

Multicarrier OFDM with Application to Adaptive Spectrum Utilization: A Design Study

A Thesis

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

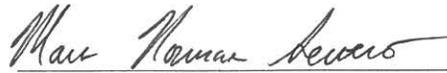
Marc N. Severo

December

2016

APPROVAL SHEET

The thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science


AUTHOR

The thesis has been read and approved by the examining committee:

Prof. Stephen G. Wilson

Advisor

Prof. Maite Brandt-Pearce

Prof. Robert M. Weikle

Dr. CJ Rieser

Accepted for the School of Engineering and Applied Science:



Craig H. Benson, Dean, School of Engineering and Applied Science

December
2016

This thesis is dedicated to my uncle,

Norman C. Severo,

His work and dedication to academic research is my inspiration

Abstract

As the demand for wireless spectrum exceeds regulatory allocation, there is a need to look at new ways to manage and utilize unused spectrum. One method is to use smart, or cognitive radios that are able to sense and utilize unused spectrum. As the bandwidth requirements for these adaptive radios increase, there is a need to aggregate the data over non-contiguous channels over a wide bandwidth. This requirement of both being adaptive and flexible proposes challenges to the design of the transmitter and receiver. These challenges are increased when the radios are used in a multi-user scenario where users are asynchronous to one another.

In this thesis we will propose solutions relating to the physical layer requirements of a wide-band Adaptive Spectrum Aggregation & Management (ASAM) system for asynchronous users. Specifically an application using multiple-user OFDM between possibly supersonic aircraft and a ground terminal will be studied. The challenges of timing estimation, frequency offset correction, and channel equalization will be evaluated and solutions explored. The performance of various channelizing transmitting and receiving architectures will be evaluated and compared. Often the performance of a channelizer is limited to its prototype FIR filter, so optimization of efficient root-Nyquist and compensated FIR filters will be presented. At the conclusion, an end-to-end simulation will be presented to measure the performance of the multiple-user OFDM system.

Acknowledgements

I would like to first thank my advisor, Prof. Stephen G. Wilson, who showed through example the importance and joy of teaching others.

I would also like to thank my committee members, Prof. Maite Brandt-Pearce, Prof. Bobby Weikle, and Dr. CJ Rieser for devoting their time to provide contributions to this work.

Thank you to everyone at the University of Virginia who has taught or supervised me, and let us also not forget Beth Eastwood-Beatty who kept this program running smoothly.

Last but not least, I would like to thank my wife, Kelly Severo, parents Armando and Roberta Severo, brother Anthony Severo, and my uncle, Norman Severo, who provided me with the daily support, encouragement, and substances needed to complete my studies.

Contents

Abstract	iv
Acknowledgements	v
List of Figures	ix
List of Tables	xii
Glossary of Terms	xiii
1 Introduction	1
1.1 Application	3
1.2 Orthogonal Frequency Division Multiplexing (OFDM) Modulation	6
1.2.1 OFDM System Overview	9
1.2.2 OFDM Channel Modulation	11
1.2.3 OFDM Channel Demodulation	12
1.3 Application System Parameters	13
1.4 Organization of Thesis	14
2 Channelizer Architectures	16
2.0.1 Polyphase Filter	18
2.0.2 Polyphase Channelizer	20
2.1 Receiver Architectures	22

2.1.1	Receiver I:	
	Polyphase Channelizer with External Resampler	23
2.1.2	Receiver II:	
	Polyphase Channelizer with Internal Rate Conversion	24
2.1.3	Comparison of Receivers	27
2.2	Proposed Transmitter Architectures	28
2.2.1	Transmitter I:	
	CIC Interpolator / Heterodyne	29
2.2.2	Transmitter II:	
	Polyphase Channelizer using Internal Rate Conversion	32
2.2.3	Comparison of Transmitters	34
3	Frame, Frequency, and Equalizer Estimation	36
3.1	Frequency Estimation	39
3.2	Timing Estimation	40
3.3	Channel Equalization	42
	3.3.1 Polynomial Curve Fitting	46
4	Prototype Filter Design	49
4.1	Truncated Root Raised Cosine Filter	51
4.2	Root-Nyquist Filter Design	53
	4.2.1 Symmetric FIR Filter Design	53
	4.2.2 Symmetric FIR Filter Design Minimizing Out of Band Power	55
	4.2.3 Asymmetric FIR Filter Design Minimizing Out of Band Power	58
4.3	CIC Compensation Filter	60
	4.3.1 Symmetric Compensated FIR Filter	61
	4.3.2 Symmetric Compensated CIC Filter Minimizing Out of Band Power	62
	4.3.3 Asymmetric Compensation FIR Filter Minimizing Out of Band Power	64
4.4	Filters Used in Simulation	65

5	Simulation and Results	68
5.1	Ideal Channel Simulations	70
5.2	Non-Ideal Channel Simulations	73
5.3	Multipath Channels	75
6	Conclusion	78
6.1	Future Work	79
7	Bibliography	81

List of Figures

1.1	Transmitter Design	2
1.2	Receiver Design	3
1.3	Major ASAM System Components	4
1.4	Frequency Plan	5
1.5	OFDM Subcarriers	6
1.6	OFDMA Resource Grid	8
1.7	OFDM Block Diagram	9
1.8	OFDM Symbols	10
1.9	OFDM Modulation Block Diagram	11
1.10	OFDM Demodulation Block Diagram	12
1.11	OFDM Subcarriers	14
2.1	Simple Receiver Channelizer with Conventional Heterodyning	17
2.2	Noble Identities	18
2.3	Polyphase Filter Structure	19
2.4	Rate Changing with Polyphase Filters	19
2.5	Resampling Downconverter, or Polyphase Channelizing Receiver	21
2.6	Spectrum and Channel Spacing	22
2.7	Receiver I: Polyphase Channelizer with External Resampler	23
2.8	Standard Fractional Resampler	24
2.9	Overall View of Receiver II	25
2.10	Input Circular Buffer States	26

2.11	Output Circular Buffer States	27
2.12	Comparison of Receiver Computational Costs per Number of Active Channel	28
2.13	CIC Interpolator Filter	29
2.14	CIC Interpolation Frequency Response for P Stages	30
2.15	Simple Transmitter Channelizer for Relatively Few Channels	31
2.16	Output Frequency Response for CIC, $L = 4, P = 3$	31
2.17	Transmitter II: Polyphase Structure with Internal Rate Conversion	32
2.18	Circular Buffer States for Transmitter II	33
2.19	Comparison of Transmitter Computational Costs per Number of Active Channel	35
3.1	OFDM Symbol Synchronization Using Correlator	37
3.2	OFDM Symbol and Frequency Synchronization Using Correlator	38
3.3	Outputs of Time and Frequency Estimator, Ideal Channel	39
3.4	Outputs of Time and Frequency Estimator, Non-Ideal Channel	41
3.5	OFDM Demodulation Timing Diagram	42
3.6	Output of Channelizer with Estimate $\hat{\theta}$ and OFDM Symbol Markings, Ideal Timing	43
3.7	Downsampled Output of Frame Parser, Ideal Timing	43
3.8	Output of Channelizer Aligned with Estimate $\hat{\theta}$, Non-Ideal Timing with 80- Sample Commutation Offset	44
3.9	Downsampled Output of Frame Parser, Non-Ideal Timing with 80-Sample Commutator Offset, Truncated Pulse, and Frequency Response	44
3.10	Frame Parser Output with 80-Sample Commutator Offset Parsing in the Mid- dle of the OFDM Symbol	45
3.11	Frame Parser Output with 80-Sample Commutator Offset Parsing in the Mid- dle of the Cyclic Prefix, and Samples Shifted	46
4.1	Frequency Response of Ideal RRC Filter	50
4.2	Truncated Root-Raised Cosine Filter	52

4.3	Symmetric Filter using Linear Program	56
4.4	Targeted Symmetric Filter using Nonlinear Program	58
4.5	Targeted Asymmetric Filter	60
4.6	Distortion Caused by CIC Interpolation	61
4.7	Compensated CIC Filter Using Linear Program	63
4.8	Targeted Symmetric Filter using Nonlinear Program	65
4.9	Targeted Asymmetric Compensated CIC Filter	66
4.10	Quantized Long Filter Used in Simulations	67
5.1	Simulation Under an Ideal AWGN Channel, Perfect Timing, Estimation and Equalization Disabled	70
5.2	Simulation Under an Ideal AWGN Channel, Perfect Timing Equalization En- abled, Estimation Disabled	71
5.3	Simulation Under an Ideal AWGN channel, Timing Estimation Enabled, Fre- quency Estimation Disabled and Equalization Enabled/Disabled	72
5.4	Simulation Under an Ideal AWGN channel, No Frequency Offset, Frequency Estimation Enabled, Timing Estimation and Equalization Disabled	72
5.5	AWGN Channel with Timing Offsets, Estimators Disabled	73
5.6	Ideal AWGN Channel with Frequency Offsets, Estimators Disabled	74
5.7	2-Ray Model Frequency Response	75
5.8	2-Ray Model Channel with Varying Fitting Orders	76
5.9	2-Ray Model and 1000 Hz Offset Channel with Varying Fitting Orders	77

List of Tables

1.1 Application System Parameters	15
---	----

Glossary of Terms

ADC	Analog-to-Digital Converter.
ASAM	Adaptive Spectrum Aggregation & Management.
AWGN	Additive White Gaussian Noise.
CIC	Cascaded Integrator Comb.
DAC	Digital-to-Analog Converter.
DFT	Discrete Fourier Transform.
DSP	Digital Signal Processing.
FCC	Federal Communications Commission.
FDE	Frequency Domain Equalizer.
FDM	Frequency Division Multiplexing.
FEC	Forward Error Correction.
FFT	Fast Fourier Transform.
FIR	Finite Impulse Response.
FPGA	Field-Programmable Gate Array.
GT	Ground Terminal.
IDFT	Inverse Discrete Fourier Transform.
IFFT	Inverse Fast Fourier Transform.
ISI	Inter-Symbol Interference.
LDPC	Low Density Parity Check.
LNA	Low Noise Amplifier.

OFDM	Orthogonal Frequency Division Multiplexing.
OFDMA	Orthogonal Frequency Division Multiple Access.
PAPR	Peak to Average Power Ratio.
PSF	Pulse Shaping Filter.
PU	Primary User.
QAM	Quadrature Amplitude Modulation.
RF	Radio Frequency.
RRC	Root Raised Cosine.
SNR	Signal-to-Noise Ratio.
SU	Secondary User.
T&E	Test & Evaluation.
TA	Test Article.
UU	Unlicensed User.

Chapter 1

Introduction

As the demand for radio and microwave spectrum increases, we are posed with new challenges of how to best allocate and utilize spectrum. Over the past few decades the wireless communities have been forced at the same time to adapt to new technologies that have higher spectral efficiency, reduce spectrum use, and bid for rights to use newly allocated spectrum. As available spectrum becomes limited and more expensive to license, organizations are seeking to use underutilized or infrequently used spectrum. While the Primary User (PU), holds a license from a regulatory commission to transmit on a given spectrum, the PU may not always occupy the spectrum at all times or at all locations. Many regulatory commissions such as the Federal Communications Commission (FCC) have allocated spectrum where Secondary Users (SU) or Unlicensed Users (UU) may operate as long as they do not cause harmful interference to the PU and can accept interference from the PU [1].

The ability to operate as an SU requires intelligent spectrum management techniques, commonly referred to as Cognitive Radio. If the SU additionally requires use of large bandwidths, availability of contiguous bandwidth may not be sufficient. In this case, Adaptive Spectrum Aggregation and Management (ASAM), techniques must be employed to allocate data into non-contiguous blocks of spectrum. A successful ASAM effort must perform the following tasks:

- Implement opportunistic radios to identify white space in spectrum
- Aggregate non-contiguous blocks of spectrum into a single duplex data channel
- Dynamically assign channels to users based on predefined and ad hoc policies

To accomplish these tasks, the radios must incorporate wideband transmitters and receivers that are both frequency-agile and efficient to implement in hardware. A generalized architecture for the transmitter is illustrated in Figure 1.1. To support non-contiguous spectrum, the modulator must distribute the message bits $m[n]$ into C channels. Then C parallel processes modulate each of the allocated channels before passing the data to a transmitter channelizer. The TX Channelizer forms up to C channels that are frequency multiplexed at baseband and provides frequency assignment. The channels are combined, undergo digital-to-analog conversion (DAC) before being quadrature modulated to a real bandpass signal $s(t)$. High-lighted blocks represent the topics that will be discussed in more detail. The elements to the left of the dashed line are implemented in a digital platform such as an Field-Programmable Gate Array (FPGA).

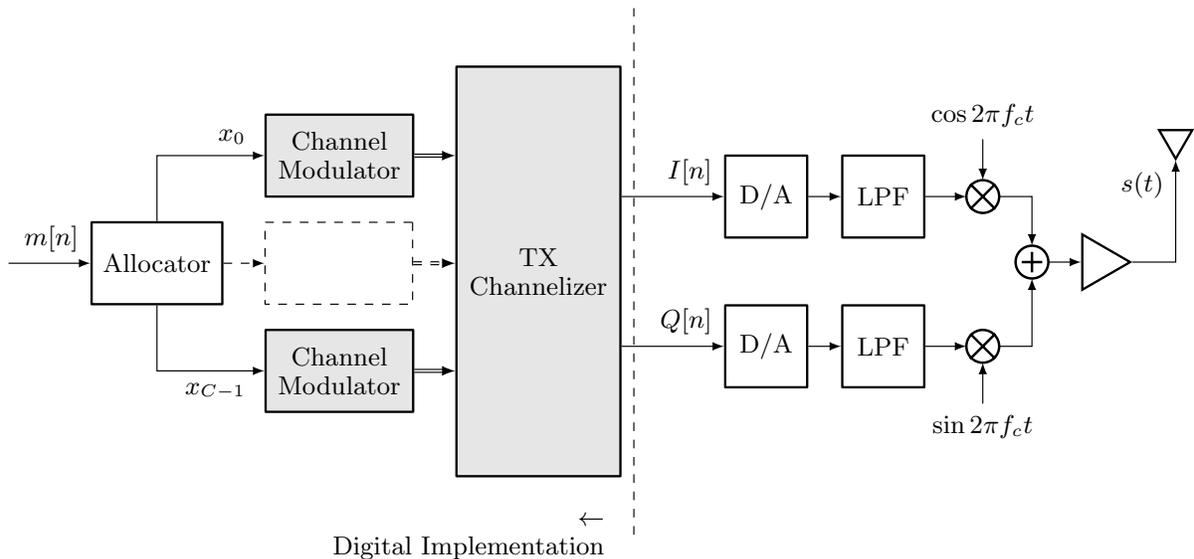


Figure 1.1: Transmitter Design

The dual of the transmitter is the receiver shown in Figure 1.2 where there are C channel

demodulators. Everything to the right of the dashed lines is implemented in digital hardware. Here the received signal $r(t)$ is equal to $s(t) * h(t)$ where $h(t)$ is the impulse response of the channel, incorporating attenuation, path delay, and multipath, if any. Additive White Gaussian Noise (AWGN) $n(t)$ is introduced due to shot and thermal noise at the receiver front end. The real bandpass signal $r(t) + n(t)$ is quadrature demodulated before analog-to-digital conversion (ADC). The complex baseband signal is then passed to the receiver channelizer which performs frequency translation and low-pass filter for individual channels. A selector chooses from the M number of available channels to K number of active channels for demodulation. Active channels then undergo demodulation and decision making. After decision-making the carriers are then combined to generate $\hat{m}[n]$.

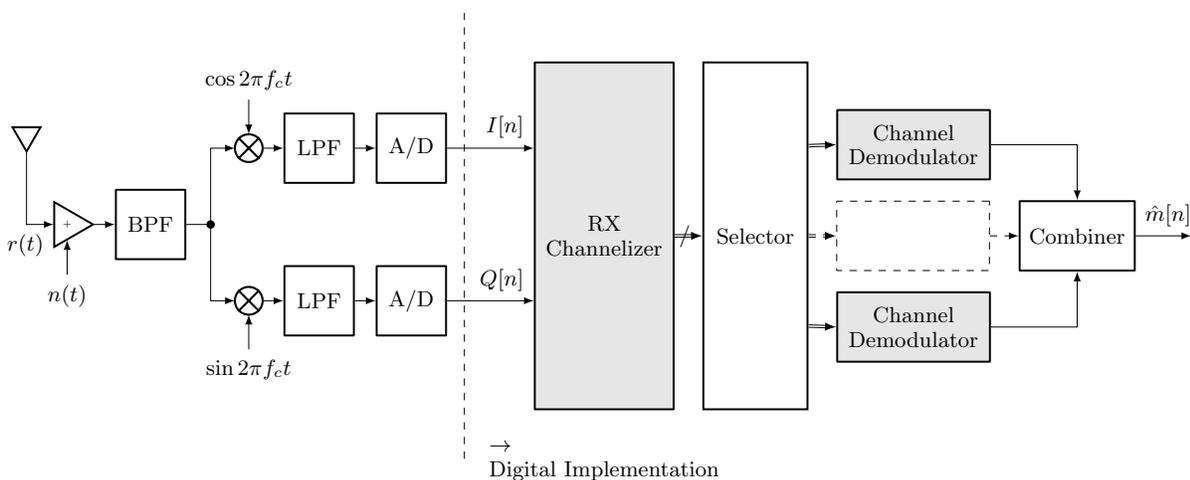


Figure 1.2: Receiver Design

1.1 Application

While the architectures presented in this thesis are generic and apply to a wide range of applications, we will at times reference a specific application¹ and requirement which should

¹Project Adaptive Spectrum Aggregation & Management funded by the National Spectrum Consortium

1.1. APPLICATION

help bring context and constraints to the study. In this application we shall set a scenario where we have one ground terminal (GT), which serves as a base station, and multiple users, or test articles (TA). The TAs are airborne and must communicate with the GT while carrying out test and evaluation (T&E) missions. For this application, a 320 MHz contiguous spectrum has been selected for use between 5.9 and 6.2 GHz. It can be assumed that the aircraft conducting T&E missions are capable of supersonic travel and speeds up to Mach 3 with a communications range of 100 km will be assumed. The TAs have a stated requirement to support up to 10 Mbps in the downlink (TA-to-GT). We assume that the application will use a high-rate Forward Error Correction (FEC) code such as a Low Density Parity Check Code (LDPC). We assume the majority of the traffic is on the downlink and that the GT requires only a single dedicated channel for transmission. The ASAM system overview is shown in Figure 1.3 which illustrates high-level requirements of the ASAM system.

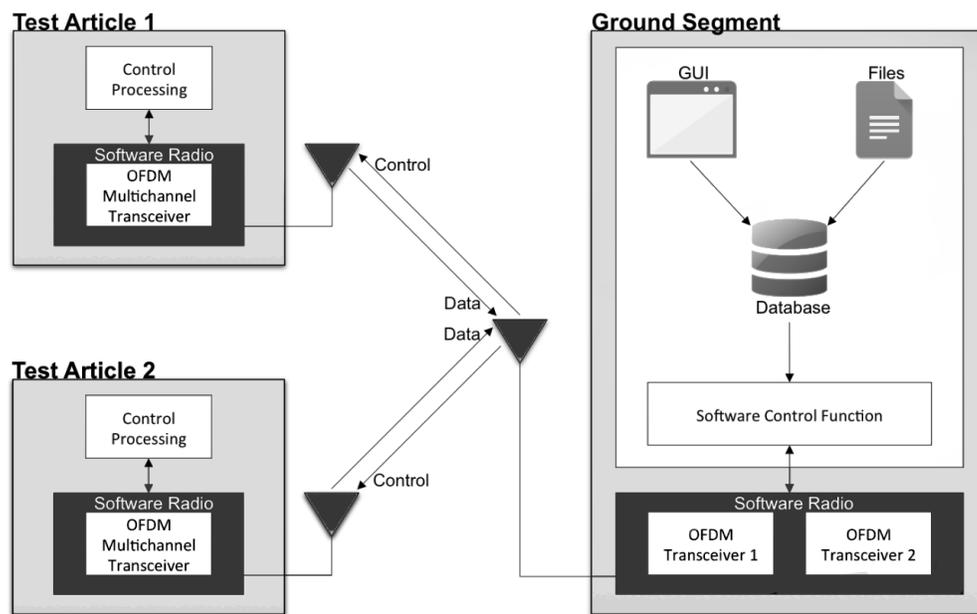


Figure 1.3: Major ASAM System Components

As there are a number of licensed carriers located in upper C-Band, the spectrum chosen for this study, the air-to-ground links must be adaptive and intelligent. On the other hand we shall assume that the GT uses a prescribed transmit frequency in C-Band where it is

licensed to operate as a primary user. The TAs will be assumed to be operating as secondary users, and thus must yield to all primary users, including intermittent transmitters.

An example frequency plan is shown in Figure 1.4, again with two users shown. Within this spectrum are a number of PUs and SUs that must be avoided. Some of these bandwidths may be known prior to the T&E mission and may be pre-programmed as reserved. Other PUs and SUs must be identified based on white space spectral sensing. Thus each receiver in the GT and TA must be capable of wideband power sensing and this information should be shared with the GT to be used in spectrum assignment.

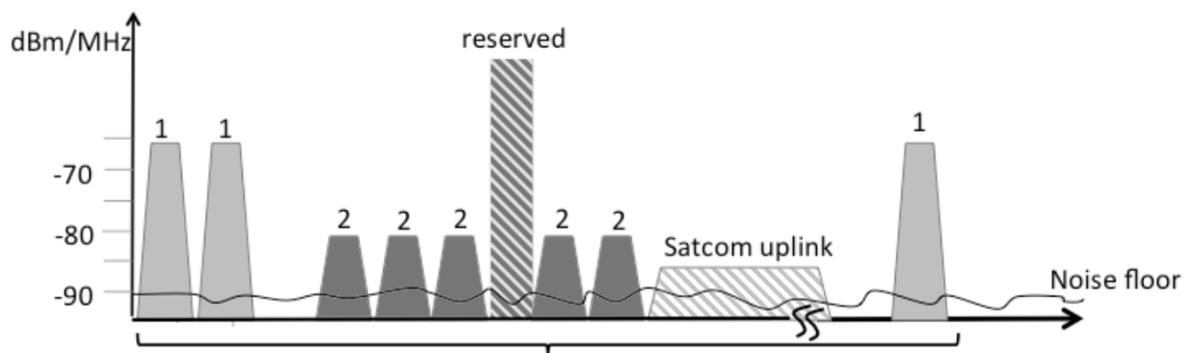


Figure 1.4: Frequency Plan

As we assume that the GT transmits on a licensed, predefined channel, discovery and initial handshaking can be performed using already existing methods. To avoid unintentional interference, the TAs will not enable transmitters until handshaking is complete and has received instructions from the GT. At predetermined intervals all transmitters will disable to enter a ‘quiet’ state so that spectrum power may be measured and PUs identified. To compute power, no demodulation is required. The outputs of the receiver channelizer block will perform simple power sensing by computing the mean-squared value for each channel. The measured power will be compared to a predetermined threshold such that decisions can be made on whether the selected channel is in use by PUs or is available. Information from

all TAs will be used by the GT when determining spectrum availability to avoid near-far issues.

1.2 Orthogonal Frequency Division Multiplexing (OFDM) Modulation

A natural modulation candidate for this application is Orthogonal Frequency Division Multiplexing (OFDM) Modulation. OFDM is a type of frequency division multiplexing (FDM) in which Quadrature Amplitude Modulation (QAM) subcarriers are spaced on the interval of the symbol duration, $1/T$ [10]. We define this to be the subcarrier spacing, f_Δ . The spectrum of this is the sum of overlapping *sinc()* functions with null-to-null bandwidth of $2f_\Delta$. This spectrum is shown in Figure 1.5. Note that assuming the subcarriers are coherent, the frequencies are orthogonal, and the subcarriers will not interfere with each other.

