NON-ORTHOGONAL WAVEFORMS (NOW)

Billy Conditsis

Bachelor of Engineering

Telecommunications

Department of Engineering

Macquarie University

July 31, 2017

Supervisor: Associate Professor Dr. Rein Vesilo

Acknowledgements

I would like to acknowledge Associate Professor Dr. Rein Vesilo for providing me with guidance on my thesis and for putting in the time and effort to further my understanding in engineering across many topics.

I would also like to thank my family for being there for me, not only to complete my thesis and degree, but for every day leading up to it.

Statement of Candidate

I, Billy Conditsis, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the Department of Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment at any academic institution.

Student's Name: Billy Conditsis

Student's Signature: Billy Conditsis

Date: 31/07/2017

Abstract

This project looks at a new interface to be developed by using non-orthogonal waveforms as opposed to orthogonal waveforms. It focuses on the particular non-orthogonal waveform called Generalised Frequency Division Multiplexing (GFDM), which consists of subcarrier filtered multicarrier modulation using circular convolution. This is important as the upcoming consumer environment is not well suited for orthogonality. This document reports the idea to abandon synchronism and orthogonality altogether, thereby leading to possible problems such as crosstalk or interference, and tests if these impairments may be optimised by a suitable transceiver structure and transmission technique. The way in which this transmission technique will be optimised is by comparison of various filtered GFDM signals through simulations using Matlab software. These simulations represent future proposed transmission examples and highlight the benefits of using specific GFDM filters over OFDM. The main problem to be compared with is Out of Band (OOB) emissions in each of the filters. Comparisons of different parameters in the filters, such as roll-off factor and subcarrier/sub-symbols are manipulated to illustrate positive as well as negative effects. This has allowed an optimal solution to the problem at hand. This optimal solution is revealed using a prototype XIA 4^{th} order filter using full roll-off and maximum sub-symbols. In doing so the performance of GFDM signal is increased via less interference. Overall, a clearer understanding of the proposed waveform is presented as well as recommended.

Contents

1 Introduction

The introduction talks about the main ideas of the project. It begins with background on the problem at hand and defines the project goals. It then explains how the goals are obtained as part of the project plan. The introduction then concludes with a project overview and an outline of the budget. At the end of the document is an abbreviations section which outlines all acronyms used in the research paper.

The current Orthogonal Frequency Division Multiplexing (OFDM) is not suited for the upcoming Internet of Things (IoT) expansion. The IoT expansion relies on a vast amount of unsynchronized machine-to-machine communication. This creates a problem as the current OFDM method uses orthogonal waveforms which are inflexible and therefore not suitable for the near future. A new air interface needs to be developed (5G) with novel features such as, absence of synchronization, very high mobility and very low latency. The current research is essential to address as there is an incredibly large user base that needs a new air interface for the future.

There are several waveforms that are being studied that meet these requirements. The types of waveforms can be classified into two categories:

1. Subcarrier filtered multicarrier modulation using **linear** convolution, namely Filter Bank Multi Carrier (FBMC)

2. Subcarrier filtered multicarrier modulation using **circular** convolution, namely Generalised Frequency Division Multiplex (GFDM)

Each system has their advantages and disadvantages and can be weighed on certain criteria. Due to time constraints and project scope, the focus will be on the circular convolution waveform, GFDM.

The 5G physical layer requires exceptional levels of flexibility, performance, reliability, efficiency, robustness and scalability [1]. When considering these requirements for 5G, the upcoming drivers of technology such as the IoT and Gigabit Wireless Connectivity must also be taken into account [2]. With these requirements, the need for a system to overtake OFDM

may be defined as having increased throughput, loose synchronization, low latency, and particularly low OOB emissions [1]. OOB emissions are important in this document as they relate to meeting the requirements of 5G. OOB emissions refer to the neighbouring signals interfering with each other. By reducing these emissions, less signal interference will occur with its neighbours. This allows for an optimisation of the GFDM signal structure and will be the main focus of this research.

GFDM is a new physical layer scheme created to overcome the major broadband and real-time issues for 5G networks [3]. It has been proposed for 5G wireless communication systems due to several reasons. The key reason is GFDM covers the majority of OFDM advantages and handles its limitations [4]. Furthermore, GFDM OOB emissions can be reduced through optimisation. In doing so, GFDM will be overcoming some of OFDM's limitations when looking at the upcoming changing environment [4]. The method in which the reduction of OOB emissions is completed in this project is through a prototype pulse shaping filter. This filter narrows the transmitting signal to ensure the least emissions are leaked. With less leakage, less interfere with its neighbouring signals will occur and more signal space is efficiently used.

GFDM is an example of a "New Waveform" since it has less OOB emissions and hence a more efficient spectrum usage [5, 6]. In order to effectively measure these OOB emissions, the approach taken is to create GFDM signal simulations. The use of Matlab software is chosen as it effectively measures these OOB emissions through the use of realistic signals. These simulations will show that GFDM is a good decision for low latency environments such as the 5G air interface [6, 7].

The next generation of wireless networks will face different challenges from new scenarios. It should satisfy the demands of the increasing mobile data rate in addition to the rapidly changing mobile data traffic that is approximately doubling every year [4]. GFDM is a key candidate for the physical layer of the next generation cellular system as it is capable of addressing all types of communications foreseen for the 5G networks. This review investigates the performance of GFDM as a candidate waveform for the physical layer of future wireless communication systems.

1.1 Project Goal

The overall goal of this project is to optimise and recommend a 5G non-orthogonal waveform. The scope of research involved in this project is limited to GFDM as a candidate waveform due to time constraints and its credibility for a reliable solution. Goals have been set up, defined by the project scope to explain how a new air interface will overcome the synchronisation issues in the upcoming environment. Creating an initial long-term goal gives direction to the research and allows for more focused goals to be later developed. This initial goal is to not only show that GFDM is a solution but to also optimise GFDM such that it is a prime candidate for the new air interface. The overall aim of this project is chosen not to be too complex but enough to challenge, test intellectuality and become of use in the future to the many users.

Currently, there are issues in GFDM such as OOB emissions which result in interference and hence degrade performance. However, this issue leaves room for improvement in the waveform and will be a focus point of the project. The performance improvement of GFDM will be achieved by showing which transmission filter yields better performance and under which parameters. This is accomplished through several short term goals listed in the Gantt chart in figure.1 below. Realistic simulation data will be developed over various filters such that the performance of each may be easily compared whilst detailing trade-offs and relationships. An initial aim of a 10% increase in performance over OFDM has been set to ensure the results are useful. This increase in performance is chosen as it is feasible with the experience and time available. These results may also be directly compared with OFDM simulations as a standard of improvement. Once complete, results will be analysed and recommendations for GFDM will be made. These results will be relevant as they will have a direct influence on GFDMs candidacy as a solution to a new air interface

1.2 Project Plan

Creating a project plan is a necessity and was constructed at the very beginning. In doing so the goals become achievable in the time allocated. The project consists of several phases and required significant planning in order be successful. Every week a consultation meeting was conducted with Dr. Rein Vesilo to record progress and maintain alignment of the thesis. The attendance of these meetings is logged and can be seen in appendix.1. To track the work completed each week and remain on schedule, a proposed arrangement of tasks is required. These tasks are displayed in the Gantt chart in figure.1 below and were chosen before commencement of the project.

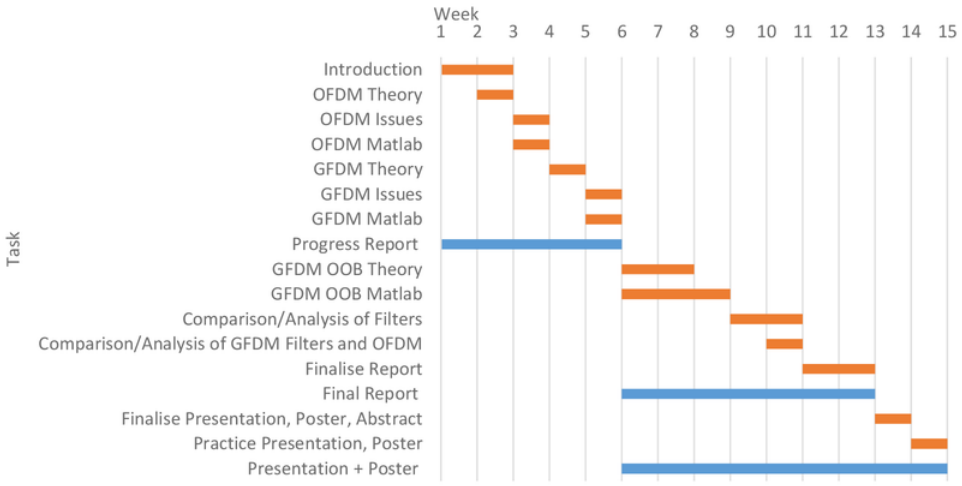

Figure.1 Gantt Chart

This Gantt Chart was referenced before the commencement of each week to ensure the work completed each week is effective and on schedule. As the weeks progressed, milestones were checked off and different tasks launched. Contingency was added on weekends and semester breaks to allow for unforeseen changes and maintain the projects schedule. Some topics were initially planned to be included in the final research paper such as the relationship of timing offset to OOB emissions in GFDM however due to the scope and time restrictions of project they were removed. As a result, the Gantt chart had been adjusted throughout the project to ensure a successful completion.

1.3 Project Overview

The project overview explains how the research idea was developed and then details onto how it was broken into sections to be completed.

The broad research question that this project developed from is, “How can we better prepare for the exponential increase in consumer technological demand?”. This question has motivated many new innovative ideas to support the predicted future needs. The approach taken in this document is to focus on one method (GFDM) and explain how it could be optimised as a solution. This is to be done in terms of OOB leakage reduction through various filters in a GFDM waveform. In order to describe this, the project is broken into four main sections. Each of these sections include contrasting information on both the current OFDM structure and the upcoming GFDM structure along with its filters.

1. OFDM Theory
2. GFDM Theory
3. Matlab Simulations
4. Results and Discussion

The outcomes of these four sections are explained in the conclusions section and provide a detailed solution to the problem of future consumer demands.

The theory sections are part of the projects literature review. Firstly, the theory of OFDM is explained. This is done as GFDM is built upon this theory. The OFDM section also outlines OFDM’s limitations and issues in regards to the future.

After the OFDM section is completed the same steps are taken for GFDM theory. The construction of GFDM is explained as well as how it is built on top of OFDM. Since the issues in OFDM have already been stated, GFDM’s theory explains how it will overcome these issues in its basic structure. GFDM then goes into its proposed limitations and mainly focuses on OOB emissions. This leads to another section explaining the different kind of transmit filters and how they relate to OOB emissions which is a large focus of this project.

Following this, Matlab OFDM simulations are used based on its mathematics explained in the theory section. Once again, building off OFDM, Matlab simulations for GFDM are formed using the OFDM simulation. They are created in a way where direct performance parameters

are measured and compared with each other. After the GFDM simulation is created it is further modified to allow for multiple transmit filters. Finally, a measurement tool for OOB emissions is created to gather results.

This leads to the final results and discussion section. It begins by displaying the performance trade-offs between GFDM filters by varying parameters. It then uses this analysis to declare an overall best suited filter. Once this is completed, benefits through GFDM are demonstrated using a direct comparison to the original OFDM. The document concludes by recommending optimal GFDM filter parameters in order to improve GFDM's candidacy as an upcoming waveform.

1.4 Project Budget

The project was allocated budget of AUD $300.00 from Macquarie University. This project is simulation based and doesn't require much equipment. The main application used is Matlab 2017a and the only equipment required is a computer which can support Matlab through the recommended hardware requirements of a 64-bit operating system, 2GB of ram, 6GB disk space and a hardware accelerated graphics card. Macquarie University provides access to multiple computers on campus all which include licences and enough processing power to meet the recommended requirements for Matlab use. This yields no real cost for use of the software or any need for extra equipment. Macquarie University's license for Matlab also allows students to download the full application at home with no extra fees. The software chosen to complete this project maximises benefits while yielding no cost in the allocated budget.

2 Background and Related Work (OFDM)

Background information on 4G's OFDM is explained to further demonstrate the requirement for GFDM. This information provides the basis for a simulation to be performed on the current OFDM system. Following this are some outlines of the current issues residing in the interface

2.1 OFDM Theory

Multi-carrier systems are systems which split their transmitting data into several components and send each of those components over separate parallel sub-channel. OFDM is a special form of this Multi-Carrier Modulation (MCM) as it uses orthogonal waveforms during its transmission over the sub-channels. The current 4G mobile network system uses MCM in OFDM's orthogonality structure today. These signals are defined to be orthogonal to ensure a well-defined signal with no leakage. In doing so, this technique suppresses the overlapping Inter-Carrier Interference (ICI) between the parallel signals and hence enables a distortion-less and efficient transmission scheme.

In order to simulate OFDM, a complete understanding of the mathematics needs to be explained. By breaking up the modulation scheme into its smallest fragments and representing them mathematically, a construction of the entire signal can be created for simulation. The first fragment to start with is the individual subcarriers. k is the number of subcarriers, T_{sym} is the symbol duration and f_k is the frequency at this subcarrier. This is shown in OFDM as $f_k = \frac{k}{T_{sym}}$ derived from the equation $f = \frac{1}{T_s}$. $\frac{1}{T_s}$ is the subcarrier symbol rate and provides orthogonality as set of pulses to satisfy stability criterion for zero Inter-Symbol Interference (ISI). This subcarrier frequency is part of a time-limited complex signal $e^{j2\pi f_k t}$ which represents the different subcarriers. Orthogonality in OFDM is then achieved by making the integral of the products of the different subcarriers fundamental period equal to zero as seen in (1).

$$\frac{1}{T_{sym}}\int_0^{T_{sym}} e^{j2\pi f_k t}e^{-j2\pi f_i t}dt = \frac{1}{T_{sym}}\int_0^{T_{sym}} e^{\left(\frac{j2\pi k}{T_{sym}}\right)t}e^{\left(-\frac{j2\pi i}{T_{sym}}\right)t}dt$$

$$= \frac{1}{T_{sym}}\int_0^{T_{sym}} e^{\frac{j2\pi(k-i)}{T_{sym}}t}dt \quad (1)$$

$$= \begin{cases} 1, \forall \text{ integer } k = i \\ 0, \text{otherwise} \end{cases}$$

Let N be the number of modulated subcarriers. Then, by using discrete time samples at $t = nT_s = \frac{nT_{sym}}{N}$ for $n = 0,1,2,\dots,N-1$, this equation can be written in the discrete time domain using the Fourier transform of the signal in (2).

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{\left(\frac{j2\pi k}{T_{sym}}\right).nT_s}e^{-\frac{j2\pi i}{T}.nT_s} = \frac{1}{N}\sum_{n=0}^{N-1} e^{\left(\frac{j2\pi k}{T_{sym}}\right).\frac{nT}{N}}e^{-\frac{j2\pi i}{T_{sym}}.nT\frac{sym}{N}}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} e^{\frac{j2\pi(k-i)}{N}n} \quad (2)$$

$$= \begin{cases} 1, \forall \text{ integer } k = i \\ 0, \text{otherwise} \end{cases}$$

This equation is derived by the below parameters.

$$s(t) = \frac{1}{N}\sum_{k=0}^{N-1} s_k(t)$$

$s(t)$ OFDM signal

N number of modulated subcarriers (normalized to N)

$s_k(t)$ modulated subcarriers

$$s_k(t) = \mathrm{Re}\left\{z_k(t)e^{j[2\pi(f_c+k\Delta f)t]}\right\}, 0 \le t \le T_s$$

$s_k(t)$ in complex envelope notation

$z_k(t)$ complex envelope of the kth subcarrier (or equivalent lowpass signal)

f_c nominal carrier frequency

Δf minimum frequency spacing

$z_k(t) = Z(k) = I_k + jQ_k, 0 \leq t < T_s$ I_k in-phase component of $z_k(t)$, I value of kth subcarrier symbol

Q_k quadrature component of $z_k(t)$, Q value of kth subcarrier symbol

OFDM is widely used in many wireless systems today including the popular IEEE 802.11a,g Wireless Local Area Network (WLAN). The process for the signal transmission begins with modulation. A modulator controls various aspects of a waveform to later be transmitted. This process for OFDM systems is displayed below in figure.2.

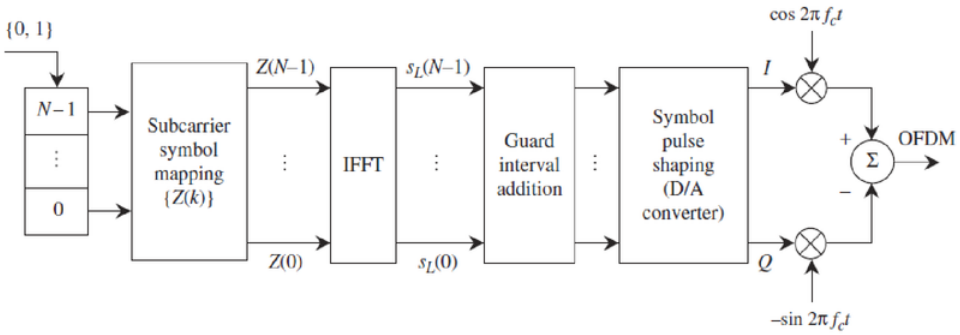

Figure.2 OFDM modulator [8]

There is an incoming data stream of 1's and 0's into the modulator. The data incoming is at the rate of $\frac{1}{T}$ bits/s. This data is then demultiplexed (serial-to-parallel conversion) into N parallel streams. Each parallel stream has a rate equal to $\frac{1}{NT}$ bits/s and independently modulates one of N orthogonal subcarriers [8]. The OFDM subcarriers are chosen to be mutually orthogonal with the minimum frequency spacing of $\frac{1}{T_s}$, which is the subcarrier symbol rate [8].

