Efficient Wideband Digital Front-End Transceivers for Software Radio Systems

A Thesis
Presented to
The Academic Faculty

by

Wajih A. Abu-Al-Saud

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
April 2004

Efficient Wideband Digital Front-End Transceivers for Software Radio Systems

Approved by:

Professor Gordon L. Stüber, Adviser

Professor H. Venkateswaran
College of Computing

Professor Mark A. Clements

Professor John R. Barry

Professor Ye (Geofferey) Li

Date Approved: April 6, 2004

To my parents who never lacked the support, encouragement, and prayers, and to my sweetheart Abeer who gave me so much love and showed a lot of patience.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Gordon L. Stüber, for his insightful guidance and helpful comments. His hard work and dedication made me a better student and researcher.

I would also like to thank the committee members of my proposal and thesis defenses, Prof. Mark Clements, Prof. John Barry, Prof. Ye Li, Prof. H. Venkateswaran from the College of Computing, and Prof. Paul Steffes. Their comments and suggestions produced a significantly better thesis.

I would like to thank King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, Saudi Arabia for supporting and funding my Ph.D. study. I would also like to thank all the faculty of the Electrical Engineering Department at KFUPM, especially the chair of the EE Department Prof. Jamil Bakhashwain, the dean of the College of Engineering Prof. Samir Al-Baiyat, my M.Sc. advisor Dr. Samy El-Hennawey, and co-advisor Dr. Mohamed Boraie, who set me on the right path to success.

I would also like to thank my colleagues in the Wireless Systems Laboratory Apurva Mody, Jun Tan, Kihong Kim, Hasung Kim, JoonBoem Kim, Galib Assadullah, and Qing Zhao. My work at the Wireless Systems Laboratory would have been much harder without your support. Thanks are also due to all my friends at Georgia Tech who showed their support, especially Loran Jatunov, Cagatay Candan, Babak Firoozbakhsh, and Roberto Uzcategui.

I would like to thank my parents whose love, support, and continuous prayers kept me going. Mom, dad, I could have never completed my Ph.D. if it wasn't for your devotion. Your love gave me the strength to keep on working. I am also thankful to my brother Zakarya, and my sisters Maryam, Reem, Fatimah, and Zainab. I love you so much. Last but not least, I would like to express my deepest thanks, love, and gratitude to my lovely wife Abeer who dedicated her time and effort to see me succeed. Dear Abeer, I could never thank you enough.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Software radios (SWR) have been proposed for wireless communication systems to enable them to operate according to incompatible wireless communication standards by implementing most analog functions in the digital section on software-reprogrammable hardware. However, this significantly increases the required computations for SWR functionality, mainly because of the digital front-end computationally intensive filtering functions, such as sample rate conversion (SRC), channelization, and equalization. For increasing the computational efficiency of SWR systems, two new SRC methods with better performance than conventional SRC methods are presented. In the first SRC method, we modify the conventional CIC filters to enable them to perform SRC on slightly oversampled signals efficiently. We also describe a SRC method with high efficiency for SRC by factors greater than unity at which SRC in SWR systems may be computationally demanding. This SRC method efficiently increases the sample rate of wideband signals, especially in SWR base station transmitters, by applying Lagrange interpolation for evaluating output samples hierarchically using a low-rate signal that is computed with low cost from the input signal.

A new channelizer/synthesizer is also developed for extracting/combining frequency multiplexed channels in SWR transceivers. The efficiency of this channelizer/synthesizer, which uses modulated perfect reconstruction (PR) filter banks, is higher than polyphase filter banks (when applicable) for processing few channels, and significantly higher than discrete filter banks for processing any number of variable-bandwidth channels where polyphase filter banks are inapplicable. Because the available methods for designing modulated PR filter banks are inapplicable due to the required number of subchannels and stopband attenuation of the prototype filters, a new design method for these filter banks is introduced. This method is reliable and significantly faster than the existing methods.

Modulated PR filter banks are also considered for implementing a frequency-domain

block blind equalizer capable of equalizing SWR signals transmitted though channels with long impulse responses and severe intersymbol interference (ISI). This blind equalizer adapts by using separate sets of weights to correct for the magnitude and phase distortion of the channel. The adaptation of this blind equalizer is significantly more reliable and its computational requirements increase at a lower rate compared to conventional time-domain equalizers making it efficient for equalizing long channels that exhibit severe ISI.

CHAPTER I

INTRODUCTION

Many of the communication standards that define the specifications for today's wireless digital communication systems, among which are the Interim Standard-95 (IS-95) [96], the Global System for Mobile Communication (GSM) [25], and code-division multiple access 2000 (cdma2000) [97], are highly incompatible. This incompatibility between communication standards is attributed to the use of different air interface specifications, for they may have different

- access methods, such as frequency-division multiple access (FDMA), code-division multiple access (CDMA), or time-division multiple access (TDMA),
- modulation schemes, such as quadrature-phase shift keying (QPSK), minimum shift keying (MSK), or Gaussian minimum shift keying (GMSK),
- radio frequency (RF) operation bands such as VHF, UHF, SHF, or EHF,
- RF allocations within a single band such as the 800 MHz, 900 MHz, or 1900 MHz cellular allocations, all of which are in the UHF band,
- channel bandwidths such as 30 kHz, 200 kHz, 1250 MHz, or 5 MHz.

The specifications of the widely adopted analog and digital cellular communication standards listed in Table 1 [12] clearly illustrate the significant incompatibilities that exist among the different cellular standards. The incompatibility between cellular and non-cellular wireless standards, such as wireless local area network (WLAN) standards, is even higher. Since reprogramming of the analog front-ends of conventional digital wireless communication systems, where many functions are performed, may be difficult or inapplicable, deploying these systems for different standards may require the replacement of virtually all the analog hardware in the radio's front-end. It may also be necessary to replace the non-reprogrammable

digital hardware and the hardware that does not provide sufficient reprogrammability for implementing the new standards in these systems.

Table 1: Comparison of different analog and digital cellular standards [12].

	AMPS	GSM/ DCS-1900	IS-136	IS-95	cdma2000	WCDMA/ UTRA
Standard Type	Analog	Digital	Digital	Digital	Digital	Digital
Generation	1	2	2	2	3	3
Frequency Band	Cellular	Cellular PCS	Cellular/ PCS	Cellular/ PCS	PCS	PCS
Uplink (MHz)	824-849	890-915/ 1850-1910	824-849/ 1850-1910	824-849/ 1850-1910	1850-1910	1920-1980
Downlink (MHz)	869-894	935-960/ 1930-1990	869-894/ 1930-1990	869-894/ 1930-1990	1930-1990	2110-2170
Access Method	FDMA	TDMA	TDMA	CDMA	CDMA	CDMA
Channel Bandwidth	30 kHz	200 kHz	30 kHz	1.25 MHz	1.25, 3.75, 7.5, 11.25, 15 MHz	5, 10, 20 MHz
Modulation Type	FM	GMSK	$\pi/4$-DPSK	QPSK and OQPSK	QPSK and BPSK	QPSK and BPSK
Max. O/P Power						
Base:	20 W	320 W	20 W	1.64 kW	1.64 kW	Unspecified
Mobile:	4 W	8 W	4 W	6.3 W	2 W	1 W
Users/Channel	3	8	3	Up to 63	Up to 253	Up to 250
Data Rate/User	19.2 kbps	22.8 kbps	13 kbps	19.2 kbps	1.5 kbps– 2.0736 Mbps	100 bps– 2.048 Mbps

The use of software radio (SWR) technology has been proposed for implementing future generations of wireless communication systems, including third-generation cellular systems (3G). SWR systems are capable of reducing or eliminating the problems experienced when deploying wireless communication systems for operation according to standards with different air interfaces. An SWR communication system typically uses a single wideband high-dynamic-range analog-to-digital converter (ADC) in the receiver side and a single wideband high-dynamic-range digital-to-analog converter (DAC) in the transmitter side. The wideband ADC and DAC of SWR systems are placed as close to the antenna as possible. Ideally, simple wideband receiving and transmitting RF stages, respectively, separate the ADC and

DAC from the SWR antenna, as shown in Fig. 1 [39, 60, 62, 111]. By placing the ADC and DAC of the system close to the antenna, most of the communication functions can be performed in the digital section of the radio. Consequently, these functions can be implemented on software-reprogrammable platforms such as DSPs and, therefore, can be reconfigured by reprogramming the processors on which they are implemented. To switch from GSM to CDMA operation in an SWR cellular base station transceiver, for example, all digital filter bandwidths and center frequencies are readjusted to the suitable values for transceiving CDMA signals, and the GSM modulator and demodulator are reprogrammed to perform CDMA spreading and despreading [59]. The parameters of other functions that may also be required, such as sample rate conversion (SRC), are also adjusted to suit CDMA reception. The reconfigurability of SWR systems gives them the flexibility required for supporting arbitrary air interfaces within the ADC and DAC bandwidth and dynamic range limitations, and the processing power limitations of the signal processors that are used [64]. SWR systems with sufficient computational power may also be capable of communicating using multiple air interfaces simultaneously. Implementing this feature in military SWR base stations, for example, gives them the capability of bridging incompatible communication networks, while implementing this feature in commercial SWR base stations allows them to provide global roaming capability to conventional mobile units. SWR technology also allows software updates and bug fixes to be downloaded to base stations and user terminals quickly and easily or distributed via smart cards without the need for issuing equipment recalls. Finally, software-reprogrammability enables wireless service providers to offer new or temporary services to users with SWR-capable equipment, such as advanced multimedia applications and vocoders for different languages, via over-the-air software downloads [59].

Although the SWR concept is applicable for applications such as satellite mobile communications, mobile military communications, air and sea traffic control, and WLAN [62], it is particularly attractive for commercial cellular systems. This is mainly due to the rapidly increasing number of cellular standards, the wide spread of cellular systems, and the increasing demand for cellular mobile units that are flexible for incorporating new services easily and quickly and permit global roaming. However, the increased flexibility and

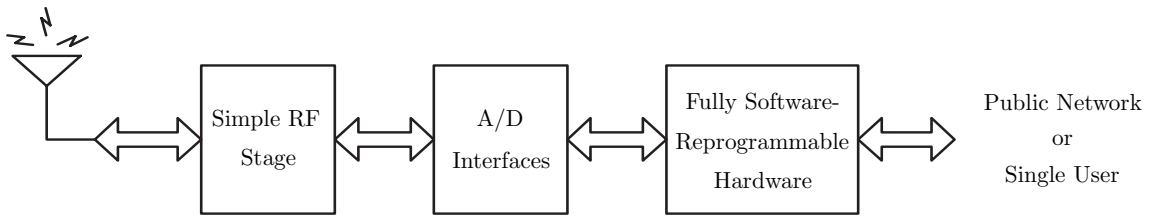


Figure 1: The ideal SWR wireless communication system [39, 60].

features that SWR systems provide to digital communication applications come at a cost. Performing many of the functions that are confined to the analog section of conventional digital communication systems, such as channelization, digitally in SWR systems results in a significant increase in the computational requirements of the receiver. The computational requirement of SWR systems is further increased by the functionality of SWR systems that enables the support of standards with different air interfaces. This functionality forces the use of digital functions that are normally not required in conventional digital communication systems such as SRC [37, 39]. Also, feeding wideband signals that may contain the complete transmission band of a particular wireless air interfaces to a single ADC may generate digital signals with very high dynamic ranges because of variations in the propagation environment of the different channels that compose the received signal [39, 77]. Likewise, the dynamic range of the digital input signals to the DAC that contain a complete transmission band of an air interface may be high because of power management that may be used in SWR transmitters [111]. The processing of high-dynamic range signals in SWR systems further increases the computational requirements because higher word-length operations and longer digital filters are required. Typical signals processed by SWR base stations may have bandwidths of 25 MHz and dynamic ranges of 90 dB [25, 60, 64]. In extreme cases, SWR systems may process signals with bandwidths and dynamic ranges that exceed 100 MHz and 100 dB, respectively. Extracting a single channel for example, from a digitized wideband signal followed by the required baseband processing of the extracted channel may require computational resources far exceeding those of fast DSPs [6, 39, 62, 85]. Therefore, SWR base stations may require a large number of currently available DSPs for

performing the required digital signal processing. The high computational requirements for processing wideband SWR signals may result in high design, implementation, operation and maintenance costs, and high power consumption. Therefore, this may render SWR systems inefficient to build and operate. Since the digital front-end can consume a sizeable fraction of the available DSP resources in an SWR system, and since DSPs are currently the bottleneck of SWRs, computationally efficient methods for performing the functions of the digital front-end are essential.

The aim of the research in this thesis is to investigate new techniques for performing computationally demanding functions in the SWR digital front-end. Lowering the computational cost and/or improving the performance of SRC, channelization, and equalization in SWR systems lowers the cost of implementing and operating these systems. For improving the computational efficiency of SRC in SWR systems, we have modified the conventional cascaded integrator-comb (CIC) filter to improve its performance and make it more suitable for SWR systems. We also develop an SRC method based on the use of Lagrange interpolation. This SRC method is particularly efficient for SWR base station transmitters where SRC by factors greater than unity of wideband signal is generally required. For channelization/synthesis of frequency-division multiplexed (FDM) wideband signals, we present a new channelizer/synthesizer that uses modulated perfect reconstruction (PR) filter banks as building blocks. The channelizer/synthesizer is capable of operating at a higher computational efficiency than polyphase filter banks for processing a small number of channels and discrete filter banks for processing any number of channels. Since the methods for designing modulated PR filter banks proposed in the literature are generally inapplicable for this channelizer/synthesizer because of demanding design parameters, we derive an efficient method for designing these filter banks. The designed modulated PR filter banks are shown to be also suitable for implementing efficient and powerful frequency-domain blind equalizers that are capable of equalizing channels with very long impulse responses and sever intersymbol interference (ISI). These blind equalizers are block equalizers that are computationally efficient for equalizing channels with impulse responses that extend over a large number of symbols periods.

The remainder of this thesis is organized as follows. Chapter 2 provides brief background on SWR systems and discusses issues related to the computational requirements of critical SWR functions. In Chapter 3, the modified cascaded-integrator-comb filters for performing SRC in SWR systems are discussed. Chapter 5 discusses the efficient channelizer/synthesizer that is introduced for channelizing and synthesizing the discrete channels in SWR receivers and transmitters, and Chapter 6 discusses the method that we derive for designing modulated PR filter banks with high number of subchannels and high stopband attenuations to be used in the channelizers/synthesizers of Chapter 5. The details of the frequency-domain blind equalizer are discussed in Chapter 7, and some concluding remarks are given in Chapter 8.

CHAPTER II

BACKGROUND

Commercial and military software-reconfigurable communication systems provide their users with many features that cannot be supported by non-software-reconfigurable communication systems. Commercial SWR wireless base stations, for example, can be deployed quickly and easily for operation according to any desired communication standard with minimal setup cost. After deployment, firmware upgrades and software bug fixes can be downloaded over the air to base stations and mobile units without the need for equipment recall. Also, a wider range of enhanced voice and multimedia services can be offered by SWR systems, and band congestion can be reduced by efficiently distributing users over the available band [6, 64, 111]. Furthermore, SWR base stations can probe the communication channel to a particular user terminal and choose the best air interface and the different parameters of the selected air interface in an attempt to achieve the highest possible quality of service [60]. Also, the ability of SWR systems to switch between air interfaces provides global roaming capability and permits backward compatibility with older communication equipment. An SWR wireless PC card, for example, may provide a notebook PC with WLAN access through access-points operating with any of the IEEE 802.11 standards. When WLAN access points are unavailable, the SWR PC card may attempt to connect to a cellular communication tower. If cellular service is also unavailable, the SWR wireless PC card may initiate a satellite communication link.

The flexibility offered by SWR systems comes at a cost. The design of SWR communication systems is much more complicated than conventional communication systems. The challenges of designing SWR systems are different for commercial and military applications. The design of SWR base stations also generates a different set of challenges than the design of SWR mobile units. For example, commercial SWR cellular systems, in extreme cases, operate at RFs that are scattered in part of the UHF band (800–2200 MHz), while military

SWR systems, in extreme cases, operate at RFs that may be scattered in all frequency bands from HF to EHF (3–30000 MHz), as illustrated in Fig. 2 [113]. Also, unlike the relatively friendly environment of commercial SWR wireless systems, where an attempt is made to minimize the co-channel interference experienced by the different users, military radios are designed to operate in environments where the signal may experience high Doppler and may be subjected to intentional interference in the form of jamming. However, the bandwidth efficiency of commercial systems may be significantly higher than that of military systems. Also, military SWR mobile units are less constrained by cost considerations that constrain commercial SWR mobile units, where inexpensive customer equipment is usually desired.

The design of SWR base stations and mobile units has different sets of challenges. For example, SWR mobile units usually communicate with a single base station using a single air interface at any particular time, while SWR base stations may be required to simultaneously communicate with a large number of mobile units that have different air interface specifications. The multimode operation of SWR base stations imposes demanding computational requirements. Although power consumption is not a critical factor in SWR base stations, designing power-efficient SWR base stations reduces the cost of operating such systems and, therefore, increases their profitability. The computational requirements of SWR mobile units are much lower than the requirements of SWR base stations because they communicate with a single base station; however, the computational requirements may be relatively high for a hand-held mobile unit considering the limited size and limited power supply. Therefore, restrictions on size and weight place constraints on the amount of power storage and consumption in SWR mobile units [64]. In conclusion, successful design of SWR systems is constrained by finding solutions to the different problems facing the design of base stations and mobile units of military or commercial SWR systems.

2.1 Brief History of SWR Systems

The general concept of SWRs, in which radio functions run on software-reprogrammable platforms, was introduced in 1992 by Joseph Mitola III [60] as a method for reducing the cost, providing an extended range of services, and reducing the deployment time of military

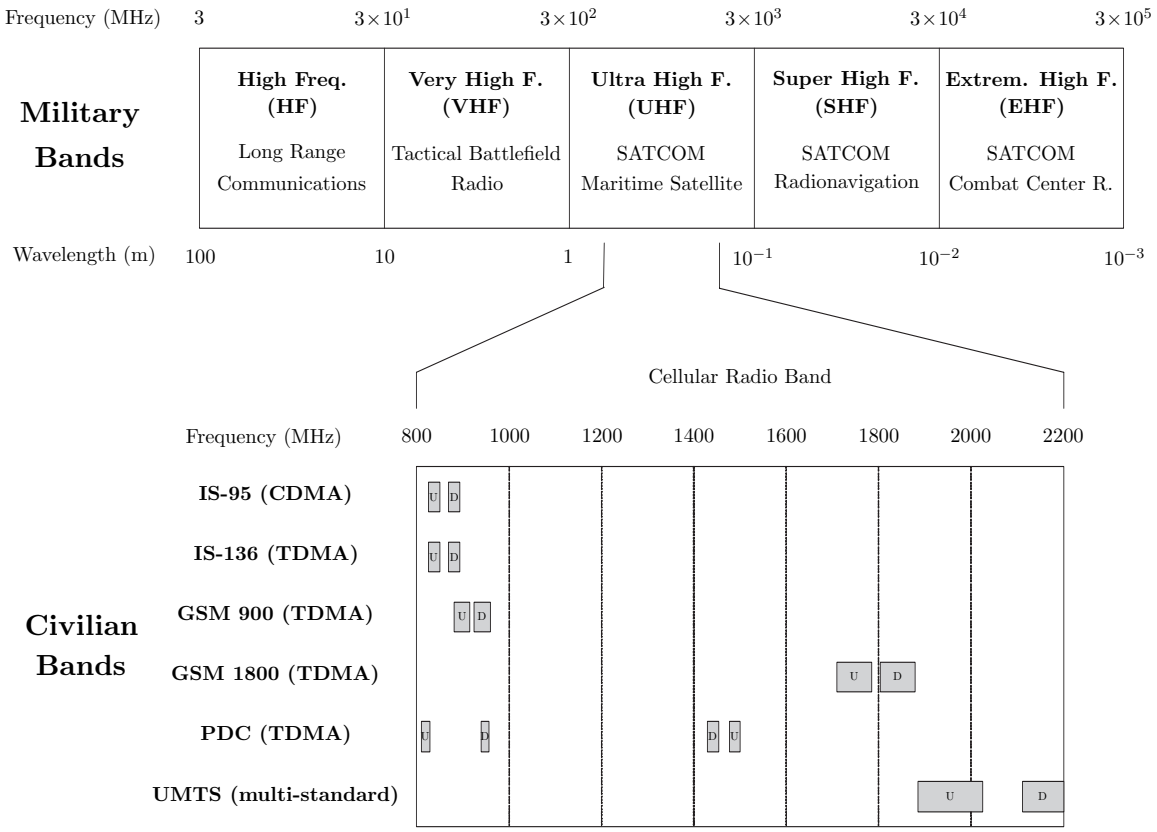


Figure 2: Frequency bands of operation of military mobile and civilian cellular radio systems [113]. The bands labelled U and D are the mobile unit uplink and downlink bands, respectively, corresponding to the base station downlink and uplink bands, respectively.

and commercial wireless communication systems. Since then, an increasing amount of research has been devoted to the design and implementation of SWR systems. The greatest interest in SWR systems has been shown by the U.S. Department of Defence (DoD) with the objective of eliminating the need for multiple radio transceivers with different air interfaces for different military applications. The interest of the DoD in SWRs has led to the design, implementation, and testing of the first SWR system: Speakeasy Phase I in August 1994. Because military communication applications use different frequency bands for different communication topologies and different transmission data rates, Phase II of the Speakeasy radio has the capability of emulating more than 15 military radios, which have different modulation schemes and different RF carrier frequencies scattered in the HF, VHF, UHF, L, C, and X-bands. The Speakeasy Phase II has a reconfigurable DSP unit with computational power of 1 billion 16-bit integer operations/s and 200 million floating-point operations/s

(MFLOPS). It has the ability to simultaneously operate in up to four modes of operation and the ability to bridge incompatible networks, thereby allowing two or more users with incompatible radio transceivers to communicate seamlessly using the Speakeasy system as a user-transparent bridge [52, 61, 98, 111].

The development and testing of the multiband and multimode (i.e., operates in different frequency bands and capable of emulating radios with different air interfaces) military Speakeasy SWR spurred the development of commercial SWR cellular systems [6, 9, 18–22, 24, 36, 37, 39, 41, 42, 45, 47, 50, 52, 59, 61, 62, 64, 66, 85, 88, 93, 101–103, 111, 113, 114, 117, 120]. In November 1999, the International Telecommunication Union (ITU) introduced the standards for the global 3G mobile communication system IMT-2000 [81]. The introduction of the IMT-2000 standard, which supports five different air interfaces, is a major motivation for developing SWRs for 3G cellular applications to facilitate global roaming and provide backward compatibility for second-generation (2G) base stations and mobile units [45].

2.2 The Bottleneck of SWR Systems

The greatest challenges in designing SWR systems result from the unavailability of ADCs and DSPs that are fast enough to support the high levels of SWR reconfigurability and the high computational requirements of the different processes in SWR systems. The speeds of currently available ADCs generally lag the conversion rates required by SWR systems for supporting wideband reception. However, it is expected that sufficiently fast ADCs that are capable of supporting the different SWR applications will soon be available because of advances in different ADC technologies. Therefore, the DSPs that are available today represent the bottleneck of SWRs [6]. Increasing the computational efficiency of the SWR functions, in particular, becomes essential for successful and profitable operation of SWR systems. Two methods have been considered in the literature for accomplishing this. Significant research has been devoted to improving the SWR architecture and exploring the commonalities between the different communication standards to reduce the complexity of implementing SWR systems [10,30,41,42,47,59,61,65,92,93,113]. Relatively less emphasis has been given to improving the SWR signal processing algorithms [26, 37, 38, 50, 117, 119, 120].

SRC and channelization are two functions that have been considered for optimization in SWR systems because of their high computational requirements, especially in SWR base stations [36, 111].

Most, if not all, processing in conventional digital wireless communication systems is performed using analog hardware or fast non-programmable digital hardware, such as ASICs. Unlike these conventional systems, where narrowband channels with low-dynamic ranges are usually transceived, SWR wireless systems may transceive extremely wideband signals with high dynamic ranges using slower highly reprogramable digital hardware, such as DSPs. This causes the digital signal processing in SWR systems to be computationally intensive, where a large number of currently available DSPs may be required for applying functions, such as SRC and channelization (or channel synthesis), on the signals that are transceived by SWR systems. Since increasing the efficiency of the signal processing algorithms in SWRs reduces the overall required computations and, therefore, lowers the required number of DSPs and/or reduces the size and power consumption of the required DSPs, improving the signal processing algorithms is key for the success of SWR systems.

2.3 Reconfigurability of SWR Systems

The main feature of SWR communication systems that distinguishes them from their conventional radio counterparts is their ability to mutate depending on the air interface(s) of the communication system(s) with which they are attempting to communicate. The mutation is performed by software-reprogramming of the signal processing components while avoiding any form of modification or upgrade to the hardware of the radio [62]. Different SWR implementations may have different levels of software-reconfigurability. An SWR system may be fully software-reprogrammable, software reconfigurable only, or partially software-reprogrammable and partially software reconfigurable. Therefore, the terms SWR and software defined radio (SDR) are often used in describing digital radios that have the capability of being altered by means of software [59, 64, 91]. The research discussed in this thesis adopts the definitions for SWRs and SDRs given in [64], which states that SDRs implement the desired set of capabilities using software-reconfigurable elements, such as

application-specific integrated circuits (ASIC), while SWRs implement the desired capabilities using software-reprogrammable elements, such as DSPs. This means that the software of an SWR defines the functions of the air interface, while the software of an SDR defines the parameters of the air interface, and possibly the particular ASIC to be used for performing a specific communication function. Therefore, SWRs provide a higher degree of reconfigurability and are more flexible than SDRs. The main scope of the research discussed here is to increase the efficiency of the computational algorithms required in SWR communication systems specifically, although applying the algorithms described herein to SDRs may reduce their hardware complexity as well.

A fully reconfigurable SWR system like the ideal SWR shown in Fig. 1 is impractical for real applications. In a real communication application, the simplicity of the RF stage imposes the use of extremely wideband and high dynamic-range ADCs and DACs for covering the frequency bands of all signals of interest and the dynamic ranges of the different signals that may be processed. Besides being computationally demanding, processing the digital signals in the system of Fig. 1 is wasteful because the useful information for a particular mobile unit or base station generally occupies a small fraction of the frequency band covered by an ideal SWR system [113]. In general, the computations that an ideal SWR cellular system requires cannot be delivered by DSPs that are available today or will be available in the foreseeable future [6]. Therefore, lower software reconfigurability levels of SWR systems may be considered for more practical implementations.

The computational requirements of the digital functions that are reconfigured for handling different air interface specifications are generally proportional to the sample rates and dynamic ranges of the signals being processed. Therefore, the bandwidth and dynamic range of the ADC and DAC, which is specified by their location in the system, determine the level of software reconfigurability of the SWR system as well as its computational requirements. Moving the ADC and DAC closer to or further away from the antenna results in a tradeoff between the complexities of the radio's analog and digital sections, which consequently results in a trade-off of the SWR system reconfigurability with its computational requirements [113]. The level of reconfigurability of an SWR system is also determined by

the type of components used for signal processing, such as ASICs, function programmable gate arrays (FPGA), dedicated DSPs, or general-purpose processors [18, 21, 61]. The SWR configurations that are applicable for SWR base stations and mobile units are discussed next. It is important to note that either bandpass or baseband digitization may be used in these configurations. In baseband digitization, analog downconverters are used to mix the signals to baseband as in-phase and quadrature (I and Q) signals that are then digitized using two balanced ADCs. In bandpass digitization, the analog band of interest is centered around a nonzero frequency and is digitized using a single bandpass ADC such that the folding of the digital band downconverts the desired band. Digital downconversion may further be used to center the signal around zero frequency. Both of these configurations are equivalent in the sense that they eventually result in similar sample rates, except that RF digitization reduces the system complexity at the expense of an increase in the complexity of the sample-and-hold circuit to reduce the noise caused by ADC aperture jitter.

2.3.1 Reception of the Complete Transmission Band of Air Interfaces

Moving the interfaces between the analog and digital sections of the SWR system away from the antenna by inserting analog bandlimiting filters significantly limits the required ADC bandwidth and DSP computations. Such a filter may be part of an IF stage with the purpose of bandlimiting the input signal to the ADC and output signal from the DAC to the transmission bandwidths of the air interface being processed [113]. This configuration is shown in Fig. 3 for an SWR cellular system. In this configuration, a software tunable analog bandpass filter passes only the transmission band of a particular air interface to the ADC. Therefore, the maximum bandwidth and dynamic range values of the fixed-sample-rate ADC are equal to the widest band and highest dynamic range of the different communication standards that may be served by the SWR system. The required resolution of the ADC is computed by considering that the maximum theoretical signal-to-noise ratio (SNR) of an ADC in dBs [112] is given by

$$\mathrm{SNR} = 6.02B + 1.76 + 10\log_{10}\left(\frac{f_s}{2f_{max}}\right) \quad \mathrm{dB}, \tag{1}$$

where B is the number of bits, f_s is the sampling frequency, and f_{max} is the maximum frequency of the input signal, and also considering that the actual SNR of an ADC falls short of the theoretical SNR by roughly two bits of resolution. For cellular SWR applications, for example, the required wordlength of the ADC may be in the range of 17–19 bits to support the highest dynamic ranges of 90-100 dBs that GSM transmission bands may have [25]. Since the largest transmission band of the different air interfaces is that of the 3G IMT-2000 standard with bandwidth of around 150 MHz [113], ADC and DAC sample rates on the order of 350 Msamples/s may be required. Although such ADCs may soon be available, the computational cost for up/downsampling and channelizing/synthesizing the digital wideband signals is extremely high. For example, applying a high-quality filtering process to the wideband digital signal to extract a single channel may require processing power on the order of 100 operations per sample [6], or approximately 35,000 million operations per second (MOPS) with wordlengths up to about 20 bits. The computations in this configuration may overwhelm SWR DSPs since the fastest currently available DSPs, like the Texas Instruments TMS320C6416 [95], perform on the order of 3000 MOPS (32-bit operations). This configuration is most suited for SWR base stations since it retains a high level of system reconfigurability and the computations may be manageable if efficient DSP algorithms are employed.

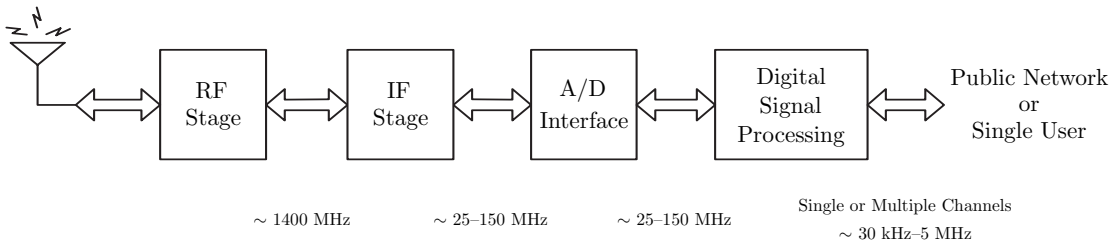


Figure 3: Typical bandwidths of the signals at different stages of an SWR cellular system employing reception of the complete transmission band of air interfaces.

2.3.2 Single Channel Reception

For an SWR client unit, where single channels are received and transmitted, the SWR configuration in which the complete transmission band of a particular air interface is transceived may be impractical because of the wasted computational power of wideband reception for extracting a relatively narrowband signal. The computations required in SWR mobile stations can be reduced significantly to manageable values by introducing an analog software tunable bandpass filter as part of a second IF stage with the purpose of passing the band of the desired channel only.

For SWR systems that operate on individual channels of a particular air interface at any specific time, as shown in Fig. 4 [113], the passband or baseband ADC and DAC bandwidths must be at least equal to the largest channel bandwidth of the different air interfaces that may be transceived. For cellular SWR mobile units, for example, the required ADC and DAC bandwidths may be 5 MHz to support wideband-CDMA (W-CDMA) and all communication standards with channels of lower bandwidths such as GSM. The required dynamic ranges of the ADCs and DACs are generally lower than the ADC and DAC dynamic ranges required in the configuration described in Section 2.3.1, because the near-far effect in the received signal has been eliminated and power-management in the transmitted signal can be performed easily in the analog RF stage. However, the reconfigurability of this SWR architecture is severely limited by the significant use of analog mixers and filters in the analog RF and IF stages. Consequently, this configuration is suitable for SWR mobile units only, where the transmission and reception of only individual channels of a particular air interface is desired. Because of the relatively narrow bandwidth and low dynamic range of the processed signals, and because many of the functions are performed in the analog section of the radio, the amount of required computations in this configuration is limited. The limited computations are generally suitable for limited-power SWR mobile units with the use of optimized signal processing functions. The use of this configuration is generally not suitable for SWR base stations that transceive multiple channels because of the limitations imposed by the significant use of analog components.

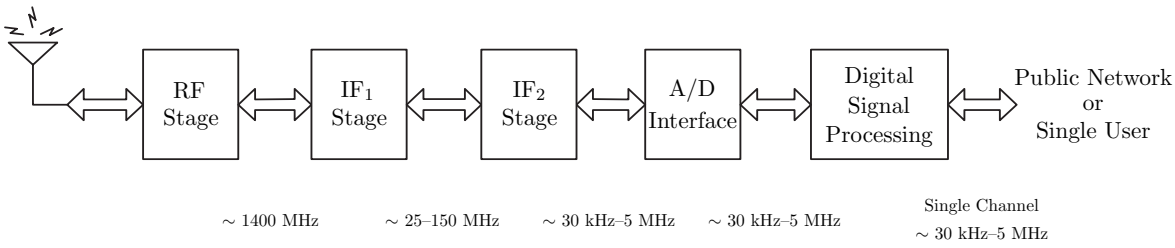


Figure 4: Typical bandwidths of signals at different stages of a cellular SWR with single-channel reception.

2.4 Computationally Demanding DSP Functions in SWR Systems

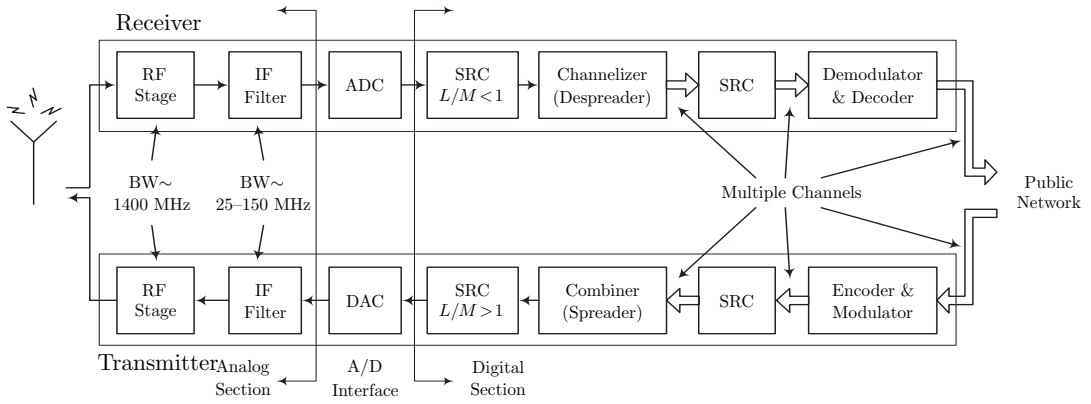
The total rate of operations that a single-input single-output process or a process such as SRC or a multi-input and/or multi-output process such as channelization and channel synthesis in an SWR system performs is a function of several factors. These factors include the sample rate(s) of the input signal(s) and output signal(s) and the number of operations that are required for computing each sample of the output signal(s). The number of operations per output sample may itself be a function of the different parameters of the input signal such as the oversampling factor and dynamic range. For processing a specific signal, the input and output sample rates of the process are generally fixed. Therefore, increasing the efficiency of this process demands that the number of operations per output sample be reduced. Since the parameters. In this subsection, we describe processes of SWR systems that are computationally demanding [36, 111], i.e., they require high number of operations per output sample.

2.4.1 Sample Rate Conversion (SRC)

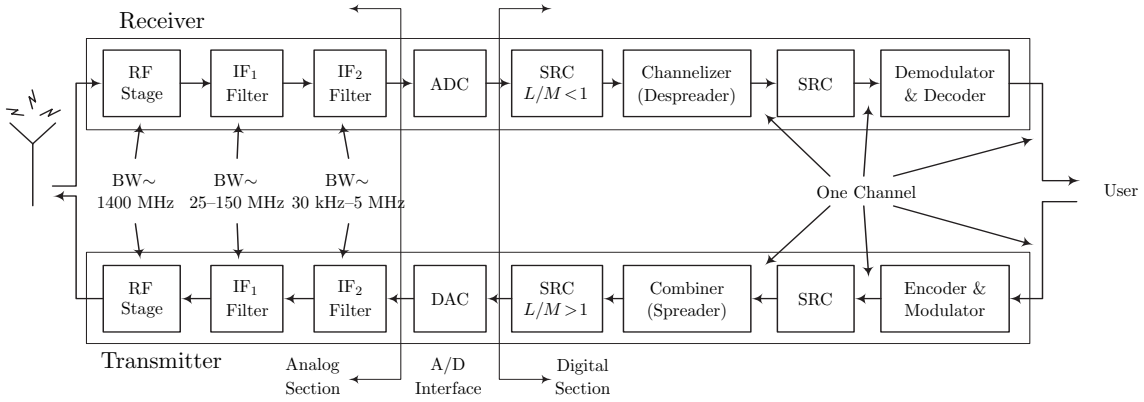
Since an SWR system uses software-reprogramable hardware that runs at a fixed master clock rate regardless of the air interface being used, the sample rates of the ADC and DAC are fixed and are often incompatible with the sample rates required for channelizing, synthesizing, and processing the baseband channels [37, 39]. Therefore, SRC is necessary for interfacing the ADC and DAC to the remainder of the SWR system. Since SRC can consume a sizeable fraction of the available DSP resources in an SWR, and since DSPs are

currently the bottleneck of SWRs due to the large amount of computations that is required, efficient SRC is highly desirable. Different methods for performing SRC by a rational factor of L/M exist [14, 73, 83, 107], and different techniques have been proposed for efficient implementation of SRC in SWR systems [37, 39, 43, 53]. Considering only rational-factor SRC in SWR systems is due to the fact that the ratios of the sample rates required for processing the signals of the different air interfaces can always be represented by rational values. Irrational SRC factors, nevertheless, can be approximated with high accuracy using rational factors L/M with sufficiently large values of L and M.

