Thesis Report

Blind OFDM Signal Parameter Estimation

Author: Kevin van der Mark, 4154509
Supervisors: prof. dr. E.W. McCune (TU Delft)
dr. ir. G.M.J. Janssen (TU Delft)
ir. Vincent Voogt (TNO)

Delft University of Technology
Delft, The Netherlands

TNO
The Hague, The Netherlands

A Thesis report submitted in fulfilment of the requirements for EE4300, Master Thesis, as part of the Master of Science EE program

August 2019

Preface

This thesis project has been completed under course code EE4300 as part of the Electrical Engineering master program at the faculty of EEMCS, Delft University of Technology.

I would like to thank TNO, and especially my supervisor Vincent Voogt, for giving me this opportunity. After working on so many projects at the university, it is very inspirational and exciting to put my theoretical knowledge into practice, knowing I made a contribution to a real world project.

I would also like to thank my university supervisors, prof. dr. Earl McCune and dr. ir. Gerard Janssen, for the support and occasional push in the right direction. It has been a very interesting time together and I learned a lot, thank you both for your valuable time.

K.P. van der Mark
The Hague, August 2019

Contents

Abstract

OFDM is a common modulation technique applied in the video downlink of drones. In order to eavesdrop on this link, the OFDM signal needs to be demodulated. OFDM demodulators separate orthogonal subcarriers to obtain single carrier signals. This process requires a lot of prior information on the signal parameters.

This thesis presents two techniques in obtaining these parameters blindly. The first approach is based on non-filtered channels, exploiting inherent OFDM signal properties. Although not realistic for practical applications it gives insight in the OFDM techniques. The second approach is based on the autocorrelation function, exploiting the correlation between the Cyclic Prefix and the end of an OFDM symbol. This techniques proves more robust to filtering and noise effects, but does introduce a noticeable error margin even present in the low noise simulations.

List of Figures

Abbreviations

ADSL	Asymmetric Digital Subscriber Line
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
CP	Cyclic Prefix
DAB	Digital Audio Broadcasting
DFT	Discrete Fourier Transform
DMB	Digital Terrestrial Multimedia Broadcast
DMT	Discrete Multi Tone
DSSS	Direct Sequence Spread Spectrum
DVB	Digital Video Broadcasting
ED	Electronic Defence
EM	electromagnetic
EW	Electronic Warfare
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
IQ	In-phase and Quadrature
ISI	Inter Symbol Interference
ISM	Industrial, Scientific and Medical
LTE	Long-Term Evolution
LTF	Long Training Field
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On-Off Keying
OSI	Open Systems Interconnection
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
STF	Short Training Field
TDM	Time Division Multiplexing
UAV	Unmanned Aerial Vehicle
WiMAX	Worldwide Interoperability for Microwave Access
ZP	Zero Padding

Symbols

A	Amplitude
ω	Angular Frequency
t	Time
ϕ	Phase
$s(t)$	Time Signal
$s_n[p]$	Subcarrier symbol
$s_i[p]$	OFDM symbol
$s_s[p]$	Transmitted OFDM symbol (including CP)
$r[p]$	Transmitted Signal
p	Sample Index
T_{SS}	Symbol Time
T_{cp}	Cyclic Prefix Time
T_S	Transmitted Symbol Time
T_{max}	Maximum Delay Spread
B_T	Transmission Bandwidth
M	Modulation level
R_S	OFDM Symbol Rate
R_{SS}	Subcarrier Symbol Rate
R_b	Transmitted OFDM Symbol Rate
$m(t)$	Unipolar Baseband Signal
N	Number of Subcarriers
n	Subcarrier Index
f	Frequency
L	Number of OFDM Symbols
i	OFDM Symbol Index
O	Modulation Order
$q[k]$	Discontinuity Index Vector
k	Discontinuity Index
j	Imaginary notation
δ	Delta Dirac
s_p	Number of Pilot Tones
n_p	Subcarrier index with Pilot Tone

1

Introduction

The connected world is all around us, and is expected to increase towards 100 billion network-enabled devices by 2030 [1]. The expectation is that the majority of these devices will be wireless such as IoT. Accommodating all these wireless transmissions requires smart engineering, resulting in a number of different techniques to e.g. reduce energy consumption and increase spectral efficiency [2]. One technique used to increase spectral efficiency is OFDM [3]. Currently, OFDM is a popular technique used in multiple wired and wireless standards and products such as ADSL, LTE, WiFi, DAB and DVB.

OFDM is also a popular technique in the video downlink of drones. Drones are increasing in popularity in regular use, but also in illegal activities such as snuggling contraband in to prisons. TNO is interested in eaves dropping this OFDM based video downlink to provide intelligence for law enforcement agencies. To enable eavesdropping, the OFDM signal parameters have to be obtained. Standardised implementations have a document set of possible parameters, but proprietary links such as the ones used in drones are not well documented. Therefore, a blind approach is needed to obtain the OFDM signal parameters. This chapter will give a short background followed by the motivation behind this research goal.

1.1 Background

Warfare used to be focused on three main fronts: ground, sea and air. Over the years, a fourth front has been added: space, followed by a fifth front: the EM spectrum. EW is the research area aimed at tackling the challenges in this fifth front, focusing on topics like, but not limited to, radar systems, radar countermeasures, covert communications and communication countermeasures [4, 5].

EW and Electronic Defence are topics gaining more and more interest world wide [6]. At TNO, this is reflected by the fast growing department of Electronic Defence. TNO is a Dutch research institute which has strong relations with the ministry of defence. The department of Electronic Defence works on technologies aimed at achieving tactical and strategic advantages in the electromagnetic spectrum.

From an Electronic Defence point of view, knowing who is transmitting and what is being transmit-

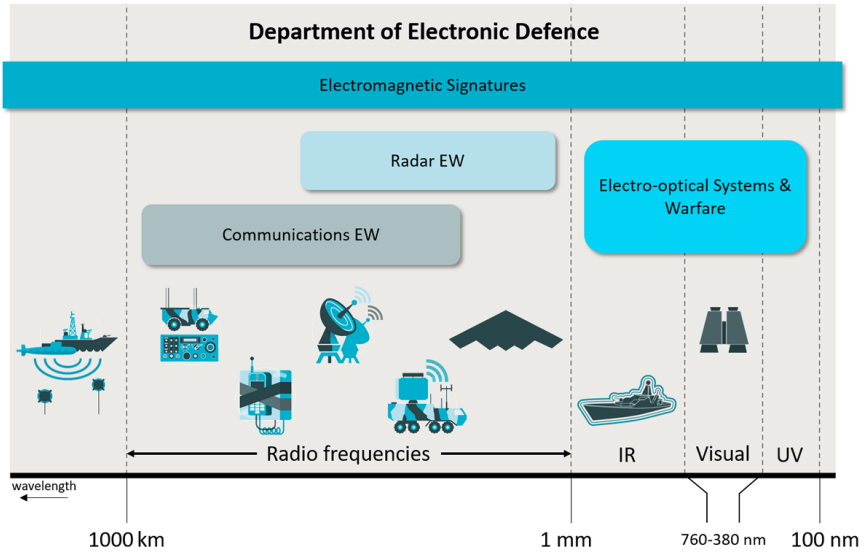


Figure 1.1: Fields of Expertise within the TNO department of Electronic Defence [9]

ted within the EM spectrum is of vital importance in achieving a tactical advantage and NATO wide investments are needed [7]. Collecting information on a signal to either decode, falsify or jamming the link gives the user a tactical advantage. Being able to decode enemy communications real-time may provide knowledge on their movement and tactics to be able to apply adequate counter actions. Injecting false information or the jamming of communications can disrupt internal communications and isolate targets, thus denying the enemy effective use of the EM spectrum [8].

1.2 Motivation

TNO is an independent research institute, focusing on innovation and knowledge acquisition. The department of ED works mainly for the Dutch Ministry of Defence, the Dutch intelligence agencies and the Ministry of Justice and Security [9]. The main fields of research of the ED department are depicted in figure 1.1.

1.2.1 TNO

Within the scope of Communications EW, the use and misuse of commercially available mini-UAV, also called drones, receives interest. The goal is to provide intelligence on drones being used in the field for the police or the military. Recent news articles show that drones can be dangerous and disruptive as, for example, happened in the UK in 2018. Airports in the UK alone experienced "at least 117 close calls between drones and planes" [10]. The second largest airport in the UK has been shut down for 33 hours in 2018, disrupting around 140000 travellers. In 2016, Dubai

Figure 1.2: The DJI Mavic 2 drone [13]

International Airport was closed for almost 2 hours over three instances due to illegal drone flights, accumulating to a total estimated financial loss of 11 million US dollars [11]. Potential collisions between drones and airplanes are the reason airport officials often preemptively close the airspace. This is a costly endeavour, but due to the safety risks one that is needed none the less. To aid the police in decreasing the response time and take down drones faster, more specialised tools are required.

Another example is drones being used to smuggle contraband into prisons [12]. Drugs, cell phones, weapons and other forbidden items are being smuggled into prisons all over the world. Companies are specialising in countering these drones, usually by jamming their control signals. In urban environments, however, jamming wide frequency bands is not always possible.

TNO is one of the parties involved in looking for solutions to these challenges. By researching different aspects related to drones, TNO is acquiring a strong knowledge position. With this knowledge, TNO is hoping to tackle the challenges of the future faster and more efficiently. The aspect of the counter drone program being addressed in this thesis is eavesdropping. Being able to receive information of an adversary's drone brings a huge tactical advantage in both military and civilian use cases. Based on the video link, information on the location and the purpose of the drone might be obtained.

1.2.2 Counter Drone

One popular and advanced drone is selected as a use case, the DJI Mavic 2 as seen in figure 1.2. This drone is modern, has a powerful camera and is mainly used for areal photography and filming, but is also capable of carrying a payload up tp 1 kg [13]. It uses the proprietary Ocusync 2.0 for its video transmission system. Not much information on the actual signal is provided, only that Ocusync 2.0 [14]

- Uses OFDM,
- Transmits around the 2.4 or 5 GHz,
- Has a SDR,
- Is based on off the shelf hardware.

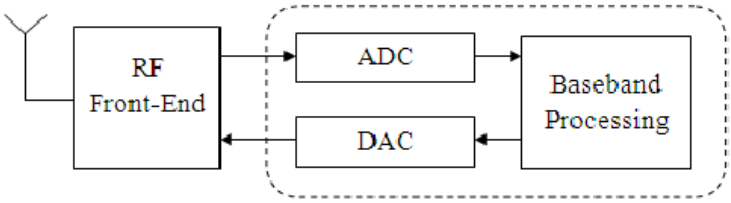


Figure 1.3: Basic SDR layout [15]

The advantages of using OFDM are its high data rate performance in urban environments. The fact that the Mavic 2 uses a SDR is very new and brings new challenges. Where previous generations used expensive dedicated hardware and FPGAs, the use of a SDR enables the manufacturer to use off the shelf hardware to generate arbitrary signals, only limited by the RF front end.

Each transceiver used to be composed of dedicated hardware, where the signal parameters were dependent on the hardware chosen. In these traditional designs, altering the centre frequency or other signal parameters required a hardware overhaul, which is relatively expensive and time consuming and often only applied in newer versions. An SDR on the other hand combines an RF-to-baseband transceiver and digital processor into one system, as illustrated in figure 1.3. This enables the user to generate arbitrary waveforms, limited only by the capabilities of the RF front end.

One more feature of the Ocusync 2.0 is that the drone and controller choose an arbitrary frequency band, within the ISM bands, for the downlink once it starts up. The uplink uses FHSS and hops throughout the selected band. During flight, if the downlink experiences interference or distortion, the drone automatically switches to another frequency band with less disturbances. This hampers the effectivity of jamming since not only the operational channel needs to be jammed but the entire frequency band.

The use of a SDR greatly increases the challenges presented. Dedicated hardware means the OFDM parameters hardly change over time, but a SDR can change its parameters even during flight. The limitations on the signal parameters are related since they are only limited by the SDR front end hardware.

1.2.3 OFDM

OFDM is currently found to be (one of) the most wide spread techniques used in high throughput data links. The difficulty in eavesdropping on these types of signals is that it requires a lot of prior knowledge on the signal parameters to demodulate the signal [16]. Figure 1.4 shows a full OFDM transmission system with bits being inserted by the sender and retrieved by the recipient. For the receiver to determine $\widehat{Data}$ from the input signal, multiple modulation parameters have to

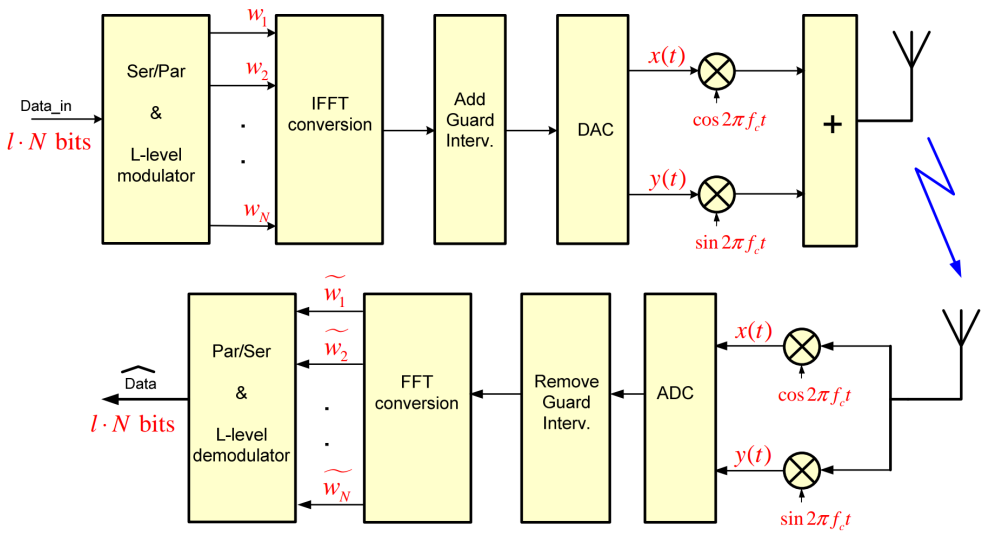


Figure 1.4: OFDM transmission system [17]

be known.

The problem is that, if these parameters are unknown, because the preamble was not received or the structure is not known, the data can not be retrieved. Especially when working with proprietary signal formats like the drone link described before, the structure of the physical layer frame is often unknown. A new blind approach needs to be derived to obtain the signal parameters needed to demodulate an OFDM signal.

At this stage, the goal is to analyse a recorded signal. There are no limitations regarding hardware such as processing power or memory usage. Once a technique has been derived, further research shall be done on improving the technique and at this stage, timing constraints shall be added. This is outside of the scope of this thesis and shall therefore not be further elaborated on.

1.3 Research Objective

Research on eavesdropping, or popularly known as "hacking", wireless signals is a hot topic. However, the steps needed to be taken to obtain data from an unknown wireless signal are bountiful. A detailed overview of the different challenges, such as blind demodulation, channel encoding estimation, decryption and frame reconstruction is given in the preceding internship report [18].

The first step in hacking a wireless signal is finding it. In some use cases, it is known if someone is communicating but not where in the frequency domain. Finding the desired signal can be done by employing multiple spectrum scanning techniques. Another use case is that a signal is found in the frequency spectrum, but its origin and purpose are unknown. In either case, once the signal has been obtained, converting it to data requires several more steps. First, the physical layer modulation

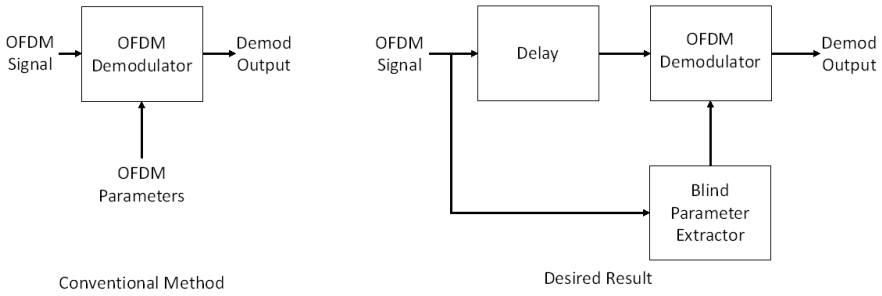


Figure 1.5: Research objective illustrated

technique applied by the user needs to be known. Theoretically, this could be anything from DSSS, FHSS or OFDM to 64-QAM. Besides signal modulation, scramblers or interleavers might be used, further increasing the complexity. Once the modulation type and use of scrambling or interleaving is known the signal can be demodulated and signal states in the IQ plane can be obtained[1]. What these signal states represent depends on the (unknown) data mapping that is being used. For documented links like WiFi, this data mapping is known, but proprietary links can use any data mapping its designers chose. This increases the complexity drastically since the possible mappings can only be verified in a much later stage. Once data is obtained, it might contain channel coding and/or be encrypted, which all takes place on the higher levels of the OSI model. Considering all above mentioned OFDM parameters and features, the technique of blindly obtaining OFDM signal parameters has been chosen for this thesis. The parameters which need to be known in order for an OFDM demodulator to function are [19]:

- Symbol length;
- Cyclic prefix length;
- Number of subcarriers;
- Information on pilot tones:
  - Location
  - Value

The conventional method is to provide these parameters directly to the OFDM demodulator based on prior knowledge. The question is: can these parameters be blindly obtained for an unknown OFDM signal and are they indeed sufficient to serve as the input for an OFDM demodulator with the aim to retrieve the signal data? This new approach is illustrated in figure 1.5. The OFDM signal reaches the demodulator with a delay while the signal parameters are obtained in the blind parameter extractor block.

[1]Assuming amplitude and/or phase modulation is being used. This is nog the case for frequency modulated signals.

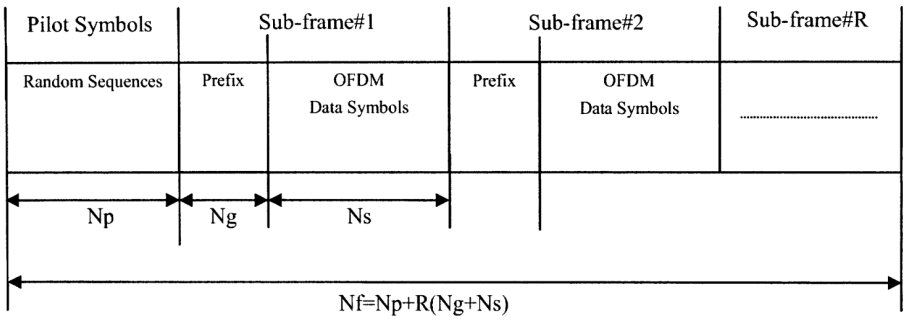


Figure 1.6: Frame structure of OFDM according to [20]

1.4 State of the Art

This section will provide a short overview of some of the current research related to the research objective of this thesis.