For the kth bit stream, a group of $\log_2 M$ bits is mapped into a subcarrier symbol $Z(k) = I_k + jQ_k, 0 \leq t < T_s$.

The sequence of N symbols is then fed into the IFFT (Inverse Fast Fourier Transform) to obtain a sequence which represents the samples of the OFDM complex envelope [8].

A guard interval is then added in the form of a Cyclic Prefix (CP), which is data before the message. This is done to eliminate inter-carrier and inter-symbol interference.

Following this, the output of the D/A (Digital/Analogue) converter is a complex I-Q (In phase-Quadrature) waveform with a p(t) interpolation pulse shape [8]. The pulse shape p(t) has the form $\frac{\sin x}{x}$, which is used for the reconstruction of a sampled signal [8]. This signal is then transmitted through a channel to be demodulated as per figure.3 below.

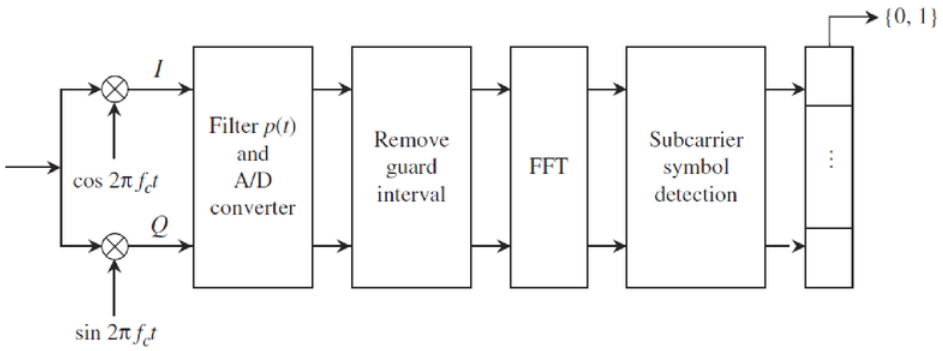

Figure.3 OFDM demodulator [8]

The demodulator does the opposite of the modulator whereby it recovers the information sent by the modulator.

The received OFDM signal is first demodulated into the I-Q baseband waveform, which contains the subcarrier symbols. [8] The baseband waveform is then passed through a filter p(t) to receive only the required signal. The A/D convertor then converts the waveform into time-domain samples [8]. After this, the guard interval is then removed as it's no longer needed as part of the message.

The FFT (Fast Fourier Transform) process occurs in order to transform the time-domain samples into the frequency-domain samples, which represent the I-Q values of the subcarrier symbols [8].

Finally, the subcarrier symbols are then de-mapped into parallel bit streams. The parallel-to-serial converter (multiplexer) combines these parallel bit streams back into one single serial bit stream at the output for the final message [8].

2.2 Issues in OFDM

OFDM is a popular method of encoding due to its orthogonality resulting in high spectral efficiency. However, there are a few issues in OFDM that lead to a reduction in overall performance when compared to new physical layer designs. These issues are outlined in this section to further understand how GFDM aims to overcome them through the use of one CP per block and non-orthogonality.

Inter-block Interference

OFDM suffers from bandwidth inefficiency due to interference. This interference is caused from a significant portion of each data packet as it is being allocated CP segments. Inter-block-interference (IBI) occurs when the length of the CP is less than the number of channels. As the length of the cyclic prefix is reduced, IBI occurs and causes a deterioration in performance [9]. GFDM aims to overcome each of these issues as there is only one CP per group of symbols which significantly increases the size to each data packet.

Carrier Frequency Offset

OFDM signals are also more sensitive to carrier frequency offset than single carrier modulation signals due to its orthogonality. The performance reduction is caused by two main factors, namely; amplitude reduction of the desired subcarrier and inter-carrier interference.

Amplitude Reduction

Amplitude reduction occurs because during the FFT the desired subcarrier is no longer sampled at the peak of the sinc-function. Also, In-phase/quadrature (IQ) mismatch may be caused due to the 90° phase shifts. Amplitude reduction and imbalance between I and Q branch can seriously destroy the orthogonality and degrade the system performance in OFDM signals [10].

Inter-carrier Interference

OFDM has a high spectral efficiency because it has mutually orthogonal sub-carriers. If the subcarriers are closely spaced without any guard bands, there will not be any inter subcarrier interference [11]. However, any mismatch between the transmitted and received frequencies

will distort the orthogonality of the subcarriers. The difference between the transmitted and received frequency is known as Carrier Frequency Offset (CFO) and occurs due to the Doppler effect. The result of the interference of one subcarrier with one or more of its neighbouring subcarriers is known as inter carrier interference and reduces performance [11].

OFDM has been found to be a poor choice in large multiuser applications where any loss of orthogonality disrupts synchronisation among users and leads to a significant loss in performance [12]. The method for GFDM transmission is based on non-orthogonality which significantly reduces the issue of CFO. CFO still exists in GFDM and is mentioned in GFDM issues. However, since GFDM has a non-orthogonal structure, it allows for larger use of unsynchronised messages and becomes a prime candidate for multiuser applications.

3 GFDM

The proposed solution for the current project is GFDM. In the following section, GFDM will be theoretically explained and then simulated using Matlab. The types of GFDM filters will then be discussed in regards to their mathematical structure. Following this, issues that reside in GFDM will be outlined. A prototype filter will be developed with the aim to reduce OOB emissions in GFDM and will be simulated later in the document.

3.1 GFDM Theory

As the name suggests, GFDM is a generalisation of OFDM such that it modulates the data in a two-dimensional time-frequency block structure [13]. Each block consists of a number of subcarriers and sub-symbols. The subcarriers are filtered with a flexible pulse-shaping filter that is circularly shifted in both time and frequency domains [13]. These are circularly shifted as GFDM uses circular convolution in the filtering process. The computation time and accuracy of multiplication in convolution operation is crucial for the performance of digital signal processing operations [14]. GFDM uses circular convolution over linear convolution as per the below.

Linear convolution calculates the output for any linear time invariant system given its input and its impulse response [15]. This effectively improves the signal spectrum localisation of MCM. Due to these additional filtering operations, successive time-domain symbols may overlap [15]. This makes each MCM symbol no longer independent but rather correlated with its neighbours [15].

To overcome this, circular-convolution-based MCM is introduced. Circular convolution is the same as linear convolution with the exception of the support signal being periodic. This method combines several MCM symbols together to form a “block” [15]. Inside each block, linear convolution remains, yet when comparing individual blocks, independence is preserved [15]. This procedure has several advantages. The first being the block frame structure allows

for the addition of a single Cyclic Prefix (CP) and a Cyclic Suffix (CS) which relaxes the requirement for synchronisation [3]. The single CP inserted for the entire block protects the information contained in time slots and results in a higher spectral efficiency when compared to OFDM, as clarified later in Matlab simulations [3, 13]. The low latency requirement, which is a major challenge in the upcoming 5G demands, may be fulfilled by GFDM due its flexibility as a non-orthogonal wave form and block structure as explained below [3, 16].

The block structure is evident in figure.4 whereby K and M are the number of subcarriers and sub-symbols. This block structure demonstrates how the subcarriers are mapped where $\mathrm{N} = \mathrm{MK}$ is the size of the payload containing $\mathrm{d_{k,m}}$ data symbols [13].

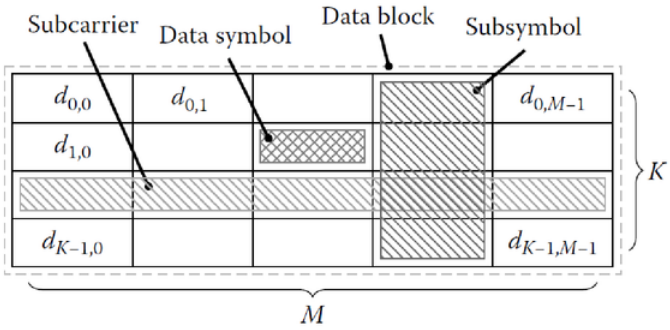

Figure.4 GFDM block structure [13]

GFDM is a MCM method whereby each subcarrier, K, is pulse shaped by a transmission filter $\mathrm{g[n]}$. Various pulse shaping transmission filters are simulated and tested later in the project to see their relationships to OOB emissions. In these filters, data symbols are spread across time and frequency. Each subcarrier in the stream is controlled by the filter through limiting its frequency response to a specific bandwidth [12]. The purpose of these filters is to allow control on how the symbols are transmitted.

GFDM is circularly shifted in time and frequency which produces the prototype filter $\mathrm{g_{k,m}[n]}$ in (1) where $\mathrm{n} = 0,1, \dots, \mathrm{N} - 1$ denotes the sampling index [13]. The modulo operation finds the remainder after a division of N.

$$\mathrm{g_{k,m}[n]} = \mathrm{g[(n - mK)modN]e^{j2\pi\frac{k}{K}n}} \qquad (1)$$

This filter represents the subcarrier mapping and modulation of a GFDM signal which is displayed below in figure.5.

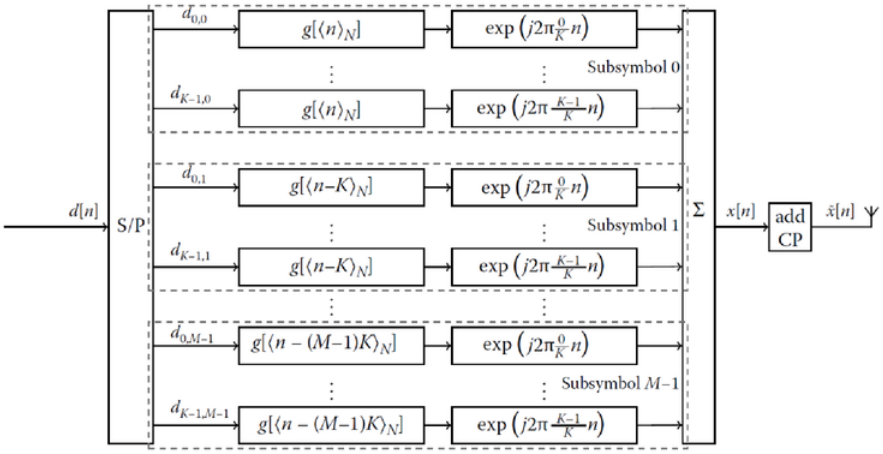

Figure.5 GFDM Modulation [13]

The formulas in (2),(3) and (4) below illustrate how the modulation process is formed. Assume $s_{k,m}$ to be the complex-valued data symbol carried by the k^{th} subcarrier and the m^{th} sub-symbol. The GFDM signal is then represented by (2) [17].

$$x[n] = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} s_{k,m} g_{k,m}[n] \tag{2}$$

The pulse shapes for each subcarrier and sub-symbol is then organised in a modulation matrix. The equation for this matrix is seen in (3).

$$A = [\mathbf{g}_{0,0} \cdots, \mathbf{g}_{K-1,0} \mathbf{g}_{0,1} \cdots \mathbf{g}_{K-1,M-1}] \tag{3}$$

$g_{k,m}$ is a column vector containing the samples from $g_{k,m}[n]$ [17]. The GFDM transmit vector is given by (4).

$$\mathbf{x} = A\mathbf{s} \tag{4}$$

s is a column vector with all data symbols $s_{k,m}$ [17]. The modulation process explained in figure.5 is a small section of the GFDM signal transmission. Figure.6 illustrates this modulation process in high level block diagrams as subcarrier mapping and digital pulse shaping.

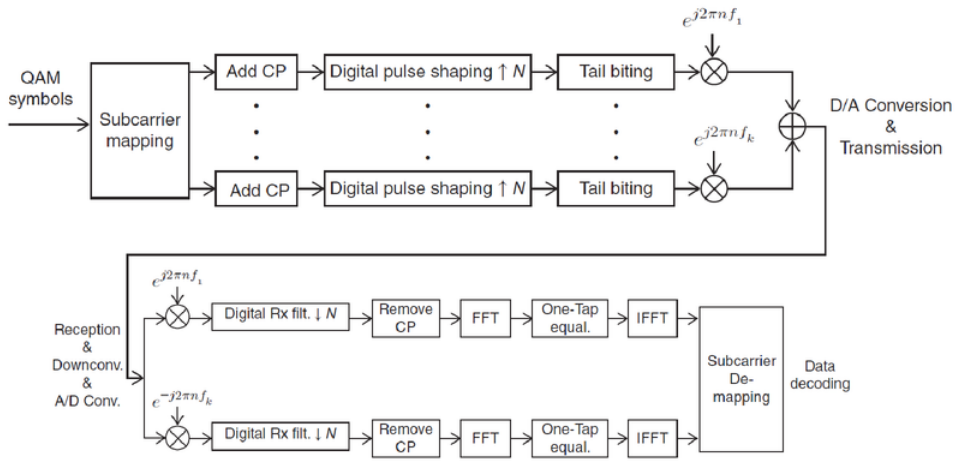

Figure.6 GFDM transmitter and receiver [15]

Following the subcarrier mapping explained earlier, the CP is added at the end of the block. As prior mentioned in OFDM issues, packet construction in GFDM only requires one CP per block.

The data symbols for each subcarrier are then filtered through a pulse shaping filter, $g[n]$, that limits the ICI to only its adjacent subcarriers and is where the optimisation problem of this project is focused.

After the filtering, tail biting occurs. Tail biting refers to the fact that GFDM uses circular convolution in the filtering process and expresses unique circular detection properties [13]. Finally, data symbols are then transmitted at an interval T and subcarrier spacing is set to $F = \frac{1}{T}$ similar to OFDM [12].

A similar process is mirrored on the receiver side transceiver. Various receiver filters are used to focus on receiving only the original transmitted signal.

After the signal has been filtered the CP is removed as it is no longer needed. Due to the tail biting and CP usage, a process call one-tap equalization is used [15]. It occurs in the frequency domain hence the Fourier Transforms and Inverse Fourier Transform steps in figure.6. It is used to remove the ISI introduced by the channel and by the filtering operations [15]. After this is completed the data for a GFDM signal has been decoded.

3.2 GFDM Filters

Filters are a great method used to optimise communication systems in transmission and reception of a signal. In each of these cases only the required signals for transmission are focused on to increase performance. In GFDM, there are a number of transmit and receive filters available with different complexities and trade-offs.

Receive filters restore those signals that may have been distorted while being transmitted across a channel. The receiver filters available for use include:

- **Matched filters (MF)**
  Can include Successive Interference Cancellation (SIC) to cancel ICI and ISI.
- **Zero-Forcing (ZF)**
  Low complexity filter for Additive White Gaussian Nosie (AWGN) or multipath channels.
- **Minimum Mean Square Error (MMSE)**
  A filter with linearithmic complexity for AWGN channels [18].

In this project there will be a focus on the optimisation of transmit filters only. Transmission filters shape the pulse such that an ICI free signal is carried across the channel. Transmit filters are used and analysed in this paper as the main variable relating to GFDM's increase in performance.

The filters constructed in this document are listed below along with their frequency responses. These filters are later simulated through Matlab for comparison. There exists a roll-off factor, α, in the filters frequency responses. This roll-off factor refers to the steepness of the transmission function with respect to frequency and can range between 0 and 1. When roll-off approaches 1 it is deemed a full roll-off with a high decrease in the side lobes of the signal. When roll-off is set a minimum, the opposite happens and the decrease is slow. This is a parameter that will be varied across the filters to see the exact relationship it has in regards to ICI. The roll-off factor is found inside of $\mathrm{lin}_\alpha(\mathrm{x})$ displayed in (1).

$$\mathrm{lin}_\alpha(\mathrm{x}) = \min\left(1, \max\left(0, \left(\frac{1+\alpha}{2\alpha} + \frac{|\mathrm{x}|}{\alpha}\right)\right)\right) \tag{1}$$

- **Raised Cosine Filter (RC)**
  The RC function is a cosine function raised up to sit above the frequency axis. The function itself and the first derivative are continuous [19].

$$G_{RC}[f] = \frac{1}{2}[1 - \cos\left(\pi \text{lin}_{\alpha}\left(\frac{f}{M}\right)\right)]$$

- **Root-Raised-Cosine Filter (RRC)**
  The RRC function is a square root of the RC filter response.

$$G_{RRC}[f] = \sqrt{G_{RC}[f]}$$

- **Dirichlet Filter**
  The Dirichlet function is a rectangular function in the frequency domain and has a roll-off factor of zero.