SWR transceivers may require several SRC blocks along the signal path for interfacing functions that operate at different sample rates. For example, SRC may be required at the ADC/channelizer interface in SWR base station receivers [37, 39] and the channel-synthesizer/DAC interface in SWR base station transmitters. Since the fixed-sample-rate ADC and DAC must accommodate the largest possible bandwidths and dynamic ranges of the signals that may be transceived, the sample rates of the ADC and DAC may be much higher than the values that are actually required for processing a particular signal. For example, processing the complete transmission band of different wireless air interfaces in the SWR base station of Fig. 5(a) may require the use of ADCs and DACs with bandwidths on the order of 150 MHz and dynamic ranges of 100 dB to accommodate the extreme IMT-2000 transmission bandwidth and the maximum GSM dynamic range [25, 77, 85, 113], although the transceived signal at a particular time may have much lower bandwidth and/or dynamic range. Similarly, the ADC and DAC of an SWR mobile unit, as that shown in Fig. 5(b), may have bandwidths of 5 MHz for processing individual channels of different air interfaces with bandwidths up to the 5 MHz channel bandwidth used in W-CDMA, even when a 200 kHz GSM channel, for example, is being processed by the ADC and DAC. In such cases, downsampling after the ADC and upsampling before the DAC may be required [50]. As shown in Fig. 5, SRC by factors that are close to unity may also be required at the channelizer/demodulator and modulator/channel-synthesizer interfaces because the sample rates of the baseband channels output by a channelizer or input to a channel-synthesizer may be different from the chip- or bit-rates of the received or transmitted data [39, 85, 120].



(a) Construction of an SWR wireless base station



(b) Construction of an SWR wireless mobile unit

Figure 5: Construction of SWR wireless systems. The bandwidths of the different stages are typical for cellular wireless systems.

SRC in SWR systems may be computationally-intensive due to the relatively wide bandwidth and high dynamic range (near-far effect in SWR receivers [39] and power management in SWR transmitters) of SWR signals. The rate of computations for upsampling in SWR transmitters may be significantly higher than the rate for downsampling in SWR receivers, because the number of samples that are computed in the former case may be several orders of magnitude higher than the number of samples that are computed in the latter. Consider, for example, an SWR base station that uses ADC and DAC sample rates of 350 Msamples/s to accommodate the different IMT-2000 standards. When the base station is actually transceiving a standard GSM 900 transmission with bandwidth of 25 MHz [25], SRC by factors close to 1/7 and 7 may be required in the receiver and transmitter sections of the

base station, respectively. Assuming the use of simple single-stage sample rate increase and decrease processes with filters of equal guard bandwidths that require equal number of operations for computing an output sample in both cases, the rate of computations for sample rate increase would roughly be 7 times higher than the rate of computations for sample rate reduction. Using efficient SRC methods such as CIC filters may allow performing sample rate reduction in SWR at a high efficiency as illustrated by the results given in [37] (Fig. 6 and Table 1), which indicate that less than 10 MPOS may be required for downsampling in an SWR system regardless of the air interface, channel bandwidth, and output sample rate. For sample rate reduction processes in SWR systems that require filters with narrower transition bandwidths, the rate of computations may be higher but generally lower than the required rate for sample rate increase in SWR transmitters. In the literature, However, significantly more emphases has been directed towards the study of sample rate reduction in SWR systems compared to that directed towards sample rate increase. In this thesis, we present two SRC methods, one of which is applicable for increasing the sample rate of the signals in SWR transmitters in particular.

2.4.2 Channelizing/Synthesizing Wideband SWR Signals

A significant fraction of the computational power of wideband SWR base station receivers may be dedicated to channelizing the received wideband signals digitally and preparing the extracted channels for baseband processing. SWR base station transmitters may also dedicate a sizable fraction of their computational power for synthesizing the individual channels into a wideband signal for transmission. The high computational cost for channelizing/synthesising SWR channels is attributed to the long filters that are required for processing wideband high-dynamic-range signals that consist of a large number of channels. Reducing the computations for channelization/synthesis of wideband signals in SWR transceivers is vital for reducing the cost and power consumption of DSPs in SWR systems.

Channelizing a subset with N_c channels from the N FDM channels contained in the received SWR wideband signal can be performed efficiently by polyphase decomposing the filters of modulated discrete filter bank (DFB) channelizers. This is applicable when the

desired sample rate of the extracted channels is $1/L$ of the sample rate of the input signal, such that

$$L = K \times N \tag{2}$$

is satisfied for an integer K, and the digital band is equally divided among all N channels [117,119,120]. If these very restrictive conditions are not satisfied because the different channels in the input signal to the channelizer have different bandwidths or the channels unequally divide the digital spectrum, then polyphase decomposition of the DFB is inapplicable. In this case, the computationally inefficient DFB channelizer may be the only applicable method for channelizing the signal. Polyphase decomposing the DFB of a channel synthesizer, which combines a set of N_s narrowband channels into an N-channel wideband signal in the SWR transmitter, requires similar conditions to those required for polyphase decomposing the channelizer filter bank. For the channel synthesizer, the input channels must have equal sample rates, and the sample rate increase factor L must also satisfy (2) for an integer K. Compared to the computations of DFB channelizers and synthesizers, the use of polyphase filter bank (PFB) channelizers and synthesizers reduces the required computations for channelizing N_c channels and synthesizing N_s channels roughly by factors equal to N_c and N_s, respectively.

The computational requirements for performing channelization/synthesis in the SWR base station and mobile unit may be extremely high. Because of the high sample rates and dynamic ranges of the input signals to the channelizer or output signals from the synthesizer, extracting a single channel from wideband received signals or synthesizing a single channel into a wideband transmitted signal may require the complete computational power of fast DSPs [6, 39, 62, 85]. Typical values of the sample rates and dynamic ranges of the input signals to the channelizer and output signals of the synthesizer may be in the ranges of 60–350 Msamples/s and 90–100 dB, respectively. The computational requirements for fully channelizing/syntheisizng such signals may require the use of 50–100 currently available DSPs. Even when DFB channelizers/synthesizers are applicable, the channelization/synthesis requirements may still be high. Therefore, efficient methods for channelizing/syntheising SWR signal are essential for constructing efficient SWR systems.

2.4.3 Blind Equalization of Wideband SWR Received Signals

The received signals in digital wireless applications, including SWR systems, experience additive noise that results from internal or external sources. The effect of additive noise can be reduced by use error correction codes to detect and correct errors in the received signal. The received signals may also have a much severe and more destructive type of corruption known as intersymbol interference (ISI), which results from the transmission of the digital information over non-ideal finite-bandwidth or multipath channels [54, 79]. In communication systems that suffer from ISI, extracting the digital information from the received signal requires an equalizer that uses a filter to invert the channel response before detecting the symbols in the received signal. The output of such an equalizer is a noisy delayed form of the transmitted signal with an amount of ISI that is small enough to permit the detection of the digital information with an acceptable probability of error. Reducing the ISI requires that coefficients of the equalizer be adjusted such that its response is close to that of the channel inverse. Fig. 6 shows the different modes for equalizer weight adaptation. In the trained equalizer mode, a training sequence is used in conjunction with an applicable adaptation algorithms to adjust the equalizer coefficients [5, 23, 67]. Blind equalization requires only the knowledge of the statistics of the transmitted signal to adapt the coefficients [8, 13, 32, 33, 44, 74]. When successful adaptation in a trained or blind equalizer is reached, adaptation directed by decisions performed on previously received symbols (decision-directed equalization) may be used to track slow variations in the channel. [49, 56, 94, 116].

Adapting the weights of an equalizer may be difficult and computationally demanding, especially for equalizers with high number of weights needed to compensate for long channels with severe ISI. The complexity of adapting an equalizer is increased further by the lack of a training sequence in blind equalizers. Equalizers are implemented either in time-domain (TD) or frequency-domain (FD) [82, 89, 90, 100] forms. While TD equalizers generally have lower input-to-output delays, FD equalizers are more computationally efficient and adapt to channels in a smaller number of iterations but possibly longer time period. In SWR signals that are transmitted through long channels with severe ISI, conventional TD and FD blind

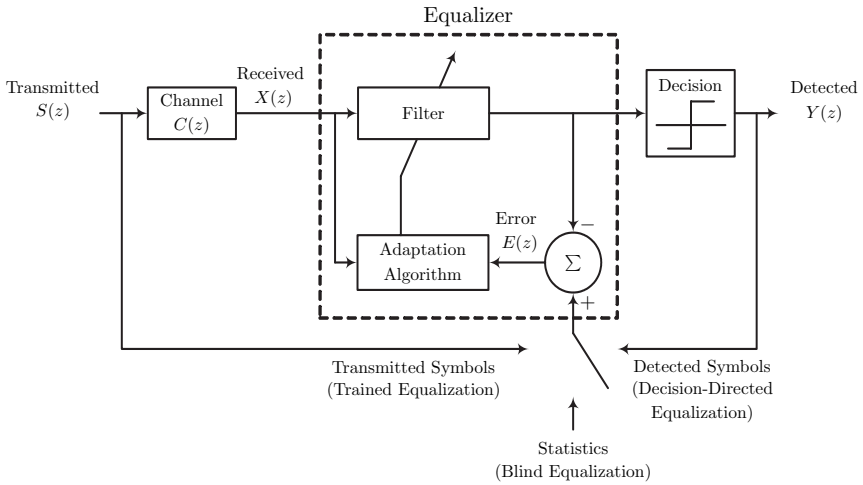


Figure 6: Modes of equalizer adaptation.

equalization methods generally fail to reduce the ISI and may even fail to adapt to the communication channel at all. Therefore, developing powerful blind equalization methods that are capable of equalizing channels that experience severe ISI and have long impulse responses that extend over a large number of symbol periods may be necessary.

CHAPTER III

MODIFIED CIC FILTER FOR SAMPLE RATE CONVERSION IN SOFTWARE RADIO SYSTEMS

Among the methods described in literature for sample rate conversion (SRC) [14, 15, 83], only a few have the computational efficiency that is required for software radio (SWR) systems. Cascaded-integrator-comb (CIC) filters [40, 110] perform SRC efficiently by using only additions/subtractions which makes them attractive for SWR applications. However, conventional CIC filters may be unsuitable for SWR, especially for SRC factors that are close to unity, because they have a limited number of tuning parameters and they exhibit a passband droop.

Fig. 7 shows a CIC filter of order $N_1 + N_2$ that performs SRC by a rational factor of R/L [40], where N_1 and N_2 are the number of comb-integrator stages in the interpolation and decimation sections, respectively. For fixed R/L, the performance of CIC filters can be altered by changing the filter order which controls the image attenuation, and/or the delay of the comb stages M that controls the filter bandwidth.

The remainder of this chapter is organized as follows. Section 3.1 briefly describes the construction of the conventional CIC filter. Section 3.2 then discuss the modified CIC filter and its computational requirements compared to the conventional CIC filter. Finally, Section 3.3 compares the results of simulating the modified CIC filter to the conventional CIC filter.

3.1 Conventional CIC Filter

The transfer function of the CIC filter for SRC by a factor R/L is obtained by reflecting the low sample rate combs across the upsampler and downsampler to the intermediate high sample rate (IHSR) section. This results in a transfer function with respect to the IHSR

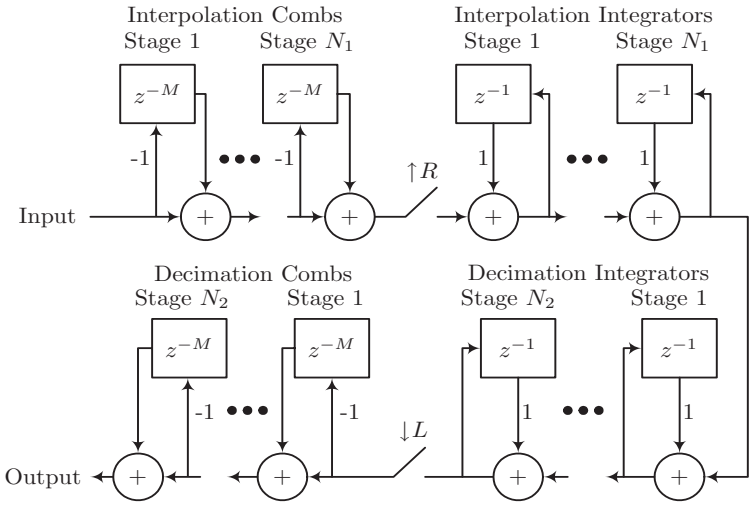


Figure 7: Conventional CIC filter of order N_1+N_2 for SRC by R/L.

given by

$$\begin{aligned} H(z) &= \frac{\left(1-z^{-RM}\right)^{N_1}\left(1-z^{-LM}\right)^{N_2}}{\left(1-z^{-1}\right)^{N}} \\ &= \left(\sum_{k=0}^{RM-1} z^{-k}\right)^{N_1} \cdot \left(\sum_{k=0}^{LM-1} z^{-k}\right)^{N_2}, \end{aligned} \tag{3}$$

where R is the interpolation factor; L is the decimation factor; N is the order of the CIC filter ($N = N_1 + N_2$); and M is the delay of each comb stage. The power response $P(f)$ of the CIC filter is

$$P(f) = \left(\frac{\sin(\pi RMf)}{\sin(\pi f)}\right)^{2N_1}\left(\frac{\sin(\pi LMf)}{\sin(\pi f)}\right)^{2N_2}, \tag{4}$$

where f is normalized with respect to the IHSR. Eq. (4) shows that the CIC filter is a lowpass filter with zeros occurring at multiples of $f = 1/RM$ and $f = 1/LM$. The distribution of zeros, over which there is limited control, is uneven resulting in a low attenuation at some image frequencies. The effect of the uneven distribution of zeros over the undesired images becomes more significant with input signals that have wide dynamic ranges because insufficiently attenuated parts of the images may alias over low power parts of the desired baseband signal.

3.2 Modified CIC Filters

The construction of CIC filters makes their frequency response unsuitable for specific SWR applications. An SWR system must be capable of processing narrowband channelized signals at wideband reception. Due to variations in the propagation environment, the wideband input signal to an SWR has a very high dynamic range. For example, in accordance with GSM 5.05, a GSM receiver should be capable of withstanding a blocking signal that is 85 dB above the desired signal (when the two signals are from 0.8 MHz to 1.6 MHz apart) [25]. Depending on the location of the high-power narrowband channels in the wideband signal, the attenuation of their images may be insufficient. To achieve better performance, we suggest an SWR receiver that locates the high-power channels and accordingly sets the zeros of the CIC filter close to their images to provide them with higher attenuation.

The CIC filter is modified by spreading the delays in the CIC filter comb stages. While the delays of the combs in the conventional CIC filter are equal to RM or LM delay units at the IHSR, the delays are either distributed evenly to provide a more uniform image attenuation or they are set around specific values to provide additional suppression to particularly strong image components. The modified CIC filter has transfer function

$$\begin{aligned} H_m(z) &= \frac{\left(1-z^{-M_1}\right)\left(1-z^{-M_2}\right)\cdots\left(1-z^{-M_N}\right)}{\left(1-z^{-1}\right)^N} \\ &= \left(\sum_{k=0}^{M_1-1} z^{-k}\right)\left(\sum_{k=0}^{M_2-1} z^{-k}\right)\cdots\left(\sum_{k=0}^{M_N-1} z^{-k}\right), \end{aligned} \tag{5}$$

and power response

$$P_m(f) = \frac{\sin^2(\pi M_1 f)\sin^2(\pi M_2 f)\cdots\sin^2(\pi M_N f)}{\sin^{2N}(\pi f)}, \tag{6}$$

where $M_1, M_2, \ldots, M_N$ is a set of comb delays in delay units of the IHSR section that provide the power response of the CIC filter with zeros at multiples of the normalized frequencies $f = 1/M_1, 1/M_2, \ldots, 1/M_N$. Fig. 8 shows the modified CIC filter of order N. For the best performance, the delays $M_1, M_2, \ldots, M_N$ are experimentally set to values in the range $\max(MR, ML)$ to $\max(2MR, 2ML)$ depending on the power spectrum of the input signal such that the modified CIC filter provides the most uniform image attenuation.

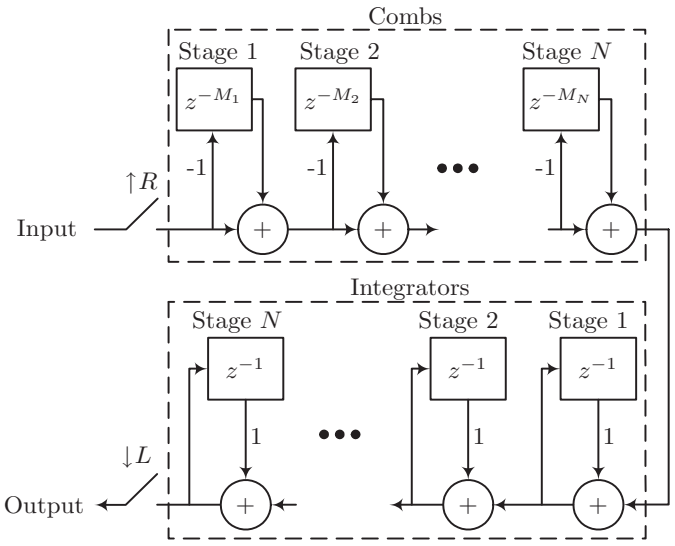


Figure 8: Modified CIC filter of order N for SRC by R/L.

Fig. 9 illustrates the power response of fourth order conventional and modified CIC filters for SRC by 9/10. The zeros of the conventional CIC filter are located at multiples of $f = 1/10$ and $1/9$, while the delays of the modified CIC filter are 16, 14, 12, and 10, which produce zeros at multiples of $f = 1/16$, $1/14$, $1/12$, and $1/10$. For a signal occupying 3/4 of the digital band, the conventional and modified CIC filters provide signal-to-noise ratios (SNR) of 15 dB and 50 dB, respectively, where the SNR is defined as the power ratio after lowpass filtering of the lowest power level in the desired signal to the highest power level in the images.

The complexities of the conventional and modified CIC filters of order $N = N_1 + N_2$ can be compared in terms of their memory requirements and number of additions (or subtractions) per output sample (APOS). While the conventional CIC filter requires $(N_1 + N_2)(M + 1)$ memory elements, the modified CIC filter requires $(N_1 + N_2)\max(3M/2 + 1, 3ML/2R + 1)$ memory elements on average. The $N_1 + N_2$ integrators of the conventional and modified CIC filters require the same number of APOS. Since the integrators operate in the IHSR section, every integrator requires R APIS. Therefore, the $N_1 + N_2$ integrator stages require a total number of $(N_1 + N_2)L$ APOS. The interpolation and decimation combs of the conventional CIC filter require N_1 APIS ($N_1 L/R$ APOS) and N_2 APOS, respectively. When expanded in

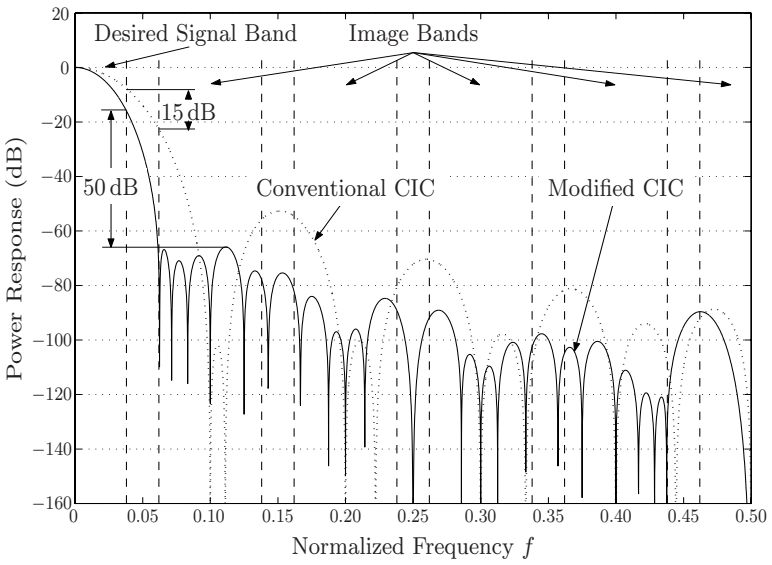


Figure 9: Power response of 4th-order conventional ($N_1 = N_2 = 2$) and modified ($N = 4$) CIC filters for SRC by $R/L = 9/10$.

tree structure, the transfer function of the modified CIC filter comb section has a maximum of $2^{N_1+N_2}$ branches and operates on a signal that has $R-1$ zero samples between consecutive sample of the input signal. This results in every branch requiring one addition/subtraction every R samples of the IHSR signal and the comb stages requiring a maximum of $2^{N_1+N_2}$ APIS or $2^{N_1+N_2}L/R$ APOS. Therefore, the conventional and modified CIC filters require $(N_1+N_2)L+N_1L/R+N_2$ and a maximum of $(N_1+N_2)L+2^{N_1+N_2}L/R$ APOS, respectively. For practical values of R/L close to unity, the modified CIC filter requires approximately 3/2 the number of memory elements and performs a maximum of $(2^{N_1+N_2}-N_1)L/R-N_2$ more APOS than the conventional CIC filter. The extra number of APOS is small for practical SRC factors and filter orders when compared to the total number of computations that the conventional CIC filter requires.

3.3 Simulation Results

In this example, the signal shown in Fig. 10 is processed by a fourth-order CIC filter to perform SRC by a factor of 9/10. The input signal, chosen to illustrate the benefits of the modified CIC filter, occupies 0.83 of the available digital band and contains 31 equal-power frequency multiplexed channels and two 25 dB higher power channels. Fig. 11(a) shows

that the conventional CIC filter fails to attenuate the high-power images resulting in visible aliasing and a SNR of −13 dB. Fig. 11(b) shows that the modified CIC filter provides a SNR of +28 dB. Fig. 12 shows the output signals of both filters where the output of the conventional CIC filter [Fig. 12(a)] contains visible aliasing while the modified CIC filter [Fig. 12(b)] does not. A second-order infinite impulse response (IIR) filter (requiring two multiplications per output sample) is used to correct for the passband droop in the output signals of both CIC filters. Fig. 12 shows that all low power channels have approximately equal power, i.e., there is little passband droop.

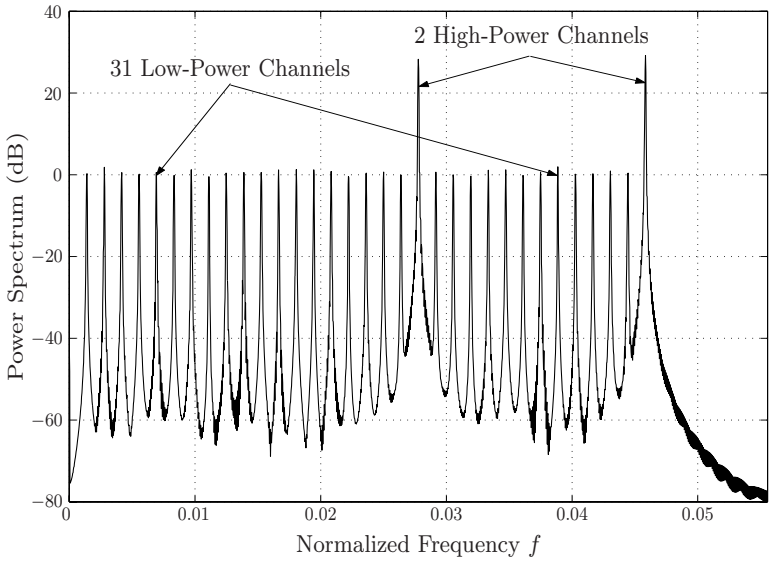


Figure 10: Input signal to conventional and modified CIC filters containing of 33 frequency multiplexed channels (f is normalized with respect to the intermediate high sample rate).

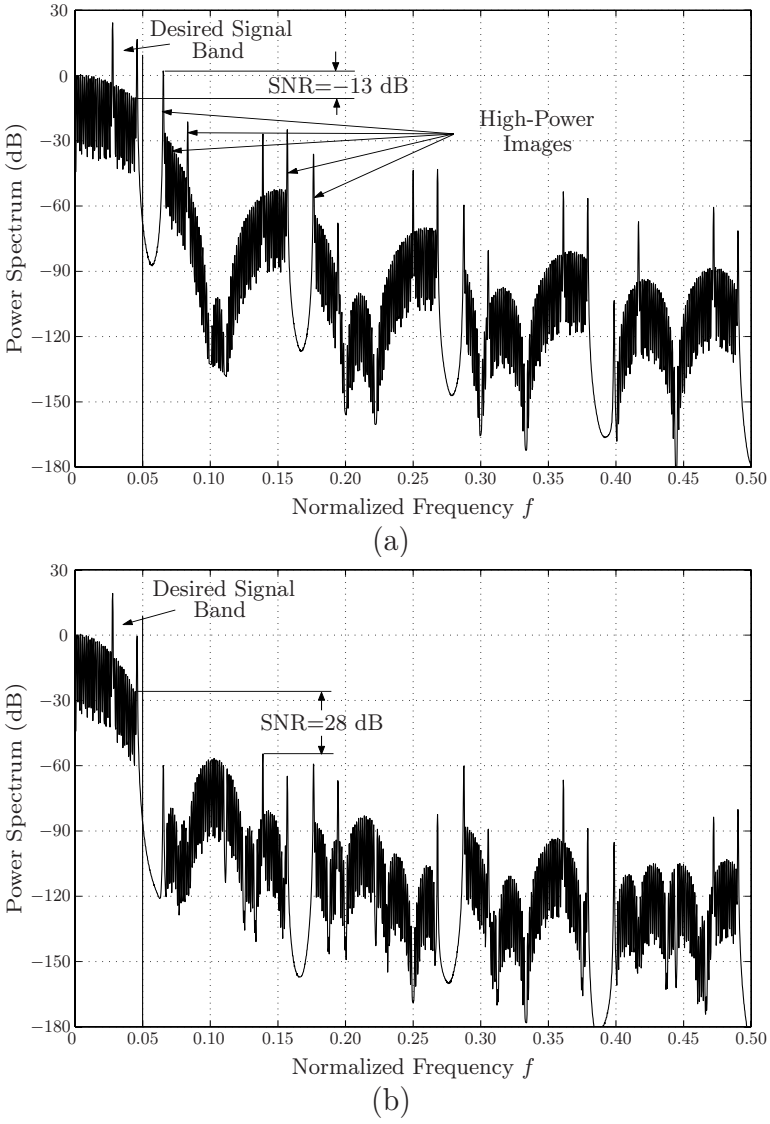


Figure 11: Filtered signals before downsampling of (a) conventional CIC filter having SNR of -13 dB and (b) modified CIC having SNR of +28 dB.

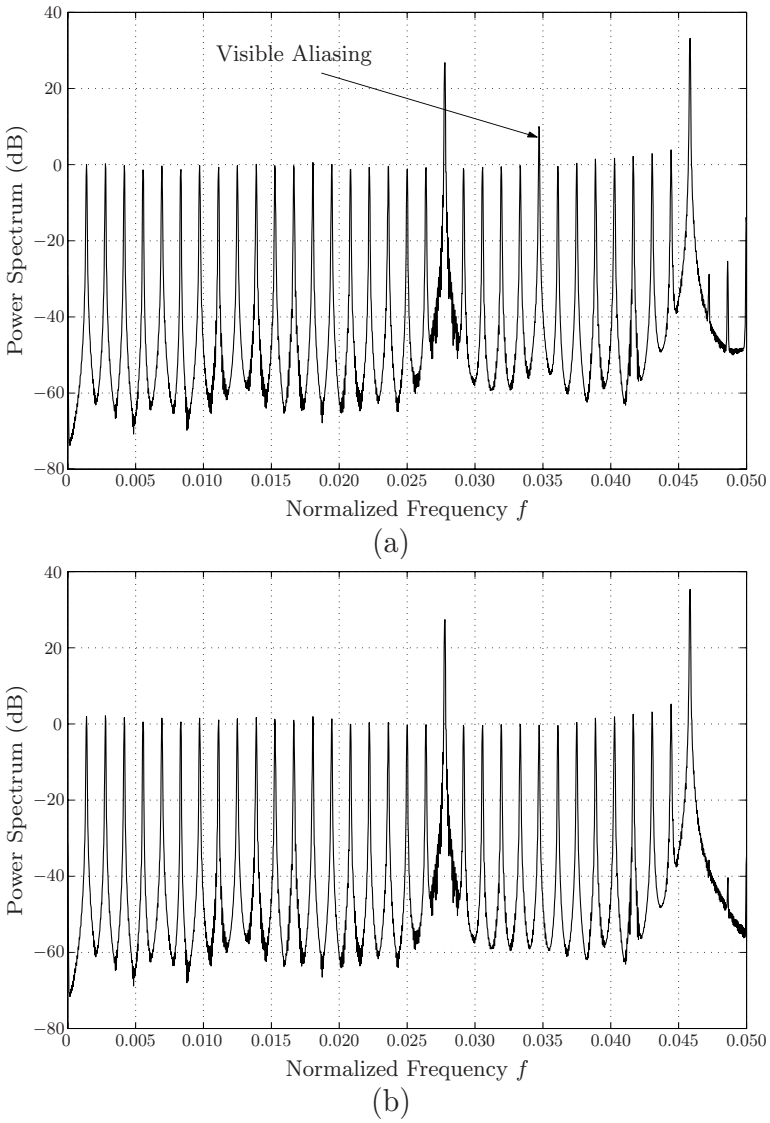


Figure 12: Output signals of (a) conventional CIC showing visible aliasing and (b) modified CIC filter. Both filters are followed by second-order IIR filter to correct for the passband droop of both filters.

CHAPTER IV

EFFICIENT SAMPLE RATE CONVERSION FOR SOFTWARE RADIO SYSTEMS USING HIERARCHICAL COMPUTATION OF OUTPUT SIGNALS

The different methods that have been proposed in the literature for performing SRC in SWR systems generally focus on reducing the sample rate of the received signals by rational conversion factors [1, 37, 39, 43]. This SRC method increases the efficiency of SRC in SWR systems by factors $L/M > 1$ by exploiting the fact that conventional SRC systems require a constant number of computations for evaluating the different output samples of a particular stage in a SRC process. For example, computing each output sample of the two stages in a two-stage conventional SRC process that increases the sample rate of a wideband input signal by a rational factor requires roughly a constant number of computations ignoring possible symmetry of the coefficients of the filters that are used, which may reduce the number of computations for some output samples. Reducing the number of computations for evaluating some of the output samples of one or more stages may result in significant lowering the overall rate of computations required by the complete SRC process. While conventional SRC methods compute the output samples of a particular stage serially, this chapter describes a SRC method that is effective for reducing the computational requirement for SRC processes with rational factors $L/M > 1$ by computing the output sample in a progressive form, allowing the information obtained from the computed samples at a particular instant to be used towards the computation of other samples. This process may reduce the average number of computations per sample for some SRC stages and for the whole SRC process significantly.

The remainder of this chapter is organized as follows. Section 4.1 describes the new SRC method, and Section 4.2 compares its computational requirements to other SRC methods.

Finally, Section 4.3 presents simulation results of this new method.

4.1 The New SRC System

The efficiency of the SRC method discussed in this chapter is generally higher than conventional SRC methods for the computationally demanding case of increasing the sample rate and, therefore, we consider rational SRC factors L/M that are greater than unity. To describe the suggested new SRC method, we will consider first a single-stage conventional SRC process with conversion factor $L/M > 1$, and then discuss extending the process to multi-stage SRC processes. Since L/M is greater than unity, the cutoff frequency of the filter that is required for the single stage SRC process is proportional to $1/L$. Assume that a $(1/L)$th-band lowpass filter [14], which has a cutoff frequency of π/L, is used after the upsampling-by-L process to eliminate the images of the desired baseband signal before downsampling by M. Therefore, the computation of each of the L/M samples that are computed per input sample requires $2A$ multiplications, where A is an integer. The average number of multiplications for computing the output samples of this single-stage SRC process can be reduced by computing a low-rate set of output samples that constitutes a fraction q of the total output samples by using the $(1/L)$th-band lowpass filter. This fraction q of output samples and the input samples can then be used for computing the remaining output samples using a lower number of multiplications per output sample. Further computational reduction may be achieved by applying the procedure described above multiple times in a recursive fashion, as will be discussed later. Unlike a multi-stage SRC process where the input samples to a particular stage are hidden from following stages, the process of computing an output sample in the SRC method described here has access to all previously computed output samples as well as all input samples.

Suppose that a sequence of samples $x(nT_1)$ for which SRC by a factor $L/M > 1$ is to be applied have been obtained by sampling the continuous-time signal $x(t)$ at a rate higher than the Nyquist sampling rate, where n is the sample index and T_1 is the sampling period. A continuous-time lowpass filter with the impulse response

$$h(t) = w\left(\frac{t}{AT_1}\right)\operatorname{sinc}\left(\frac{t}{T_1}\right), \quad -\infty < t < \infty, \tag{7}$$

where $w(t/AT_1)$ is a symmetric window with $w(0) = 1$ and $w(t/AT_1) = 0$ for $|t| > AT_1$, can be used for evaluating the continuous-time waveform $x(t)$ at any desired t from the samples $x(nT_1)$ as

$$x(t) = \sum_{n=\left\lfloor \frac{t}{T_1} \right\rfloor - A+1}^{\left\lfloor \frac{t}{T_1} \right\rfloor + A} x(nT_1)\, w\left(\frac{t-nT_1}{AT_1}\right) \operatorname{sinc}\left(\frac{t-nT_1}{T_1}\right), \tag{8}$$

where $\lfloor t \rfloor$ is the largest integer not grater than t. The filter $h(t)$ with the cutoff frequency $f_c = 1/2T_1$ Hz, is suitable for SRC by factors $L/M > 1$. For factors $L/M < 1$, $h(t)$ can be used only if the bandwidth of the input signal $x(nT_1)$ is limited to a maximum of L/M of the digital band so that aliasing is avoided when the sample rate is reduced. In the following discussion, we consider the case of SRC in SWR systems by rational factors $L/M > 1$. We also restrict the SRC factors to rational values since the ratios of the sample rates of the different air interfaces that SWR systems transceive are always rational numbers. Using (8), SRC can be performed by periodically resampling $x(t)$ at some desired sample rate $1/T_2$, to give

$$x(lT_2) = \sum_{n=\left\lfloor l\frac{T_2}{T_1} \right\rfloor - A+1}^{\left\lfloor l\frac{T_2}{T_1} \right\rfloor + A} x(nT_1)\, w\left(\frac{lT_2-nT_1}{AT_1}\right) \operatorname{sinc}\left(\frac{lT_2-nT_1}{T_1}\right), \tag{9}$$

where l is an integer, and $T_2 = MT_1/L$. Compared to an ideal SRC process, where an infinite number of input samples contribute to each output sample, the image attenuation of the non-ideal SRC process given in (9) is finite and the transition bandwidth is nonzero because only $2A$ input samples are used for computing each sample of the output signal. Different windows can be used in (9), the simplest of which is the rectangular window, which has poor spectral characteristics in the sense that it provides the undesired SRC images with low attenuation. For better SRC performance, windows that are continuous at the edges like the Blackman and Kaiser (with large values of the parameter β) windows can be used [73]. Substituting for T_2 in terms of T_1 and expanding the sinc$(\cdot)$ function in (9) gives

$$x\left(\frac{lMT_1}{L}\right) = \frac{\sin\left(\pi\frac{lM}{L}\right)}{\pi} \sum_{n=\left\lfloor \frac{lM}{L} \right\rfloor - A+1}^{\left\lfloor \frac{lM}{L} \right\rfloor + A} \frac{(-1)^n x(nT_1)\, w\left(\frac{lM-nL}{AL}\right)}{\frac{lM}{L} - n}, \tag{10}$$

where the identity $\sin(t + n\pi) = (-1)^n \sin(t)$ has been used. Excluding the effect of the window $w(t/AT_1)$, (10) shows that an input sample influences the magnitude of an output sample in proportion to the inverse time difference between the two samples. Equation (10) also shows that an output sample with an integer value of l/L is coincident and equal to an input sample and, therefore, need not be calculated.