OFDM Blind Parameter Identification in Cognitive Radios [20] Ishii and Wornell describe an OFDM parameter identification algorithm for cognitive radio purposes. They obtain the number of symbols, the number of subcarriers, symbol length and pilot tone information using a correlation-based algorithm. They assume an OFDM frame structure as shown in figure 1.6. By differentiating between correlated data and random prefixes, estimates for e.g. symbol length can be made. However, the assumption of a strict frame build-up containing randomised preambles limits its implementation. Standardised OFDM implementations such as 802.11 do not meet this assumption.

Second-Order Cyclostationarity of Mobile WiMAX and LTE OFDM Signals and Application to Spectrum Awareness in Cognitive Radio Systems [21] Al-Habashna *et al.* devised a technique based on second-order cyclostationarity to differentiate between WiMAX and LTE OFDM signals. The possible parameters of WiMAX and LTE are presented and matched against obtained values. These obtained values have a relatively high margin of error, but by comparing the obtained values to the possible values from the specifications these errors are compensated. Undocumented links do not have a limited set of parameters which means this technique can not match them and the error can not be compensated.

Analysis of OFDM Parameters using Cyclostationary Spectrum Sensing in Cognitive Radio [22] Using cyclostationary analysis, primary users of OFDM are detected for cognitive radio purposes. The basic extracted features focus on centre frequency and occupied bandwidth. Based on these features, primary users and secondary users are categorised and unoccupied bands

in the frequency spectrum are selected for secondary users to occupy. The purpose of this paper is to use the frequency spectrum more efficiently which is why no further parameters needed for demodulation are taken into consideration. Even though OFDM is taken in a wider sense compared to previously described research, multiple parameters like subcarriers, symbol length and CP length are not considered.

Study of Cognitive Radio Spectrum Detection in OFDM Systems [23] An overview is presented of three main methods of spectrum detection focused on OFDM; matched filter detection, energy detection and cyclostationary detection. Matched filter detection is quickly skipped as it has a need for prior information. For energy detection to work at lower levels of SNR pilot tones are needed which also requires prior information on the signal. The cyclostationary technique is viable for detecting whether an OFDM signal is present, but does not provide information on the number of subcarriers or a fine estimate of the CP length.

In summary, most literature found is focused on cognitive radio applications. For this, less information is needed than for military applications that focus on signal demodulation. A possible reason for the lack of literature on the military applications might be the fact that military researchers do not publish their findings.

1.5 Contributions

Two techniques to blindly obtain OFDM signal parameters have been derived. First, a technique based on discontinuity detection has been created to blindly obtain OFDM signal parameters. This technique is successful in blindly obtaining the required parameters of noiseless, simulated signals. Applying this technique to simulated signals with practical additions like noise and filtering, however, did not provide the required results. The obtained understanding of OFDM did result in another, autocorrelation based technique. This technique proves more robust to the addition of filtering and noise.

A strong foundation and understanding of OFDM has been derived, opening the door for future researchers regarding blind OFDM signal parameter estimation. The two main contributions are:

- A discontinuity based blind parameter estimation technique for OFDM
- An autocorrelation based blind parameter estimation technique for OFDM

1.6 Outline

Chapter 2 will give an introduction to OFDM. The technique behind OFDM is explained and an overview is given on what is needed for OFDM to work in practical systems. Chapter 3 gives a theoretical solution to the problem of blind OFDM signal parameter estimation, explaining how inherent effects such as discontinuities can be exploited to obtain the necessary parameters. This remains however a theoretical solution, which is why chapter 4 elaborates on some practical consideration such as filtering. Chapter 5 presents the results of both the discontinuity based technique from chapter 3 and the autocorrelation based technique from chapter 4 and shows the limitation in relation to windowing and noise effects. Closing this thesis, chapter 6 provides a short, concluding summary and recommendations for future work.

2

OFDM Signal Structure

2.1 Multipath Delay Spread Tolerance

A wireless signal may reflect on physical objects, which in urban environments means a signal may bounce against vehicles, buildings etc. These reflections result in multiple delayed copies of the same signal, following different paths, reaching the receiver. This effect is illustrated in figure 2.1. These signals are known as multipath signals [17]. They may differ in time, phase and amplitude. The time difference between the first signal and the last signal reaching the receiver is called the delay spread, noted by T_{max}. This effect can lead to ISI, which distorts the signal. To prevent ISI, a guard interval, or Cyclic Prefix (CP), is needed[1]. The CP is a copy of the last section of an OFDM symbol waveform, copied and placed before the start of the symbol. This is done for each OFDM symbol waveform. An illustration of this technique is shown in figure 2.2. The length of the CP should be longer than the maximum delay spread given by T_{max}: $T_{cp} > T_{max}$.

$$T_{cp} > T_{max} \tag{2.1}$$

T_{SS} The symbol time is represented by T_{SS} and is without the CP, as shown in figure 2.2.

[1]Other implementations exist, like zero padded guard bands, but will not be discussed in this thesis.

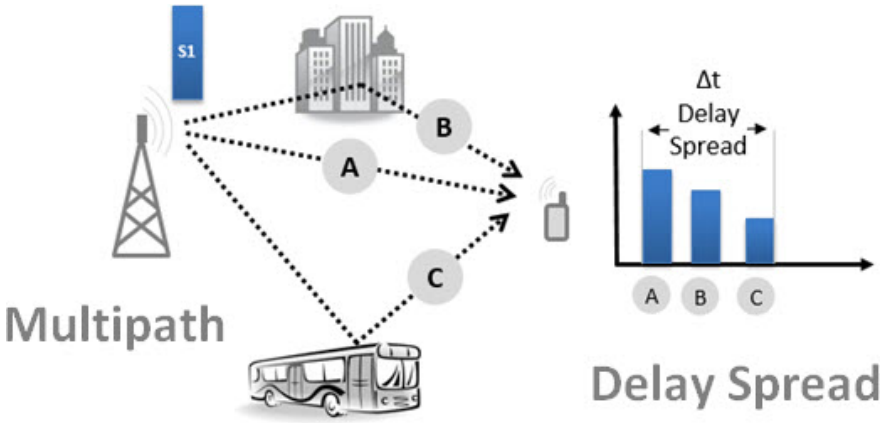


Figure 2.1: Multipath delay spread [24]

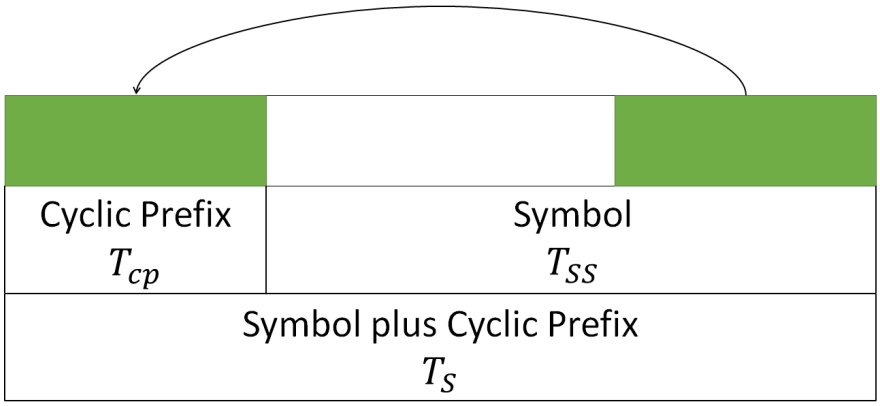


Figure 2.2: Symbol and CP timings

$\boldsymbol{T_{cp}}$ The length of the CP is given by T_{cp} as shown in figure 2.2.

$\boldsymbol{T_S}$ $T_S = T_{SS} + T_{cp}$ as shown in figure 2.2.

$\boldsymbol{T_{max}}$ The time delay spread between the first and last multipath signal reaching the receiver.

2.2 OFDM Signal Properties

When describing a signal being transmitted through a wireless medium, there are two main domains: time and frequency. These resources can be utilised in such a way to increase throughput efficiency. One example is when the frequency spectrum is fixed, this spectrum may be assigned to different users over time using TDM. The counter example is when the time domain is fixed, the frequency domain may be shared by dividing it into non-overlapping frequency bands using FDM. An illustration of this technique is shown in figure 2.3. The figure shows flat, wide and non-overlapping frequency bands, separated by overlapping parts called guard bands to prevent interference. Summing multiple arbitrarily modulated subcarriers results in what is known as a DMT type signal. Each DMT symbol can be written as the sum of N modulated carriers

$$s(t) = \sum_{k=1}^{N} A_n cos(\omega_c t + \phi_n) \tag{2.2}$$

where the individual modulation of each subcarrier is defined by the phasor notation $A_n \angle \phi_n$. Instead of using wide spaced frequency bands with large guard band, the use of orthogonal signals was proposed in 1966 with the idea of increasing bandwidth efficiency [25]. This idea combined with the implementation of the DFT gave rise to the concept of OFDM.

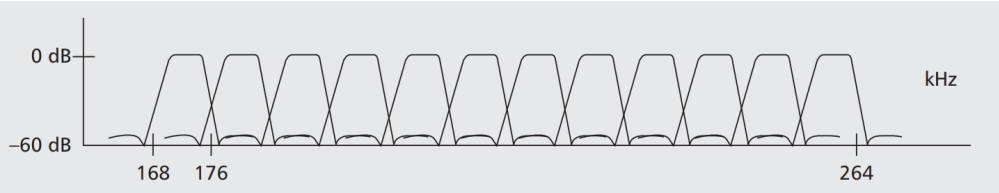


Figure 2.3: Frequency spectrum for 12 telephone subchannels [3]

2.2.1 Orthogonality

At this stage, N subcarriers are defined, centred around a centre carrier frequency but they are not necessarily orthogonal. To achieve this, the definition of orthogonality for functions s_i and s_k is examined. In order to achieve orthogonality between two subcarrier signals s_i and s_k,

$$\int_0^{T_{SS}} s_l(t) \cdot s_k^*(t)dt = 0 \quad \text{for} \quad l \neq k\,, \;\; l,k \in \mathbb{Z} \tag{2.3}$$

where $T_{SS} = \frac{N}{R_S}$ is the subcarrier symbol time and R_S is the input symbol rate which is split up in N parallel data streams of subcarrier symbol rate $R_{SS} = \frac{R_S}{N}$. One subcarrier signal is given by

$$s_i(t) = e^{j2\pi f_n t} \quad \text{for} \quad i \in \mathbb{Z} \tag{2.4}$$

where

$$f_n = f_0 + n\Delta f \tag{2.5}$$

Inserting equation 2.4 into equation 2.3 gives

$$\begin{aligned} \int_0^{T_{SS}} s_l(t) \cdot s_k^*(t)dt &= \int_0^{T_{SS}} e^{j2\pi(f_0 + l\Delta f - f_0 - k\Delta f)t}dt \\ &= \int_0^{T_{SS}} e^{j2\pi(l-k)\Delta f t}dt \\ &= \frac{e^{j2\pi(l-k)\Delta f T_{SS}} - e^{j2\pi(l-k)\Delta f 0}}{j2\pi(l-k)\Delta f} = 0 \end{aligned} \tag{2.6}$$

if and only if

$$e^{j2\pi n} = 1 \quad \text{for} \quad n \in \mathbb{Z} \tag{2.7}$$

which is the case if

$$n = l - k \quad \text{for} \quad n \in \mathbb{Z}, \;\; n \neq 0 \tag{2.8}$$

and

$$\Delta f = \frac{1}{T_{SS}} \tag{2.9}$$

The symbol time is then related to the frequency separation to obtain orthogonality between the subcarriers by

$$T_{SS} = \frac{1}{\Delta f} \tag{2.10}$$

2.2.2 FFT/IFFT

An OFDM signal at a sampling frequency equal to the symbol rate R_S results in a symbol representation by

$$\frac{1}{N}\sum_{n=0}^{N-1} A_{n,i}e^{j(2\pi f_n pT_S+\phi_{n,i})} = \frac{1}{N}\sum_{n=0}^{N-1}\left(A_{n,i}e^{j(2\pi f_0 pT_S+\phi_{n,i})}\right)e^{j2\pi np\Delta fT_S} \tag{2.11}$$

Comparing equation 2.11 to the generalised discrete IFFT, given by

$$g(pT) = \frac{1}{N}\sum_{n=0}^{N-1} G(n\Delta f)e^{j2\pi np/N} \tag{2.12}$$

shows equations 2.11 and 2.12 are equal where

$$A_n e^{j(2\pi f_0 pT_S+\phi_n)} \tag{2.13}$$

is the frequency domain signal and

$$\Delta f = \frac{1}{NT_S} \tag{2.14}$$

which is the same requirement as for orthogonality

$$\Delta f = \frac{1}{T_{SS}} = \frac{1}{NT_S} \tag{2.15}$$

as shown in section 2.2.1. This means that due to the fact that OFDM consists of orthogonal subcarriers, the signal can be defined by the inverse Fast Fourier transform. Taking an FFT from one OFDM symbol will therefore separate the subcarrier into individual subcarrier symbols.

2.2.3 Block Oriented Modulation

OOK is an example of a single carrier modulation scheme [26]. An OOK signal is represented by

$$s_i(t) = Am(t)cos(\omega_c t) \tag{2.16}$$

where $m(t)$ is a unipolar baseband data signal. The time representation of this signal is shown in figure 2.4. One clear observation to be made is that OOK streams data serially. Bits are modulated one after another, with no dependency between them. OFDM on the other hand, is a multi carrier modulation scheme which combines bits into blocks, and modulates them in parallel. Each subcarrier is modulated using single carrier modulation and summed up to obtain an OFDM symbol. This principle is illustrated in time and frequency domain in figure 2.5. In an OFDM system with N subcarriers, the i-th symbol can be represented by

$$s_i(t) = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1} A_{n,i}e^{j(2\pi f_n t+\phi_{n,i})}, & \text{for } 0 \le t < T_{SS} \\ 0, & \text{otherwise} \end{cases} \tag{2.17}$$

Symbol The term symbol is being used to describe the OFDM waveform of length T_{SS}, equal to the modulation order m times the number of subcarriers N.

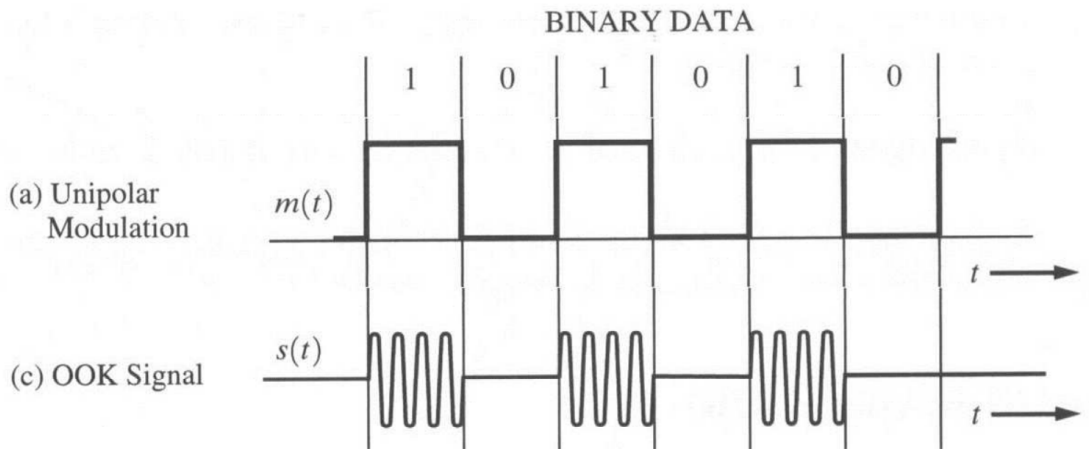


Figure 2.4: Bandpass OOK and unipolar signals [26]

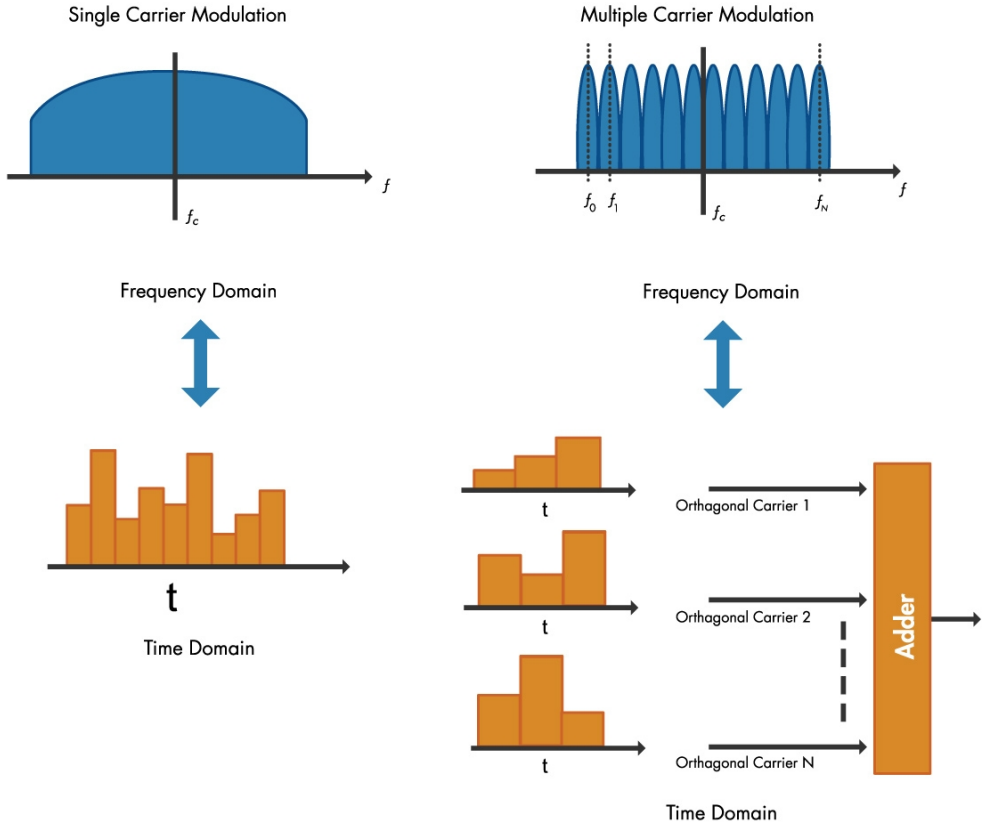


Figure 2.5: Single carrier modulation and OFDM in time and frequency domains [27]

N N is the number of subcarriers and the number of samples needed to demodulate the OFDM symbol, also known as the minimal FFT size. The individual subcarriers are denoted by n.