- **Xia 1st/4th order Filter**
  Xia is family of real, non-linear functions which contain an ISI free property [20]. The first order Xia frequency response is displayed in (2)

$$G_{XIA1}[f] = \frac{1}{2}[1 - \exp\left(-j\text{lin}_{\alpha}\left(\frac{f}{M}\right)\text{sign}(f)\right)] \quad (2)$$

  The fourth order Xia frequency response is shown in (3) where the polynomial $\rho_4(x)$ maps the range (0,1) onto itself as seen in (4) [21].

$$G_{XIA4}[f] = \frac{1}{2}[1 - \exp\left(-j\pi\rho_4\left(\text{lin}_{\alpha}\left(\frac{f}{M}\right)\right)\text{sign}(f)\right)] \quad (3)$$

$$\rho_4(x) = x^4(35 - 84 + 70x^2 - 20x^3) \quad (4)$$

3.3 Issues in GFDM

In order for GFDM to be a successful new waveform it must have low latency, low OOB emission and be robust against CFO. This raises some issues currently residing in GFDM but also opens the opportunity to target these issues to make it a preferred waveform solution.

In this section latency and OOB emissions are explained as the main issues which are then to be reduced through the simulations.

Latency

Imperfect synchronisation can affect the performance of large multiple access long term evolution (LTE) networks. Due to GFDMs non-orthogonality, imperfect synchronisation directly affects inter-subcarrier interference. The more interference there is, the higher the latency becomes. To overcome this interference, GFDM aims to relax the current requisites of oscillator accuracy of 0.1ppm in LTE up to 10–100 times (1–10 ppm). In doing so, it allows for the design of simpler transmitters, leaving out complex synchronization procedures and reducing signalling overhead [13]. This leaves GFDM with a much lower relative latency compared to OFDM. This doesn't actually remove the sources of interference. Interference for a given subcarrier comes from the surrounding subcarriers when bandwidth-limited filters are used [13]. Hence which filter is used plays a large role in latency reduction and will be tested later in the document.

OOB emissions

OOB emissions refers to the unwanted leakage outside of proposed transmitted message. These emissions occur just outside of the channels bandwidth and are enclosed by the OOB domain emission limits. Figure.7 defines the emissions occurring to the right of a transmission signal.

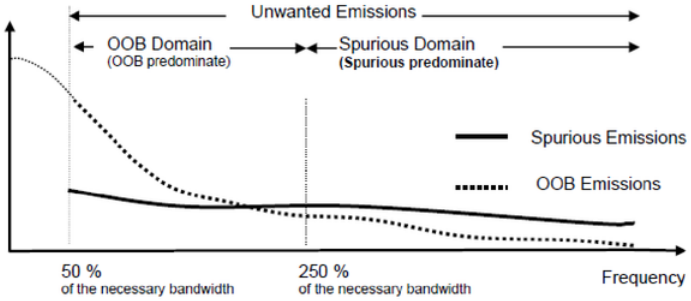

Figure.7 OOB domain definition [22]

The OOB radiation of the GFDM signal is defined as the ratio between the amount of energy that is emitted within the allocated bandwidth, B, and amount of energy emitted outside of it in the specific OOB frequency range [7].

$$O = \frac{|B|}{|OOB|} \cdot \frac{\int_{f \epsilon OOB} P(f) df}{\int_{f \epsilon B} P(f) df} \quad (5)$$

The aim of this project is to minimise this result from the OOB equation in (5). The way in which this will be accomplished is through subcarrier filtering. Subcarrier filtering reduces the GFDM OOB as well as reduce latency. The emission reduction of subcarrier filtered GFDM signals are improved over those compared with OFDM and will be explained in the document. Different filters will be compared with each other as to how well they reduce OOB emissions. The abrupt transitions between the GFDM blocks limit the overall reduction that can be achieved in total OOB emissions.

Another method to reduce interference in receive filters is to produce specific data to cancel out the interference. This method is referred to as Successive Interference Cancellation (SIC). The way in which this is accomplished is as follows. First a particular subcarrier is detected. It is then modulated and its pulse shaping is accomplished [23]. This process occurs before being converted to the subcarriers and generates an approximate transmitted signal. The approximated signal is then subtracted from the received signal to cleanse the effect of adjacent subcarrier interference [23]. SIC algorithm may iteratively reduce the impact of the self-interference on the GFDM SER performance when a demodulator is employed [13]. Once all subcarriers are correctly filtered, the ICI-free signal is demodulated. This process can be repeated multiple times to achieve a better SER performance [13]. At the cost of complexity on the receiver side, the SIC may achieve a SER performance comparable with orthogonal systems such as OFDM [13]. SIC is a separate solution to GFDM interference looked at in this document as it occurs on the receiver side of the transceiver. This is mentioned at the end of the document in future work as it can build upon the research constructed in this paper.

4 Matlab Implementation

Accurate simulations are created below consisting of OFDM, GFDM and GFDMs filters to collect results and compare performance. In order to create accurate simulations, the software Matlab is used. The version used is Matlab 2017a student edition with its default installation packages including the signal processing toolbox.

4.1 OFDM Signal Simulation

An accurate simulation of OFDM through Matlab is required to represent the current benefits and limits it has. The simulation will provide information on what parameters prioritise an efficient GFDM solution. It will also display a direct comparison between the GFDM filtered signals and OFDM. In order to do this through Matlab, the code needs to be realistically created to effectively simulate each part of the OFDM signal, hence being broken into 7 main sections illustrated in figure.8 derived from OFDM theory.

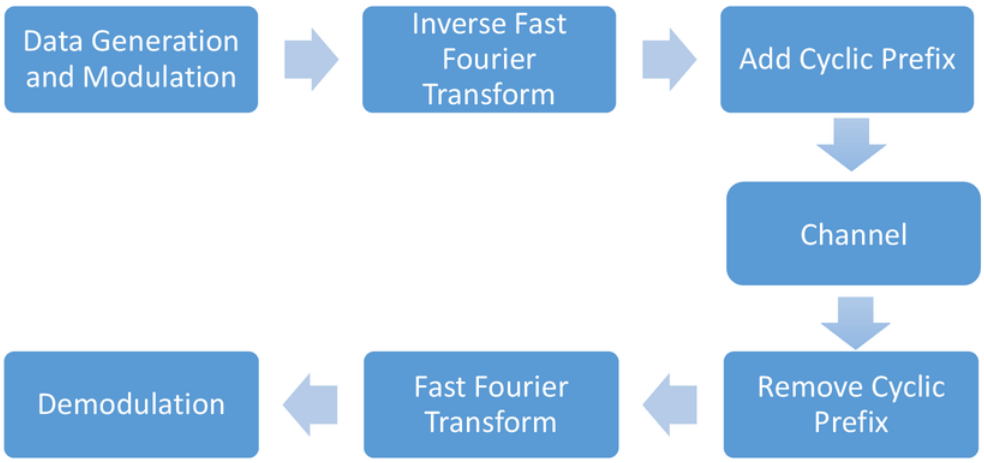

Figure.8 Block Diagram OFDM Scheme

Each of these sections uses the mathematical understanding of OFDM and links together to create a random signal. The output will vary across as Binary Phase-Shift Keying (BPSK) and

Quadrature Phase-Shift Keying (QPSK) comparing symbol error rate. The entirety of this code linking the sections together is seen in appendix.2 and is explained in sections below.

1. **Data Generation and Modulation [9]**

The data generated to be transmitted over OFDM system is randomly generated at length N and initially set as BPSK.

```
am = [-1,1];
M = 2;
dat_ind = ceil(M*rand(1,N));
data = am(dat_ind);
```

It can be set to QPSK if modified to the below to allow for 4 phases instead of 2.

```
am = [1,1i,-1,-1i];
M = 4;
dat_ind = ceil(M*rand(1,N));
data = am(dat_ind);
```

2. **Inverse Fast Fourier Transform [9]**

The IFFT is set at N points and is quite simple due to Matlab's inbuilt functions.

```
data_t = ifft(data);
```

3. **Add Cyclic Prefix [9]**

After this, the last few data points are repeated at the beginning to create the CP with CP_length symbols at the beginning of each data block.

```
data_cp = [data_t(end-CP_length+1:end), data_t];
```

4. **Channel [9]**

The transmission occurs over frequency selective fading channels modelled as Finite Impulse Response (FIR) filters of order L. The FIR filter is the output of a channel computed by filtering input signal with the channel and then adding noise to it. The channel, h, is assumed to have normalised sub-channels drawn from a Rayleigh distribution.

```
h = complex(randn(L+1,1), randn(L+1,1))*sqrt(0.5/(L+1));
```

Noise is then generated to be added to the transmission over the zero-mean complex gaussian channel.

```
noise = complex(randn(1,Total_length), randn(1,Total_length))*sqrt(0.5/N);
```

5. Remove Cyclic Prefix [9]

The CP is then removed from received data, discarding the CP_length symbols.

```
rec_sans_cp = rec(CP_length+1:end)
```

6. Fast Fourier Transform [9]

Following this, the data is extracted by discarding the CP transformed into the frequency domain, equalised to account for the channel and to yield the final received signal. FFT is then used to convert the channel into the frequency domain to normalise the number of sub-channel.

```
rec_f = fft(rec_sans_cp); % FFT
h_f = sqrt(rho)*fft(h,N); % Equivalent channel on each subcarrier
```

7. Demodulation [9]

After this, the received signal is then demodulated using maximum likelihood (ML) estimation on the received symbols to obtain an estimate of the transmitted signal. For BPSK:

```
det1 = abs(rec_f+h_f).^2; % Calc the Euclidean dist assuming -1 was
transmitted
det2 = abs(rec_f-h_f).^2; % Calc the Euclidean dist assuming +1 was
transmitted
det = [det1, det2]; % Concatenating the two vectors
[min_val, ind] = min(det, [], 2); % Find the symbol the received signal is
closest to
dec = 2*((ind-1)>0.5)-1; % BPSK decoding
```

And for QPSK:

```
det1 = abs(rec_f-am(1)*h_f).^2; % Calc the Euclidean dist assuming 1 was
transmitted
det2 = abs(rec_f-am(2)*h_f).^2; % Calc the Euclidean dist assuming +1i was
transmitted
```

```
det3 = abs(rec_f-am(3)*h_f).^2; % Calc the Euclidean dist assuming -1 was
transmitted
det4 = abs(rec_f-am(4)*h_f).^2; % Calc the Euclidean dist assuming -1i was
transmitted
det = [det1, det2, det3, det4]; % Concatenating the vectors
[min_val, ind] = min(det, [], 2); % Find the symbol the received signal is
closest to
dec = am(ind); % Generating the decoded symbols
```

In order to bring all of this together and achieve a result to compare with, a Monte-Carlo simulation is performed to estimates the Bit Error Rate (BER) of the received signals [9]. BER refers to the probability of error P_e occurring in the transmitted time interval. In OFDM, it is expressed as the mean of the probability of error of all the individual subcarriers as shown in (1) [9].

$$P_e = \frac{1}{N}\sum_{k=1}^{N} P_e[k] \tag{1}$$

$P_e[k]$ refers to the channel dependent on the probability of error in the k^{th} subcarrier [9]. This would change depending on the phases used. BPSK is shown in (2) and QPSK in (3).

$$P_e[k] = Q(\sqrt{2y_k}) \tag{2}$$

$$P_e[k] = 2Q(\sqrt{y_k}) - Q^2(\sqrt{y_k}) \tag{3}$$

y_k refers the instantaneous Signal-to-Noise Ratio (SNR) and Q is the Q-function. Figure.9 below shows the output of the code used in appendix.2 for the two modulation signals, BPSK and QPSK. The SNR of the channels is varied and the probability of error is estimated for each case of SNR [9]. As expected, the symbol error rate of the QPSK system performs lower than the BPSK case due to the extra phases.

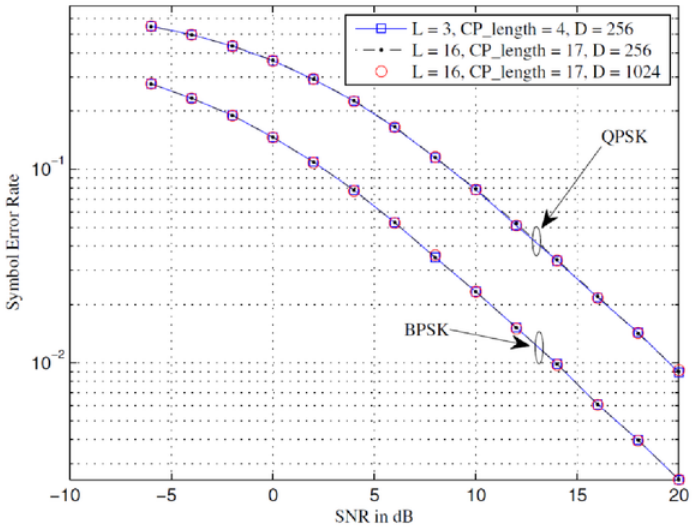

Figure.9 OFDM BER vs SNR in QPSK and BPSK [9]

As the length of the CP reduces, Inter-Block-Interference (IBI) increases and causes deterioration in performance. GFDM does not use block-based transmission and only uses one cyclic prefix per group of symbols so a direct improvement in performance can already be estimated.

4.2 GFDM Signal Simulation

Continuing from the OFDM simulation, another simulation is performed in order to display a GFDM signal. A random GFDM and OFDM signal are created with equal length to compare with each other. This simulation is constructed on the transmission side of the GFDM theory is displayed in a block diagram in figure.10. It is also broken into parts and explained in each section similar to the OFDM simulation.

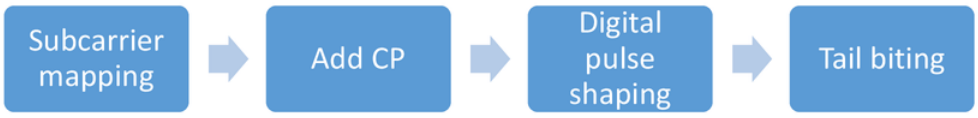

Figure.10 GFDM Transmitter Scheme

Starting with the subcarrier mapping, there are a few input parameters that make up the GFDM block that are initialised as per the below:

- $\mathrm{K} = 512$ (Number of subcarriers, where the set is only from 100:200)
- $\mathrm{Mon} = 14$ (Number of sub-symbols)

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
gfdm = get_defaultGFDM('TTI');
gfdm.K = 512;
gfdm.Kset = 100:200;  % Only allocate some subcarriers
gfdm.Mon = 14;


ofdm = gfdm;
ofdm.M = 1;
ofdm.Mon = 1;
ofdm.L = ofdm.K; %match the sub symbol to 1 subcarrier for frequency domain


nB = 3; % Number of GFDM blocks
```

These simulations are created over 3 GFDM blocks to ensure consistency over just 1 block. These default parameters give the value of $\mathrm{N} = \mathrm{KM} = 7168$ where each subcarrier carries M sub-symbols giving total parallel sub-streams N. In order for the simulation to remain consistent the size of this block must remain the same.

After this the CP is added. To maintain simplicity in this simulation, the CP is not needed for transmission.

```
assert(~isfield(gfdm, 'Ncp') || gfdm.Ncp == 0);
```

Following this is the digital pulse shaping filter. There are more input parameters that may be varied such as the type of filter used and the roll-off factor used:

- Filter: RC filter
- $a = 0.1$ (Roll-off factor)

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
gfdm.pulse = 'rc';
gfdm.a = 0.1;

ofdm.pulse = 'rc_td';  % use RC_TD with roll-off 0 to make a rectangular
filter
ofdm.a = 0;
```

This code contains the RC filter being used in GFDM and a rectangular filter for OFDM. The next section in this document will explain what the RC filter is doing along with the other filters mentioned in this project.

Lastly is the tail biting properties. Once again, since this simulation is only based on a simple transmission, the tail biting effects will vanish resulting in a regular FBMC burst so they are not required.

Using these parameters, a simulation is created to estimate the Power Spectral Density (PSD) of a GFDM and OFDM signal. Power spectral density refers to the frequency response of a signal. It demonstrates where the average power is distributed as a function of frequency. The formula (1) explains how PSD is calculated.

$$\mathrm{PSD} = |\mathrm{F}(\omega)| \tag{1}$$

After using the above GFDM default parameters, the PSD calculation may begin. Firstly, the Fourier transform of the signal is taken, then it's magnitude is taken.

