan 1999.

[37] X. Chen and T.W. Parks. Design of IIR filters in the complex domain. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1443–1446. ICASSP, 1988.

[38] X. Chen and T.W. Parks. Design of IIR filters in the complex domain. *IEEE Transactions on Acoustics, Speech and Signal Processing [see also IEEE Transactions on Signal Processing]*, 38(6):910–920, Jun 1990.

[39] Ying-Jui Chen, S. Oraintara, T.D. Tran, K. Amaratunga, and T.Q. Nguyen. Multiplierless approximation of transforms with adder constraint. *IEEE Signal Processing Letters*, 9(11):344–347, Nov 2002.

[40] Nam Ik Cho and Sang Uk Lee. Optimal design of finite precision FIR filters using linear programming with reduced constraints. *IEEE Transactions on Signal Processing*, 46(1):195–199, Jan 1998.

[41] T.A.C.M. Claasen and W.F.G Mecklenbräuker. On the transposition of linear time-varying discrete-time networks and its application to multirate digital systems. *Philips Journal of Research*, 33(1/2):78–102, 1978.

[42] J.O. Coleman. Cascaded coefficient number systems lead to FIR filters of striking computational efficiency. In *The* 8^{th} *IEEE International Conference on Electronics, Circuits and Systems. ICECS 2001.*, volume 1, pages 513–516. IEEE, Sep 2001.

[43] R.E. Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[44] F. de Dinechin and V. Lefèvre. Constant multipliers for fpgas. Research Report 2000-18, Laboratoire de l'informatique du Parallélisme (INRIA), Lyon, May 2000.

[45] A.G. Dempster and M.D. Macleod. Constant integer multiplication using minimum adders. *IEE Proceedings Circuits, Devices and Systems*, 141(5):407–413, Oct 1994.

[46] A.G. Dempster and M.D. Macleod. Use of minimum-adder multiplier blocks in FIR digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(9):569–577, Sep 1995.

[47] A.G. Dempster and M.D. Macleod. Generation of signed-digit representations for integer multiplication. *IEEE Signal Processing Letters*, 11(8):663–665, Aug 2004.

[48] Zinnur Doğanata and P.P. Vaidyanathan. On one-multiplier implementations of FIR lattice structures. *IEEE Transactions on Circuits and Systems*, 34(12):1608–1609, Dec 1987.

[49] G. Evangelista. Least mean squared error-design of complex FIR filters with quantized coefficients. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(8):778–784, Aug 2001.

[50] S.H. Galal, H.F. Ragaie, and M.S. Tawfik. RC sequence asymmetric polyphase networks for RF integrated transceivers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(1):18–27, Jan 2000.

[51] G.C. Gielis, R. van de Plassche, and J. van Valburg. A 540-MHz 10-b polar-to-Cartesian converter. *IEEE Journal of Solid-State Circuits*, 26(11):1645–1650, Nov 1991.

[52] M.J. Gingell. Single sideband modulation using sequence asymmetric polyphase networks. *Electrical Communication*, 48(1 and 2):21–25, 1973.

[53] G.H. Hardy and E.M. Wright. *An Introduction to the theory of numbers*. Oxford University Press, 1979.

[54] R.I. Hartley. Subexpression sharing in filters using canonic signed digit multipliers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(10):677–688, Oct 1996.

[55] R.J. Hartnett and G.F. Boudreaux-Bartels. Improved filter sharpening. *IEEE Transactions on Signal Processing*, 43(12):2805–2810, Dec 1995.

[56] O. Herrmann, L.R. Rabiner, and D.S.K. Chan. Practical design rules for optimum finite impulse response low-pass digital filters. *The Bell System Technical Journal*, 52:769–799, Jul-Aug 1973.

[57] R.M. Hewlitt and Jr. Swartzlantler, E.S. Canonical signed digit representation for FIR digital filters. In *SiPS 2000, IEEE Workshop on Signal Processing Systems*, pages 416–426, 2000.

[58] J.V. Hu and L.R. Rabiner. Design techniques for two-dimensional digital filters. *IEEE Transactions an Audio and Electroacoustics*, AU-20(4):249–257, Oct 1972.