Now assume that output samples $x(lMT_1/L)$ have already been computed for a small fraction of the possible values of l using (10) for example. The output samples for all the remaining values of l can be represented in terms of the input samples and this set of computed output samples. This representation of $x(lMT_1/L)$ can be obtained by assuming that $x([l+k]MT_1/L)$ for all $k \in \Phi$ have been computed, where $\Phi = \{\phi_1, \phi_2, \cdots, \phi_b\}$ is a set of b nonzero integers with $\phi_1 < \phi_2 < \cdots < \phi_b$ such that $\{l + \phi_1, l + \phi_2, \cdots, l + \phi_b\}$ represents the indexes of all the output samples used along with the input signal for computing the output sample $x(lMT_1/L)$. Representing the $(l+k)$th output sample $x([l+k]MT_1/L)$ in a form similar to that of the lth output sample $x(lMT_1/L)$ in (10) gives

$$x\left(\frac{[l+k]MT_1}{L}\right) = \frac{\sin\left(\pi\frac{[l+k]M}{L}\right)}{\pi} \sum_{n=\left\lfloor\frac{[l+k]M}{L}\right\rfloor-A+1}^{\left\lfloor\frac{[l+k]M}{L}\right\rfloor+A} \frac{(-1)^n x(nT_1)\, w\left(\frac{[l+k]M-nL}{AL}\right)}{\frac{[l+k]M}{L} - n}. \tag{11}$$

Define S_l and S_{l+k} to be the summations appearing in (10) and (11), respectively, i.e.,

$$S_l \equiv \sum_{n=\left\lfloor\frac{lM}{L}\right\rfloor-A+1}^{\left\lfloor\frac{lM}{L}\right\rfloor+A} \frac{(-1)^n x(nT_1)\, w\left(\frac{lM-nL}{AL}\right)}{\frac{lM}{L} - n}, \tag{12}$$

$$S_{l+k} \equiv \sum_{n=\left\lfloor\frac{[l+k]M}{L}\right\rfloor-A+1}^{\left\lfloor\frac{[l+k]M}{L}\right\rfloor+A} \frac{(-1)^n x(nT_1)\, w\left(\frac{[l+k]M-nL}{AL}\right)}{\frac{[l+k]M}{L} - n}. \tag{13}$$

The summation S_{l+k} for different values of $k \in \Phi$ of the sample $x([l+k]MT_1/L)$ represents additional information that can be used for computing S_l, from which the value of the target output sample $x(lMT_1/L)$ can be easily computed. Using the Lagrange interpolation function (LIF) [83], which is defined for a set of points $\{(t_1, f(t_1)), (t_2, f(t_2)), \cdots, (t_N, f(t_N))\}$ as

$$f(t) = \sum_{j=1}^{N} \frac{\prod_{\substack{s=1 \\ s\neq j}}^{N} (t - t_s)}{\prod_{\substack{s=1 \\ s\neq j}}^{N} (t_j - t_s)} f(t_j), \tag{14}$$

we can define $R_\Phi(l)$ as the LIF of the points $(kM/L, S_l - S_{l+k})$, for all $k \in \Phi$, i.e.,

$$R_\Phi(l) \equiv \sum_{k\in\Phi} \left\{ \left[\alpha_k \prod_{\substack{s\in\Phi \\ s\neq k}} \left(\frac{sM}{L}\right) \right] (S_l - S_{l+k}) \right\}, \tag{15}$$

where the coefficients $\{\alpha_k, k \in \Phi\}$ of the Lagrange polynomials satisfy

$$\sum_{k\in\Phi} \left[\alpha_k \prod_{\substack{s\in\Phi \\ s\neq k}} \left(\frac{sM}{L}\right) \right] = 1. \tag{16}$$

Therefore, $R_\Phi(l)$ in (15) simplifies to

$$R_\Phi(l) = S_l - \sum_{k\in\Phi} \left\{ \left[\alpha_k \prod_{\substack{s\in\Phi \\ s\neq k}} \left(\frac{sM}{L}\right) \right] S_{l+k} \right\}. \tag{17}$$

Substituting S_l and S_{l+k} defined in (12) and (13) into (17) permits the representation of $R_\Phi(l)$ in terms of all the input samples that contribute to S_{l+k} for all $k \in \Phi$. If the elements of $\Phi = \{\phi_1, \phi_2, \cdots, \phi_b\}$ satisfy $\phi_1 < 0 < \phi_b$, roughly $2A + M(\phi_b - \phi_1)/L$ input samples will contribute to $R_\Phi(l)$ as shown by

$$R_\Phi(l) = \sum_{n=-A+M\phi_1/L+1}^{A+M\phi_b/L} \lambda_l(n)\, x\left[\left(n + \left\lfloor \frac{lM}{L} \right\rfloor\right) T_1\right]. \tag{18}$$

The coefficient $\lambda_l(n)$ in (18) for a particular n is obtained by collecting the coefficients of the input sample $x[(n + \lfloor lM/L \rfloor)T_1]$ when S_l and S_{l+k}, for all $k \in \Phi$, that are given in (12) and (13), respectively, are substituted into (17). Fig. 13 plots $|\lambda_l(n)|$ against $|n|$ for a Kaiser window ($\beta = 9.5$ and size $2A = 122$) for a virtual SRC factor $L/M = 1$. The figure plots $|\lambda_l(n)|$ for $\Phi = \emptyset$ (i.e., the magnitude of the coefficients of a Kaiser-windowed lowpass filter), and $\Phi = \{-1, 1\}$, $\{-3, -1, 1, 3\}$, $\{-6, -2, 2, 6\}$, $\{-5, -3 - 1, 1, 3, 5\}$, $\{-10, -6, -2, 2, 6, 10\}$, and $\{-20, -12, -4, 4, 12, 20\}$. The plots in Fig. 13 are normalized with respect to the maximum value of each plot. Fig. 13 shows that $|\lambda_l(n)|$ decays at different rates for different Φ, meaning that some terms in the summation in (18) can be dropped if their contribution to $R_\Phi(l)$ is negligible. For other values of L/M, the plots of $|\lambda_l(n)|$ generally have the same form as the plots in Fig. 13, but have different rates of decay.

A Kaiser-windowed ($\beta = 9.5$ and $2A = 122$) lowpass filter, for example, has a transition bandwidth that is 10% of the passband width and provides a stopband attenuation of

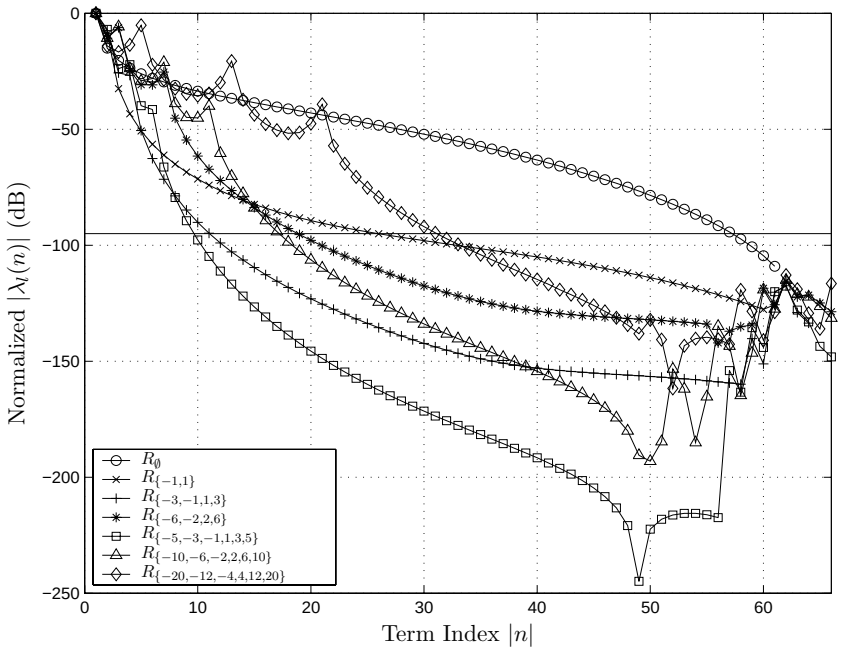


Figure 13: Normalized magnitude of the coefficients $\lambda_l(n)$ in $R_\Phi(l)$ versus $|n|$ for different Φ sets for the Kaiser window of size $2A = 122$ and a virtual value of $L/M = 1$.

approximately 95 dB. For a Kaiser window, plots of the coefficients $|\lambda_l(n)|$ shown in Fig. 13 for $\Phi = \emptyset$, $\{-1, 1\}$, $\{-3, -1, 1, 3\}$, $\{-6, -2, 2, 6\}$, $\{-5, -3-1, 1, 3, 5\}$, $\{-10, -6, -2, 2, 6, 10\}$, and $\{-20, -12, -4, 4, 12, 20\}$ drop below the -95 dB level for $|n|$ greater than or equal to 57, 25, 11, 18, 9, 17, and 31, respectively. Consequently, R_Φ can be computed to a high degree of accuracy using fewer than $2A+M(\phi_b-\phi_1)/L$ terms such that the error in computations caused by dropping terms is much less than the error introduced by the finite precision of the computations. Due to the small discontinuities at the Kaiser window edges, some plots of $|\lambda_l(n)|$ in Fig. 13 experience a sudden increase as $|n|$ approaches A, which is equal to 61 in this case. Careful inspection of these plots for $|n|$ near 61 reveals that they remain below -112 dB, which is much lower than the stopband attenuation of the filter and, therefore, the effect of this sudden increase of $|\lambda_l(n)|$ on the accuracy of the computed output samples is negligible.

Using (17), we can write S_l in terms of $R_\Phi(l)$ and S_{l+k} for $k \in \Phi$, and substitute the result into (10). This results in

$$S_l = \sum_{k\in\Phi} \left[\alpha_k \prod_{\substack{s\in\Phi \\ s\neq k}} \left(\frac{kM}{L} \right) \right] S_{l+k} + R_\Phi(l), \tag{19}$$

and

$$x\left(\frac{lM}{L}T_1\right) = \frac{\sin\left(\pi\frac{lM}{L}\right)}{\pi}\, S_l. \tag{20}$$

Using (19) and (20) yields a systematic method for computing the target output sample $x(lMT_1/L)$ from the input samples that are embedded in $R_\Phi(l)$ and the additional information given by S_{l+k} for all $k \in \Phi$. Since $|\lambda_l(n)|$ for small values of $|n|$ and Φ different from $\emptyset$ generally decays faster than the decay of the coefficients of a windowed lowpass filter, $R_\Phi(l)$ can be computed to a specified accuracy by using only $2B$ ($\ll 2A$) terms. Fig. 13 shows that the value of B required to achieve the desired accuracy generally decreases as size b of the set Φ increases and as the magnitude of its elements decrease. The minimum required value of B is determined by the value of L/M, the desired SRC image attenuation, and the error due to finite-precision computations. Proper selection of the value of B will guarantee that the contribution of the summation terms in (18) to $R_\Phi(l)$ for which $n < -B+1$ and $n > B$ is negligible and, therefore, these terms can safely be dropped from the summation. Fig. 13 and similar figures for other values of L/M can be used for estimating B, where B is roughly equal to the minimum n above which the value of $|\lambda_l(n)|$ for a given Φ drops sufficiently below the desired image attenuation or the finite-precision error.

Reducing the number of terms in the summation of $R_\Phi(l)$ in (18) to $2B$ terms, and substituting it into (19) and (20) results in the approximation

$$x\left(\frac{lM}{L}T_1\right) \approx \frac{\sin\left(\pi\frac{lM}{L}\right)}{\pi}\hat{S}_l, \tag{21}$$

where $\hat{S}_l$ is an approximation of S_l given by

$$\hat{S}_l = \sum_{k\in\Phi}\left[\alpha_k \prod_{\substack{s\in\Phi\\ s\neq k}}\left(\frac{sM}{L}\right)\right] S_{l+k} + \sum_{n=-B+1}^{B} \lambda_l(n)\, x\left[\left(n+\left\lfloor\frac{lM}{L}\right\rfloor\right)T_1\right]. \tag{22}$$

Assuming that S_{l+k} for a particular l and all $k \in \Phi$ are available, (21) and (22) can be used for computing a highly accurate value of $x(lMT_1/L)$ by using a total of $2B + b + 1$ multiplications, where all coefficients are pre-computed for all values of (l mod L). The multiplication by the periodic $\sin(\pi lM/L)/\pi$ in (21) can be embedded into the coefficients $\alpha_k \prod_{\substack{s\in\Phi\\ s\neq k}}\left(\frac{lM}{L}\right)$ and $\lambda_l(n)$ in (22) to reduce the number of multiplications to $2B+b$. Therefore,

this SRC method discussed here requires the computation of a low-rate sequence of S_l values at nonconsecutive periodic series of l values using (12). Such a low-rate sequence of S_l values can then be used for evaluating the corresponding output samples $x(lMT_1/L)$ using (20) as well as for getting the approximate values of S_l using (22), which in turn can be used for evaluating other output samples. Multiple steps of this process can be performed until S_l (or its approximation $\hat{S}_l$) becomes available for all l so that $x(lMT_1/L)$ can be evaluated for all l.

To compute the SRC output samples $x(lMT_1/L)$ using the SRC method described here, the sequence of the required S_l or its approximation, $\hat{S}_l$, can be computed in multiple steps to increase the efficiency of the SRC process. This allows the use of the results obtained from previous steps in the following steps. Consider that the computation of S_l or $\hat{S}_l$ for all integer values l are computed in P steps. We define Ψ_j, for $j = 1, 2, \cdots, P$, to be different sets consisting of all the values of l at which S_l (or $\hat{S}_l$) have been already computed and are available at the beginning of the jth step. The computation of S_l for a particular l in the jth step is then performed using any set $\Phi = \{\phi_1, \phi_2, \cdots, \phi_b\}$ that satisfies

$$\{l + \phi_1, l + \phi_2, \cdots, l + \phi_b\} \subset \Psi_j. \tag{23}$$

That is, $R_\Phi(l)$ in (17) is represented using S_{l+k} for values of $k \in \Phi$, or equivalently $l+k \in \Psi$, only. It is important to note that different Φ sets may be required for the computation of S_l for different values of l depending on the relative position of S_l to S_{l+k} that are available. Higher computational efficiency of our SRC method is achieved by optimizing the elements of the Ψ_j sets. Increasing the sparsity of the elements in Ψ_j will increase the values of B and b that are required for achieving a specified computational accuracy of S_l in the jth step. By trading-off the sparsity of the sets Ψ_j, the computations of the SRC method can be optimized. Fig. 14 shows a flow chart of the steps in the SRC method described here, where the flow of computations goes from left to right and top to bottom. Proper delays may be required to guarantee the availability of all the required intermediate results before carrying out the different computations.

Computing S_l for specific values of l may be more computationally efficient than other

values of l. The increased efficiency in computing S_l at particular values of l results from using symmetry of the coefficients of the input samples in (12) to reduce the number of multiplications, where the number of multiplications required for computing S_l may be reduced by approximately one half. This coefficient symmetry can be used to provide a more efficient SRC process for SRC factors that are generally close to unity. In this scenario, a stage of the SRC process with factor L/M can be broken into two stages with SRC factors L_1/M_1 and L_2/M_2 such that $L/M = (L_1/M_1) \cdot (L_2/M_2)$. The SRC factor L_1/M_1 of the first stage can be chosen such that S_l^* for some or all integer l take advantage of the symmetry and therefore can be computed efficiently. Following the computation of this efficiently computed sequence S_l^*, the second SRC stage is applied where another sequence S_l is computed at low cost and directly used for obtaining the output samples $x(lMT_1/L)$. For example, the computation of S_l^* is efficient for $L_1/M_1 = 2$, where lM_1T_1/L_1 lies over an epoch of an input sample for even values of l and halfway between epochs $\lfloor l/2 \rfloor T_1$ and $(\lfloor l/2 \rfloor + 1)T_1$ of the $(\lfloor l/2 \rfloor)$th and $(\lfloor l/2 \rfloor + 1)$th input samples for odd values of l, respectively. Therefore, the computation of S_l^* in the first stage for any odd value of l can be performed using a modified form of (12) that takes symmetry in consideration. This modified form of (12) at $l = 2i + 1$ for integer i is given by

$$S^*_{l=2i+1} = \sum_{n=1}^{A} \frac{(-1)^n w\left(\frac{1-2n}{2A}\right)}{\frac{1}{2} - n} \{x([2i+n]T_1) - x([2i-n+1]T_1)\} \,. \tag{24}$$

The approximation of an S_l value given in (22) can also be modified to take advantage of symmetry for a symmetric Φ, i.e., the elements of Φ are symmetric about zero, as

$$\begin{aligned} \hat{S}^*_{l=2i+1} \;\approx\; & \sum_{k \in \Phi^*} \left[\alpha_k \prod_{\substack{s \in \Phi \\ s \neq k}} \frac{s}{2} \right] \{S_{2i+1+k} + S_{2i+1-k}\} \\ & + \sum_{n=1}^{B} \lambda_{2i+1}(n) \{x([2i+n]T_1) - x([2i-n+1]T_1)\} \,, \end{aligned} \tag{25}$$

where $\Phi^* = \{\phi_1, \phi_2, \cdots, \phi_{b/2}\}$. The sequence of S_l^* for a sparse set of odd values of l can be computed using (24) and then used for computing the approximations $\hat{S}_l^*$ of S_l^* for other values of l using (25) in multiple steps until S_l^* or $\hat{S}_l^*$ have been computed for all odd l. The second stage of the SRC process can then be applied for computing S_l and the output

samples using the results of the first stage. The efficiency of computing S_l^* at odd l implies that for roughly the same cost twice as many S_l^* values can be computed using (24) as S_l using (12) may be computed. This generally reduces the SRC computational cost without sacrificing performance.

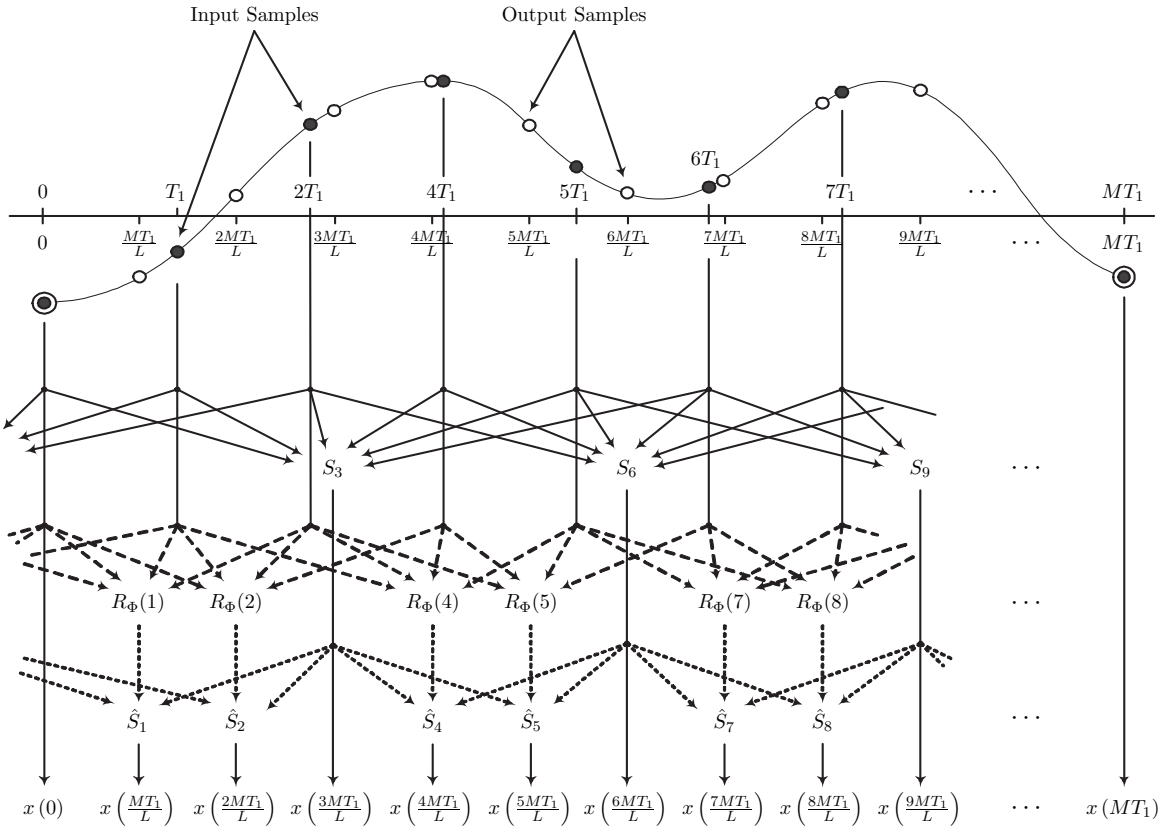


Figure 14: Flow of computations in a typical implementation of the SRC method described in this chapter, where computations flow from left to right and top to bottom. Proper delays are required before performing computations to guarantee that intermediate results are available. Solid lines from input samples to S_3, S_6, S_9 are computations in (12), dashed lines from input samples to $R_\Phi(1), R_\Phi(2), \cdots, R_\Phi(8)$ are computations in (18), and dotted lines from S_3, S_6, S_9 and $R_\Phi(1), R_\Phi(2), \cdots, R_\Phi(8)$ to $\hat{S}_1, \hat{S}_2, \hat{S}_4, \hat{S}_5, \hat{S}_7, \hat{S}_8$ are computations in (22).

4.2 Computational Requirements of the New SRC Method

The computations requirements of the new SRC method are determined by several factors, including the choice of conversion factors for the different SRC stages, the desired spurious-free dynamic range (SFDR) of the SRC process, and the bandwidth, oversampling ratio,

and sample rate of the input signal. In a two-stage SRC process, for example, the efficiency of this method compared to a conventional SRC method generally increases as the ratio of the conversion factors in the first to second stages increases, as the SFDR increases, and as the oversampling ratio of the SRC input signal decreases, i.e., as the sample rate approaches the Nyquist rate.

The computational requirements of the new SRC method will be compared to conventional single- and two-stage SRC methods that use equiripple FIR lowpass filters in all stages. The order of an FIR lowpass filter that is required for attenuating the undesired SRC images by a fixed amount in a particular stage of a conventional SRC process with factor $L/M > 1$ is proportional to the value of L and inversely proportional to the transition bandwidth [14]. Therefore, a conventional single-stage SRC process that requires a filter of order $2CL$ will perform $2C$ MPOS, where the value of C depends on the oversampling ratio of the input signal and the maximum passband and stopband ripples. This can be extended to a two-stage SRC process with first- and second-stage conversion factors of $L_1/M_1 > 1$ and $L_2/M_2 > 1$, respectively, where the overall conversion factor is $L/M = (L_1/M_1) \cdot (L_2/M_2)$. Assuming $(1/L_1)$th- and $(1/L_2)$th-band lowpass filters of orders $2DL_1$ and $2EL_2$ are used for the first and second stages, respectively, where D and E depend on the oversampling ratio of the input signal and the maximum passband and stopband ripples, the first and second stages would perform $2DM_2/L_2$ and $2E$ MPOS, respectively. Therefore, the total required MPOS for this two-stage SRC process is $2DM_2/L_2+2E$. The values of C for the single-stage and D for the two-stage SRC processes are typically close to the required value of A in the SRC process discussed here.

Our SRC method may significantly reduce the computations required for SRC compared to a conventional single- or multi-stage SRC approach. Assume that the computation of S_l for all the required values of l in a particular SRC stage is performed in $P + 1$ hierarchical steps indexed $0, 1, \cdots, P$. In the zeroth step, a fraction q_0 of all S_l values are computed using (12) at a cost of $2A$ multiplications for each computed S_l. The jth step, where $j = 1, 2, \cdots, P$, of the proposed method then approximates a fraction q_j of all the S_l values using (22) at an average cost of $2\bar{B}_j + \bar{b}_j$ multiplications per computed S_l, where $\bar{B}_j$ and

$\bar{b}_j$ represent averages of B_j and b_j. The use of averages of B_j and b_j is attributed to the fact that approximating S_l may require different Φ sets for different values of l, which may result in the use of different values of B_j and b_j for different values of l. Finally, one MPOS may be required for obtaining an output sample from S_l or its approximation for any value of l. Therefore, this SRC stage performs a maximum number of MPOS of

$$\text{MPOS} = 2q_1A + 1 + \sum_{j=1}^{P} q_j \left(2\bar{B}_j + \bar{b}_j\right). \tag{26}$$

The use of symmetry as described by (24) and (25) may further reduce the MPOS given by (26) for some SRC factors. The required MPOS of the new SRC method in (26) assumes that all the coefficients of the input samples in (12) and (22) and all the coefficients of S_{l+k} for all k in (22) for the different Φ sets that would be used are pre-computed and stored. The Φ sets required for different output samples are periodic if each step of the process computes S_l for a periodic set of l values. Therefore, only a finite number of coefficients need to be computed and stored.

To illustrate the computational efficiency of the SRC method discussed here, we consider two different SRC processes with factors 7 and 7/5 and compare the MPOS required by the method described here for performing these processes to the MPOS required by conventional SRC methods. Configuration I of the proposed method is a single-stage five-step SRC process. For the SRC factor $L/M = 7$, the zeroth step of this configuration evaluates S_l for $l = 24i$ at all integer i using a Kaiser window with $A = 61$ and $\beta = 9.5$. Then, the values of S_l that have not been computed in previous steps for $l = 12i$, $l = 8i$, $l = 3i$, and $l = i$ at all integer i are computed in steps 1, 2, 3, and 4, respectively, using the average values $\bar{B}_1 = 19.58$, $\bar{B}_2 = 7.58$, $\bar{B}_3 = 4.87$, $\bar{B}_4 = 2.76$ and $\bar{b}_1 = 8.55$, $\bar{b}_2 = 9.00$, $\bar{b}_3 = 7.13$, $\bar{b}_4 = 7.45$. Taking into consideration the ratios of the total S_l that are computed in each step, it is seen that this configuration requires on average 19.35 MPOS. Compared to a single-stage conventional SRC process with $L/M = 7$ that uses a filter with $C = 57$, the required MPOS is 114 MPOS. A two-stage process with $L_1/M_1 = 3$ and $L_2/M_2 > 7/3$ would require values of $D = 58$ and $E = 4$, which result in the SRC process requiring 57.71 MPOS. Since the new SRC method requires optimization of the different values of l at which S_l is computed

and the required averages $\bar{B}_j$ and $\bar{b}_j$ in the different steps of the different stages, evaluating the MPOS for the new SRC method at different values of L/M is nontrivial.

Configuration II of the SRC method described above uses a two-stage process that depends on coefficient symmetry illustrated by (24) and (25), where the conversion factor of the first stage is 2. This SRC method uses a four-step process in the first stage and a single step in the second stage. The zeroth step of the first stage evaluates $\hat{S}_l$ at $l = 8i$ for all integer i that do not correspond to input samples by using (12) with $A = 61$. Step 1 then computes $\hat{S}_l$ for all $l = 4i$ that have not been computed with $\Phi_1 = \{-20, -12, -4, 4, 12, 20\}$ and $B_1 = 34$. Step 2 computes $\hat{S}_l$ for $l = 2i$ with $\Phi_2 = \{-10, -6, -2, 2, 6, 10\}$ and $B_2 = 19$. Using all previously computed $\hat{S}_l$, Step 3 computes $\hat{S}_l$ for $l = i$ with $\Phi_3 = \{-5, -3, -1, 1, 3, 5\}$ and $B_3 = 10$. Performing this step gives known values of $\hat{S}_l$ for all integers l. Therefore, the second stage of the process with a single step computes S_l of the second stage for all integer l that correspond to the desired output sample using a small value of $A = 14$. Since the fractions q_0, q_1, q_2, and q_3 of all $\hat{S}_l$ values computed in the different steps of the first stage are equal to 1/8, 1/8, 1/4, and 1/2, respectively, this configuration requires $24.25M/L + 15(L-1)/L$ MPOS on average. A two-stage conventional SRC system that provides similar performance with a half-band filter in the first stage would have $D = 60$ and $E = 7$ and, therefore, perform an average of $60M/L+14$ MPOS. This demonstrates that the new SRC method in this case requires less MPOS than a conventional two-stage SRC method with a half-band filter in the first stage for a range of SRC factors.

4.3 Simulation Results

Configuration I of the SRC process described in Section 4.2 is applied to an input signal for SRC by a factor $L/M = 7$, where a minimum image attenuation of 95 dB and a transition bandwidth equal to 10% of the passband are desired. The wideband signal that is shown in Fig. 15, which has an amplitude resolution of 19 bits and is constructed by adding 38 equal-power frequency division multiplexed (FDM) channels that occupy 95% of the available digital band, is input to this configuration. The output signal of the SRC process is shown in Fig. 16. It is seen that our SRC provides all image spurs with an attenuation of at least

95 dB relative to the peak power of the desired baseband signals, and therefore provides a SFDR of 95 dB using 19.35 MPOS on average compared to 114 and 57.71 MPOS for the single- and two-stage conventional SRC methods, respectively, at the same SRC factor $L/M = 7$.

The input signal in Fig. 15 is also applied to Configuration II of the new SRC method described in Section 4.2, where the intermediate high-sample-rate and output signals of this configuration are shown in Fig. 17(a) and 17(b), respectively. This configuration performs SRC by a factor of $L/M = 7/5$ in two stages. The intermediate high-sample-rate signal indicates that this configuration also achieves a SFDR of 95 dB using 30.18 MPOS. The computational requirement of Configuration II is significantly lower than the computational requirement of the two-stage SRC method with a half-band filter in the first stage, which requires 56.85 MPOS. All the computations in the two examples given in this section are performed using fixed-point operations, where the signals, including all intermediate results, are represented with 19 bits of accuracy. Lowering the accuracy of computations below 19 bits will reduce the achievable SFDR of the new and traditional SRC methods below the desired 95 dB, because the quantization noise becomes dominant over the noise of the 95 dB attenuated images when the accuracy of computations is lower than 19 bits.

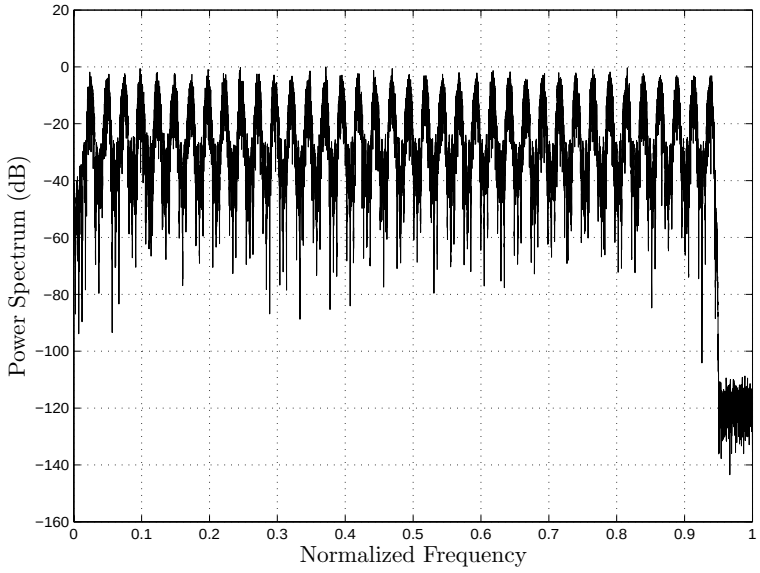


Figure 15: The input signal to the suggested SRC system occupying 0.95 of the available digital band.

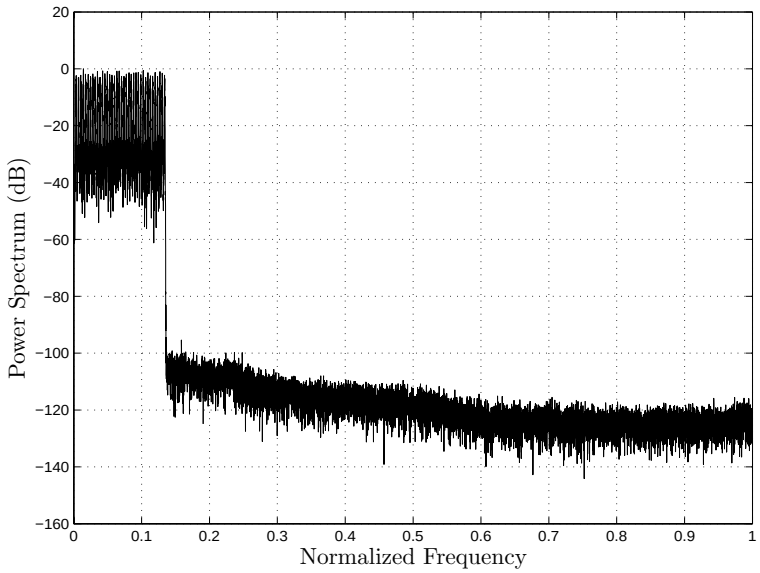
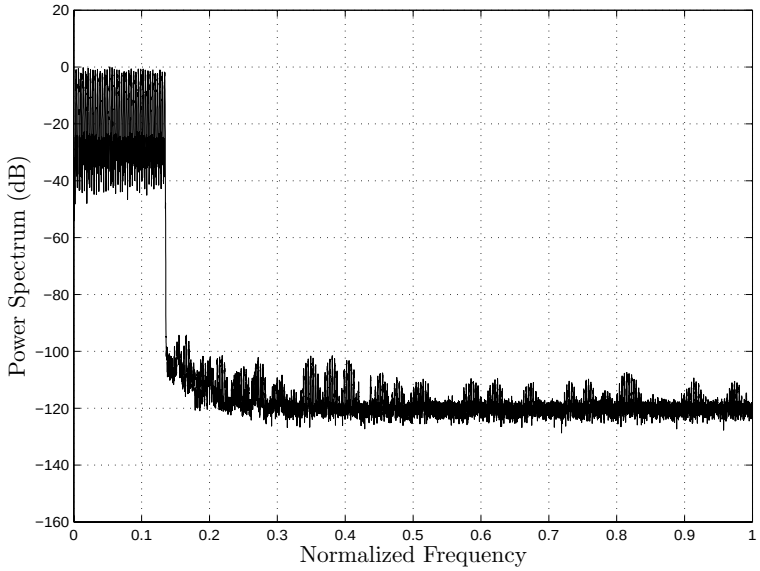
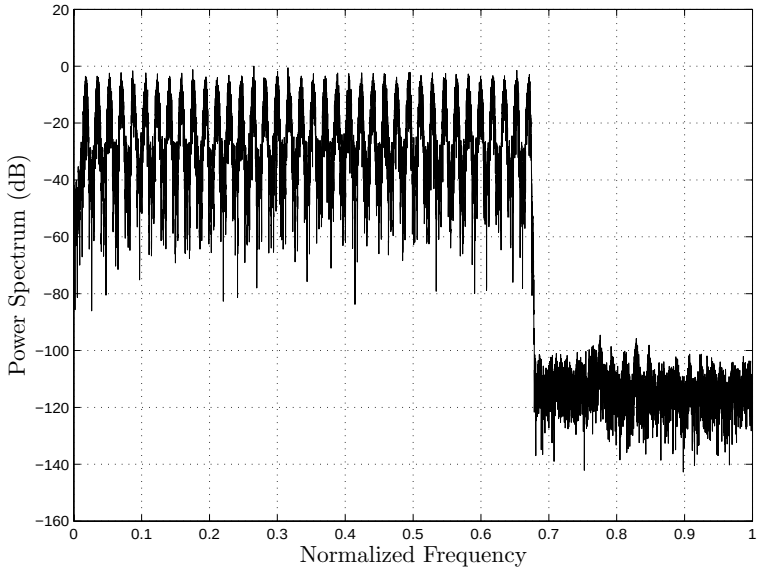


Figure 16: The output signal of Configuration I of the suggested SRC method with SRC factor $L/M = 7$. This configuration is a single-stage five-step process that requires only 19.35 MPOS on average.



(a)



(b)

Figure 17: (a) The filtered high-sample-rate signal and (b) output signal of Configuration II of the suggested SRC system for $L/M = 7/5$. The system uses $\Phi_2 = \{-20, -12, -4, 4, 12, 20\}$ with $B_2 = 34$, $\Phi_3 = \{-10, -6, -2, 2, 6, 10\}$ with $B_3 = 19$, $\Phi_4 = \{-5, -3, -1, 1, 3, 5\}$ with $B_4 = 10$ in the first stage and uses $\Phi_0 = \{-2, -1, 0, 1, 2\}$ with $B_0 = 5$ for the second stage. This configuration requires 30.18 MPOS on average.

CHAPTER V

EFFICIENT WIDEBAND CHANNELIZER FOR SOFTWARE RADIO SYSTEMS USING MODULATED PERFECT RECONSTRUCTION FILTER BANKS

Due to the variations in the propagation environment of the FDM channels in the SWR received signal and the desired SWR flexibility obtained by using single wideband ADC and DAC, SWR channelizers are forced to operate on channels with high dynamic ranges and high sample rates. The signals processed by an SWR transceiver may contain the complete transmission band of a wireless air interface with a dynamic range in excess of 90 dB and bandwidth of 25–150 MHz, i.e., a sample rate of 350 Msamples/s. Satisfying the computational requirements for fully channelizing or synthesizing such signals may demand the use of a large number of currently available DSPs for the SWR base station transceiver functionality [6, 39, 63, 85].

This chapter describes a simple channelization method that is capable of significantly reducing the number of computations for channelization in cases where polyphase decomposition of the channelizer filters is inapplicable and discrete filter bank (DFB) channelizers must be used. The new channelizer can also be modified for synthesizing a set of baseband channels into a wideband signal for transmission. As for DFB channelizers in SWR systems, SRC may be required after this channelizer to change the sample rates of the channels to the proper rates required for further baseband processing. The new channelizer uses modulated perfect reconstruction (PR) filter banks. Because the required modulated PR filter banks have extremely high number of subchannels and prototype filters with high stopband attenuations, design methods of these filter banks that are described in the literature are generally impractical. A procedure that is based on an efficient PR filter bank design method in the literature is suggested for designing the required modulated PR filter banks

in the next chapter.

The remainder of this chapter is outlined as follows. Section 5.1 discusses the channelization of SWR signals. In Section 5.2, the modulated filter banks that are used in the new channelizer are described. Section 5.3 describes the channelization method. Section 5.4 compares the computations for the new and the DFB channelization methods, and Section 5.5 shows simulation results of the new channelizer.

5.1 Channelization of SWR Signals

Extracting a subset of the N channels contained in the received SWR wideband signal can be performed efficiently by polyphase decomposition of the filters of modulated DFB channelizers when the desired sample rate of the extracted channels is $1/L$ of the sample rate of the input signal, such that $L = K \times N$ is satisfied for an integer K, and the digital band is equally divided among all N channels [117,119,120]. Since SWR systems are expected to transceive channels of different air interface specifications simultaneously, the bandwidths of the different channels transceived at a particular time may be different and therefore, these conditions may are not satisfied. In this case, polyphase decomposition of the filter bank may be inapplicable and DFB channelizers may be the only applicable method for channelizing the signal. Compared to the DFB channelizer, polyphase filter bank (PFB) channelizers reduce the required computations for channelization by a factor roughly equal to the number of extracted channels.