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
% Beginning of PSD
f = linspace(-gfdm.K/2, gfdm.K/2, 2*length(sGFDM)+1); f = f(1:end-1)';
plot(f, mag2db(fftshift(abs(fft(sOFDM, 2*length(sOFDM)))))/2, 'b');
plot(f, mag2db(fftshift(abs(fft(sGFDM, 2*length(sGFDM)))))/2, 'r');
```

Figure.11 displays the average power plotted on a dB scale against frequency. Both GFDM and OFDM are displayed on the same graph where GFDM is red and OFDM is blue. The length to compare each PSD is in the 100-200 subcarriers set. Therefore, the comparison of each transmission signal is be seen in the 100Hz to 200Hz bandwidth bracket on the frequency axis below.

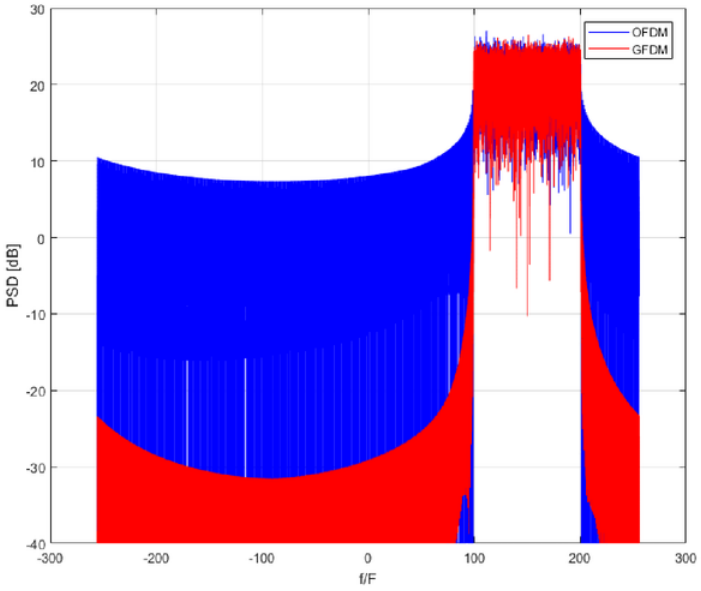

Figure.11 Power Spectral Density Vs frequency in OFDM and GFDM

The spectrum is slightly higher in a GFDM which concludes that both emit a strong signal. OFDM already has a high spectral density so for GFDM to be able to match it and overcome it provides an advantage for a proposed network framework. When looking at what's outside the bandwidth, a clear difference in OFDM and GFDM is already seen. This is where the OOB emissions occur and will be inspected in detail through Matlab simulations.

4.3 GFDM Filter Representations

The idea of the optimal pulse-shaping filter is to tailor a signals waveform. In GFDM this would occur in the transmitter and receiver whilst the waveform travels over the current channel condition [24]. This section will explain each of GFDM's transmission filters representations in Matlab.

The filtering of a transmission signal occurs after the subcarrier mapping is completed. The filters used in this document are simulated then modified with the aim of reducing OOB emissions. As already mentioned, the transmitter filters focused on this project are:

1. RC
2. RRC
3. Dirichlet
4. Xia 1st/4th order

Each of these filters is used as a parameter listed in the GFDM simulation code.

```
gfdm.pulse = 'rc';
gfdm.pulse = 'rrc';
gfdm.pulse = 'dirichlet';
gfdm.pulse = 'xia1st_td';
gfdm.pulse = 'xia4th_td';
```

Below shows their Matlab implementation along with a description of their mathematical properties. The code detailing all of these GFDM filters in Matlab is seen in appendix.4.

The number of guard carriers are set to 1 across all filters. The OOB leakage is minimised when 1 guard carrier is used [25]. Altering number of guard carriers will change OOB leakage but will not be changed between the filters in this code in order to maintain consistency.

1. RC

Raised cosine pulses are often used instead of sinc pulses, because sinc pulses have an impulse response of infinite duration [13]. They are simulated in the time domain and adhere to the required roll-off factor being less than 1.

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
function g = rc(M, K, a)
% RC - Return Raised Cosine filter (time domain)
%
t = linspace(-M/2, M/2, M*K+1); t = t(1:end-1); t = t';

g = (sinc(t) .* cos(pi*a*t) ./ (1-4*a*a*t.*t));
g = fftshift(g);
g(K+1:K:end) = 0;

g = g / sqrt(sum(g.*g));
```

2. RRC

Root raised cosine is based on the raised cosine but with an extra root taken. The relationship between subcarriers and sub-symbols remains the same across the two due to the similar structure.

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
function g = rrc(M, K, a)
% RRC - Return Root Raised Cosine filter (time domain)
%

t = linspace(-M/2, M/2, M*K+1); t = t(1:end-1); t = t';

g = (sin(pi*t*(1-a))+4*a.*t.*cos(pi*t*(1+a))./(pi.*t.*(1-(4*a*t).^2));
g(find(t==0)) = 1-a+4*a/pi;
g(find(abs(t) == 1/(4*a))) = a/sqrt(2)*((1+2/pi)*sin(pi/(4*a))+(1-
2/pi)*cos(pi/(4*a)));

g = fftshift(g);
g = g / sqrt(sum(g.*g));
```

3. Dirichlet

The Dirichlet pulse is defined by a perfect rectangular function in the frequency domain. Dirichlet pulse are therefore be regarded as a "reversed OFDM" as the rectangle pulse shaping is applied in the frequency domain instead of the time domain [25]. Being a perfect rectangular function, the Dirichlet pulse is a corner case of the Xia pulse when the roll-off goes to zero, hence, no need for a roll-off factor input [24].

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
function g = dirichlet(M, K)
% DIRICHLET - Return Dirichlet Kernel

o = ones(1, M);
z = zeros(1, K*M-M);

G = [o z];
G = circshift(G, [0, -floor(M/2)]);

g = ifft(G);
g = g / sqrt(sum(abs(g).^2));
g = g';
```

4. Xia 1st/4th order

The Xia pulses, are a family of real non-symmetric pulse shaping filters that are defined by a roll-off factor and a roll-off function in the frequency domain [24]. The most common are 1st order and 4th order which are both looked at in the simulations.

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
function g = rampbasedfir(M, K, a, pulse)
% rampbasedfir - Return Ramp Based filter
```

```
switch pulse
    case   'r1st_fd', type= 0;
    case    '1st_fd', type= 1;
    case   'r4th_fd', type= 2;
    case    '4th_fd', type= 3;
    case    'rrc_fd', type= 4;
    case     'rc_fd', type= 5;
    case 'rrc4th_fd', type= 6;
    case  'rc4th_fd', type= 7;
    case 'xia1st_fd', type= 8;
    case 'xia4th_fd', type= 9;
    case   'r1st_td', type=10;
    case    '1st_td', type=11;
    case   'r4th_td', type=12;
    case    '4th_td', type=13;
    case    'rrc_td', type=14;
    case     'rc_td', type=15;
    case 'rrc4th_td', type=16;
    case  'rc4th_td', type=17;
    case 'xia1st_td', type=18;
    case 'xia4th_td', type=19;
    otherwise error('Unknown transmitter filter!');
end;


if type<10
[rise,fall]=get_ramp(M,a,type);
g=ifft(([fall;zeros((K-2)*M,1);rise]));%sum(g.^2)=1
else
    if M>1,
        type=type-10;
        g=zeros(M*K,1);
        [rise,fall]=get_ramp(K,a,type);
        g(1:K)=fall;
        g(end-K+(1:K))=rise;
    else
        g=ones(K,1);
    end;
end
g=g/sqrt(sum(abs(g).^2));
```

4.4 OOB Emission Measurements

In order to measure the OOB emissions across various GFDM filters, numerous steps are required in order to achieve a result. Due to this, the time taken to complete this section was the longest in the project. The OOB measurements are broken into three sections, each progressing from the last:

1. Rectangular pulse
2. Sine wave
3. GFDM

The first section starts with a simple rectangular pulse and calculates the OOB emissions from single pulse. Building on top of this, the aim is to then move onto a slightly more complicated sine wave. In doing so, the code will evolve and calculate the energy under the sine wave. As a result, the code will once again be modified to suit an actual GFDM waveform. In each of these steps, the total energy is calculated in both the frequency and time domain to ensure correct methodology. This methodology is derived from Parseval's theorem below. Parseval's theorem states that sum/integral of the square of a function is equal to the sum/integral of the square of its Fourier transform in the frequency domain.

$$E = \int_{-\infty}^{\infty} |f(t)|^2 dt = \int_{-\infty}^{\infty} |F(\omega)|^2 d\omega \quad (1)$$

1. Rectangular Pulse

By starting with a simple rectangular pulse the energy calculation under the pulse is predicted and calculated. The dingle pulse occurs from 0 to 1 with amplitude 2, so an energy result of 4W is expected from the code in both the time and frequency domain and is what will be tested. The rectangular pulse is first defined in the time domain below.

```
%RECTANGULAR PULSE
Fs=100; %change sampling frequency
t0=1; %change end length
t = 0:1/Fs:t0;
X=@(t) 2*rectangularPulse(0,t0,t); %Change amplitude
L = length(t);
Lsig = length(X);

%ENERGY TIME DOMAIN
Energytime=(abs(integral(X,-Inf,Inf))^2) %parsevals equation
```

The sampling frequency refers to the number of samples to be taken across 1 second of the signal. This is set at 100Hz at first to ensure an accurate result and a smooth diagram. The total length of the signal is then set to 1 second. Figure.12 shows 100 samples of the signal over the 1 second time axis. Then the rectangular pulse is then created. The pulse is multiplied by 2, allowing the amplitude to be 2 and the pulse is set to rise at 0 and fall and 1. This result is achieved in figure.12 below. The energy calculated is now in the form of the right side of

Parseval's equation seen above in (1). It took the absolute value of the integral over the entire rectangular pulse and achieved an energy result of 4W as expected.

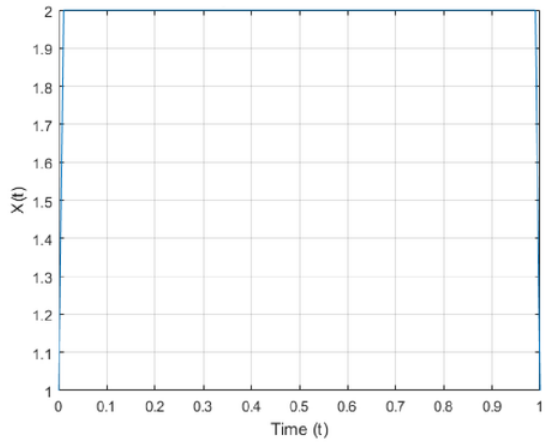

Figure.12 Rectangular Pulse in Time Domain

After this, the Fourier transform is taken so that the right side of Parseval's equation can be calculated. The absolute value of this is taken and rounded using the floor function. P1 and P2 allow the pulse to scale on the frequency domain axis. Figure.13 shows the frequency response of the rectangular pulse.

```
%FFT
Y = fft(X(t));
P2 = abs(Y/Lsig);
P1 = P2(1:floor(Lsig/2+1));
P1(2:end-1) = 2*P1(2:end-1);
```

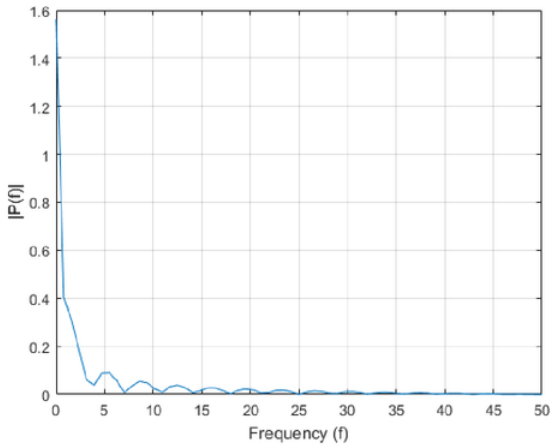

Figure.13 Rectangular Pulse in Frequency Domain

Once the frequency response is achieved it needed to be squared and the area under the curve would give the energy result. As mentioned before the power spectral density demonstrates where the average power is distributed as a function of frequency. This is displayed on a log scale in figure.14.

```
%%PSD
psdx = (1/(Lsig*Fs)) * abs(P1).^2; %parsevals equation
PdBW = 10*log10(Pplot);

%%ENERGY FREQUENCY DOMAIN
Energyfreq = sum(psdx)/Fs % in watts
```

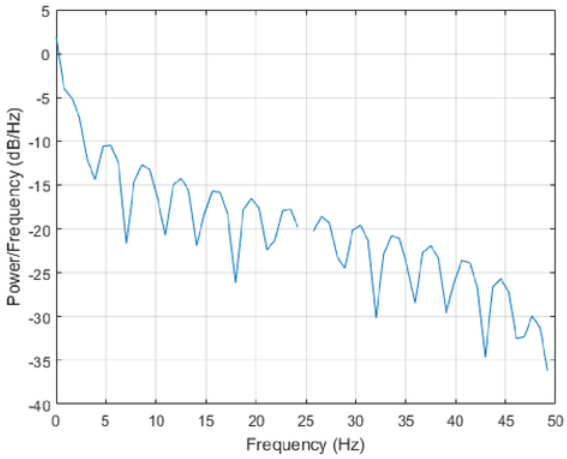

Figure.14 PSD Using FFT of Rectangular Pulse

The PSD is squared and then the integrated over the entire signal. This is completed using Matlab's sum function. The result calculated is also 4W and proves a correct test of the code used. The rest of the Matlab code used to create these figures are seen in appendix.5.

2. Sine Wave

The rectangular pulse has proved successful, so the next step is to move onto a sine wave. This time, a sine function of $\sin(2\pi t)$ from 0 to 1 is created in a modified version on the rectangular pulse code but with a predicted result of 0.5W. This code is improved to become more efficient as the complexity progressed. The sampling frequency is lowered to a n^2 factor and the square of the absolute value of the curve is taken before calculating the time domain integral. The sine wave in the time domain is plotted in figure.15.

```
%TIME DOMAIN SINGNAL
Fs=16; % sampling frequency
t0=1; % end length
```

```
t = 0:1/Fs:t0-1/Fs;
X=@(t) sin(2*pi*t); %signal
absX2=@(t) abs(X(t)).^2;
Ltim = length(t); % time length
Lsig = length(X(t)); % signal length

%ENERGY TIME DOMAIN
Energytime=integral(absX2,0,t0) % time domain parsevals
```

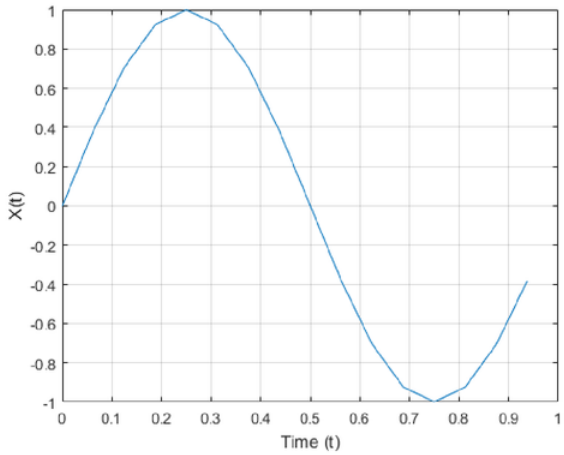

Figure.15 Sine Wave in Time Domain

Again, the integral is taken over the entire wave and the energy calculated is equal to 0.5W as predicted. The next step is to plot the sine waves frequency response. This time zero padding is created for the frequency domain signal as it simplifies the code. The Fourier transform is then taken, followed by the absolute value. The sine wave in the frequency domain is plotted in figure.16.

```
%PLOT FFT
n = 2^nextpow2(Ltim); %zero padding
Y = fft(X(t),n); % FFT
f = Fs*(0:n-1)/n; % freq axis
P = abs(Y/n); % P for parsevals eqn
```

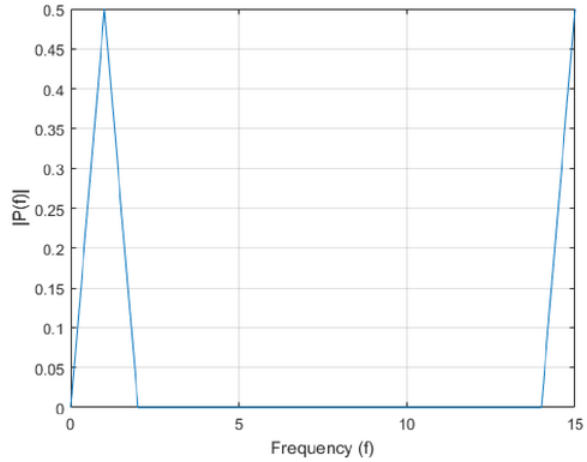

Figure.16 Sine Wave in Frequency Domain

Now that the frequency domain is correctly displayed, the right side of Parseval's equation can be calculated. The absolute value of the frequency response is taken and the axis is converted to a dB scale. The PSD of this sine wave is displayed in figure.17 below. Once again the sum function is used for integral of the PSD to calculate the energy under the curve.