[59] J.A. Huisken, F.A.M. van de Laar, M.J.G. Bekooij, G.C.H. Gielis, P.W.F. Gruijters, and F.P.J. Welten. A power-efficient single-chip OFDM demodulator and channel decoder for multimedia broadcasting. *IEEE Journal of Solid-State Circuits*, 33(11):1793–1798, Nov 1998.

[60] K. Ichige, M. Iwaki, and R. Ishii. Accurate estimation of minimum filter length for optimum FIR digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(10):1008–1016, Oct 2000.

[61] M. Ikehara, M. Okuda, K. Kiyose, and S. Tikahashi. Complex approximation for FIR filters with equiripple magnitude and phase response. In *Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on Circuits and Systems*, volume 4, pages 73–76. IEEE, 2000.

[62] A.H.J. Immink. *System and receiver design for two-dimensional optical storage*. PhD thesis, Technical University Eindhoven, 2005.

[63] L.B. Jackson. On the interaction of roundoff noise and dynamic range in digital filters. *The Bell System Technical Journal*, 49(2):159–184, Feb 1970.

[64] A.G. Jaffer and W.E. Jones. Constrained least-squares design and characterization of affine phase complex FIR filters. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, 1993,* volume 1, pages 685–691, Nov 1993.

[65] A.G. Jaffer and W.E. Jones. Weighted least-squares design and characterization of complex FIR filters. *IEEE Transactions on Signal Processing [see also IEEE Transactions on Acoustics, Speech, and Signal Processing]*, 43(10):2398–2401, Oct 1995.

[66] T. Jamil. The complex binary number system. *Potentials of the IEEE*, 20(5):39–41, Dec 2001.

[67] T. Jamil, N. Holmes, and D. Blest. Towards implementation of a binary number system for complex numbers. In *Proceedings of IEEE SoutheastConn Nashville, Tennessee*, pages 268–274, Apr 2000.

[68] B. Jaumard, M. Minoux, and P. Siohan. Finite precision design of FIR digital filters using a convexity property. *IEEE Transactions on Acoustics, Speech and Signal Processing [see also IEEE Transactions on Signal Processing]*, 36(3):407–411, Mar 1988.

[69] J.F. Kaiser and R.W. Kamming. Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-25(5):415–422, Oct 1977.

[70] J.F. Kaiser and K. Steiglitz. Design of FIR filters with flatness constraints. In *IEEE Int. Conference Acoustics, Speech and Signal processing, Boston M.A.*, pages 197–200, Apr 1983.

[71] D. Karaboga, D.H. Horrocks, N. Karaboga, and A. Kalinli. Designing digital FIR filters using tabu search algorithm. In *ISCAS '97., Proceedings of 1997 IEEE International Symposium on Circuits and Systems*, volume 4, pages 2236–2239, 1997.

[72] S. Khmelnik. A method and system for processing complex numbers. *International Patent Application*, WO 01/50332. Priority: Jan 5, 2000, Filed: Jan 5, 2001, Published: Jul 12, 2001.

[73] S. Khmelnik and S. Selyutin. Arithmetic unit for complex number processing. *http://www.iproscorp/article26_e.pdf*, pages 1–14, 2002.

[74] D.E. Knuth. An imaginary number system. *Communications of the ACM*, pages 245–247, 1960.

[75] D.M. Kodek. Design of optimal finite wordlength FIR digital filters using integer linear programming techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28(3):304–308, Jun 1980.

[76] M. Lang. *Algorithms for the constrained design of digital filters with arbitrary magnitude and phase responses*. PhD thesis, Vienna University of Technology, June 1999.

[77] I.H. Lee, S.S. Wu, and A.Y. Wu. Cost-efficient multiplier-less FIR filter structure based on modified décor transformation. In *International Conference on Acoustics, Speech, and Signal Processing*. ICASSP, 2001.

[78] V. Lefèvre. Multiplication by an integer constant. Research Report 1999-06, Laboratoire de l'informatique du Parallélisme (INRIA), Lyon, Jan 1999.