The ability of SWR systems to efficiently channelize signals with variable-bandwidth channels, where the conditions for polyphase decomposition of the filters of DFB channelizers are not satisfied, may be necessary to provide the flexibility to process single- or multi-standard signals having channels of differing bandwidths [26]. Various filter bank channelization methods that may be applicable for channelizing specific single- or multi-standard SWR signals have been proposed [26,117–120]. However, the only known channelization method for SWR signals that contain single- or multi-standard channels of differing bandwidths where no restriction is placed on the individual channels is the DFB channelizer. The channelizer described in this chapter is capable of channelizing SWR signals that

contain single- or multi-standard channels with nonequal bandwidths at an efficiency that is much higher than that of the DFB channelizer, especially for large number of extracted channels.

5.2 Exponentially-Modulated Perfect Reconstruction Filter Banks

The design and implementation of PR and near-PR finite impulse response (FIR) filter banks have been extensively studied in the literature [16, 17, 31, 48, 68–72, 87, 104–106, 109, 115]. We are interested in this chapter in a class of filter banks known as modulated PR filter banks. Since the filters of modulated PR filter banks are obtained by modulating a prototype filter, they can be efficiently constructed using polyphase decomposed forms of the filter banks. It was shown in [57, 78, 108] that modulated PR filter banks exist for linear-phase prototype filters of length $2M$, where M is the number of subchannels in the cosine-modulated filter bank. This result was extended in [58] for prototype filters of length $2mM$, where m is an arbitrary positive integer, and an efficient method for designing prototype filters of modulated PR filter banks was proposed in [51]. Compared to the quadrature mirror filter (QMF) bank design methods, the design method of [51] requires the optimization of at most half the number of parameters that are optimized in pseudo-QMF bank design methods, and a smaller fraction of the number of parameters that are optimized in modulated-QMF bank design methods for similar filter lengths.

Most discussion on modulated PR filter banks in the literature is dedicated to cosine-modulated PR filter banks. In the efficient method for designing prototype filters of cosine-modulated PR filter banks proposed in [51], length-$2mM$ prototype filters $h(n)$ are designed by optimizing $m\lfloor M/2 \rfloor$ parameters. The kth filters, $\hat{h}_k(n)$ and $\hat{f}_k(n)$, of the analysis and synthesis sections, respectively, of the filter bank in this design method are obtained by cosine modulating the real-valued linear-phase filter $h(n)$ as

$$\begin{aligned} \hat{h}_k(n) &= 2h(n)\cos\left[(2k+1)\tfrac{\pi}{2M}\left(n-\tfrac{2mM-1}{2}\right)+(-1)^k\tfrac{\pi}{4}\right], \\ \hat{f}_k(n) &= 2h(n)\cos\left[(2k+1)\tfrac{\pi}{2M}\left(n-\tfrac{2mM-1}{2}\right)-(-1)^k\tfrac{\pi}{4}\right], \end{aligned} \tag{27}$$

for $n = 0, 1, \cdots, 2mM-1$, and $k = 0, 1, \cdots, M-1$. Due to the cosine modulation, the PR

filter bank requires only M branches for PR of the input signal.

In SWR systems, the real-valued analog signal received by the antenna is eventually converted into a complex-valued digital signal $X(z)$ that is feed to the channelizer in the form of real-valued in-phase and quadrature signals with bandwidths that may be close to half the sample rates and therefore, the bandwidth of $X(z)$ may be close to the sample rate. The use of cosine-modulated PR filter banks in the channelizer discussed here, which is based on modulated PR filter banks, causes aliasing of the components of $X(z)$ that are in the digital frequency bands $[0, \pi]$ and $[\pi, 2\pi]$ over each other in the channelized output signals. Consequently, the complex-exponential modulation (CEM) is used in the PR filter banks of this channelizer to eliminate this aliasing at the expense of requiring $2M$ branches in the filter bank compared to M branches for the cosine-modulated PR filter bank. It is shown in the Appendix that the cosine modulation in (27) is actually a form of the CEM. As a result, PR prototype filters designed for cosine-modulated PR filter banks are also applicable for CEM PR filter banks. The CEM that will be used for the channelizer is given by

$$h_k(n) = f_k(n) = h(n)e^{j\frac{k\pi}{M}\left(n-\frac{2mM-1}{2}\right)} = h(n)W_{2M}^{-k\left(n-\frac{2mM-1}{2}\right)}, \tag{28}$$

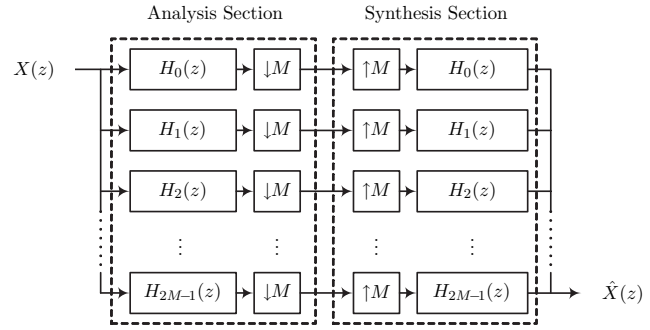
where $W_M = e^{-j2\pi/M}$, $n = 0, 1, \cdots, 2mM–1$, and $k = 0, 1, \cdots, 2M–1$. The discrete filter form of the CEM filter bank is shown in Fig. 18(a), and its output can be expressed in terms of the input $X(z)$ as

$$\hat{X}(z) = \frac{1}{M}\sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{2M-1} H_k\left(zW_M^p\right) H_k(z), \tag{29}$$

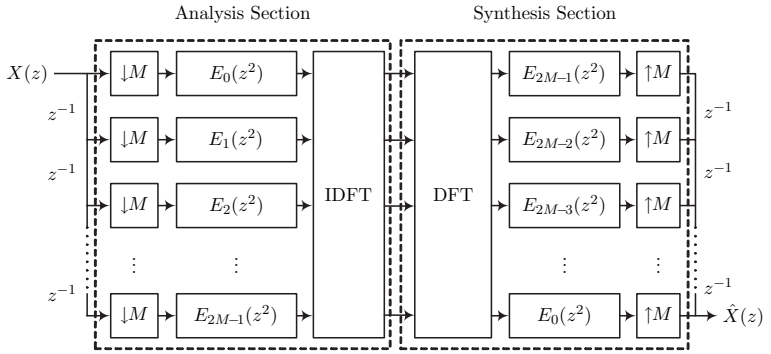
where $H_k(z)$ is the z-transform of $h_k(n)$. The signal $\hat{X}(z)$ in (29) can be decomposed into two parts as

$$\hat{X}(z) = \frac{1}{M}X(z)\sum_{k=0}^{2M-1} H_k^2(z) + \frac{1}{M}\sum_{p=1}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{2M-1} H_k\left(zW_M^p\right) H_k(z), \tag{30}$$

where the first term is the desired signal and the second term is the aliasing signal. The aliasing term nearly cancels out when $H(z)$, from which all filters of the bank are derived, is a PR prototype filter. The remaining aliasing signal has a power spectrum density (PSD) that is extremely lower than the PSD of the desired signal over the complete frequency spectrum [51].



(a)



(b)

Figure 18: The CEM PR filter bank in (a) DFB form, and (b) PFB form.

To increase the computational efficiency of the CEM filter bank of Fig. 18(a), the analysis and synthesis filters are expressed in terms of the length-m polyphase components of $H(z)$ by writing

$$\begin{aligned} H_k(z) &= F_k(z) = \sum_{n=0}^{2mM-1} h(n) W_{2M}^{-k\left(n-\frac{2mM-1}{2}\right)} z^{-n} \\ &= \sum_{q=0}^{2M-1} \sum_{r=0}^{m-1} h(q+2Mr) W_{2M}^{-k\left(q+2Mr-\frac{2mM-1}{2}\right)} z^{-q-2Mr} \\ &= \sum_{q=0}^{2M-1} W_{2M}^{-k\left(q-\frac{2mM-1}{2}\right)} z^{-q} \underbrace{\sum_{r=0}^{m-1} h(q+2Mr) z^{-2Mr}}_{E_q(z^{2M})}. \end{aligned} \tag{31}$$

Denoting $E_q(z)$ to be the qth polyphase component of $H(z)$ permits writing $H_k(z)$ and $F_k(z)$ as

$$H_k(z) = F_k(z) = W_{2M}^{k\frac{2mM-1}{2}} \sum_{q=0}^{2M-1} E_q(z^{2M}) W_{2M}^{-kq} z^{-q}, \tag{32}$$

which allows the representation of the analysis and synthesis filters as sums of the modulated $2M$ polyphase components of $H(z)$. The CEM of the polyphase components in the analysis and synthesis sections can be performed easily using a $2M$-point inverse discrete Fourier transform (IDFT) and a $2M$-point discrete Fourier transform (DFT), respectively. For highly composite values of $2M$, especially power-of-two values, the Cooley-Tukey $2M$-point inverse fast Fourier transform (IFFT) and fast Fourier transform (FFT), respectively, can be used instead [73]. The polyphase decomposed CEM filter bank is shown in Fig. 18(b). For consistency with most of the literature, the number of subchannels that is associated with a PR prototype filter in this chapter will be set to the number of subchannels in the corresponding cosine-modulated filter bank. Therefore, M-subchannel PR prototype filters can be used in either M-subchannel cosine-modulated filter banks or $2M$-subchannel CEM filter banks.

5.3 The New Channelization Method

The use of DFB channelizers for channelizing SWR signals is computationally demanding, especially for channelizing high dynamic-range wideband SWR signals composed of a large number of FDM channels. Although polyphase decomposition of the channelizer filter bank can significantly reduce the number of computations for channelization, it is applicable only if some stringent conditions are satisfied. The strict conditions for using PFB channelizers imply a limitation in the ability of SWR systems to receive and process single- or multi-standard signals comprised of channels having differing bandwidths. The traditional method for extracting N_e channels from a signal with N variable-bandwidth channels is to use a DFB channelizer with N_e variable bandwidth filters and constant transition bandwidths equal to the minimum guard bandwidth in the signal. The required number of computations for the DFB channelizer may be impractical for large values of N_e.

In this section, a new channelizer is described, which can channelize large-N signals that do not satisfy the conditions for polyphase decomposition of the DFB. It is assumed that the N channels in the complex-valued input signal $X(z)$ occupy most of the digital band $[0, 2\pi]$, and that the channels $X_i(z)$, for $i = 0, 1, \cdots, N-1$, possibly have different channel

and guard bandwidths. Moreover, the channels are generally compacted, where the guard bands are small compared to the channel bandwidths. Let the spectrum of the ith channel, $X_i(z)$, in $X(z)$, for $i = 0, 1, \cdots, N-1$, satisfy

$$X_i(e^{j\omega}) = \begin{cases} X(e^{j\omega}), & \omega_i^l \le \omega \le \omega_i^u \\ 0, & \text{elsewhere} \end{cases}, \tag{33}$$

where ω_i^l and ω_i^u are the lower and upper frequency limits of $X_i(z)$, respectively, such that $\omega_i^l < \omega_i^u < \omega_{i+1}^l$. The ith guard bandwidth G_i is defined to be the width of the band between the ith and $(i+1)$th channels, or

$$G_i = \omega_{i+1}^l - \omega_i^u, \qquad i = 0, 1, 2, \cdots, N-1, \tag{34}$$

where $\omega_N^l = \omega_0^l + 2\pi$. Let $G_{\min}$ be the minimum among all N guard bands, i.e.,

$$G_{\min} = \min_{i=0,1,\cdots,N-1} \left(\omega_{i+1}^l - \omega_i^u \right). \tag{35}$$

Assuming that it is possible to design an M-subchannel length-$2mM$ lowpass PR prototype filter $h(n)$ with frequency response $H(e^{j\omega})$ that satisfies $20\log_{10}|H(e^{j\omega})/H(e^{j0})| \le -A_s$ dB for all $\omega \in [\pi/M, 2\pi - \pi/M]$, where A_s is the stopband attenuation of $H(e^{j\omega})$ in dBs, we will define $\omega \in [0, \pi/M] \bigcup [2\pi - \pi/M, 2\pi]$ as the transition band of $H(e^{j\omega})$ and the remainder of the digital band as the stopband. Consequently, the kth CEM analysis filter, $h_k(n)$, and synthesis filter, $f_k(n)$, obtained from (28) have $\omega \in [\pi(-1+k)/M, \pi(1+k)/M]$ as transition band and the remainder as stopband. The designation of the transition bands of these filters as such indicates that they individually do not have passbands. However, the combination of a set of $h_k(n)$ or $f_k(n)$ filters with consecutive values of k results in filters with nonzero passbands. We will restrict the stopband edges ω_s of the PR prototype filters used in this chapter to $\omega_s = \pi/M$. This provides the best performance for the channelizer, which corresponds to the highest attenuation of the undesired aliasing in the output signals.

The CEM PR filter banks required for the channelizer discussed in this section have relatively large power-of-two number of subchannels $2M$. The power-of-two value for $2M$ for channelizing a particular $X(z)$ is chosen such that none of the transition bands of the analysis filters of the $2M$-subchannel CEM filter bank overlaps the band of more than

one channel in $X(z)$, i.e., none of the guard bands in $X(z)$ is fully contained within the transition band of a single analysis filter. Therefore, a specific value of M can be used in the channelizer if there exist an arbitrary set of integers $\{k_i, i = 0, 1, \cdots, N-1\}$ satisfying

$$\omega_i^u \leq \frac{\pi k_i}{M} \leq \omega_{i+1}^l - \frac{\pi}{M}. \tag{36}$$

For $X(z)$ with a specific value of $G_{\min}$, the worst case value for an integer R, where $M = 2^R$, is $\pi/2^R = G_{\min}/2$, which results in a value of M that is given by

$$M = 2^{\left\lceil \log_2\left(\frac{2\pi}{G_{\min}}\right)\right\rceil}. \tag{37}$$

For values of M that satisfy (37), the analysis filters of a $2M$-subchannel CEM filter bank are able to divide the spectrum of $X(z)$ into $2M$ narrowband signals that can be grouped into N mutually exclusive sets of M_i subchannels, for $i = 0, 1, \cdots, N-1$, such that the band of $X_i(z)$ is fully contained within the transition bands of the inner $M_i - 2$ filters of the ith set. Recombining the ith set of narrowband signals allows the reconstruction of the ith channel of $X(z)$.

Let Q_i^l and Q_i^u specify the lower and upper values of k, respectively, of a contiguous set of CEM analysis filters $H_k(z)$ that is associated with the ith channel $X_i(z)$, such that

$$Q_i^l = \left\lfloor \frac{M\omega_i^l}{\pi} \right\rfloor, \quad \text{and} \quad Q_i^u = \left\lceil \frac{M\omega_i^u}{\pi} \right\rceil, \tag{38}$$

and let $M_i = Q_i^u - Q_i^l + 1$ be the number of filters in this set. Assuming M satisfies (36) or (37) for a specific $X(z)$, the outputs $V_k(z)$ of the analysis section of a $2M$-subchannel CEM PR filter bank, for $k = 0, 1, \cdots, 2M-1$, are given by

$$V_k(z) = \frac{1}{M} \sum_{p=0}^{M-1} H_k\left(z^{\frac{1}{M}} W_M^p\right) X\left(z^{\frac{1}{M}} W_M^p\right). \tag{39}$$

The ith channel can now be extracted from $X(z)$ by reconstructing it from the M_i signals $V_k(z)$, for $k = Q_i^l, Q_i^l + 1, \cdots, Q_i^u$. For the modified CEM filter bank in Fig. 19, where $V_k(z)$ for $k = Q_i^l, Q_i^l + 1, \cdots, Q_i^u$ are fed to the M_i inputs of the synthesis section starting at the (Q_i^l mod 2)th synthesis branch (i.e., the zeroth branch for even values of Q_i^l and first branch for odd values of Q_i^l) and the inputs to all other synthesis branches are set to zero,

the signal labelled $Y_i(z)$ becomes

$$Y_i(z) = \frac{1}{M} \sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=\eta_i}^{Q_i^u - Q_i^l + \eta_i} H_{k+Q_i^l-\eta_i}\left(zW_M^p\right) H_k(z), \tag{40}$$

where $\eta_i = (Q_i^l \bmod 2)$. Using (32), $H_{k+Q_i^l-\eta_i}(zW_M^p)$ can be written as

$$H_{k+Q_i^l-\eta_i}(zW_M^p) = W_{2M}^{(Q_i^l-\eta_i)\frac{2mM-1}{2}} H_k\left(zW_M^{p+\frac{Q_i^l-\eta_i}{2}}\right). \tag{41}$$

Since $(Q_i^l - \eta_i)$ is even, the term indexed $p = M - (Q_i^l - \eta_i)/2$ in the outer summation of (40) can be separated and using (41), $Y_i(z)$ becomes

$$\begin{aligned} Y_i(z) = {} & \frac{1}{M} W_{2M}^{(Q_i^l-\eta_i)\frac{2mM-1}{2}} X\left(zW_M^{-\frac{Q_i^l-\eta_i}{2}}\right) \sum_{k=\eta_i}^{Q_i^u - Q_i^l + \eta_i} H_k^2(z) \\ & + \frac{1}{M} W_{2M}^{(Q_i^l-\eta_i)\frac{2mM-1}{2}} \sum_{\substack{p=0 \\ p \neq M-(Q_i^l-\eta_i)/2}}^{M-1} X\left(zW_M^p\right) \sum_{k=\eta_i}^{Q_i^u - Q_i^l + \eta_i} H_k\left(zW_M^{p+\frac{Q_i^l-\eta_i}{2}}\right) H_k(z), \end{aligned} \tag{42}$$

where the first term is the desired signal corresponding to the ith extracted channel and the second term is the aliasing signal due to downsampling by M in the analysis section. The summation in the first term of (42) for PR filters $H_k(z)$ is equivalent to a filter that passes the ith channel $X_i(z)$ of the input signal and attenuates the remaining $N-1$ channels. Unlike the aliasing term in (30), the aliasing term in $Y_i(z)$ is significantly larger because $V_k(z)$, for $k = 0, 1, \cdots, Q_i^l - 1, Q_i^u + 1, Q_i^u + 2, \cdots, 2M-1$, are not included in the second term of (42) to provide the nearly complete alias cancellation. Therefore, even when filter banks that satisfy the PR criterion exactly are used in the channelizer discussed here, the output channels are not exactly equal to the channels of the input signal.

Since the bandwidth of $Y_i(z)$, for $i = 0, 1, \cdots, N-1$, is narrower than the bandwidth of $X(z)$ for values of $N > 1$, the sample rate of $Y_i(z)$ can be lowered by a maximum factor that depends on the bandwidth of $X_i(z)$. Define $\hat{M}_i$ to be a power-of-two integer that is related to M_i by

$$\hat{M}_i = 2^{\lceil \log_2 M_i \rceil - 1}. \tag{43}$$

The sample rate of $Y_i(z)$ can be safely reduced by a factor of $M/\hat{M}_i$ without causing significant aliasing to the ith output channel since the other $N-1$ input channels have

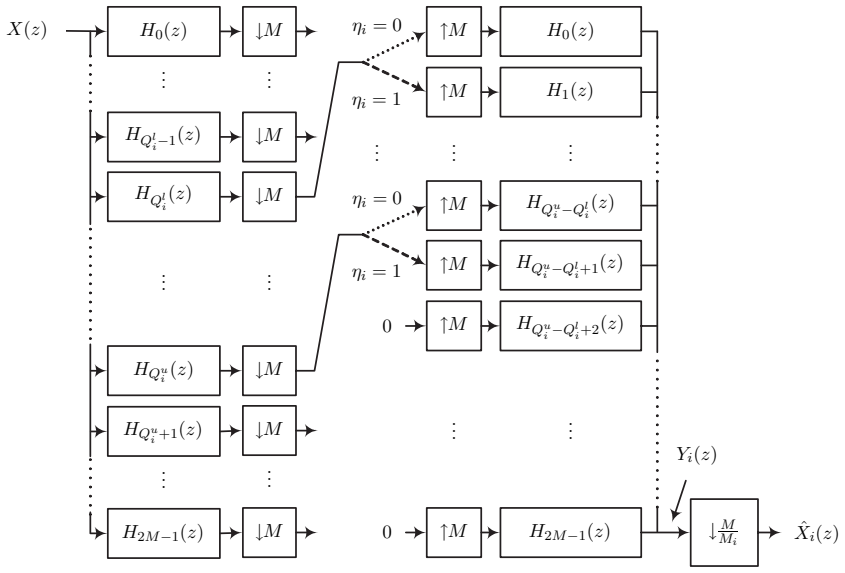


Figure 19: The building block of the new channelizer used for extracting the ith channel in the input signal.

been attenuated. Downsampling $Y_i(z)$ by a factor $M/\hat{M}_i$ produces the ith channelizer output signal $\hat{X}_i(z)$ given by

$$\hat{X}_i(z) = \frac{\hat{M}_i}{M^2} W_{2M}^{(Q_i^l-\eta_i)\frac{2mM-1}{2}} \sum_{q=0}^{\frac{M}{\hat{M}_i}-1} \sum_{p=0}^{M-1} \left[X\left(z^{\frac{\hat{M}_i}{M}} W_M^{p+q\hat{M}_i}\right) \right.$$
$$\left. \sum_{k=\eta_i}^{Q_i^u-Q_i^l+\eta_i} H_k\left(z^{\frac{\hat{M}_i}{M}} W_M^{p+q\hat{M}_i+\frac{Q_i^l-\eta_i}{2}}\right) H_k\left(z^{\frac{\hat{M}_i}{M}} W_M^{q\hat{M}_i}\right) \right]. \quad (44)$$

The signal $\hat{X}_i(z)$ in (44) is constructed by aliasing shifted, downsampled, and filtered forms of $X(z)$. The filtering provided by the inner-most summation of (44) guarantees that all channels except $X_i(z)$ are attenuated before aliasing takes place. By repeating the single channel extractor of Fig. 19 for all the desired channels using a single analysis section for all channels but different synthesis sections for the different channels that are extracted, the complete construction of the new channelizer shown in Fig. 20 is obtained.

The channelizer in Fig. 20 is in discrete filter form and therefore, is not computationally optimized, where the number of required computations per extracted channel far exceeds that of the traditional DFB channelizer. Since the analysis and synthesis filters in Fig. 20 are obtained by modulating $H(z)$, they can be polyphase decomposed in a form similar to

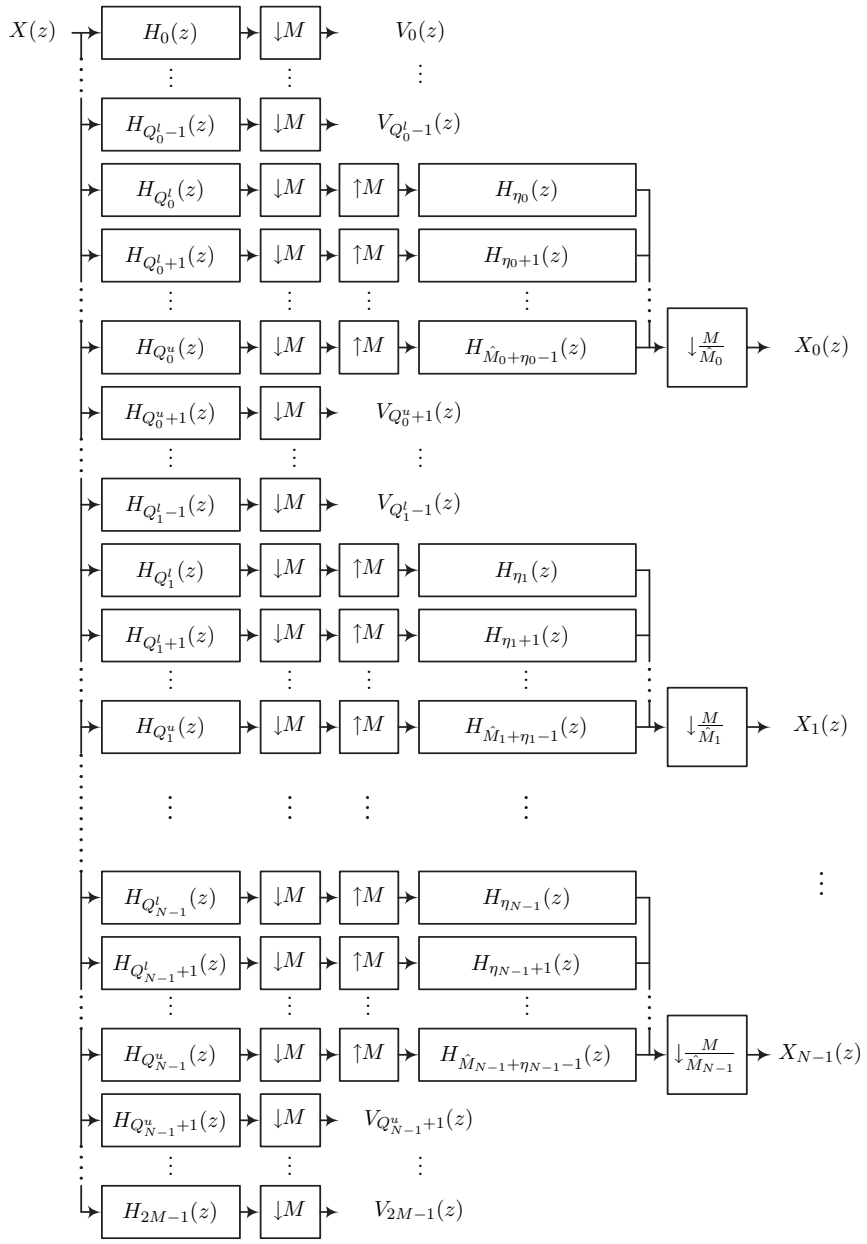


Figure 20: The new channelizer in discrete filter form.

the filter bank of Fig. 18(b) using the polyphase components $E_q(z)$, for $q = 0, 1, \cdots, 2M-1$, of $H(z)$. The polyphase decomposition of the synthesis sections of Fig. 20 results in the cascade of upsampling by M, delay elements, and downsampling by $(M/\hat{M}_i)$ for the ith channel. Since M and $\hat{M}_i$ are powers-of-two, with $M \geq \hat{M}_i$, the upsampling by M can be decomposed into cascades of upsampling by $\hat{M}_i$ followed by upsampling by $(M/\hat{M}_i)$ as shown in Fig. 21(a). The polyphase identity [107] applied to delay elements as shown in Fig. 21(b) simplifies the cascade, where only $\hat{M}_i$ polyphase components are retained. Since all but the (η_i)th to the $(Q_i^u - Q_i^l + \eta_i)$th inputs to the $2M$-point FFT block are zero, $Q_i^u - Q_i^l + 1 \leq 2\hat{M}_i$, and only the $(M/\hat{M}_i)$th, $(2M/\hat{M}_i)$th, $\cdots$, $(2M)$th outputs are required, a $2M$-point decimated-in-time FFT block [73] can be replaced with a $2\hat{M}_i$-point FFT block. If $(Q_i^u - Q_i^l + \eta_i) = 2\hat{M}_i$ for a particular channel, the value of $\hat{M}_i$ can be doubled or the last analysis signal, $V_{Q_i^u}(z)$, for this channel can be fed to the zeroth input of the FFT. The construction of the simplified synthesis section for the ith channel is shown in Fig. 21(c), where the polyphase components $E_q^{(\hat{M}_i)}(z)$, for $q = 0, 1, \cdots, 2\hat{M}_i - 1$, are given by

$$E_q^{(\hat{M}_i)}(z) = E_{q\frac{M}{\hat{M}_i}}(z) = \sum_{r=0}^{m-1} h\left(q\frac{M}{\hat{M}_i} + 2Mr\right) z^{-r}. \tag{45}$$

This results in the efficient reduced-complexity polyphase representation of the new channelizer shown in Fig. 22. For an input signal with equal-power channels, the output channels of Fig. 22 have a signal-to-noise ratio (SNR) that is on the order of the stopband attenuation A_s of $H(z)$, where the SNR is defined as the ratio of the maximum PSD of the desired channel to the maximum PSD of aliasing. The sample rate of the ith output signal is proportional to $\hat{M}_i$, and some channels may require SRC to bring their sample rates to the proper values required for further baseband processing. As shown later, doubling the value of $\hat{M}_i$ generally results in a marginal increase in the number of computations for extracting the ith channel. Therefore, the value of $\hat{M}_i$ for a channel that requires SRC can be doubled if this reduces the computations for SRC significantly.

5.3.1 Aliasing and Amplitude Distortion

Aliasing in the ith output signal, $\hat{X}_i(z)$, results from downsampling by M and $(M/\hat{M}_i)$ in the analysis and synthesis sections, respectively. To estimate the aliasing error in $\hat{X}_i(z)$,

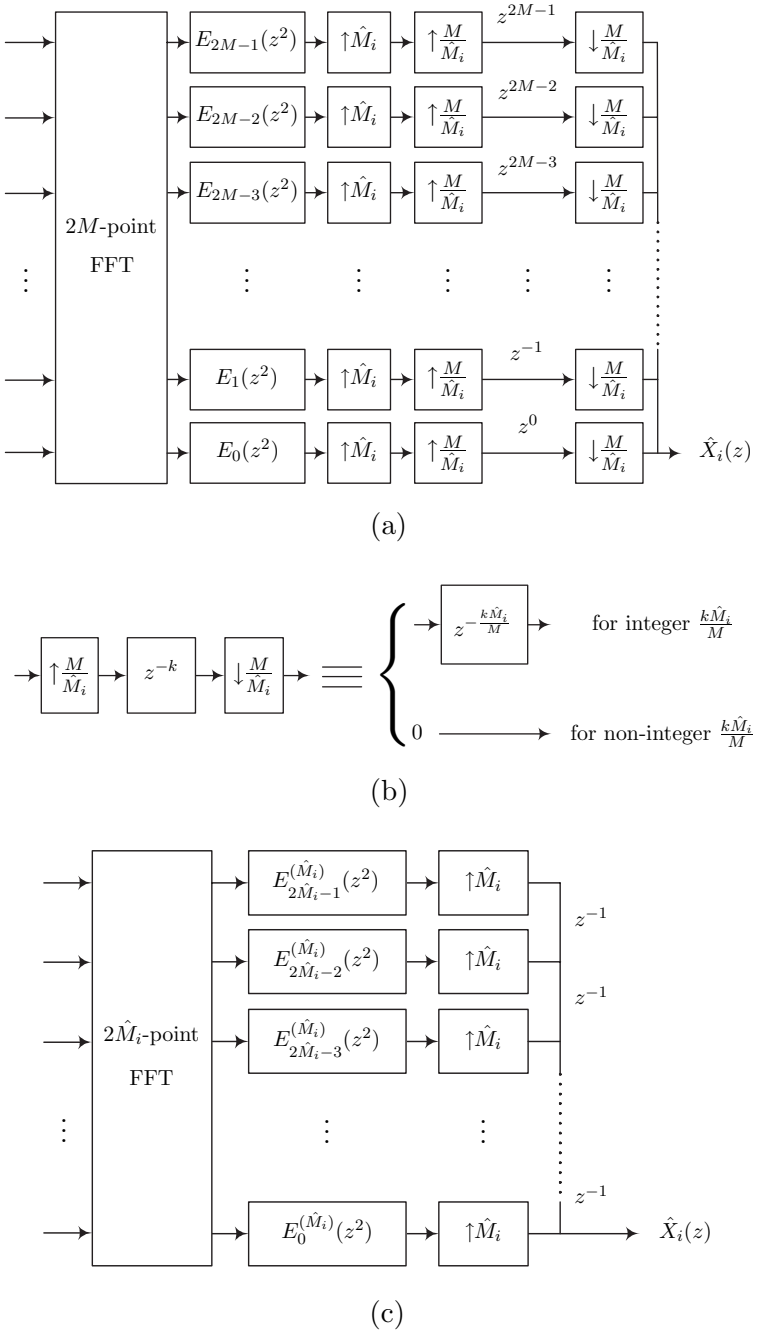


Figure 21: Simplification of the synthesis sections of the new channelizer. (a) The original polyphase decomposed synthesis section of the ith channel, (b) the polyphase identity applied to pure delays, and (c) the simplified synthesis section of the ith channel.

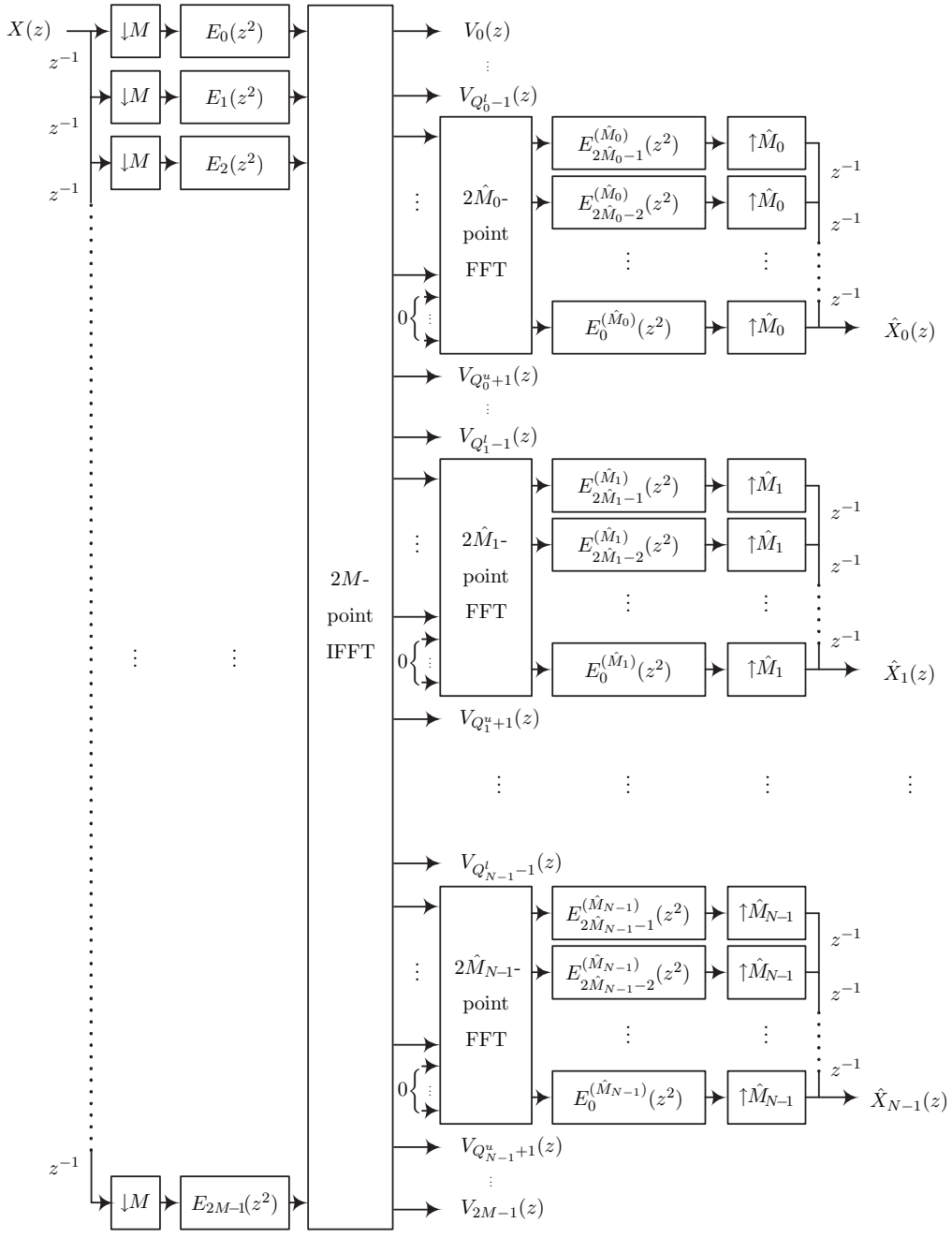


Figure 22: The new channelizer in PFB form.

consider the signal $Y_i(z)$ in (42), which experiences two errors. The first error caused by downsampling by M is

$$\xi_{Y1}(e^{j\omega}) = \frac{1}{M}\left[\sum_{\substack{p=0\\ p\neq M-(Q_i^l-\eta_i)/2}}^{M-1}\left|\sum_{k=\eta_i}^{Q_i^u-Q_i^l+\eta_i} H_k(e^{j\omega})H_k\left(e^{j\omega}W_M^{p+\frac{Q_i^l-\eta_i}{2}}\right)\right|^2\right]^{\frac{1}{2}}. \tag{46}$$

This error results in the first aliasing component in $\hat{X}_i(z)$. The finite attenuation of the undesired channels in $Y_i(z)$ gives rise to the error

$$\xi_{Y2}(e^{j\omega}) = \frac{1}{M}\left|\sum_{k=\eta_i}^{Q_i^u-Q_i^l+\eta_i} H_k^2(e^{j\omega})\right| \tag{47}$$

outside the interval $\pi(\eta_i-1)/M \leq \omega \leq \pi(Q_i^u-Q_i^l+\eta_i+1)/M$ that results in the second aliasing component in $\hat{X}_i(z)$. Assuming that the PR prototype filter $H(z)$ of the CEM filter bank that is used in the channelizer has a stopband attenuation of A_s dB and a stopband edge of $\omega_s = \pi/M$, it is easy to show that the magnitudes of $\xi_{Y1}(e^{j\omega})$ and $\xi_{Y2}(e^{j\omega})$ are roughly A_s and $2A_s$ dB lower than the peak magnitude of $H^2(z)$, respectively, which means that $\xi_{Y2}(e^{j\omega})$ can be ignored. Therefore, the aliasing in $\hat{X}_i(z)$ becomes

$$\xi_X(e^{j\omega}) = \frac{\hat{M}_i}{M^2}\left[\sum_{q=0}^{\frac{M}{\hat{M}_i}-1}\sum_{\substack{p=0\\ p\neq M-(Q_i^l-\eta_i)/2}}^{M-1}\left|\sum_{k=\eta_i}^{Q_i^u-Q_i^l+\eta_i} H_k\left(z^{\frac{\hat{M}_i}{M}}W_M^{q\hat{M}_i}\right)H_k\left(z^{\frac{\hat{M}_i}{M}}W_M^{p+q\hat{M}_i+\frac{Q_i^l-\eta_i}{2}}\right)\right|^2\right]^{\frac{1}{2}}. \tag{48}$$

The value of ω_s of the PR prototype filter used in the new channelizer is restricted to π/M because a larger ω_s significantly increases the peak values of $\xi_{Y1}(e^{j\omega})$ and $\xi_X(e^{j\omega})$ and a smaller ω_s reduces the A_s that can be achieved during the design of the prototype filter. The peak magnitude of $\xi_{Y1}(e^{j\omega})$ was found to be generally within 1 dB from $-A_s$ dB, and the peak magnitude of $\xi_X(e^{j\omega})$ was 2–3 dB higher. For example, the peak magnitudes of $\xi_{Y1}(e^{j\omega})$ and $\xi_X(e^{j\omega})$ using $H(z)$ with $M = 512$ and $m = 13$ that has $A_s = 98.04$ dB were -98.09 dB and -95.39 dB, respectively. Since the use of finite-precision computations causes a noise floor in the channelized signals, the required precision of the computations and filter coefficients for a particular application must be high enough to preserve the desired SFDR of the system. Therefore, for a channelizer with a desired 95 dB SFDR, for example, computations and filter coefficients with 18–20 bits of accuracy would be required.