L The number of OFDM symbols in a signal is denoted by L, with the individual symbols being denoted by i.

Single symbol The i-th symbol is represented by

$$s_i[p] = \sum_{n=1}^{N} A_n e^{j\phi_n} e^{j2\pi f_n n} \quad \text{for} \quad f_n = f_0 + n\Delta f \tag{2.18}$$

where A_n is the amplitude of the n-th subcarrier and ϕ_n is the phase of the n-th subcarrier.

Full waveform An OFDM signal is represented by

$$s_s(t) = \sum_{i=0}^{L-1} [s_c p(t)\delta(t - i(T_{SS} + T_{cp})) + s_i\delta(t - T_{cp} - i(T_{SS} + T_{cp}))], \quad \text{for} \quad 0 \leq i \leq L\,, n \in \mathbb{Z} \tag{2.19}$$

with

$$s_c p(t) = s_i(t)w(t) \tag{2.20}$$

for $w(t)$ a windowing function with

$$w(t) = 1 \quad \text{for} \quad 0 \leq t \leq T_{cp} \quad \text{otherwise } w(t) = 0 \tag{2.21}$$

Waveform notation A sampled OFDM waveform consisting of symbols and their CP is denoted by $s_s[p]$. A single symbol without a CP is represented by $s_i[p]$ for $0 < i \leq L$ and the individual subcarrier signals which result in the OFDM symbol are denoted by $s_n[p]$ for $0 < n \leq N$.

2.2.4 Waveform Discontinuity

As clearly visible from the figures and given by the use of the FFT, the resulting waveform is periodic. The start of each symbol is solely depended on the input data which are summed up over the subcarriers. This means the initial value for each symbol may differ. When plotting multiple symbols after one another, this becomes visible as clear discontinuities as shown in figure 2.6. Where single carrier signals are a stream of bits, OFDM is the result of a summation of the symbols over N subcarriers.

2.2.5 Comparing Spectral Efficiency

OFDM, as mentioned before, uses orthogonal subcarriers. This decreases the occupied bandwidth compared to single carrier modulation with identical symbol rate R. The absolute transmission

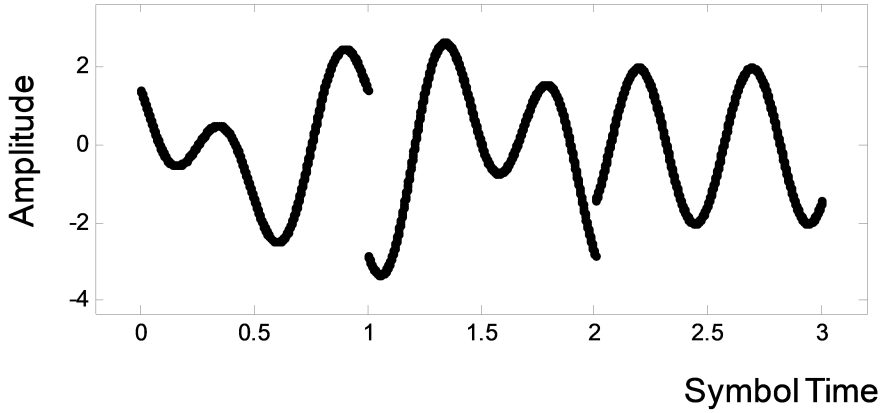


Figure 2.6: OFDM waveform containing three symbols without CP [28]

bandwidth for a QAM signal with rectangular pulses is [26]

$$B_T = \frac{2R}{\log_2 M} \text{ Hz} \tag{2.22}$$

The spectral efficiency of a QAM signal with rectangular pulses is given by

$$\eta_{QAM} = \frac{R}{B_T} = \frac{\log_2 M}{2} \frac{\text{bits/s}}{\text{Hz}} \tag{2.23}$$

For OFDM the transmission bandwidth is

$$B_{TOFDM} = \frac{N+1}{T_S S} = \frac{N+1}{NT_S} \approx \frac{1}{T_S} = R_S \text{ Hz} \quad \text{for} \quad N >> 1 \tag{2.24}$$

The actual bit rate of OFDM signals is reduced due to the CP, since each OFDM symbol is transmitted with an extra overhead. This means the actual bit rate equals

$$R_B = \frac{R_S \log_2 M}{1+cp} \quad \text{for} \quad 0 \leq cp \leq 1 \tag{2.25}$$

and the spectral efficiency equals

$$\begin{aligned} \eta_{OFDM} &= \frac{R_b}{B_{TOFDM}} \\ &= \frac{N \log_2 M_{sub}}{(1+N)(1+cp)} \\ &= \frac{log_2 M_{sub}}{1+cp} \frac{\text{bits/s}}{\text{Hz}} \quad \text{for} \quad N > 10 \end{aligned} \tag{2.26}$$

$\boldsymbol{M}$ Modulation level, where M_{sub} is the modulation level of the subcarrier modulation scheme.

$\boldsymbol{R_S}$ Symbol rate of one OFDM symbol.

$\boldsymbol{R_b}$ Bit rate of the signal, $R_b = R_S \log_2 M$.

$\boldsymbol{R_{SS}}$ Subcarrier symbol rate, equal among all subcarriers: $R_{SS} = \frac{R_S}{N}$.

2.3 OFDM Signal Modulation

In this section, QPSK is being used to illustrate the steps taken to go from bits to an OFDM transmit symbol. Theoretically, any type of subcarrier modulation can be used. Practically, the hardware used often limits the possible modulation types. The number of subcarriers and the location and value of the pilot tones need to be known before generating the OFDM waveform. In this example there are 6 subcarriers,$N = 6$, and the pilot tones are located on subcarriers $n = -2$ and $n = 2$ and have a constant BPSK modulated 1 set.

2.3.1 From Bits to Blocks

The first step is converting the input bit stream into blocks, depending on the subcarrier modulation order. The modulation order is given by the number of bits per symbol. For a single carrier modulation with m levels,

$$O = \log_2(m) \tag{2.27}$$

which in the case of QPSK, or 4-PSK, means the modulation order is $\log_2(4) = 2$ bits per symbol. Data is grouped in to blocks of size ON, the modulation order times the number of subcarriers. This means that for 4 subcarriers, the input information stream is grouped per block of 8 bits in to symbols of 2 bits,

$$[00011011] => [00][01][10][11] \tag{2.28}$$

O The order of the subcarrier modulation is denoted by O, $O = \log_2(m)$ where m is the modulation level, i.e. for QPSK $m = 4$.

2.3.2 From Blocks to Signal States

The next step is converting the input bit symbols in to signal states on the IQ plane, in accordance with the modulation and bit mapping used. In this example, QPSK with Gray coding is being used as illustrated in figure 2.7. For data input [00][01][10][11], this results in the data elements illustrated by figure 2.8

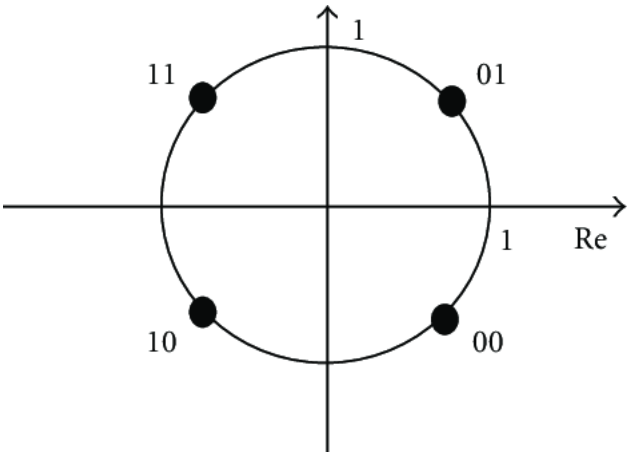


Figure 2.7: QPSK Gray Coding [29]

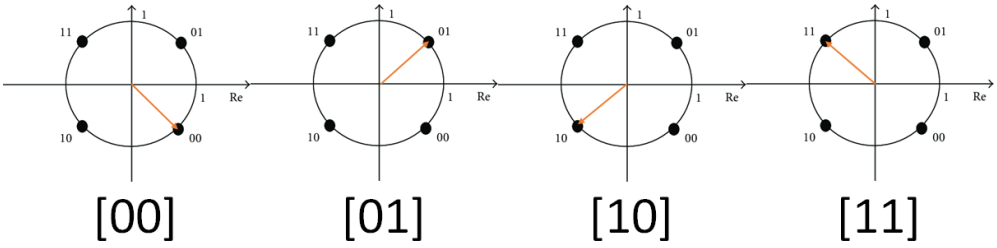


Figure 2.8: Data elements per data subcarrier

or in IQ notation by

$$\begin{aligned}
[00] &= \frac{1}{2}\sqrt{2} - j\frac{1}{2}\sqrt{2} \\
[01] &= \frac{1}{2}\sqrt{2} + j\frac{1}{2}\sqrt{2} \\
[10] &= -\frac{1}{2}\sqrt{2} - j\frac{1}{2}\sqrt{2} \\
[11] &= -\frac{1}{2}\sqrt{2} + j\frac{1}{2}\sqrt{2}
\end{aligned} \tag{2.29}$$

2.3.3 From Data Elements to Subcarrier Signals

Using the FFT, the data elements are converted to waveforms. Note that the subcarriers -2 and 2 are not used at this stage, because they are reserved for pilot tones. Data is mapped onto the remaining subcarriers following the single carrier modulation scheme used, as shown in figure 2.9. The length of each subcarrier is one symbol time, or T_{SS}.

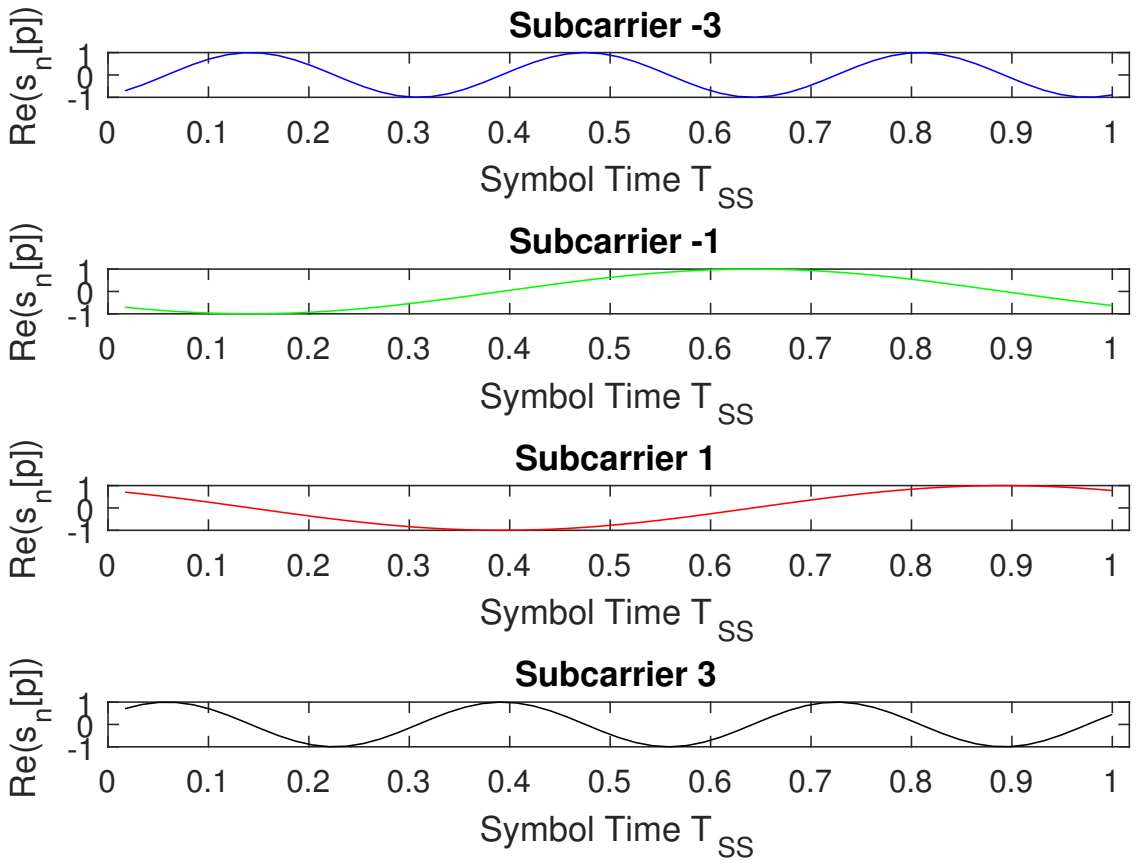


Figure 2.9: Individual Subcarrier Waveforms

2.3.4 Pilot Tones

The next step is inserting the pilot tones. The pilot tones, [1] are BPSK modulated, are inserted on the reserved subcarriers. Note that the pilot tones do not carry data but are special symbols added for synchronisation purposes. If the number of subcarriers is limited, the data rate of the OFDM signal goes down with the insertion of pilot tones since the number of data subcarriers decreases, $n_d = n - n_p$. If the number of the pilot tones are considered "extra" subcarriers, the bandwidth increases which decreases the spectral efficiency. This is illustrated in figure 2.10. The length of each pilot tone is one symbol time,T_{SS}.

s_p The number of pilot tones is denoted by s_p, $0 < s_p < n$.

s_p The location of the pilot tones is denoted by n_p where n_p is the subcarrier index where the pilot tone is located.

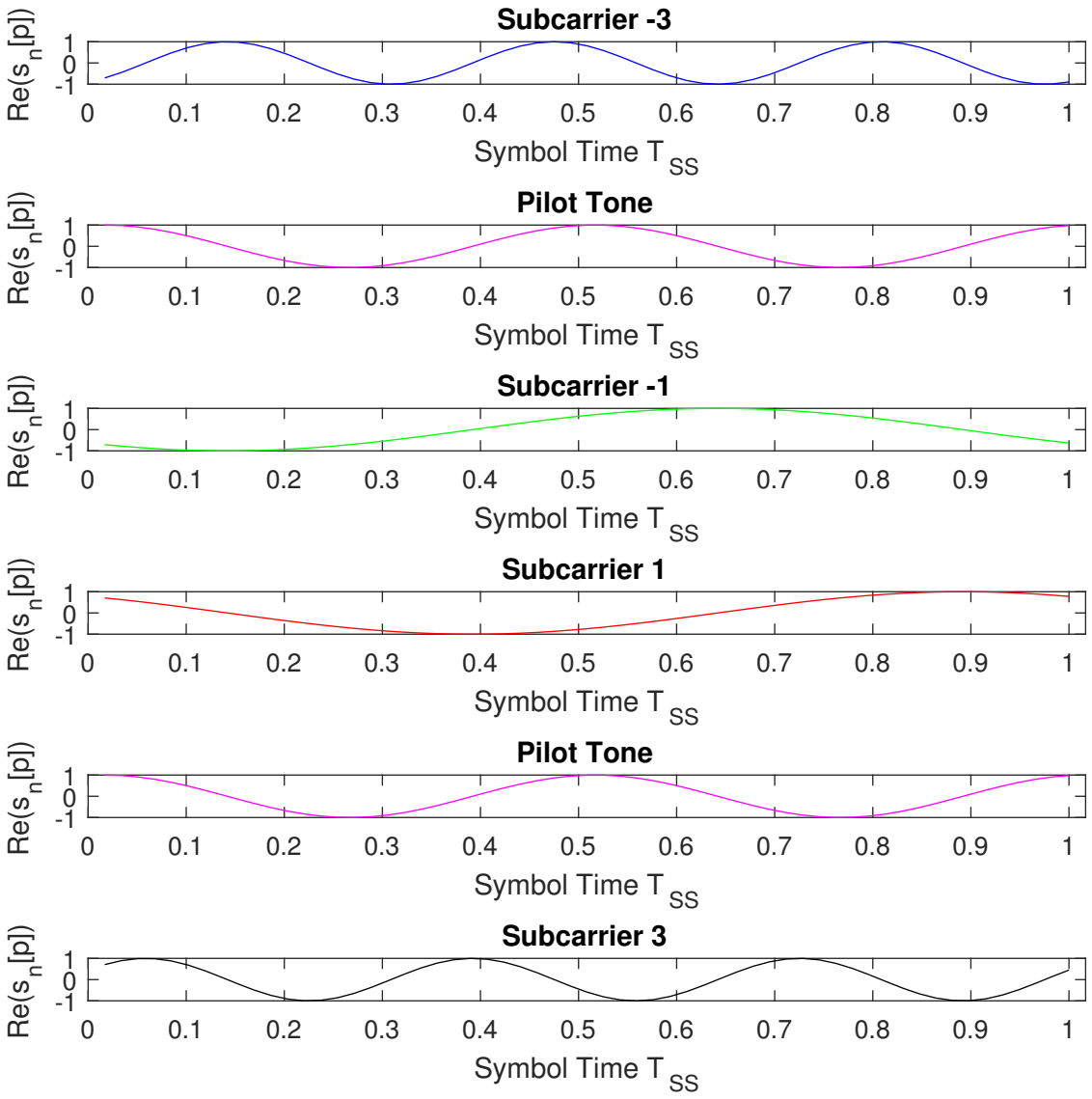


Figure 2.10: Inserting the pilot tones

2.3.5 From Summation to Symbols

Once all waveforms are generated in parallel, they are summed up into one symbol following equation 2.17, as shown in figure 2.11. The summed up symbol also has a length of one symbol time, or T_{SS}.