[79] V. Lefèvre. Multiplication by an integer constant. Research Report 4192, Institut National de Recherche en Informatique et en Automatique (INRIA), Villers-Lès-Nancy, May 2001.

[80] V. Lefèvre. Multiplication by an integer constant: Lower bounds on the code length. Research Report 4493, Institut National de Recherche en Informatique et en Automatique (INRIA), Villers-Lès-Nancy, Jul 2002.

[81] W. Lertniphonphun and J.H. McClellan. Complex frequency response FIR filter design. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1301–1304, 1998.

[82] Dongning Li. Minimum number of adders for implementing a multiplier and its application to the design of multiplierless digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(7):453–460, Jul 1995.

[83] Y.C. Lim. Frequency-response masking approach for the synthesis of sharp linear phase digital filters. *IEEE Transactions on Circuits and Systems*, 33(4):357–364, Apr 1986.

[84] Y.C. Lim and A.G. Constantinides. Linear phase FIR digital filter without multipliers. *Proceedings ISCAS 1979, IEEE International Symposium on Circuits and Systems*, pages 185–188, 1979.

[85] Y.C. Lim and Y. Lian. The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 40(2):88–95, Feb 1993.

[86] Y.C. Lim and S.R. Parker. FIR filter design over a discrete powers-of-two coefficient space. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(3):583–591, Jun 1983.

[87] Marcos Martínez-Peiró, Eduardo I. Boemo, and Lars Wanhammar. Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm. *IEEE Transactions on Circuits and Systems II*, 49(3):196–203, Mar 2002.

[88] S. Maubach. Complex primes. *Private Communication*, Sep 2004.

[89] J.H. McClellan, T.W. Parks, and L.R. Rabiner. *FIR Linear Phase Filter Design Program*, pages 5.1–1 – 5.1–13. IEEE Press, John Wiley & Sons, 1979.

[90] C. Mead and L. Conway. *Introduction to VLSI systems*. Addison-Wesley Publishing Company, Inc., 1980.

[91] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC Press, Inc., 1997.

[92] B.J. Minnis, P.A. Moore, A.W. Payne, A.C. Caswell, and M.E. Barnard. A low-IF polyphase receiver for GSM using log-domain signal processing. In *Digest of Papers. Radio Frequency Integrated Circuits (RFIC) Symposium*, pages 83–86. IEEE, 2000.

[93] G.E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), Apr 1965.

[94] Zhi-Jian (Alex) Mou. Novel use of symmetries in linear phase digital filters. In *Proceedings. ICASSP 1992 San Francisco. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, Mar 1992.

[95] Zhi-Jian (Alex) Mou. Symmetry exploitation in digital interpolators / decimators. *IEEE Transactions on Signal Processing*, 44(10):2611–2615, Oct 1996.

[96] V. Nagesha. On computing the discrete Fourier transform of a linear phase sequence. *IEEE Transactions on Signal Processing [see also IEEE Transactions on Acoustics, Speech, and Signal Processing]*, 39(1):225–226, Jan 1991.

[97] M. Narasimhan and J. Ramanujam. Improving the computational performance of ILP-based problems. In *International Conference on Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers*, pages 593–596. IEEE/ACM, 1998.

[98] Y. Neuvo, D. Cheng-Yu, and S. Mitra. Interpolated finite impulse response filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(3):563–570, Jun 1984.

[99] T.Q. Nguyen. The eigenfilter for the design of linear-phase filters with arbitrary magnitude response. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1981–1984. ICASSP, Apr 1991.

[100] A.M. Nielsen. *Number systems and digit serial arithmetic*. PhD thesis, Odense University Denmark, June 1997.

[101] A.H.H.J. Nillesen. Non-integral delay circuit. *United States Patent*, US 5,349,548. Filed: May 13, 1993, Granted: Sep 20, 1994.

[102] A.H.H.J. Nillesen. Non-integral delay circuit. *United States Patent*, US 5,625,581. Filed: Nov 16, 1994, Granted: Apr 29, 1997.

[103] A.H.H.J. Nillesen. FIR-filter with variable phase delay (VPD FIR-filter). *Internal Communication*, Feb 1992.