```
%%PSD
P = abs(Y/n); % P for parsevals eqn
PdBW = 10*log10(P);

%%ENERGY FREQ DOMAIN
psdf = P.^2*(n/Fs);% n/Fs to allow for non n^2 sampling freq
EnergyFreq = sum(psdf) % freq domain parsevals
```

The results from the energy under the frequency curve is 0.5W as predicted. Parseval's theorem has been implemented correctly here and needs to be replicated in a GFDM scenario. The accompanying Matlab code for the sine wave is seen in appendix.6.

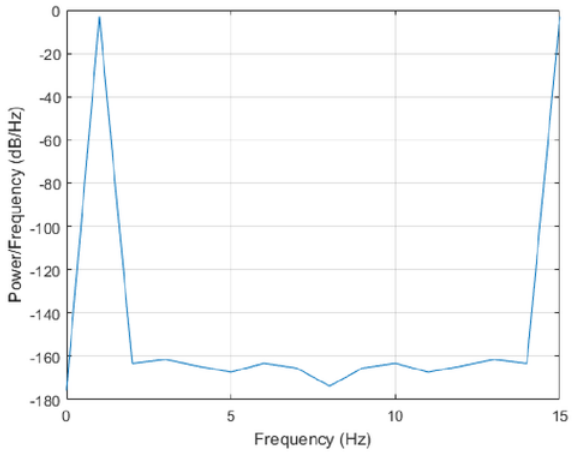

Figure.17 Sine Wave Energy

3. GFDM Wave

The last step is to move onto a GFDM wave by combining the sine energy code and the GFDM Matlab code in section 4.2. This simulation includes the levels of GFDM OOB emissions across all the filters.

The GFDM Matlab code already displays the PSD of GFDM, so a step backward is needed in order to calculate the frequency response, p(f). To create the right side of the Parseval's equation, the dB scale is removed and the frequency response is squared. Combining this with the sine energy code, a total energy result under the entire signal can now be achieved.

This simulation has changed the original 100 to 200 subcarriers from 100 to 120 subcarriers with a roll-off of 0.9 for all filters. In doing so a bandwidth of 20Hz is set to better calculate OOB and is displayed over the frequency axis of -300Hz to 300 Hz. First, the frequency axis mentioned is set up to display the entire waveform for each filter.

```
%% Plot the resulting P(f)
f = linspace(-gfdm1.K/2, gfdm1.K/2, 2*length(sGFDM1)+1); f = f(1:end-1)';
%frequency axis in frequency domain
```

Then the absolute value is taken as well as the fast Fourier transforms of each of the filters as plotted in figure.18 below.

```
RC  = fftshift(abs(fft(sGFDM1, 2*length(sGFDM1))))/2; %RC signal to be
plotted
RRC = fftshift(abs(fft(sGFDM2, 2*length(sGFDM2))))/2; %RRC
DIR = fftshift(abs(fft(sGFDM3, 2*length(sGFDM3))))/2; %Dirichlet
XIA1 = fftshift(abs(fft(sGFDM4, 2*length(sGFDM4))))/2; %Xia 1st
XIA4 = fftshift(abs(fft(sGFDM5, 2*length(sGFDM5))))/2; %Xia 4th
```

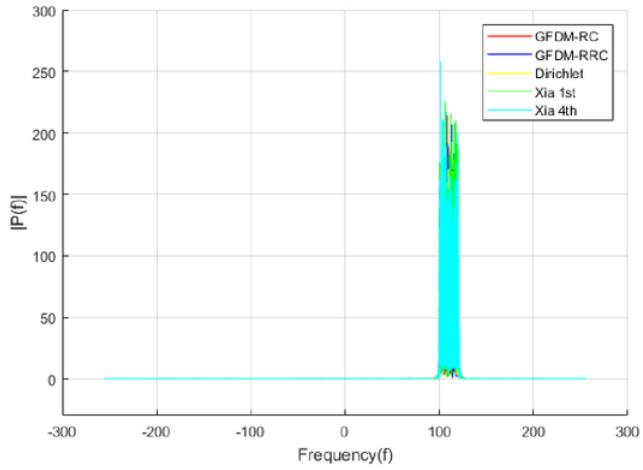

Figure.18 GFDM Frequency domain all filters

After the frequency domain is displayed, much like the sine and the rectangular examples, the PSD is calculated and is plotted in figure.19. The code has been improving and now used magnitude to dB function for the PSD.

```
%Plot PSD
plot(f, mag2db(RC), 'r'); %take magnitude and in dB scale
plot(f, mag2db(RRC), 'b');
plot(f, mag2db(DIR), 'y');
plot(f, mag2db(XIA1), 'g');
plot(f, mag2db(XIA4), 'c');
```

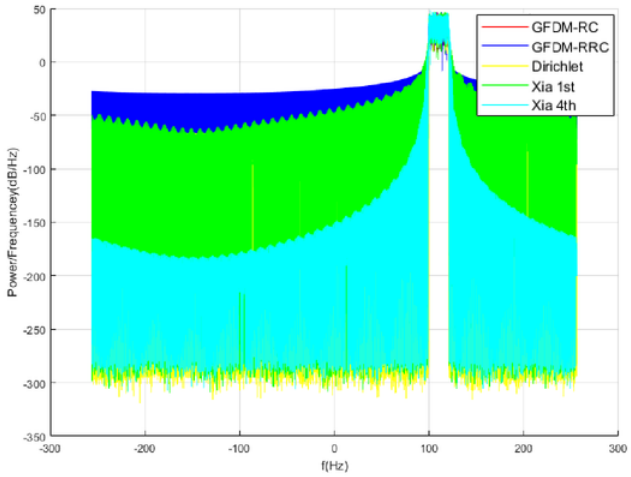

Figure.19 PSD of all GFDM filters

The final result in which this section has been aiming to achieve may now be calculated. The energy under these filters is calculated by first squaring the absolute values and then using an integration approximation, the trapezoidal rule, to find the energy. The energy required however is not under the entire wave, yet rather in the OOB domain only. This OOB domain is calculated using figure.20.

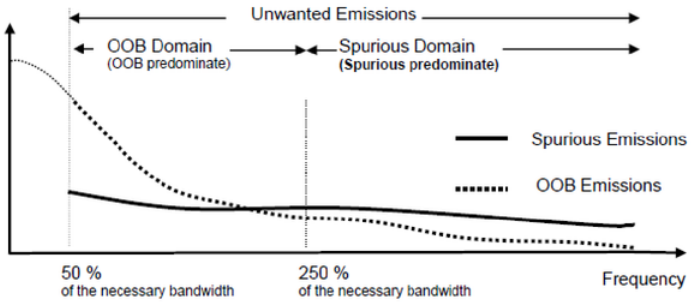

Figure.20 OOB Domain Definition [22]

The bandwidth of the GFDM wave simulated is equal to 20Hz as the allocated subcarriers occur between 100Hz to 120Hz signal. Using the 50% of the necessary bandwidth from figure.20 the beginning of the OOB domain is set at 130Hz. Similarly, the end of the OOB domain is set at 250% of the necessary bandwidth, 170Hz. These are the limits in which the

trapezoidal rule calculates its energy from. The entirety of this code for this is seen in appendix.7

```
%% Energy in frequency domain
psdfRC = RC.^2; %end of parsevals equation
psdfRRC = RRC.^2;
psdfDIR = DIR.^2;
psdfXIA1 = XIA1.^2;
psdfXIA4 = XIA4.^2;