The peak-to-peak amplitude distortion in $\hat{X}_i(z)$ [51] caused by passband ripples of the filter $T_i(e^{j\omega})$ that extracts $\hat{X}_i(z)$ is evaluated by considering

$$T_i(e^{j\omega}) = \frac{1}{M} \sum_{k=\eta_i}^{Q_i^u - Q_i^l + \eta_i} H_k^2(e^{j\omega}). \tag{49}$$

The peak passband ripple $(T_{\max} - T_{\min})/(T_{\max} + T_{\min})$, where $T_{\min}$ and $T_{\max}$ are the minimum and maximum values of $|T_i(e^{j\omega})|$ in the passband $[\pi\eta_i/M, \pi(Q_i^u - Q_i^l + \eta_i)/M]$, was found to be on the order of $-2A_s$ dB for PR filter banks with different values of M, m, and M_i. For the PR prototype filter with $M = 512$ and $m = 13$, the peak passband ripple was found to be -197.6 dB, which results in a peak-to-peak amplitude distortion of -191.6 dB.

5.3.2 Use of the Channelizer for Channel Synthesis

The use of PFBs for channel synthesis requires similar conditions to the those that enable the use of PFBs for channelization be satisfied. The input signals to a PFB synthesizer must have the same sample rate and channel bandwidth, and the ratio L of the sample rate of the synthesized signal to the sample rate of the input signals must be an integer multiple of the number of channels. The channelizer can be easily modified for synthesizing a set of channels that do not satisfy these conditions to generate a wideband signal with FDM channels. For synthesizing a signal with sample rate R_s that has a minimum guard band of $G_{\min}$ between channels, the sample rates of the different channels are independently adjusted to the lowest power-of-two multiples of R_s/M, where M is obtained by using (37). The channelizer in Fig. 22 is then used in the reverse form to perform channel synthesis. This is done by inverting the different blocks of the channelizer, i.e., downsampling by M, $2M$-point IFFT, $2\hat{M}_i$-point FFTs, and upsampling by $2\hat{M}_i$ in reverse become upsampling by M, $2M$-point FFT, $2\hat{M}_i$-point IFFTs, and downsampling by $2\hat{M}_i$, respectively. The input and outputs of the channelizer are also reversed so that they become the output and inputs of the synthesizer, respectively.

5.4 Computational Complexity of the New Channelizer

The computations for the new channelizer described in this chapter and the DFB channelizer are compared for complex-valued input signals with sample rates R_s. The input signals are assumed to have N channels from which N_e channels with indexes $\Omega \subset \{0, 1, \cdots, N-1\}$ are to be extracted with SNRs of 95 dB. Due to the downsampling by M, the filters and transforms in the new channelizer of Fig. 22 operate at a sample rate of R_s/M. For power-of-two values of M and $\hat{M}_i$, a $2M$-point IFFT and $2\hat{M}_i$-point FFT, respectively, require $[4M\log_2(2M) - 4M]$ and $[4\hat{M}_i\log_2(2\hat{M}_i) - 4\hat{M}_i]$ real multiplications per application [73]. Therefore, the analysis IFFT and synthesis FFTs for extracted channels perform $[4M\log_2(2M) - 4M]R_s/M$ and $[\sum_{i\in\Omega}(4\hat{M}_i\log_2(2\hat{M}_i) - 4\hat{M}_i)]R_s/M$ real multiplications per second (RMPS), respectively, and the analysis and synthesis filtering perform $(4mM)R_s/M$ and $(\sum_{i\in\Omega} 4m\hat{M}_i)R_s/M$ RMPS, respectively. The total RMPS of the new channelizer becomes

$$\Gamma_{\text{New}} = 4R_s\left(m + \log_2 M + \frac{1}{M}\sum_{i\in\Omega}\left(\hat{M}_i\log_2\hat{M}_i\right) + \frac{m}{M}\sum_{i\in\Omega}\hat{M}_i\right) \quad \text{RMPS.} \tag{50}$$

Assuming that length-L equiripple baseband FIR filters with variable bandwidths are used in a DFB channelizer such that the ith channel is frequency shifted to baseband and extracted at a reduced sample rate close to $(\omega_i^u - \omega_i^l)R_s/2\pi$, the number of RMPS for extracting the N_e channels is approximately

$$\Gamma_{\text{DFB}} = 2R_sL\sum_{i\in\Omega}\frac{\omega_i^u - \omega_i^l}{2\pi} \quad \text{RMPS.} \tag{51}$$

When applicable, the number of RMPS of the PFB is at least

$$\Gamma_{\text{PFB}} = 2R_s\left(\frac{L}{N} + \log_2 N\right) \quad \text{RMPS.} \tag{52}$$

Assuming equal passband and stopband ripples of the filters in the DFB channelizer, the required value of L for a specific $G_{\min}$ and channelizer SNR [46] is given by

$$L = \frac{\text{SNR} - 13}{2.324 \cdot G_{\min}}. \tag{53}$$

For a channelizer with a SNR of 95 dB, for example, $L = 35.284/G_{\min}$. Substituting for $G_{\min}$ in (37) gives a bound on the required value of M in terms of L as

$$0.1781L \le M < 0.3561L. \tag{54}$$

Substituting the upper limit of M given by (54) and the value of $m = 13$ that is required in the new channelizer for achieving a SNR of 95 dB into (50) gives the worst case RMPS of the new channelizer in terms of L. Since the value of $\sum_{i\in\Omega} \hat{M}_i$ is upper bounded by $2M$, and, consequently, $\sum_{i\in\Omega} \left(\hat{M}_i \log_2 \hat{M}_i\right)$ is upper bounded by $2M \log_2 2M$, the maximum RMPS for the new channelizer is $R_s(12m + 8 + 12\log_2 M)$, or at most $R_s(146.12 + 12\log_2 L)$ RMPS. Fig. 23 shows the approximate number of RMPS of the DFB channelizer, the PFB channelizer (assuming applicability), and the worst case of the new channelizer versus the number of extracted channels, N_e, for a SNR of 95 dB. It is seen that Γ_{New} is lower than Γ_{PFB} for $N_e < 12$, and is far lower than Γ_{DFB} for all values of N_e. Furthermore, (50) indicates that doubling the value of M or doubling $\hat{M}_i$ for a specific channel increases the number of RMPS of the new channelizer by a relatively small amount. Therefore, if SRC is required for a particular channel that has $M_i \approx 2\hat{M}_i$, the value of $\hat{M}_i$ can be doubled to reduce the length of the filters and the computational requirements of the SRC at the expense of a small increase in the computations for the channelizer.

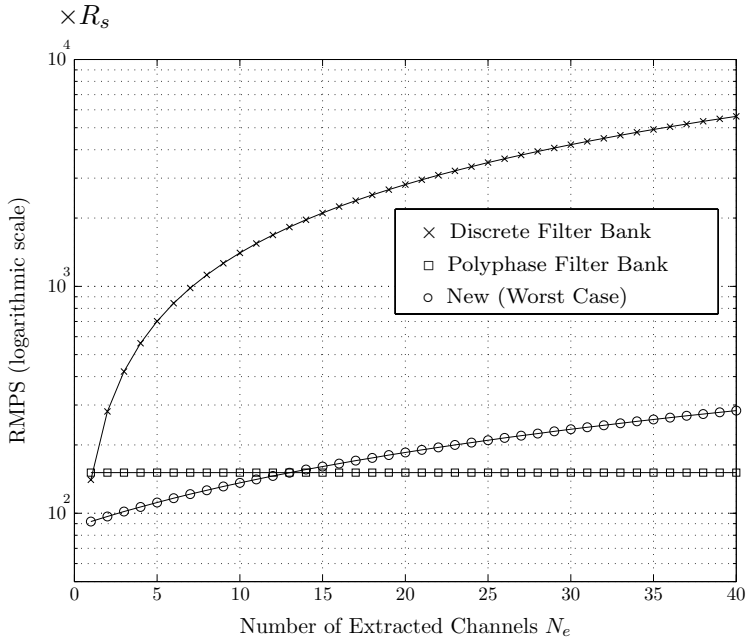
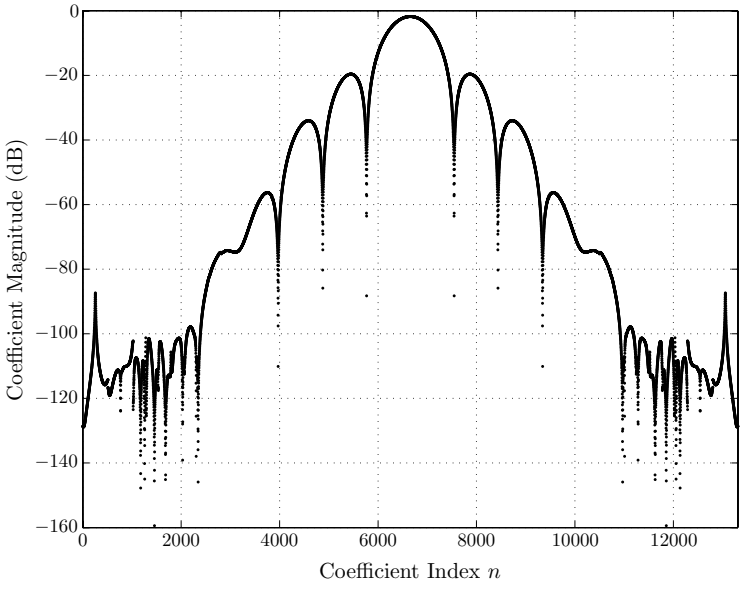


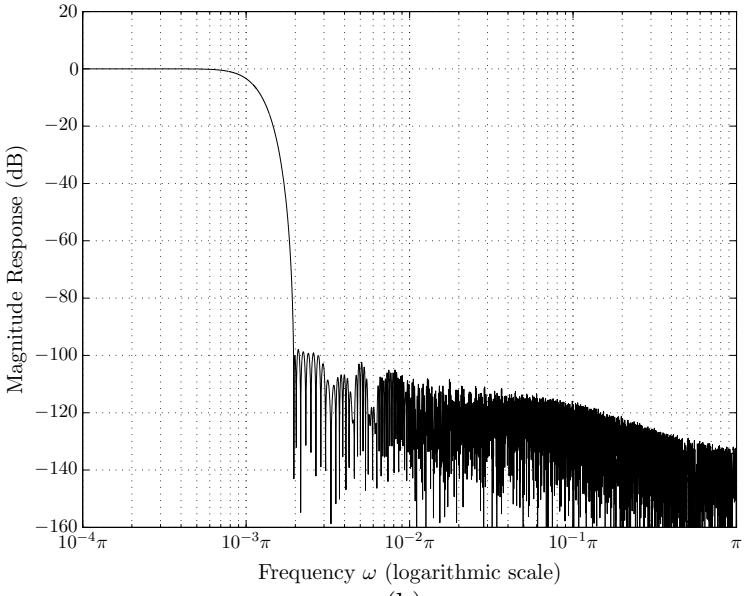
Figure 23: Approximate RMPS versus N_e for the DFB, the PFB (when applicable), and the worst case of the new channelizer.

5.5 Simulation Results

The new channelization method is applied to the complex-valued input signal of Fig. 25(a) with $N = 40$ channels of variable bandwidths ranging from 0.02π to 0.06π and guard bands ranging in width from 0.004π to 0.006π. A 512-subchannel length-13312 PR prototype filter shown in Fig. 24 that was designed using the procedure described in Chapter 6 was used in the channelizer. Fig. 25(b) shows the 16th input channel at a low sample rate. The values of Q_{16}^{l} and Q_{16}^{u} for this channel in the input signal are 368 and 396, respectively. The new channelizer frequency shifts the input signal in Fig. 25(a) to bring the band of a desired channel close to zero frequency. For example, the magnitude response in Fig. 25(c) is for the filter $T_{16}(e^{j\omega})$ that effectively extracts the 16th input channel. Fig. 25(d) represent the maximum aliasing $\xi_{Y1}(e^{j\omega})$ that the 16th extracted channel shown in Fig. 25(e) may experience. This aliasing raps around when the sample rate of the signal in Fig. 25(e) is reduced by $M/\hat{M}_i$ to form the aliasing $\xi_X(e^{j\omega})$ that the output 16th channel shown in Fig. 25(f) experiences. The signals in Fig. 25(e) and (f) are the extracted 16th channel at the sample rate of the input signal and at the reduced output sample rate, respectively. The 16th output channel experiences a SNR of at least 96.45 dB. The new channelizer requires a total of approximately $143R_s$ RMPS for full channelization of the signal in Fig. 25(a), while the DFB channelizer requires approximately $2LR_s = 5616R_s$ RMPS.



(a)



(b)

Figure 24: PR prototype filter with $M = 512$, $m = 13$, $\omega_s = \pi/512$, and $A_s = 98.04$ dB designed with the method presented in Chapter 6. (a) Coefficient magnitude and (b) magnitude response (using log scale for ω to magnify details of the transition band).

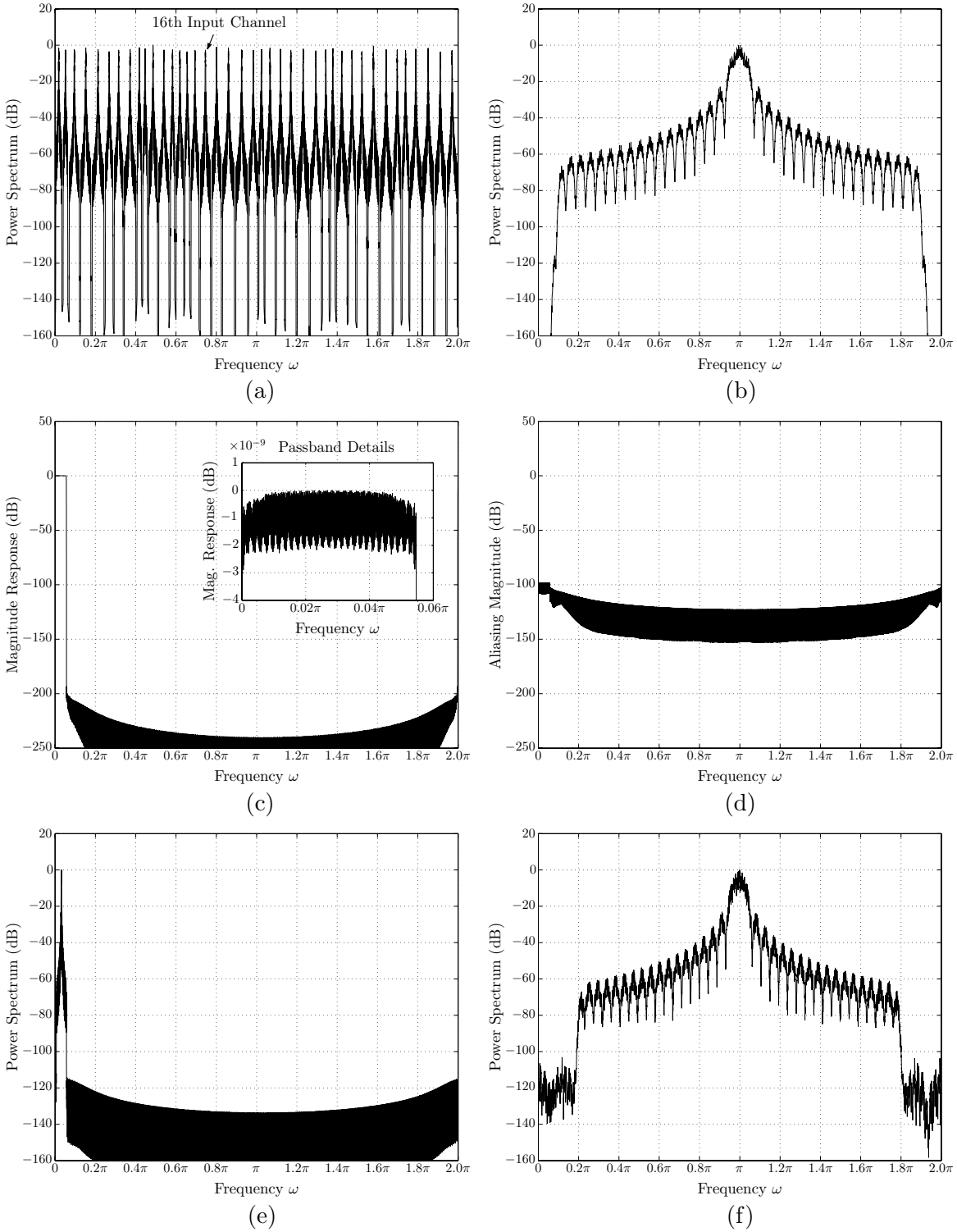


Figure 25: (a) Input to the new channelizer, (b) PSD of the 16th channel, (c) response of $T_{16}(e^{j\omega})$ with passband details, (d) aliasing $\xi_{Y1}(e^{j\omega})$ in the 16th channel with peak value of -98.09 dB, (e) PSD of the extracted 16th channel at sample rate of input signal, and (f) PSD of the extracted 16th channel at the output sample rate with SNR of 95.39 dB.

CHAPTER VI

DESIGN OF MODULATED PR PROTOTYPE FILTERS WITH LARGE VALUES OF SUBCHANNELS M

The design of perfect reconstruction (PR) and near-PR finite impulse response (FIR) filter banks has been extensively studied in the literature [16, 17, 31, 48, 69–72, 87, 104–106, 109, 115, 121]. Resorting to the use of near-PR filter banks instead of PR filter banks is usually attributed to the relative ease of designing near-PR filter banks. Designing PR filter banks with large number of subchannels that contain filters with high stopband attenuations is difficult due to the complexity of parameter optimization caused by the high non-linearity between the design parameters, filter coefficients, and the frequency response of the filters. This high non-linearity usually results in limiting the amount of stopband attenuation that can be achieved for a specific number of subchannels. Because of this, the methods for designing modulated PR filter banks that are described in the literature are generally impractical for designing PR filter banks with large number of subchannels and high stopband attenuations.

In this chapter, we consider the design of a class of PR filters known as modulated PR filter banks. Modulated PR filter banks are generally favored over non-modulated PR filter banks because they only require the design of a lowpass prototype filter, from which filters of the bank are obtained by modulation, and they can be implemented efficiently in a polyphase filter bank form. It was shown in [57,78,108] that modulated PR filter banks exist for linear-phase FIR prototype filters of length $2M$, where M is the number of subchannels in the cosine-modulated filter bank. This result was extended in [57] for prototype filters of length $2mM$, where m is an arbitrary positive integer, and an efficient method for designing the prototype filters of modulated PR filter banks was proposed in [51]. Compared to quadrature mirror filter (QMF) bank design methods, the design method of [51] requires optimizing at most half the number of parameters that are optimized in pseudo-QMF bank

design methods and a smaller fraction of the number of parameters that are optimized in PR-QMF bank design methods for similar filter lengths. The design method described in this chapter, which is based on the efficient design method of [51], admits the efficient design of PR filter banks that have a large number of subchannels and filters with high stopband attenuations.

The remainder of this chapter is outlined as follows. Section 6.1 describes the efficient modulated PR filter bank design method and Section 6.2 gives a design example of the presented design method.

6.1 *Efficient Method for Designing Modulated PR Prototype Filters with Large Even Values of* M

It was assumed in Section 5.3 that an M-subchannel PR prototype filter with high stopband attenuation A_s and large power-of-two M exists. Channelizing a 40-channel SWR signal with $G_{\min} = 2\pi/400$ and 95 dB dynamic-range, for example, requires a PR prototype filter with $M = 512$ and $m = 13$ that has A_s slightly higher than 95 dB for building a 1024-subchannel CEM PR filter bank. Designing such PR prototype filter is non-trivial and has not been considered in the literature. It was stated in [72] that prototype filters of modulated PR filter banks for values of $M > 2$ with A_s close to 100 dB have not been shown to exist due to the high non-linearity that relates the design parameters to the filter coefficients. To the best of our knowledge, the literature has only reported designs of modulated PR filter banks with relatively small values of $M \leq 32$ and prototypes with relatively low values of $A_s \leq 75$ dB that require optimizing on the order of 100 parameters. The complexity experienced when designing modulated PR filter banks with extreme values of M and A_s for the channelizer discussed in Chapter 5 may be several orders of magnitude higher than the complexity of the design examples in the literature. Furthermore, the existence of a large number of local minima in the optimization functions when designing PR filter banks with $M > 2$ and $m > 10$ limits the values of A_s that can be achieved.

The design method proposed in [51] is efficient in the sense that it requires optimizing at most half the number of parameters that are optimized in the pseudo- or PR-QMF

bank designs as shown in Table 2. However, its complexity increases sharply with the number of design parameters, and attempts for designing prototype filters with high values of $A_s > 75$ dB for $M > 2$ were unsuccessful. The design method in [7] was used for designing prototype filters of modulated PR filter banks with power-of-two values of $M = 32$. However, the complexity of this method increases sharply as M increases, because it uses the same objective functions that are used in [51] and it requires optimizing the parameters of prototype filters of all the modulated PR filter banks with $2, 4, 8, \cdots, M/2$-subchannels first. We may safely conclude that designing PR filter banks with high values of $M > 500$ and A_s close to 100 dB, for example, is generally very difficult, if not impossible by using the currently existing design methods in the literature.

Table 2: Number of parameters simultaneously optimized in different PR filter bank design methods for prototype filters of length $2mM$ [51, 104]

Design Method	Number of Parameters	Number of Optimizations
Pseudo-QMF Bank	mM	1
PR-QMF Bank	$(2m-1)(M-1)+\binom{M}{2}$	1
Mod. PR Filter Bank	$m\lfloor M/2 \rfloor$	1
Our Design Method	m	$1+\lfloor M/2 \rfloor$

A simple method for designing modulated PR filter banks with virtually any even value of M is presented in this section. The method described here, which is based on the method described in [51], will work well for large values of M and high A_s, and its complexity increases linearly with the value of M. Considering only even values of M, which is the case for the channelizer discussed in Chapter 5 and most practical applications, the design method of [51] requires the simultaneous optimization of $mM/2$ parameters denoted $\theta_{q,p}$, for $q = 0, 1, \cdots, M/2 - 1$, and $p = 0, 1, \cdots, m - 1$. For a specific q, the m parameters $\theta_{q,p}$, for $p = 0, 1, \cdots, m - 1$, generate the pair of polyphase components $E_q(z)$ and $E_{q+M}(z)$ of $H(z)$ as

$$\begin{bmatrix} E_q(z) \\ E_{M+q}(z) \end{bmatrix} = \begin{bmatrix} \cos\theta_{q,m-1} & z^{-1}\sin\theta_{q,m-1} \\ \sin\theta_{q,m-1} & -z^{-1}\cos\theta_{q,m-1} \end{bmatrix} \begin{bmatrix} \cos\theta_{q,m-2} & z^{-1}\sin\theta_{q,m-2} \\ \sin\theta_{q,m-2} & -z^{-1}\cos\theta_{q,m-2} \end{bmatrix} \cdots$$

$$\begin{bmatrix} \cos\theta_{q,1} & z^{-1}\sin\theta_{q,1} \\ \sin\theta_{q,1} & -z^{-1}\cos\theta_{q,1} \end{bmatrix} \begin{bmatrix} \cos\theta_{q,0} \\ \sin\theta_{q,0} \end{bmatrix}. \tag{55}$$

The remaining M polyphase components are obtained from the symmetry of $H(z)$ as

$$\begin{cases} E_{M-q-1}(z) = z^{-(m-1)}E_{M+q}(z^{-1}), \\ E_{2M-q-1}(z) = z^{-(m-1)}E_q(z^{-1}), \end{cases} \quad q = 0, 1, \cdots, M/2-1. \tag{56}$$

The process of optimizing the $mM/2$ parameters $\theta_{q,p}$ starts with the initialization

$$\theta_{q,p} = \begin{cases} \frac{\pi}{4}, & p = 0, & q = 0, 1, \cdots, \frac{M}{2}-1, \\ \frac{\pi}{2}, & p = 1, 2, \cdots, m-1, & q = 0, 1, \cdots, \frac{M}{2}-1. \end{cases} \tag{57}$$

Substituting all $\theta_{q,p}$ into (55) and (56) gives the $2M$ polyphase components of $H(z)$, from which the frequency response $H(e^{j\omega})$ is evaluated and used in either of the objective functions

$$\Phi_1 = \int_{\frac{\pi}{2M}+\alpha}^{\pi} |H(e^{j\omega})|^2 d\omega, \qquad \Phi_2 = \max_{\omega\in\left[\frac{\pi}{2M}+\alpha,\pi\right]} |H(e^{j\omega})|, \tag{58}$$

where $0 \le \alpha \le \pi/2M$. The design method of [51] optimizes the $mM/2$ parameters by minimizing Φ_1 first and then minimizing Φ_2. Clearly, the simultaneous optimization of $mM/2 \gg 100$ parameters as required by the channelizer of Chapter 5 may be difficult because the optimization complexity increases sharply as mM increases, and because of the demanding numerical computation of $H(e^{j\omega})$ with sufficiently high frequency resolution.

The optimization in [51] for designing large-M modulated PR filter banks can be significantly simplified by converting it to $(M/2+1)$ optimizations each with m parameters. Since only the m parameters of $\theta_{q,p}$ for a specific q and $p = 0, 1, \cdots, m-1$ contribute to $E_q(z)$, $E_{M-q-1}(z)$, $E_{M+q}(z)$, and $E_{2M-q-1}(z)$, the prototype filter $H(z)$ can be written in terms of $B_q(z)$, for $q = 0, 1, \cdots, M/2-1$, as

$$H(z) = \sum_{q=0}^{\frac{M}{2}-1} \left(B_q(z^M)z^{-q} + z^{-(2mM-1)}B_q(z^{-M})z^q \right), \tag{59}$$

where

$$B_q(z) = E_q(z^2) + z^{-1}E_{M+q}(z^2). \tag{60}$$

Therefore, $M/2$ length-$4m$ 2-subchannel PR prototype filters can be extracted from $H(z)$, where the qth filter that is completely specified by $\theta_{q,p}$, for $p = 0, 1, \cdots, m-1$, is $(B_q(z^2)+$

$z^{-(4m-1)}B_q(z^{-2})$). The $M/2$ sets of parameters can now be decoupled and optimized separately by using $M/2$ reference functions for the $M/2$ sets of $\theta_{q,p}$, for $q=0,1,\cdots,M/2-1$, that generate $B_q(z)$, for $q=0,1,\cdots,M/2-1$, respectively.

Assume that the parameters $\theta_p^{(2)}, p=0,1,\cdots,m-1$, of a 2-subchannel length-$4m$ PR filter $H^{(2)}(z)$ have been fully optimized using the method of [51], which can generate 2-subchannel PR prototype filters with high stopband attenuations. For the desired M, we can now define a length-$2mM$ filter $H^{(M)}(z)$ such that

$$h^{(M)}(n) = \sum_{i=0}^{4m-1} h^{(2)}(i) g\left(n - \frac{M(2i-4m+1)}{4}\right), \tag{61}$$

for $n=0,1,\cdots,2mM-1$, where $g(n)$ is the impulse response of a length-$2mM$ type-II real symmetric FIR filter with frequency response $G(e^{j\omega})$ that satisfies

$$\begin{cases} |G(e^{j\omega})-1| \ll \delta, & 0<\omega\le\pi/M, \\ 0\le|G(e^{j\omega})|\le 1+\delta, & \pi/M<\omega\le 3\pi/M, \\ |G(e^{j\omega})| \ll \delta, & 3\pi/M<\omega\le\pi, \end{cases} \tag{62}$$

where δ is the peak stopband ripple of $h^{(2)}(n)$. The process described by (61) represents an upsampling of the coefficients of $h^{(2)}(n)$ by a factor of M followed by downsampling by a factor of 2 such that the set of coefficients that is retained in the downsampling process is the set that does not contain the coefficients of $h^{(2)}(n)$. The impulse and frequency responses of the filters $h^{(2)}(n)$, $g(n)$, and $h^{(M)}(n)$ that are given in Fig. 26(a) and (b) illustrate this process for $m=3$ and $M=8$. In Fig. 26, a 2-subchannel PR filter $h^{(2)}(n)$ is interpolated using a length-48 filter $g(n)$ to give an 8-subchannel length-48 near-PR filter $h^{(8)}(n)$. In general, any filter $g(n)$ with a response that satisfies (62) can be used. However, it has been found that $g(n)$ filters that are designed using the robust Parks-McClellan equiripple FIR filter design method [75, 76], which is capable of producing very long filters for well chosen design parameters, provide the best results in terms of achieving the highest values of A_s of the desired PR prototype filters for a particular value of m. The near-PR filter $H^{(M)}(z)$ that results from (61) can be decomposed using its $2M$ length-m polyphase components

$E_q^{(M)}(z)$ as

$$H^{(M)}(z) = \sum_{q=0}^{\frac{M}{2}-1} \left(B_q^{(M)}(z^M) z^{-q} + z^{-(2mM-1)} B_q^{(M)}(z^{-M}) z^q \right), \quad (63)$$

where $B_q^{(M)}(z)$, for $q = 0, 1, \cdots, M/2-1$, is

$$B_q^{(M)}(z) = E_q^{(M)}(z^2) + z^{-1} E_{M+q}^{(M)}(z^2). \quad (64)$$

Now, the desired PR prototype filter $H(z)$ can be obtained by optimizing $\theta_{q,p}$, for $q = 1, 2, \cdots, M/2$ and $p = 0, 1, \cdots, m-1$, to minimize the much simpler objective function Ψ that is given by

$$\Psi = \sum_{i=0}^{2mM-1} \left| h(i) - h^{(M)}(i) \right|^2. \quad (65)$$

Using the polyphase representation and symmetry of $h(n)$ and $h^{(M)}(n)$, Ψ can be written as

$$\Psi = \sum_{q=0}^{2M-1} \sum_{n=0}^{m-1} \left| h(mq+n) - h^{(M)}(mq+n) \right|^2 = \sum_{q=0}^{M-1} \sum_{n=0}^{2m-1} \left| b_q(n) - b_q^{(M)}(n) \right|^2 = 2 \sum_{q=0}^{M/2-1} \psi_q, \quad (66)$$

where $b_q(n)$, and $b_q^{(M)}(n)$ are the impulse responses of $B_q(z)$ and $B_q^{(M)}(z)$, respectively, and

$$\psi_q = \sum_{n=0}^{2m-1} \left| b_q(n) - b_q^{(M)}(n) \right|^2. \quad (67)$$

Since ψ_q is positive for all q, Ψ is minimized by minimizing ψ_q, for all $q = 0, 1, \cdots, M/2-1$, independently. Minimizing ψ_q for a particular q requires that only $\theta_{q,p}$ for $p = 0, 1, \cdots, m-1$ be optimized. Upon minimizing Ψ, $H(z)$ that inherently satisfies PR is obtained. Because the complexity for minimizing Φ_1 or Φ_2 increases rapidly with increasing M, while it increases linearly with M for minimizing Ψ, replacing the $(mM/2)$-parameter optimization with $(M/2+1)$ optimizations of m parameters reduces the optimization complexity significantly. We estimate that the optimization complexity is reduced by a factor of at least $M/2$. While the original optimization method may be computationally impractical for large M, the modified optimization method is computationally feasible, particularly because the modified method does not require that the frequency response of $H(z)$ be computed at all and it does not suffer from local minima that limit the values of A_s that can be achieved. The modified modulated PR prototype filter design procedure is summarized in the following steps

1. Using the method in [51], the m parameters $\theta_p^{(2)}$, for $p = 0, 1, \cdots, m-1$, of $H^{(2)}(z)$ with stopband edge of $\omega_s = \pi/2$ are optimized. The value of m is selected such that the stopband attenuation of $H^{(2)}(z)$ exceeds that of the desired M-subchannel prototype $H(z)$ by several dBs.

2. Using the Parks-McClellan optimum equiripple filter design method, a length-$2mM$ filter $g(n)$ satisfying (62) is designed and used for computing the coefficients of $H^{(M)}(z)$ using (61).

3. The two sets of parameters $\theta_{M/4-1,p}$ and $\theta_{M/4,p}$, for $p = 0, 1, \cdots, m-1$, of $H(z)$ are initialized such that $\theta_{M/4-1,p} = \theta_{M/4,p} = \theta_p^{(2)}$, and are independently optimized to minimize $\psi_{M/4-1}$ and $\psi_{M/4}$.

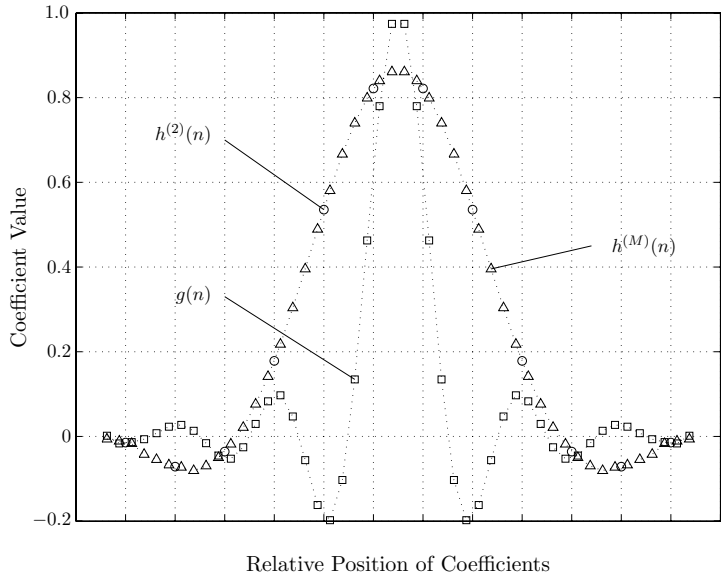
4. For each $q = M/4+1, M/4+2, \cdots, M/2-1$, in this order, the set $\theta_{q,p}$, for $p = 0, 1, \cdots, m-1$, is initialized with the optimum $\theta_{q-1,p}$ and optimized to minimize ψ_q.

5. For each $q = M/4-2, M/4-3, \cdots, 0$, in this order, the set $\theta_{q,p}$, for $p = 0, 1, \cdots, m-1$, is initialized with the optimum $\theta_{q+1,p}$ and optimized to minimize ψ_q.

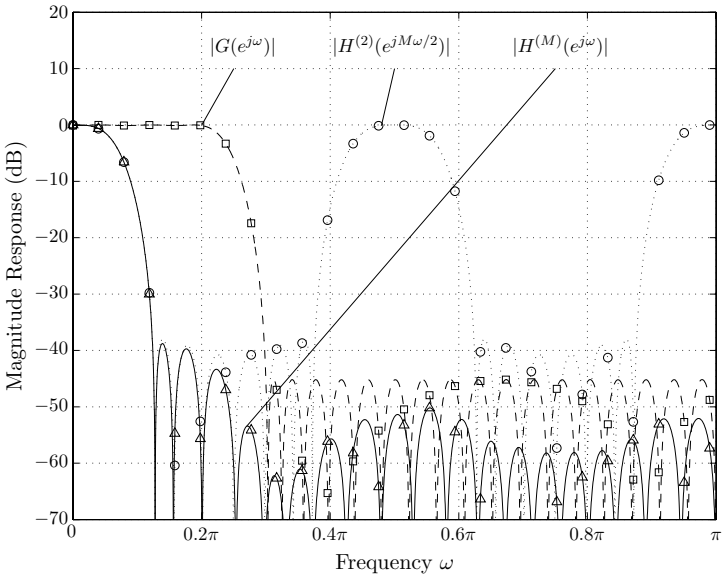
Theoretically, the presented design method is also applicable for odd values of M. However, the highest stopband attenuations of the prototype filters of modulated PR filter banks for odd values of M that can be achieved using this method are limited to approximately 50 dB even for large values of m. This limitation is attributed to the fact that two of the $2M$ polyphase components of odd-M PR prototype filters must be pure delays. That is, the $\lfloor M/2 \rfloor$th and $\lfloor 3M/2 \rfloor$th polyphase components of $H(z)$ must have the form βz^{-1} for the resulting filter bank to satisfy PR, where β is a real valued constant. These two polyphase components are obtained by setting the parameters in (55) as given in (57) [51]. The restriction on these two polyphase components results in limiting the highest stopband attenuation that can achieved for odd values of M.