[104] A.H.H.J. Nillesen. Video interpolation with VPD FIR-filter. *Internal Communication*, Mar 1993.

[105] T. Nillesen, F. Bruekers, and A. van den Enden. Workshop on the FIR-filter design program DESIGN. Technical Report 1994-02-17, Philips Research Laboratories, Eindhoven, The Netherlands, 1994.

[106] S. Nordebo and Zhuquan Zang. Application of infinite dimensional linear programming to FIR filter design with time domain constraints. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1297–1300, 1998.

[107] B. Nowrouzian, A.T.G. Fuller, and F. Ashrafzdeh. Constrained min-max optimization of finite-precision multi-rate digital filters over the canonical signed-digit coefficie. In *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, volume 2, pages 683–686, 1997.

[108] Woo Jin Oh and Yong Hoon Lee. Implementation of programmable multiplierless FIR filters with powers-of-two coefficients. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(8):553–556, Aug 1995.

[109] M. Oner. A genetic algorithm for optimisation of linear phase FIR filter coefficients. In *Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers*, volume 2, pages 1397–1400, 1998.

[110] D. Pal. Low power parallel multiplier for complex numbers. *United States Patent*, US 5,694,349. Filed: Mar 29, 1996, Granted: Dec 2, 1997.

[111] T.W. Parks and J.H. McClellan. A program for the design of linear phase finite impulse response digital filters. *IEEE Transactions on Audio and Electroacoustics*, AU-20(3):195–199, Aug 1972.

[112] W. Penney. A binary system for complex numbers. *Journal of the ACM*, 12(2):247–248, Apr 1965.

[113] E.J. Pol and P. Boer. More alternatives for larger integers. *Private Communication*, Dec 2007.

[114] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: the art of scientific computing*. Cambridge university press, 1999.

[115] L.R. Rabiner. Linear program design of finite impulse response (FIR) digital filters. *IEEE Transactions on Audio and Electroacoustics*, AU-20(4):280–288, Oct 1972.

[116] L.R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[117] J. Radecki, J. Konrad, and E. Dubois. Design of multidimensional finite-wordlength FIR and IIR filters by simulated annealing. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(6):424–431, Jun 1995.

[118] J.B. Rault, Y.F. Dehery, J.Y. Roudaut, A.A.M.L. Bruekers, and R.N.J. Veldhuis. Digital transmission system using subband coding of a digital signal. *United States Patent*, US 5,214,678. Filed: May 31, 1990, Granted: May 25, 1993.

[119] D. Reefman, A.A.M.L. Bruekers, F.L.A.J. Kamperman, and A.A.M. Staring. Watermark insertion and extraction into and from a high quality signal is performed using sample rate conversion. *United States Patent*, US 6,690,812. Filed: May 15, 2001, Granted: Feb 10, 2004.

[120] G.W. Reitwiesner. Binary arithmetic. In F.L. Alt, editor, *Advances in computers*, volume 1, pages 231–308. Academic Press, New York, London, 1960.

[121] A.J. Rijnberg, R. Taori, and A.A.M.L. Bruekers. Generating coefficients for a prediction filter in an encoder. *United States Patent*, US 7,224,747. Filed: Jan 5, 2001, Granted: May 29, 2007.

[122] H. Samueli. An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients. *IEEE Transactions on Circuits and Systems*, 36(7):1044–1047, Jul 1989.

[123] H.W. Schüßler. *Digitale Signalverarbeitung 1*. Springer-Verlag, 1994.

[124] A. Sempel, E.F. Stikvoort, A.A.M.L Bruekers, A.W.M. van den Enden, R.J. van de Plassche, and G.C.M. Gielis. Reception of modulated carriers having asymmetrical sidebands. *United States Patent*, US 6,324,233. Filed: Jul 22, 1998, Granted: Nov 27, 2001.

[125] P. Siohan and A. Benslimane. Finite precision design of optimal linear phase 2-D FIR digital filters. *IEEE Transactions on Circuits and Systems*, 36(1):11–22, Jan 1989.