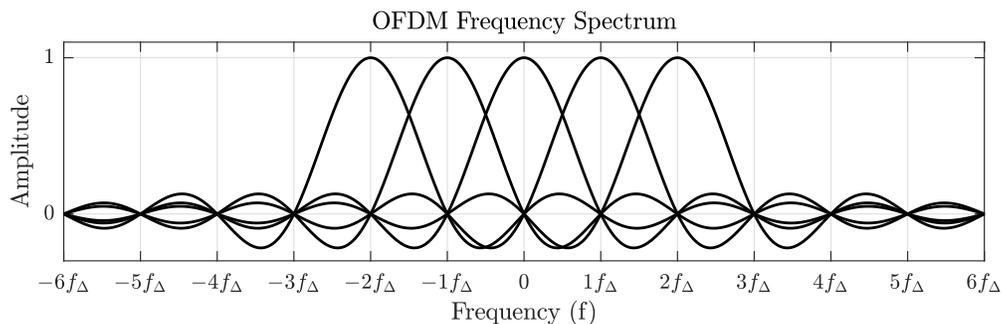


Figure 1.5: OFDM Subcarriers

There are three key benefits to OFDM:

- allows for denser channel packing
- provides efficient channel estimation for wideband channels
- modulation structure lends itself well to multi-user OFDM and Orthogonal Frequency

Division Multiple Access (OFDMA) applications

For these reasons, many applications such as LTE, 802.11a/g/n/ac, WiMAX, and DVB-T use this modulation scheme.

There are a few drawbacks of OFDM:

- high peak-to-average power ratio (PAPR)
- timing estimation is more challenging than single-carrier QAM requiring the synchronization of OFDM symbols
- requires more computations than previous modulation methods

A competing modulation method that might be considered is single-carrier QAM. While single carrier QAM is simpler to implement, it is more challenging to perform wide band equalization on. The ease of implementing wideband equalization is a key reason for using OFDM in the application under review. In this application we assume that the delay spread may be large so the coherence bandwidth will be too small for easy implementation of conventional single-carrier QAM.

Another reason for selecting OFDM is that it scales easily in multi-user scenarios. While the TAs will use single-carrier OFDM, the GT could use OFDMA. The OFDM spectrum may be viewed in both frequency and time to show that subcarriers could be placed into groups to form what is called resource blocks. Some resource blocks will be dedicated to control signals while others may be dynamically allocated to users. The dynamic nature of the resource block means that if one user requires a higher throughput for a given period, all that is needed is that user to be allocated more resource blocks. An example structure is shown in Figure 1.6 for two users.

The resource block assignment shown is rather arbitrary but it highlights the flexibility allowed in OFDMA in that resource block assignments are allowed to change over time based

1.2. ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM) MODULATION

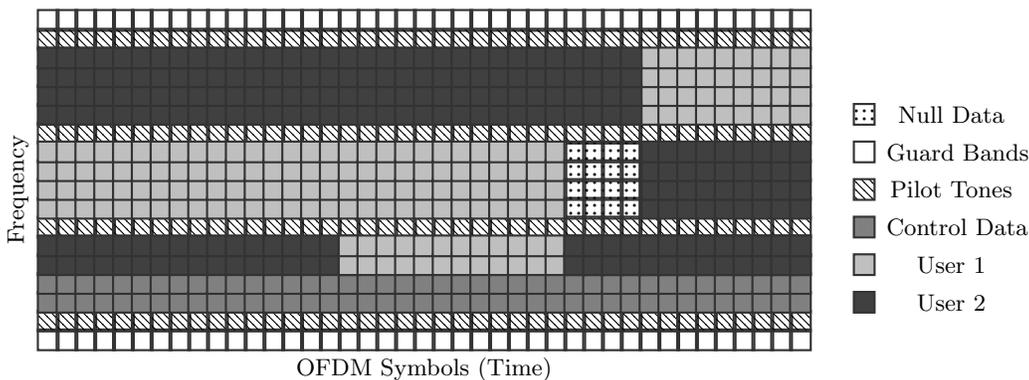


Figure 1.6: OFDMA Resource Grid

on current demands of the application. Also shown in Figure 1.6 are guard bands and pilot tones. These subcarriers are considered to be overhead as they carry no data. Guard bands are placed at the edges of the spectrum as these subcarriers will suffer from uncompensatable inter-symbol interference (ISI) distortion. This will be discussed in Chapter 3.

Pilot tones are used to support frequency-domain channel equalization, a key benefit of using OFDM modulation. The pilot tones are distributed throughout the frequency spectrum and consist of known symbols. The frequency domain equalizer (FDE) will use these known symbols to produce an estimate of the channel's response through an interpolation process. This FDE is possible due to the addition of a cyclic prefix that is added to the OFDM frame [2].

The pilot tones shown in Figure 1.6 are shown to be on the same subcarrier for every OFDM frame in time. While this is how our OFDM structure is defined, it is common for communication standards (e.g. LTE) to scatter the pilot tones more sparsely to reduce modulation overhead. This is an acceptable practice if a slow time-varying channel is assumed. When the pilot tones are more sparse, the equalizer will retain memory of past OFDM frames and incorporate previous pilot tones into the current frames' channel estimator.

From Figure 1.5 it can be noted that the out-of-band emissions of the combined signal are

quite high due to the slowly decaying sidelobes. It is common to perform pulse shape filtering post OFDM modulation to reduce the sidelobes and increase spectral efficiency. We are not concerned about in-band decay that exist due to nulled subcarriers in this application as users are using single carrier OFDM. The process of designing efficient and optimal pulse shape filters (PSF) will be discussed in Chapter 4.

1.2.1 OFDM System Overview

We may express OFDM modulation as [10]

$$\tilde{s}[m] = \sum_{n=0}^{M-1} x_n e^{j2\pi mn/M} \quad m = 0, 1, \dots, M-1 \quad (1.1)$$

where x_n are complex QAM symbols. This equation is equivalent to the Inverse Discrete Fourier Transform (IDFT) thus for modulation the IDFT may be used and the DFT may be used for demodulation. When using modern computing hardware, the IFFT and FFT are efficient implementations that are used to compute these transforms.

A typical OFDM system is shown in Figure 1.7. After the inverse FFT, the signal is up-sampled at an oversampling rate OS and the filtered using a pulse shaping filter. The signal is then converted from discrete time to continuous time using a Digital-to-Analog converter before transmission. At the receiver, the continuous time signal is converted back to digital by the Analog-to-Digital converter at the oversampled rate OS . The signal is filter using the matched filter of the transmitter, decimated, and then demodulated using the forward FFT.



Figure 1.7: OFDM Block Diagram

1.2. ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM) MODULATION

Between the transmitter and the receiver, the signal $s(t)$ is sent over a channel $p(t)$ such that the received signal $r(t)$ is

$$r(t) = x(t) * p(t) \quad (1.2)$$

If we can say (1.2) is equal to

$$\tilde{r}(t) = \tilde{x}(t) \otimes \tilde{p}(t) \quad (1.3)$$

then we may say

$$\tilde{Y}[m] = \tilde{X}[m]\tilde{P}[m] \quad (1.4)$$

which means we can recover X by dividing Y by the channel response P . So if we are able to force the channel to obey cyclic convolution we will have a convenient way to perform channel equalization in the frequency domain.

This is accomplished by the addition of adding a cyclic extension to the transmission to create an OFDM symbol shown in Figure 1.8. The limitation is that in order for cyclic convolution to be used, the cyclic prefix must be at least as long as the impulse response of the channel $p(t)$. The channel may be estimated by using pilot tones which are subcarriers with known values.

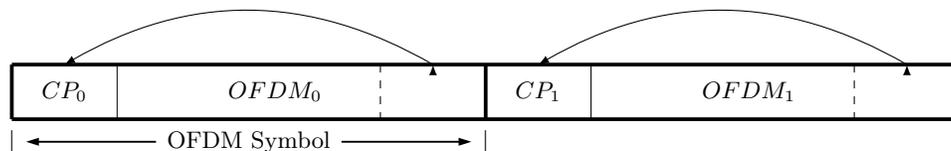


Figure 1.8: OFDM Symbols

1.2.2 OFDM Channel Modulation

Each active channel² requires its own OFDM modulator, shown in Figure 1.9. The OFDM modulator takes in a stream of bits converting them to M-ary QAM symbols while performing a serial-to-parallel operation at the same time. The number of data symbols at the output per frame is the size of the N -point IFFT minus the number of pilot and guard tones used. The next step generates the OFDM frame X_N by performing subcarrier mapping on D_n , including insertion of the guard and pilot tones. The frequency-mapped OFDM frame X_N is then converted into a time domain signal through use of the IFFT before cyclic prefix insertion. The data is then serialized as s_n at a rate of R_s which will then go to the channelizer block in Figure 1.1. Not shown in this block diagram is the upsampling and pulse shaping that is typical in OFDM systems. For the architectures presented, the upsampling and pulse shaping occur in the channelizers.

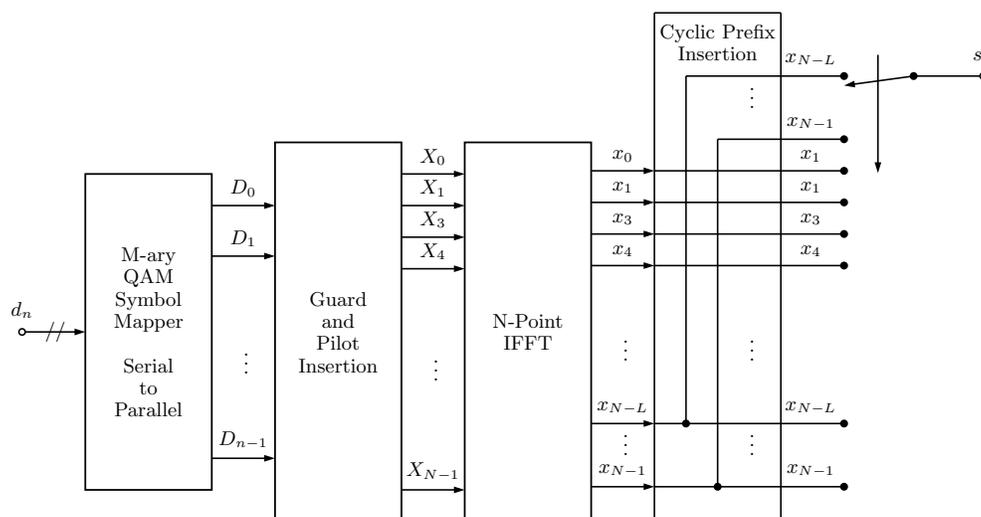


Figure 1.9: OFDM Modulation Block Diagram

²We will use the terms channel and carrier interchangeably

1.2.3 OFDM Channel Demodulation

The process for OFDM demodulation is similar to reversing the process of modulation, with the exception that frame parsing is required prior to demodulation and channel estimation is required after demodulation. The block diagram of this process is shown in Figure 1.10 where r_n is data from the output of the receiver's channel selector block.

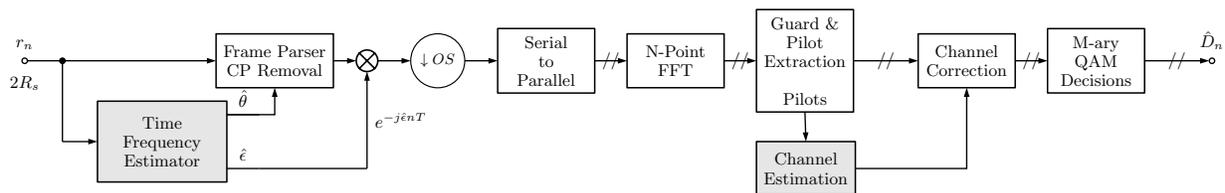


Figure 1.10: OFDM Demodulation Block Diagram

Frame parsing is required as the receiver does not have information regarding the arrived time of OFDM symbols. The process of timing and frame estimation will be discussed in Chapter 3. The parser uses information from the timing estimator to determine when to extract frames, removing the cyclic prefix in the process. A frequency offset correction is applied based on the estimated frequency offset $\hat{\epsilon}$. Typically, the time and frequency estimator are running at a higher rate than R_s so the signal needs to be decimated by the oversampling factor OS prior to serial-to-parallel conversion. From here on, the data is handled in parallel blocks. The N -point FFT converts the time domain signal back to the frequency domain. The data and pilots are extracted separately so that channel estimation may be done using the pilot tones and then applied to the data. After channel correction, M-ary QAM decisions are performed on the data, generating the demodulated data \hat{d}_n .

1.3 Application System Parameters

Multi-carrier OFDM modulation has been selected for this application. The 320 MHz spectrum will be split up into 256 channels spaced by 1.25 MHz. Root-Nyquist pulse shaping with a roll-off factor $\alpha = .25$ will be used. This implies 1 Msps at the output of the OFDM modulator. An FFT size of $N = 128$ is selected which provides a subcarrier spacing of 7.8 kHz. We assume a maximum transmission distance of 100 km and will use a cyclic prefix overhead of 12.5% for channel correction, allowing for a channel with a time dispersion of $12 \mu\text{s}$.³ The total OFDM symbol duration is $144 \mu\text{s}$ which will require each channels' FFT to operate at approximately 7000 FFT/s.

The subcarrier assignment that will be used is shown in Figure 1.11. A total of 18 subcarriers will be used as a guard band, with nine guard subcarriers on each side and the center DC carrier nulled to prevent any issues of a DC offset in the RF hardware. Thirteen pilot tones will be used which are distributed evenly between the data subcarriers. The remaining 96 subcarriers are available for data and are modulated using M-ary QAM. The effective symbol rate is 667 kSps which for 4-QAM translates to 1.333 Mbps of uncoded data and 2.667 Mbps of uncoded data for 16-QAM. In the simulation study we will use 4-QAM, however it should be noted that constellation sizes may easily be increased provided enough transmitter power is available. When the TAs require a higher bit rate than is supported by a single channel, additional channels will be allocated for use, which may not be contiguous, based on available spectrum to achieve the required throughput.

To quantify performance of the system and our filter designs, we specify a required ISI signal-to-noise ratio (ISI) to be greater than 40 dB and a peak sidelobe \mathcal{S}_P to be less than -50 dB relative to the mainlobe power density. A summary of system parameters is provided in Table 1.1.

³The channel delay spread is assumed to be less than 12,000 feet

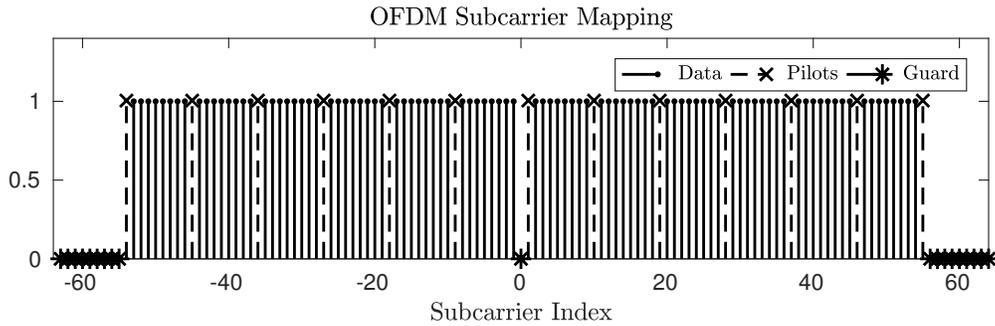


Figure 1.11: OFDM Subcarriers

1.4 Organization of Thesis

The scope of this thesis is to study the physical layer challenges that arise from designing and building wideband multi-channel transceivers. The high sample rate required to capture wideband signals drives the need for efficient architectures in order to be realized in available hardware. Two transmit and two receiver architectures will be reviewed in more detail including a metric for quantifying hardware complexity such that computational costs may be compared.

Below we will study filter design, optimization, and implementation techniques. The wideband architectures proposed require very long filters that can be challenging to design using conventional methods. We will present several methods of generating a short filter such that the long prototype filter may be generated. We will explore solutions to produce a prototype filter capable of correcting for in-band distortion caused by CIC interpolation. Additionally we will measure performance differences between symmetric and asymmetric FIR filters of the same length.

The thesis will simulate an end-to-end implementation of a two-user scenario using a filter design method presented. The system will be tested in non-ideal settings such as timing error, frequency offsets, and multipath effects which poses challenges to channel equalization. We

Application System Parameters	
Maximum Transmission Distance:	100 km
Total Bandwidth:	320 MHz
Number of Channels:	256
Channel Spacing:	1.25 MHz
Root-Nyquist Roll-off Factor:	0.25
Modulation Type:	OFDM
Constellation:	M-ary QAM (square)
OFDM Symbol Duration:	144 μ s
FFT Size:	128
Cyclic Prefix Overhead:	12.5% (16 Symbols)
Cyclic Prefix Duration:	12 μ s
OFDM FFT Rate:	7000 FFT/s
Subcarrier Spacing:	7.8 kHz
Data Subcarriers:	96
Guard Subcarriers:	18 + null DC subcarrier
Pilot Tones:	13
Effective Symbol Rate:	667 kSps
Required ISI	≥ 40 dB
Peak Sidelobe \mathcal{S}_P	≤ -50 dB

Table 1.1: Application System Parameters

will often reference our specific application to provide examples although the modulation and demodulation architectures presented are easily generalized and may be applied to a number of applications.

Chapter 2

Channelizer Architectures

The ability to design a multi-channel transmitter or receiver that utilizes a single RF radio is done through a channelizer. The primary purpose of the channelizer is to provide the rate change and frequency translation required to modulate or demodulate several baseband channels to/from a single RF carrier.

To quantify the cost of each architecture, we need to define a metric for cost. Cost may invoke several definitions such as the number of logic gates, amount of memory, or amount of computations required. Additionally, not all operations are equal. In our metric for computational cost, we shall adopt a simple definition of operations per second. We will make an assumption that additions and multiplications each have the same cost of one computation per real valued sample.¹ For FFT computations, we will assume an FFT algorithm [5] which has $4N \log_2 N - 6N + 8$ number of combined real additions and real multiplies. We will assume that upsampling, decimation, and buffer management are free operations and that lookup tables will be utilized whenever possible. Further, we will assume that for complex data the cost of addition is twice that of real data and the cost of complex multiplication is four times that of real data.

¹The accuracy of this metric may vary greatly between hardware as different hardware may be optimized for particular operations.

FIR Filters in their direct form for an N -tap filter will consist of N multipliers and $N - 1$ additions. If, however the filter is a symmetric FIR filter, the number of multipliers may be reduced to $(N + 1)/2$ for odd N and $N/2$ if N is even [20].

The most intuitive form of a receiver channelizer is demonstrated in Figure 2.1. This receiver performs independent heterodyning on each channel followed by a low pass filter, and then decimation by a factor of M . The process of performing low-pass filtering prior to decimation performs the purpose of matched filtering to maximize the Signal-to-Noise Ratio (SNR) and to perform decimation without producing aliases. The receiver in Figure 2.1 shows k number of channels being channelized out of C total channels. The architecture for the transmitter (not shown) is simply the dual of the receiver.

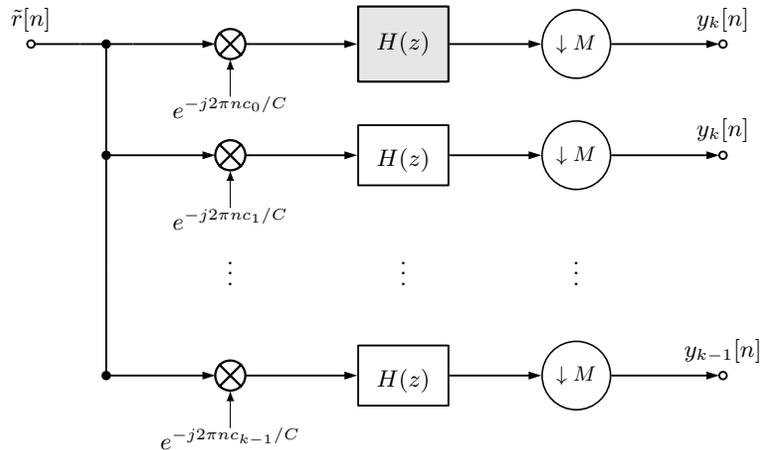


Figure 2.1: Simple Receiver Channelizer with Conventional Heterodyning

In our application study, an upsampling rate of $M = 320$ is required for number of channels $C = 256$ with a roll-off factor of $\alpha = 0.25$. Assuming a pulse shaping filter that is a 12 symbol pulse shaping filter, H_{PSF} would be a long filter with 3841-taps operating at 320 Msps. Even assuming $h_{PSF}[n]$ to be a symmetric filter, the number of computations exceeds 3×10^{12} computations per-second per-channel. It would be very taxing to implement such a receiver in hardware and considering this architecture does not scale well when adding channels, such a design is nearly impossible to use in our application.

2.0.1 Polyphase Filter

A common approach to reduce the computational burden of filtering using a long filter while interpolating or decimating is to implement a polyphase filter bank. One of the Noble Identity is used which states “The output from a filter $H(z^M)$ followed by an M -to-1 down sampler is identical to an M -to-1 down sampler followed by the filter $H(z)$ ” [7] and a similar statement applies to interpolation. These identities are shown in Figure 2.2 and allows filtering of h_{PSF} to occur after downsampling.

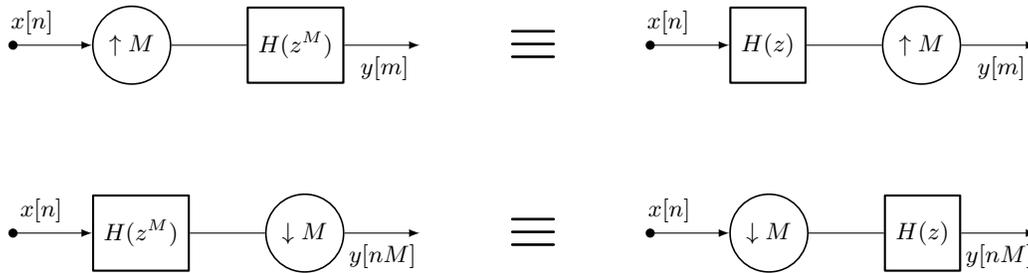


Figure 2.2: Noble Identities

To develop the polyphase architecture for a filter with z -transform $H(z)$ we can write out the summation

$$H(z) = \sum_{n=0}^{N-1} h[n]z^{-N} \quad (2.1)$$

using a polyphase decomposition [8] as

$$H(z) = H_0(z^M) + z^1 H_1(z^M) + z^3 H_3(z^M) + \dots + z^{-(M-1)} H_{M-1}(z^M) \quad (2.2)$$

Where

$$H_i(z) = h(i)z^{-i} + h(M+i)z^{-(M+i)} + H(2M+i)z^{-2(M+i)} + \dots \quad (2.3)$$

The block diagram for (2.3) is expressed in Figure 2.3 [8].

Applying the Noble property to Figure 2.3 moves the downsampling operation before $H(z^M)$

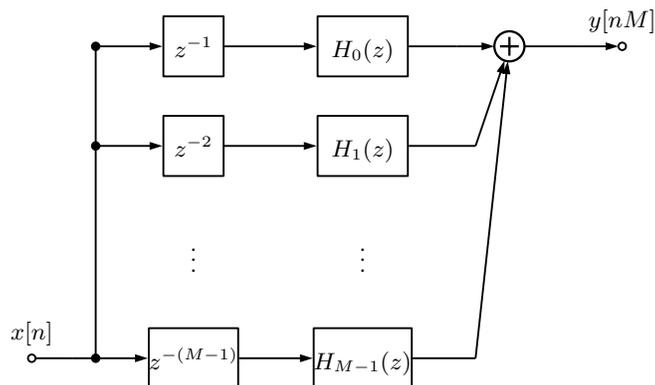


Figure 2.3: Polyphase Filter Structure

and may substitute the shaded block in Figure 2.1. Further, the delays may be thought of as commutators in DSP and FPGA hardware. The resulting polyphase interpolator and decimator filters are shown in Figure 2.4. Rate conversion occurs in the commutators which cycle at rate M and the filter $H_m(z)$ is a phase-shifted downsampled variant of the prototype filter $H_{PSF}(z)$. A design note on this structure is that there is no computational benefit to using a symmetric FIR filter, meaning asymmetrical FIR filters with greater attenuation characteristics may be considered.