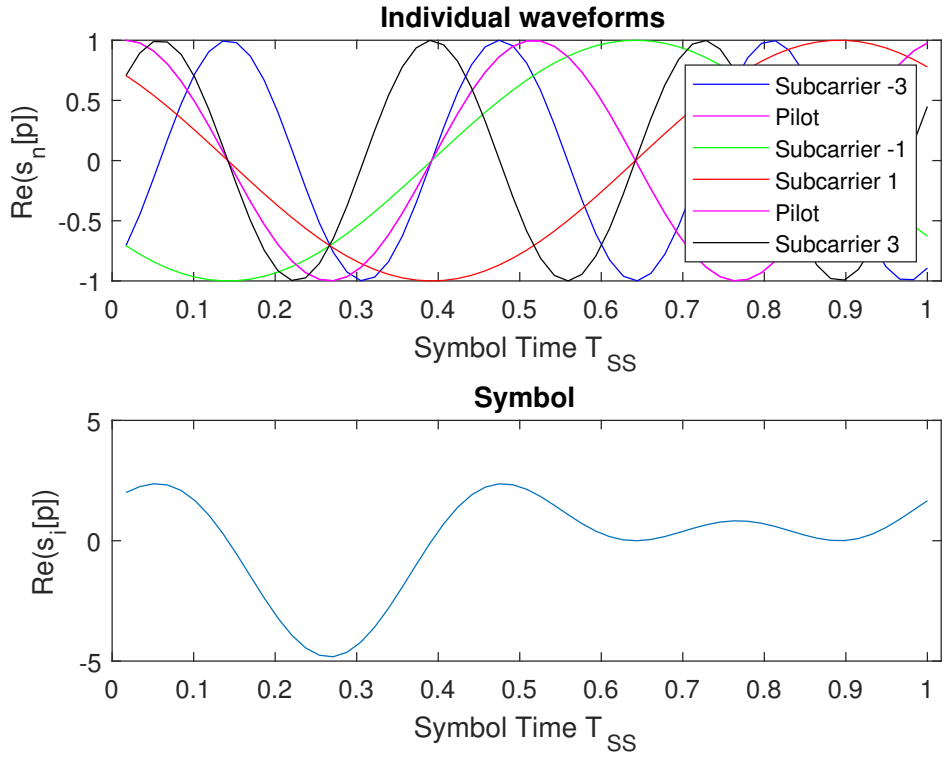


Figure 2.11: Summing the waveforms into one symbol

2.3.6 Cyclic Prefix Insertion

To prevent ISI, the CP is added per symbol. A signal consisting of multiple symbols M is given by

$$s_s(t) = \sum_{i=0}^{M-1} s_i(t)\delta(t - iT_{SS}) \tag{2.30}$$

where $\sum_{i\in\mathbb{Z}} \delta(t - iT_{SS}) = \text{Ш}_{T_{SS}}(t)$ is the Dirac comb with intervals equal to the symbol time T_{SS}. The result is a series of Dirac delta functions. The CP is then added per symbol by making an exact copy of length T_{cp} from the end of the OFDM symbol, which due to the periodicity of the signal becomes

$$s_i(t) = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1} A_{n,i} e^{j(2\pi f_n + \phi_{n,i})t}, & \text{for } -T_{cp} \leq t < T_{SS} \\ 0, & \text{otherwise} \end{cases} \tag{2.31}$$

The complete symbol plus CP has a length of $T_S = T_{SS} + T_{cp}$. In this example, a CP of 25% of the symbol time is added before the OFDM symbol, as shown in figure 2.12.

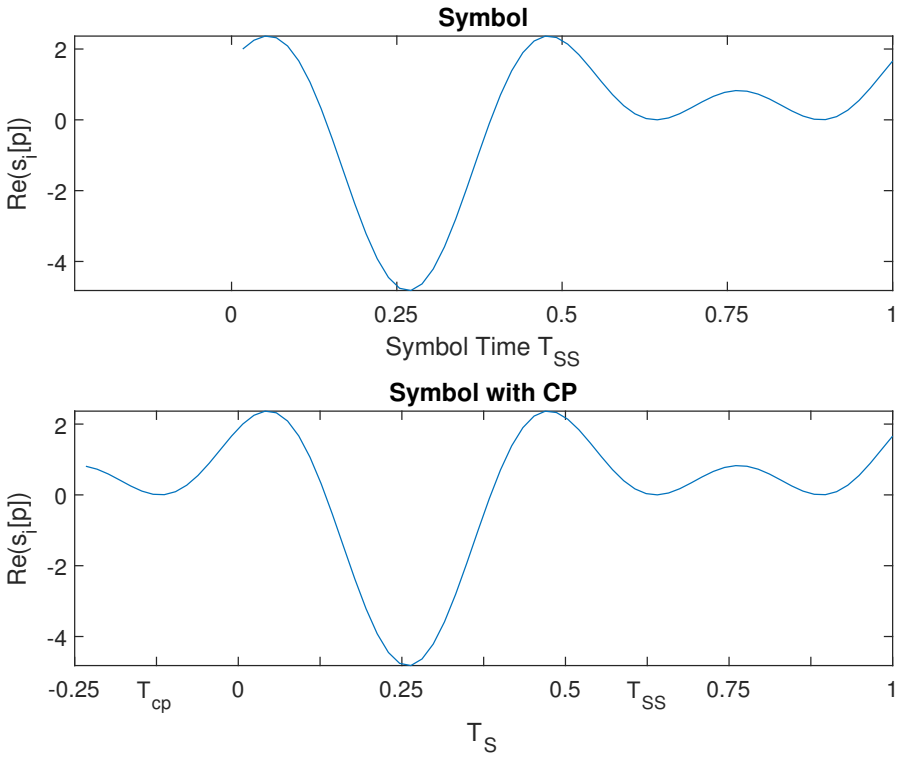


Figure 2.12: Adding 25% Cyclic Prefix

This process is done for each symbol, resulting in a stream of symbols each representing a block of information input data.

2.4 OFDM Signal Demodulation

The receiver receives the signal as complex data. The receiver has to reverse the steps taken by the transmitter in order to obtain the information input data.

2.4.1 Reverse Process

Separating the received OFDM symbol is essentially by the same process performed by the transmitter but in reverse. This is summarised as:

- Move the signal to baseband;
- Determine the symbol timing;
- Remove CP;
- Convert using FFT;
- Separate per subcarrier and demodulate;

- Convert to information bits.

In this thesis, we assume the signal is in baseband. The next step is determining the symbol timings. In practical systems, the symbol time or symbol rate and CP is known by the receiver before hand.

With this knowledge, a symbol with its CP can be extracted. The result is the opposite of what is illustrated in figure 2.12. The symbol without the CP is then converted using an FFT to single symbol states per subcarrier. The type of subcarrier modulation is also assumed known by the receiver, which enables the conversion of the individual waveforms to data. This entire process is illustrated in figure 2.13.

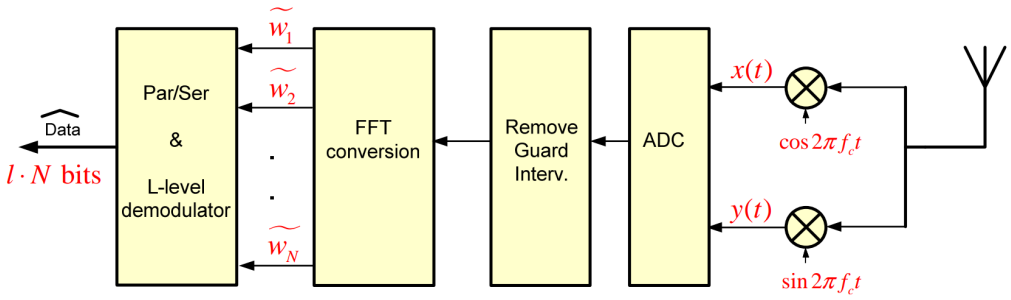


Figure 2.13: OFDM receiver model [17, slide 40]

2.4.2 Synchronisation

Synchronisation is needed because timing and frequency offsets often result in a loss of orthogonality between the subcarriers. A more complete overview of the effects of non-synchronous OFDM is presented by H. Zhou et-al [30].

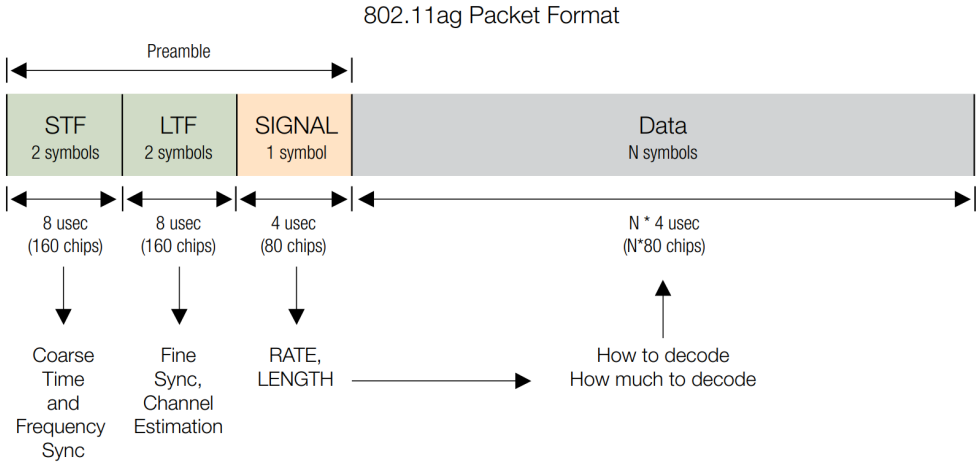


Figure 2.14: Physical layer frame of 802.11a/g [31]

As shown in figure 2.14, in a practical implementation part of the physical layer frame can be reserved

for synchronisation purposes. In 802.11a/g, there are two synchronisation fields in the physical layer frame: the STF and the LTF. In the example of 802.11g, the pilot tones are divided as illustrated in figure 2.15. The STF is 2 symbols long and during this time, $\frac{1}{4}$ of the 52 subcarriers are being used for initial timing synchronisation and frequency estimation. Among these 13 subcarriers are the 4 pilot tones. The LTF is also 2 symbols long and uses all 52 subcarriers for fine synchronisation.

Besides the example above, other techniques exist to achieve synchronisation in OFDM systems. An overview of multiple synchronisation techniques is presented by C.D. Parekha et-al [32].

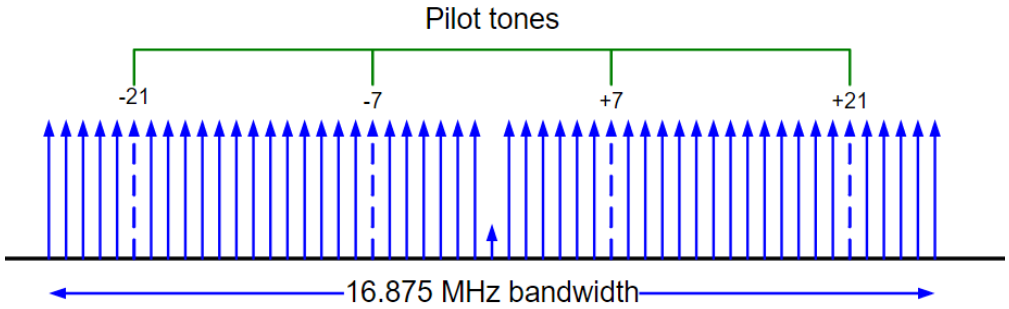


Figure 2.15: Structure of a 20 MHz OFDM 802.11g channel [17, slide 43]

2.4.3 Parameters needed

In order to achieve the demodulation steps, as described above, some essential parameters are needed. To extract a symbol and its CP, both lengths must be exactly known. This is also necessary for the next step, removing the CP. To convert the signal to the frequency domain using an FFT, the length of this N-point FFT must be known. Finally, for further synchronisation information on the pilot tones such as the locations and values must be known.

In summary, the key parameters for an OFDM demodulator are:

- Cyclic prefix length T_{cp};
- Symbol length T_{SS};
- Number of subcarriers N;
- Information on pilot tones:
  - Location of the pilot tones
  - Data value of the pilot tones

2.5 Conditions

This chapter presented an overview of a standard OFDM signal. Other OFDM techniques exist, such as

- ZP-OFDM [33]
- OFDM without CP [34]

These techniques will not be discussed further in this thesis. The standard form of OFDM seems most common, based on popular practical implementations such as WiFi and LTE. Adapting the techniques presented for these other types of OFDM is recommended for future work. The initial assumptions are chosen to simplify the process and gain a better understanding of OFDM. They include:

- An OFDM signal is present;
- The IQ baseband signal is available;
- A CP between 0, not including 0, and the symbol length: $0 < T_{cp} < T_{SS}$
- All data subcarriers have the same and constant modulation technique over the sample set;
- There are pilot tones at constant locations and with constant values;
- The number of samples per symbol is:
  - a power of 2 and
  - at least equal to or higher than the number of subcarriers

3

OFDM Parameter Estimation

This chapter introduces a innovative technique to obtain the OFDM parameters necessary as input for an OFDM demodulator. The parameters deemed necessary, as explained in section 2.4, are:

- Cyclic prefix length T_{cp};
- Symbol length T_{SS};
- Number of subcarriers N;
- Information on pilot tones:
  - Location
  - Value

As explained in chapter 2, an OFDM signal contains (multiple) OFDM transmitted symbol(s) of length T_S. Each transmitted symbol consists of an OFDM symbol of length T_{SS} and a CP of length T_{cp}, as illustrated in figure 3.1.

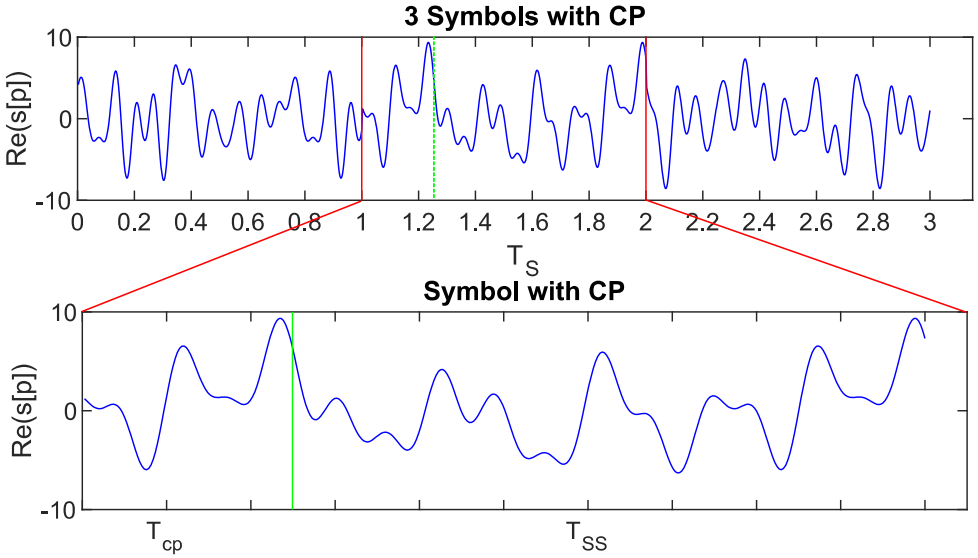


Figure 3.1: T_S, T_{SS} and T_{cp} illustrated

3.1 Test Signals

Before signals can be analysed, they have to be generated. The test signals in this thesis are generated following the steps in appendix A and the code presented in Appendix B.4. Existing OFDM signal generation tools such as the Wireless Waveform Generator from Matlab are able to generate OFDM signals, but to gain a better understanding of the techniques behind OFDM the choice was made to design an OFDM signal generator from scratch [35]. This OFDM waveform generator provides greater flexibility and can provide the intermediate signals and other variables.

The test signals for this chapter are generated using the following parameters:

- Random input data;
- 8-PSK subcarrier modulation;
- 4 constant pilot tones, bit value 1, modulated using BPSK;
- Total number of subcarriers: $N = 52$;
- Cyclic Prefix length: 25%, $T_{cp} = 256\ samples$;
- Symbol time: $T_{SS} = 1024\ samples$;
- Number of symbols: $M = 50$.

The information input data chosen for the test signals is random, using a uniformly distributed pseudorandom integer function for values of 0 and 1 for equal probability. The other signal parameters have been chosen from within the scope of the signal generator, striking a balance between complexity, known practical implementations and simulation time. The result is a baseband, complex IQ data stream of the generated signal denoted by $r[p]$.

$\boldsymbol{r[p]}$ The sampled OFDM signal is defined as $r[p]$ where p denotes the sample for $0 < p < L$ and $p \in \mathbb{Z}$.

3.2 Symbol Boundaries

The first challenge is finding the symbol boundaries. Knowing where symbols start and end is the first step taken in obtaining the necessary parameters.

3.2.1 Signal Property

As explained in section 2.2.3, OFDM is a block oriented modulation technique which as explained in section 2.2.4, means there is no guaranteed waveform continuity. This is a signal property which can be exploited in order to find the symbol boundaries. Since a discontinuity occurs between symbols,

the space between two consecutive discontinuities, located at $q[k]$ and $q[k-1]$, is the symbol time plus the corresponding CP time

$$q[k+1] - q[k] = T_S[p] \tag{3.1}$$

$\boldsymbol{q[k]}$ Vector $q[k]$ is the series of samples where discontinuities occur. $q[k+1]-q[k]$ gives the number of samples between two consecutive discontinuities.

As explained in section 2.5, the sampling of $r[p]$ in this chapter is assumed to be constant and meet all requirements. Therefore it is assumed $T_S[p]$ is constant for all $q[k+1] - q[k]$ and will be noted as T_S hereafter. The case of $T_S[p]$ being not constant due to sampling errors will be discussed in the next chapter.

3.2.2 Method

To calculate T_s, the locations of $q[k]$ and $q[k+1]$ need to be determined. $q[k]$ is the start of a symbol and $q[k+1]$ is the start of the next symbol. This can be done by determining the amplitude difference between two consecutive samples using

$$\dot{s}[p] = r[p+1] - r[p]\,, \quad \text{for} \quad 0 < n \leq L\,,\, n \in \mathbb{Z} \tag{3.2}$$

Using a threshold detection algorithm, the values for p can be identified for which a discontinuity occurs. These values of p are then stored in vector $q[k]$

3.2.3 Resulting Algorithm

There is a threshold taken into consideration called ϵ, where the discontinuities are selected over normal sample transitions. The result is a peak finder over the differential of the signal.