[126] P. Siohan and C. Roche. Optimal design of 1-D and 2-D FIR multiplierless filters. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2877–2880. ICASSP, 1991.

[127] Young-Seog Song and Yong Hoon Lee. Design of sparse FIR filters based on Branch-and-Bound algorithm. In *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, volume 2, pages 1445–1448, 1997.

[128] K. Steiglitz, T.W. Parks, and J.F. Kaiser. METEOR: a constraint-based FIR filter design program. *IEEE Transactions on Signal Processing*, 40(8):1901–1909, Aug 1992.

[129] E. Stikvoort. Analoge polyphasefilters. *Tijdschrift van het NERG*, 68(2):84–92, 2003.

[130] R.L. Streit and A.H. Nuttall. A general Chebyshev complex function approximation procedure and an application to beamforming. *Journal Acoustic Society of America*, 72(1):181–190, 1982.

[131] Chen-Cheng Tseng and Su-Ling Lee. Minimax design of stable IIR digital filter using iterative linear programming method. In *Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on Circuits and Systems*, volume 4, pages 89–92. IEEE, 2000.

[132] P.P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial. *Proceedings of the IEEE*, 78(1):56–93, Jan 1990.

[133] P.P. Vaidyanathan. *Multirate systems and filter banks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[134] R.J. van de Plassche, A.A.M.L. Bruekers, and G.C.M. Gielis. Multi-standard reception. *United States Patent*, US 6,643,502. Filed: Jul 23, 1998, Granted: Nov 4, 2003.

[135] A.W.M. van den Enden. *Efficiency in multirate and complex digital signal processing*. PhD thesis, Technical University Eindhoven, September 2001.

[136] A.W.M. van den Enden and N.A.M. Verhoeckx. *Discrete-Time Signal Processing: An Introduction*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[137] W. Verhaegh. Efficiently generating all different factorizations. *Private Communication*, Mar 1994.

[138] I.M. Vinograd (editor in chief). *Encyclopaedia of mathematics*. Kluwer academic publishers, 1989.

[139] M. Vlček, P. Zahradnik, and R. Unbehauen. Analytical design of FIR filters. In *Proceedings of ICECS ’99. The 6th IEEE International Conference on Electronics, Circuits and Systems*, volume 1, pages 71–74, 1999.

[140] M. Yagyu, M. Shiratori, and A. Nishihara. Design of linear phase FIR digital filters using minimal number of adders and subtractors. In *IEEE Asia Pacific Conference on Circuits and Systems*, pages 286–289, 1996.

[141] Y.J. Yu. *Multiplierless multirate FIR filter design and implementation*. PhD thesis, National University of Singapore, 2003.

[142] Y.J. Yu, Y.C. Lim, and T. Saramäki. Restoring coefficient symmetry in polyphase implementation of linear phase FIR filters. *Circuits, Systems, and Signal Processing*, 25(2):253–264, Apr 2006.

Samenvatting

Symmetrie en Efficiëntie bij Complexe FIR Filters

De belangrijkste bijdrage van dit proefschrift is een reeks van nieuwe methodes voor het ontwerpen van symmetrische en efficiënte complexe FIR filters, zoals: i) het reduceren over de complexe geheeltallige coëfficiënten van gegeneraliseerde-Hermitisch-symmetrische filters tot Hermitisch-symmetrische filters, ii) de introductie van alternatieve structuren voor complexe filters, en iii) een algemeen toepasbaar recept voor het herstellen van symmetrie in 'multirate' (meer dan één bemonsteringsfrequentie) polyphase filter structuren.