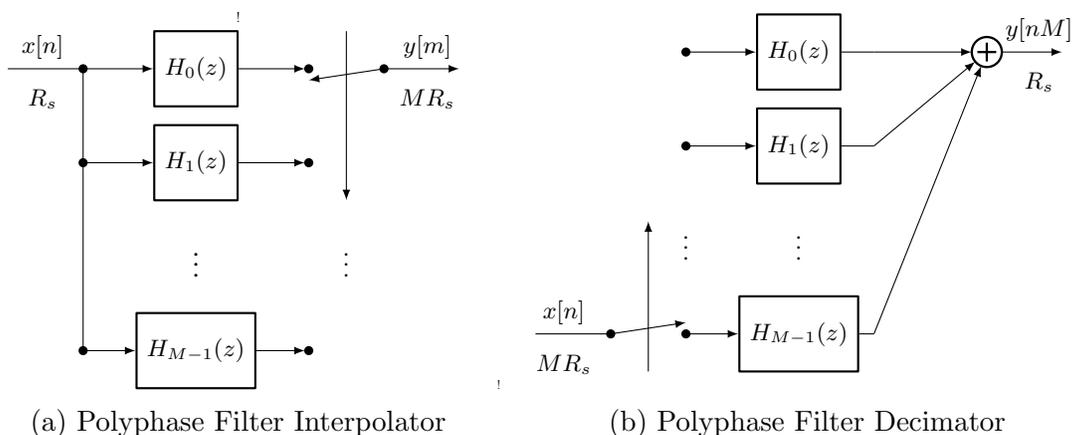


Figure 2.4: Rate Changing with Polyphase Filters

For context, we will take a look at our original channelizer in Figure 2.1. We consider the same requirements of a 12-symbol prototype filter with rate change $M = 320$ so that the

total filter length is 3840-taps long. Thus $H(z)$ has length 12 and takes the form:

$$\begin{aligned}
H_0(z) &= [h_0, h_{320}, \dots, h_{N-319}] \\
H_1(z) &= [h_1, h_{321}, \dots, h_{N-318}] \\
&\vdots \\
H_{M-1}(z) &= [h_{319}, h_{639}, \dots, h_{N-1}]
\end{aligned} \tag{2.4}$$

The benefit is now the filters $H_m(z)$ are operating at a speed of 1 Msps instead of 320 Msps. The savings on computations per second is significant, dropping to 1.7×10^{10} computations per channel per second. While the savings in computations is beneficial, the design can still be improved further by incorporating channelization as discussed next.

2.0.2 Polyphase Channelizer

Following Harris [8] to the polyphase filter structure already discussed, consider a case where all channels are populated in Figure 2.1. Apply the frequency translation property of the Z-Transform [8, 13] to translate a lowpass filter to a bandpass filter. Using this property we can say if $h[n]$ has a transform $H(z)$ then the sequence $h[n]e^{j\theta n}$ has a Z-transform $H(ze^{-j\theta})$ [8]

$$\begin{aligned}
H(z) &= h(0) + h(2)z^{-2} + h(3)z^{-3} + \dots + h(N-1)z^{N-1} \\
&= \sum_{n=0}^{N-1} h(n)z^{-n}
\end{aligned} \tag{2.5}$$

and

$$\begin{aligned}
G(z) &= h[0] + h[1]e^{j\theta}z^{-1} + h[2]e^{j2\theta}z^{-2} + \dots + h[[N-1]e^{j(N-1)\theta}z^{-(N-1)} \\
&= \sum_n^{N-1} h[n] [e^{-j\theta}z]^{-n}
\end{aligned} \tag{2.6}$$

then

$$G(z) = H(z)\Big|_{z=e^{-j\theta}z} = H(ze^{j\theta}). \quad (2.7)$$

Applying this translation to Figure 2.4b produces Figure 2.5.

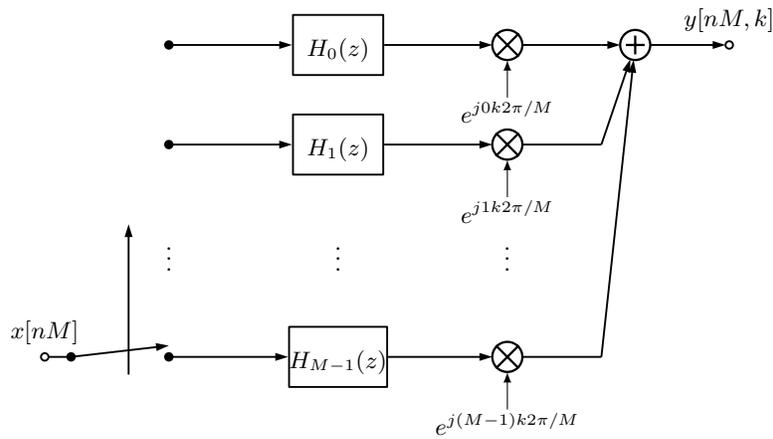


Figure 2.5: Resampling Downconverter, or Polyphase Channelizing Receiver

Note that the output is computed as a phase coherent summation of the output series [8] $y_r[nM]$. This output weighting is equivalent to the IDFT of the output of the filter bank which is expressed as

$$y[nM, k] = \sum_{r=0}^{M-1} y_r[nM] e^{j2\pi rk/M} \quad (2.8)$$

which allows the computations to be performed using the IFFT algorithm. It follows that if we can do this to a single channel, we may do the same for additional channels without significant addition of computations. The takeaway is that we may heterodyne a large number of channels simultaneously by incorporating a polyphase structure with little additional computational cost.

2.1 Receiver Architectures

In this section we will propose two receiver architectures. A key requirement of the receiver is for both the GT and TAs to be able to perform wideband spectral sensing during the quiet periods. The standard heterodyne approach is an unfeasible solution when using a large number of channels so we will only evaluate methods that utilize the polyphase channelizer.

Due to the excess bandwidth factor α required for pulse shaping, the output sampling rate of the polyphase channelizer needs to be $C_s = R_s(1 + \alpha)$ which is not a multiple of symbol rate R_s . This requires resampling that converts the rate to an integer value of R_s so that timing synchronization, decimation, and demodulation without ISI may be performed. The frequency spacing and bandwidth relationship is illustrated in Figure 2.6. The required resampling ratio is the rational fraction of the desired output oversampling rate L_O , divided by the channel spacing C_s . We will denote this rational fraction as $R_U : R_D$ where R_U is the numerator and R_D is the denominator of the rational fraction.

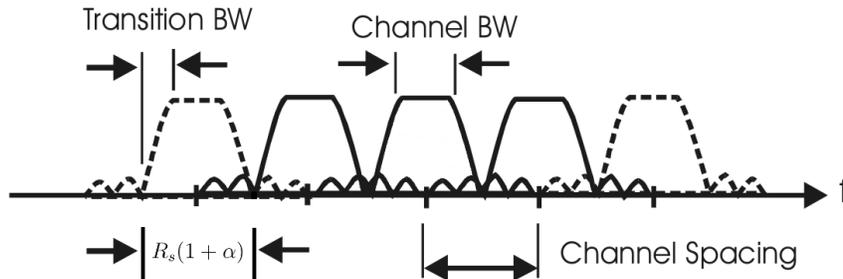


Figure 2.6: Spectrum and Channel Spacing [8]

2.1.1 Receiver I:

Polyphase Channelizer with External Resampler

The first receiver under study is a polyphase channelizer that performs this resampling externally. To derive this receiver, we extend Figure 2.5 to apply for multiple channels and replace the phase coherent summation with the IDFT, or IFFT, as (2.8) allows. The IFFT output node k corresponds to the k th channel of the demodulator. The components to the left of the Selector performs the same function as the structure in Figure 2.1.

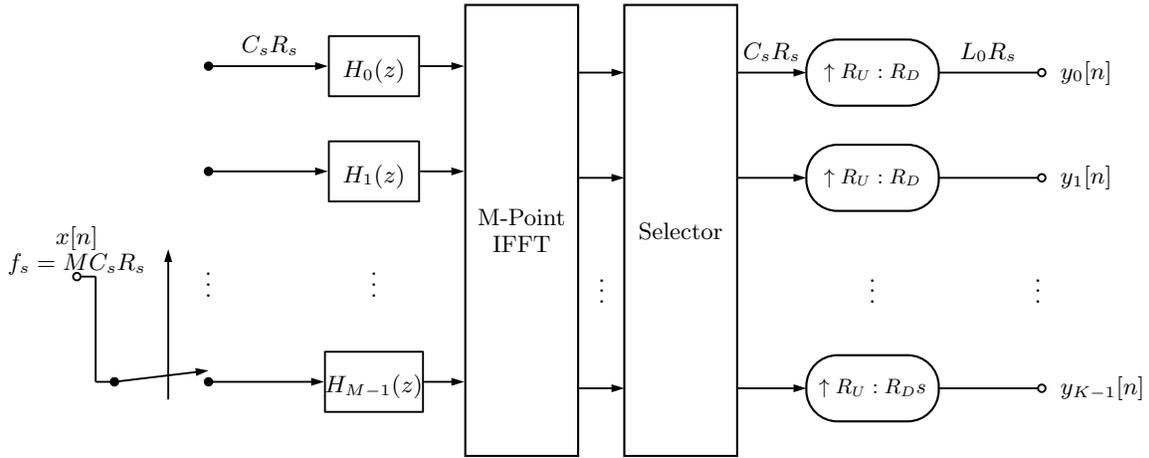


Figure 2.7: Receiver I: Polyphase Channelizer with External Resampler

In our application of study, we have 256 channels that are spaced $C_s = 1.25$ MHz so the polyphase channelizer filters and IFFT operate at 1.25 Msps. An output oversampling rate of $L_0 = 2$ is desired to provide 1/2 symbol resolution to the demodulator. Thus an 8 : 5 resampler is required. The resampler may be implemented by using a polyphase interpolator followed by decimation, as shown in Figure 2.8. The resampling filter must be chosen such that it provides good out-of-band rejection, otherwise aliases of the signal may cause ISI.

The resampler introduces additional computations that does not scale efficiently for a large number of channels. However we observe that the resampler is only required for the K active

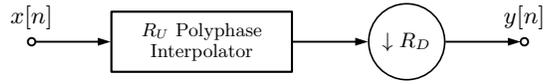


Figure 2.8: Standard Fractional Resampler

channels requiring demodulation. Spectrum sensing may still be done at the output of the IFFT as no resampling is needed for this task, so implementing a small number of resamplers in hardware is acceptable.

2.1.2 Receiver II:

Polyphase Channelizer with Internal Rate Conversion

For this second architecture, the goal will be to remove the need for using interpolation filters to perform the rate change. We will see how this can be done using circular buffers and a simple state machine, two operations that we consider to be computationally free. In FPGA and computer hardware, buffer management may be done by handling pointer addresses such that data does not need to be copied from one array to another. It will be the role of the state machine to perform 2-dimensional circular buffer rotations every FFT cycle. This method is derived in Harris and explained in more detail in [7, 8]. The overall generic structure is shown in Figure 2.9.

The rate change here occurs at the input to the input circular buffer and the ability to achieve this rate conversion is left up to proper circular buffer management [8]. Unlike in the Receiver I architecture, the commutator only fills the first MC_s/L_0 slots providing an output rate of L_0R_s . These slots are denoted with a filled node in Figure 2.9 and the empty nodes denote input placeholders. The input and output circular buffers are 2-dimensional buffers that are shifted up and right at the end of each commutation cycle. The amount of the shift and the size of the buffers depends on the system parameters so we will use our application to explain the use of the circular buffers.

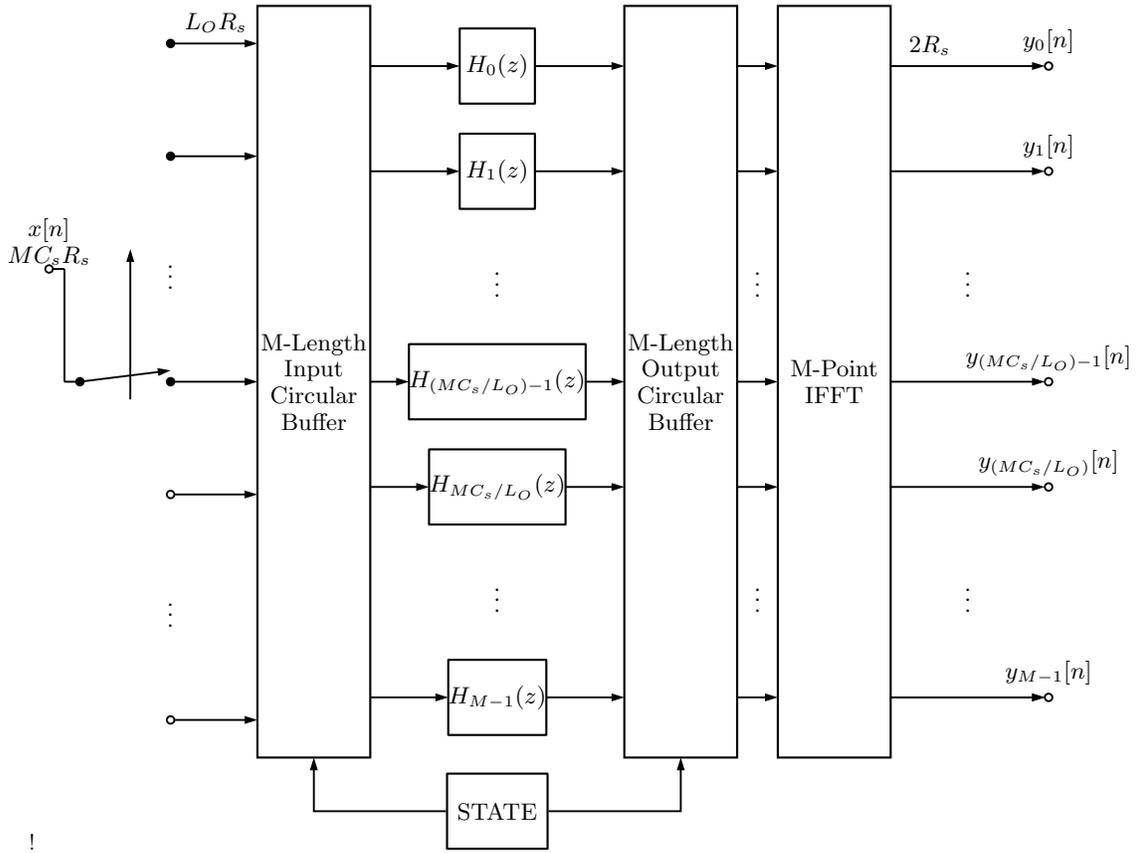


Figure 2.9: Overall View of Receiver II

In our system we have 256 channels so we assign the size of the FFT and buffers to $M = 256$ with a channel bandwidth of 1 MHz. Our excess bandwidth parameter is $\alpha = 0.25$ so our final sampling rate is $f_s = 320$ MHz. We wish to have a final sampling rate of $L_0 R_s = 2$ MHz as this is the desired input rate to the OFDM demodulators. This means that the input commutator will cycle from the 159th node to the 0th node and back to the 159th. The input circular buffer is 256 long but the inputs of locations 160 to 255 are always empty.

The contents of the input circular buffer is shown in Figure 2.10 for four cycles of the commutator. The number of columns is the length of the polyphase filter sections, $H_m(z)$, while the number of rows correspond to the total number of channels. For simplicity we will use a polyphase filter length of 3 in the graph and thus 3 columns are shown. The rows are divided into 8 buffers of length 32 each and denoted as B_i . The commutator then fills 5 of

2.1. RECEIVER ARCHITECTURES

the 8 buffers starting at the indicator arrow and rotating up during the commutation cycle. After the commutator cycle is done filling B_0 through B_4 , the buffer is rotated upwards prior to being shifted right on the first 5 rows, denoted by the double line. This process is shown for four commutator cycles of 160 samples each, for 640 incoming samples total. For each commutation cycle the output of the input buffer contains 256 samples, providing the 8 : 5 rate change required. The rotation of the buffers prevent discontinuities that would otherwise occur performing this rate change.

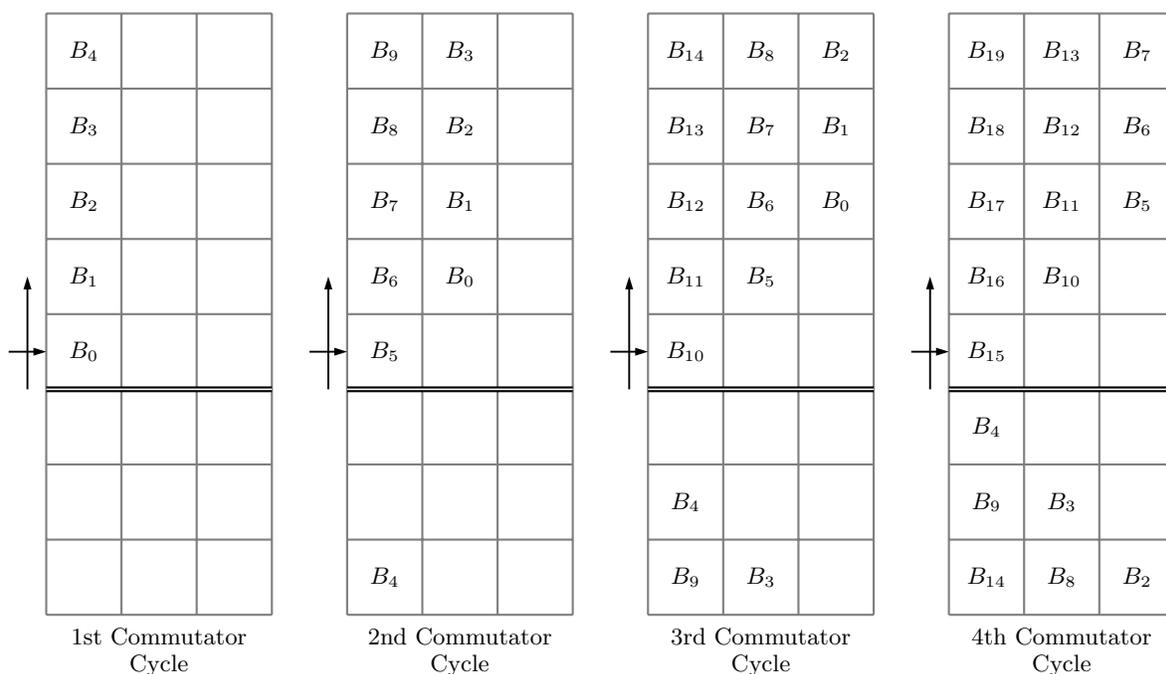


Figure 2.10: Input Circular Buffer States

After each cycle of the commutator, the circular buffer output is filtered using the polyphase filter design, taking note that the filter states are shifting at every cycle.

The Output Circular Buffer must complete the circular rotation required to prevent discontinuities in the incoming data [8]. The operation of this buffer is to rotate the column every commutator cycle by a multiple of the buffer size B_i . For our example that means we perform a shift of either 0, 32, 64, ... after each commutator cycle. We have a total of 8

states, 4 of which are shown in Figure 2.11 to help illustrate the process.

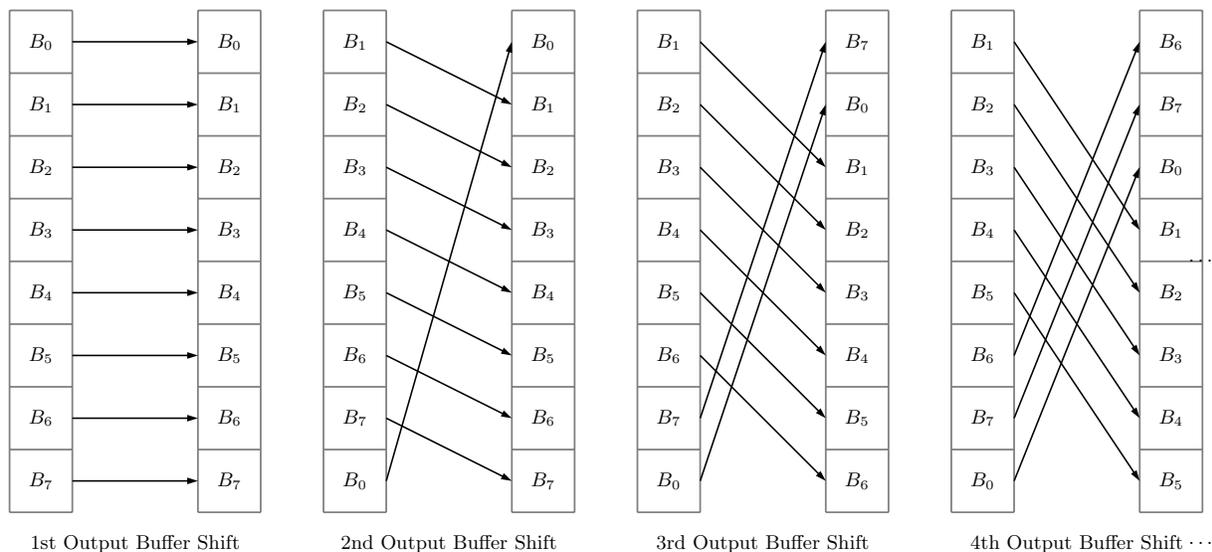


Figure 2.11: Output Circular Buffer States (4 of 8 States Shown)

The result is a receiver channelizer architecture that provides no additional computations to channelize 1 or 256 channels. The rate change is handled internally and entirely by buffer management and a simple state machine, eliminating the need for an external resampler.

2.1.3 Comparison of Receivers

The total addition and multiplication computations per second is shown in Figure 2.12 for the two receivers discussed. Note that Receiver II is independent of number of channels due to our assumption that buffer management is free. The two polyphase channelizer designs are similar with the external rate conversion process expected to be more efficient until there are about 20 users. This estimate is based on the external resampler needing a size of approximately 104 taps, based on the estimation [7]

$$N = \frac{f_s}{\Delta f} \frac{\text{Attn, dB}}{22} \quad (2.9)$$

The external resampler is more efficient with a small number of active channels as the IFFT operates at a rate of $1.25R_s$ compared to internal resampler's IFFT operating at $2R_s$. As more external resamplers are required, the computational cost of resampling begins to dominate and becomes less efficient. However as they both use a polyphase architecture, their performance is similar.

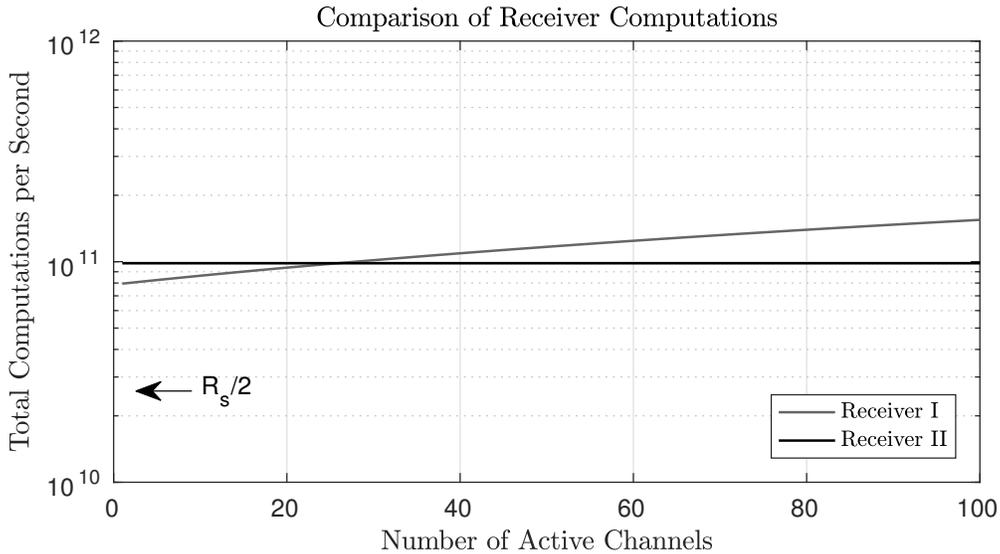


Figure 2.12: Comparison of Receiver Computational Costs per Number of Active Channel

2.2 Proposed Transmitter Architectures

The requirement for the transmitter differs from the requirements of the receiver in that not all channels must be channelized simultaneously. For a small number of channels, such as in the ground terminal which is assumed to only need one channel, a heterodyne channelizer approach may be more efficient. However the TAs require transmitting on multiple channels so a design that scales well is useful. We will present an architecture for each case.

2.2.1 Transmitter I: CIC Interpolator / Heterodyne

The first proposed transmitter architecture is a more efficient implementation of the heterodyning architecture discussed earlier. The channelizer will still use the heterodyne approach; however, instead of upsampling by 320 with a long interpolating filter, we perform a more modest upsample by L and then follow that with CIC (Cascaded Integrator Comb) interpolation at a much higher rate, $320/L$. The CIC interpolator provides an efficient means to perform large interpolation rates as it may be implemented using only additions and subtractions [11]. As the name implies, the CIC interpolator uses a comb section followed by upsampling and an integration section, which may be cascaded to achieve better stopband attenuation. Figure 2.13 shows this structure.