%oob domain
OOBRC = trapz(f>130&f<170,psdfRC) %use trapezoidal rule as approx
integration between OOB domain
OOBRRC = trapz(f>130&f<170,psdfRRC)
OOBDIR = trapz(f>130&f<170,psdfDIR)
OOBXIA1 = trapz(f>130&f<170,psdfXIA1)
OOBXIA4 = trapz(f>130&f<170,psdfXIA4)
```

5 Results and Discussion

This section is broken into three sections.

1. GFDM OOB Emissions
2. GFDM Filters
3. Comparison of GFDM Filters with OFDM

The first section demonstrates the effects each of the filters have on OOB emissions. The second section then optimises each filter based on varying input parameters. The final section compares the optimised filters to the original OFDM simulations to achieve results which support GFDM's candidacy. To conclude there is a discussion on the benefits GFDM provides from the attained Matlab simulation results.

The purpose of these results are to obtain an answer to which transmit filters has the lowest OOB emissions and under which parameters. In order to maintain consistency with results, the initial input parameters to be varied later are set as:

- $\mathrm{a} = 0.9$ (high roll-off factor)
- $\mathrm{K} = 512$ (subcarriers from a smaller set of 100 to 120)
- $\mathrm{Mon} = 14$ (maximum sub-symbols)

5.1 GFDM OOB Emission Performances

As defined before, there is only a specific domain in which OOB emissions exist. The way in which OOB emissions are calculated in this project are from the frequency side of Parseval's theorem.

$$\mathrm{E} = \int_{-\infty}^{\infty} |\mathrm{f(t)}|^2 \mathrm{dt} = \int_{-\infty}^{\infty} |\mathrm{F}(\omega)|^2 \mathrm{d}\omega$$

The frequency responses of each of the filters have already been simulated in the OOB domain. The absolute value and square of these of the results is then integrated over the defined OOB domain. This domain is taken on the right side of the signal between 130Hz and

170hHz due to the definition of the OOB leakage. The lower the OOB emissions are, the lower the resulting interference and hence an increase in performance. With this in mind, the simulations yielded the below results across the different filters with the default parameters. In order to ensure consistency, a mean of 100 samples of each filter is taken. This is due to the random function used in the simulation of each wave to be realistic. The results are obtained below in table.1.

Transmit Filter Type	OOB emissions in OOB domain (W)
RC	2.3251e-05
RRC	0.0065
DIR	1.5562e-04
XIA1	2.5866e-05
XIA4	3.6719e-11

Table 1. Mean of 100 OOB emission samples

This result shows large differences across the various pulse shapes when using default parameters. Xia 4th order has the lowest emissions by a great margin. When compared to the other filters, raised cosine, Dirichlet and Xia 1st order perform similarly and Root raised cosine performs poorly. Figure.21 displays these results around the 120Hz to 130Hz area which is just before the OOB domain. This demonstrates the excess leakage occurring right after the allocated bandwidth.

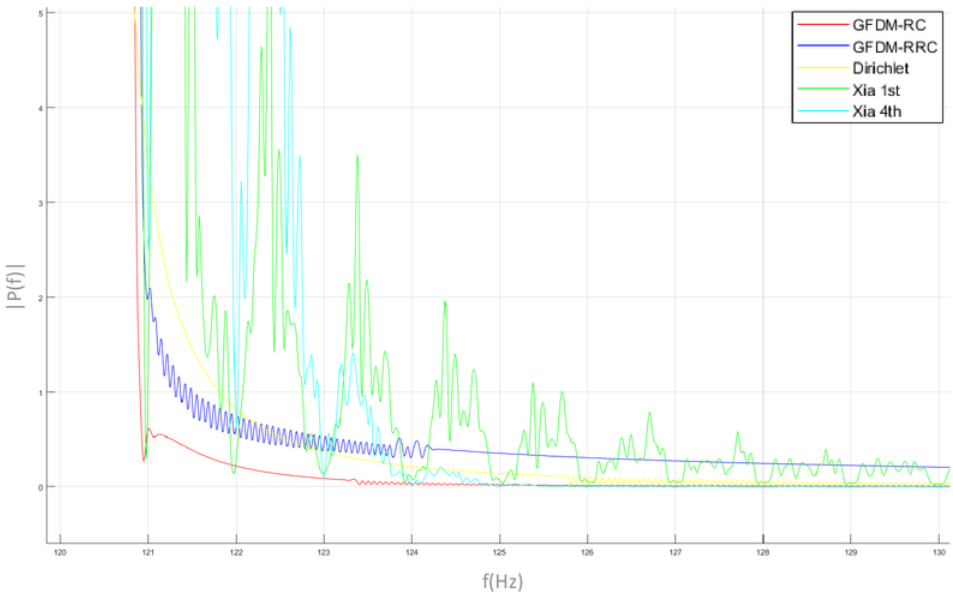

Figure.21 Non-OOB domain energy

Figure.22 shows the continuation of this leakage across each filter. This diagram is now inside of the OOB domain. It is clear here that average results achieved in table.1 reflect those plotted below.

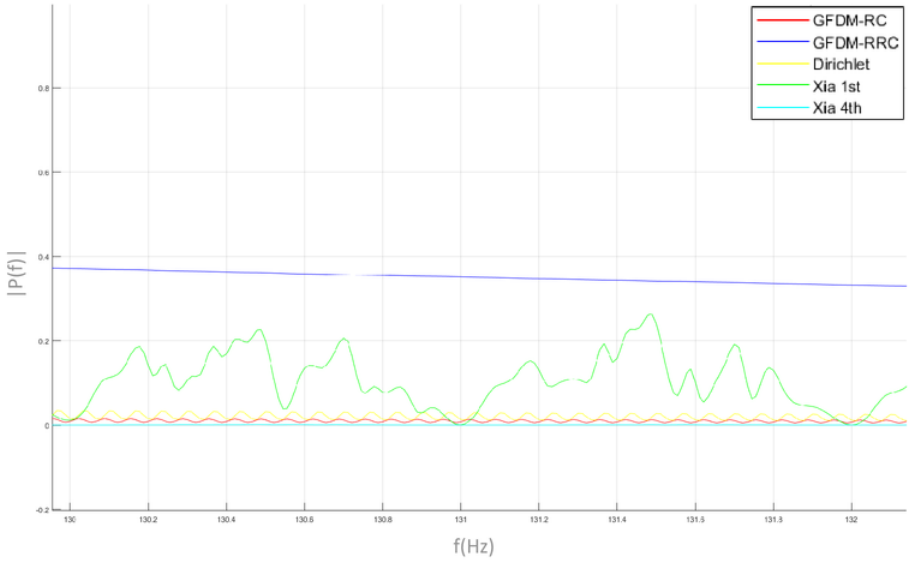

Figure.22 OOB domain energy

Figure.22 displays the varying OOB emissions across the filters inside of the OOB domain. The next step after showing the OOB emissions is to adjust the roll-off factor to see if any changes in the OOB domain occur.

5.2 GFDM Filter Performances

Each filter has achieved OOB emission results using the default input parameters below. The next step is to vary each of the default parameters chosen and see how it relates to OOB emissions.

- $\alpha = 0.9$ (roll-off factor)
- $\mathrm{K} = 512$ (subcarriers from 100 to 120)
- $\mathrm{Mon} = 14$ (sub-symbols)

Starting off by varying the roll-off, a possible increase or decrease in OOB emissions in each of the filters is predicted to occur. After varying the roll-off factor to three different values, an average of 100 samples are taken. The results obtained are in table.2.

Filter	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$ (default)
RC	1.3891e-04	7.1490e-05	2.3251e-05
RRC	0.0025	0.0070	0.0065
DIR	1.6417e-04	1.4398e-04	1.5562e-04
XIA1	0.0051	7.9150e-05	2.5866e-05
XIA4	0.0323	7.8890e-09	3.6719e-11

Table.2 Roll-off factor variation

The roll-off factor varies across filters. Altering the roll-off factor has further increased or reduced OOB emissions depending on the filter. With this knowledge, a new test is taken with another 100 sampled averages. This time the roll factor increases from 0.1 to 0.9. This is done to see if there is a direct relationship between roll-off factor and OOB emissions.

Filter	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
RC	1.57E-04	0.0016	9.49E-05	0.0032	7.10E-05	4.95E-05	4.45E-05	0.0026	1.80E-05
RRC	0.0024	0.0056	0.0069	0.0075	0.0084	0.0093	0.0090	0.0085	0.0065
DIR	1.55E-04	1.56E-04	1.60E-04	1.62E-04	1.56E-04	1.58E-04	1.54E-04	1.56E-04	1.56E-04
XIA1	0.0075	7.39E-04	3.04E-04	1.28E-04	8.76E-05	6.23E-05	3.76E-05	3.01E-05	2.61E-05
XIA4	0.034	0.0036	8.62E-05	1.94E-06	7.82E-09	3.02E-09	7.14E-10	2.00E-10	3.74E-11

Table.3 Roll-off factor variation with an increase of 0.1

Varying the roll-off factor yielded some interesting results. RC, XIA 1st order and Xia 4th order all show a decrease in OOB emissions as roll-off is increased. This is caused by the abrupt transition between the GFDM blocks when the transmit filter is not ISI-free [25]. RRC filter performed poorly overall yet had an optimal point at 0.1 roll-off. Once again Dirichlet performs the same due to its construction not being based on a roll-off factor. The Dirichlet pulse shaping filter creates no self-interference but has a OOB radiation comparable to the RC [25]. These results are displayed in figure.23 below on a dB scale for comparison with an added trend line.

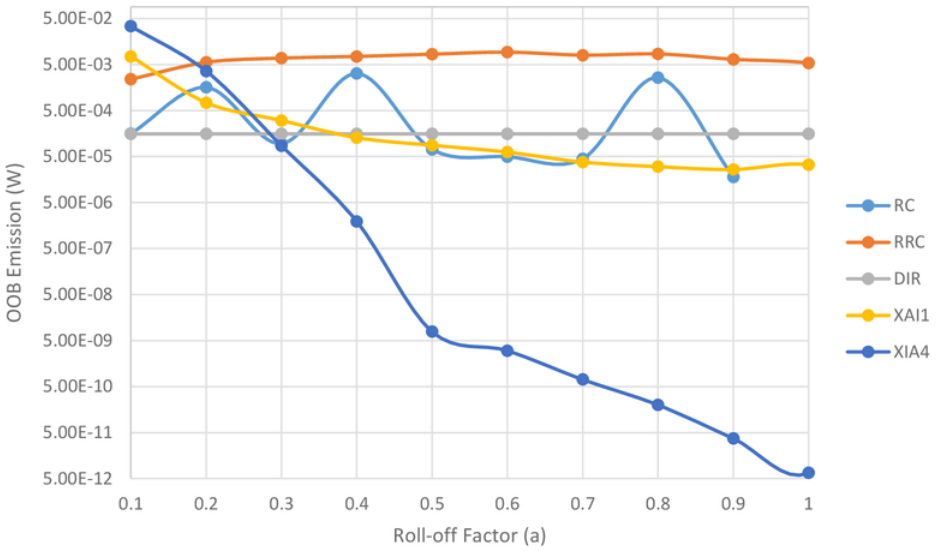

Figure.23 Roll-off factor variation trend lines

As a conclusion from these results, XIA 4th order shows dominance in having lowest OOB emissions from the 0.4 roll-off to the full roll-off mark. Initially, at 0.1 roll-off, Xia 4th order had the highest emissions of all the filters.

Another quantity that will be varied in the GFDM signal is the number of subcarriers and sub-symbols. As mentioned before in GFDM theory, $\mathrm{N} = \mathrm{MK}$ is the number of subcarriers multiplied by the number of sub-symbols. By keeping N the same and varying both the subcarriers and sub-symbols a test is created to see if there is a change in OOB emissions. The default values are set to keep $\mathrm{N} = 7168$ where $\mathrm{M} = 14$ and $\mathrm{K} = 512$.

Filter	M*K(1*7168=7168)	M*K (8*896=7168)	M*K (14*512=7168)
RC	1.29E-09	1.65E-09	1.82E-05
RRC	4.11E-06	6.52E-06	0.0071
DIR	2.72E-05	4.23E-05	1.22E-04
XIA1	2.73E-05	2.98E-05	2.70E-05
XIA4	4.74E-11	4.62E-11	3.56E-11

Table.4 subcarriers vs sub-symbols

There is a slight change in emissions for Dirichlet and XIA filters as the sub-symbols are increased. Contrastingly, there are large changes in the RC and RRC filters when the sub-

symbols are decreased. If the lowest possible M is set, RC and RRC emissions are reduced. This now allows RC to be a close second choice for lowest OOB emissions. The results are more clearly displayed on the log scale in figure.24 where the number of sub-symbols are rounded to ensure N is kept constant.

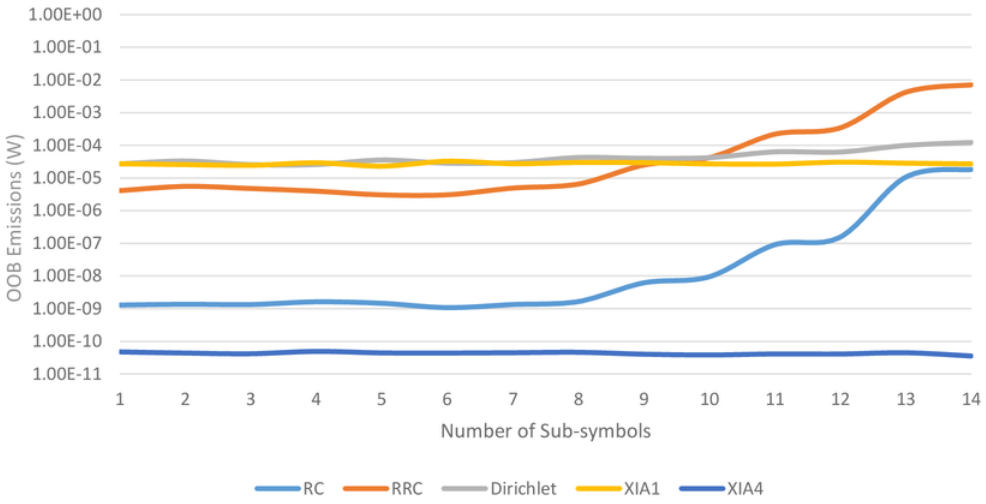

Figure.24 OOB emissions Vs Number of Sub-symbols

It is clear from figure.24 that number of sub-symbols has a large influence on RC and its subsequent RRC filter.

The results obtained from this section of the document detail that setting the below input parameters results in the greatest GFDM OOB minimisation.

1. Filter: XIA 4th filter
2. Roll-off factor: approach full roll-off ($\alpha = 1$)
3. Sub-symbols: Maximum
4. Subcarriers: Minimum (as a result of maximum sub-symbols)

This can conclude that the most efficient filter to use in terms of OOB leakage reduction is Xia 4th filter with a high roll-off factor. Maximising the number of sub-symbols in this filter will yield optimal results in terms of lowest OOB leakage. On the other hand, if a low roll-off is to be used a RC filter would be recommended. For practical applications, the RC with lower α in the range of $0 \leq \alpha \leq 0.2$ is preferable [7]. In this case, minimising the number of sub-symbols will yield optimal results in terms of lowest OOB leakage in the RC filter.

5.3 Comparison of GFDM Filters with OFDM

After successfully optimising parameters across GFDM filters a comparison is created so illustrate how effective these results are against OFDM. The code from OOB Matlab is altered once again now to allow the original OFDM simulations. Appendix.8 displays this code. The frequency response and PSD of OFDM and GFDM are once again calculated below. Figure.25 displays the GFDM frequency response of all filters along with OFDM and figure.26 displays the PSD of all GFDM filters along with OFDM.

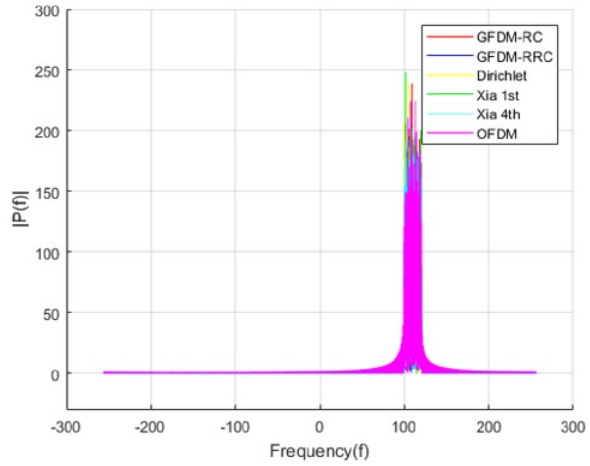

Figure.25 Frequency Response GFDM + OFDM

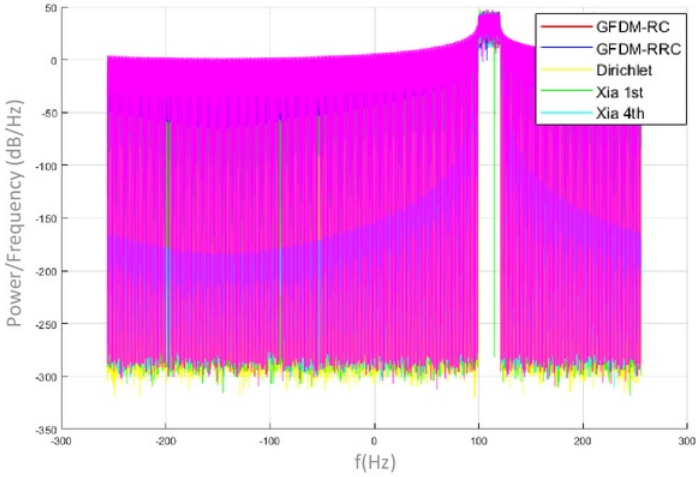

Figure.26 GFDM filters and OFDM comparison

From the figures it is not very clear how different OFDM is from the proposed GFDM filters. The graphs have many overlapping waveforms as there are 6 wave comparisons. This is an undesired result and is decided to be displayed in a different form. A numerical result is tabulated below to better illustrate results. OFDM OOB emissions are set at 0.9 roll-off and maximum sub-symbols are used. These parameters are chosen to be the comparison point with OFDM as the lowest emissions occur under in the Xia 4^{th} order filter with these conditions.

Filter	OOB Emissions (W)
RC	1.7910e-05
RRC	0.0044
DIR	1.7609e-04
XIA1	3.5434e-05
XIA4	4.0314e-11
OFDM	0.0140

Table.5 GFDM filters and OFDM OOB comparison

From here it is evident that every filter contains lower OOB emissions than OFDM. RC, Dirichlet and XIA order filters perform exceptionally better here in terms of OOB leakage when compared to OFDM. These results are plotted in figure.27 on a dB scale to further illustrate which filters perform with the best in terms of least OOB emissions.

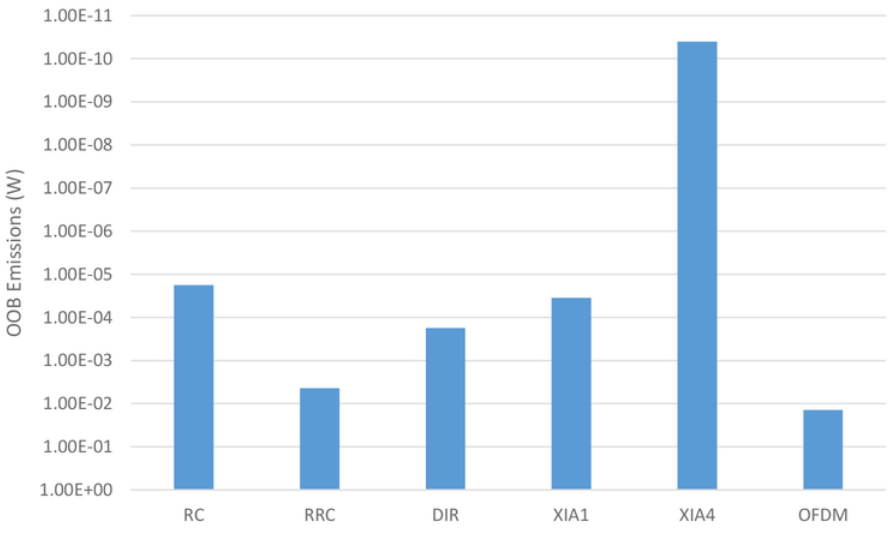

Figure.27 GFDM Filter efficiency compared to OFDM

Figure.27 informs that the lower the bar, the less efficient the waveform. GFDM filters. RRC is lowest performer out of the GFDM filters yet still performs three times better than OFDM. The RC filter displays a OOB leakage that is about 40 dB or 50 dB below that of OFDM [25]. It is clear to say that all of the present pulse shaping filters outperform OFDM regarding OOB radiation [25]

6 Conclusions and Future Work

The main outcome of this project was to optimise and recommend a 5G non-orthogonal waveform. In this project, GFDM's OOB emissions are reduced and the waveform is presented as a solution for the upcoming 5G.

6.1 Conclusions

The work completed in this document explains how GFDM is a prime candidate solution to 5G networks as well as optimising its performance. Currently, intercarrier interference exists in GFDM transmission due to OOB leakage and result in high latency. When considering consumer trends for upcoming networks, low latency is a requirement.

Simulations underwent during this project demonstrate an understanding of the OOB leakages, then develop a way to measure and minimise them. The way in which these OOB emissions are minimised are through specific filtering and an alteration of GFDM input parameters. In doing so, the benefits of GFDM's transmission scheme are calculated and represented. These calculations allowed a direct comparison to that of the current OFDM transmission performance. When compared with each other, GFDM has proven great promise in terms of lower interference and latency.

The exact result in which GFDM has inherits lower interference exists in all transmission filters tested; Raised Cosine, Root Raised Cosine, Dirichlet, Xia 1st order and Xia 4th order. The filters which performed with the best performance in terms of lowest OOB radiation are Raised Cosine and Xia 4th order. This occurs due to their ability to use certain parameter's such as roll-off factor and subcarriers/sub-symbols effectively. These filters are recommended to be used as they benefit the proposal of GFDM as a future candidate waveform in the most efficient way.

6.2 Future Work

There are several possible paths to continue the research of GFDM development as a candidate waveform. Building on the work completed in this project, possible trade-offs may be explored when using optimal transmission filters and parameter settings. This could include a complete GFDM simulation where not only transmission is optimised but also a receiver across a simulated channel is included. This would require usage of CP and tail biting. By simulating GFDM across an AWGN and Rayleigh channel, a more realistic approach may now be illustrated to further support this waveform. Following this, the receive filters MF, ZF and MMSE may also be compared with each other using a similar procedure taken in this project yet include the one-tap equalisation as it already reduces the ISI caused by the filters. Even new interference reduction methods such as windowed GFDM and guard symbol GFDM should be further researched. After all of these possible simulations have been constructed and optimised, a direct comparison to OFDM can be calculated, comparable to what has been completed here. In doing so, the results will further demonstrate the benefits GFDM waveform can provide for the upcoming 5G air interface.

SIC was mentioned in this paper as an interference reduction method. This added complexity occurs on the receiver side of GFDM which was not the focus of this project. SIC along with receiver optimisation is worthy of investigation in future projects when using the research constructed across various filters in this project. This opens up some ideas as the use of these receiver methods could still increase performance in GFDM and other competing interfaces.

There are other issues in GFDM that may be further analysed and optimised. Timing offset is an issue that should be considered in GFDM. There was an initial plan to consider the reduction of timing offset as a relation to OOB reduction but was removed due to time constraints. The relationship between the two is something that should be considered using the research in this paper.

GFDM is the focus waveform for 5G networks in this project as defined in the scope. There are still future developments across all areas of other competing waveforms. Other waveforms such as FBMC may also be considered as a candidate. By comparing exact

performance measurements such as OOB emissions across all future waveforms, a standard can be compared with to ensure 5G has the best interface suited.

7 Abbreviations

5G (Fifth Generation)

Additive White Gaussian Noise (AWGN)

Binary Phase-Shift Keying (BPSK)

Bit Error Rate (BER)

Carrier Frequency Offset (CFO)

Cyclic Prefix (CP)

Cyclic Suffix (CS)

Digital/Analogue (D/A)

Fast Fourier Transform (FFT)

FBMC (Filter Bank Multi-Carrier)

Finite Impulse Response (FIR)

Frequency Sampling Filter (FSF)

Generalised Frequency Division Multiplexing (GFDM)

Guard Symbol (GS)

Guard Symbol GFDM (GS-GFDM)

In Phase-Quadrature (I-Q)

Inter-Block-Interference (IBI)

Inter-Carrier Interference (ICI)

Inter-Symbol Interference (ISI)

Inverse Fast Fourier Transform (IFFT)

IoT (Internet of Things)

Matched Filter (MF)

Minimum Mean Square Error (MMSE)

Multi-Carrier Modulation (MCM)

NOW (Non-Orthogonal Waveforms)

Orthogonal Frequency Division Multiplexing (OFDM)

Out-of-band (OOB)

Power Spectral Density (PSD)

Quadrature Phase-Shift Keying (QPSK)

Raised Cosine (RC)

Root-Raised-Cosine (RRC)

Signal-to-Noise Ratio (SNR)

SMART (Specific, Measurable, Attainable, Relevant, Time Bound)

Successive Interference Cancellation (SIC)

Timing Offset (TO)

Windowed GFDM (W-GFDM)

Wireless Local Area Network (WLAN)

Wireless Regional Area Network (WRAN)

Zero-Forcing (ZF)

8 References

[1] I. Gaspar *et al.*, "GFDM - A Framework for Virtual PHY Services in 5G Networks," 2015.

[2] M. Van Eeckhaute, A. Bourdoux, P. De Doncker, and F. Horlin, "Performance of emerging multi-carrier waveforms for 5G asynchronous communications," *EURASIP Journal on Wireless Communications and Networking,* vol. 2017, no. 1, pp. 1-15, 2017.

[3] G. Wunder *et al.*, "5GNOW: non-orthogonal, asynchronous waveforms for future mobile applications," *Communications Magazine, IEEE,* vol. 52, no. 2, pp. 97-105, 2014.

[4] A. Mokdad, P. Azmi, and N. Mokari, "Radio resource allocation for heterogeneous traffic in GFDM-NOMA heterogeneous cellular networks," *IET Communications,* vol. 10, no. 12, pp. 1444-1455, 2016.

[5] T. Suzuki, T. Sato, and T. Yoshioka, "Analysis of the interference from GFDM to OFDM signals in same band," *IEICE Communications Express,* vol. 6, no. 3, pp. 126-131, 2017.

[6] S. Han, Y. Sung, and Y. H. Lee, "Filter Design for Generalized Frequency-Division Multiplexing," 2016.

[7] N. Michailow *et al.*, "Generalized Frequency Division Multiplexing for 5th Generation Cellular Networks," *Communications, IEEE Transactions on,* vol. 62, no. 9, pp. 3045-3061, 2014.

[8] T. T. Ha, *Theory and Design of Digital Communication Systems*. Cambridge University Press, 2010.

[9] A. B. Narasimhamurthy, *OFDM systems for wireless communications*. San Rafael, Calif.: San Rafael, Calif: Morgan & Claypool Publishers, 2010.

[10] D. Peng, X. Ma, X. Yao, and H. Zhang, "Receiver IQ mismatch estimation in PDM CO-OFDM system using training symbol," *Optical Fiber Technology,* vol. 36, pp. 417-421, 2017.

[11] A. Jayaprakash and G. Reddy, "Robust Blind Carrier Frequency Offset Estimation Algorithm for OFDM Systems," *An International Journal,* vol. 94, no. 3, pp. 777-791, 2017.

[12] B. Farhang-Boroujeny and H. Moradi, "Derivation of GFDM Based on OFDM Principles," *IEEE International Conference on Communications (ICC) 2015 London,* 2015.

[13] F. Hu, *Opportunities in 5G Networks: A Research and Development Perspective*. CRC Press, 2016.

[14] P. Bhalawi and A. Raghuwanshi, "Improved Computational Time for Circular / Linear Convolution using FFT by Matrix Multiplication," *International Journal of Engineering Innovations and Research,* vol. 6, no. 2, pp. 60-64, 2017.

[15] F. L. Luo and C. Zhang, *Signal Processing for 5G: Algorithms and Implementations*. Wiley, 2016.

[16] R. Gerzaguet *et al.*, "The 5G candidate waveform race: a comparison of complexity and performance," *EURASIP Journal on Wireless Communications and Networking,* vol. 2017, no. 1, pp. 1-14, 2017.

[17] I. Gaspar, M. Matthe, N. Michailow, L. Leonel Mendes, D. Zhang, and G. Fettweis, "Frequency-Shift Offset-QAM for GFDM," *Communications Letters, IEEE,* vol. 19, no. 8, pp. 1454-1457, 2015.

[18] P.-C. Chen, B. Su, and Y. Huang, "Matrix Characterization for GFDM: Low Complexity MMSE Receivers and Optimal Filters," *IEEE Transactions on Signal Processing,* vol. 65, no. 18, pp. 4940-4955, 2017.

[19] A. Sahin, I. Guvenc, and H. Arslan, "A Survey on Multicarrier Communications: Prototype Filters, Lattice Structures, and Implementation Aspects," *Communications Surveys & Tutorials, IEEE,* vol. 16, no. 3, pp. 1312-1338, 2014.

[20] X. Xiang-Gen, "A family of pulse-shaping filters with ISI-free matched and unmatched filter properties," *Communications, IEEE Transactions on,* vol. 45, no. 10, pp. 1157-1158, 1997.

[21] J. Wu, X. Ma, X. Qi, Z. Babar, and W. Zheng, "Influence of Pulse Shaping Filters on PAPR Performance of Underwater 5G Communication System Technique: GFDM," *Wireless Communications and Mobile Computing,* vol. 2017, 2017.

[22] E. C. Commitee, "ECC Recommendation - Unwanted Emissions," 2012.

[23] R. Datta, G. Fettweis, Z. Koll´r, and P. Horv´th, "FBMC and GFDM Interference Cancellation Schemes for Flexible Digital Radio PHY Design," *IEEE,* 2011.

[24] S. Antapurkar, A. Pandey, and K. Gupta, *GFDM performance in terms of BER, PAPR and OOB and comparison to OFDM system*. 2015.

[25] M. Matthé, N. Michailow, I. Gaspar, and G. Fettweis, "Influence of pulse shaping on bit error rate performance and out of band radiation of generalized frequency division multiplexing. Paper presented at the ICC'14 - workshop on 5G technologies (ICC'14 WS - 5G)," in *Influence of pulse shaping on bit error rate performance and out of band radiation of generalized frequency division multiplexing. Paper presented at the ICC'14 - workshop on 5G technologies (ICC'14 WS - 5G)*Australia: Sydney, 2014.

9 Appendix

Appendix.1 Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
1	2/8/17	Initial meeting		A. Vuk
2	9/8/17	Regular meeting		A. Vuk
3	16/8/17	"		A. Vuk
4	23/8/17	"		A. Vuk
5	30/8/17	"		A. Vuk
6	6/9/17	"		A. Vuk
7	13/9/17	"		A. Vuk
Semester Break	27/9/17	"		A. Vuk
8	4/10/17	"		A. Vuk
9	11/10/17	"		A. Vuk
10	18/10/17	"		A. Vuk
11	25/10/17	"		A. Vuk
12	1/11/17	"		A. Vuk

Appendix.2 OFDM Simulation [9]