Hoofdstuk 1: Inleiding Op het gebied van de digitale signaalbewerking ('Digital Signal Processing' of 'DSP'), spelen filters een belangrijke rol. Digitale filters die bijvoorbeeld gebruikt worden in radiozenders en -ontvangers, werken met een zeer hoge bemonsteringsfrequentie. Voor deze belangrijke categorie filters is efficiëntie cruciaal. Toepassing van filters met een verschillend gedrag voor positieve en negatieve frequenties is in veel gevallen voordelig, bijvoorbeeld in 'multirate' systemen. In zulke filters zijn sommige coëfficiënten complex. Dit proefschrift richt zich in het bijzonder op methodes ter verbetering van de efficiëntie van symmetrische filters. Eindige-lengte impulsresponsie ('Finite Impulse Response' of 'FIR') filters met een symmetrische impulsresponsie hebben een lineaire-fase-frequentieresponsie.
Dit inleidende hoofdstuk geeft het verhaal achter de titel van dit proefschrift en schetst het gebied van DSP in zijn algemeenheid en van het ontwerpen van digitale filters in het bijzonder. Ook het belang van de complexe filters wordt toegelicht. De inspiratie voor het schrijven van dit proefschrift is ontstaan uit de ervaringen, opgedaan bij de ontwikkeling en het gebruik van het DESFIL softwarepakket voor filterontwerp. Van DESFIL wordt daarom enige achtergrondinformatie gegeven. Veel van de resultaten die in dit proefschrift worden gepresenteerd, kunnen gebruikt worden in toekomstige versies van pakketten voor filterontwerp zoals DESFIL. Daarna wordt de achtergrond gegeven van de drie hoofdonderzoeksvragen die in dit proefschrift aan de orde komen. Deze vragen zijn de volgende.

Is het van belang om gegeneraliseerde-Hermitisch-symmetrische filters te ontwerpen?
Welke structuren implementeren gegeneraliseerde-Hermitisch-symmetrische filters?
Is het mogelijk om de symmetrie te herstellen in polyphase filter structuren?

Tot slot wordt het overzicht van dit proefschrift gepresenteerd en worden enige notaties geïntroduceerd.

Hoofdstuk 2: Symmetrische filters Vanwege hun lineaire-fase eigenschap vormen symmetrische filters een interessante klasse van FIR-filters. Bovendien kunnen symmetrische FIR-filters efficiënt geïmplementeerd worden. Niet-symmetrische FIR filters komen beknopt aan de orde in Hoofdstuk 4 en Hoofdstuk 5. In dit hoofdstuk wordt de klassieke definitie van Hermitische symmetrie uitgebreid naar een meer algemene definitie die ook geschikt is voor complexe filters, gegeneraliseerde-Hermitische oftewel (σ, μ)-symmetrie, waarbij σ de *vorm van symmetrie* en μ het *centrum van symmetrie* is, met $|\sigma| = 1$, $\sigma \in \mathbb{C}$ en $\mu \in \mathbb{Z}/2$. De relevantie van deze nieuwe definitie, die een gelijke behandeling van filters met een even en oneven lengte mogelijk maakt, wordt uitvoerig aangetoond. Diverse interessante eigenschappen die gebruikt worden in volgende hoofdstukken, worden in dit hoofdstuk gepresenteerd en afgeleid. Verder wordt er speciale aandacht besteed aan symmetrische filters met eindige-precisie coëfficiënten. Voor deze filters worden nieuwe theorema's met betrekking tot het reduceren van willekeurige (σ, μ)-symmetrische FIR-filters tot $(1, \mu)$- of (j, μ)-symmetrische filters gepresenteerd. Op basis van deze theorema's is een procedure ontworpen voor het reduceren van dergelijke (σ, μ)-symmetrische filters. De mogelijke besparingen in rekenkundige kosten door het toepassen van de reductieprocedure worden in detail besproken.

Hoofdstuk 3: Eerste- en tweede-orde-filters Voorbeelden van eenvoudige filters zijn FIR-filters van een lage orde. Voor de eerste- en tweede-orde FIR-filters wordt bekeken wat de mogelijkheden zijn om de transmissienulpunten in het z-vlak te plaatsen, voor het geval dat de coëfficiëntwaarden een beperkt bereik hebben. Verder wordt aangetoond dat de nieuw gedefinieerde (j, μ)-symmetrische complexe filters voordelen kunnen hebben boven de $(1, \mu)$-symmetrische complexe filters, afhankelijk van de gegeven specificatie.