6.2 Design Example

The design method of [51] was used for designing 2-subchannel PR prototype filters $H^{(2)}(z)$ for several values of m, where the optimized parameters $\theta_p^{(2)}$, for $p = 0, 1, \cdots, m-1$, are



(a)



(b)

Figure 26: Example of computing $H^{(M)}(z)$ with $M = 8$ from $H^{(2)}(z)$. (a) Impulse responses and (b) magnitude responses of $H^{(2)}(z)$, $G(z)$, and $H^{(M)}(z)$. The indexes of the coefficients in (a) start at $n = 0$ for each filter and the magnitude response of $H^{(2)}(z)$ in (b) is compressed in frequency.

given in Table 3. The corresponding $H^{(2)}(z)$ filters can be used in the modified modulated PR filter bank design procedure for obtaining prototype filters with the desired M and different stopband attenuations. The coefficients and magnitude response of a length-13312 PR prototype filter with $M = 512$ and $m = 13$ using $\theta_p^{(2)}$ for $m = 13$ from Table 3 were shown in Fig. 24(a) and (b) in Chapter 5. The stopband attenuation of the filter is 98.04 dB, and a total of 3328 parameters are optimized. Extensive testing of this method for designing modulated PR filter banks reveals that the use of a particular set of parameters $\theta_p^{(2)}$ produces prototype filters with stopband attenuations that are almost independent of the value of M.

Table 3: Parameters of two-channel modulated PR prototype filters for different values of m designed using the method of [51]

m	5	6	8	10	12	13	16
$\theta_0^{(2)}$	0.40219198	−1.67693504	0.65146437	0.53253686	−1.81942840	−0.99313915	−0.76813676
$\theta_1^{(2)}$	1.52549094	1.69757099	1.49392758	−1.60665880	0.53952129	2.17398213	1.44245600
$\theta_2^{(2)}$	2.47127910	−2.47520327	2.31327153	2.36657638	−1.57064445	−2.93523868	2.85452795
$\theta_3^{(2)}$	1.80925804	1.43855803	1.86981649	−1.12785327	2.85058881	1.55711415	0.52115001
$\theta_4^{(2)}$	1.25104224	1.99366341	−0.73623506	−1.25807305	−3.03039324	0.16200082	1.57773272
$\theta_5^{(2)}$		1.77243580	1.52296633	−1.70972783	1.52197876	−1.40917029	−0.54633659
$\theta_6^{(2)}$			−2.81754127	0.84831837	2.74628759	1.32902615	2.98867165
$\theta_7^{(2)}$			1.59262319	−2.14691373	−0.40947119	−2.36145492	−2.61155761
$\theta_8^{(2)}$				1.63404187	−1.50268944	−2.33216815	−1.54452991
$\theta_9^{(2)}$				2.07506752	0.84976274	3.01449850	2.91562060
$\theta_{10}^{(2)}$					−1.69535306	−1.53665178	2.65672516
$\theta_{11}^{(2)}$					−2.01445882	0.36295938	−1.27653894
$\theta_{12}^{(2)}$						−2.13685862	−1.67666867
$\theta_{13}^{(2)}$							1.18806135
$\theta_{14}^{(2)}$							−0.39418399
$\theta_{15}^{(2)}$							−1.43650270
ω_s	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$
A_s	48.96 dB	58.61 dB	75.42 dB	82.44 dB	92.50 dB	99.26 dB	109.17 dB

CHAPTER VII

BLIND FREQUENCY DOMAIN EQUALIZATION OF SEVERE ISI CHANNELS WITH LONG RESPONSES

Two main classes of blind equalization have been considered in the literature: second-order statistics (SOS) blind equalizers that use the cyclostationarity of oversampled received signal, and higher-order statistics (HOS) blind equalizers that rely on third-order statistics or higher to equalize the phase of the received signal [11, 27, 29, 34, 35, 44, 80, 86]. The use of HOS in the latter class of blind equalizers is attributed to the fact that second-order statistics lack the necessary phase information, which renders them unusable for equalizing the phase of two-dimensional digital signals transmitted over non-minimum-phase channels [13]. Although blind equalization requires longer time and more computations to adapt, it may be preferred over trained equalization in many applications when the initial transmission or the retransmission of a training sequence after a communication interruption is inapplicable or undesired [35]. For communication systems with single transmitter and multiple receivers, the transmission of a training sequence may be impractical if different receivers access the transmitted signal at different times. With the lack of training sequences, blind equalizers may be the only applicable equalization method for many applications.

In demanding communication applications that operate at high symbol rates like some SWR applications, the communication channels may have spreads extending over tens or hundreds of symbol periods and magnitude responses that exhibit high frequency-selective fading on the order of 50 dB, for example. For such applications to operate, equalization algorithms that are able to equalize these channels are necessary. While several TD-based blind equalizers have the ability of equalizing channels with severe ISI, the increase in complexity and the reduced ability to adapt as the channel length increases make them unsuitable for equalizing long channels. The FD blind equalizers that have been proposed in

the literature for equalizing very long channels are generally computationally efficient. However, they are able to equalize channels with relatively low ISI that have frequency-selective fading on the order of 20 dB or less because of the use of simple TD–FD transformations such as the FFT [100]. To the best of our knowledge, equalization of channels with long impulse responses and severe ISI that have high frequency-selective fading of 50 dB or more for QAM constellations have not been proposed. In this chapter, we present a relatively computationally efficient FD blind equalizer based on modulated exact- or near-perfect reconstruction filter banks for performing the TD–FD transformations in the equalizer. The equalizer presented in this chapter has the basic structure of the conventional FD equalizer described in [90] and shown in Fig. 27. The error functions used in the adaptation process in the equalizer discussed here are similar to those proposed by conventional LMS-based TD blind equalizers proposed in [28, 29, 86]. However, the speed of adaptation in our equalizer is increased and the computational requirements are reduced by modifying the LMS algorithms of the TD equalizers to FD-transformed RLS algorithms. The RLS algorithms in our equalizer are further modified to significantly improve the adaptation to long channels with severe ISI by breaking the set of equalizer weights into magnitude and phase weights that are optimized separately. Simulation results of the presented equalizer clearly show that it is able to equalize long channels with complicated frequency responses. The simulation results also show that this equalizer outperforms the FD blind equalizer that uses simple IFFT and FFT blocks for the TD–FD transforms for all channels with severe ISI at the expense of a relatively small increase in computation.

The remainder of this chapter is organized as follows. In Section 7.1, a brief discussion of HOS blind equalizers is given. Section 7.2 discusses the new FD blind equalizer, and Section 7.3 evaluates its computational requirements. Finally, the results of simulating our equalizer under different conditions are discussed in Section 7.4.

7.1 Bussgang HOS Blind Equalizers

The HOS (third-order or higher) of a two-dimensional digital signal contain magnitude and phase information of the signal that enables their use in blind equalizers to adjust

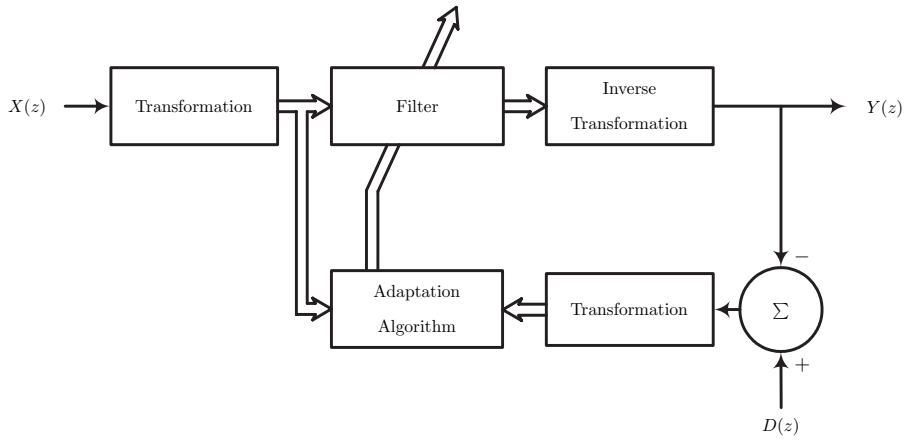


Figure 27: Basic FD equalizer ($D(z)$ is the desired response) [90].

for the magnitude and phase distortion of complex-valued channels. Since higher-order cumulants or polyspectra are zero for Gaussian random processes [35], the use of this type of HOS makes blind equalizers resilient to additive Gaussian noise because the higher-order cumulants and polyspectra of a noisy received signal are equal to the those of a noiseless form of the received signal [13]. Two special cases of the well studied Bussgang subclass of HOS blind equalization are the Sato algorithm [86] and Godard (or constant modulus) algorithm [29, 44, 55, 99]. The generalized Sato algorithm (GSA) [28] for blind equalization of two-dimensional symbol constellations minimizes

$$\begin{aligned}\mathbf{E}\Big\{|e^2_{\mathrm{GSA}}(n)|\Big\} = \mathbf{E}\Big\{&\Big[\Re\{y(n)\} - \gamma_{\mathrm{GSA}} \cdot \mathrm{sgn}(\Re\{y(n)\})\Big]^2 \\ &+ \Big[\Im\{y(n)\} - \gamma_{\mathrm{GSA}} \cdot \mathrm{sgn}(\Im\{y(n)\})\Big]^2\Big\},\end{aligned} \tag{68}$$

where $\mathbf{E}[\cdot]$ is the expectation operator, $y(n)$ is nth equalizer output sample, $\Re\{y(n)\}$ and $\Im\{y(n)\}$ are the real and imaginary parts of $y(n)$, respectively, $\mathrm{sgn}(\cdot)$ is the sign function, and

$$\gamma_{\mathrm{GSA}} = \frac{\mathbf{E}\Big[\Re\Big\{s^2(n)\Big\}\Big]}{\mathbf{E}\Big[\Big|\Re\{s(n)\}\Big|\Big]} = \frac{\mathbf{E}\Big[\Im\Big\{s^2(n)\Big\}\Big]}{\mathbf{E}\Big[\Big|\Im\{s(n)\}\Big|\Big]}, \tag{69}$$

where $s(n)$ is the nth transmitted symbol. The GSA uses an LMS update for the vector of coefficients $\mathbf{C}$ given by

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\Big\{\Big[\Re\{y(n)\} - \gamma_{\mathrm{GSA}} \cdot \mathrm{sgn}(\Re\{y(n)\})\Big]$$

$$+j\Big[\Im\{y(n)\} - \gamma_{\mathrm{GSA}} \cdot \mathrm{sgn}(\Im\{y(n)\})\Big]\Big\} y(n)\mathbf{X}^*(n), \tag{70}$$

where $\mathbf{X}(n)$ is a vector of the recently received samples, μ is the update step size, and * is the complex conjugate operator. The Godard algorithm (GA) minimizes

$$\mathbf{E}\Big[e_{\mathrm{GA}}^2\Big] = \mathbf{E}\Big[\Big(|y(n)|^2 - B_{\mathrm{GA}}\Big)^2\Big], \tag{71}$$

where

$$B_{\mathrm{GA}} = \frac{\mathbf{E}\Big[\Big|s(n)\Big|^4\Big]}{\mathbf{E}\Big[\Big|s(n)\Big|^2\Big]}. \tag{72}$$

The filter coefficient update for the Godard algorithm is also an LMS-like update given by

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\Big[\Big|y(n)\Big|^2 - B_{\mathrm{GA}}\Big] y(n)\mathbf{X}^*(n). \tag{73}$$

The GSA and GA TD blind equalizers generally have low computational requirements for short equalizers. However, the high computational complexity and slow convergence for long equalizer lengths renders them unsuitable for equalizing long channels.

In this chapter, we describe the details of a FD blind equalizer that attempts to minimize errors signals similar in form to those of the GSA and GA. While the conventional GSA and GA use LMS-like adaptation algorithms, this blind equalizer uses an RLS adaptation algorithm to speed the adaptation significantly for long complex-valued channels. Because this equalizer processes blocks of samples of the received signal at once, its efficiency measured as computations per equalized sample increases relative to conventional TD equalizers as the number of equalizer weights increases. While the complexity and delay of the equalizer increase linearly, its computational requirement increases in a logarithmic rate with the number of equalizer weights, which may result in a significant saving in computations for equalizers with an extremely large number of weights that may be required for equalizing channels with long impulse responses.

7.2 New Frequency-Domain Blind Equalizer

Adapting a TD equalizer to a channel with complex-valued impulse response requires the simultaneous optimization of the magnitudes and phases of all the equalizer coefficients.

Since the modification of a TD equalizer coefficient directly affects the whole frequency response of the equalizer, the optimization of a TD blind equalizer for long channels is often lengthy and computationally demanding. A FD equalizer, however, uses a transformation that divides the input signal into a set of partially or highly orthogonal narrowband signals that each can be equalized using a set of a few coefficients [90]. Each set of coefficients forms a filter that equalizes a signal with significantly narrower bandwidth than the received signal. The different sets may be partially or highly decoupled and, therefore, may be optimized independently. Applying an inverse transformation and properly recombining the equalized narrowband signals produces an equalized form of the received signal. The discrete and inverse discrete Fourier transforms (DFT an IDFT) may be suitable transformation and inverse transformation for the FD equalization applications shown in Fig. 27. Practically, a FD equalizer that uses the DFT and IDFT experiences debilitating limitations. The DFT and IDFT result in equalizer coefficients that are significantly dependent because the mainlobe of the frequency response of a particular DFT output is wide and the sidelobes are relatively large and decay slowly. The first sidelobe, for example, is only 13.26 dB lower than the mainlobe. Therefore, the orthogonality of the outputs of a DFT is generally low.

Dividing the input signal of the equalizer into highly orthogonal narrowband signals requires the use of higher-quality transformations such as filter banks that have filters with relatively high stopband attenuation and low filter-passband overlap. Since the output of an equalizer in the absence of channel ISI and noise should be a delayed form of the input signal, an appropriate transformation and inverse transformation for Fig. 27 would be the analysis and synthesis parts of a perfect reconstruction (PR) filter bank, respectively. A PR filter bank is computationally efficient when its filters can be polyphase decomposed, which requires that the bank be a modulated PR filter bank [16, 48, 51, 58, 109]. The outputs of the analysis section of a modulated PR filter bank are generally more orthogonal to each other compared to the outputs of a DFT.

For a complex-exponentially modulated (CEM) PR filter bank with $2M$ subbands and a length-$2mM$ prototype filter $h(n)$, the kth analysis and synthesis filters $h_k(n)$ are given

by

$$h_k(n) = h(n)W_{2M}^{-k\left(n-\frac{2mM-1}{2}\right)}, \tag{74}$$
$$n = 0, 1, \cdots, 2mM-1, \qquad k = 0, 1, \cdots, 2M-1,$$

where $W_M = e^{-j2\pi/M}$, and m is a positive integer that determines the length and, therefore, the stopband attenuation of the prototype and the filters in the bank. The z-transforms of the analysis and synthesis filters can be written in terms of their polyphase components as

$$\begin{aligned} H_k(z) &= \sum_{n=0}^{2mM-1} h(n)W_{2M}^{-k\left(n-\frac{2mM-1}{2}\right)}z^{-n} \\ &= \sum_{q=0}^{2M-1}\sum_{r=0}^{m-1} h(q+2Mr)W_{2M}^{-k\left(q+2Mr-\frac{2mM-1}{2}\right)}z^{-q-2Mr} \\ &= \sum_{q=0}^{2M-1} W_{2M}^{-k\left(q-\frac{2mM-1}{2}\right)}E_q(z^{2M})z^{-q}, \end{aligned} \tag{75}$$

where $E_q(z)$, for $q = 0, 1, \cdots, 2M-1$, are the $2M$ polyphase components of $H(z)$ such that

$$E_q(z) = \sum_{r=0}^{m-1} h(q+2Mr)z^{-r}. \tag{76}$$

The IDFT-DFT cascade is therefore a special case of the modulated PR filter bank shown in Fig. 18 with $E_q(z)$ being equal to some constant c for all $q = 0, 1, \cdots, 2M-1$. Define the correlation C_p between the frequency responses of the kth and $(k+p \bmod 2M)$th analysis filters, $H_k(z)$ and $H_{(k+p \bmod 2M)}(z)$, that have passbands centered $p\pi/M$ radians apart as

$$\begin{aligned} C_p &= \frac{\int_0^{2\pi} |H_k(e^{j\omega})H_{(k+p \bmod 2M)}(e^{j\omega})|d\omega}{\int_0^{2\pi} |H_k(e^{j\omega})|^2 d\omega}, \\ & \quad p = 0, 1, \cdots, 2M-1, \end{aligned} \tag{77}$$

where mod is the modulus operator. A plot of C_p, $p = 0, 1, \cdots, 2M-1$, for a $(2M = 64)$-point IDFT filter bank is shown in Fig. 28. The figure also shows C_p for $(2M = 64)$-band CEM-modulated exact-PR filter banks with $m = 5$ and $m = 13$ that have prototype stopband attenuations of 50.34 dB and 98.58 dB, respectively, and stopband edge frequencies $\omega_s = \pi/M$. Note that C_p, $p \neq 0$ for the IDFT bank is relatively large compared to that of the modulated PR filter banks with $m = 5$ and $m = 13$. For modulated PR filter banks, C_p, $p \neq 0$, is roughly equal to the stopband attenuation of the filters in the banks, except

for $p = 1$ and $2M - 1$. Therefore, two narrowband signals produced by the analysis section of a CEM-modulated PR filter bank with a sufficiently large value of m may be considered orthogonal if they are separated by at least one output signal. Moreover, the relatively high stopband attenuation of the analysis filters and the narrow bandwidth of the filters when prototype filters with $\omega_s = \pi/M$ are used reduce the aliasing from the downsampling-by-M operation to a value on the order of the stopband attenuation of the prototype filter.

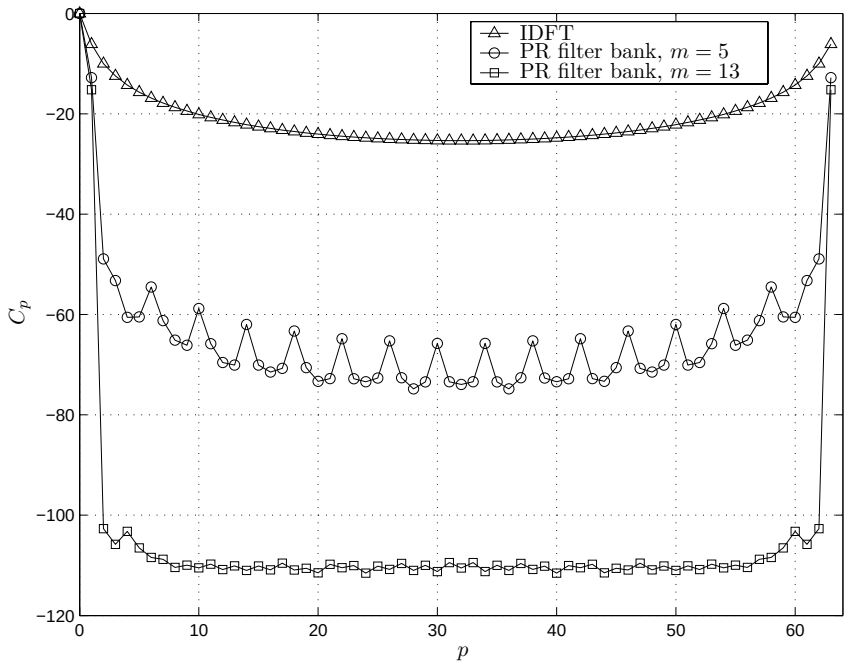


Figure 28: The magnitude of the correlation C_p versus p for the DFT, and the modulated PR filter banks with $m = 5$ and $m = 13$.

The adaptation of blind equalizer in systems that use symbol constellations with either amplitude or phase information but not both is significantly simpler than the adaptation for symbol constellations with both amplitude and phase information such as QAM. The simplicity of the adaptation in the former case results from the knowledge of either the phase of amplitude of the transmitted symbols, which can be used to simplify the adaptation process. The adaptation of blind equalizers for QAM signals using conventional FD adaptation techniques is difficult when the ISI introduced by the channel is severe. For a QAM constellation, several factors contribute to both the magnitude and phase of the

error between a particular sample of the received signal and the corresponding transmitted symbol. The magnitude and phase distortions of the communication channel, channel noise, and possible miss-synchronization between the transmitter and receiver contribute to both the magnitude and phase parts of the error between a transmitted sample and the corresponding received sample. Applying simple adaptation algorithms often fails to adapt blind equalizers for complicated channels with long impulse responses that lack a strong line-of-sight path. The suggested blind equalizer with the basic structure of Fig. 27 uses the analysis and synthesis sections of a modulated PR filter bank for the transformation and inverse transformation blocks, respectively. The required modulated PR filter bank may have a large number of subbands, $2M$, to equalize long channels and a prototype filter that has high stopband attenuation A_s and stopband edge frequency $\omega_s = \pi/M$.

Unlike TD equalizer weights, which are the coefficients of a filter, the weights in the proposed FD blind equalizer are the complex-valued gains, G_k, for $k = 0, 1, \cdots, 2M-1$, shown in Fig. 29. These weights scale the narrowband outputs of the analysis section of the filter bank. The complex gains G_k in the proposed equalizer are decomposed into three factors: R, $|A_k|$, and $P_k/|P_k|$, such that

$$G_k = R \cdot |A_k| \cdot \frac{P_k}{|P_k|}. \tag{78}$$

As will be discussed later, the factor $|A_k|$ adjusts the magnitude of the kth narrowband signal $Q_k(z)$ in the proposed equalizer shown in Fig. 30, $P_k/|P_k|$ adjusts the phase of $Q_k(z)$, and R is a unit vector that rotates the constellation of the equalizer output signal to align it with that of the transmitted signal.

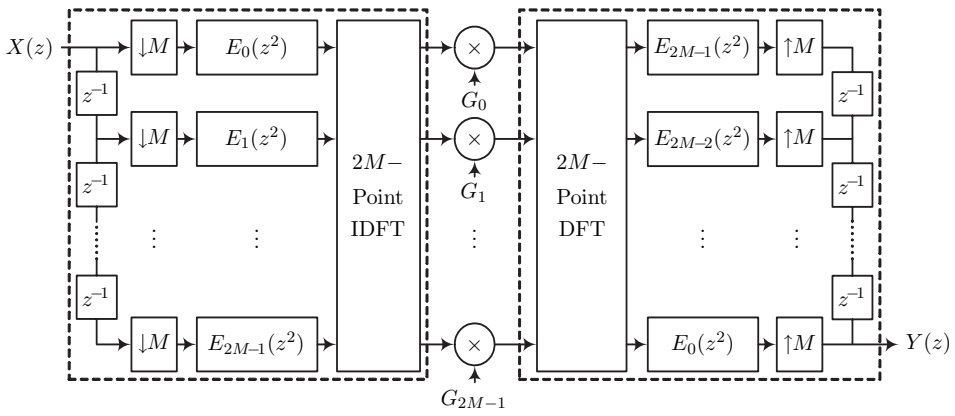


Figure 29: The FD FIR filter. $E_k(z^2)$, $k = 0, 1, \cdots, 2M-1$ are the polyphase components of the prototype filter $H(z)$ of the filter bank.

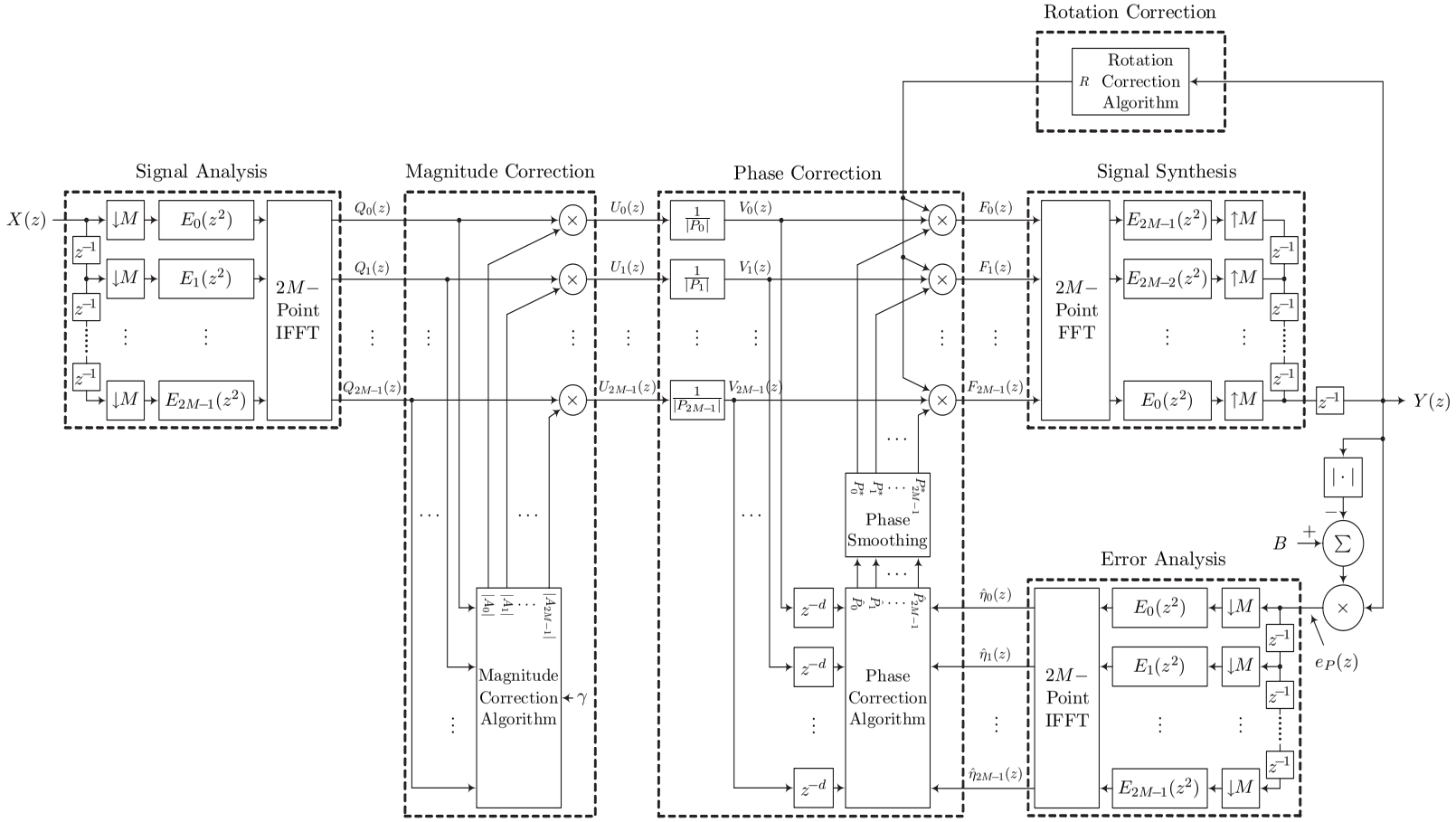


Figure 30: The construction of the new FD blind equalizer.

After R, A_k, and P_k for all $k = 0, 1, \cdots, 2M-1$ have been optimized such that an error signal is minimized, samples of the transmitted and equalized signals will differ because of additive noise and residual ISI due to imperfect channel inversion. In the absence of noise the error between the transmitted signal $T(z)$ and the equalizer output signal $Y(z)$ is

$$T_e(z) \quad = \quad T(z) - Y(z), \tag{79}$$

where $Y(z)$ shown in Fig. 29 is given by

$$Y(z) = \frac{1}{M} \sum_{p=0}^{M-1} C\left(zW_M^p\right) X\left(zW_M^p\right) \sum_{k=0}^{2M-1} G_k H_k\left(zW_M^p\right) H_k(z), \tag{80}$$

and $C(z)$ is the z-transform of the communication channel. The successful equalizer adaptation reduces the power of the error signal $T_e(z)$ sufficiently to permit detection of the received symbols with a relatively low probability of error. Fig. 31 shown the magnitude response of a typical channel with two frequencies at which the response drops below the peak response of the channel by approximately 35 dB. The figure also shows the power spectrum density (PSD) of the error $T_e(z)$ for a noiseless 16-QAM received signal with independent identically-distributed (iid) symbols that is equalized by a modulated PR filter bank equalizer with a value of $M = 32$. As illustrated in Fig. 31, the highest values of the PSD of $T_e(z)$ occur at the zeros of the channels since the equalization of channel zeros is difficult because they represent a loss of information. Channel poles also cause a similar effect as channel zeros. To improve the performance of the equalizer, the value of M used in the modulated PR filter bank is increased, which permits a more accurate approximation of the channel inverse response and, therefore, reduces the error caused by the imperfect equalization of channel poles and zeros. It is seen that $\Re\{t_e(n)\}$ and $\Im\{t_e(n)\}$ generally approximate zero-mean Gaussian distributed random processes, where $t_e(n)$ is the TD representation of $T_e(z)$. The powers of $\Re\{t_e(n)\}$ and $\Im\{t_e(n)\}$ are determined by the complexity of the channel and, particularly, by the number of zeros and poles of the channel. Bit errors in the output of the equalizer are, therefore, caused by the combination of the two two-dimensional zero-mean Gaussian random processes: the channel noise and the error of equalization $T_e(z)$. The noise is generally the dominant cause of bit errors at low SNRs and the error signal $T_e(z)$ becomes the dominant source of bit errors at high SNRs.

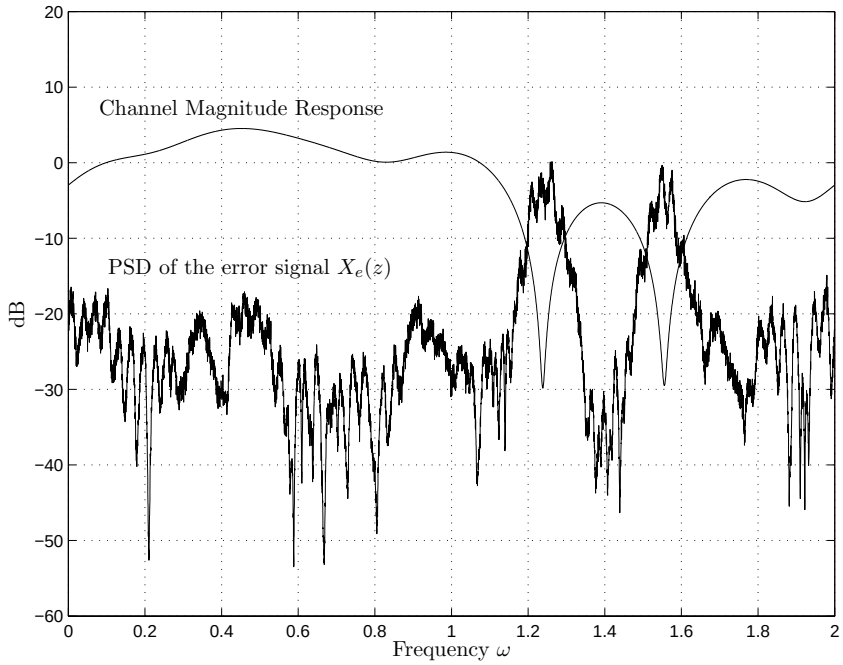


Figure 31: The magnitude of the correlation C_p versus p for the DFT, and the modulated PR filter banks with $m = 5$ and $m = 13$.

The function of all equalizers, including the proposed one, is not to perfectly reproduce the transmitted signal but only mitigate the ISI sufficiently to adequate signal detection. Hence, the use of modulated near-PR filter banks may be preferred over modulated exact-PR filter banks in the proposed equalizer, since near-PR filter banks require shorter prototype filters than exact-PR filter banks to achieve the same stopband attenuation. The increase in the probability of bit error due to the use of near-PR filter banks as opposed to their exact-PR counterparts is generally very small.

The adaptation of the equalizer parameters in the suggested blind equalization method is performed by optimizing the complex gains G_k, for $k = 0, 1, \cdots, 2M-1$, between the analysis and synthesis sections of a near- or exact-PR filter bank in three simultaneous processes. While optimizing these complex parameters is relatively simple in the case of trained or decision-directed equalization, it is difficult for blind equalization, especially for severe ISI-causing long channels. The availability of only the statistics of the desired output of a blind

equalizer makes the adaptation of the weights much slower and more complicated, especially with received signals that have closed-eye constellations. The slow adaptation is attributed to the high number of samples that are required to estimate HOS of the received signal. The adaptation algorithms proposed for FD equalizers in the literature perform the weight updates by adjusting the magnitudes and phases of the weights using a single adaptation block. In the next subsections, we describe an optimization method that adapts the phases of the equalizer weights separately from the magnitudes by using magnitude– and phase–adjusting modified RLS algorithms. The phase–adjusting RLS algorithm further includes a block to that smooths the phase response of the equalizer, which is found to significantly improve the adaptation of the phase weights, especially for complicated channels.

7.2.1 Adapting the Magnitude Weights A_k

White QAM signals with iid symbols become colored when transmitted through a channel that causes magnitude distortion. To correct for the magnitude distortion of the channel, we define A_k, $k = 0, 1, \cdots, 2M-1$, as a set of complex-valued weights such that $|A_k|$ is used to multiply the signal $Q_k(z)$ as shown in the magnitude correction block in Fig. 30. Since R in (78), as will be discussed later, is a unit vector, $|G_k| = |A_k|$, for $k = 0, 1, \cdots, 2M-1$. The magnitudes of A_k are responsible for adjusting the PSD of the received signal to approximate the PSD of the transmitted signal. For a white transmitted signal, the magnitude correction block works to whiten the received signal and therefore, the cascade of the channel and the equalizer described here with properly optimized A_k approximates an all-pass filter. For a signal $U_k(z) = |A_k|Q_k(z)$, the kth weight A_k can be optimized such that

$$\begin{aligned}\mathbf{E}\Big[\Big|\eta_k(n)\Big|^2\Big] = \mathbf{E}\Big\{ &\Big[\Re\{u_k(n)\} - \gamma_A \operatorname{sgn}(\Re\{u_k(n)\})\Big]^2 \\ &+ \Big[\Im\{u_k(n)\} - \gamma_A \operatorname{sgn}(\Im\{u_k(n)\})\Big]^2\Big\} \end{aligned} \tag{81}$$

is minimized, where $u_k(n)$ is the TD representation of $U_k(z)$ shown in Fig. 30. The optimum value for γ_A depends on several factors: the statistics $\mathbf{E}[|\Re\{s(n)\}|]$ of the transmitted signal, the response of the prototype filter $H(z)$, and the length m of its $2M$ polyphase components.

Table 4 lists optimized values of γ_0 for four different filter banks, where

$$\gamma_A = \gamma_0 \cdot \mathbf{E}\Big[\Big|\Re\{s(n)\}\Big|\Big]. \tag{82}$$

The values of γ_0 in Table 4 are those that resulted in the lowest average number of iterations required by the equalizer to adapt to a set of randomly generated channels of different lengths at different SNRs. The use of significantly lower or higher values of γ_0 generally result in the failure of the equalizer to adapt to the channel. The values of γ_0 in Table 4 assume that the $2M$-point DFTs and IDFTs used in the filters have scaling factors of 1 and $1/2M$, respectively, and that the coefficients of prototype filter are normalized such that

$$\sum_{n=0}^{2mM-1} |h(n)|^2 = M. \tag{83}$$

Table 4: Values of γ_0 for different modulated filter banks

Filter Bank Type	γ_0
IDFT/DFT	$1.0000/\sqrt{2M}$
Modulated PR Filter Bank with $m = 3$	$0.9945/\sqrt{2M}$
Modulated PR Filter Bank with $m = 5$	$0.9922/\sqrt{2M}$
Modulated PR Filter Bank with $m = 13$	$0.9392/\sqrt{2M}$

The magnitude correction algorithm updates the magnitudes and the phases of the weights A_k; however, only the magnitudes $|A_k|$ are used in the equalizer. The phases of A_k, represented by $A_k/|A_k|$, are retained for the adaptation algorithm only. Using the initializations $D_k(0) = \delta_A$ and $A_k(0) = 1$, for $k = 0, 1, \cdots, 2M-1$, where δ_A is a small positive number, an RLS algorithm given by the set of equations

$$S_k(n) = \frac{D_k(n)\, Q_k(n)}{\lambda + D_k(n)\, |Q_k(n)|^2}, \tag{84}$$

$$U_k(n) = |A_k(n)| Q_k(n), \tag{85}$$

$$\eta_k(n) = -\Big[\Big(\Re\{U_k(n)\} - \gamma_A \operatorname{sgn}(\Re\{U_k(n)\})\Big) + j\Big(\Im\{U_k(n)\} - \gamma_A \operatorname{sgn}(\Im\{U_k(n)\})\Big)\Big], \tag{86}$$

$$A_k(n+1) = A_k(n) + \Re\Big\{S_k(n)\eta_k^*(n)\Big\}, \tag{87}$$

$$D_k(n+1) = \lambda^{-1}\Big[D_k(n) - S_k(n)\, D_k(n)\, Q_k^*(n)\Big], \tag{88}$$

$$k = 0, 1, \cdots, 2M-1,$$

can be used for updating the kth weight A_k. The initialization $A_k(0) = 1 \ \forall \ k$ assumes an automatic gain control (AGC) that adjusts the rms value of the equalizer input signal to a value close to unity. If AGC is not used, $A_k(0)$ are all set to the rms value of the input signal.