Data: Input signal, complex
Result: T_S and vector $q[k]$
$q[k] = r[p] \quad \text{for} \quad \dot{s}[p] > \epsilon$;
$T_S = q[k+1] - q[k]$;

Algorithm 1: Determining discontinuity locations $q[k]$

3.2.4 Simulation Results

The result is vector $q[k]$ and T_S, the symbol time plus the CP time. This method is illustrated in figure 3.2. The first waveform shows the real component of $r[p]$. The differential is shown as the second figure, where clear spikes are visible. The location of these spikes are used as input for vector $q[k]$ and are plotted over the real waveform in the bottom figure. The sample number of the symbol transitions are now known and stored in $q[k]$.

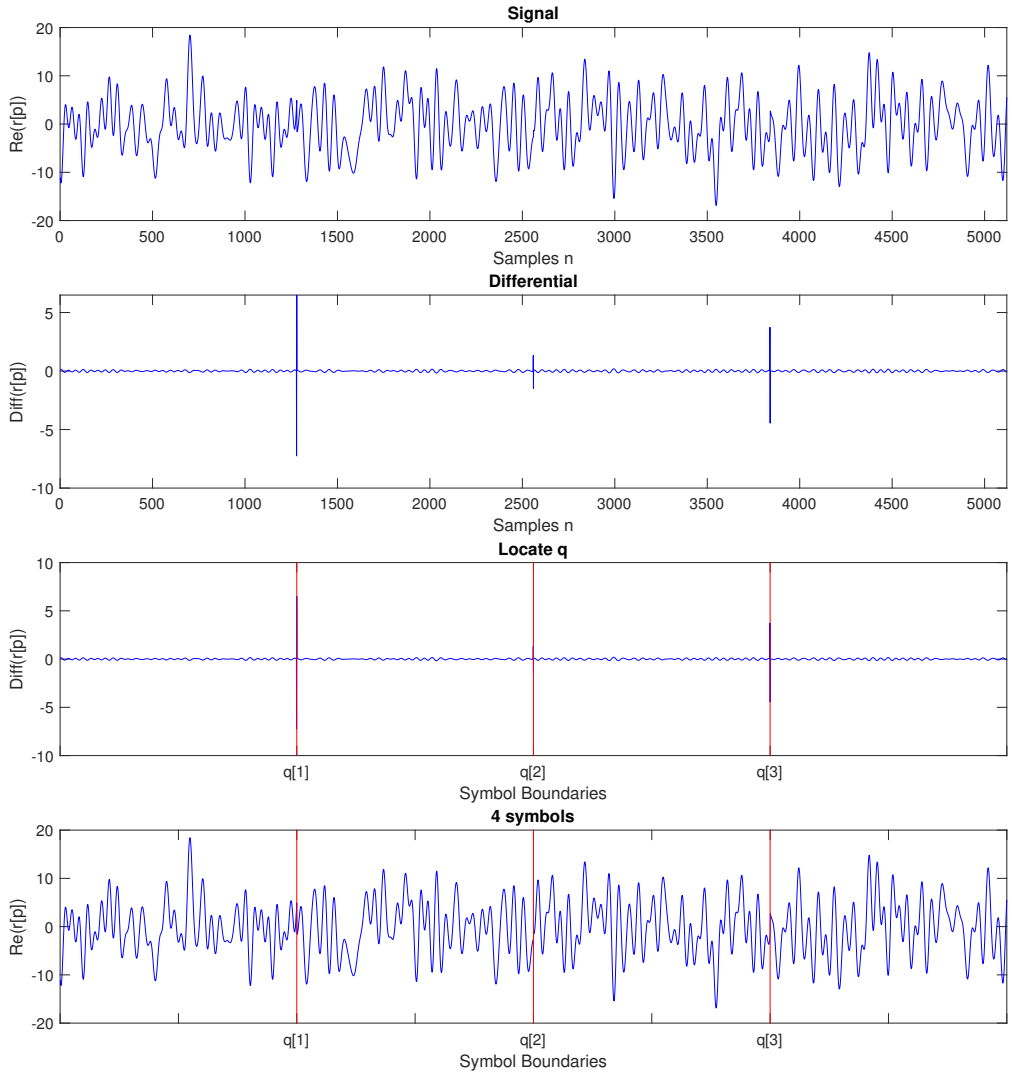


Figure 3.2: Determining the locations of $q[k]$

3.3 Symbol and Cyclic Prefix Length

The next parameters that will be determined are the symbol length T_{SS} and the CP length, T_{cp}.

3.3.1 Signal Property

As explained in section 2.3.6, the CP is an exact copy of a certain percentage of the end of the OFDM symbol which is put infront of the symbol. This property can be exploited to find where samples have identical copies within the symbol time T_S. There will always be exactly one identical copy in a noiseless OFDM symbol. This is because the first subcarriers around the centre frequency have exactly one phase rotation, which means the amplitude and phase is unique over the symbol

time T_{SS}. Any summation including these subcarriers will also result in unique values over the symbol time T_{SS}.

3.3.2 Method

From section 3.2 the start of the symbols is obtained in the form of vector $q[k]$. Since $q[k]$ indicates the start of the CP, there has to be an exact copy of this sample before $q[k+1]$. T_{SS} is the value of c for which $r[q[k]+c]-r[q[k]]=0$, given by

$$T_{SS} = c \quad \text{for} \quad r[q[k]+c]-r[q[k]]=0 \tag{3.3}$$

As explained in section 3.2, the assumption for correct sampling is also made here which means T_{SS} will be a constant. With T_{SS} and T_S known, T_{cp} is obtained following

$$T_{cp} = T_S - T_{SS} \tag{3.4}$$

3.3.3 Resulting Algorithm

The discrete OFDM signal is denoted by $r[p]$ and has a length L. The start of the symbols in vector $q[k]$, such that $r[q[k]]$ is the value of the first sample of a symbol. The implementation of this method is given in algorithm 2.

Data: Discrete input signal $r[p]$, complex; vector $q[k]$; T_S
Result: T_{SS} and T_{cp}
Find value n such that
$r[q[1]] = r[q[1]+n] + \epsilon$;
for $q = 2 : T_S$ **do**
  if $r[q[k]] - r[q[k]+q] > \epsilon$ **then**
    break
  end
end
$T_{cp} = q$
$T_{SS} = T_S - T_{cp}$

Algorithm 2: Determining the length of a symbol and the CP

3.3.4 Simulation Results

The steps taken in 2 are illustrated in figure 3.3. The first figure is where section 3.2.4 left off. A copy of the first sample is located within the samples up to the next symbol, $2 < q < T_S$. The location of the sample with the same amplitude and phase is marked as shown in the second figure. The number of samples at the start of the symbol is increased and correlated with the samples at the end of the symbol as shown in the third figure. Once the sample sets are no longer equal, this

process is stopped and the CP is obtained as shown in the fourth figure. The bottom figure shows what remains per transmitted symbol, the OFDM symbol.

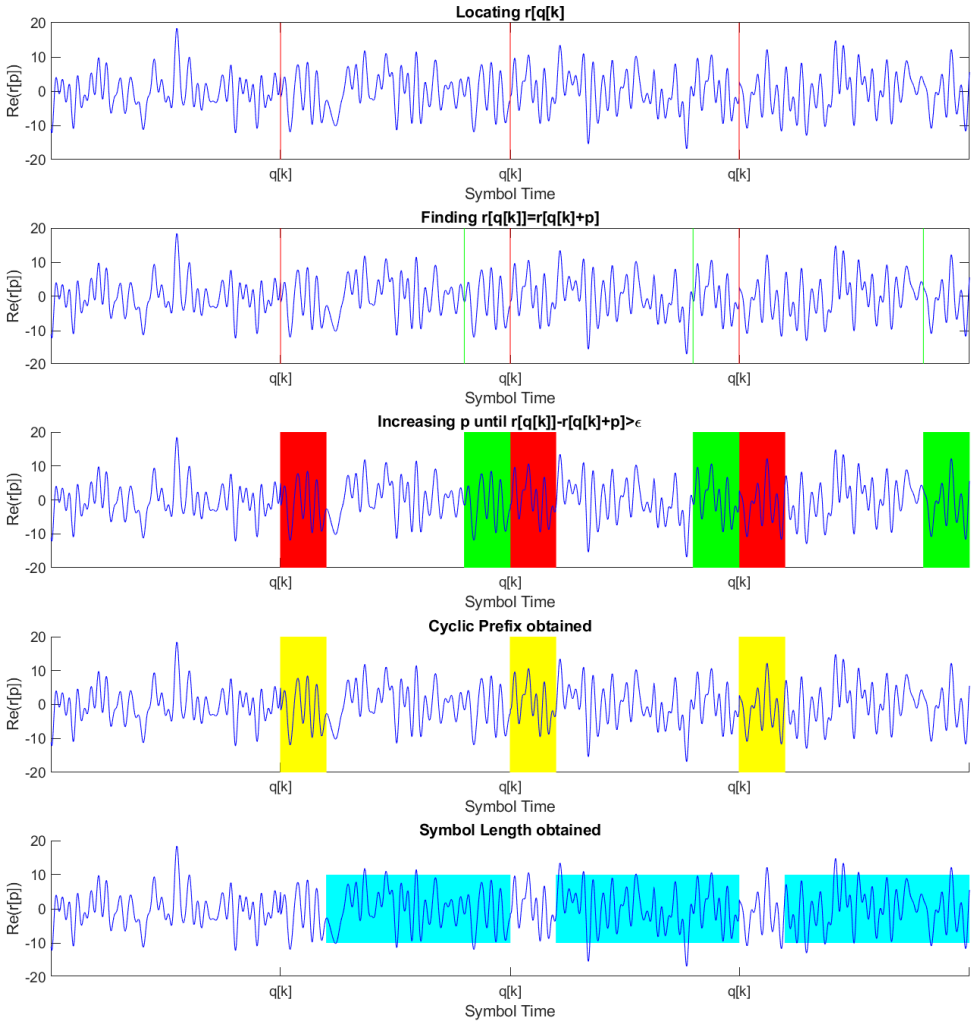


Figure 3.3: Obtaining the symbol length and the cyclic prefix length

3.4 Number of Subcarriers

With the obtained parameters of section 3.3, individual symbols can be extracted and placed in parallel. As explained in section 2.4, the next step is converting the symbol using an FFT and separating the subcarriers.

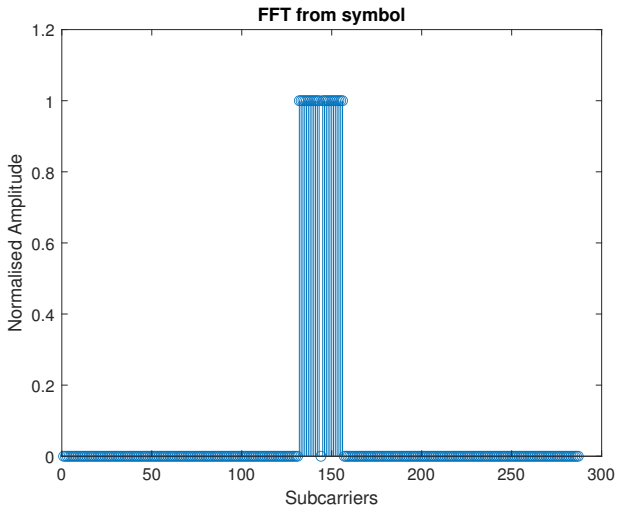


Figure 3.4: FFT of one symbol

3.4.1 Signal Property

As described in appendix 2.2.2, the FFT of a symbol results in the individual subcarrier symbols. Based on the conditions from section 2.5, "The number of samples per symbol is a power of 2 and at least equal to or higher than the number of subcarriers". This condition enables the use of the FFT for subcarrier separation. Less samples per symbol would result in a loss of orthogonality since the subcarriers will overlap and cause interference.

3.4.2 Method and Simulation

The Fourier transform is taken from the i-th symbol to obtain S_i,

$$s_i[p] \xrightarrow{\mathscr{F}} S_i[F] \tag{3.5}$$

Determining the non-zero locations of S_i gives the frequency bins which hold subcarriers.

Data: Symbol $s_i[p] \in \mathbb{C}$;
Result: N
$s_i[p] \xrightarrow{\mathscr{F}} S_i[F]$
if $S_i[p] > 0$ **then**
  $S_i[p] =$ subcarrier, $N = N + 1$
end

Algorithm 3: Determining the number of subcarriers

The result is illustrated in figure 3.4. Each peak represents a subcarrier, having a normalised amplitude of 1 due to the use of PSK as subcarrier modulation scheme.

3.5 Pilot tones

The next step is obtaining the required information on the pilot tones.

3.5.1 Signal Property

As explained in section 2.5, the pilot tones are assumed to be on constant locations and have constant values throughout the signal. The location on the IQ plane of each subcarrier can be extracted using an FFT. Since all carriers have random data, except the pilot tones, all carriers should have random locations on the IQ plane. The pilot tones however, are constant values which can be used by the receiver for synchronisation. They are basically sinusoidal waves, separated over multiple subcarriers.

3.5.2 Method & Algorithm

Using the variance function, the variance of each subcarrier over M symbols is calculated. The result is sorted from low to high after which the differential is taken to determine when the array goes from small variance to high variance. The location of this peak gives the number of pilot tones among the subcarriers, n_p. Calculating the location of n_p lowest values in the variance vector gives the location of the pilot tones, s_p.

Data: Signal $s_m[p] \in \mathbb{C}$;
Result: Vector s_p
for $i = 1 : M$ **do**
| $s_i[p] = s_m[q[i] : q[i+1] - 1]$
end
$s_i[p] \xrightarrow{\mathscr{F}} S_i[F]$
if $S_i[p] > 0$ **then**
| $S_i[p] = S_k[p]$
end
Var($S_k[p]$) over M symbols
$n_p = max(diff(sort(Var(S_k[p]))))$ over M
$s_p = min(Var(S_k[p]))$

Algorithm 4: Determining the location of the pilot tones

3.5.3 Simulation Results

The steps from algorithm 4 are tested on the simulated signal. Each step is presented in figure 3.5. From the results it is clear $n_p = 2$ and the locations are $s_p = [7, 18]$.

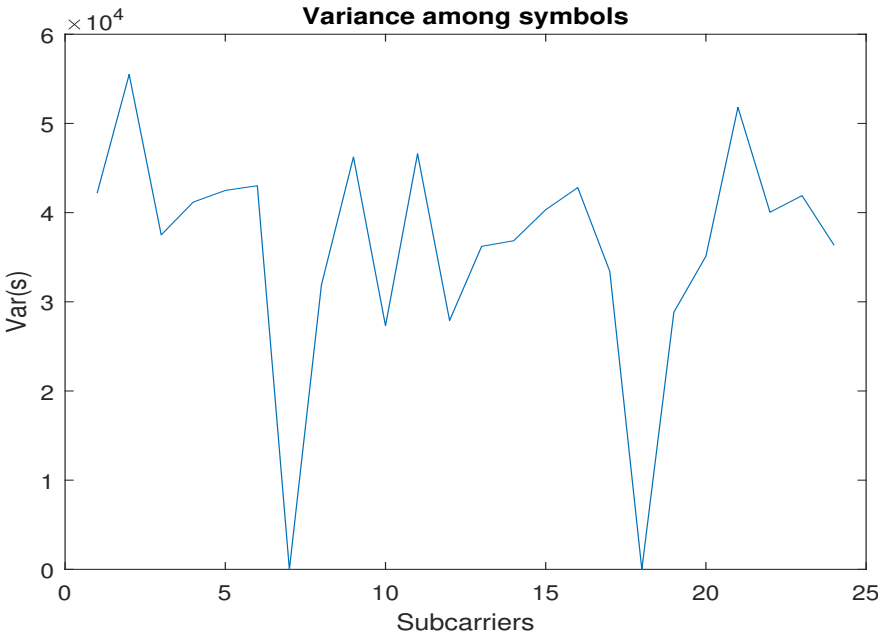


Figure 3.5: Determining the location of the pilot tones

3.6 Subcarrier Modulation

With the information obtained so far, it is possible to extract the data carriers from each OFDM symbol. This information is not the goal of this thesis, but is used to check the quality of the previous steps.

3.6.1 Resulting Algorithm

The constellation is obtained using the following algorithm.

Data: Subcarrier data, complex;

Result: Constellation

Scatterplot(subcarrier data)

Algorithm 5: Obtaining the constellation

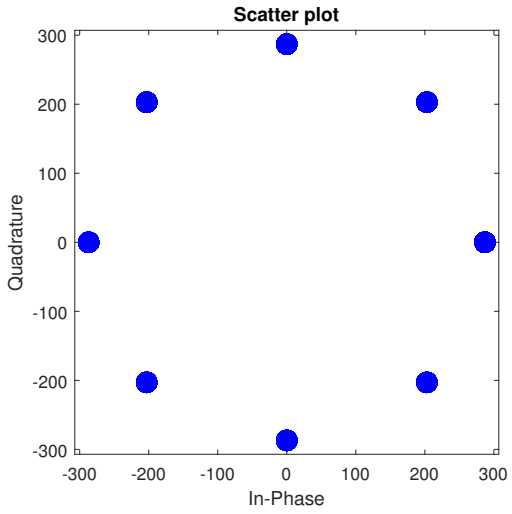


Figure 3.6: Constellation obtained form the subcarriers

3.6.2 Simulation Results

The constellation obtained using the method described above is shown in figure 3.6.

3.7 Edge Effects

Recording a signal which starts at exactly the first sample of a CP is unlikely. Edge effect can therefore occur where partial symbols are recorded at the start or at the end of a signal.

3.7.1 Non-complete Symbols

Calculating the number of symbols in the sample set of size L can be done with

$$M_{symbols} = \frac{L_{sampleset}}{L_{symbol}}. \tag{3.6}$$

but is not always correct. Equation 3.6 only works if the start of the signal is also the start of the first CP and the signal holds an integer number of symbols. Possible partial symbols and the start and end of the signal are not accounted for at this stage.

3.7.2 Signal Property

Based on the information obtained in section 3.2, the start of a symbol is known within the signal data. Using this as a reference, with the symbol length T_{SS} and the CP length T_{cp} obtained in section 3.3, it is possible to remove the partial symbols from the signal.

3.7.3 Method

The start of a symbol is located at $q[k]$. Calculating how many integer symbols, including the CP, can be extracted before $q[k]$ is done by using the modulus.