```
clear all
clc

N = 16; % Number of subcarriers in each OFDM symbol
L = 3; % Channel order
CP_length = 4; % Cyclic prefix length
B = 10; % Number of OFDM symbols per transmitted frame
mc_N = 5000; % Number of iterations to achieve sufficient errors
SNR_db = 0:2:20; %SNR in dB
SNR = 10.^(SNR_db/10); % SNR values
Pe = zeros(size(SNR_db)); % Initializing the error vector
Total_length = (CP_length+N)*B; % Total length of each frame
am = [-1,1]; % For BPSK
M = 2; % For BPSK

for SNR_loop = 1:length(SNR_db)
    rho = SNR(SNR_loop);
    err = 0;

    for mc_loop = 1:mc_N
        dat_ind = ceil(M*rand(B,N));
        data = am(dat_ind);

        % Reshaping the data into a BxN matrix,...
        %...used later for error detection
        data_reshape = reshape(data, 1, B*N);

        tx_data = data;

        for b = 1:B
            % Taking the IFFT
            data_t(b,:) = ifft(tx_data(b,:));
        end

        % Adding Cyclic prefix
        data_cp = [data_t(:,end-CP_length+1:end), data_t];

        % Reshape the BxN matrix to obtain the frame (1xTotal_length)
        data_tx =reshape(data_cp.',1,Total_length);
        h = complex(randn(L+1,1), randn(L+1,1))*sqrt(0.5/(L+1));

        %Noise
noise=complex(randn(1,Total_length),randn(1,Total_length))*sqrt(0.5/N);

        % Received signal
        rec = sqrt(rho)*(filter(h,1,data_tx))+noise;

        % Reshape the recd signal into CP_length+N x B array
        rec_reshaped = (reshape(rec, CP_length+N, B)).';

        % Remove CP
        rec_sans_cp = rec_reshaped(:,CP_length+1:end);

        for bb = 1:B
            % Taking the FFT
            rec_f(bb,:) = fft(rec_sans_cp(bb,:));
        end
```

```
        % Calculating the equivalent channel on each subcarrier
        h_f = sqrt(rho)*fft(h,N);

        for b2 = 1:B

            % Extracting the OFDM symbol from the "rec_f" matrix
            rec_symbol = transpose(rec_f(b2,:));

            % Calc Euclidean dist assuming -1
            det1 = abs(rec_symbol+h_f).^2;

            % Calc Euclidean dist assuming +1
            det2 = abs(rec_symbol-h_f).^2;

            % Concatenate the two vectors
            det = [det1, det2];

            % Find symbol the recd signal is closest to
            [min_val, ind] = min(det, [], 2);


            % Generate the decoded symbols
            dec(b2,:) = 2*((ind-1)>0.5)-1;
        end

        % Reshape the decoded symbols to calc error
        dec_reshape = reshape(dec, 1, B*N);

        % Comparing dec_reshape against...
        %...data_reshape to calculate errors
        err = err + sum(dec_reshape~=data_reshape);
    end
    % Calculate the probability of error
    Pe(SNR_loop) = err/(mc_N*B*N);
end

% Semilog plot of Pe vs. SNR_db
semilogy(SNR_db,Pe)

gantt chart, WBS, attendance form
```

Appendix.3 GFDM Simulation


```
% This example shows how a random GFDM signal can be created. Additionally,
an OFDM signal with an equal length is created, and the Spectrum of both is
compared.

%% Create parameter sets for GFDM and OFDM
gfdm = get_defaultGFDM('TTI');
gfdm.K = 512;
gfdm.Kset = 100:200;  % Only allocate some subcarriers
gfdm.pulse = 'rc';
gfdm.a = 0.1;
gfdm.Mon = 14;

ofdm = gfdm;
ofdm.pulse = 'rc_td';  % use RC_TD with roll-off 0 to make a
ofdm.a = 0;            % rectangular filter
ofdm.M = 1;
ofdm.Mon = 1;
ofdm.L = ofdm.K; %match the sub symbol to 1 subcarrier for frequency domain
nB = 3; % Number of GFDM blocks to generate

%% Generate the signals % Currently only works without CP
assert(~isfield(gfdm, 'Ncp') || gfdm.Ncp == 0);

% Allocate enough space for the signals
blockLen = gfdm.M*gfdm.K;
sGFDM = zeros(nB * blockLen, 1);
sOFDM = zeros(size(sGFDM));

for b = 1:nB
    % Create GFDM signal by modulation of random data
    D = do_map(gfdm, do_qammodulate(get_random_symbols(gfdm), gfdm.mu));
    x = do_modulate(gfdm, D);
    sGFDM((b-1)*blockLen+(1:blockLen)) = x;

    % Create an OFDM signal
    for m = 1:gfdm.M
        D2    =    do_map(ofdm,    do_qammodulate(get_random_symbols(ofdm),
ofdm.mu));
        x2 = do_modulate(ofdm, D2);
        sOFDM((b-1)*blockLen+(m-1)*gfdm.K+(1:gfdm.K)) = x2;
    end
end

%% Plot the resulting PSD
f = linspace(-gfdm.K/2, gfdm.K/2, 2*length(sGFDM)+1); f = f(1:end-1)';
plot(f, mag2db(fftshift(abs(fft(sOFDM, 2*length(sOFDM)))))/2, 'b');
hold on;
plot(f, mag2db(fftshift(abs(fft(sGFDM, 2*length(sGFDM)))))/2, 'r');
hold off;
ylim([-40, 30]);
xlabel('f/F'); ylabel('PSD [dB]');
grid()
legend({'OFDM', 'GFDM'});
```

Appendix.4 GFDM Filters

```
% Copyright (c) 2014 TU Dresden
% All rights reserved.
% See accompanying license.txt for details.
%


function g = get_transmitter_pulse(p)
% GET_TRANSMITTER_PULSE - Return g00 for the given GFDM paramater set.
%
% See the code for available filter names

if ~isfield(p, 'cache.g')
    if strcmp(p.pulse, 'rc')
        g = rc(p.M, p.K, p.a);
    elseif strcmp(p.pulse, 'rrc')
        g = rrc(p.M, p.K, p.a);
    elseif strcmp(p.pulse, 'dirichlet')
        g = dirichlet(p.M, p.K);
    elseif strcmp(p.pulse, 'rect_fd')
        g = rect_fd(p.M, p.K);
    elseif strcmp(p.pulse, 'rect_td')
        g = rect_td(p.M, p.K);
    elseif ~isempty(regexp(p.pulse, '_[tf]d$'))
        g = rampbasedfir(p.M, p.K, p.a, p.pulse);
    else
        error('Unknown pulse shaping filter')
    end
else
    g = p.cache.g;
end


function g = rc(M, K, a)
% RC - Return Raised Cosine filter (time domain)
%
t = linspace(-M/2, M/2, M*K+1); t = t(1:end-1); t = t';

g = (sinc(t) .* cos(pi*a*t) ./ (1-4*a*a*t.*t));
g = fftshift(g);
g(K+1:K:end) = 0;

g = g / sqrt(sum(g.*g));

function g = rrc(M, K, a)
% RRC - Return Root Raised Cosine filter (time domain)
%

t = linspace(-M/2, M/2, M*K+1); t = t(1:end-1); t = t';

g = (sin(pi*t*(1-a))+4*a.*t.*cos(pi*t*(1+a))./(pi.*t.*(1-(4*a*t).^2));
g(find(t==0)) = 1-a+4*a/pi;
g(find(abs(t) == 1/(4*a))) = a/sqrt(2)*((1+2/pi)*sin(pi/(4*a))+(1-
2/pi)*cos(pi/(4*a)));

g = fftshift(g);
g = g / sqrt(sum(g.*g));
```

```
function g = dirichlet(M, K)
% DIRICHLET - Return Dirichlet Kernel

o = ones(1, M);
z = zeros(1, K*M-M);

G = [o z];
G = circshift(G, [0, -floor(M/2)]);

g = ifft(G);
g = g / sqrt(sum(abs(g).^2));
g = g';

function g = rect_fd(M, K) %OFDM
% rect_fd - Return filter that is a symmetric rect in the frequency domain.
assert(mod(M, 2) == 0);
G = zeros(M*K, 1);
G(1:M/2) = 1;
G(end-M/2+1:end) = 1;
g = ifft(G);
g = g / sqrt(sum(abs(g.*g)));

function g = rect_td(M, K) %OFDM
    g = zeros(M*K, 1);
    g(1:K) = 1;
    g = g / sqrt(sum(g.*g));
end
```

Appendix.5 Rectangular Pulse Energy

```
%RECTANGULAR PULSE
Fs=100; %change sampling frequency
t0=1; %change end length
t = 0:1/Fs:t0;
X=@(t) 2*rectangularPulse(0,t0,t); %Change amplitude
L = length(t);
Lsig = length(X);
figure
plot(t,X(t))
grid on
title('Sine Wave in Time Domain')
xlabel('Time (t)')
ylabel('X(t)')

%ENERGY TIME DOMAIN
Energytime=(abs(integral(X,-Inf,Inf))^2) %parsevals equation

%FFT
Y = fft(X(t));
P2 = abs(Y/Lsig);
P1 = P2(1:floor(Lsig/2+1));
P1(2:end-1) = 2*P1(2:end-1);
%doing FFT again to allow for plot
n = 2^nextpow2(L);
Y2 = fft(X(t),n);
f = Fs*(0:(n/2))/n;
P = abs(Y2/n);
Pplot=P(1:n/2+1);
figure
plot(f,Pplot)
grid on
title('Sine Wave in Frequency Domain')
xlabel('Frequency (f)')
ylabel('|P(f)|')

%%PSD
psdx = (1/(Lsig*Fs)) * abs(P1).^2; %parsevals equation
%use from plot fft
PdBW = 10*log10(Pplot);
figure
plot(f,PdBW)
grid on
title('Power Spectral Density Using FFT')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')

%%OR PERIODOGRAM
figure
periodogram(X(t),[],f,Fs)

%%ENERGY FREQUENCY DOMAIN
Energyfreq = sum(psdx)/Fs % in watts
```

Appendix.6 Sine Pulse Energy

```
%TIME DOMAIN SINGNAL
Fs=16; % sampling frequency
t0=1; % end length
t = 0:1/Fs:t0-1/Fs;
X=@(t) sin(2*pi*t); %signal
absX2=@(t) abs(X(t)).^2;
Ltim = length(t); % time length
Lsig = length(X(t)); % signal length

figure(1)
plot(t,X(t))
grid on
title('Rectangular Pulse in Time Domain')
xlabel('Time (t)')
ylabel('X(t)')

%ENERGY TIME DOMAIN
Energytime=integral(absX2,0,t0) % time domain parsevals

%PLOT FFT
n = 2^nextpow2(Ltim); %zero padding
Y = fft(X(t),n); % FFT
f = Fs*(0:n-1)/n; % freq axis
P = abs(Y/n); % P for parsevals eqn

figure(2)
plot(f,P)
grid on
title('Rectangular Pulse in Frequency Domain')
xlabel('Frequency (f)')
ylabel('|P(f)|')

%%PLOT PSD
PdBW = 10*log10(P);
figure(3)
plot(f,PdBW)
grid on
title('Power Spectral Density Using FFT')
xlabel('Frequency (Hz)')
ylabel('Power/Frequency (dB/Hz)')

%%OR PERIODOGRAM
figure(4)
periodogram(X(t),[],f,Fs)

%%ENERGY FREQ DOMAIN
psdf = P.^2*(n/Fs);% n/Fs to allow for non n^2 sampling freq
EnergyFreq = sum(psdf) % freq domain parsevals
```

Appendix.7 GFDM Energy and OOB

```
%for j=1:100 %enable to create 100 samples for avg calculated at end

%% Create parameter sets for GFDM1 RC
gfdm1 = get_defaultGFDM('TTI');
gfdm1.K = 512;
gfdm1.Kset = 100:120;  % Only allocate some subcarriers
gfdm1.pulse = 'rc';
gfdm1.a = 0.9;
gfdm1.Mon = 14;
nB = 3; % Number of GFDM blocks to generate

%% Create parameter sets for GFDM2 RRC
gfdm2 = get_defaultGFDM('TTI');
gfdm2.K = 512;
gfdm2.Kset = 100:120;  % Only allocate some subcarriers
gfdm2.pulse = 'rrc';
gfdm2.a = 0.9;
gfdm2.Mon = 14;

%% Create parameter sets for GFDM3 Dirichlet
gfdm3 = get_defaultGFDM('TTI');
gfdm3.K = 512;
gfdm3.Kset = 100:120;  % Only allocate some subcarriers
gfdm3.pulse = 'dirichlet';
%no need for roll-off factor as Dirichlet contains rectangular pulse
similarity
gfdm3.Mon = 14;

%% Create parameter sets for GFDM4 Xia 1st
gfdm4 = get_defaultGFDM('TTI');
gfdm4.K = 512;
gfdm4.Kset = 100:120;  % Only allocate some subcarriers
gfdm4.pulse = 'xia1st_td';
gfdm4.a = 0.9;
gfdm4.Mon = 14;

%% Create parameter sets for GFDM5 Xia 4th
gfdm5 = get_defaultGFDM('TTI');
gfdm5.K = 512;
gfdm5.Kset = 100:120;  % Only allocate some subcarriers
gfdm5.pulse = 'xia4th_td';
gfdm5.a = 0.9;
gfdm5.Mon = 14;

%% Generate the GFDM signals
%all without CP
assert(~isfield(gfdm1, 'Ncp') || gfdm1.Ncp == 0);
assert(~isfield(gfdm2, 'Ncp') || gfdm2.Ncp == 0);
assert(~isfield(gfdm3, 'Ncp') || gfdm3.Ncp == 0);
assert(~isfield(gfdm4, 'Ncp') || gfdm4.Ncp == 0);
assert(~isfield(gfdm5, 'Ncp') || gfdm5.Ncp == 0);