7.2.2 Adapting the Phase Weights P_k

The magnitude correction block is followed by a phase correction block in the proposed equalizer as shown in Fig. 30. The phase correction block computes a set of weights $P_k, k = 0, 1, \cdots, 2M-1$, and the phases $P_k/|P_k|$ are used to modify the phases of the magnitude-corrected narrowband signals $U_k(z)$. After the magnitude distortion of the received signal has been corrected by adjusting the weights A_k, the remaining ISI is mainly caused by the phase distortion. This phase distortion results in an error between an output sample of the equalizer and the corresponding transmitted sample. The radial part of this error provides sufficient information that permits adjust the phase weights. Since both the magnitude and phase distortions of the channel contribute to the radial part of this error, and since updating the magnitude weights A_k takes precedence over updating the phase weights P_k, the adaptation of P_k generally occurs when A_k have approached the optimum values.

Assume that $A_k, k = 0, 1, \cdots, 2M-1$ have been optimized. The phase weights are optimized to minimize the power of the error $\hat{e}_P(n)$ in the output signal given by

$$\mathbf{E}\left[\left|\hat{e}_P(n)\right|^2\right] = \mathbf{E}\left[\left|y(n)(B_P - |y(n)|)\right|^2\right], \tag{89}$$

where $B_P = \mathbf{E}\{|s(n)|^2\}$. The error $\hat{e}_P(n)$ is a TD error that is transformed by the error analysis block of the equalizer to yield $\hat{\eta}_k(n), k = 1, 2, \ldots, 2M$, using a transformation similar to that of the equalizer input signal as shown in Fig. 30. The FD error signals $\hat{\eta}_k(n)$ can be used for adapting P_k. The optimization of P_k in the phase correction block of Fig. 30 using an RLS algorithm demands that only the phases but not the magnitudes of the input signals to this block be altered. This is achieved by dividing the inputs to the

phase correction block, $U_k(n)$, by $|P_k|$ producing the signals

$$V_k(n) = \frac{U_k(n)}{|P_k(n)|}, \qquad (90)$$
$$k = 0, 1, \cdots, 2M-1.$$

The weight P_k, for all k are then updated using properly delayed forms of the signals $V_k(n)$ and the FD error signals $\hat{\eta}_k(n)$ by using the RLS algorithm described by the set of equations

$$\hat{S}_k(n) = \frac{\hat{D}_k(n)\, V_k(n-d)}{\lambda_P + \hat{D}_k(n)\, |V_k(n-d)|^2}, \qquad (91)$$

$$\hat{P}_k(n+1) = P_k(n) + \hat{S}_k(n)\, \hat{\eta}_k^*(n), \qquad (92)$$

$$P_k(n+1) = \left|\hat{P}_k(n+1)\right| \left[(1-2\alpha)\frac{\hat{P}_k(n+1)}{|\hat{P}_k(n+1)|} + \alpha\left(\frac{\hat{P}_{k-1}(n+1)}{|\hat{P}_{k-1}(n+1)|} + \frac{\hat{P}_{k+1}(n+1)}{|\hat{P}_{k+1}(n+1)|}\right)\right], \qquad (93)$$

$$\hat{D}_k(n+1) = \hat{\lambda}_P^{-1}\Big[\hat{D}_k(n) - \hat{S}_k(n)\, \hat{D}_k(n)\, V_k^*(n-d)\Big], \qquad (94)$$

$$k = 0, 1, \cdots, 2M-1,$$

where $\hat{P}_{2M}(n+1) = \hat{P}_0(n+1)$, and α is a positive small number. The RLS algorithm described in (91-94) uses the initializations $P_k(0) = 1$ and $\hat{D}_k(0) = 1/\delta_P$, for all $k = 0, 1, \cdots, 2M-1$. Our observation of the magnitudes $|P_k|$ reveals that they continuously decrease during the optimization. The magnitudes $|P_k|$ play a role in the optimization process but not in the equalization and, therefore, the magnitudes of P_k can be reset if they approach the processor precision, although this may cause the loss of several iterations to enable contents of all affected memory elements to be flushed out.

The purpose of updating $\hat{P}_k$ and using (94) for obtaining the actual weights P_k is that using $P_k(n+1) = \hat{P}_k(n+1)$, for example, instead of (94) often results in the failure of the RLS algorithm to adapt P_k properly to channels with significant amplitude and/or phase distortion. This is caused by the remaining ISI from the imperfect whitening of the received signal interfering with the process of adapting P_k. To improve the convergence of the P_k, and to permit convergence in situations where convergence was found to difficult to achieve, the phase smoothing process described by (94) and represented by the phase smoothing block in Fig. 30 is applied. This process is based on the fact that the phases of P_k

for adjacent values of k are close when using filter banks with large values of M. Generally, values of α in the range $0.005 - 0.01$ work well for exact- and near-PR filter banks having various values of M and m. The final outputs of the phase correction block $F_k(z)$ are then given by

$$F_k(n) = P_k^*(n)R(n)V_k(n), \tag{95}$$

where the weight $R(n)$ causes a rotation of the constellation of the output signal as discussed in the next subsection. Applying the inverse transformation of the equalizer to the set of F_k signals produces the output signal of the equalizer.

7.2.3 Adapting the Rotation Weight R

After the magnitude A_k and phase P_k weights in the equalizer have been optimized, the equalizer output signal may exhibit a rotated signal constellation. The reason is that the error $\hat{e}_P(n)$ used to optimize the P_k weights provides information about the relative phases of the different $U_k(n)$ signals, but not their absolute phases. This rotation must be corrected before making decisions on samples of the equalized signal. For a constellation with symbols that have one of N symbol-magnitudes $\{a_1, a_2, \cdots, a_N\}$, with $a_1 \leq a_2 \leq \cdots \leq a_N$, the rotation angle θ_R is initialized with $\theta_R = 0$ and updated using

$$\theta_R(n+1) = \begin{cases} \theta_R(n), & D_a\Big(|y(n)|\Big) \neq a_N \\ \theta_R(n) + \mu_R \Im\Big\{y(n)\Big[y(n) - \hat{D}_s^{a_N}(y(n))\Big]^*\Big\}, & D_a\Big(|y(n)|\Big) = a_N, \end{cases} \tag{96}$$

where μ_R is the adaptation step size. The operation $D_a(|y(n)|)$ is a decision that outputs the magnitude from $\{a_1, a_2, \cdots, a_N\}$ that is nearest to $|y(n)|$, and $\hat{D}_s^{a_N}$ is a decision that outputs the symbol from the set of symbols having magnitude a_N that is nearest to $y(n)$. The weight R that is used to correct for the rotation in the constellation of the equalized signal is

$$R(n+1) = \cos\Big[\theta_R(n+1)\Big] + j \sin\Big[\theta_R(n+1)\Big]. \tag{97}$$

The process of updating θ_R uses only symbols that belong to the set having the largest magnitude because they give the highest accuracy. The system may exhibit rotation ambiguity equal to the angles separating the different symbols with magnitude a_N.

7.3 Computational Requirements of the Suggested FD Equalizer

The main parameters that determine the capabilities and computational requirements of the proposed blind equalizer are the stopband attenuation A_s and stopband edge frequency ω_s of the prototype filter and the number of subbands $2M$ of the filter bank. While M determines the maximum channel length that can be equalized, the value of A_s determines the maximum channel distortion that can be corrected and ω_s determines the amount of correlation between the different equalizer weights and, therefore, the equalizer adaptation speed. The values of A_s and M are also directly related to the computational requirements of the equalizer. Since the equalizer uses DFT and IDFT blocks, the computations are most efficient when M is a power-of-two, where fast Fourier transforms (FFT) and inverse fast Fourier transforms (IFFT) can be used. This section derives the computational requirements of the proposed equalizer during the phases of adaptation and equalization and compares them to those of conventional TD blind equalizers.

The downsampling-by-M operation in the analysis sections of the equalizer results in most of the processing being performed at the low sample rate of $(1/M)$th the sample rate of the equalizer input signal. Therefore, each of the three sets of polyphase filters performs $4m$ real multiplications per output sample (MPOS) and each of the IFFTs and FFT performs $4\log_2(2M)-4$ MPOS. The computational requirements for the RLS algorithms of A_k and P_k for all $k = 0, 1, \cdots, 2M-1$ are approximately 66 and 122 MPOS. The adaptation process of θ_R for a 16-QAM signal constellation requires approximately 2 MPOS on average and less for higher-order QAM constellations. Therefore, the total number of real MPOS that the equalizer requires during the adaptation phase is, approximately,

$$\Gamma_{\text{Adaptation}} = \Big[12m + 12\log_2(2M) + 178\Big] \quad \text{MPOS}, \tag{98}$$

and during the equalization phase, assuming no further adaptation is performed, is

$$\Gamma_{\text{Equalization}} = \Big[8m + 8\log_2(2M)\Big] \quad \text{MPOS}. \tag{99}$$

The proposed equalizer has been observed to reliably adapt to severe-ISI channels of lengths on the order of $2M/11$. For an $(L = 80)$-channel with a 50 dB spread between the highest

and lowest values of its magnitude response, for example, a power-of-two $M = 512$ and a value of $m = 5$ for the modulated PR filter bank of the proposed equalizer may be required for a 16-QAM signal. This equalizer requires approximately 358 and 120 MPOS during adaptation and equalization phases, respectively.

For equalizing channels of length L, conventional equalizers may require FIR filters of lengths $2L - 3L$ for low SNR conditions and lengths up to $5L$ for high SNR conditions for providing the best performance [84]. Therefore, assuming that conventional GA or GSA blind equalizers are able to equalize channels of length $L = 80$, for example, that have complicated frequency responses, the length of the complex-valued FIR filters required for arbitrary SNRs would be on the order of $5L$. This means that conventional blind equalizers require on the order of $20L$ MPOS for the filtering process and $20L$ MPOS for the filter coefficients update process. For the ($L = 80$)-channel, a conventional TD blind equalizer therefore requires roughly 3200 MPOS during adaptation and 1600 MPOS during equalization, which are approximately 9 and 13 times the MPOS, respectively, required by the proposed equalizer. The required number of weights in the proposed and conventional blind equalizers generally increase as the channel lengths increase. However, the efficiency of the proposed equalizer compared to that of a conventional TD blind equalizer increases, since the computations of the proposed blind equalizer are on order $O(\log_2 M)$ while the computations of the conventional equalizer are on order $O(M)$. Conventional block blind equalizers generally fail to equalize severe-ISI channels because of the significant correlation between the equalizer weights that restricts the values these weights can have.

7.4 Simulation Results

The new FD blind equalization method was implemented and tested for different channels with different characteristics and lengths at different SNR ratios. The SNR used in the simulations is defined as the ratio of the power of the desired part of the signal at the input of the equalizer to the power of the additive noise, and has the form

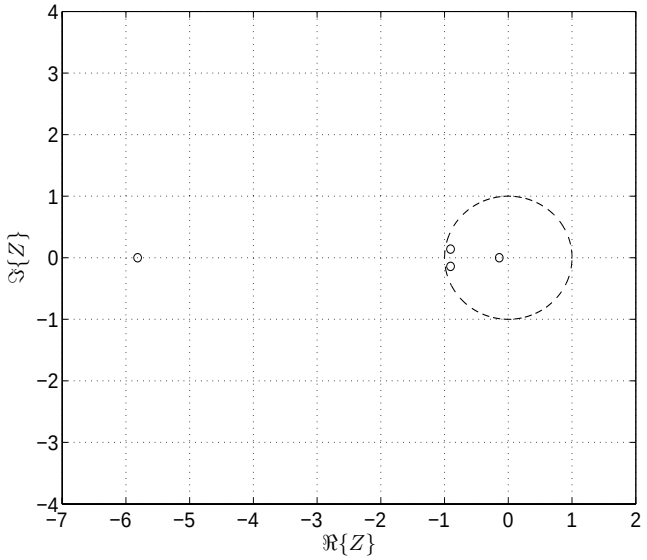
$$\text{SNR} = \frac{\sigma_S^2 \sum_{n=0}^{L-1} |c(n)|^2}{\sigma_N^2}, \tag{100}$$

where σ_S^2 is the power of the transmitted signal, $c(n)$ is the nth coefficient of the channel, and σ_N^2 is the power of the additive noise.

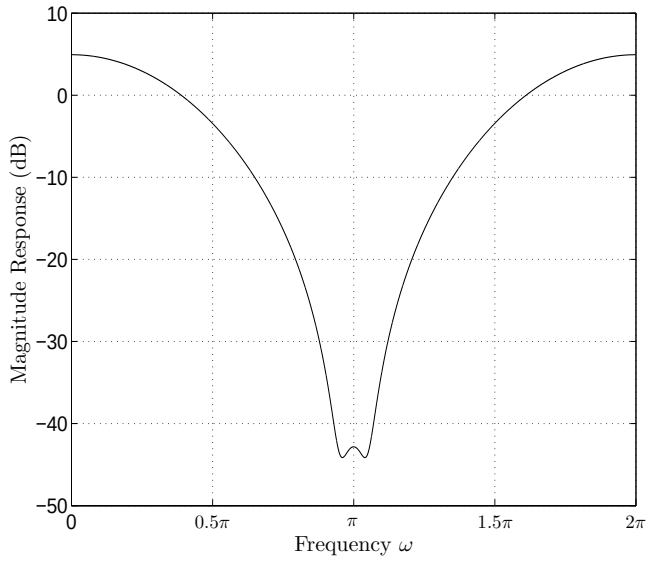
Example 1

The performance of the new blind equalization method in terms of the bit error rate (BER) in the equalized signals versus the SNR is studied using near-PR filter bank equalizers with $m = 3$ and 5 and for $M = 128$ and 256. The measurement of bit errors occurs while the equalizer weights are fixed after an initial set of iterations during which the equalizer weights are allowed to adapt. The performance of the new equalization method is compared to that of a IFFT/FFT equalizer that uses the same adaptation method. The input signals to the different equalizers are 16– and 64–QAM signals that have been transmitted through the real-valued channel with the zero-pole and magnitude response plots that are shown in Fig. 32. This channel is characterized by a gradual drop in the magnitude response that results in an attenuation of roughly 49 dB of 0.03 of the signal band centered at half its sampling rate. Such a channel illustrates the effect of the loss of different portions of the received signal due to noise on the ability of an equalizer to adapt and equalize the received signal. The parameters used in the adaptation algorithm for all equalizers, including the IFFT/FFT equalizers, are given in Table 5.

The BER plots (assuming the use of gray coding) of the different equalizers are shown in Fig. 33. Although equalizers that use prototypes with A_s less than the magnitude response spread of the channel of approximately 49 dB are able to successfully adapt for high SNRs, they fail to reduce the BER of the equalized signals below 0.05 regardless of the SNR. For the new equalizer using prototypes with $A_s > 49$ dB (i.e., $m = 5$, the BER generally drops as the SNR is increased. The effect of increasing M from 128 to 256 at a particular SNR is seen to reduce the BER since more accurate approximation of the channel inverse can be achieved. Changing the modulation from 16–QAM to 64–QAM significantly increases the BER at a specific SNR since the Euclidean distance between the different symbols is reduced. The values of $|A_k|$ and phases of P_k for $k = 0, 1, \cdots, 511$ of the equalizer with $m = 5$ and $M = 256$ at a SNR of 50 dB are shown in Fig. 34(a) and (b), respectively. Parts (c) and (d) of Fig. 34 show a set of samples of the input and output (equalized) signals of



(a)



(b)

Figure 32: The channel used in Example 1, (a) zero-pole plot, and (b) magnitude response $|C(e^{j\omega})|$.

Table 5: Parameters of the new FD blind equalizer in Example 1

Parameter	Value
Modulation Type	16– and 64-QAM, iid symbols
Maximum Number of Iterations for Adaptation	8000
SNR	25–65 dB
M	128 and 256
m	1 (IFFT/FFT), 3 and 5
ω_s of the Prototype Filter	π/M
A_s of Prototype Filter	~ 13 dB (IFFT/FFT) ~ 40 dB($m = 3$) ~ 50 dB($m = 5$)
δ_A	0.01
λ_A	0.999
δ_P	0.01
λ_P	0.985
γ_0	As in Table 4
μ_R	0.0005
ϵ	0.00655

the equalizer. For lower values of the SNR, where the PSD of the noise exceeds the PSD of the desired part of the received signal at some frequencies, the magnification of the noise stops when its PSD at these frequencies reaches the PSD of the transmitted signal and, therefore, noise enhancement is limited.

Example 2

This example illustrates the performance of the new and the IFFT/FFT equalizers under severe ISI resulting from the transmission of a 16–QAM signal through a complex-valued FIR channel of length 90 with random coefficients. The real and imaginary parts of the impulse response, the magnitude response, and the zero-pole plot of the channel are shown in Fig. 35. The magnitude response of the channel clearly shows the severity of the ISI that a transmitted signal through such a channel experiences, where the channel contains several near-zero values of the magnitude response. The length and magnitude response complexity of this channel is roughly the maximum that the new equalizer with $M = 512$

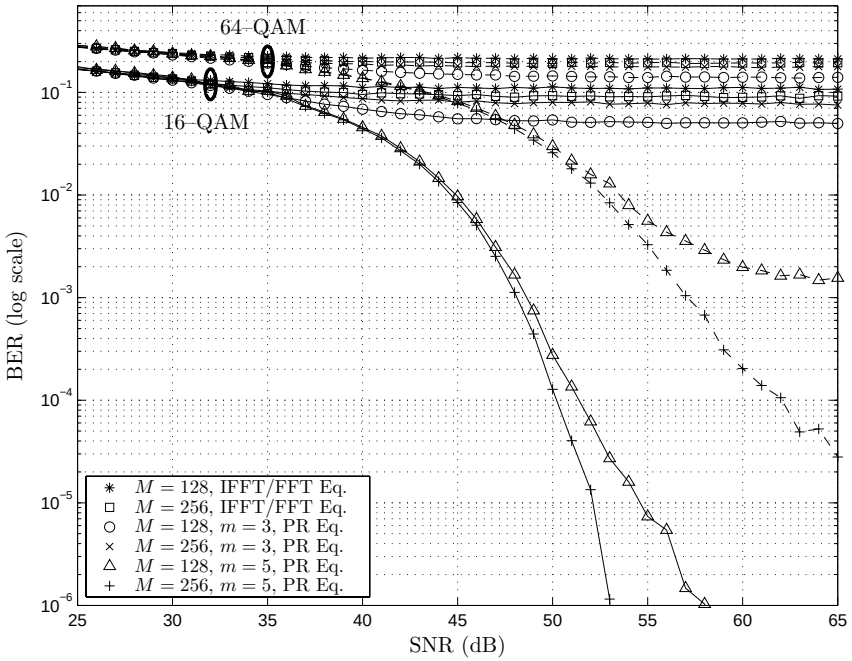


Figure 33: BER versus SNR for 16– and 64–QAM signals transmitted through the channel of Fig. 32 and equalized using the IFFT/FFT, and the $m = 3$ and $m = 5$ PR equalizers with $M = 128$ and $M = 256$.

can successfully equalize. Channels that are more complicated may require a larger power-of-two value of M. The parameters of the new and the IFFT/FFT blind equalizes are given in Table 6.

The BER versus the SNR for the new and the IFFT/FFT equalizers are shown in Fig. 36. For the PR equalizer with $m = 5$, both Gaussian and uniform noise distributions are considered. While the $m = 5$–PR equalizer fails to converge for several low SNR values, the IFFT/FFT equalizer, which also uses the suggested adaptation algorithm, fails to converge for several high SNR values and is unable to reduce the BER below approximately 0.03 when the SNR is increased. The floor of the BER plot for the IFFT/FFT equalizer results from the limited spread of the values of A_k to approximately 25 dB, which results in the loss of a significant portion of the received signal and the failure to adapt several times. The performance of the new equalizer in the presence of uniformly distributed noise is seen to be very close to the performance with Gaussian noise.

For a 16–QAM received signal with a SNR of 30 dB, Fig. 37(a) and (b) show the

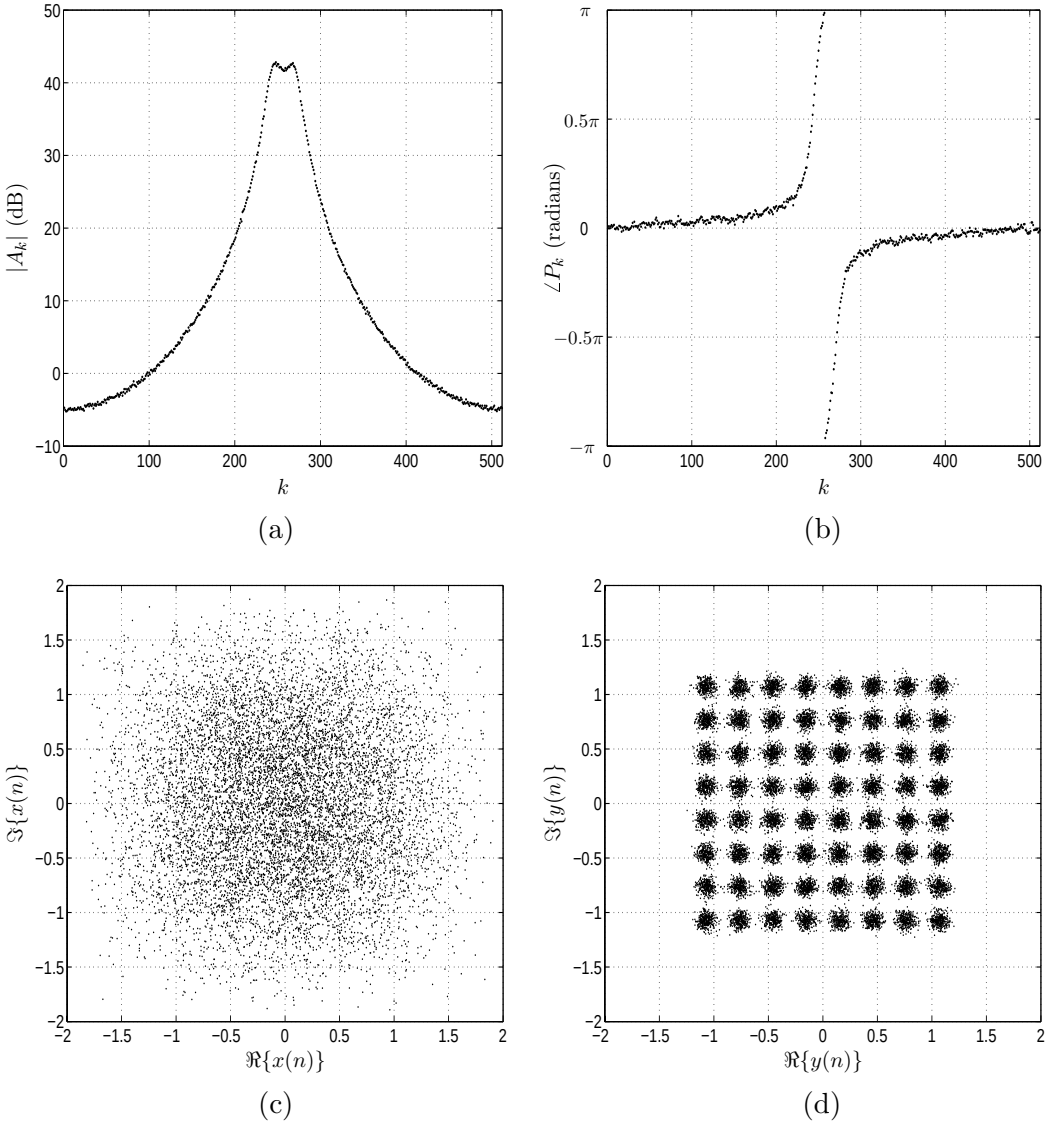


Figure 34: Performance of the new FD equalizer in Example 1 for a 64–QAM constellation at a SNR of 50 dB, (a) Magnitude of optimized A_k, (b) Angle of optimized P_k, (c) a set of samples of the input signal $x(n)$, and (d) a set of samples of the equalized output signal $y(n)$.

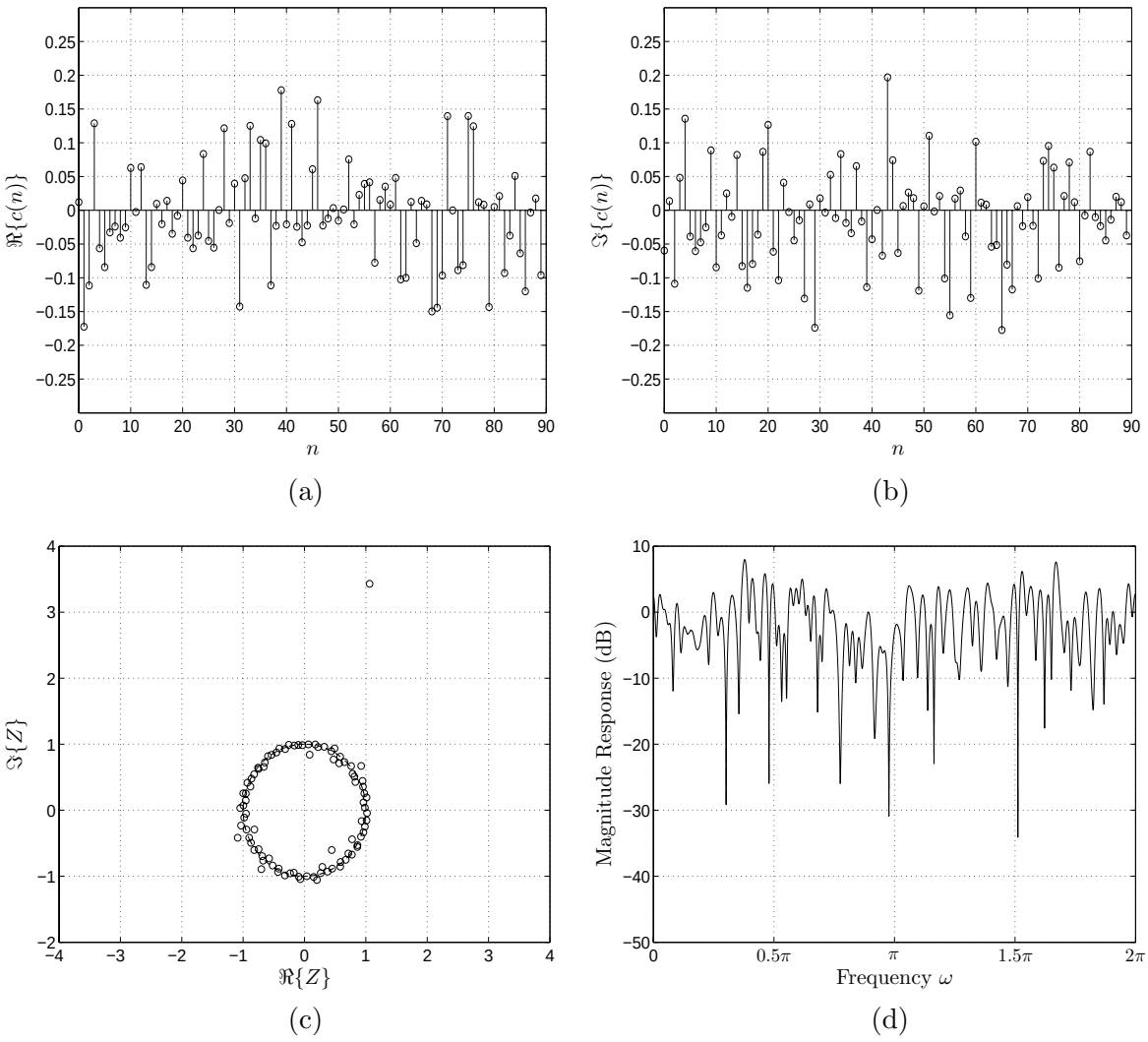


Figure 35: The channel used in Example 2, (a) $\Re\{c(n)\}$, (b) $\Im\{c(n)\}$, (c) zero-pole plot, and (d) magnitude response $|C(e^{j\omega})|$.

Table 6: Parameters of the new blind equalizer in Example 2

Parameter	Value
Modulation Type	16-QAM, iid symbols
Maximum Number of Iterations for Adaptation	4000
SNR	10–50 dB
M	512
m	1 (IFFT/FFT) and 5
Number of Weights $2M$	1024
ω_s of the Prototype Filter	$\pi/512$
A_s of Prototype Filter	13.2 dB (IFFT/FFT) 47.85 dB($m = 5$)
δ_A	0.01
λ_A	0.999
δ_P	0.01
λ_P	0.985
γ_0	As in Table 4
μ_R	0.0001
ϵ	0.00655

adapted $|A_k|$ and $\angle P_k$ of the $m = 5$–PR equalizer, respectively. The figure also shows part of the received and equalized signals. The adaptation of the different parts of the equalizer and the equalizer as a whole are illustrated in Fig. 38. The adaptation of the weights A_k and P_k are given in terms of the means of $|\log_{10}(|A_k|)|$ and $|\angle P_k|$ over the values of $k = 0, 1, \cdots, 2M-1$ for every iteration, respectively, the adaptation of R is given in terms of $\theta_R = \angle R$, and the adaptation of the whole equalizer is given in terms of the mean of the squared error (MSE) of the output signal $y(n)$. Fig. 38(a) shows that the adaptation of A_k continues throughout the process but slows in speed as the adaptation process progresses. The adaptation of the P_k illustrated in Fig. 38(b) roughly continues at a constant rate until the adaptation the is achieved, where it progresses slowly afterwards. The phase of R remains unadapted until the weights P_k have approached their final values, where the it quickly adjusts the rotation of the constellation to the proper value and nearly remains constant as shown in Fig. 38(c). The MSE of $y(n)$ shown in Fig. 38(d), which is an indicator of the adaptation of the whole equalizer, remains almost constant until all the

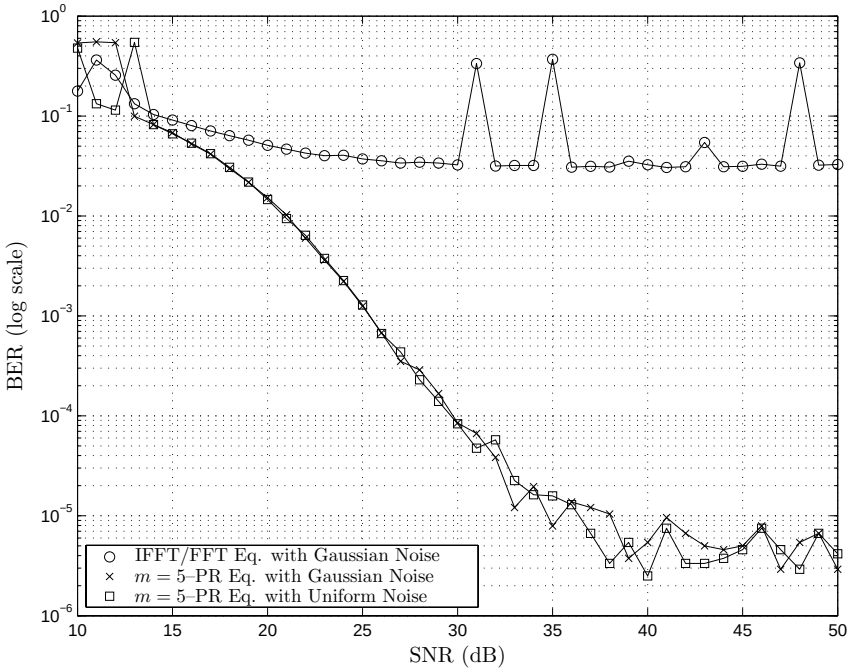


Figure 36: BER versus SNR for 16–QAM signals transmitted through the channel of Fig. 35 and equalized using the IFFT/FFT and the $m = 5$–PR equalizers with $M = 512$. For the $m = 5$–PR equalizer, noise that is Gaussian and unform distributed are considered.

equalizer weights have been adjusted successfully, where it then drops to its minimum.

Example 3

The new equalizer is tested for equalizing channels of different lengths using 7 sets of channels with lengths ranging from 20 to 80 coefficient. Each set contains 25 channels of a particular length with coefficients that are samples of a Gaussian random variable. An equalizer with $M = 512$ and $m = 5$ was used for equalizing the different channels, where the parameters of the equalizer were fixed to the values shown in Table 7. The plots of the BER averaged over the 25 channels of each set versus the SNR for the new equalizer and the IFFT/DFT equalizer are given in Fig. 39. For relatively long channels (i.e., $L = 60$, 70, and 80 in this example), Fig. 39 shows that the new equalizer experiences a BER floor due to the domination of the error caused by the remaining ISI in the equalized signal over the error due to the channel noise. The suggested adaptation algorithm successfully adapted the PR filter bank equalizer for almost all the channels over the SNRs shown in Fig. 39. For the IFFT/FFT equalizer, successful adaptation is achieved for almost all short

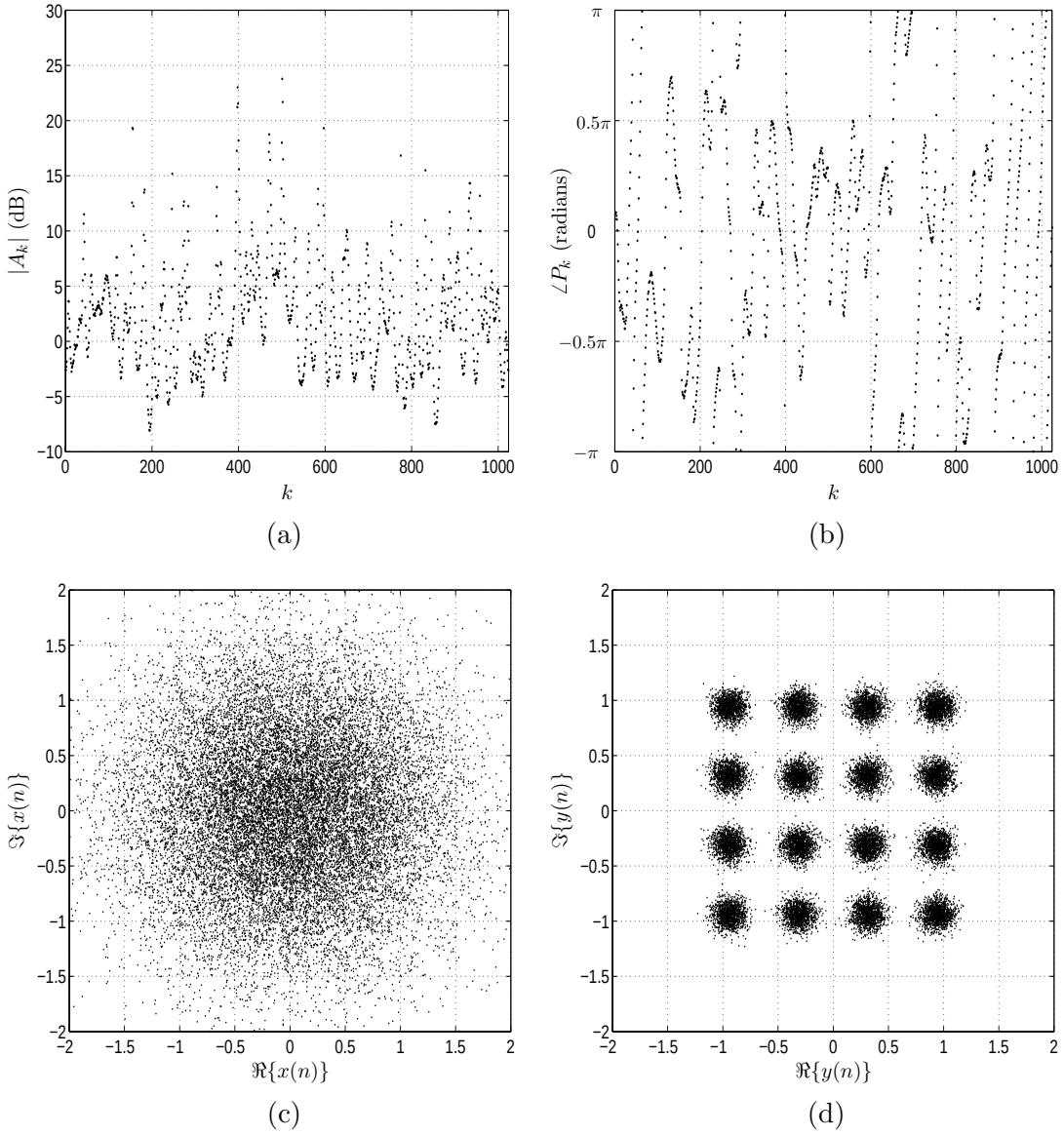


Figure 37: Performance of the new equalizer in Example 2 for a 16–QAM constellation at a SNR of 30 dB, (a) Magnitude of optimized A_k, (b) Angle of optimized P_k, (c) a set of samples of the input signal $x(n)$, and (d) a set of samples of the equalized output signal $y(n)$.