Hoofdstuk 4: Transversale en complexe structuren De transversale filterstructuur is een van de vele mogelijke structuren voor zowel symmetrische als niet-symmetrische FIR-filters. Belangrijke eigenschappen van deze structuur zijn: i) coëfficiënten zijn identiek aan de waarden van de impulsresponsie, ii) de coëfficiënten zijn invariant onder de polyphase decompositie voor 'multirate' filters, en iii) pipelining kan op triviale wijze toegepast worden. Natuurlijk kunnen deze transversale structuren op hun beurt deel uitmaken van een samengestelde structuur.

Met als doel filterstructuren kostenefficiënt te maken, toont dit hoofdstuk hoe (σ, μ)-symmetrie verschijnt in de transversale structuur en hoe die benut kan worden. Dan volgt een overzicht van enkele bekende structuren en van structuren die geïnspireerd zijn door de nieuwe definitie voor symmetrie. Indien twee filters gemeenschappelijke ingangen of uitgangen hebben, bestaan er interessante structuren. Diverse alternatieven om complexe filters of coëfficiënten te splitsen in hun individuele reële en imaginaire delen worden besproken en in detail vergeleken. Ook nieuwe structuren om geconjugeerde coëfficiënten efficiënt te combineren worden in het vergelijk betrokken.

Hoofdstuk 5: Polyphase structuren Een van de belangrijkste concepten bij 'multirate' filteren is de polyphase decompositie en de nauw gerelateerde polyphase filter structuur. Dit concept maakt een efficiënte realisatie van interpolerende en decimerende filters mogelijk. Echter toepassing van deze decompositie op lineaire-fase-filters zal in veel gevallen de symmetrie tenietdoen die benut had kunnen worden om de rekenkundige kosten te verminderen zoals is besproken in het vorige hoofdstuk.
Centraal in dit hoofdstuk is het herstel van symmetrie in polyphase structuren. Volgens een nieuw theorema kan elk reëel of complex 'multirate' (σ, μ)-symmetrisch filter met geheeltallige of rationale interpolatie- of decimatie-factoren geconstrueerd worden met gebruikmaking van symmetrische filters in een polyphase structuur. Een mogelijke procedure om de symmetrie te herstellen, wordt gegeven en verder toegepast in diverse voorbeelden om zijn waarde te tonen.

Hoofdstuk 6: Conclusies In dit laatste hoofdstuk worden de drie hoofdonderzoeksvragen beantwoord en wordt een opsomming gegeven van mogelijk interessante onderwerpen voor toekomstig onderzoek.

Appendices Een scala van appendices ondersteunt de discussies en analyses in de kern van dit proefschrift. Eerst is er een verzameling van veelgebruikte equivalenties en hun bewijs betreffende 'multirate' en complexe systemen, gevolgd door beknopte inleidingen in: pipelining, analoge polyphase filters en het algoritme van Euclides. Verder worden interessante alternatieve constructies voor de realisatie van vermenigvuldigingen met geheeltallige en complexe-geheeltallige coëfficiënten besproken samen met diverse voorbeelden. Tot slot worden getallen met een complexe basis en de complexe priemgetallen beknopt geïntroduceerd.

Biography

Fons Bruekers was born on September 23, 1957 in Nederweert, The Netherlands. After graduating from the HAVO at 'Philips van Horne Scholengemeenschap', Weert, in 1975, he moved to the HTS at 'Instituut voor Hoger Beroeps Onderwijs', Eindhoven, were he received a BSc degree in electrical engineering, in 1979. As a conscript in the period from 1979 to 1981 he participated in a project on the development of a mobile digital communication network.

Since 1981, he has been working at Philips Research Laboratories, Eindhoven, where he was engaged in several research programmes, such as local area networks, filterbanks, filter design tools, lossless audio coding, audio watermarking, biometrics, and safety. In the early nineties the work on filter design tools resulted, amongst others, in the DESFIL software package that offered very advanced design options. He contributed to many courses on filter design in general and on the application of DESFIL in particular. Today, the tool is still in use in many laboratories inside and outside Philips. DESFIL was an important source of inspiration for the issues treated in this thesis.

Since 2001, he has been a principal scientist at the Philips Research Laboratories. He has (co)authored over 15 publications and holds over 35 granted US patents in various fields.

Index

T

U

V

Z