% Allocate enough space for each of the signals
blockLen = gfdm1.M*gfdm1.K;
sGFDM1 = zeros(nB * blockLen, 1);

blockLen2 = gfdm2.M*gfdm2.K;
```

```
sGFDM2 = zeros(nB * blockLen2, 1);

blockLen3 = gfdm3.M*gfdm3.K;
sGFDM3 = zeros(nB * blockLen3, 1);

blockLen4 = gfdm4.M*gfdm4.K;
sGFDM4 = zeros(nB * blockLen4, 1);

blockLen5 = gfdm5.M*gfdm5.K;
sGFDM5 = zeros(nB * blockLen5, 1);

for b = 1:nB
    % Create RC GFDM signal by modulation of random data
    D = do_map(gfdm1, do_qammodulate(get_random_symbols(gfdm1), gfdm1.mu));
    x = do_modulate(gfdm1, D);
    sGFDM1((b-1)*blockLen+(1:blockLen)) = x;

    % Create RRC GFDM signal by modulation of random data
    D2 = do_map(gfdm2, do_qammodulate(get_random_symbols(gfdm2),
gfdm2.mu));
    x2 = do_modulate(gfdm2, D2);
    sGFDM2((b-1)*blockLen2+(1:blockLen2)) = x2;

    % Create Dirichlet GFDM signal by modulation of random data
    D3 = do_map(gfdm3, do_qammodulate(get_random_symbols(gfdm3),
gfdm3.mu));
    x3 = do_modulate(gfdm3, D3);
    sGFDM3((b-1)*blockLen3+(1:blockLen3)) = x3;

    % Create Xia 1st GFDM signal by modulation of random data
    D4 = do_map(gfdm4, do_qammodulate(get_random_symbols(gfdm4),
gfdm4.mu));
    x4 = do_modulate(gfdm4, D4);
    sGFDM4((b-1)*blockLen4+(1:blockLen4)) = x4;

    % Create Xia 4th GFDM signal by modulation of random data
    D5 = do_map(gfdm5, do_qammodulate(get_random_symbols(gfdm5),
gfdm5.mu));
    x5 = do_modulate(gfdm5, D5);
    sGFDM5((b-1)*blockLen5+(1:blockLen5)) = x5;
end

%% Plot the resulting P(f)
f = linspace(-gfdm1.K/2, gfdm1.K/2, 2*length(sGFDM1)+1); f = f(1:end-1)';
%frequency axis in frequency domain

RC  = fftshift(abs(fft(sGFDM1, 2*length(sGFDM1))))/2; %RC signal to be
plotted
RRC = fftshift(abs(fft(sGFDM2, 2*length(sGFDM2))))/2; %RRC
DIR = fftshift(abs(fft(sGFDM3, 2*length(sGFDM3))))/2; %Dirichlet
XIA1 = fftshift(abs(fft(sGFDM4, 2*length(sGFDM4))))/2; %Xia 1st
XIA4 = fftshift(abs(fft(sGFDM5, 2*length(sGFDM5))))/2; %Xia 4th

%Plot P(f)
figure(1)
hold on
plot(f,RC, 'r'); %RC red
plot(f,RRC, 'b'); %RRC blue
plot(f,DIR, 'y'); %Dirichlet yellow
```

```
plot(f,XIA1, 'g'); %Xia 1st green
plot(f,XIA4, 'c'); %Xia 4th cyan
hold off
ylim([-30 300])
xlabel('Frequency(f)'); ylabel('|P(f)|');
grid()
title('Frequency Domain');
legend({'GFDM-RC','GFDM-RRC','Dirichlet','Xia 1st','Xia 4th'});

% Turn into upper envelope for easier comparaison
[yupper1,ylower1]=envelope(RC,1,'peak'); %take envelope every 1 sample
[yupper2,ylower2]=envelope(RRC,1,'peak');
[yupper3,ylower3]=envelope(DIR,1,'peak');
[yupper4,ylower4]=envelope(XIA1,1,'peak');
[yupper5,ylower5]=envelope(XIA4,1,'peak');

figure(2)
hold on
plot(f,yupper1, 'r'); %RC red
plot(f,yupper2, 'b'); %RRC blue
plot(f,yupper3, 'y'); %Dirichlet yellow
plot(f,yupper4, 'g'); %Xia 1st green
plot(f,yupper5, 'c'); %Xia 4th cyan
hold off
ylim([-30 300])
xlabel('Frequency(f)'); ylabel('|P(f)|');
grid()
title('Frequency Domain - Envelopes');
legend({'GFDM-RC Envelope','GFDM-RRC Envelope','Dirichlet Envelope','Xia
1st Envelope','Xia 4th Envelope'});

%Plot PSD
figure(3)
hold on
plot(f, mag2db(RC), 'r'); %take magnitude and in dB scale
plot(f, mag2db(RRC), 'b');
plot(f, mag2db(DIR), 'y');
plot(f, mag2db(XIA1), 'g');
plot(f, mag2db(XIA4), 'c');
hold off
xlabel('f(Hz)'); ylabel('Power/Frequencey(dB/Hz)');
grid()
title('Power Spectral Density');
legend({'GFDM-RC','GFDM-RRC','Dirichlet','Xia 1st','Xia 4th'});

%% Energy in frequency domain
psdfRC = RC.^2; %end of parsevals equation
psdfRRC = RRC.^2;
psdfDIR = DIR.^2;
psdfXIA1 = XIA1.^2;
psdfXIA4 = XIA4.^2;

%total energy
%EnergyFreqRC = sum(psdfRC)
%EnergyFreqRRC = sum(psdfRRC)
%EnergyFreqDIR = sum(psdfDIR)
%EnergyFreqXIA1 = sum(psdfXIA1)
%EnergyFreqXIA4 = sum(psdfXIA4)

%oob domain
```

```
OOBRC = trapz(f>130&f<170,psdfRC)
OOBRRC = trapz(f>130&f<170,psdfRRC)
OOBDIR = trapz(f>130&f<170,psdfDIR)
OOBXIA1 = trapz(f>130&f<170,psdfXIA1)
OOBXIA4 = trapz(f>130&f<170,psdfXIA4)

%EnergyFreqRC(j) = sum(psdfRC);
%EnergyFreqRRC(j) = sum(psdfRRC);
%EnergyFreqDIR(j) = sum(psdfDIR);
%EnergyFreqXIA1(j) = sum(psdfXIA1);
%EnergyFreqXIA4(j) = sum(psdfXIA4);

%OOBRC(j) = trapz(f>130&f<170,psdfRC); %use trapezoidal rule as approx
integration between OOB domain
%OOBRRC(j) = trapz(f>130&f<170,psdfRRC);
%OOBDIR(j) = trapz(f>130&f<170,psdfDIR);
%OOBXIA1(j) = trapz(f>130&f<170,psdfXIA1);
%OOBXIA4(j) = trapz(f>130&f<170,psdfXIA4);
%end

%mean(EnergyFreqRC)
%mean(EnergyFreqRRC)
%mean(EnergyFreqDIR)
%mean(EnergyFreqXIA1)
%mean(EnergyFreqXIA4)

%mean(OOBRC)
%mean(OOBRRC)
%mean(OOBDIR)
%mean(OOBXIA1)
%mean(OOBXIA4)
```

Appendix.8 GFDM OOB Filters and OFDM

```
%for j=1:100 %enable to create 100 samples for avg calculated at end

%% Create parameter sets for GFDM1 RC
gfdm1 = get_defaultGFDM('TTI');
gfdm1.K = 512;
gfdm1.Kset = 100:120;  % Only allocate some subcarriers
gfdm1.pulse = 'rc';
gfdm1.a = 0.9;
gfdm1.Mon = 14;
nB = 3; % Number of GFDM blocks to generate

%% Create parameter sets for GFDM2 RRC
gfdm2 = get_defaultGFDM('TTI');
gfdm2.K = 512;
gfdm2.Kset = 100:120;  % Only allocate some subcarriers
gfdm2.pulse = 'rrc';
gfdm2.a = 0.9;
gfdm2.Mon = 14;

%% Create parameter sets for GFDM3 Dirichlet
gfdm3 = get_defaultGFDM('TTI');
gfdm3.K = 512;
gfdm3.Kset = 100:120;  % Only allocate some subcarriers
gfdm3.pulse = 'dirichlet';
%no need for roll-off factor dirichlet contains rectangular pulse
similarity
gfdm3.Mon = 14;

%% Create parameter sets for GFDM4 Xia 1st
gfdm4 = get_defaultGFDM('TTI');
gfdm4.K = 512;
gfdm4.Kset = 100:120;  % Only allocate some subcarriers
gfdm4.pulse = 'xia1st_td';
gfdm4.a = 0.9;
gfdm4.Mon = 14;

%% Create parameter sets for GFDM5 Xia 4th
gfdm5 = get_defaultGFDM('TTI');
gfdm5.K = 512;
gfdm5.Kset = 100:120;  % Only allocate some subcarriers
gfdm5.pulse = 'xia4th_td';
gfdm5.a = 0.9;
gfdm5.Mon = 14;

%% Create parameter sets for OFDM
ofdm = gfdm1;
ofdm.pulse = 'rc_td';  % use RC_TD with rolloff 0 to make a
ofdm.a = 0;            % rectangular filter
ofdm.M = 1;
ofdm.Mon = 1;
ofdm.L = ofdm.K;

%% Generate the GFDM signals
%all without CP
assert(~isfield(gfdm1, 'Ncp') || gfdm1.Ncp == 0);
assert(~isfield(gfdm2, 'Ncp') || gfdm2.Ncp == 0);
assert(~isfield(gfdm3, 'Ncp') || gfdm3.Ncp == 0);
```

```
assert(~isfield(gfdm4, 'Ncp') || gfdm4.Ncp == 0);
assert(~isfield(gfdm5, 'Ncp') || gfdm5.Ncp == 0);

% Allocate enough space for each of the signals
blockLen = gfdm1.M*gfdm1.K;
sGFDM1 = zeros(nB * blockLen, 1);

blockLen2 = gfdm2.M*gfdm2.K;
sGFDM2 = zeros(nB * blockLen2, 1);

blockLen3 = gfdm3.M*gfdm3.K;
sGFDM3 = zeros(nB * blockLen3, 1);

blockLen4 = gfdm4.M*gfdm4.K;
sGFDM4 = zeros(nB * blockLen4, 1);

blockLen5 = gfdm5.M*gfdm5.K;
sGFDM5 = zeros(nB * blockLen5, 1);

sOFDM = zeros(size(sGFDM1));


for b = 1:nB
    % Create RC GFDM signal by modulation of random data
    D = do_map(gfdm1, do_qammodulate(get_random_symbols(gfdm1), gfdm1.mu));
    x = do_modulate(gfdm1, D);
    sGFDM1((b-1)*blockLen+(1:blockLen)) = x;

    % Create RRC GFDM signal by modulation of random data
    D2 = do_map(gfdm2, do_qammodulate(get_random_symbols(gfdm2),
gfdm2.mu));
    x2 = do_modulate(gfdm2, D2);
    sGFDM2((b-1)*blockLen2+(1:blockLen2)) = x2;

    % Create Dirichlet GFDM signal by modulation of random data
    D3 = do_map(gfdm3, do_qammodulate(get_random_symbols(gfdm3),
gfdm3.mu));
    x3 = do_modulate(gfdm3, D3);
    sGFDM3((b-1)*blockLen3+(1:blockLen3)) = x3;

    % Create Xia 1st GFDM signal by modulation of random data
    D4 = do_map(gfdm4, do_qammodulate(get_random_symbols(gfdm4),
gfdm4.mu));
    x4 = do_modulate(gfdm4, D4);
    sGFDM4((b-1)*blockLen4+(1:blockLen4)) = x4;

    % Create Xia 4th GFDM signal by modulation of random data
    D5 = do_map(gfdm5, do_qammodulate(get_random_symbols(gfdm5),
gfdm5.mu));
    x5 = do_modulate(gfdm5, D5);
    sGFDM5((b-1)*blockLen5+(1:blockLen5)) = x5;

    % Create an OFDM signal
    for m = 1:gfdm1.M
        D6 = do_map(ofdm, do_qammodulate(get_random_symbols(ofdm),
ofdm.mu));
        x6 = do_modulate(ofdm, D6);
        sOFDM((b-1)*blockLen+(m-1)*gfdm1.K+(1:gfdm1.K)) = x6;
    end
```

```
end

%% Plot the resulting P(f)
f = linspace(-gfdm1.K/2, gfdm1.K/2, 2*length(sGFDM1)+1); f = f(1:end-1)';
%frequency axis in frequency domain

RC  = fftshift(abs(fft(sGFDM1, 2*length(sGFDM1))))/2; %RC signal to be
plotted
RRC = fftshift(abs(fft(sGFDM2, 2*length(sGFDM2))))/2; %RRC
DIR = fftshift(abs(fft(sGFDM3, 2*length(sGFDM3))))/2; %Dirichlet
XIA1 = fftshift(abs(fft(sGFDM4, 2*length(sGFDM4))))/2; %Xia 1st
XIA4 = fftshift(abs(fft(sGFDM5, 2*length(sGFDM5))))/2; %Xia 4th

OFDM = fftshift(abs(fft(sOFDM, 2*length(sOFDM))))/2;


%Plot P(f)
figure(1)
hold on
plot(f,RC, 'r'); %RC red
plot(f,RRC, 'b'); %RRC blue
plot(f,DIR, 'y'); %Dirichlet yellow
plot(f,XIA1, 'g'); %Xia 1st green
plot(f,XIA4, 'c'); %Xia 4th cyan
plot(f,OFDM, 'm'); %OFDM black
hold off
ylim([-30 300])
xlabel('Frequency(f)'); ylabel('|P(f)|');
grid()
title('Frequency Domain');
legend({'GFDM-RC','GFDM-RRC','Dirichlet','Xia 1st','Xia 4th','OFDM'});

% Turn into upper envelope for easier comparaison
[yupper1,ylower1]=envelope(RC,1,'peak'); %take envelope every 1 sample
[yupper2,ylower2]=envelope(RRC,1,'peak');
[yupper3,ylower3]=envelope(DIR,1,'peak');
[yupper4,ylower4]=envelope(XIA1,1,'peak');
[yupper5,ylower5]=envelope(XIA4,1,'peak');
[yupper6,ylower6]=envelope(OFDM,1,'peak');

figure(2)
hold on
plot(f,yupper1, 'r'); %RC red
plot(f,yupper2, 'b'); %RRC blue
plot(f,yupper3, 'y'); %Dirichlet yellow
plot(f,yupper4, 'g'); %Xia 1st green
plot(f,yupper5, 'c'); %Xia 4th cyan
plot(f,yupper6, 'm'); %OFDM black
hold off
ylim([-30 300])
xlabel('Frequency(f)'); ylabel('|P(f)|');
grid()
title('Frequency Domain - Envelopes');
legend({'GFDM-RC Envelope','GFDM-RRC Envelope','Dirichlet Envelope','Xia
1st Envelope','Xia 4th Envelope','OFDM Envelope'});

%Plot PSD
figure(3)
```

```
hold on
plot(f, mag2db(RC), 'r'); %take magnitude and in dB scale
plot(f, mag2db(RRC), 'b');
plot(f, mag2db(DIR), 'y');
plot(f, mag2db(XIA1), 'g');
plot(f, mag2db(XIA4), 'c');
plot(f, mag2db(OFDM), 'm');
hold off
xlabel('f(Hz)'); ylabel('Power/Frequencey(dB/Hz)');
grid()
title('Power Spectral Density');
legend({'GFDM-RC','GFDM-RRC','Dirichlet','Xia 1st','Xia 4th','OFDM'});

%% Energy in frequency domain
psdfRC = RC.^2; %end of parsevals equation
psdfRRC = RRC.^2;
psdfDIR = DIR.^2;
psdfXIA1 = XIA1.^2;
psdfXIA4 = XIA4.^2;
psdfOFDM = OFDM.^2;


%oob domain
OOBRC = trapz(f>130&f<170,psdfRC) %use trapezoidal rule as approx
integration between OOB domain
OOBRRC = trapz(f>130&f<170,psdfRRC)
OOBDIR = trapz(f>130&f<170,psdfDIR)
OOBXIA1 = trapz(f>130&f<170,psdfXIA1)
OOBXIA4 = trapz(f>130&f<170,psdfXIA4)

OOBOFDM = trapz(f>130&f<170,psdfOFDM)

%OOBRC(j) = trapz(f>130&f<170,psdfRC);
%OOBRRC(j) = trapz(f>130&f<170,psdfRRC);
%OOBDIR(j) = trapz(f>130&f<170,psdfDIR);
%OOBXIA1(j) = trapz(f>130&f<170,psdfXIA1);
%OOBXIA4(j) = trapz(f>130&f<170,psdfXIA4);
%OOBOFDM(j) = trapz(f>130&f<170,psdfOFDM);
%end

%mean(OOBRC)
%mean(OOBRRC)
%mean(OOBDIR)
%mean(OOBXIA1)
%mean(OOBXIA4)
%mean(OFDM)
```