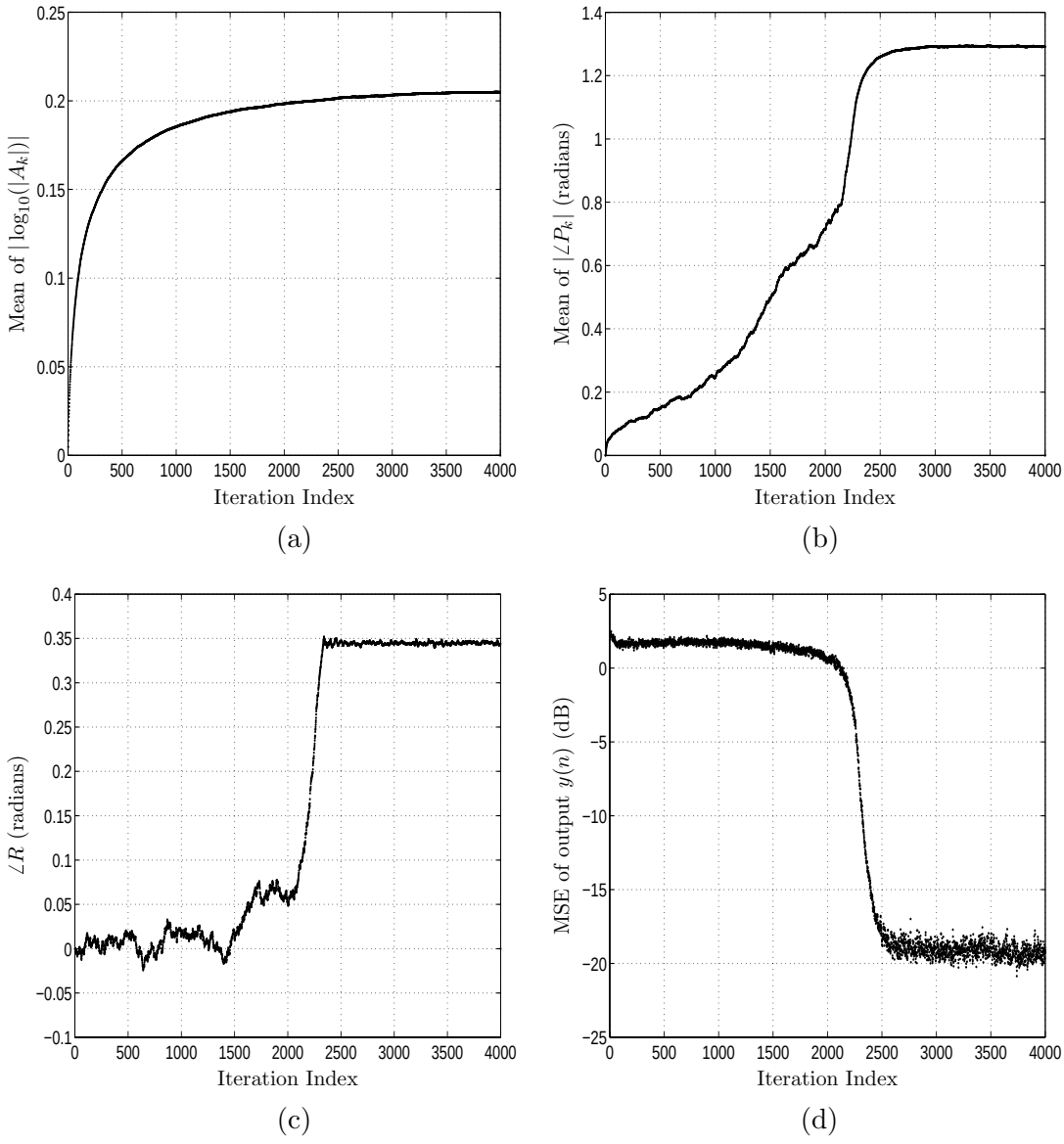


Figure 38: Adaptation of the blind equalizer weights, (a) Measure of adaptation of the magnitude of A_k, (b) Measure of adaptation of the phase of P_k, (c) Measure of adaptation of the phase of R, and (d) MSE of the output signal $y(n)$.

channels while adaptation fails for a large portion of long channels. The average BER even for the set of channels of length 20 floors for high SNRs at approximately 0.01, rendering the IFFT/FFT nearly useless for equalizing such channels.

Table 7: Parameters of the new blind equalizer in Example 3

Parameter	Value
Modulation Type	16-QAM, iid symbols
Maximum Number of Iterations for Adaptation	4000
SNR	$\leq 15 - 45$ dB
M	512
m	1 (IFFT/FFT) and 5
ω_s of the Prototype Filter	$\pi/512$
A_s of Prototype Filter	13.2 dB (IFFT/FFT) 47.85 dB($m = 5$)
δ_A	0.01
λ_A	0.999
δ_P	0.01
λ_P	0.985
γ_0	As in Table 4
μ_R	0.0001
ϵ	0.00655

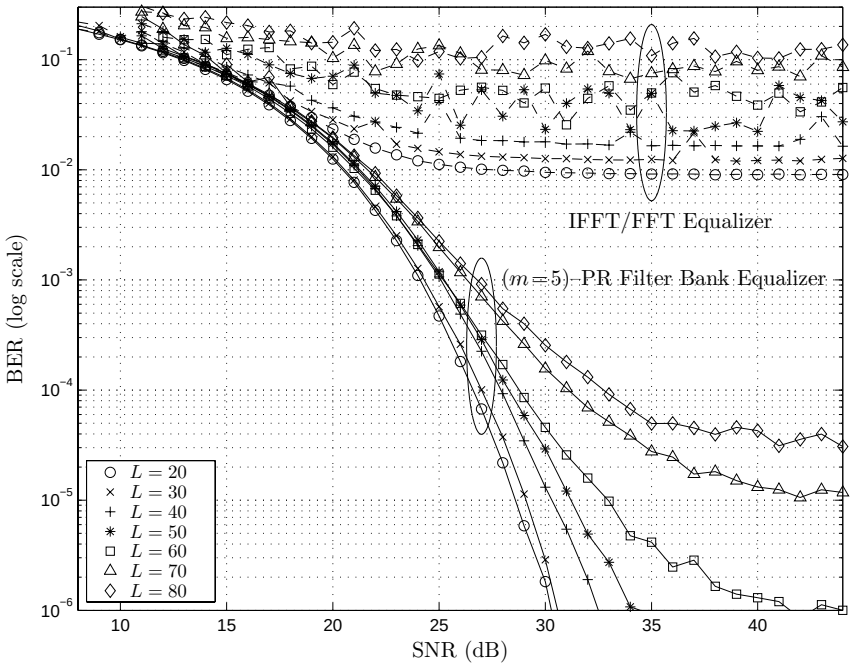


Figure 39: The average BER versus the SNR for channels with $L = 20, 30, \cdots, 80$ coefficients that are equalized using both the $(m = 5)$–PR filter bank and the IFFT/DFT equalizers.

CHAPTER VIII

CONCLUDING REMARKS

The objective of the research described in this thesis was to investigate new methods for performing some of the computationally demanding DSP functions in the front-end of SWR systems in an effort to reduce their computational requirements and, therefore, reduce the cost, DSP count, radio size, and power consumption of SWR base and mobile station transceivers. The major contributions of this research are

- **Modified CIC Filters:** A modification to conventional CIC filters with improved frequency characteristics was described,
- **Efficient SRC for SWR Base Station Transmitters:** A computationally efficient method for performing SRC by factors greater than unity for use in SWR systems was presented,
- **Efficient Channelizer/Synthesizer Design:** A computationally efficient channelizer and channel synthesizer for extracting and recombining baseband channels in SWR systems was described,
- **Design Method for Modulated PR Filter Banks:** A fast and powerful method for designing the prototype filters of modulated PR filter banks was discussed,
- **Design of Robust FD Blind Equalizers:** The design and adaptation method of a robust and reliable FD blind equalizer for equalizing SWR wideband signals ware presented.

These contributions and the conclusions derived from them are summarized as follows.

8.1 Modified CIC Filter for Sample Rate Conversion in Software Radio Systems

A modification to the conventional CIC class of filters for SRC in SWR systems was described in Chapter 3. This modification was shown to enhance the frequency characteristics of the conventional CIC filter. The modified CIC filter provides higher SNRs and better image attenuation compared to the conventional CIC filter by adjusting the zeros of the filter to target high-power image components. SWR systems can take advantage of this flexibility when the wideband input contains narrowband channels with a high dynamic range. An SWR receiver can measure the power of different channels and correspondingly adjust the delays of the CIC filter to minimize aliasing caused by high-power narrowband channels. The modified CIC filter gains this improved performance over the conventional CIC filter at the expense of a small increase in the number of computations for a range of SRC factors.

8.2 Efficient Sample Rate Conversion for Software Radio Systems Using Hierarchical Computation of Output Signals

An efficient SRC method for SRC factors $L/M > 1$ was discussed in Chapter 4. This SRC method is capable of reducing the number of MPOS that are required for performing SRC of wideband and high dynamic range signals for a range of conversion factors that is required at different stages of SWR communication systems in general, and in SWR base station transmitters in particular. The computational efficiency of this method is significantly higher than the efficiency of other SRC methods for conversion factors that are greater than but close to unity and the efficiency generally decreases as the conversion factor increases. This SRC method may require a fraction in the range 0.25–0.50 of the MPOS required by efficiently-constructed multi-stage conventional SRC systems, especially when the SRC factor is greater than but close to unity.

8.3 Efficient Wideband Channelizer for Software Radio Systems Using Modulated Perfect Reconstruction Filter Banks

In Chapter 5, we presented a new channelization method that uses modulated PR filter banks as building blocks. This channelizer significantly reduces the computations for channelization in SWR systems when the input signal is composed of channels with different bandwidths, where polyphase decomposition of the DFB channelizer is inapplicable. This channelizer is shown to have higher computational efficiency than DFB channelizers, especially when the number of extracted channels is large. For wideband signals with equal-bandwidth channels, the efficiency of the channelizer is also higher than PFB channelizers for when only few channels are extracted. The new channelizer can also be used for efficiently performing synthesis of channels with nonequal bandwidths and different sample rates. The computational complexity of the channelizer/synthesizer, which is of order $O(\log_2 M)$, is significantly lower than the computational complexity of DFB channelizers, which is of order $O(M)$, when channelizing signals with large number of channels that are separated by small guard bands.

8.4 Design of Modulated PR Prototype Filters with Large Values of Subchannels M

A new method for designing modulated PR filter banks was described in Chapter 6. While conventional methods for designing modulated exact-reconstruction filter banks fail to generate filters with both high number of subchannels and prototype filters with high stopband attenuations, the new method is capable of designing such modulated exact-PR filter banks efficiently and quickly. The efficiency of the design method was illustrated by designing a prototype filter for a 512-subchannel modulated PR filter bank with stopband attenuation $A_s = 98.04$ dB and stopband edge $\omega_s = \pi/512$. The complexity of the design method increases linearly with the number of subchannels of the filter bank while the complexity of the design methods in the literature increase at a much higher rate and generally fail to design modulated PR filter banks with stopband attenuations on the order of 100 dB for

filter banks with more than two subchannels.

8.5 Blind Equalization of QAM Signals Transmitted over Severe ISI-Causing Channels

Chapter 7 described a FD blind equalization method that is capable of equalizing long channels with severe ISI that conventional TD or FD blind equalizers have not been shown to be able to equalize. The blind equalizer, which is suitable for QAM signals and other two-dimensional symbol constellations, equalizes the transmission channel by adapting the individual magnitudes and the individual phases of its FD weights independently. Further, a rotation of the equalized signal is performed to compensate for the average error in the adaptation of the phase part of the equalizer weights. The computational requirements of the $2M$-weight equalizer are of order $O(log_2 2M)$ and the delay of the system is of order $O(2M)$. The computational requirements of the new equalizer represent a small fraction of the computational requirements of conventional TD blind equalizers, which have order $O(2M)$ when the number of weights in the equalizers is high. Simulations of the equalizer show that it is capable of equalizing complex-valued channels of lengths on the order of one-twelfth the number of weights with high reliability.

We propose the following extensions to the research performed in this thesis.

- **Use of Filter Banks for SRC:** The research in this thesis has shown the potential for modulated near- and exact-perfect reconstruction filter banks for implementing several signal processing algorithms such as channelizing/synthesizing narrow band channels and equalization. We propose extending the use of modulated filter banks for SRC. Modulated perfect reconstruction filter banks may have great potential for performing SRC due to their inherent multi-rate structure,

- **Combining Blocks of the SWR with Similar Construction:** One of the widely considered fields of research in SWR is dedicated to finding the commonalities between the different wireless communication standards to reduce the computations of the SWR

system. The channelization/channel-synthesis and blind equalization algorithms discussed in Chapters 5 and 7, respectively, and possibly the SRC systems if they can be implemented using modulated filter banks would all have the same basic construction. We think that significant amounts of computations can be eliminated by combing the different blocks of the SWR front-end,

- **Extending FD Blind Equalizers to Time-Varying Channels:** The FD blind equalization method described in Chapter 7 assumed a constant channel over the period of equalizer adaptation. We propose studying the effect of complicated channels that are slowly time-varying on the performance of this FD blind equalizer,

- **Study of Practical Implementation of the Algorithms on ASICs and FPGAs:** The research in this thesis assumes that the different algorithms are implemented on general-purpose DSPs. Implementing these algorithms on ASICs or FPGAs is accompanied by a new set of problems, including the limited reconfigurability of these units. We propose an extension to this research that studies the possibility of using ASICs and FPGAs for the DSP algorithms discussed here.

APPENDIX A

RELATION BETWEEN COSINE-MODULATED AND EXPONENTIALLY-MODULATED PR FILTER BANKS

Representing $\hat{H}_k(z)$ and $\hat{F}_k(z)$ in terms of $H(z)$ using (27) and substituting into the output signal of the cosine-modulated filter bank

$$\hat{X}(z) = \frac{1}{M} \sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{M-1} \hat{H}_k\left(zW_M^p\right) \hat{F}_k(z) \tag{101}$$

yields

$$\begin{aligned}
\hat{X}(z) \;=\; & \frac{1}{M} \sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{M-1} \Bigg[W_{2M}^{(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_M^p W_{4M}^{(2k+1)}\right) H\left(zW_{4M}^{(2k+1)}\right) \\
& + W_{2M}^{-(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_M^p W_{4M}^{-(2k+1)}\right) H\left(zW_{4M}^{-(2k+1)}\right) \\
& + j(-1)^k H\left(zW_M^p W_{4M}^{(2k+1)}\right) H\left(zW_{4M}^{-(2k+1)}\right) \\
& - j(-1)^k H\left(zW_M^p W_{4M}^{-(2k+1)}\right) H\left(zW_{4M}^{(2k+1)}\right) \Bigg].
\end{aligned} \tag{102}$$

We can define $S_p(z)$ as the sum of the last two terms in (102) over all values of $k = 0, 1, \cdots, M-1$, i.e.,

$$\begin{aligned}
S_p(z) \;=\; & \sum_{k=0}^{M-1} j(-1)^k H\Big(zW_M^p W_{4M}^{(2k+1)}\Big) H\Big(zW_{4M}^{-(2k+1)}\Big) \\
& - \sum_{k=0}^{M-1} j(-1)^k H\Big(zW_M^p W_{4M}^{-(2k+1)}\Big) H\Big(zW_{4M}^{(2k+1)}\Big).
\end{aligned} \tag{103}$$

Expanding the two summations of $S_p(z)$ in (103) gives

$$\begin{aligned}
S_p(z) \;=\; & j\Bigg[H\Big(zW_{4M}^{4p+1}\Big) H\Big(\hat{z}W_{4M}^{-1}\Big) - H\Big(zW_{4M}^{4p+3}\Big) H\Big(\hat{z}W_{4M}^{-3}\Big) + \cdots \\
& \qquad - H\Big(zW_{4M}^{4p+2M-1}\Big) H\Big(\hat{z}W_{4M}^{-2M+1}\Big) \Bigg] \\
& - j\Bigg[H\Big(zW_{4M}^{4p-1}\Big) H(\hat{z}W_{4M}) - H\Big(zW_{4M}^{4p-3}\Big) H\Big(\hat{z}W_{4M}^{3}\Big) + \cdots \\
& \qquad - H\Big(zW_{4M}^{4p-2M+1}\Big) H\Big(\hat{z}W_{4M}^{2M-1}\Big) \Bigg].
\end{aligned} \tag{104}$$

It can be easily shown that the first set of terms of $S_p(z)$ in (104) cancel the second set of terms for any $p = 0, 1, \cdots, M-1$. Hence, $S_p(z) = 0$ for all $p = 0, 1, \cdots, M-1$ and any z. Therefore, (102) becomes

$$\begin{aligned}
\hat{X}(z) &= \frac{1}{M}\sum_{p=0}^{M-1} X\left(zW_M^p\right)\left[\sum_{k=0}^{M-1} W_{2M}^{(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_M^p W_{4M}^{(2k+1)}\right) H\left(zW_{4M}^{(2k+1)}\right)\right. \\
&\quad \left. + \sum_{k=2M-1}^{M} W_{2M}^{(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_M^p W_{4M}^{(2k+1)}\right) H\left(zW_{4M}^{(2k+1)}\right)\right] \\
&= \frac{1}{M}\sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{2M-1} W_{2M}^{(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_M^p W_{4M}^{(2k+1)}\right) H\left(zW_{4M}^{(2k+1)}\right) \\
&= \frac{1}{M}\sum_{p=0}^{M-1} X\left(zW_M^p\right) \sum_{k=0}^{2M-1} \tilde{H}_k\left(zW_M^p\right) \tilde{F}_k\left(z\right), \qquad (105)
\end{aligned}$$

where

$$\tilde{H}_k(z) = \tilde{F}_k(z) = W_{4M}^{(2k+1)\left(\frac{2mM-1}{2}\right)} H\left(zW_{4M}^{(2k+1)}\right). \qquad (106)$$

Therefore, the modulation of (27) in a cosine-modulated filter bank and the modulation

$$\tilde{h}_k(n) = \tilde{f}_k(n) = h(n) e^{j\frac{\pi(2k+1)}{2M}\left(n-\frac{2mM-1}{2}\right)} \qquad (107)$$

in a CEM filter bank are identical. Since the cosine modulation of (27) is a form of the CEM in (107), prototypes designed for cosine-modulated PR filter banks are also applicable for CEM PR filter banks.

REFERENCES

[1] Abu-Al-Saud, W. A. and Stüber, G. L., "Modified CIC filter for sample rate conversion in software radio systems," *IEEE Signal Proc. Letters*, vol. 10, pp. 152–154, May 2003.

[2] Abu-Al-Saud, W. A. and Stüber, G. L., "Blind frequency domain equalization of severe ISI channels with long responses," *submitted to IEEE Trans. Commun.*

[3] Abu-Al-Saud, W. A. and Stüber, G. L., "Efficient sample rate conversion for software radio systems," *submitted to IEEE Trans. Signal Proc.*

[4] Abu-Al-Saud, W. A. and Stüber, G. L., "Efficient wideband channelizer for software radio systems using modulated perfect reconstruction filter banks," *to appear in IEEE Trans. Signal Proc.*

[5] Albert, A. E. and Gardner, Jr., L. S., *Stochastic Approxmiation and Nonlinear Regression*. Cambridge, MA: MIT Press, 1976.

[6] Baines, R., "The DSP bottleneck," *IEEE Commun. Mag.*, vol. 33, pp. 46–54, May 1995.

[7] Begovic, R. and Saramaki, T., "A systematic technique for designing prototype filters for perfect reconstruction cosine modulated and modified DFT filter banks," in *ISCAS*, pp. 33–36, 2001.

[8] Bellini, S., "Bussgang techniques for blind equalization," in *GLOBECOM*, pp. 1634–1640, 1986.

[9] Buracchini, E., "The software radio concept," *IEEE Commun. Mag.*, vol. 38, pp. 138–143, Sep. 2000.

[10] Butler, M., Providakes, J., and Blythe, G., "The layered radio," in *MILCOM*, pp. 174–179, 1998.

[11] Cadzow, J. A. and Li, X., "Blind deconvolution," *Dig. Signal Processing J.*, vol. 5, pp. 3–20, Jan. 1995.

[12] Chandran, N. and Valenti, M. C., "Three generations of cellular wireless systems," *IEEE Potentials*, vol. 20, pp. 32–35, Feb./Mar. 2001.

[13] Chi, C.-Y., Chen, C.-Y., Chen, C.-H., and Feng, C.-C., "Batch processing algorithms for blind equalization using higher-order statistics," *IEEE Signal Processing Mag.*, vol. 20, pp. 25–49, Jan. 2003.

[14] Crochiere, R. E. and Rabiner, L. R., "Interpolation and decimation of digital signals - a tutorial review," *Proc. IEEE*, vol. 69, pp. 300–331, March 1981.

[15] Crochiere, R. E. and Rabiner, L. R., *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[16] Cruz-Roldán, F., Amo-López, P., Maldonado-Bascón, S., and Lawson, S. S., "An efficient and simple method for designing prototype filters for cosine-modulated pseudo-QMF banks," *IEEE Signal Processing Lett.*, vol. 9, pp. 29–31, 2002 Jan.

[17] Cruz-Roldán, F., López-Ferreras, F., Amo-López, P., and Osés-del Campo, J. D., "Arbitrary-length spectral factor applied to the design of pseudo-QMF cosine-modulated filter banks," *IEEE Trans. Circuits Syst. II*, vol. 48, pp. 321–325, March 2001.

[18] Cummings, M. and Haruyama, S., "FPGA in the software radio," *IEEE Commun. Mag.*, vol. 37, pp. 108–112, Feb. 1999.

[19] Cummings, M. and Heath, S., "Mode switching and software download for software defined radio: the SDR forum approach," *IEEE Commun. Mag.*, vol. 37, pp. 104–106, Aug. 1999.

[20] Dick, C., Harris, F., and Rice, M., "Synchronization in software radios. Carrier and timing recovery using FPGAs," in *Symposium Field-Programmable Custom Computing Machines*, pp. 195–204, 2000.

[21] Dick, C. and Harris, F., "Configurable logic for digital communications: some signal processing perspectives," *IEEE Commun. Mag.*, vol. 37, pp. 107–111, Aug. 1999.

[22] Domanin, D., Gatti, U., Malcovati, P., and Maloberti, F., "A multipath polyphase digital-to-analog converter for software radio transmission systems," in *ISCAS*, (Geneva), pp. 361–364, 2000.

[23] Douglas, S. C. and Meng, T. H.-Y., "Normalized data nonlinearities for LMS adaptation," *IEEE Trans. Signal Processing*, vol. 42, pp. 1352–1365, June 1994.

[24] Efstathiou, D., Fridman, J., and Zvonar, Z., "Recent developments in enabling technologies for software defined radio," *IEEE Commun. Mag.*, vol. 37, pp. 112–117, Aug. 1999.

[25] European Telecommunications Standards Institute (ETSI), "Radio tansmission and reception." *GSM 5.05*, March 1986.

[26] Fung, C. Y. and Chan, S. C., "A multistage filterbank-based channelizer and its multiplier-less realization," in *IEEE Int. Symp. Circuits Syst.*, pp. 429–432, 2002.

[27] Giannakis, G. B. and Mendel, J. M., "Identification of nonminimum phase system using higher order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 360–377, March 1989.

[28] Godard, D. N. and Thirion, P. E., "Method and device for training an adaptive equalizer by means of an unknown data siganl in a QAM transmission system." *U.S. Patent 4 227 152*, Oct. 7, 1980.

[29] Godard, D. N., "Self-recovering equalization and carrier tracking in two dimensional data communication systems," *IEEE Trans. Commun.*, vol. 28, pp. 1867–1875, Nov. 1980.

[30] Gunn, J. E., Barron, K. S., and Ruczczyk, W., "A low-power DSP core-based software radio architecture," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 574–590, April 1999.

[31] Harteneck, M., Weiss, S., and Stewart, R. W., "Design of near perfect reconstruction oversampled filter banks for subband adaptive filters," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1081–1085, Aug. 1999.

[32] Hatzinakos, D. and Nikias, C. L., "Estimation of multipath channel response in frequency selective channels," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 12–19, Jan. 1989.

[33] Hatzinakos, D. and Nikias, C. L., "Blind equalization using a tricepstrum-based algorithm," *IEEE Trans. Commun.*, vol. 39, pp. 669–682, May 1991.

[34] Hatzinakos, D. and Nikias, C. L., "Blind equalization using a tricepstrum-based algorithm," *IEEE Trans. Commun.*, vol. 39, pp. 669–682, May 1991.

[35] Haykin, S., *Adaptive Filter Theory*. Upper Saddle River, NJ: Printice Hall, 2001.

[36] Hentschel, T., Fettweis, G. P., and Bronzel, M., "Channelization and sample rate adaptation in software radio terminals," in *Third ACTS Mobile Communication Summit*, pp. 121–126, 1998.

[37] Hentschel, T. and Fettweis, G., "Sample rate conversion for software radio," *IEEE Commun. Mag.*, vol. 38, pp. 142–150, Aug. 2000.

[38] Hentschel, T. and Fettweis, G., "Continuous-time digital filters for sample-rate conversion in reconfigurable radio terminals," *Frequenz: Journal of Telecommunications (Germany)*, vol. 55, pp. 188–194, May/June 2001.

[39] Hentschel, T., Henker, M., and Fettweis, G., "The digital front-end of software radio terminals," *IEEE Pers. Commun.*, vol. 6, pp. 40–46, Aug. 1999.

[40] Hogenauer, E. B., "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-29, pp. 155–162, April 1981.

[41] Hong, N. and Mathiopoulos, P. T., "A new software radio based distributed base station architecture and its application to 3G UMTS employing signal combining techniques," in *VTC*, pp. 240–244, Spring 2002.

[42] Huang, X. H., Du, K. L., Lai, A. K., and Cheng, K. K., "A unified software radio architecture," in *IEEE Third Workshop on Signal Processing Advances in Wireless Communications*, pp. 330–333, 2001.

[43] Jain, V. K., "Very high SFDR interpolation filters for software radio," in *IEEE Workshop Signal Proc. Syst.*, pp. 397–406, 2000.

[44] Johnson, C. R., Schniter, P., Endres, T. J., Behm, J. D., Brown, D. R., and Casas, R. A., "Blind equalization using the constant modulus criterion: a review," *Proc. IEEE*, vol. 86, pp. 1927–1950, Oct. 1998.

[45] Jondral, F., Wiesler, A., and Machauer, R., "A software defined radio structure for 2nd and 3rd generation mobile communications standards," in *IEEE 6th International Symposium on Spread Spectrum Techniques and Applications*, pp. 637–640, 2000.

[46] Kaiser, J. F., "Nonrecursive digital filter design using the I_0-sinh window function," in *Proc. IEEE Int. Symp. Circuits Sys.*, pp. 20–23, April 1974.

[47] Karimi, H. R. and Friedrichs, B., "Wideband digital receivers for multi-standard software radios," in *IEE Colloquium on Adaptable and Multistandard Mobile Radio Terminals*, pp. 5/1–5/7, 1998.

[48] Karp, T. and Fliege, N. J., "Modified DFT filter banks with perfect reconstruction," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1404–1414, Nov. 1999.

[49] Kennedy, R. A., Pulford, G., Anderson, B. D., and Bitmead, R. R., "When has a decision-directoed equalizer converged?," *IEEE Trans. Commun.*, vol. 37, pp. 879–884, Aug. 1989.

[50] Kiessling, M. and Mujtaba, S. A., "A software radio architecture for multi-channel digital upconversion and downconversion using generalized polyphase filterbanks with frequency offset correction," in *Personal, Indoor and Mobile Radio Communications*, pp. 105–109, 2002.

[51] Koilpillai, R. D. and Vaidyanathan, P. P., "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, April 1992.

[52] Lackey, R. J. and Upmal, D. W., "Speakeasy: The military software radio," *IEEE Commun. Mag.*, vol. 33, pp. 56–61, May 1995.

[53] Laddomada, M., Presti, L. L., Mondin, M., and Ricchiuto, C., "An efficient decimation sinc-filter design for software radio applications," in *IEEE Third Workshop on Signal Processing Advances in Wireless Communications*, pp. 337–339, 2001.

[54] Lee, E. A. and Messerschmitt, D. G., *Digital Communications*. Norwell, MA: Kluwer Academic Publishers, 1994.

[55] Litwin, L. R., "Blind channel equalization," *IEEE Potentials*, vol. 18, pp. 9–12, Oct.-Nov. 1999.

[56] Macchi, O. and Eweda, E., "Convergence analysis of self-adaptive equalizers," *IEEE Trans. Inform. Technol.*, vol. 30, pp. 161–176, March 1984.

[57] Malvar, H. S., "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 969–978, June 1990.

[58] Malvar, H. S., "Modulated QMF filter banks with perfect reconstruction," *Electron. Lett.*, vol. 26, pp. 906–907, June 1990.

[59] Margulies, A. S. and Mitola, III, J., "Software defined radios: a technical challenge and a migration strategy," in *IEEE 5th International Symposium on Spread Spectrum Techniques and Applications*, pp. 551–556, 1998.

[60] Mitola, III, J., "Software radios survey, critical evaluation and future directions," in *National Telesystems Conf.*, pp. 13/15–13/23, May 1992.

[61] Mitola, III, J., "Software radio architecture: a mathematical perspective," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 514–538, April 1999.

[62] Mitola, J., "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, pp. 26–38, May 1995.

[63] Mitola, J., "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, pp. 26–38, May 1995.

[64] Mitola, J., "Technical challenges in the globalization of software radio," *IEEE Commun. Mag.*, vol. 37, pp. 84–89, Feb. 1999.

[65] Moessner, K. and Tafazolli, R., "Terminal reconfigureability-the software download aspect," in *First Int. Conf. 3G Mobile Communication Technologies*, pp. 326–330, 2000.

[66] Murotake, D., Oates, J., and Fuchs, A., "Real-time implementation of a reconfigurable IMT-2000 base station channel modem," *IEEE Commun. Mag.*, vol. 38, pp. 148–152, Feb. 2000.

[67] Nagumo, J. I. and Noda, A., "A learning method for system identification," *IEEE Trans. Autom. Control.*, vol. 12, pp. 282–287, June 1967.

[68] Nayebi, K., Barnwell, III, T. P., and Smith, M. J. T., "The design of perfect reconstruction nonuniform band filter banks," in *Int. Conf. Acoust., Speech, Signal Processing*, pp. 1781–1784, 1991.

[69] Nguyen, T. Q. and Koilpillai, R. D., "The theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 44, pp. 473–483, March 1996.

[70] Nguyen, T. Q. and Vaidyanathan, P. P., "Maximally decimated perfect-reconstruction FIR filter banks with pairwise mirror-image analysis (and synthesis) frequency responses," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 693–706, May 1988.

[71] Nguyen, T. Q. and Vaidyanathan, P. P., "Structures for M-channel perfect-reconstruction FIR QMF banks which yield linear-phase analysis filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 433–446, March 1990.

[72] Nguyen, T. Q., "Near-perfect-reconstruction pseudo-QMF banks," *IEEE Trans. Signal Processing*, vol. 42, pp. 65–76, Jan. 1994.

[73] Oppenheim, A. V. and Schafer, R. W., *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[74] Pan, R. and Nikias, C. L., "The complex cepstrum of higher order cumulants and nonminimum phase system identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 186–205, Feb. 1988.

[75] Parks, T. W. and McClellan, J. H., "A program for the design of linear phase finite impulse response filters," *IEEE Trans. Audio Electoacoustics*, vol. AU-20, pp. 195–199, Aug. 1972.

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

[77] Posti, H., Järvelä, R., and Leppänen, P. A., "Receiver dimensioning in a hybrid multicarrier GSM base station," *IEEE Personal Commun.*, vol. 6, pp. 56–64, Aug. 1999.

[78] Princen, J. P. and Bradley, A. B., "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1153–1161, Oct. 1986.

[79] Proakis, J. G., *Digital Communications*. New York, NY: McGraw-Hill, 2000.

[80] Qureshi, S. U., "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387, Sep. 1985.

[81] Radiocommunicaiton Study Groups, "Detailed specifications of the radio interfaces of IMT2000." *ITU International Telecommunication Union*, 1999.

[82] Rafaely, B. and Elliott, S. J., "A computationally efficient frequency-domain LMS algorithm with contraints on the adaptive filter," *IEEE Trans. Signal Processing*, vol. 48, pp. 1649–1655, June 2000.

[83] Ramstad, T. A., "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-32, pp. 577–591, June 1984.

[84] Riera-Palou, F., Noras, J. M., and Cruickshank, D. G. M., "Variable length equalizers for broadband mobile systems," in *VTC*, pp. 2478–2485, 2000.

[85] Salkintzis, A. K., Nie, H., and Mathiopoulos, P. T., "ADC and DSP challenges in the development of software radio base stations," *IEEE Pers. Commun.*, vol. 6, pp. 47–55, Aug. 1999.

[86] Sato, Y., "A method of self-recovering equalization for multilevel amplitude-modulation systems," *IEEE Trans. Commun.*, vol. 23, pp. 679–682, June 1975.

[87] Schuller, G. D. and Smith, M. J. T., "New framework for modulated perfect reconstruction filter banks," *IEEE Trans Signal Processing*, vol. 44, pp. 1941 –1954, Aug. 1996.

[88] Shepherd, R., "Engineering the embedded software radio," *IEEE Commun. Mag.*, vol. 37, pp. 70–74, Nov. 1999.

[89] Shynk, J. J., Chan, C. K., and Petraglia, M. R., "Blind adaptive filtering in the frequency domain," in *IEEE International Symp. Circuits Systems*, pp. 275–278, 1990.

[90] Shynk, J. J., "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Mag.*, vol. 9, pp. 14–37, Jan. 1992.

[91] Software Defined Radio Forum. *http://www.sdrforum.org*.

[92] Srikanteswara, S., Hosemann, M., Reed, J. H., and Athanas, P. M., "Design and implementation of a completely reconfigurable soft radio," in *RAWCON*, pp. 7–11, 2000.

[93] Srikanteswara, S., Reed, J. H., Athanas, P., and Boyle, R., "A soft radio architecture for reconfigurable platforms," *IEEE Commun. Mag.*, vol. 38, pp. 140–147, Feb. 2000.

[94] Swindlehurst, A. L., "Normalized adaptive decision directed equalization," *IEEE Signal Proc. Letters*, vol. 5, pp. 18–20, Jan. 1998.

[95] Texas Instruments website. *http://www.ti.com*.

[96] TIA/EIA/IS-95, "Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular system," July 1993.

[97] TIA/EIA-cdma2000, "Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular system," July 1999.

[98] Torre, F. M., "Speakeasy-a new direction in tactical communications for the 21st century," in *Proc. Tactical Commun. Conf.*, pp. 139–142, April 1992.

[99] Treichler, J. R. and Agee, B. G., "A new approach to multipath correction of constant modulus siganls," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 349–472, April 1983.

[100] Tseng, C.-H. and Lin, C.-B., "A frequency-domain approximate RLS algorithm for blind equalization of mobile communication channels," in *PIMRC*, pp. 873–877, 1996.

[101] Turletti, T. and Tennenhouse, D., "Complexity of a software GSM base station," *IEEE Commun. Mag.*, vol. 37, pp. 113–117, Feb. 1999.

[102] Tuttlebee, W. H., "Software radio technology: a European perspective," *IEEE Commun. Mag.*, vol. 37, pp. 118–123, Feb. 1999.

[103] Tuttlebee, W., "Software radio-impacts and implications," in *IEEE Fifth International Symposium on Spread Spectrum Techniques and Applications*, pp. 541–545, 1998.

[104] Vaidyanathan, P. P., Nguyen, T. Q., Doganata, Z., and Saramaki, T., "Improved technique for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1042–1056, July 1989.

[105] Vaidyanathan, P. P. and Swaminathan, K., "Alias-free, real coefficient m-band QMF banks for arbitrary m," *IEEE Trans. Circuits Syst.*, vol. 34, pp. 1485–1496, Dec. 1987.

[106] Vaidyanathan, P. P., "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect-reconstruction property," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 476–492, April 1987.

[107] Vaidyanathan, P. P., *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[108] Vetterli, M. and Gall, D. L., "Perfect reconstruction FIR filter banks: some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1057–1071, July 1989.

[109] Viholainen, A., Stitz, T. H., Alhava, J., Ihalainen, T., and Renfors, M., "Complex modulated critically sampled filter banks based on cosine and sine modulation," in *ISCAS*, pp. 833–836, 2002.

[110] Wasserman, L. and Willson, Jr., A. N., "A variable-rate filtering system for digital communications," in *ICASSP*, pp. 1497–1500, 1999.

[111] Weidong, L. and Yan, Y., "Software radio: technology & implementation," in *International Conf. Commun. Technology*, (Beijing, China), pp. S27/01/1–S27/01/5, Oct. 1998.

[112] Wepman, J. A., "Analog-to-digital converters and their applications in radio receivers," *IEEE Commun. Mag.*, vol. 33, pp. 39–45, May 1995.

[113] Wiesler, A. and Jondral, F. K., "A software radio for second- and third-generation mobile systems," *IEEE Trans. Vehicular Technology*, vol. 51, pp. 738–748, July 2002.

[114] Wolmarans, E. M. and Truter, A. J., "Software radio: implementation aspects," in *EUROCOMM*, pp. 38–42, 2000.

[115] Xu, H., Lu, W., and Antoniou, A., "Efficient iterative design method for cosine-modulated QMF banks," *IEEE Trans. Signal Processing*, vol. 44, pp. 1657–1668, July 1996.

[116] Xu, J. and Wang, Y., "New decision-directed equalization algorithm for QAM communication systems," in *GLOBECOM*, pp. 1330–1334, 1996.

[117] Yung, W. H., Jian, M., and Ho, Y. W., "Polyphase decomposition channelizers for software radios," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 353–356, 2000.

[118] Zahirniak, D. R., Sharpin, D. L., and Fields, T. W., "A hardware-efficient, multirate, digital channelized receiver architecture," *IEEE Trans. Airosp. Electron. Syst.*, vol. 34, pp. 137–152, Jan. 1998.

[119] Zangi, K. C. and Koilpillai, R. D., "Efficient filterbank channelizers for software radio receivers," in *IEEE Int. Conf. Commun.*, pp. 1566–1570, 1998.

[120] Zangi, K. C. and Koilpillai, R. D., "Software radio issues in cellular base stations," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 561–573, April 1999.

[121] Zhang, Z. and Jiao, L., "A simple method for designing pseudo QMF banks," in *ICCT*, pp. 1538–1541, 2000.

VITA

Wajih A. Abu-Al-Saud was born in Saudi Arabia in 1971. He received the B.Sc. degree (with highest honors) and the M.Sc. degree, both in Electrical Engineering, from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1994 and 1996 respectively. From 1994 to 1996, he was a teaching assistant, and from 1996 to 1998 he was a lecturer in the Electrical Engineering department at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. During the summer of 1994, he was an intern in the Telecommunications Department of Saudi Aramco, Dhahran, Saudi Arabia. Since 1998, he has been working towards the Ph.D. degree in Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA.

His research interests include software radio systems, signal processing for telecommunication systems, blind equalization, and multi-rate systems and filter banks.