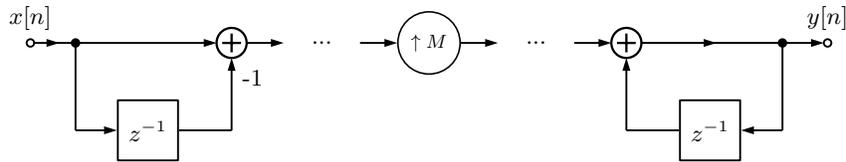


Figure 2.13: CIC Interpolator Filter

The transfer function for the CIC may be written as [11]

$$H_{CIC}(z) = \left[\frac{1 - z^{-M}}{1 - z^{-1}} \right]^P \quad (2.10)$$

where M is the CIC interpolation rate and P is the number of cascaded sections. The frequency response is then be found to be [7]

$$|H_{CIC}(f)|^2 = \left(\frac{\sin \frac{\omega M}{2}}{\sin \frac{\omega}{2}} \right)^{2P} \quad (2.11)$$

This frequency response is *sinc*-like with nulls at multiples of f_s/M . To reduce the magnitude

2.2. PROPOSED TRANSMITTER ARCHITECTURES

of the sidelobes, the number of CIC sections P must be increased. This relation is shown in Figure 2.14 for $P = 2, 3, 4$ and $L = 4$. Note that for higher number of sections P , the inband distortion becomes greater, requiring either higher L or more compensation to the pulse shaping filter. More details on designing compensated filters is discussed in Chapter 4. The figure also shows areas where aliases will appear in the spectrum.

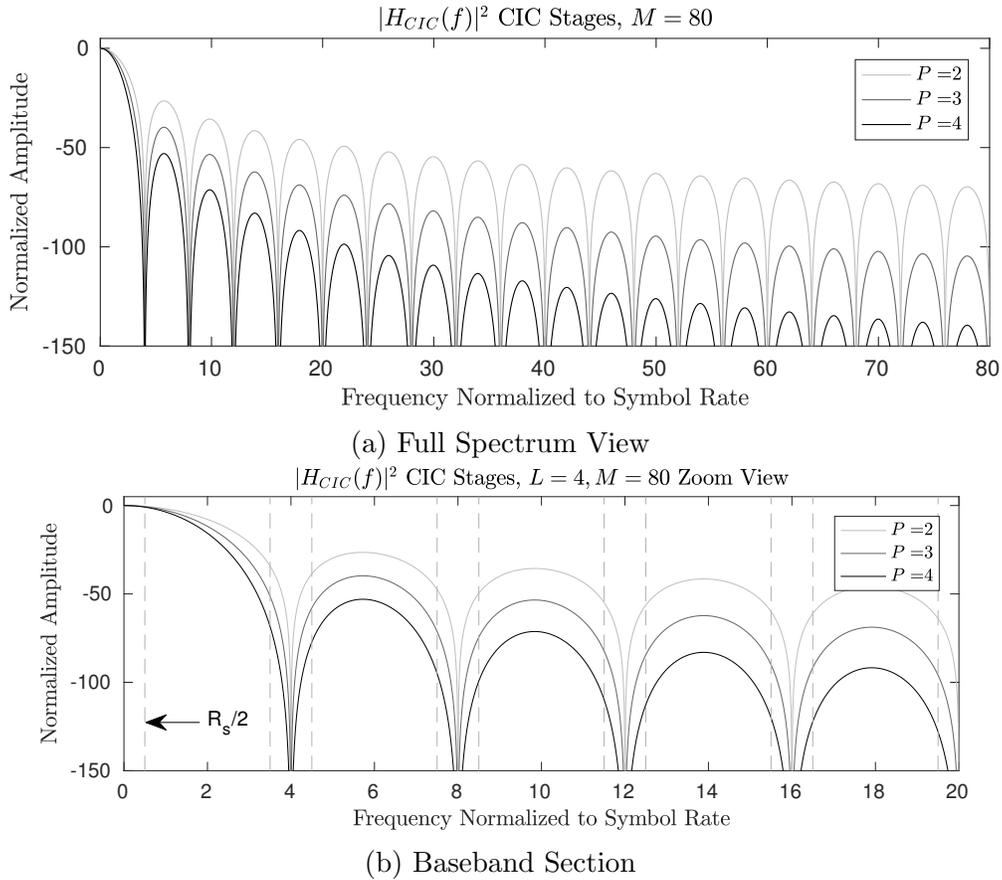


Figure 2.14: CIC Interpolation Frequency Response for P Stages

The final CIC interpolating channelizer is shown in Figure 2.15. For our application, we will use $L = 4$, $M = 80$, and $P = 3$ to achieve the conversion from $R_s = 1$ Msps to 320 Msps. Having 3 stages of CIC is just enough to bring the peak sidelobe below -50 dB, as Figure 2.16 shows. If it is desired to further attenuate the aliases, the number of stages could be increased. However either L or the prototype filter length will need to be increased to

compensate for the additional in-band distortion.

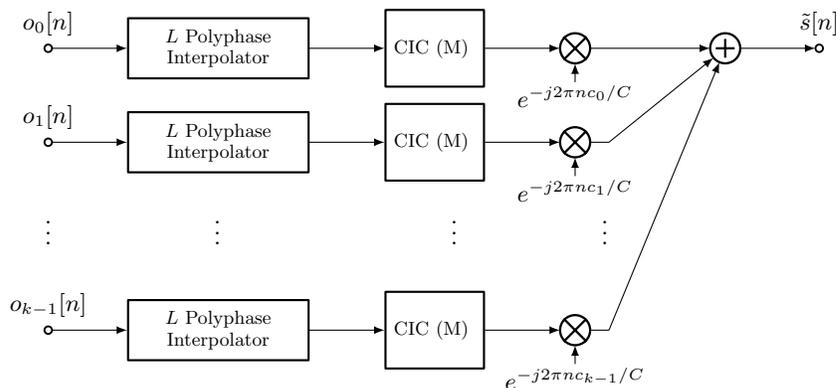


Figure 2.15: Simple Transmitter Channelizer for Relatively Few Channels

The effects on aliasing may be shown with an example filter response as shown in Figure 2.16 for $L = 4, P = 3$, with the filter extending beyond $R_s/2$ due to the $\alpha = 0.25$ excess bandwidth factor. We observe how the pulse shaping filter aliases are aligned with the nulls of the CIC filter so L and P will affect the peak sidelobe level \mathcal{S}_P .

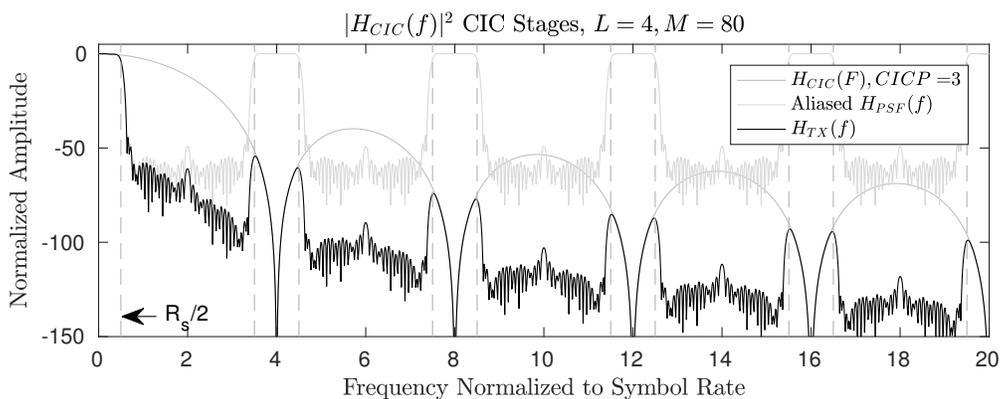


Figure 2.16: Output Frequency Response for CIC, $L = 4, P = 3$

2.2.2 Transmitter II:

Polyphase Channelizer using Internal Rate Conversion

The second transmitter design performs rate conversion in a similar method to the Receiver II, the receiver with internal rate conversion, however it is important to note that the transmitter is not the dual of the receiver [6]. Functionally they are equivalent in that they use a polyphase channelizer and perform rate conversion internally using buffer management but the buffer structure is different [6] as shown in Figure 2.17.

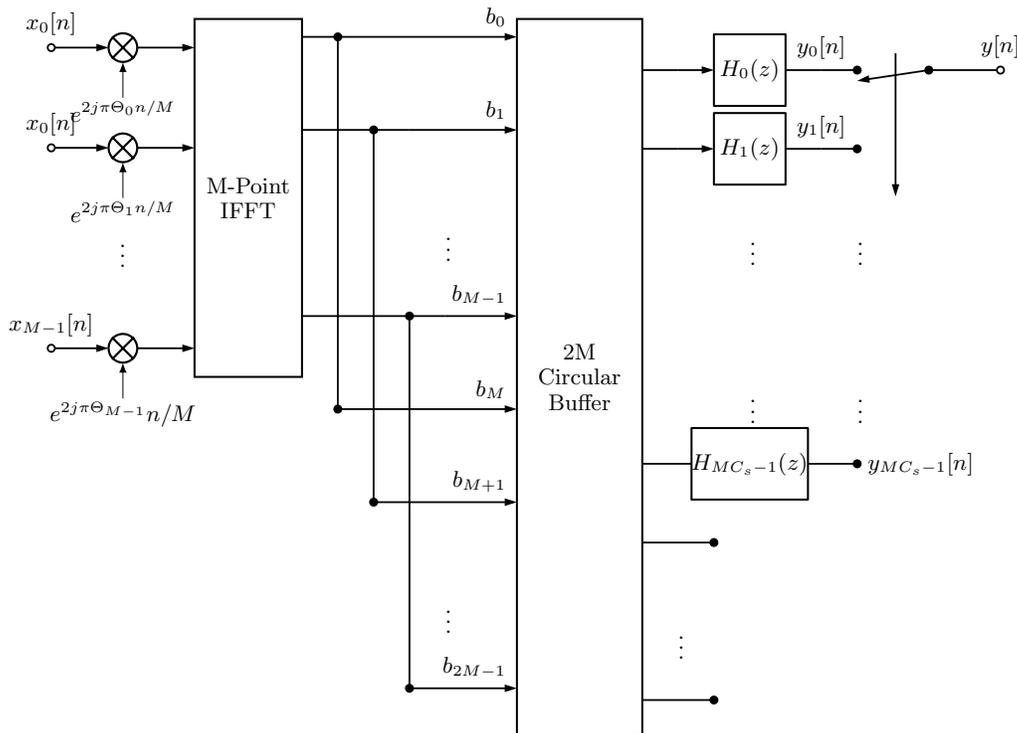


Figure 2.17: Transmitter II: Polyphase Structure with Internal Rate Conversion

This transmitter performs the rate change in two steps. First the M-point IFFT performs a $1 : M$ rate change as discussed earlier regarding the polyphase channelizing receiver. The second rate change occurs in the output commutator of $1 : MC_s$. Thus we note that for every M samples at the input provides us MC_s samples at the output. This provides us

with the rate conversion needed to space the channels by C_s . Like the Receiver II this is done internally using a buffer management system.

The buffer in the transmitter performs a circular shift to compensate for discontinuities that are due to the rate change in the commutators. The buffer rotations are shown in Figure 2.18 for three cycles. To understand this process, consider our application study where we need

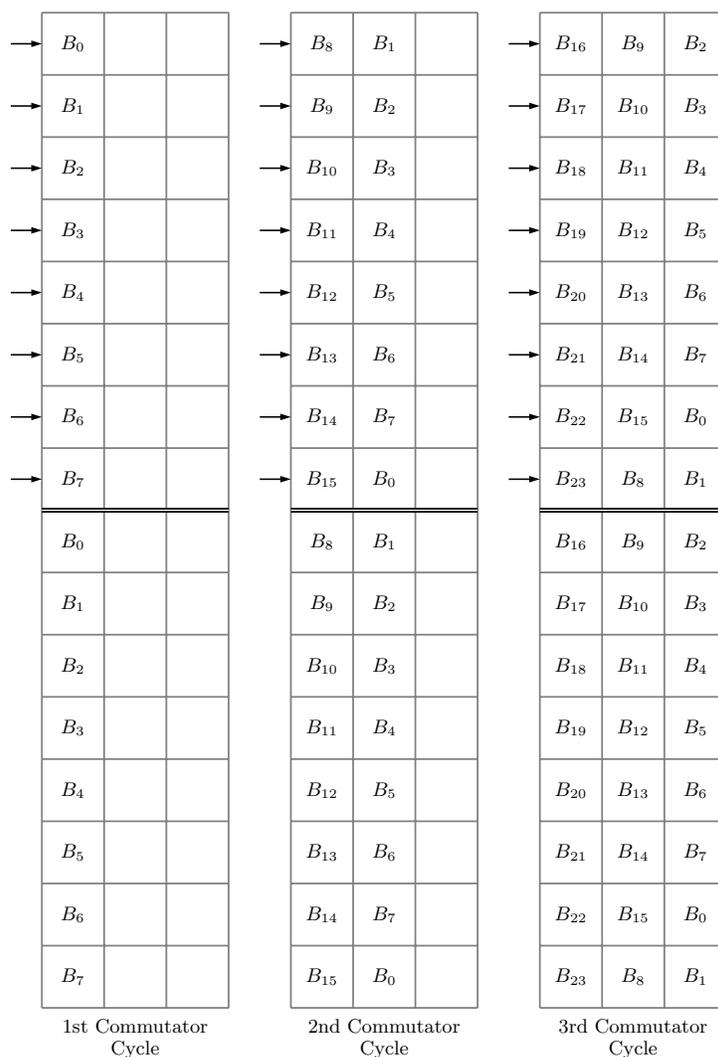


Figure 2.18: Circular Buffer States for Transmitter II

to upsample by a rate of 5 : 4 prior to channelization due to the excess bandwidth parameter α . That means for every 256 samples we clock in, we need to output 320 samples. This is

accomplished by using an $M = 256$ IFFT which corresponds to the 256 available channels. The data enters a buffer of length $2M$ or 512 in our example. To fill the buffer, data is replicated such that if i is the index of the buffer, $b_{i+M} = b_i$. As data is clocked into the circular buffer at rate R_s , the buffer is shifted by 64 samples and the output of the circular buffer clocks out 320 samples.

The circular shift in the buffer is similar to a phasor rotation [6] with the rotation proportional to the rate change over the number of channels. In our example, we have a factor of $64/256$ or $1/4$, so the phasor rotation correction required at the input of the IFFT transmitter is

$$\Theta_i = C_i \bmod 4 \quad \text{for } C_i = 0, 1, \dots, C - 1 \quad (2.12)$$

Since we are using OFDM for modulation, we may perform this phasor rotation as a circular shift in the frequency domain. If we circularly shift our OFDM subcarriers prior to the OFDM IFFT, we may eliminate the need for the multiplications in Figure 2.17.

2.2.3 Comparison of Transmitters

The amount of total addition and multiplication computations per second is shown in Figure 2.19. Again the Transmitter II is independent of the number of channels considering our assumption that buffer management is free. While the CIC Interpolator transmitter is advantageous for a small number of channels, it does not scale well and if more than 8 or 10 transmitters will be used, the polyphase channelizer is more efficient. This is because there is very little reuse in the heterodyning process of the CIC filter.

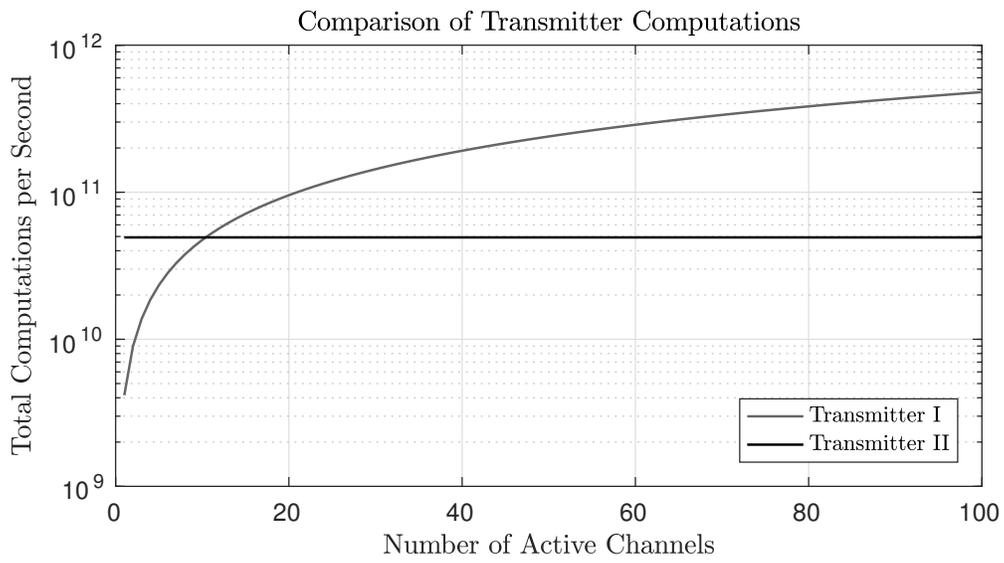


Figure 2.19: Comparison of Transmitter Computational Costs per Number of Active Channel

Chapter 3

Frame, Frequency, and Equalizer Estimation

A challenge with using OFDM is that we must identify when the start of an OFDM symbol occurs. If we make an assumption that the channel timing is changing slowly, one method for obtaining OFDM synchronization is to introduce a training sequence [14] of OFDM symbols. This method requires that some number of OFDM symbols be dedicated for synchronization purposes which increases the overhead of the channel. How frequently these synchronization symbols must be sent depends on how fast the channel is assumed to be changing. These synchronization blocks carry no data and add to the overhead of the channel.

Another method for symbol synchronization is available which takes advantage of the cyclic prefix [15–17] used in OFDM modulation. The method is implemented as a running correlator that produces a *maximum likelihood* estimation of the start of an OFDM symbol. This approach provides a benefit in that it does not add any overhead symbols and is capable of providing an estimate for every OFDM symbol. Figure 3.1 shows the implementation of this estimator.

The delay N in Figure 3.2 corresponds to the length of the OFDM symbol minus the cyclic

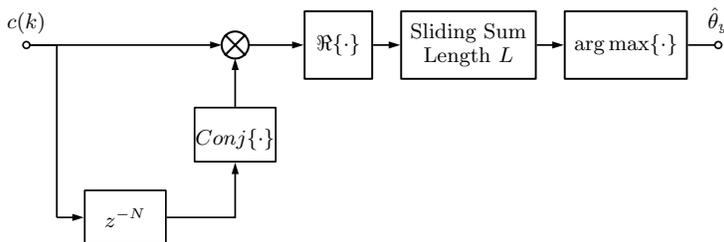


Figure 3.1: OFDM Symbol Synchronization Using Correlator

prefix so for an FFT size of 128 and a sampling rate of $2R_s$, then N would be 256. At instances when the current sample $r(k)$ are nearly identical to that of $r(k - N)$, as is the case towards the end of an OFDM symbol, the output of the multiplier will have a strong real component due to the conjugation. The parameter L is the length of the cyclic prefix, so when sequential correlated samples occur, the output grows and generates a peak. It is at this time we may identify that we are at the end of an OFDM symbol. If one wishes to reduce the variance of the estimator, multiple OFDM symbols may be averaged together.

Methods described in [15,16] build on this principle to generate an estimator that produces both timing and frequency offset estimations. The resulting block diagram of the joint *maximum likelihood* estimator is illustrated in Figure 3.2. The correlator section resides on the top left section while the bottom left section computes energy. The sliding sums serve the same purpose as the previous diagram and the two leaky integrators provide frame averaging; the length of M here is the length of the OFDM symbol including the cyclic prefix. The leaky integrators provide an efficient way to implement smoothing as the memory is reduced compared to a moving sum or other FIR low pass implementation. The leaky integrator parameter α controls the amount of smoothing and must be between 0 and 1 for stability. After frame averaging is parameter extraction which provides the estimates of the frame

time $\hat{\theta}$ and frequency offset $\hat{\epsilon}$. The parameter ρ in the diagram is defined to be

$$\rho = \frac{SNR}{SNR + 1} \quad (3.1)$$

$$SNR = \frac{\sigma_s^2}{\sigma_n^2} \quad (3.2)$$

so if we may assume a large signal-to-noise ratio, we may approximate with $\rho = 1$.

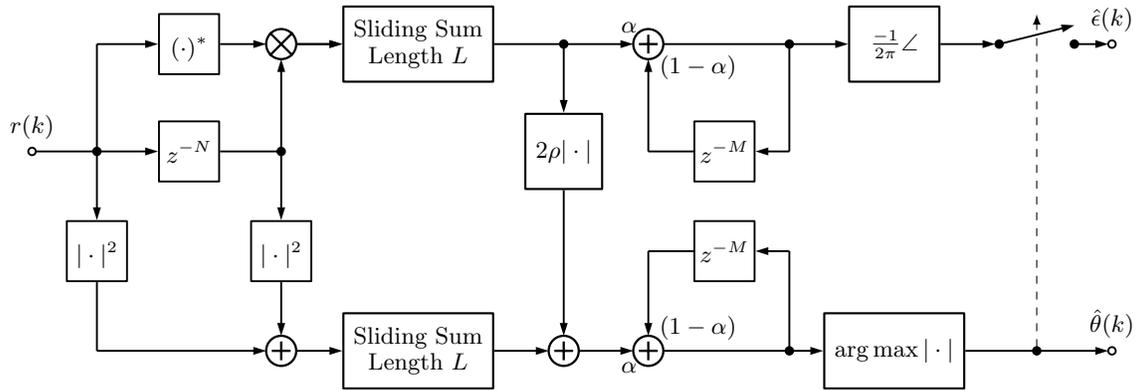


Figure 3.2: OFDM Symbol and Frequency Synchronization Using Correlator

A sample output of the estimates $\hat{\theta}$ and $\hat{\epsilon}$ are shown in Figure 3.3. This capture begins at the estimator's initialization and runs through 13 OFDM symbols with a leaky integrator value $\alpha = 0.25$. Due to the smoothing performed by the leaky integrator, the estimates become more reliable over time. Note that when the timing estimator $\hat{\theta}$ is at a maximum, the frequency offset $\hat{\epsilon}$ is roughly constant. This is the time when $\hat{\epsilon}$ should be sampled as there is low coupling between the two estimators. The angle between the correlated cyclic prefix data should be zero under no frequency offset conditions. If there is a frequency offset the correlator will produce a correction value whose mean is between $\pm 0.5f_\Delta$ where f_Δ is the tone spacing.

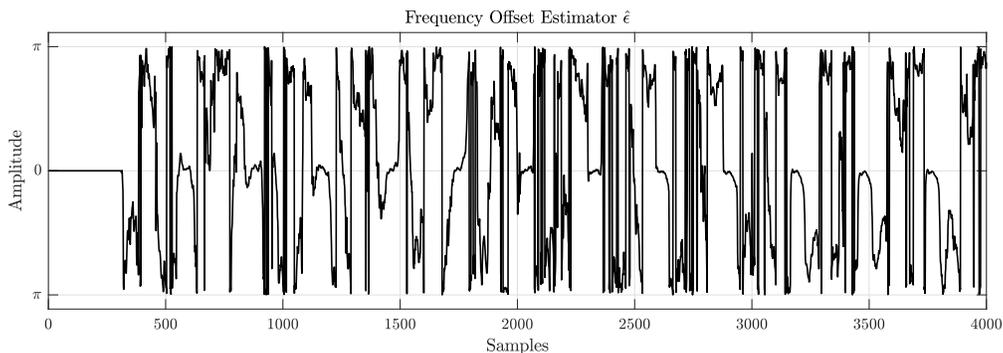
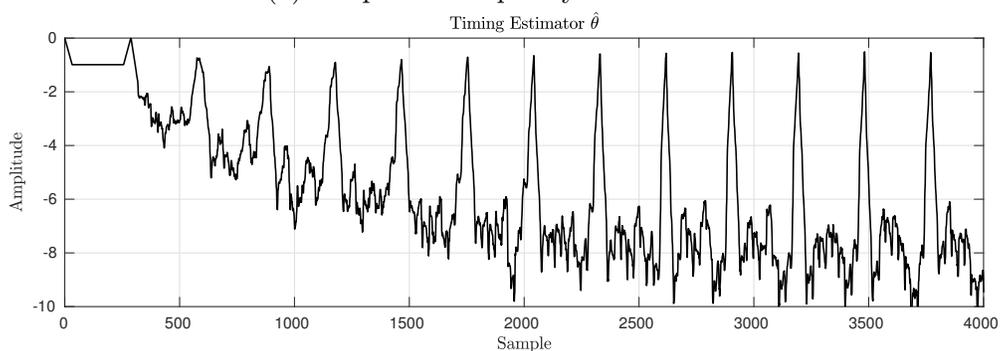
(a) Output of Frequency Estimator $\hat{\epsilon}$ (b) Output of Timing Estimator $\hat{\theta}$

Figure 3.3: Outputs of Time and Frequency Estimator, Ideal Channel

3.1 Frequency Estimation

A limitation of synchronizing based on correlation of the cyclic prefix is that it provides a frequency estimate only up to half of a tone spacing f_{Δ} . This could be problematic in our application as we assume supersonic aircraft. The detailed study of this matter is left for future work, however a couple of proposals are worth mentioning. The work described in [14] provides a method capable of correcting for frequency offsets larger than a half-tone spacing while simultaneously determining symbol timing. The caveat is that this method requires using two OFDM symbols to perform the estimation via a training sequence. This process introduces additional overhead and must be repeated faster than the channel is varying.