3.7.4 Resulting Algorithm

Data: Signal, $s[p] \in \mathbb{C}$, T_S and $q[k]$;
Result: Signal only containing complete symbols
$Rem_s = q[k]$ modulo T_{SS}
$Rem_e = (L - q[k])$ modulo T_{SS}
$S_{new} = s_m[Rem_s : Rem_e]$

Algorithm 6: Removing edge effects

3.7.5 Simulation Results

The steps from algorithm 6 are illustrated in figure 3.7.

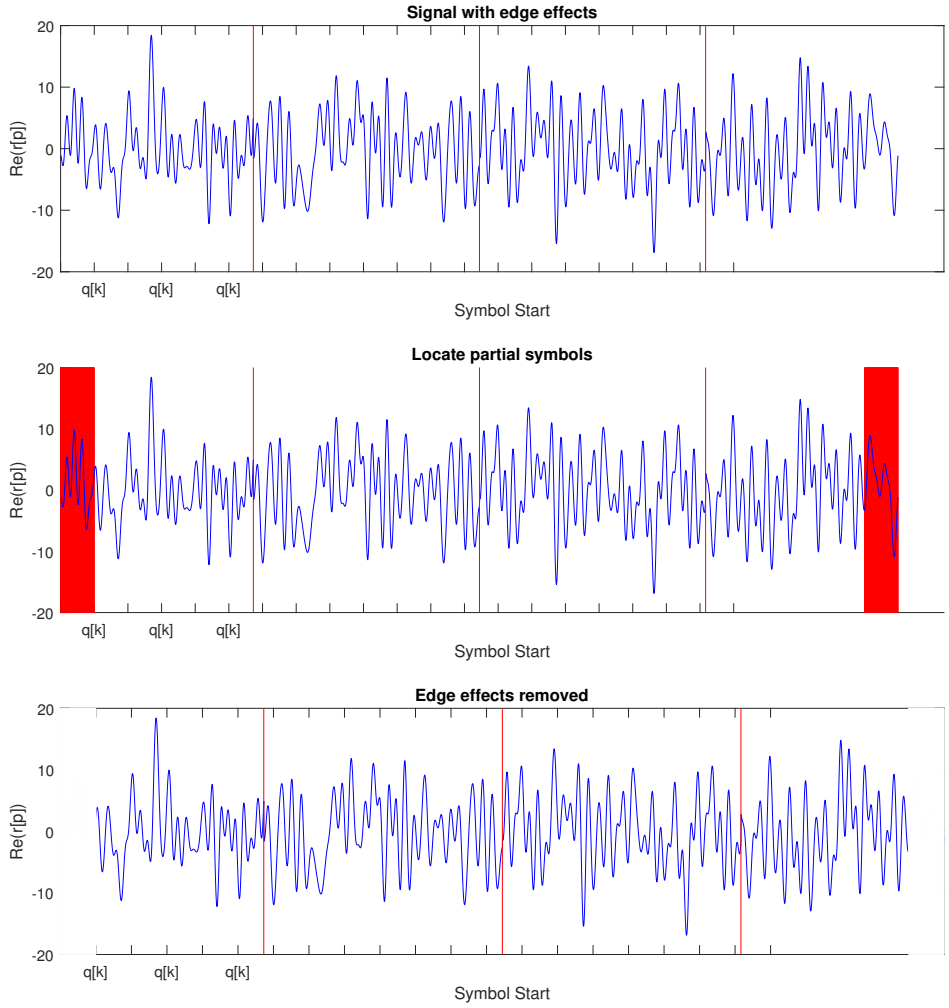


Figure 3.7: Removing edge effects

4

Practical Considerations

4.1 Symbol Boundaries

In chapter 3 a technique was introduced to detect OFDM signal boundaries within an OFDM waveform based on the discontinuities between symbols. This section will introduce bandlimmiting and windowing effects on an OFDM signal, which has a great impact on the transitions between OFDM symbols.

4.1.1 Bandlimiting and Windowing

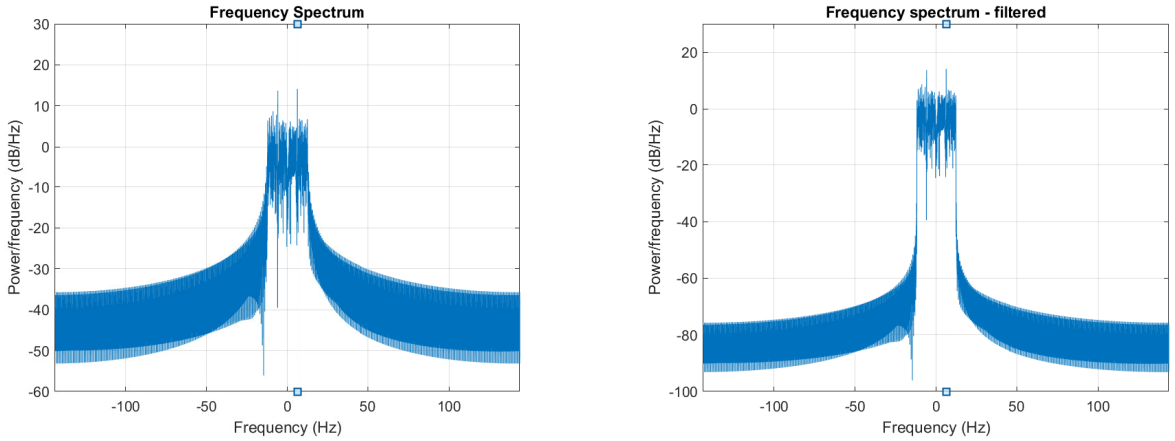
Figure 4.1a shows the frequency spectrum of the test signal from chapter 3. The standard implementation of OFDM has large side lobes which need to be reduced in order to conform to the FCC spectrum mask [36]. This side lobe reduction can be accomplished in multiple ways, including filtering and windowing. Figure 4.1b shows the same signal but with a Tukey window applied. Equation 2.31 can be abstracted to

$$s(t) = \sum_{i=0}^{M-1} s_i \delta(t - iT_S) \tag{4.1}$$

and s_i is the i-th OFDM symbol, consisting of the data and CP. The use of the Dirac comb in combination with the random input information, creates a likely discontinuity between symbols[1]. Bandwidth limiting smooths the transition between symbols, thereby decreasing or possibly removing the discontinuities. The introduction of filters alters equation 4.1 to

$$s(t) = \sum_{i=0}^{M-1} s_i * H(t - iT_S) \tag{4.2}$$

[1]It is important to note that this is not always the case: when using a lower order modulation technique on the subcarriers, in combination with a low number of subcarriers, the variation in possible starting points decreases.



(a) (b)

Figure 4.1: OFDM signal frequency spectrum unfiltered (a) and filtered (b)

where $H(t - iT_S)$ is the filter transfer function. An example of a common filter used in combination with OFDM is the Tukey window, given by

$$h[p] = \begin{cases} \frac{1}{2}\left[1 + cos\left(\pi\left(\frac{2p}{\alpha N} - 1\right)\right)\right], & 0 \leq p < \frac{\alpha N}{2} \\ 1, & \frac{\alpha N}{2} \leq p \leq N(1 - \frac{\alpha}{2}) \\ \frac{1}{2}\left[1 + cos\left(\pi\left(\frac{2p}{\alpha N} - \frac{2}{\alpha} + 1\right)\right)\right], & N(1 - \frac{\alpha}{2}) < p \leq N \end{cases} \tag{4.3}$$

where α determines the slope of the window. For $\alpha = 0$ the window is rectangular because the conditions for equation 4.3 only validate the case $H[p] = 1$. At $\alpha = 1$ it becomes a Hann window since the centre condition is from $\frac{N}{2}$ to $\frac{N}{2}$. The Tukey window has been selected because it doesn't influence the subcarrier symbol data whilst repressing the side lobe levels as shown in figure 4.2 [37, 38]. The implementation of a filter results in smooth transitions between symbols. An implementation for the Tukey window has been provided, and can be found in appendix B.3. Since windowing itself is not part of the research goal, only the effects will be discussed.

In section 3.2 a technique was presented to locate the signal boundaries. Applying this technique to the filtered OFDM signal does not present the desired results. Part of the test signal, the filtered version and both their differentials are shown in figure 4.3. From this figure is it clear that discontinuity detection using the differential no longer obtains the symbol boundaries. A new approach is needed to obtain the OFDM signal parameters T_S, T_{SS} and T_{cp}.

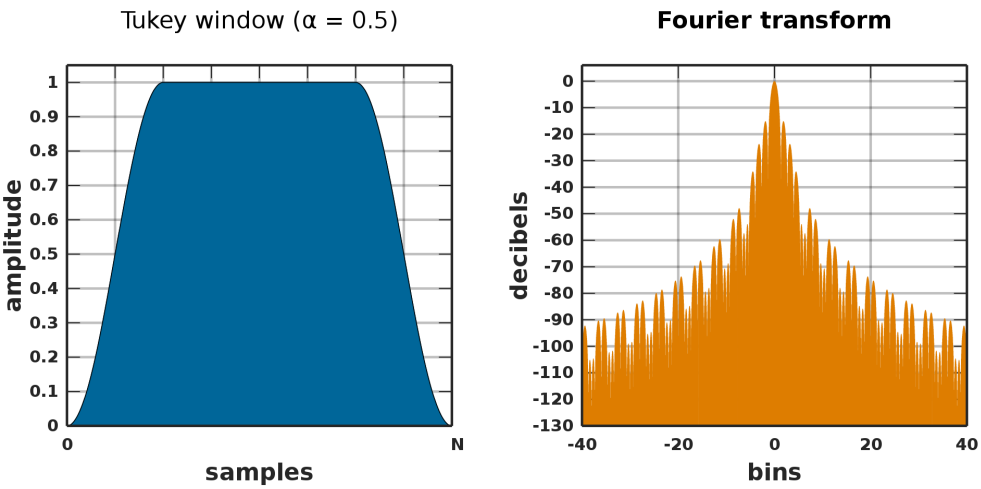


Figure 4.2: Tukey window function and its Fourier transform, $\alpha = 0.5$

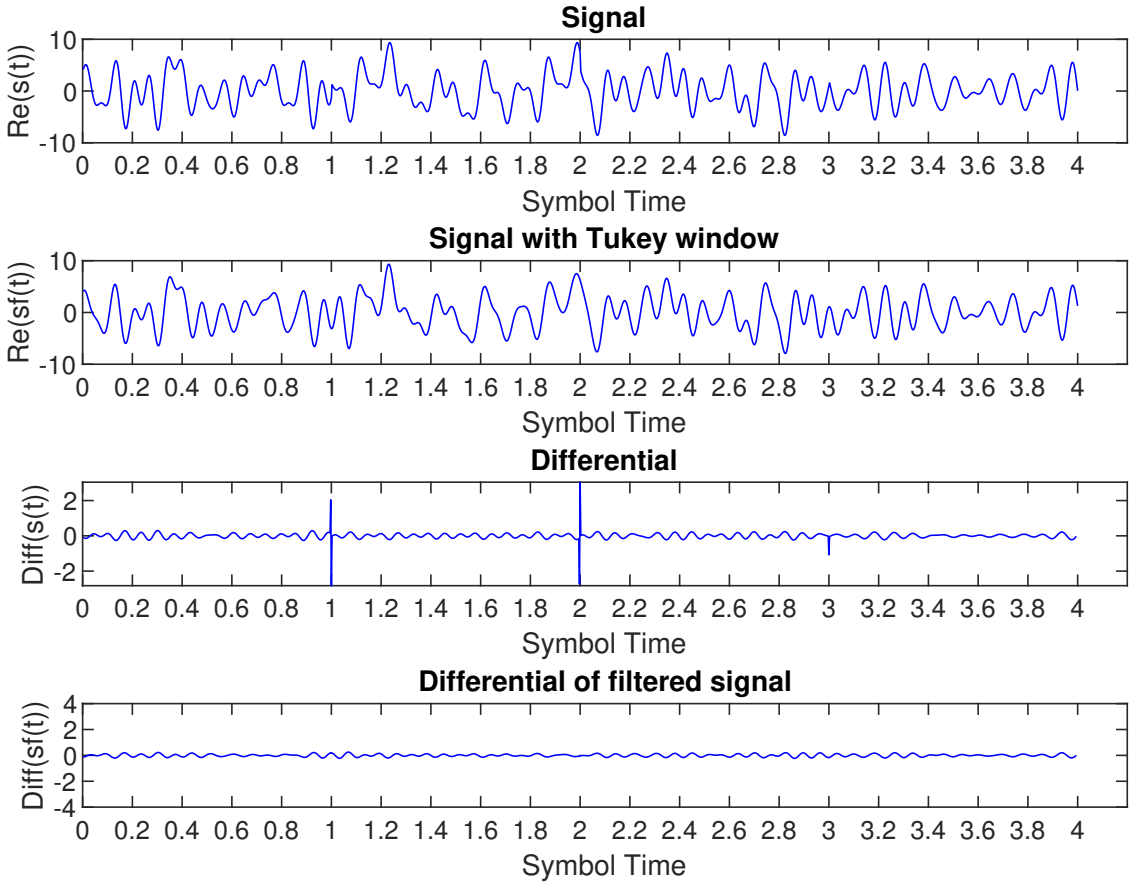


Figure 4.3: Locating signal boundaries comparison between filtered and unfiltered OFDM signals

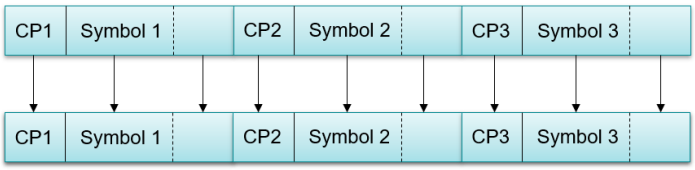


Figure 4.5: Autocorrelation energy peak and side peaks illustrated

4.1.2 Signal Property

The signal property which will be exploited in this section is the CP and its characteristics. Since the CP is a copy from the end of a OFDM symbol, there is a strong correlation between the samples of the CP and the samples at the end of the OFDM symbol. This correlation can be calculated using the autocorrelation function.

4.1.3 Method

The autocorrelation function is given by

$$R_s[\tau] = E[s(t)s^*(t+\tau)] \tag{4.4}$$

and the result is shown in figure 4.4. The main peak from the figure is the energy peak, located at $\tau = 0$. The side peaks are the result of the correlation between the CP and the original part at the end of the OFDM symbol as illustrated in figure 4.6. The shift needed in τ to obtain these peaks are $\tau = T_{SS}$ and $\tau = -T_{SS}$.

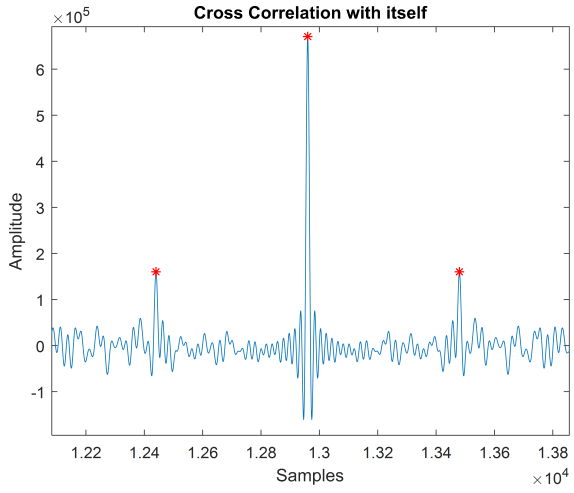


Figure 4.4: Autocorrelation of an OFDM signal

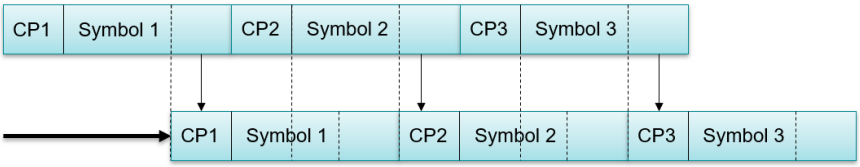


Figure 4.6: Autocorrelation energy peak and side peaks illustrated

4.1.4 Resulting Algorithm

The theory from the previous section results in the following algorithm.

Data: Input signal $s(t)$
Result: T_{SS}
for $\tau = -L : L$ **do**
| $R_s[\tau] = E[s(t)s^*(t+\tau)]$
end
T =findpeaks(R_s)
$T_{SS} = |T[1] - T[2]|$

Algorithm 7: Determining the length of a symbol

4.1.5 Simulation Results

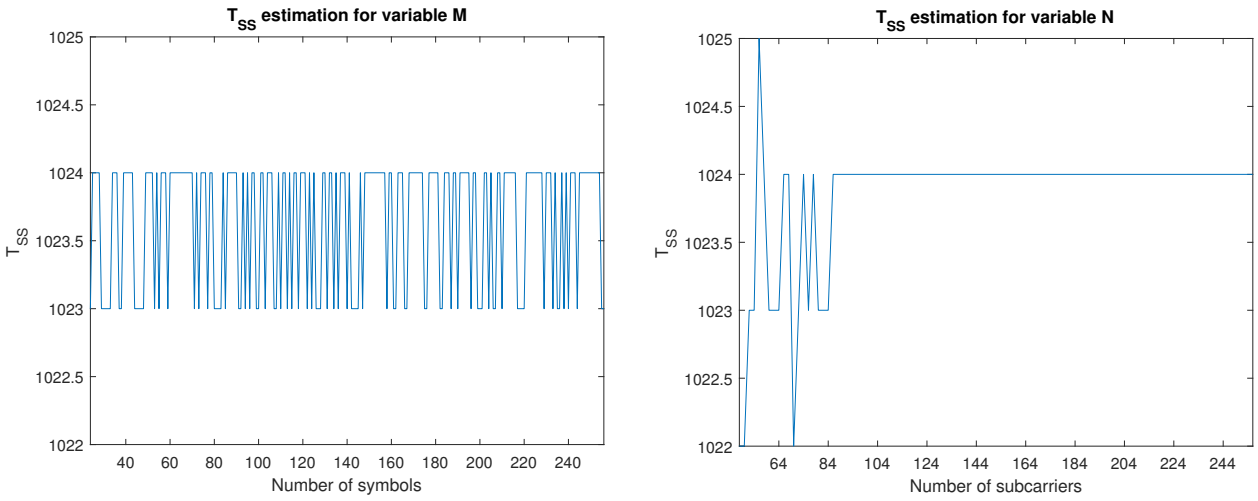
To verify the technique described above, test signals have been generated using a variable number of subcarriers and symbols. Figure 4.7a shows the calculation of T_{SS} for 20 to 256 symbols. Figure 4.7b shows the result for 24 to 256 subcarriers. Only the even numbers are used due to the symmetry needed for OFDM. The symbol length T_{SS} has been set to 1024 samples and based on the results, an error margin of $\epsilon \leq |2|$ samples is defined.