Another possible option would be to perform additional processing on Figure 3.2. We note

that the estimator will produce an estimate that will coerce the frequency offset to an integer value of the tone spacing. Once the frequency offset is determined, a tracking algorithm may be implemented to unwrap the frequency estimate over time, correcting for the abrupt 2π transitions at the output of the frequency estimator. As a new estimate of $\hat{\epsilon}$ is provided every OFDM symbol, detection of the frequency offset passing over boundaries of nf_Δ should be possible as long as the initial frequency offset is acquired.

The issue then becomes how to determine initial acquisition of the frequency offset when the offset is larger than half of a tone spacing. This is a more challenging task, but knowing that the estimator coerces the offset to nf_Δ increments, where n is an integer, the demodulator could apply different offsets such as $-f_\Delta, f_\Delta, 2f_\Delta, -2f_\Delta$ until the frequency offset is acquired.

3.2 Timing Estimation

Using the information from the estimate $\hat{\theta}$, we must establish which samples to perform OFDM demodulation on. The signal at this point must undergo downsampling to bring the rate from $2R_s$ to R_s while removing the cyclic prefix extension at the same time. There are a couple of considerations when determining which samples to discard.

The first is that in non-ideal channels the timing estimator might exhibit high noise or multiple peaks. Consider a two-path channel model where the two paths have similar amplitudes separated by some delay. The output of $\hat{\theta}$ will show a broader peak than the single-ray channel, as shown in Figure 3.4. In such a scenario, the estimator may experience increase jitter from symbol to symbol. It will be expected that the channel equalization section of the OFDM demodulator will need correct for reasonable errors in time estimation. As the sampling offset should not vary much from OFDM symbol to symbol, some sort of algorithm such as a median filter is proposed. The median filter is a nonlinear filter which handles impulse noise well.

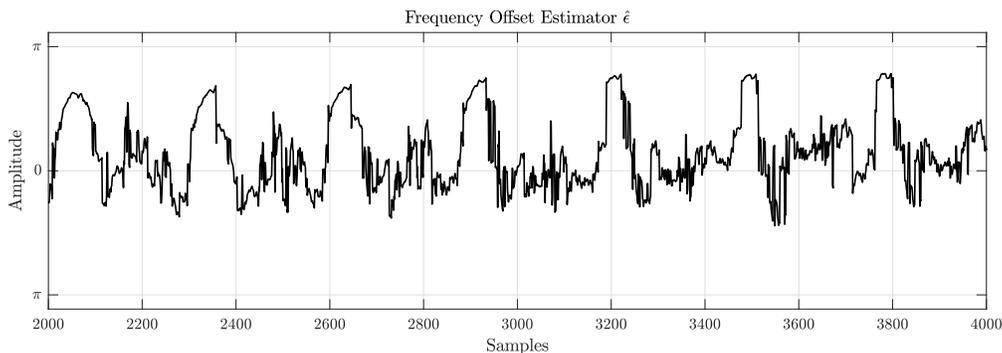
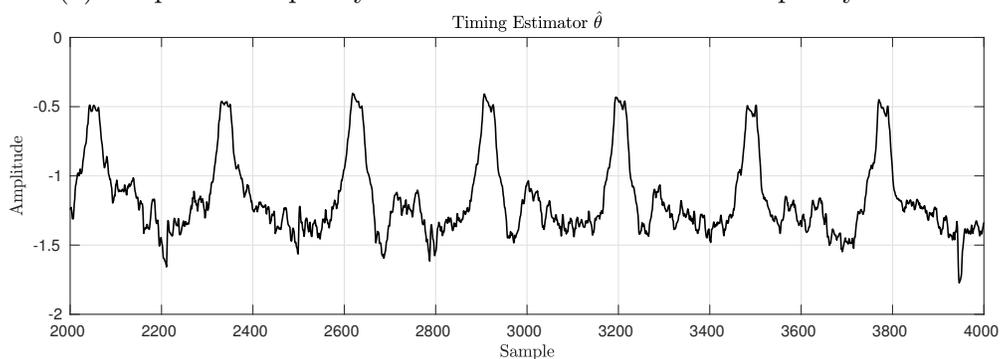
(a) Output of Frequency Estimator $\hat{\epsilon}$ with 3000 kHz Frequency Offset(b) Output of Timing Estimator $\hat{\theta}$ with 10 μs Time Spread

Figure 3.4: Outputs of Time and Frequency Estimator, Non-Ideal Channel

A second consideration is that ideal timing is very impractical in asynchronous multi-user OFDM systems such as our application. Consider the architecture of the polyphase channelizer receiver used in this system. At the front of the channelizer is a commutator which functions as a decimator. The channelizer is free-running, meaning at no point is its phase synchronized with an individual user. Attempting to synchronize the channelizer commutation phase with any given user is ineffective as multiple users are asynchronous. The conclusion is that we will always have some amount of ISI and it will be required for the equalizer to perform compensation. In order for the equalizer to perform this task, we must select the best time to parse the OFDM frame based on the timing estimate.

3.3 Channel Equalization

The purpose of the channel equalizer is to correct for both multipath channel distortion and ISI due to commutator timing. In OFDM this is accomplished by sacrificing some number of subcarriers to use as pilot symbols which are of known value. After the receiver FFT, we measure the magnitude and phase of these pilot tones, interpolate these points, and apply the estimation to our OFDM frame assuming that

$$\tilde{y}(t) = \tilde{x}(t) \otimes \tilde{p}(t) \quad (3.3)$$

$$\tilde{Y}[m] = \tilde{X}[m]\tilde{P}[m] \quad (3.4)$$

where $\tilde{x}(t)$ is our sent data, $\tilde{p}(t)$ is our channel impulse response, and $\tilde{y}(t)$ is our received data. The use of the cyclic prefix allows (3.4) to be true as long as the cyclic prefix duration is longer than the delay spread of the channel.

To illustrate the relation between the timing estimator $\hat{\theta}$ and when we should perform frame parsing, it is helpful to look at a toy case. Let us assume that the output of the OFDM modulator's IFFT is $x = [1, 0, 0, \dots, 0]$ prior to the addition of the cyclic prefix. Thus the first sample of $OFDM_n$ is one with all other samples zeros, as illustrated in Figure 3.5. The purpose of sending this impulse signal is that it allows us to follow the impulse through transmitter interpolation channelization, over a channel, and through receiver channelization so we may evaluate end-to-end timing. For ideal timing, the impulse should be recovered at the first OFDM time sample without ISI extending to other time samples.

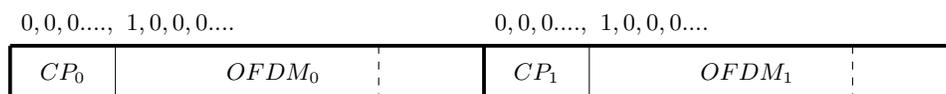


Figure 3.5: OFDM Demodulation Timing Diagram

The output of the receiver channelizer is shown in Figure 3.6 along with the output of the estimator superimposed (dashed line). From the timing estimator we may identify the start and stop of the OFDM symbols and cyclic prefix regions as shown. With this information, the frame parser may select the start and stop of the OFDM frame and downsample, extracting the cyclic prefix in the process. The results are shown in Figure 3.7 and the data matches our transmission.

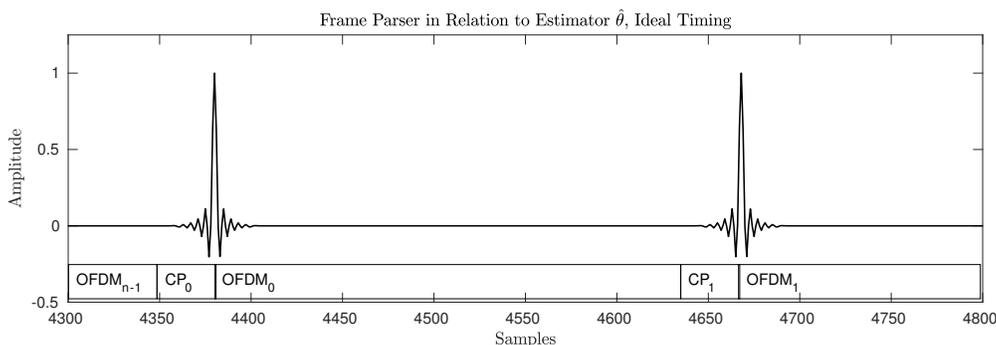


Figure 3.6: Output of Channelizer with Estimate $\hat{\theta}$ and OFDM Symbol Markings, Ideal Timing

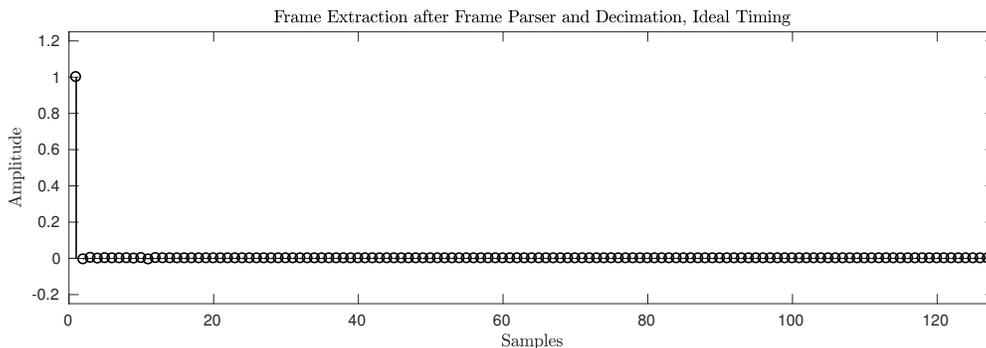


Figure 3.7: Downsampled Output of Frame Parser, Ideal Timing

Next we need to address the consequences of non-ideal timing. We are only capable of producing timing corrections with resolution $R_s/2$ so smaller timing offsets caused by either propagation delay or the decimation in the commutator must be corrected by the equalizer. Figure 3.8 shows the output of the channelizer with OFDM symbol boundaries superim-

3.3. CHANNEL EQUALIZATION

posed. A timing offset of 80 samples was introduced at the input of the commutator, which corresponds to a timing error of $R_s/4$.

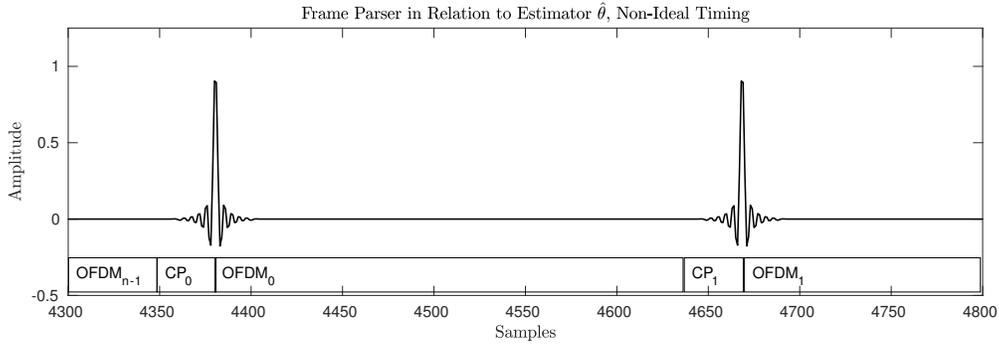


Figure 3.8: Output of Channelizer Aligned with Estimate $\hat{\theta}$, Non-Ideal Timing with 80-Sample Commutation Offset

Figure 3.9 provides a view of the output of the frame parser if we parsed the frame in the same way that we did previously. The effects of timing mismatch are more obvious in this view as we observe ISI as a result of imperfect timing. We also observe that the impulse response has been truncated as the result of cyclic prefix removal. The truncation of the pulse makes the response non-causal so we are unable to satisfy the cyclic convolution requirement in order to apply (1.4). This provides additional complication to the equalizer polynomial curve fit.

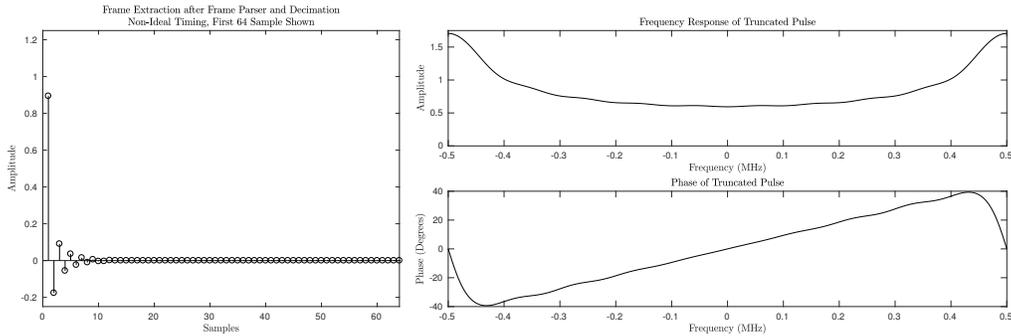


Figure 3.9: Downsampling Output of Frame Parser, Non-Ideal Timing with 80-Sample Commutator Offset, Truncated Pulse, and Frequency Response

To overcome this issue we must select a different starting time to parse the OFDM symbol.

The solution is to parse the OFDM symbol such that the entire pulse is captured. This means selecting a sample in the *middle* of CP_0 as our first sample. Thus we are extracting part of both the beginning and the end of the OFDM symbol. As the cyclic prefix is duplicate data from the last L samples, the result is no loss in data; however the data undergoes a phase shift due to the time shift of sampling before the start of $OFDM_0$. The pulse, frequency response, and phase is shown in Figure 3.10.

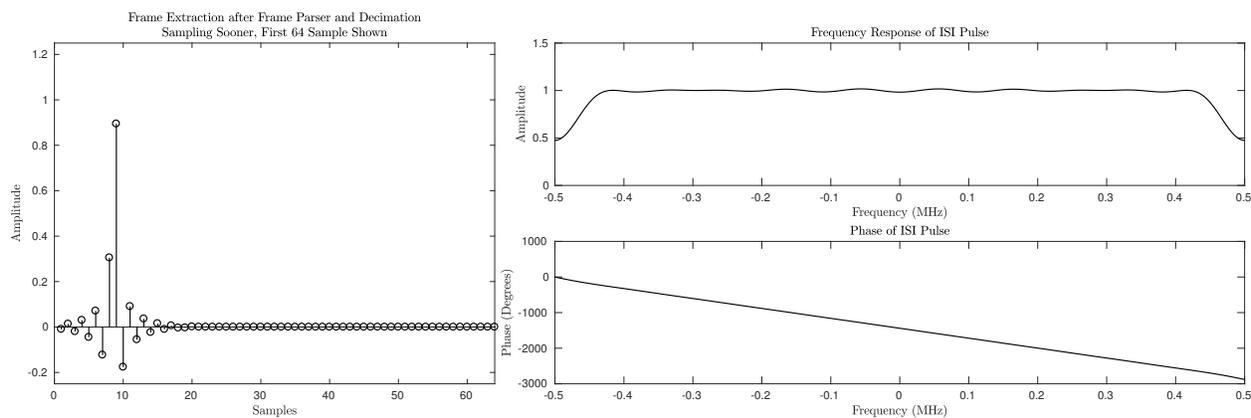


Figure 3.10: Frame Parser Output with 80-Sample Commutator Offset Parsing in the Middle of the OFDM Symbol

The large phase shift introduced may cause problems in the channel equalization process, as it may not be possible to properly unwrap the phase of the pilots such that no 2π -discontinuities exist. The solution is to circularly rotate the samples since we are using cyclic convolution. Performing this step allows us to perform ISI correction using the pilot tones and (1.4) and rotating the samples produces Figure 3.11

We notice that at the center of the band and extending outwards, the frequency response in Figure 3.11 has constant amplitude and a linear phase. At the band edges however, ISI exhibits non-linear phase and attenuation. The severity of the non-linear phase characteristics depends on the timing error. At the most extreme case, the band edges become nulls. This explains one of the reasons for using guard bands in the OFDM subcarrier mapping.

3.3. CHANNEL EQUALIZATION

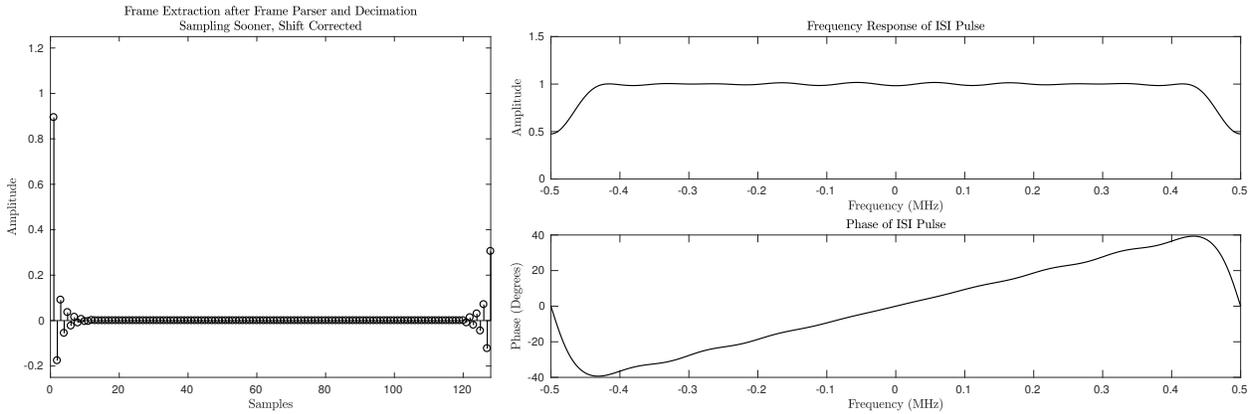


Figure 3.11: Frame Parser Output with 80-Sample Commutator Offset Parsing in the Middle of the Cyclic Prefix, and Samples Shifted

3.3.1 Polynomial Curve Fitting

The method selected to perform channel equalization is polynomial curve fitting¹. Polynomial curve fitting enables equalization that will be performed on the magnitude and phase independently. As shown in Figures 3.10 and 3.11, jitters in the timing estimator $\hat{\theta}$ will result in a linear phase rotation versus frequency. Thus a polynomial curve fit is an adequate choice for correcting timing errors. However, a primary purpose of the equalizer is to correct for channel effects that stem from multi-path and scattering. The frequency response of a multiple path transmission can be expressed as [10]

$$\tilde{P}(f) = \sum_i a_i e^{-j2\pi f t_i} \quad (3.5)$$

where a_i are the attenuation or gain on each path and t_i is the arrival time over each path. The effects of multipath and scattering produces a periodic frequency with peaks and nulls throughout the spectrum. In order for the polynomial to fit to such as response, the polynomial order must be adequate however an order that is too high will overfit and

¹MATLAB command `polyfit` is used

introduce additional error.

The general polynomial model is [12]

$$\mu_{Y|x} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p \quad (3.6)$$

which is to say the mean of the response variable Y a function of a predictor variable X , β is the model parameters, and p is the order of the polynomial. To accommodate noise or model error the matrix formula for this may be expressed as [12]

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_1^1 + \beta_2 x_1^2 + \cdots + \beta_p x_1^p + E_1 \\ Y_2 &= \beta_0 + \beta_1 x_2^1 + \beta_2 x_2^2 + \cdots + \beta_p x_2^p + E_2 \\ &\vdots \\ Y_n &= \beta_0 + \beta_1 x_n^1 + \beta_2 x_n^2 + \cdots + \beta_p x_n^p + E_n \end{aligned} \quad (3.7)$$

We may then express the equation as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2^1 & x_2^2 & \cdots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & x_n^2 & \cdots & x_n^p \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} E_0 \\ E_1 \\ \vdots \\ E_p \end{bmatrix} \quad (3.8)$$

where \mathbf{E} is an modeling error vector and solve the multiple regression algorithm using a least mean squares technique by using the form

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{E} \quad (3.9)$$

by taking the pseudo inverse such that

$$\hat{\beta} = (\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H \mathbf{Y} \quad (3.10)$$

3.3. CHANNEL EQUALIZATION

The Polynomial order impacts the ability to properly apply channel correction. The order must be sufficiently high such that a curve may be fitted without too much error but increasing the order increases the demand of hardware computations and may overfit

Another consideration when using a polynomial curve fit, or any curve fit, is that noise will negatively impact the estimation process as the curve fit will perturb the estimate. Assuming the channel is slow fading and there is a method in place to reduce the jitter from timing estimation in the frame parser, a possible method to reduce noise in the polynomial curve fit would to be average multiple pilot tones over time.

Chapter 4

Prototype Filter Design

In Chapter 2 we discussed how a long filter may be implemented in polyphase filter banks. In this chapter we will discuss how to generate this long filter, known as the prototype filter. The general process for designing such long filters is to optimize a shorter filter and then interpolate. The reason for designing a shorter filter and then interpolating is because the computational requirement to run optimization programs on large filters is impractical on most current computing hardware.

It is important to consider the characteristics of the prototype filter as it introduces ISI and has non-zero stopband leakage power. To maximize SNR at the receiver [9], a root-Nyquist pulse shaping filter shall be used with its matched filter being used at the receiver.

We will investigate various filter design methods. As such we will compare them using three metrics: Peak stopband power density \mathcal{S}_P , total out-of-band power P_{OOB} , and inter-symbol-interference, *ISI*. First we formulate the frequency response of a given, N -tap FIR filter with real coefficients h to be

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h_n e^{-j\omega n} \quad (4.1)$$

Since the FIR filter is real, the FIR filter is symmetric in the frequency response $|H(e^{j\omega})|$. Thus we will focus on the filter's frequency response between 0 and π and will define three

points, ω_p , ω_t , and ω_s . The frequency at the boundary between passband and transition region is denoted ω_p , the half-power frequency is ω_t , and ω_s is the frequency at the start of the stopband region. Figure 4.1 shows these frequencies.

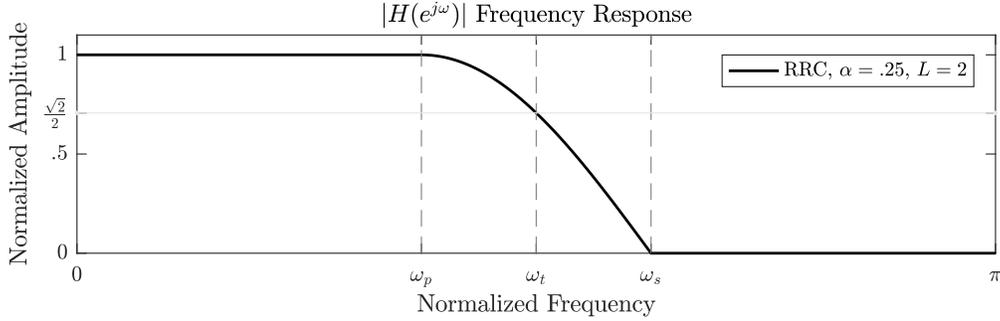


Figure 4.1: Frequency Response of Ideal RRC Filter

The peak stopband power density \mathcal{S}_P is defined as

$$\mathcal{S}_P = \max_{\omega \geq \omega_s} \left[\frac{|H(e^{j\omega})|^2}{|H(0)|^2} \right] \quad (4.2)$$

which normalizes for passband gain. The total out-of-band power is defined as

$$P_{OOB} = \frac{1}{2\pi} \int_{\omega_s}^{\pi} |H(e^{j\omega})|^2 d\omega \quad (4.3)$$

All of the filters considered are root-Nyquist, meaning we assume the filter at the receiver is the time-reversed transmitted filter: $h_R[n] = h_T[-n]$. Inter-Symbol-Interference (*ISI*) is measured by the ratio of the mean-square power of sample at perfect timing to the mean-square power of all other downsampled samples. If we define

$$p[n] = h[n] * h[-n] \quad (4.4)$$

and index n such that the peak of p occurs at $n = 0$, then we may downsample $p[n]$ by L

$$p_d[n] = p[nL] \quad (4.5)$$

and obtain ISI by

$$ISI = \frac{|p_d[0]|^2}{\sum_{n \neq 0} |p_d[n]|^2}. \quad (4.6)$$

These three metrics will be used to quantify filter performance. Our “short” filters will have length 51 or 53 taps, eventually producing a “long” filter with 4240 taps after interpolation. The short filter oversampling factor will be $L = 4$ and a excess bandwidth factor will be $\alpha = 0.25$ for all test cases.

4.1 Truncated Root Raised Cosine Filter

A common root-Nyquist filter candidate is the Root-Raised-Cosine (RRC) filter. The continuous-time impulse response of the RRC filter is defined as [4]

$$h(t) = \frac{1}{\sqrt{T_s}} \frac{4\alpha \frac{t}{T_s} \cos\left(\pi \frac{t}{T_s}(1 + \alpha)\right) + \sin\left(\pi \frac{t}{T_s}(1 - \alpha)\right)}{\pi \frac{t}{T_s} \left(1 - (4\alpha \frac{t}{T_s})^2\right)} \quad (4.7)$$

where α is defined as the roll-off factor and T_s is $1/R_s$. For $t = 0$ and $t = \pm T_s/(4\alpha)$, equation (4.7) is undefined. Here in continuous time we can take L’Hospital’s rule to find the indeterminates as

$$h(t) = \begin{cases} \frac{1}{\sqrt{T_s}} \left(1 - \alpha + \frac{4\alpha}{\pi}\right), & t = 0 \\ \frac{\alpha}{\sqrt{2T_s}} \left[\left(1 + \frac{2}{\pi}\right) \sin\left(\frac{\pi}{4\alpha}\right) + \left(1 - \frac{2}{\pi}\right) \cos\left(\frac{\pi}{4\alpha}\right) \right], & t = \pm \frac{T_s}{4\alpha} \end{cases}$$

4.1. TRUNCATED ROOT RAISED COSINE FILTER

The frequency response of the infinite length RRC pulse as shown in Figure 4.1 is

$$H(f) = \begin{cases} \frac{T_s}{\sqrt{T_s}} & 0 \leq |\omega| < \frac{\pi}{T_s}(1 - \alpha) \\ \frac{T_s}{\sqrt{2T_s}} \sqrt{1 + \cos \left[\frac{\pi T_s}{2\alpha} \left(\frac{|\omega|}{\pi} - \frac{1-\alpha}{T_s} \right) \right]} & \frac{\pi}{T_s}(1 - \alpha) \leq |\omega| < \frac{\pi}{T_s}(1 + \alpha) \\ 0 & |\omega| \geq \frac{\pi}{T_s}(1 + \alpha) \end{cases} \quad (4.8)$$

We will first quantify the performance of the truncated RRC filter which is done by sampling at $1/(LT_s)$ and then truncate¹, as shown in Figure 4.2. The truncated RRC filter suffers a penalty compared to the infinite length RRC filter. The transition extends well beyond the desired ω_s frequency and as such, \mathcal{S}_P and P_{OOB} are large. The subsequent sections will discuss alternate designs that outperform the truncated RRC filter.

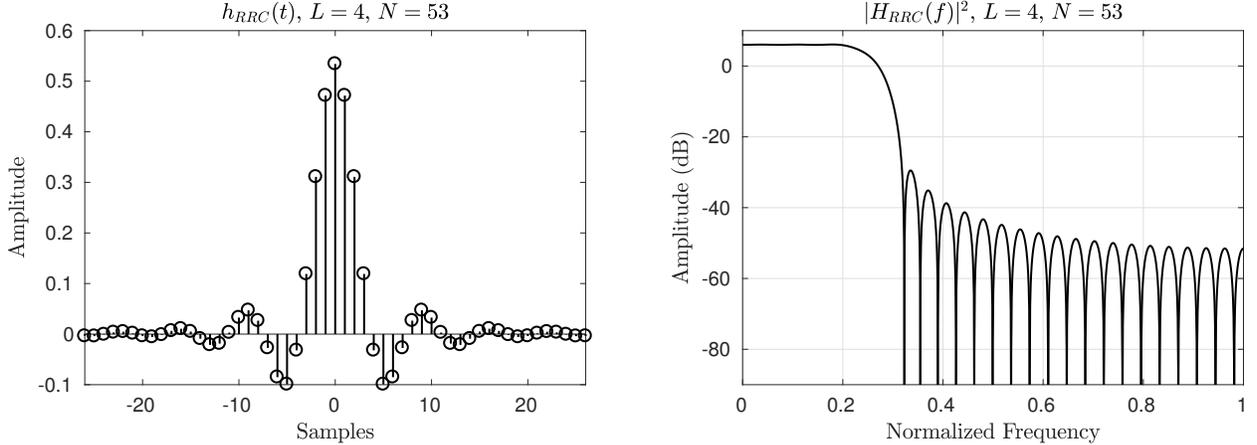


Figure 4.2: Truncated Root-Raised Cosine Filter
 $ISI = 40.4$ dB, $\mathcal{S}_P = -26.4$ dB, $P_{OOB} = -45.4$ dBW

¹This is what MATLAB does using `rcosdesign` or `firrcos`

4.2 Root-Nyquist Filter Design

The first set of filters that will be designed are root-Nyquist filters. These filters will serve as better alternates to the truncated RRC filter for pulse shaping. This section will study generation of short filters that are short enough to be suitable for solving in optimization models. These short filters may then be interpolated to generate the long filter required to use in the polyphase filter banks.

4.2.1 Symmetric FIR Filter Design

The first filter design we will target is a symmetric filter found using a linear program. This filter design is similar to the Wasserman [18] approach to filter design with the exception that there is no CIC compensation. A general linear program² is specified by

$$\min_{\underline{x}} f^T \underline{x} \text{ such that } \begin{cases} A\underline{x} \leq \underline{b}, \\ A_{eq}\underline{x} = \underline{b}_{eq}, \\ \underline{lb} \leq \underline{x} \leq \underline{ub} \end{cases} \quad (4.9)$$

where $f^T \underline{x}$ is the objective function for variables \underline{x} , with \underline{x} consisting of the filter coefficients we are trying to obtain along with the weight parameters. Parameters \underline{lb} and \underline{ub} are the lower and upper bounds that \underline{x} is restricted to.

We consider the goal of the linear program to find coefficients that produce a better fit to the RRC frequency response compared to the truncated RRC filter. This procedure does not directly target *ISI* or *P_{OOB}*; rather good filter characteristics are a result of producing a good match to the RRC, and producing a good balance of requirements may require some

²MATLAB function `linprog` is used

4.2. ROOT-NYQUIST FILTER DESIGN

amount of trial and error. To construct this into a linear model, we consider an N odd, symmetric FIR filter. From (4.1), if the filter is symmetric, the frequency response is

$$H(e^{j\omega}) = h_0 + \sum_{n=1}^{(N-1)/2} 2h_n \cos(\omega n) \quad (4.10)$$

We define our target frequency response, $H_{TARG}(e^{j\omega})$ to be the ideal RRC frequency response in (4.8). We set up the problem to be constrained to specified tolerances δ_i which depends on whether the frequency is in the passband, transition band, or stopband. Each region is assigned a weight W_i for each section which is included with \underline{h} to form \underline{x} . Thus we define our linear program as

Minimize $f = W_p\delta_p + W_{t1}\delta_{t1} + W_{t2}\delta_{t2} + W_s\delta_s$ subject to

$$\left. \begin{array}{l} H(e^{j\omega}) - \delta_p \leq H_{TARG}(e^{j\omega}) \\ -H(e^{j\omega}) - \delta_p \leq -H_{TARG}(e^{j\omega}) \end{array} \right\} 0 \leq \omega \leq \omega_p$$

$$\left. \begin{array}{l} H(e^{j\omega}) - \delta_{t1} \leq H_{TARG}(e^{j\omega}) \\ -H(e^{j\omega}) - 0 \leq -H_{TARG}(e^{j\omega}) \end{array} \right\} \omega_p < \omega \leq \omega_t \quad (4.11)$$

$$\left. \begin{array}{l} H(e^{j\omega}) - 0 \leq H_{TARG}(e^{j\omega}) \\ -H(e^{j\omega}) - \delta_{t2} \leq -H_{TARG}(e^{j\omega}) \end{array} \right\} \omega_t < \omega \leq \omega_s$$

$$\left. \begin{array}{l} H(e^{j\omega}) - \delta_s \leq H_{TARG}(e^{j\omega}) \\ -H(e^{j\omega}) - 0 \leq -H_{TARG}(e^{j\omega}) \end{array} \right\} \omega_s < \omega \leq \pi$$

$$\underline{x} = [\delta_p, \delta_{t1}, \delta_{t2}, \delta_s, h_0, h_1, \dots, h_{\frac{N-1}{2}}] \quad (4.12)$$

The tolerances δ_i provide a hard limitation to the permissible band ripple in each interval.

While it is an effective method for limiting passband ripple or stopband attenuation, if the δ_i 's are too small, the linear program will fail to converge to a solution. Relaxing these parameters will allow the linear program to converge to a solution, however depending on the W_i 's the solution may not be desirable, leading to a process of trial and error. Since ISI and P_{OOB} are not directly constrained, we may modify the passband and transition W_i and δ_i to improve on ISI. We observe that the ideal RRC filter will pass through amplitude $\frac{\sqrt{2}}{2}$ at ω_t so we force the linear program to produce a frequency that also passes through the -3 dB point at ω_t by allowing only a positive error between ω_s and ω_t and only a negative error between ω_t and ω_s . The only parameter that is directly constrained is \mathcal{S}_P which is done by establishing an upper bound for δ_p . Also required when setting up the linear program is to take sufficient samples in frequency such that the passband and stopband ripples are not under-sampled. A length of approximately $5N$ frequency points is used to provide sufficient resolution.

The results of the linear program are shown in Figure 4.3. The solution provides a stopband with constant sidelobes, however one could modify the objective functions to include more parameters in the stopband to alter the response and generate a filter with decaying sidelobes. Doing this will lower P_{OOB} but may increase \mathcal{S}_P . Compared to the truncated RRC, ISI is similar but provides a reduction in \mathcal{S}_P and P_{OOB} . As this method of filter generation requires some trial and error, the parameters were varied until a satisfactory ISI goal of 40 dB was reached while keeping \mathcal{S}_P small.

4.2.2 Symmetric FIR Filter Design Minimizing Out of Band Power

A significant disadvantage of using the linear program in the previous section is that it only provides results constrained to \mathcal{S}_P and does not target all of our desired parameters directly. In this section we will discuss a targeted filter optimization process to design a symmetric root-Nyquist FIR filter.

4.2. ROOT-NYQUIST FILTER DESIGN

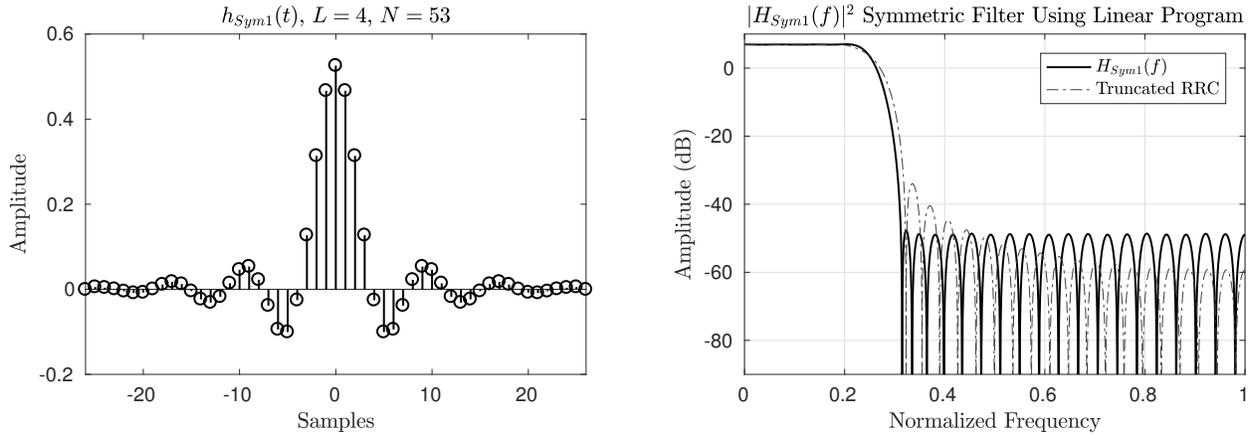


Figure 4.3: Symmetric Filter using Linear Program
 $ISI = 40.7$ dB, $\mathcal{S}_P = -47.4$ dB, $P_{OOB} = -50.0$ dBW
 $W_p = 15, W_{t1} = 1, W_{t2} = 1, W_s = 5, \delta_p = .01, \delta_{t1} = 1, \delta_{t2} = 1, \delta_s = .0075$

In this process we will not attempt to directly target a RRC filter, but we will enforce the filter to be root-Nyquist by constraining ISI as defined in (4.6). Our target will be to minimize P_{OOB} constrained to ISI and \mathcal{S}_P . To accomplish this we consider a nonlinear program³ given by

$$\min_{\underline{h}} f(\underline{h}) \text{ such that } \left\{ \begin{array}{l} c(\underline{h}) \leq 0 \\ c_{eq}(\underline{h}) = 0 \end{array} \right\} \text{ nonlinear} \quad (4.13)$$

$$\left\{ \begin{array}{l} A\underline{h} \leq \underline{b} \\ A_{eq}\underline{h} = \underline{b}_{eq} \end{array} \right\} \text{ linear}$$

$$\underline{lb} \leq \underline{h} \leq \underline{ub}$$

Our objective function, P_{OBB} , therefore becomes the discrete version of (4.10) with $\Delta\omega$ being

³MATLAB's `fmincon` is used

the frequency sample spacing,

$$f(\underline{h}) = \frac{\Delta\omega}{2\pi} \sum_{\omega=\omega_s}^{\pi} \left[h_0 + \sum_{n=1}^{(N-1)/2} 2h_n \cos(\omega n) \right]^2 \quad (4.14)$$

and our constraint equations are defined as

$$c_{eq}(\underline{h}) = p[0] - 1 \quad (4.15)$$

$$c_1(\underline{h}) = \max_{\omega \geq \omega_s} \left[\frac{H(e^{j\omega})^2}{H(0)^2} \right] - \mathcal{S}_{TARG} \quad (4.16)$$

$$c_2(\underline{h}) = -ISI + ISI_{TARG} \quad (4.17)$$

$$-1 \leq \underline{h} \leq 1 \quad (4.18)$$

An initial state \underline{h}_I must be provided and the chosen initial state for our program is the N-tap truncated RRC filter. This program will only produce a result if both the objective function is met and all constraints are satisfied. In Figure 4.4 are the results for this targeted symmetrical filter design. The constraints specified were to have a minimum ISI of -40 dB and \mathcal{S}_P of -46 dB. While a lower \mathcal{S}_P is desired in our application, there is a trade off between \mathcal{S}_P and P_{OBB} and there is a limitation to the performance we can achieve from a symmetric FIR filter.

4.2. ROOT-NYQUIST FILTER DESIGN

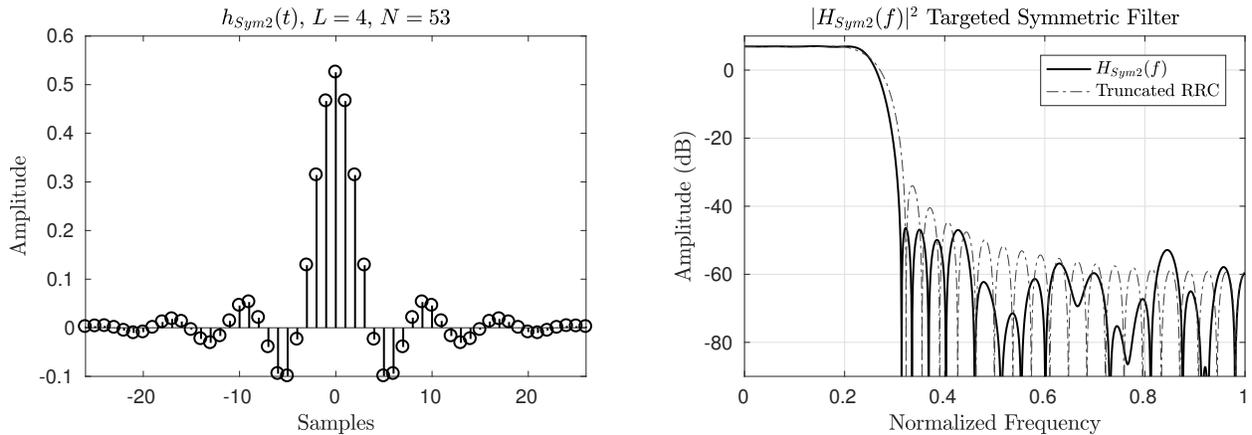


Figure 4.4: Targeted Symmetric Filter using Nonlinear Program
 $ISI = 40.0$ dB, $\mathcal{S}_P = -46.3$ dB, $P_{OOB} = -54.4$ dBW

4.2.3 Asymmetric FIR Filter Design Minimizing Out of Band Power

We will now build on the nonlinear program described in the previous section to design an asymmetrical filter. In our design there is no requirement that the filters be symmetric and generally asymmetric filters can achieve greater performance compared to symmetric filters. To design an asymmetric filter, we consider the nonlinear program described in (4.13) and (4.18) and rebuild our objective function with the generic equation in (4.1).

$$f(h) = \frac{\Delta\omega}{2\pi} \sum_{\omega=\omega_s}^{\pi} \left| \sum_{n=0}^{N-1} h_n e^{-j\omega n} \right|^2 \quad (4.19)$$

We will use the constraint functions $c(\underline{h})$ and $c_{eq}(\underline{h})$ to place constraints on \mathcal{S}_P and ISI . Our constraints and upper/lower bounds are defined as

$$c_{eq}(h) = p[0] - 1 \quad (4.20)$$

$$c_1(h) = \max_{\omega \geq \omega_s} \left[\frac{|H(e^{j\omega})|^2}{|H(0)|^2} \right] - \mathcal{S}_{TARG} \quad (4.21)$$

$$c_2(h) = -ISI + ISI_{TARG} \quad (4.22)$$

$$-1 \leq h \leq 1 \quad (4.23)$$

There are many local minima for the above program. In order for the nonlinear program to produce the best solution, multiple different initial states should be evaluated. A proposal is to use a truncated RRC function that is circularly shifted by a given amount. Using this method, a program can cycle through iterations of various initial states \underline{h}_I and select the solution with the best characteristics:

- whether the nonlinear program converged to a local minimum
- from the options above, select the filter with the lowest P_{OOB} .

For our initial test case of an $N = 53$ -tap FIR filter with oversampling $L = 4$ and roll-off factor of $\alpha = 0.25$, we selected a target ISI of 40 dB and \mathcal{S}_P of -50 dB. We cycled through all possible shifts of the truncated RRC filter as initial states and selected the iteration that produced the minimal P_{OOB} that obeyed our ISI and \mathcal{S}_P constraints. The resulting asymmetrical filter is shown in Figure 4.5.

Of the filters designed in this section, the asymmetric filter design is the only one capable of achieving our required \mathcal{S}_P and ISI constraints for $N = 53$ coefficients. Additionally, unlike the first design, this optimization process targets minimizing P_{OOB} , an important consideration in our application.

4.3. CIC COMPENSATION FILTER

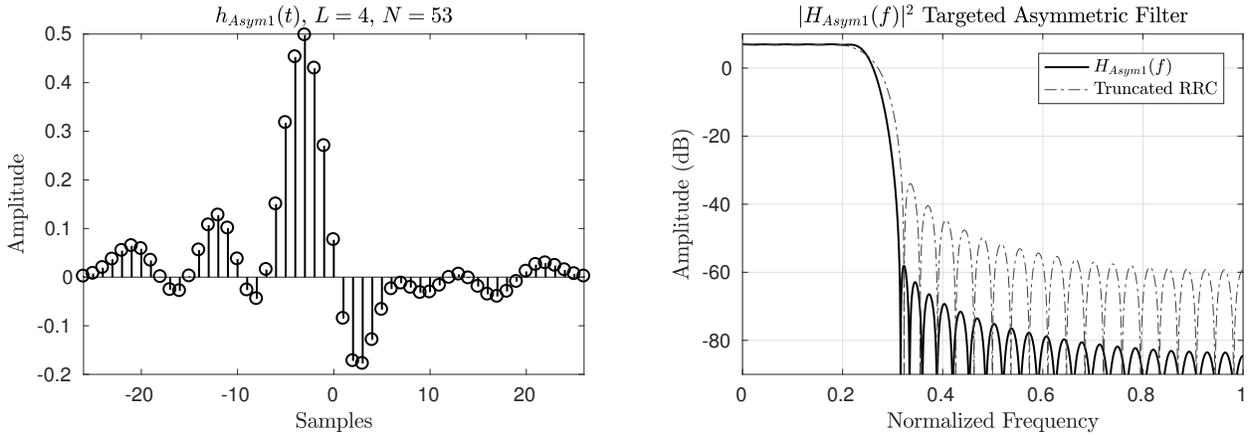


Figure 4.5: Targeted Asymmetric Filter
 $ISI = 40.0$ dB, $\mathcal{S}_P = -52.2$ dB, $P_{OOB} = -69.3$ dBW

4.3 CIC Compensation Filter

In this section we will extend the optimization of the previous three techniques to design a compensated CIC prototype filter. While a CIC interpolator provides an efficient method for interpolation as it requires only additions and subtractions, a drawback is that the CIC interpolator introduces in-band distortion. We can see this in-band distortion by analyzing the transfer function which is defined as [7]

$$H_{CIC}(z) = \left[\frac{1 - z^{-M}}{1 - z^{-1}} \right]^P \quad (4.24)$$

where M is the interpolation factor and P is the number of cascaded comb and integrator sections. The frequency response is therefore [18]:

$$H_{CIC}(e^{j\omega}) = \left(\frac{\sin \frac{\omega M}{2}}{\sin \frac{\omega}{2}} \right)^P \quad (4.25)$$

When upsampling with a CIC interpolator, we must consider the transmitted frequency

response which we shall define as the combined response of the pulse shaping filter H_{PSF} and the CIC interpolator:

$$H_{TX} = H_{PSF}H_{CIC} \quad (4.26)$$

The effect of the CIC on the frequency response is shown in Figure 4.6. The figure shows the distortion introduced to the passband region on the ideal RRC response and the required compensation filter to correct for it. While this compensation is often done by adding an additional FIR filter prior to the CIC interpolation stage, it is more efficient to combine the compensation filter with the pulse shaping filter [18]. This will be our goal when performing filter design.

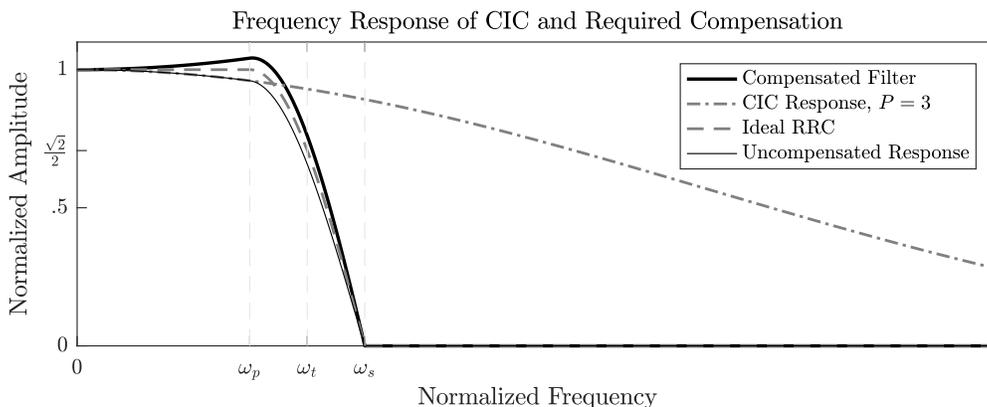


Figure 4.6: Distortion Caused by CIC Interpolation

4.3.1 Symmetric Compensated FIR Filter

The method for designing a symmetric compensated CIC prototype is similar to the process for a non-compensated filter. The objective functions are unchanged from (4.11) with the exception we use H_{TX} as defined in (4.26) in lieu of H . This is the technique used by Wasserman [18]. The objective is still to match to an ideal RRC frequency response while setting parameters on weights and tolerances, thus like the non-compensated case this program does not directly target our parameters rather they are the result of trial and error.