4.2 Cyclic Prefix Length

The technique described in section 3.3 is dependent on T_S being obtained. With T_{SS} known but T_S unknown, a new approach is needed to obtain either T_S or T_{cp}.

4.2.1 Signal Property

The theory behind the signal properties described in section 3.3.1 still holds, but is not as exact due to the effect of filtering. The correlation between the CP and the end of the OFDM symbol is still larger than the correlation between two OFDM symbols, as shown in section 4.1.



(a) (b)

Figure 4.7: T_{SS} calculations for variable subcarriers and symbols

4.2.2 Method

Since T_{SS} has been obtained in section 4.1, it is known how many samples there are between the CP and the end of the OFDM symbol as shown in equation 2.31 which is defined from $-T_{cp} < t < T_{SS}$. To determine the CP length, a window if increasing size is correlated with a window of the same size at a distance T_{SS} apart. For

$$R_s[k] = E[s(p : p + k)s^*(p + T_{SS} : p + k + T_{SS})] \tag{4.5}$$

p is increased for each symbol at a distance of T_{SS} in samples and $0 < k < T_{SS}$ to estimate the CP length.

The assumption made here is that the CP length will be greater than 0 and smaller than T_{SS}. This assumption is based on the CP length in existing techniques, presented in table 4.1.

Standard	CP Length as fraction of T_{SS}
DAB	$\frac{24.6}{100}$
DVB	$\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$
DMB	$\frac{1}{4}, \frac{1}{6}, \frac{1}{9}$
802.11a	$\frac{1}{4}$

Table 4.1: Examples of CP length in standards

4.2.3 Resulting Algorithm

The implementation from section 4.2.2 is summarised in algorithm 8.

Data: Input signal $s[p]$, T_{SS}
Result: T_{cp},T_S
for $m = 1 : M$ **do**
  $p = m * (T_{SS} + k)$ **for** $k = 1 : T_{SS}$ **do**
    $R_s[k] = E[s(p : p + k)s^*[p + T_{SS} : p + k + T_{SS}]$
  end
end
$Q = diff(R_s[k])$
$T_{cp} = argmin(Q)$
$T_S = T_{cp} + T_{SS}$

Algorithm 8: Determining the length of the cyclic prefix

4.2.4 Simulation Results

The result is a correlation with increasing window size, as shown in figure 4.8. k is increased and the correlation between the windows at a distance T_{SS} is calculated, as shown in figure 4.9. This process is repeated for an increasing window as shown in the second figure of 4.8. Once the correlation suddenly drops, the first sample of the next CP and the first sample of the symbol are selected, which are likely uncorrelated. This shows clearly in the differential from figure 4.9 and is represented by the red window from the bottom figure from 4.8.

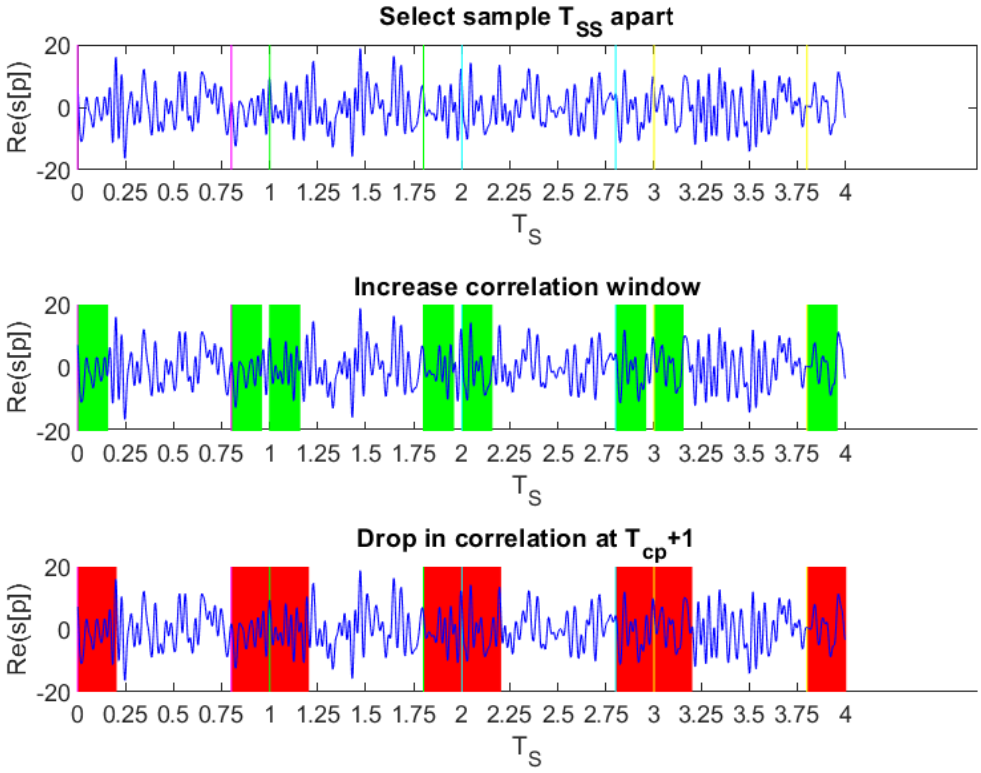


Figure 4.8: Correlation of samples T_{SS} apart until $T_{cp} + 1$

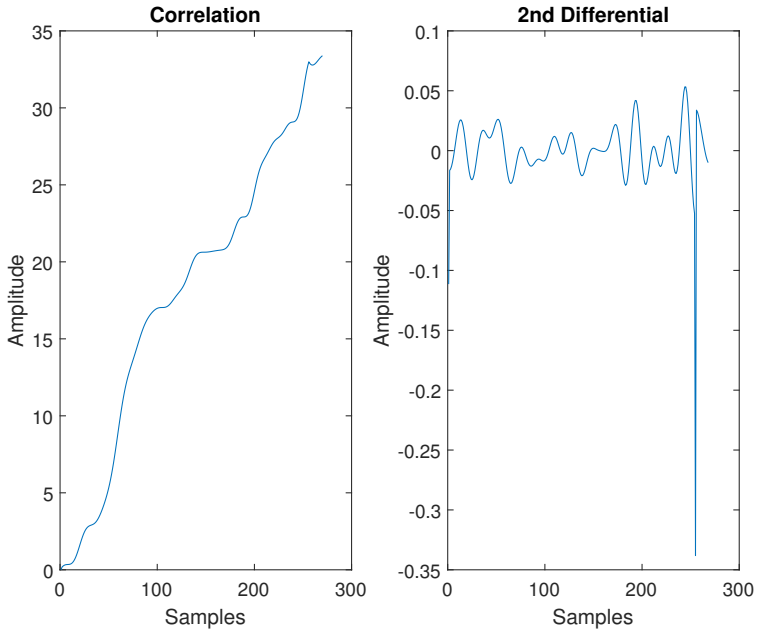


Figure 4.9: Correlation of samples T_{SS} apart and the resulting 2nd order differential

4.3 Sample Rate Alignment

Frequency binning is an important aspect of OFDM, there should always be an integer set of samples per symbol which is at least equal to the number of subcarriers. Non-integer samples per symbol can lead to spectral leakage, resulting in incorrect subcarrier separation and data retrieval. OFDM has become such a popular technique due to the technical advances in digital signal processing. One of the main components, the Fourier transform, can be implemented at a much higher speed due to the use of the FFT [3]. The FFT uses 2^Q samples, greatly decreases complexity compared to other values. The use of a power of 2 has a complexity of $O(Nlog(N))$ where other values have a complexity of $O(N^2)$. As explained in section 2.2.2, OFDM relies on the use the FFT and therefore OFDM waveforms contain 2^Q number of subcarriers, either data carriers or zero padded to best fit the FFT. To achieve orthogonality, as explained in section 2.2.1, the frequency bin spacing should be $k\Delta f$ for $k \in \mathbb{Z}$.

4.3.1 Signal Property

A symbol has been taken from an oversampled signal. The time domain signal, frequency domain signal and resulting constellation have been plotted in figure 4.10. Increasing the sampling rate by a factor of 2 smooths the time domain signal and add zeros to the outer frequency bins. The number of subcarriers stays the same and, besides amplitude changes, the constellation also remains the same.

4.3.2 Method

The presented solution for this is to re-sample the signal based on the number of samples in the calculated T_{SS} from the method described in section 4.1. The number of samples in T_{SS} are denoted by N_{SS} and the signal is re-sampled to the closest power of 2 above the current number of samples per symbol. The re-sample factor rsf is calculated using

$$rsf = \frac{2^{\lceil log_2(N_{SS}) \rceil}}{N_{SS}} \tag{4.6}$$

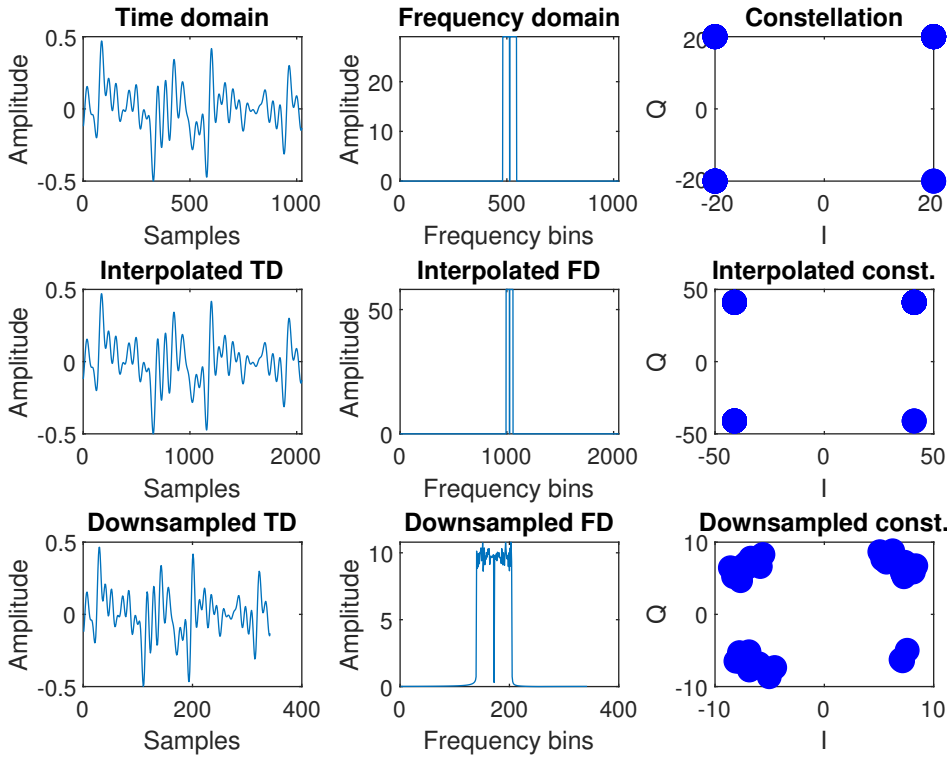


Figure 4.10: Effects of interpolation by 2 and downsampling by 3

4.3.3 Resulting Algorithm

The implementation is summarised in algorithm 9.

Data: Input signal $s(t)$, T_{SS}
Result: $r[p]$,T_{SS}
while $log_2(N_{SS}) \neq |log_2(N_{SS})|$ **do**
  $rsf = \frac{2^{\lceil log_2(N_{SS}) \rceil}}{N_{SS}}$
end
$r[p] = resample(r[p], rsf)$
Calculate T_{SS} following algorithm 7 and verify

Algorithm 9: Sample rate alignment

4.3.4 Simulation Results

The result is shown in figure 4.11, the noise effects on the frequency and constellation plots are successfully removed. One very important notation to be made is that the total number of samples should add up in the complete data set, because measured data will always have an integer number of samples. This means that, in this example, if the entire data set should consist of 2 symbols, totalling to $341\frac{1}{3} * 2 = 682\frac{2}{3}$ samples, this is of course not possible. The actual data set will consist of 682 or 683 samples, and re-sampling to the closest power of two will still result in errors due to

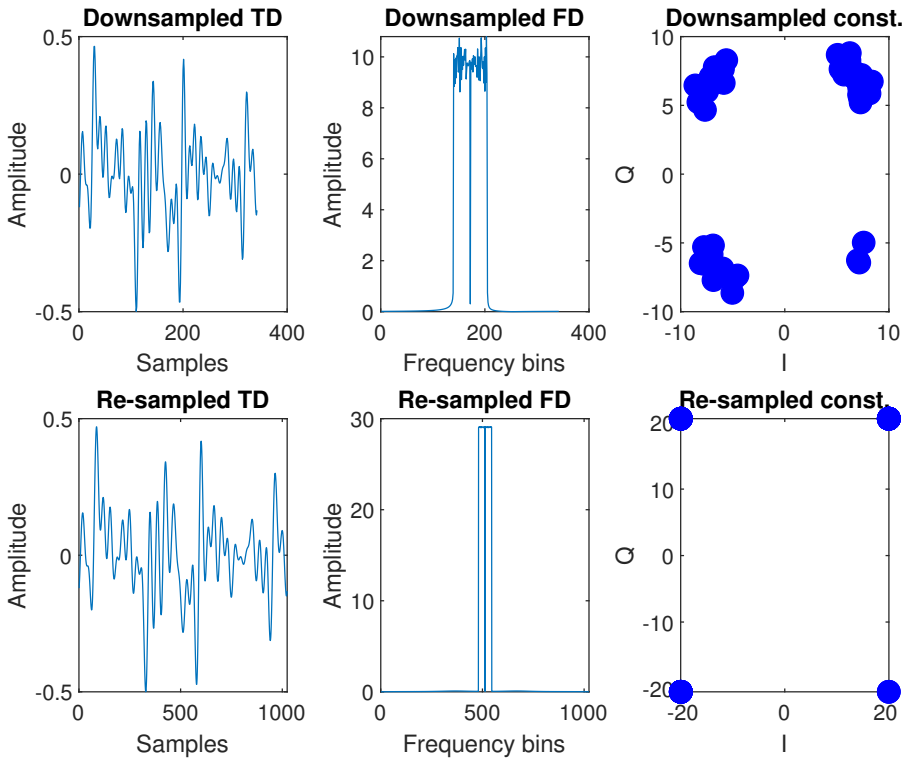


Figure 4.11: Effects of re-sampling to the closest power of 2

the incomplete data set.

5

Results

This thesis presents two techniques to blindly obtain OFDM signal parameters. In this chapter, both techniques are applied on numerous test signals, noiseless and with filtering and added noise. If discontinuities are obtainable from an OFDM signal, the technique presented in chapter 3 is preferred. A more practically viable solution is the technique presented in chapter 3 but it does introduce an error margin as discussed in section 4.1.5.

5.1 Noiseless and Unfiltered

To test the capabilities of the presented discontinuity and autocorrelation based techniques, six OFDM signals have been generated by a third party. Besides the complex data stream and the fact that they hold OFDM signals, no further information was provided. This section will elaborate on the outcome of both techniques in analysing these signals and obtaining the desired parameters. The results for the discontinuity based technique are shown in figure 5.1. The same signals are processed using the autocorrelation based technique described in chapter 4. These results are shown in figure 5.2.

From the results, it is clear the discontinuity based technique performs better compared to the autocorrelation based technique when the signals are noiseless and without filtering. All parameters estimated from the signals are presented in figure 5.1 and are without error. The autocorrelation based technique shows slight deviations in determining the CP length as shown in figure 5.2a, where it is one sample off. The next figures, 5.2b, 5.2c and 5.2d, show large errors. Figures 5.2e and 5.2f show the last two signals and their estimated parameters without error.

The main distinction here is the difference in the extremes. Where the discontinuity based technique could handle 0% and 3100% CP length as shown in figures 5.1c and 5.1d, the autocorrelation based technique is dependent on having a CP and did not yield the correct results as shown in figures 5.2c and 5.2d. In practical implementations, a form of CP will be included to handle multipath effects. For this reason, there will not be more focus on dealing with an OFDM signal with no CP.