To add compensation to the prototype filter, (4.10) is changed to

$$H(e^{j\omega}) = h_0 + \sum_{n=1}^{(N-1)/2} 2h_n \cos(\omega n) \left(\frac{\sin \frac{\omega}{2}}{\sin \frac{\omega}{2M}} \right)^P M^{-P} \quad (4.27)$$

where M^{-P} is a gain correction in the CIC. The frequency response H_{CIC} of the CIC is scaled as we are interested in the frequency response of the narrow passband region of our pulse shaping filter. Referencing Figure 2.14b, this scaling computes the frequency response from 0 to 2, frequency normalized to the symbol rate. All other frequencies beyond this are aliases of the filter we are optimizing for.

Figure 4.7 presents the results of the linear program and it provides a great illustration of how the linear program targets the combined response. Observe how for the compensated filter the sidelobes are not constant and grow over frequency, however after interpolation the sidelobes are equal. When measuring the performance of these compensated CIC filters, it is important that we consider the combined response and not just the response of the short filter.

4.3.2 Symmetric Compensated CIC Filter Minimizing Out of Band Power

Next we will look at applying the nonlinear model presented in (4.13) to design a compensated CIC filter. Our objective and constraints remain the same, however they must be modified to account for the in-band distortion introduced by CIC interpolation, and applying the CIC correction is more challenging in this case. The goal is to generate a long prototype filter that may be used in our polyphase filter bank. The uncompensated filter design functions by designing a low rate filter and then interpolating based on a process described in [3]. The CIC, however, is not an ideal interpolator so we must account for this

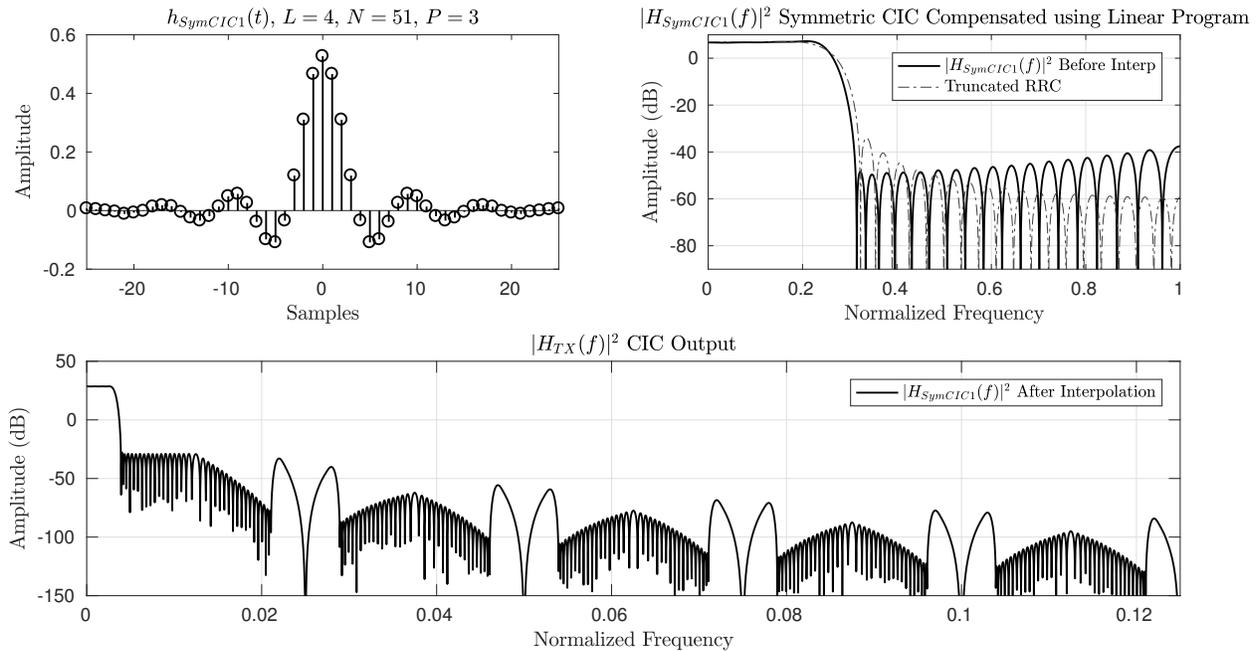


Figure 4.7: Compensated CIC Filter Using Linear Program
 $ISI = 40.2$ dB, $\mathcal{S}_P = -49.0$ dB, $P_{OGB} = -50.8$ dBW

when establishing our objective function and our constraint function. It would be desirable to generate our long filter directly, however generating a total rate change of 320 is computationally difficult. It is more desirable to generate a short filter and then interpolate. This requires characterizing our CIC interpolator in terms of initial rate L (4 in our case) instead of rate LM , or 320. As discussed in the previous section we know the frequency response of the CIC filter, and can easily scale it to fit within our region of interest. However the overall impulse response is required in order to compute the ISI in our constraint equation, and while this is known at the up-sampled rate, we need the equivalent impulse response at lower rate L in order to operate the nonlinear program efficiently.

We must modify our objective function so that it includes the narrowband correction in (4.25).

This modifies our objective equation to

$$f(h) = \sum_{\omega=\omega_s}^{\pi} \left[h_0 + \sum_{n=1}^{(N-1)/2} 2h_n \cos(\omega n) \left(\frac{\sin \frac{\omega}{2}}{\sin \frac{\omega}{2M}} \right)^P \right]^2 M^{-P} \quad (4.28)$$

The constraint equations must also incorporate CIC compensation. Knowing the frequency response of $H_{CIC}(e^{j\omega})$ we can adjust the constraint regarding peak sidelobe. The constraint regarding *ISI* is a bit more challenging since we wish to solve the problem at rate L which is before the CIC. Thus we are interested in obtaining the impulse response of (4.25) so that we may use convolution to obtain the *ISI* of the compensated design.

Considering computational methods are available, one method of doing this is to take the IDFT of (4.25), taking into consideration truncation errors. Then when computing *ISI*, we may apply CIC correction by convolving our h with this downsampled CIC impulse prior to calculating *ISI*. The results of this is shown in Figure 4.8, using the result of from Figure 4.4 as our initial state for the nonlinear program.

4.3.3 Asymmetric Compensation FIR Filter Minimizing Out of Band Power

The same methods may be applied to generate an asymmetrical filter as done before. Here we must use the generic form (4.19) to compute the frequency response. This modifies our objective equation to be

$$f(h) = \frac{\Delta\omega}{2\pi} \sum_{\omega=\omega_s}^{\pi} \left| \sum_{n=0}^{N-1} h_n e^{-j\omega n} \left(\frac{\sin \frac{\omega}{2}}{\sin \frac{\omega}{2M}} \right)^P \right|^2 M^{-P} \quad (4.29)$$

The constraint equations fundamentally remain the same, only adjusting for the changes to

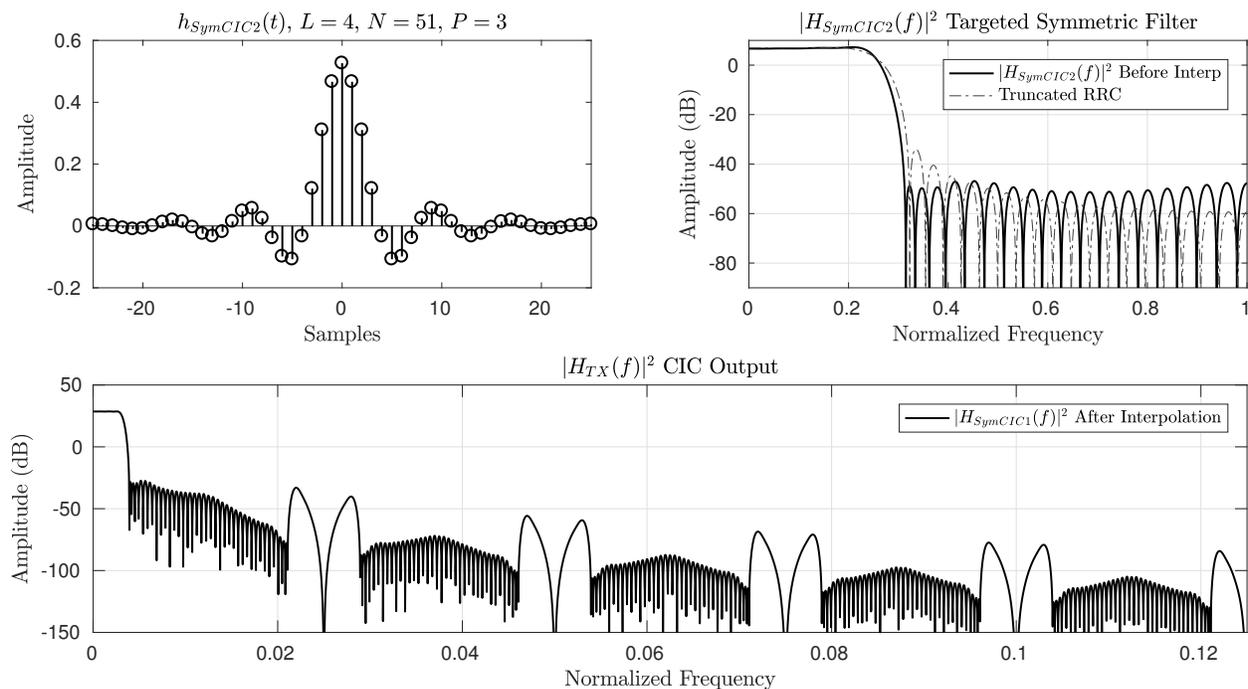


Figure 4.8: Targeted Symmetric Filter using Nonlinear Program
 $ISI = 40.0$ dB, $\mathcal{S}_P = -48.6$ dB, $P_{OOB} = -53.4$ dBW

compute $H(e^{j\omega})$. For our initial state, \underline{h}_I , we use the filter that was found in Figure 4.5. The result of the asymmetric CIC compensated filter is shown in Figure 4.9.

4.4 Filters Used in Simulation

We have presented several filter design options that can be used for our long filters. For simplicity we shall specify one that shall be used for simulation. As we chose to simulate using the Transmit II and Receiver II architecture, we shall use the filter designed in 4.5 as our starting point. To generate our long filter, we take this result and interpolate using methods in [3]⁴.

⁴MATLAB's command `interp` does this

4.4. FILTERS USED IN SIMULATION

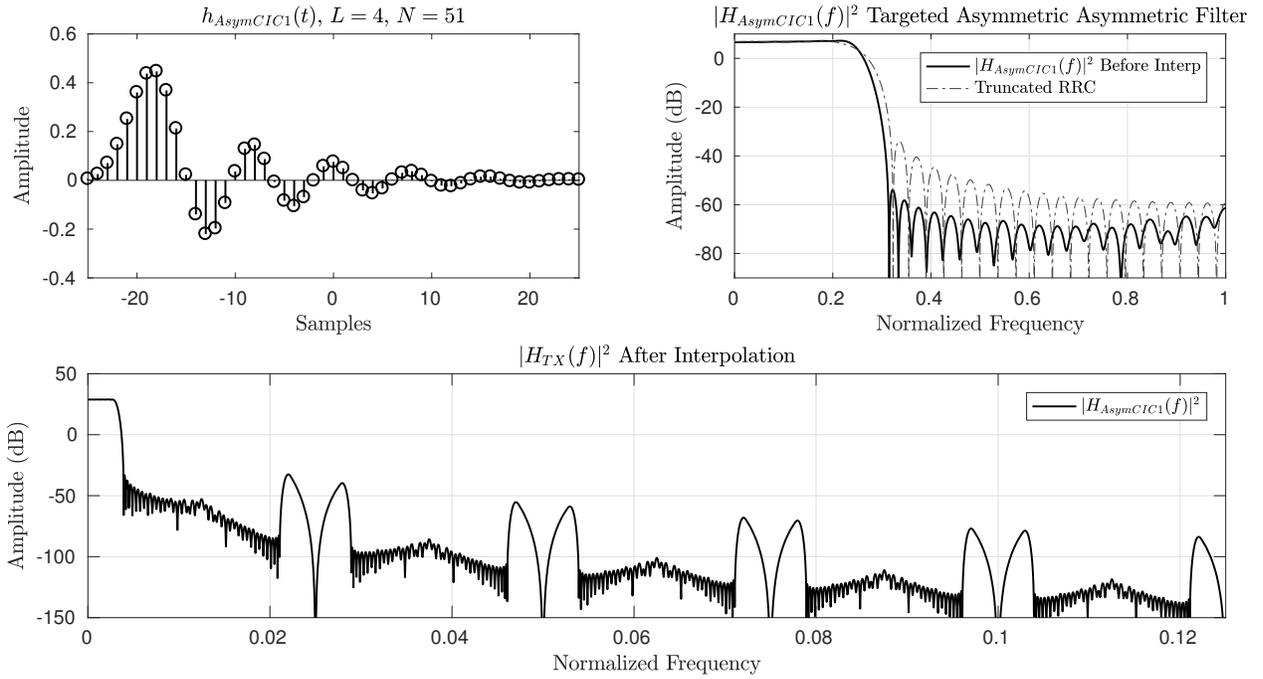


Figure 4.9: Targeted Asymmetric Compensated CIC Filter
 $ISI = 40.0$ dB, $\mathcal{S}_P = -53.3$ dB, $P_{OOB} = -59.6$ dBW

Further we wish to quantize the filter as the filter should be implemented in hardware with fixed-point fractional integer coefficient sizes. We will assume 12-bit fractional integers for the filter coefficients. The process for generating the long prototype filters is as follows: Create an optimized filter at a shorter length, say $L = 4$ times the channel rate, then interpolate using computational tools by rate $M = 80$, and then quantize to 12-bit signed fractional integer words.

The filter used in simulations is shown in Figure 4.10, showing detail of the passband region as well as the extended stopband. The filter was generated from an $N = 53$ short filter, the same one shown in Figure 4.5. After interpolating by a factor of $M = 80$, the total filter size is 4240-taps long. Compared to the unquantized short filter the characteristics are similar. The interpolation process did not alter our parameters and quantization had limited impact, mostly impacting P_{OOB} . As our target ISI and \mathcal{S}_P are met, we reason that using

12-bit word length is sufficient given these requirements.

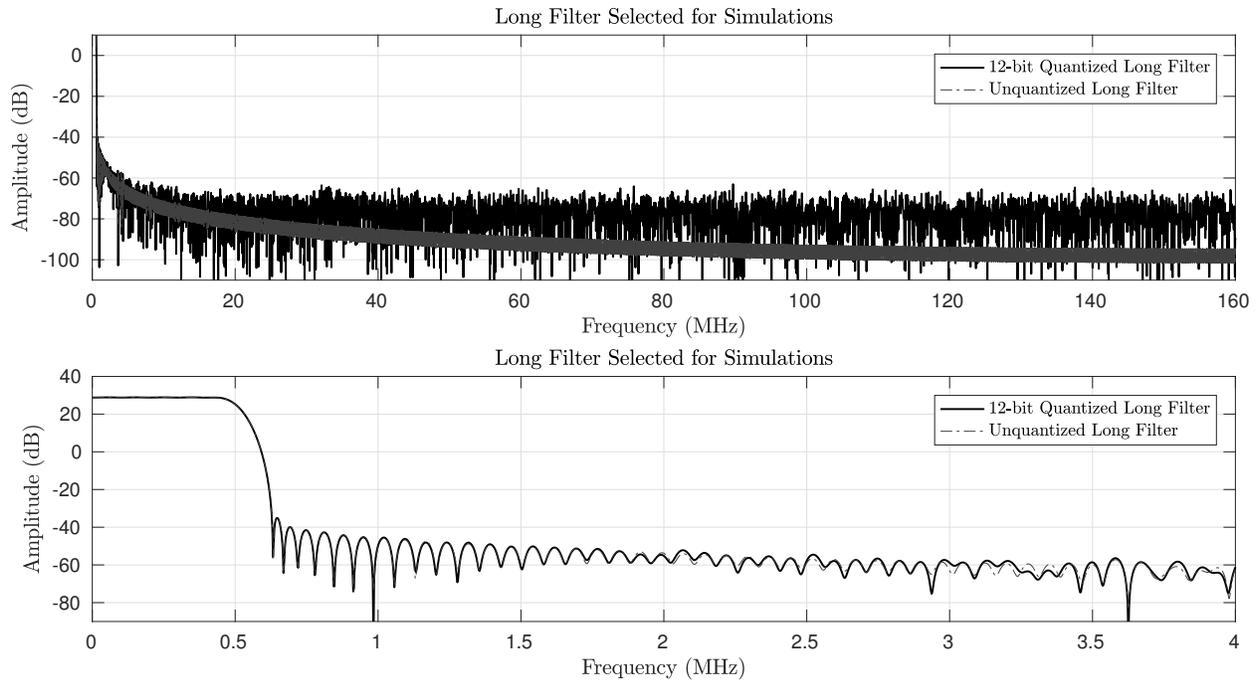


Figure 4.10: Quantized Long Filter Used in Simulations
 $ISI = 40.2$ dB, $\mathcal{S}_P = -53.5$ dB, $P_{OOB} = -61.8$ dBW

Chapter 5

Simulation and Results

In this chapter we will present an end-to-end simulation of our transmitter and receiver. The simulation is implemented in Mathworks MATLAB[®] and when available, MATLAB functions are utilized. The simulation is set up as described in this thesis and block diagrams in Figures 1.1, 1.2, 1.9, 1.10, 2.9, and 2.17 may be referenced. While each of the transmitting and receiving channelizers were implemented and tested, only results from the Transmitter II / Receiver II combination will be presented for simplicity. These architectures were chosen as they are the most efficient when simulating a large number of active channels.

We will consider a two-user scenario to perform Monte Carlo simulations to generate error probability curves. The test setup for all SNR iterations, unless specified otherwise, is as follows:

- For the transmitter, Transmitter II with internal rate conversion is used
- For the receiver, Receiver II with internal rate conversion is used
- The subcarrier assignments follows Figure 1.11
- Unless noted, the channel estimator uses a 6th-order polynomial for curve fitting
- User 1 is assigned all even-numbered channels from 0, 2, 4, ..., 254
- User 2 is assigned all odd-numbered channels from 1, 3, 5, ..., 255

- The OFDM processor does demodulation only on User 1 data
- The parameters in Table 1.1 are followed
- The simulation discards the first several iterations to allow the estimator to settle
- The simulation pseudo random number generator is seeded to a known state on each SNR iteration
- Unless specified otherwise, a leaking integrator α value of 0.25 is used

All channels are transmitting data with additive white Gaussian noise (AWGN) artificially added to the simulation with demodulation enabled for User 1. A reason for using all channels is it reduces overhead when performing a large number of Monte Carlo simulations. Due to the channelizing architecture, it is more efficient to channelize and demodulate 128 channels simultaneously than channelize and demodulate a single channel 128 times.

The users are allocated alternating channels. This is necessary to test inter-channel-interference between asynchronous users. Unless stated otherwise, user 2's channel is an ideal channel in all simulations while User 1's channel will be varied based on test condition. We may define the data at the receiver to be

$$\tilde{r}(t) = \tilde{s}_1(t) * \tilde{p}_1(t) + \tilde{s}_2(t) * \tilde{p}_2(t) + n(t) \quad (5.1)$$

where $\tilde{s}_i(t)$ is the output of the user's transmitter, \tilde{p}_i is the complex envelope of the channel's impulse response, and $n(t)$ is complex Gaussian white noise.

The transmission is implemented in a manner to simulate streaming data such that the program can run indefinitely if desired. For the purpose of these simulations, the program will run for a total of 1000 OFDM symbols on each carrier. For 128 channels each carrying 96 4-QAM symbols, this is approximately 25 million bits, although the simulation will terminate the iteration once 400 errors are counted.

For simulations that calculate symbol error probability, we use the definition of symbol error

probability P_s for an ideal AWGN channel: [19]

$$P_s = \frac{2(M_1 - 1)}{M_1} Q \left(\sqrt{\frac{2E_b}{N_0} \frac{3 \log_2 M_1}{(M_1^2 - 1)}} \right) \quad (5.2)$$

where M_1 is the 1-D size for QAM modulation, and E_b/N_0 is the energy per bit to noise density ratio.

5.1 Ideal Channel Simulations

The first simulation presented is an ideal AWGN channel using ideal timing. Time and frequency estimation are turned off and equalization is turned off. The purpose of this simulation is to generate the system's P_s and compare it to the theoretical values and verify that our additive noise is properly scaled. The result is shown in Figure 5.1.

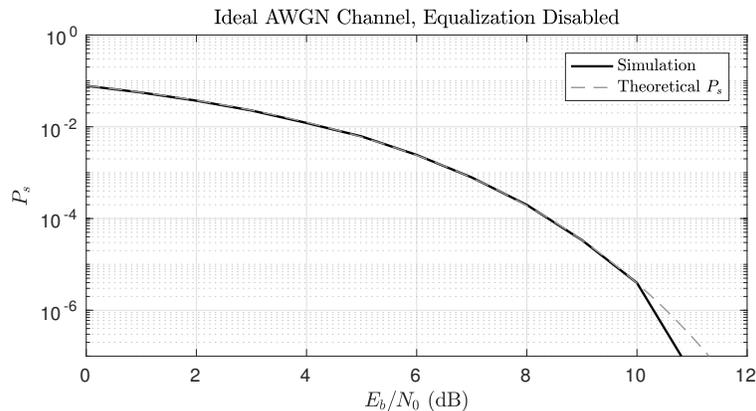


Figure 5.1: Simulation Under an Ideal AWGN Channel, Perfect Timing, Estimation and Equalization Disabled

The results closely match the theoretical and provides a good basis for later simulations.

The next simulation introduces channel equalization to the OFDM demodulator over an ideal AWGN channel, with estimation still disabled. The results are presented in Figure 5.2.

We notice about a 1 dB shift to the right at high SNR due to the introduction of imperfect

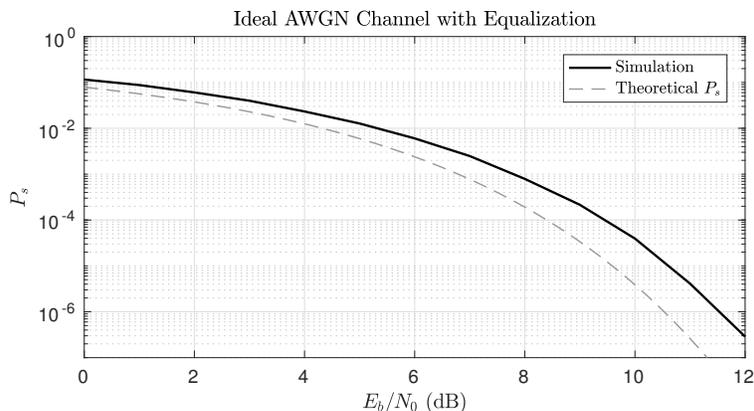


Figure 5.2: Simulation Under an Ideal AWGN Channel, Perfect Timing Equalization Enabled, Estimation Disabled

channel equalization. This penalty is a result of the polynomial curve fitting applying a fit to noisy pilot tones. This brings up possible questions of what may be done to remove this offset, such as pilot tone averaging, but this will not be studied further at this time. This plot may be considered a baseline for performance of our ability to correct for non-ideal channels when equalization is on.

Next we wish to look at the performance loss as a result of using the timing estimator, under an ideal channel. For this simulation, channel equalization and frequency estimation are disabled while timing estimation is enabled. The results are presented in Figure 5.3. We observe a significant penalty introduced due to jitter in the estimate $\hat{\theta}$. The timing estimator runs at a clock rate of $2R_s$ so if the estimator is incorrect by one sample, a noticeable phase shift will occur in demodulation. In the second plot we enabled channel equalization with a 1st-order polynomial fit to correct for this phase shift, and the results match our observations in Figure 5.2. We identify that equalization is essential when timing estimation is enabled.

The next simulation tests the performance loss of the frequency estimator under an ideal channel. For this simulation, channel equalization and timing estimation are disabled and frequency estimation is enabled. The results are presented in Figure 5.4. There is no noticeable

5.1. IDEAL CHANNEL SIMULATIONS

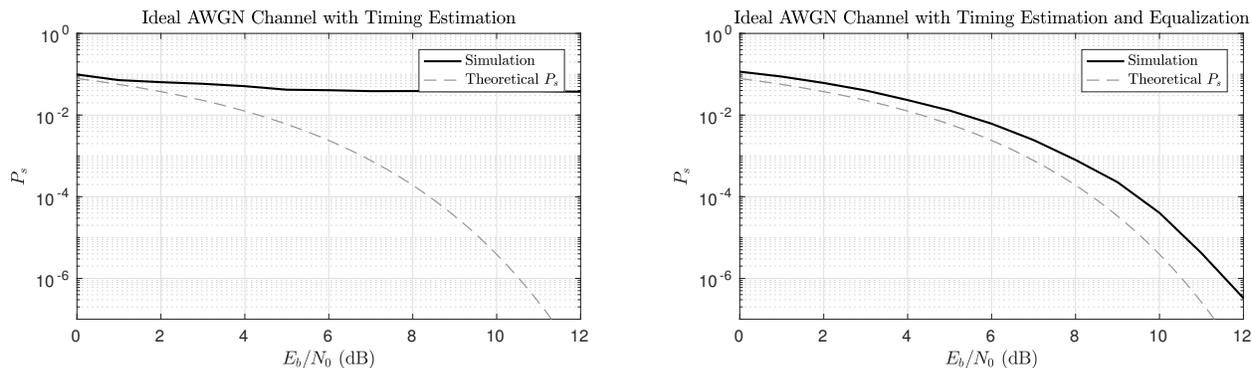


Figure 5.3: Simulation Under an Ideal AWGN channel, Timing Estimation Enabled, Frequency Estimation Disabled and Equalization Disabled (left) and Enabled (Right)

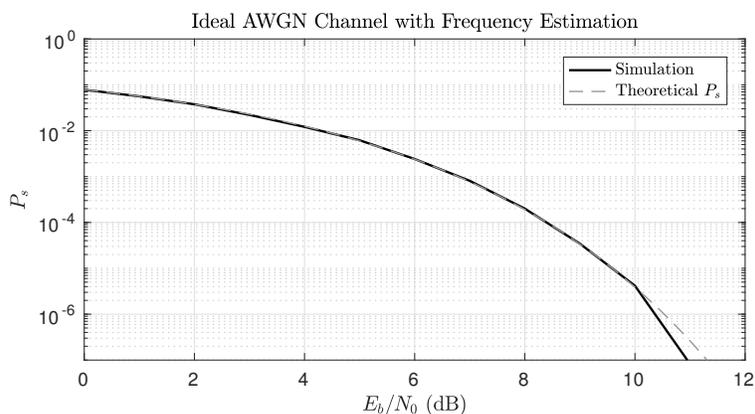


Figure 5.4: Simulation Under an Ideal AWGN channel, No Frequency Offset, Frequency Estimation Enabled, Timing Estimation and Equalization Disabled

change when using the frequency estimator compared to Figure 5.2 so the frequency estimator jitter is small, although some small amount of error is to be expected. This simulation suggests the penalty of using the frequency estimator in an ideal AWGN channel is negligible. While this configuration is provided to test error introduced due to the frequency estimate, it is impracticable to use frequency estimation without equalization as a non-zero frequency offset causes a rotating phase offset which requires the equalizer to correct.

5.2 Non-Ideal Channel Simulations

This section will introduce non-ideal channels to the simulations which include timing offsets, frequency offsets, and multi-path fading channels.

The first case presented is a measurement of the performance of the timing estimator. Various amounts of small timing offsets are introduced as delays in User 1's channel. As discussed in Chapter 3, these timing offsets produce ISI in the OFDM demodulator and requires correction. To execute these tests, the time and frequency estimators are disabled while the simulation executes with channel equalization both enabled and disabled. Four fixed timing offsets were selected and are shown in Figure 5.5. For very small offsets, the error

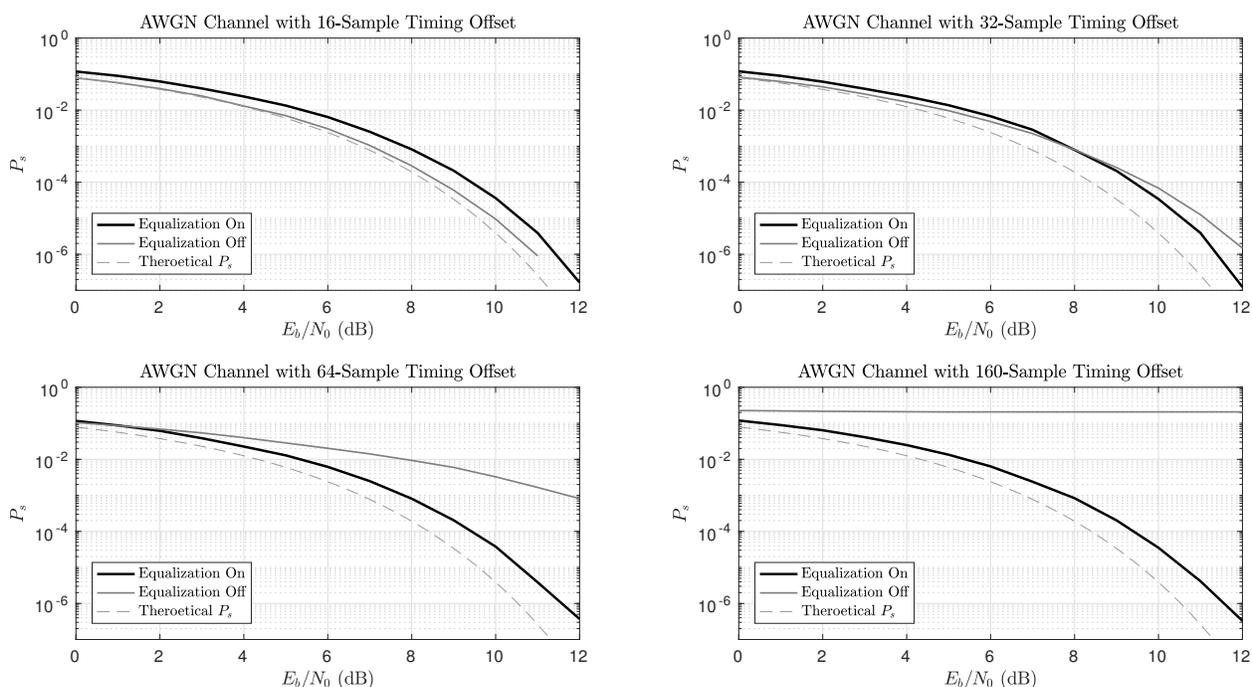


Figure 5.5: AWGN Channel with Timing Offsets, Estimators Disabled

introduced due to ISI is not significant as shown in the 16-Sample case, which is a timing error of $R_s/20$. However when the timing error exceeds $R_s/10$, equalization is beneficial and

5.2. NON-IDEAL CHANNEL SIMULATIONS

needed to maintain integrity of the system. In all cases the equalizer works, correcting for ISI for the data-carrying subcarriers. The simulation provides demonstration that large offsets may be corrected in the equalization process as the results for timing offsets do not exceed the performance loss of using channel equalization, as shown in Figure 5.2.

The next simulation will present measurements on the ability to correct for frequency offsets. For this test, channel equalization is enabled and timing estimation is disabled. It is required to perform channel equalization to correct for the rotating phase offset that occur with frequency offsets. Four fixed frequency offsets were selected and are shown in Figure 5.6. The

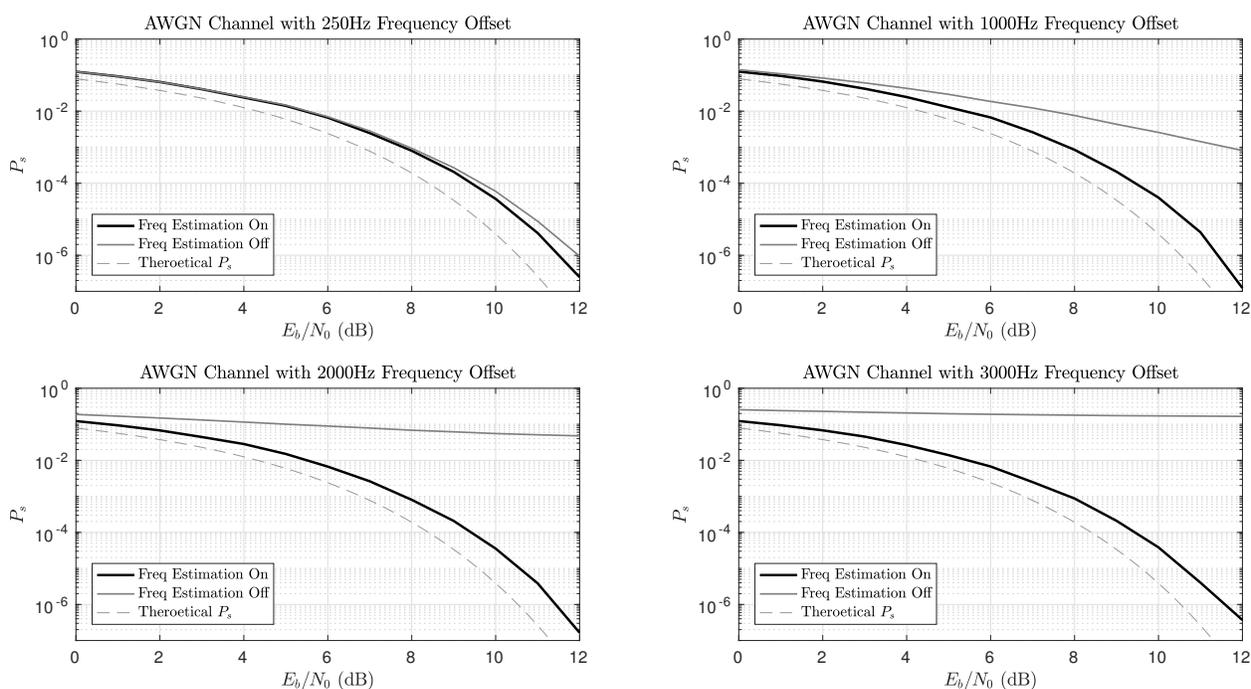


Figure 5.6: Ideal AWGN Channel with Frequency Offsets, Estimators Disabled

simulation demonstrates the frequency estimator's ability to correct for frequency shifts as long as the frequency offset is less than one-half a tone spacing, producing symbol error probabilities matching the case without a frequency offset (Figure 5.2). For the case where the frequency offset was 4000 Hz (not show), the estimator failed to correct since the subcarrier spacing is 7.8 kHz.

5.3 Multipath Channels

The following simulations present cases that are more similar to the real world by simulating a multi-path environment using a 2-ray model. For these simulations, time estimation, frequency estimation, and equalization are enabled. The two ray model used has an impulse response defined as

$$\tilde{p}(t) = 0.8\delta(t) + 0.5e^{-j\pi/4}\delta(t - 2.25\mu s) \quad (5.3)$$

which has a magnitude frequency response shown in Figure 5.7

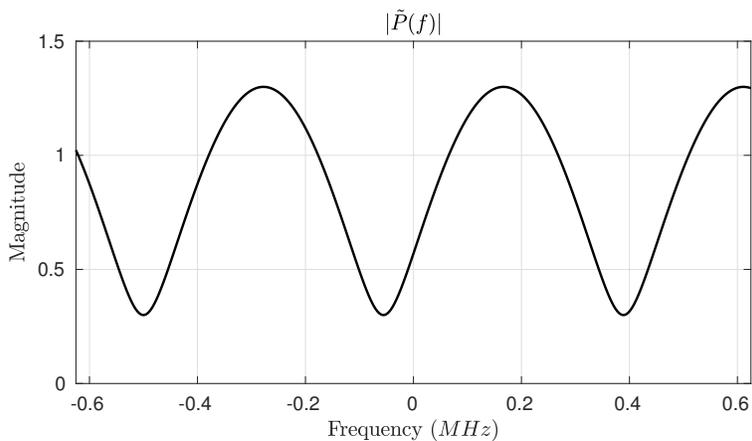


Figure 5.7: 2-Ray Model Frequency Response

The first case considered will have no frequency offset. For comparison, the simulation compares the performance of equalization with timing and frequency estimation to a case with no estimations or equalization. For the case where equalization is enabled, we vary the degree of polynomial in the equalizer to measure the performance in frequency selective situations. The results are shown in Figure 5.8. These simulations show that in order for equalization to be of use, the polynomial order must be sufficiently large enough to handle variations in the channel. Under the scenario tested, using an order of 5 or less was not sufficient. These curves are further right, meaning that symbol error probabilities are higher than expected, so further study is needed. It was noted in other simulations not presented

5.3. MULTIPATH CHANNELS

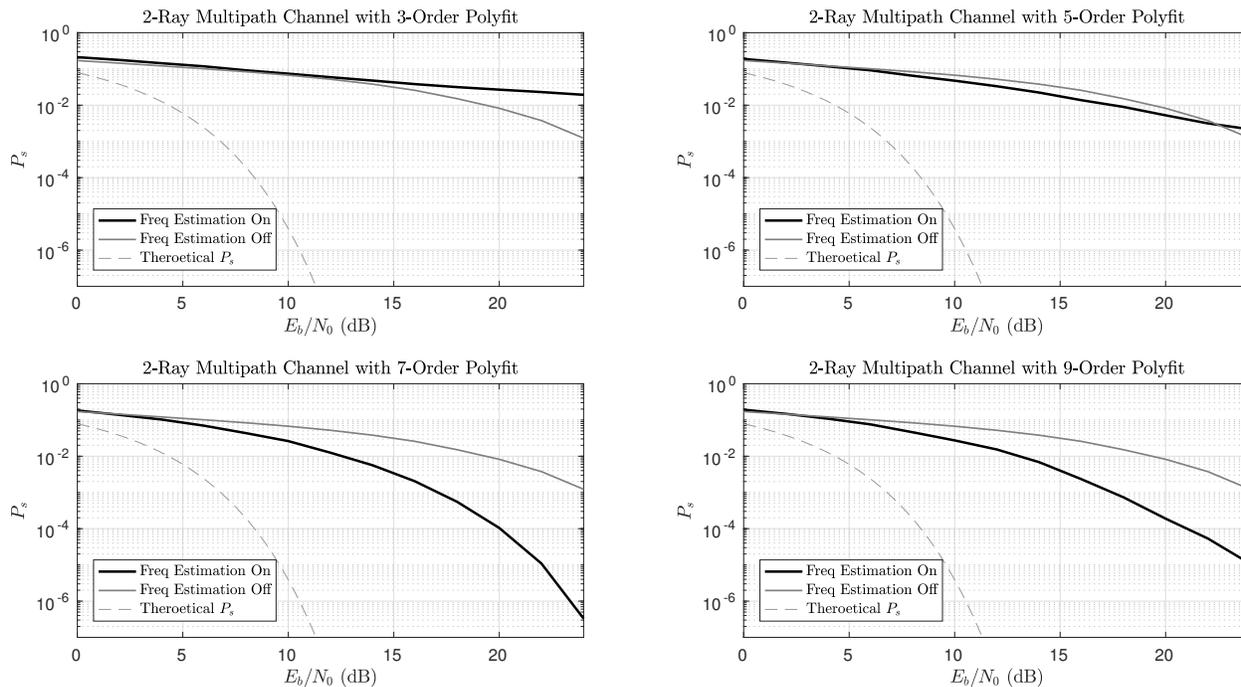


Figure 5.8: 2-Ray Model Channel with Varying Fitting Orders

that when the delay spread was $2 \mu\text{s}$ or less the polynomial curve fit produced good results but for results $2.25 \mu\text{s}$ and larger the polynomial curve fit method is insufficient.

The last simulation performed uses the same multipath channel in (5.3) and introduces a 1000 Hz Doppler shift. Due to the frequency offset it is not useful to compare with an un-equalized case so only the effects of polynomial channel equalization will be presented. The results are shown in Figure 5.9.

The results are similar to the case without a frequency offset for a given polynomial order (Figure 5.8). As with the previous case, lower order polynomials provided insufficient corrections, although a 9th order polynomial was not any better.

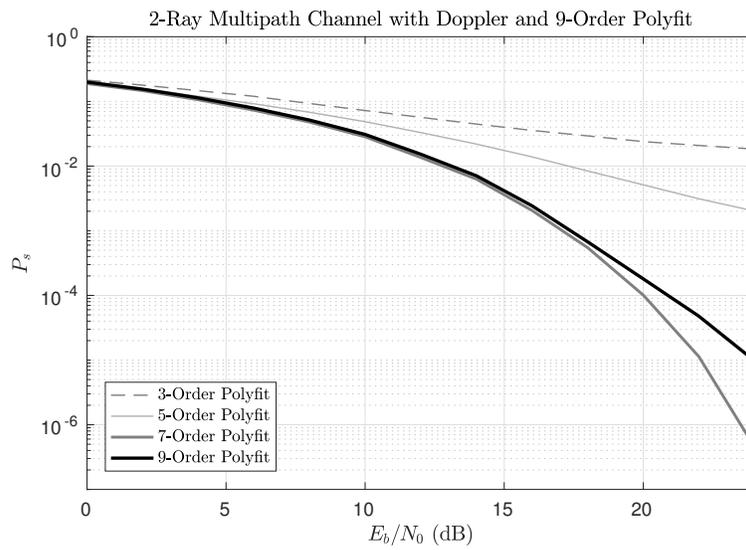


Figure 5.9: 2-Ray Model and 1000 Hz Offset Channel with Varying Fitting Orders

Chapter 6

Conclusion

In this thesis we have identified several challenges that exist when developing wideband radios for use in asynchronous multi-user applications. We studied a particular application and identified the need to use efficient channelizing hardware and presented two transmitter and two receiver architectures suitable for this application. We identified that some architectures may be more beneficial when a small number of channels are active while the other is more efficient when more channels are used. While discussing these architectures, we provided examples using our application.

Channel equalization and estimation techniques were presented which are suitable for asynchronous multi-user OFDM applications. The methods provided are efficient and effective, though with some limitations identified. The issue of timing and timing correction in OFDM was presented with solutions proposed to overcome these challenges.

It was identified that the performance of the system, such as computational cost, *ISI*, peak sidelobes, and out-of-band emissions is largely influenced by the performance and efficiency of the prototype filter. This filter is a long filter and optimizing long filters have computational limitations. To overcome these limitations we proposed multiple filter design techniques and compared the performance result of using asymmetrical filters compared to symmetric

filters. We presented filter design techniques that produce an optimized result that targets minimizing out-of-band emissions constrained to a set of design parameters.

Lastly, we performed an end-to-end simulation that tested and validated the system using the parameters of our studied application. The simulations performed selective tests targeted at measuring an individual component in the estimation and equalization process. At the end, a multi-path channel was simulated with a Doppler offset to test the combined performance of equalization and estimation, as would be necessary when this system is deployed.

6.1 Future Work

The work presented in the thesis presents several opportunities for advancements and continued studies. As the application studied moves towards deployment, improvement to each of these sections may be possible. We will highlight some of the challenges encountered in this study that we feel deserves further study.

One of the topics deserving more attention is the issue of frequency estimation. While this thesis presents a solution in Chapter 3, the presented method of frequency estimation is limited to one-half of the tone spacing. For the application studied this is insufficient. While we hypothesized that the frequency offset may be tracked and thus designed to handle larger offsets, these ideas have not been tested and deserve further study. [14] Alternately, it may be possible to use the estimation method presented in this thesis if the TAs are able to estimate the frequency shift by computing air speed and trajectory. This method has not been explored.

As observed in our results, there is a penalty associated with using channel equalization, as the polynomial curve fit, or any curve fit, does not differentiate between noise and the desired signal. This is an area that could be improved and deserves additional research. We

6.1. FUTURE WORK

hypothesize that averaging pilot tones could be one method of reducing this penalty, however one must take into consideration the jitter from the output of the frame estimator and how the equalization process is responsible for correcting for these timing offsets.

A more significant issue observed regarding the polynomial curve fit is that it has limited ability to fit to multi-path channels with large delay spreads. For smaller delay spreads of $2 \mu\text{s}$ or smaller, the polynomial curve fit performed well. However, as the delay spread increases beyond $2 \mu\text{s}$, the frequency response becomes more sinusoidal that is rapidly changing over the channel bandwidth. The polynomial basis functions are not a good choice for curve fitting in this scenario. Using a different basis such as a Fourier expansion would be more suitable in this application and deserves further research.

Additionally, the 2-Ray simulations produced higher bit error rates than were expected. Further study is needed regarding polynomial fitting and performance on frequency selective channels.

This thesis did not take into account any sort of block or spatial coding. It has been identified that FEC codes such as LDPC codes are of interest in the application, but these were not studied. Block codes also are likely of interest to provide redundancy over the frequency spectrum. Additionally spatial codes that utilize Multiple-In Multiple-Out (MIMO) transmission was not part of this study. The simulation provided could be improved to extend beyond a two-ray model by including time-varying channels.

Another consideration is how robust the estimators are to non-ideal conditions such as nonlinear amplifiers. As OFDM is being used, the transmitted signal has a high PAPR. This may translate into clipping or saturation in the amplifiers which are nonlinear effects. One proposal is to incorporate a coding scheme that lowers the PAPR. Another consideration would be to correct for nonlinear behaviors in the equalization process, if the amplifier characteristics are known.

Chapter 7

Bibliography

- [1] In the matter of redesignation of the 17.7-19.7 ghz frequency band, blanket licensing of satellite earth stations in the 17.7-20.2 ghz and 27.5-30.0 ghz frequency bands, and the allocation of additional spectrum in the 17.3-17.8 ghz and 24.75-25.25 ghz. Order on Reconsideration 04-123, Federal Communications Commission, 2004.
- [2] S. Coleri, M. Ergen, A. Puri, and A. Bahai. Channel estimation techniques based on pilot arrangement in OFDM systems. *IEEE Transactions on Broadcasting*, 48(3):223–229, Sep 2002.
- [3] Digital Signal Processing Committee. *Programs for Digital Signal Processing*. IEEE Press, Piscataway, NJ, USA, 1979.
- [4] S. Daumont, B. Rihawi, and Y. Lout. Root-raised cosine filter influences on PAPR distribution of single carrier signals. In *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*, pages 841–845, March 2008.
- [5] P. Duhamel and H. Hollmann. ‘split radix’ FFT algorithm. *Electronics Letters*, 20(1):14–16, January 1984.
- [6] F. Harris and C. Dick. Polyphase channelizer performs sample rate change required for both matched filtering and channel frequency spacing. In *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 1283–1287, Nov 2009.
- [7] F. J. Harris. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
- [8] F. J. Harris, C. Dick, and M. Rice. Digital receivers and transmitters using polyphase filter banks for wireless communications. *IEEE Transactions on Microwave Theory and Techniques*, 51(4):1395–1412, Apr 2003.

-
- [9] S. Haykin. *Communication Systems*. Wiley Publishing, 4th edition, 2001.
- [10] S. Haykin and M. Moher. *Modern Wireless Communication*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.
- [11] E. Hogenauer. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):155–162, Apr 1981.
- [12] J. S. Milton and J. C. Arnold. *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*. McGraw-Hill Higher Education, 4th edition, 2002.
- [13] S. K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill School Education Group, 2nd edition, 2001.
- [14] T. M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for OFDM. *IEEE Transactions on Communications*, 45(12):1613–1621, Dec 1997.
- [15] J. J. van de Beek, P. O. Borjesson, M. L. Boucheret, D. Landstrom, J. M. Arenas, P. Odling, C. Ostberg, M. Wahlqvist, and S. K. Wilson. A time and frequency synchronization scheme for multiuser OFDM. *IEEE Journal on Selected Areas in Communications*, 17(11):1900–1914, Nov 1999.
- [16] J. J. van-de Beek, M. Sandell, and P. O. Borjesson. On synchronization in OFDM systems using cyclic prefix. page 663667, 1996.
- [17] J. J. van de Beek, M. Sandell, M. Isaksson, and P. Ola Borjesson. Low-complex frame synchronization in OFDM systems. In *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, pages 982–986, Nov 1995.
- [18] L. Wasserman and A. N. Willson. A variable-rate filtering system for digital communications. 3:1497–1500 vol.3, Mar 1999.
- [19] S. G. Wilson. *Digital Modulation and Coding*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.
- [20] S. Winograd. *Arithmetic Complexity of Computations*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1980.