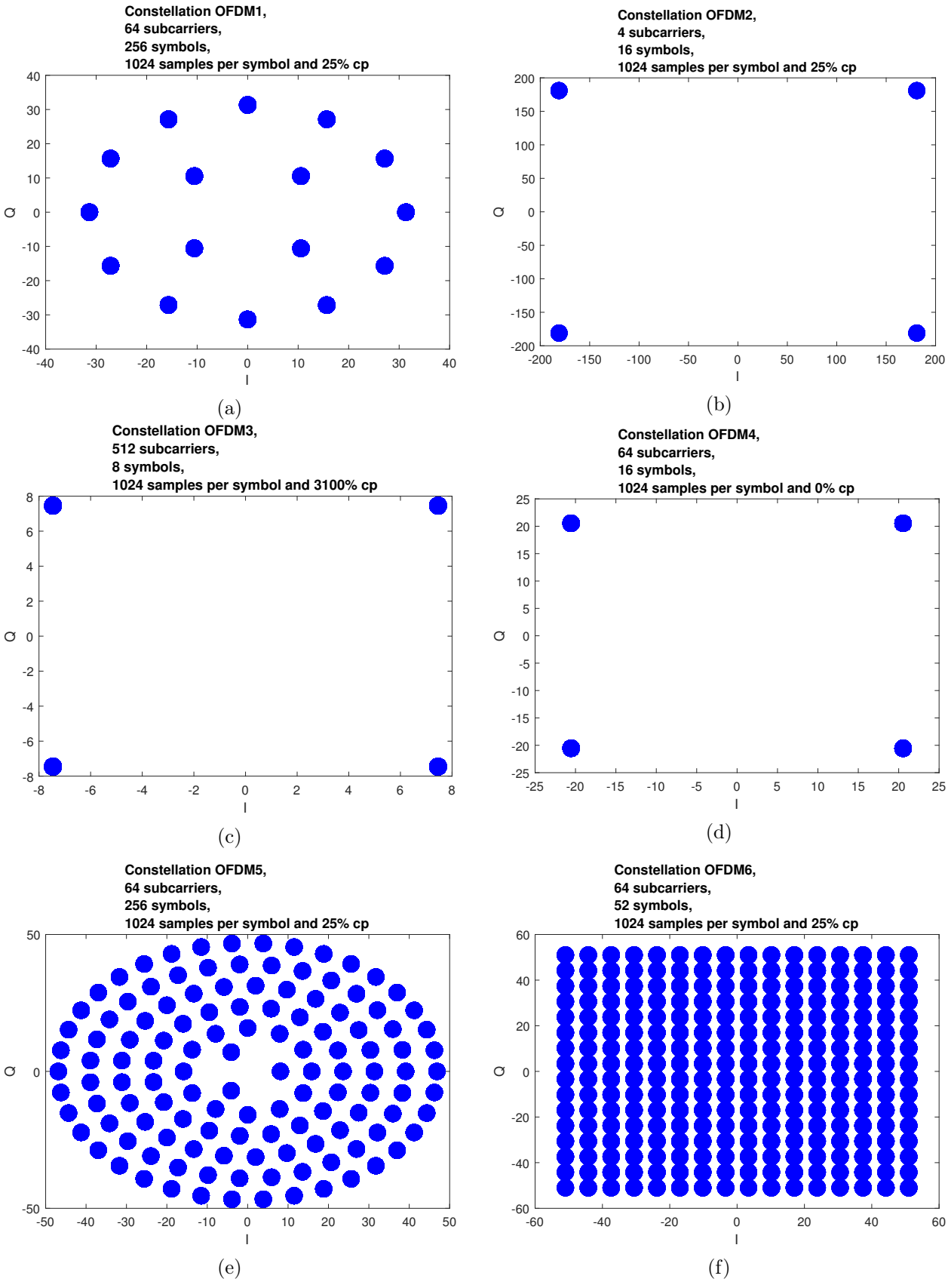


Figure 5.1: Results of the discontinuity based technique

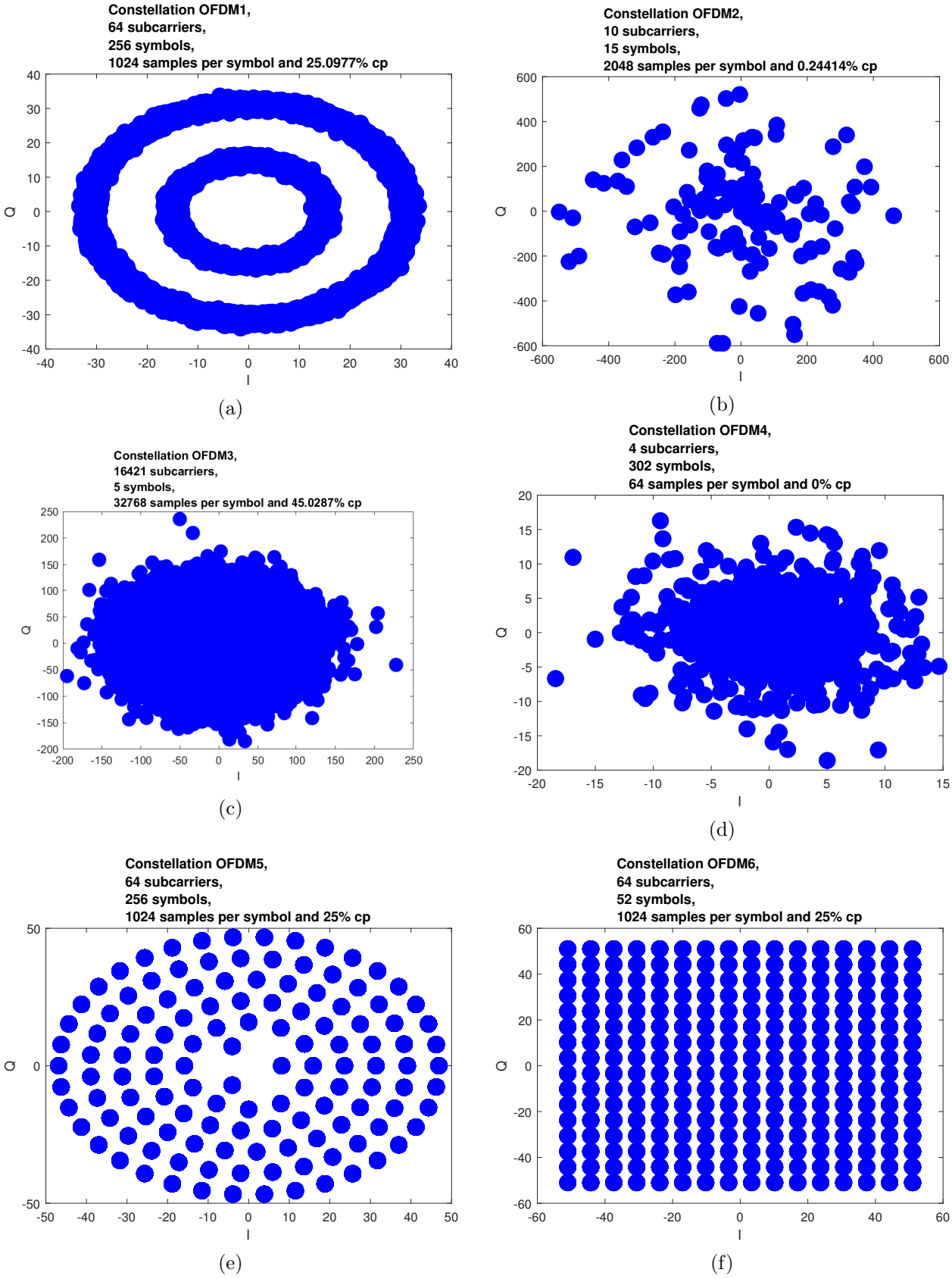


Figure 5.2: Results of the autocorrelation based technique

5.2 Filtered & Noise Tolerance

Test signals have been generated using the OFDM signal generator described in Appendix A and filtered using the Tukey window as described in section 4.1. On top of the filtering, AWGN noise has been added decreasing the SNR.

The tests described in this section have been performed using the following parameters:

- Random input data;
- Subcarrier modulation: QPSK;
- Subcarrier count: $N = 64$, $N = 128$;
- Relative Cyclic Prefix Length: $T_{cp} = 25\%$, $T_{cp} = 12.5\%$, $T_{cp} = 6.25\%$;
- Noise has beend added decreasing the SNR from 30dB to 0 dB in steps of 5dB;
- Symbol count: $M = 64$;

These parameters have been selected based on real world examples such as WiFi, except the symbol count. The symbol count has been set to 64 to strike a balance between simulation time and complexity. Each parameter set has been tested 50 times.

5.2.1 Presented Results

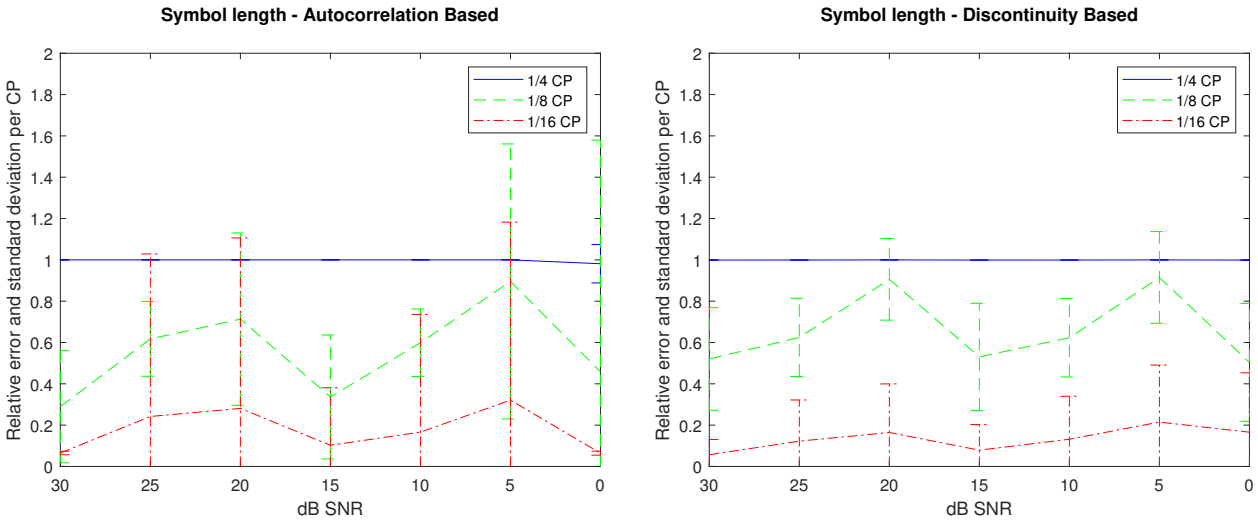
The results presented in this section are derived by dividing the obtained value for a parameter by the input value,

$$X = \frac{X'}{X_{true}} \tag{5.1}$$

This means the perfect result would be 1, and all results are greater then 0. The mean is obtained by averaging the calculated X over the 50 measurements. The standard deviation is calculated over the 50 measurements and displayed in the figures by using vertical lines.

5.2.2 Symbol length

The symbol length has been determined for the different levels of SNR, averaged over the different subcarrier values N and plotted for the different CP lengths. The result for the autocorrelation based technique is presented in figure 5.3a and the discontinuity based result is presented in figure 5.3b. As clearly visible from the figures, both techniques work very well with the higher relative CP length. For lower CP lengths, both approaches show a significant increase in incorrect results. The lowest CP length tested, $\frac{1}{16}$th performs very poor in both techniques.

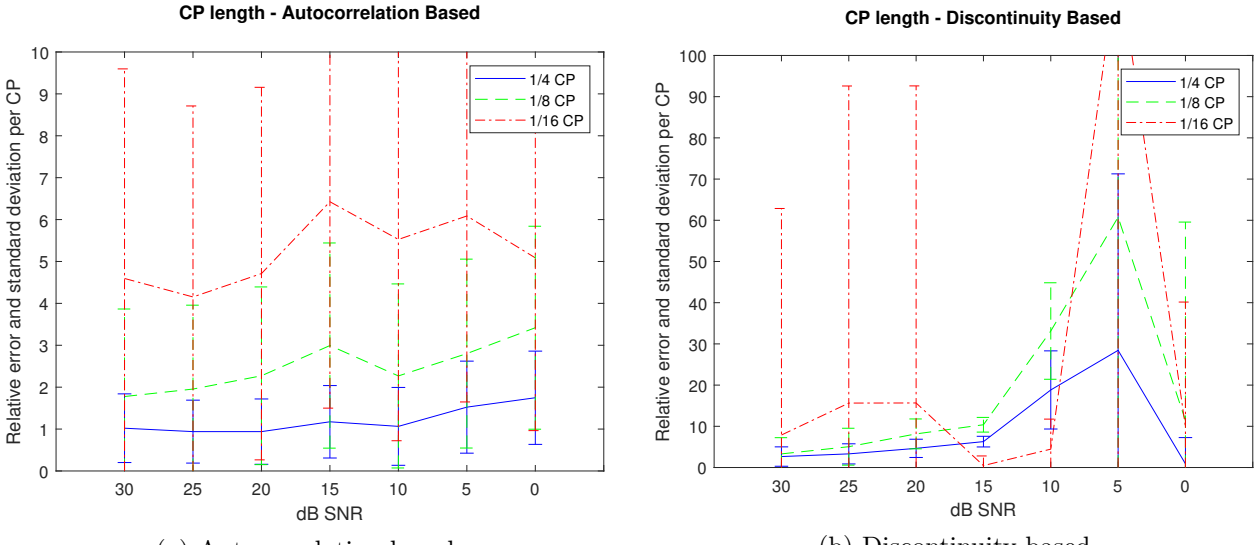


(a) Autocorrelation based (b) Discontinuity based

Figure 5.3: Determining T_{SS} for different levels of SNR

5.2.3 CP Length

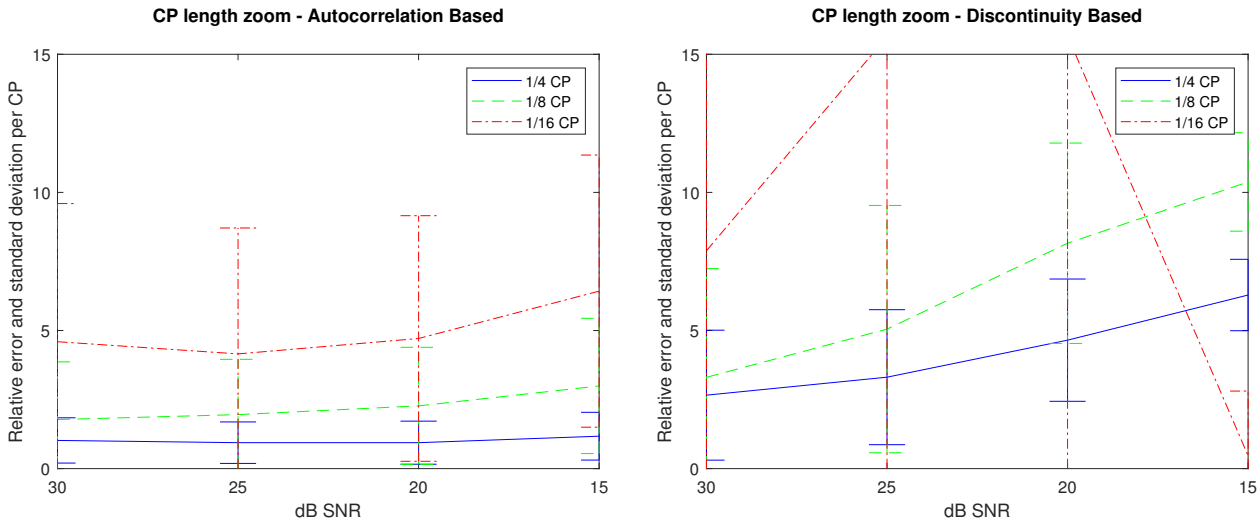
The CP length simulations have been performed by the same approach as described in section 5.2.2. The autocorrelation based technique, as illustrated in figure 5.4a, shows positive results for the longest CP length, $\frac{1}{4}$th of the symbol length. Until the 10dB SNR mark, the results are very close to ideal. Below 10dB SNR the error margin and standard deviation increases and results are less reliable. The smaller CP lengths score poorly among all levels of SNR. The discontinuity based technique, presented in figure 5.4b, shows very poor results over all CP lengths and SNR values.



(a) Autocorrelation based (b) Discontinuity based

Figure 5.4: Determining relative T_{cp} for different levels of SNR

Since figure 5.4b shows very large variance for the lower SNR levels it is hard to compare the presented figures. Figure 5.5 compares the two techniques by zooming in on the first few levels of SNR, from 30dB to 10dB SNR. Here is is clearly visible that, even in the higher SNR levels, the autocorrelation based technique outperforms the discontinuity based approach.



(a) Autocorrelation based

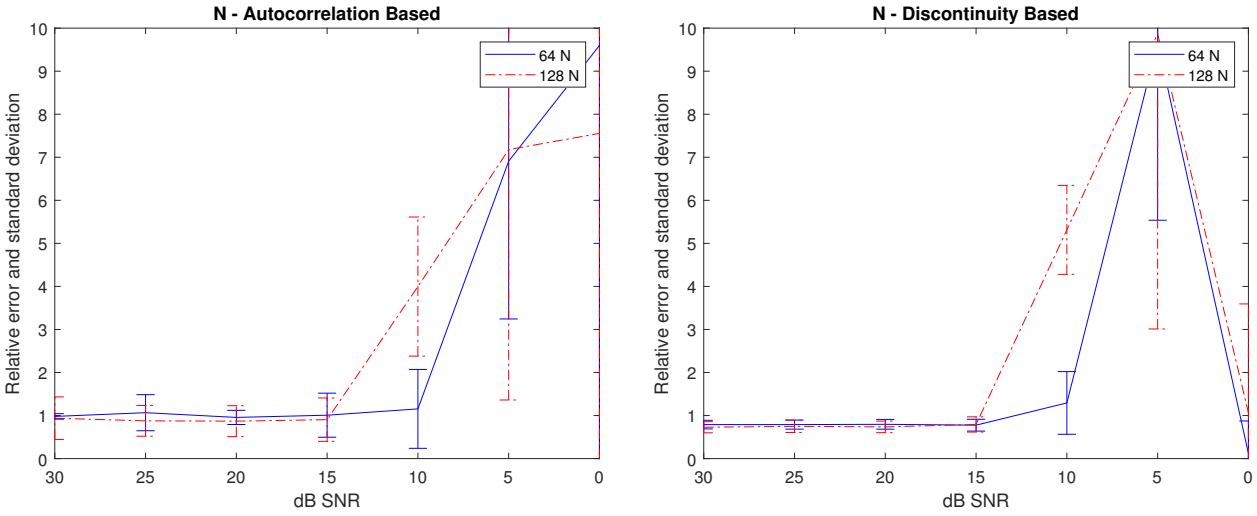
(b) Discontinuity based

Figure 5.5: Zoomed in CP estimation, first few SNR levels

5.2.4 Number of Subcarrier

Due to the poor performance of the smaller CP lengths, the CP length has been set to $\frac{1}{4}$ of the symbol length for the results presented in this section. Including the smaller CP lengths results in more errors in estimating the number of subcarriers due to the dependencies between the parameters.

Figure 5.6a shows the result for the subcarrier estimation for the autocorrelation based technique. The number of errors are relatively low until an SNR level of 15dB has been reached. Under the higher noise levels it is not possible to correctly obtain the number of subcarriers. The results of the discontinuity based approach, as presented in figure 5.6b, perform similar up to the 15dB SNR mark, with with a much smaller deviation. With higher noise levels, the number of errors increases exponentially to the point of failure, as shown by the sudden drop at the 0dB SNR level.



(a) Autocorrelation based

(b) Discontinuity based

Figure 5.6: Determining the number of subcarriers for varying SNR

5.3 Limitations

The results presented in this chapter clearly present the limitation of both techniques, the CP length. Both techniques heavily relay on the correlation between the CP and the identical part at the end of the OFDM symbol. Relative longer CP lengths still produce results, despite the filtering and levels of noise, because there are enough correlated samples between the CP and the end of the symbol.

6

Conclusion & Recommendations

6.1 Summary

This thesis set out to find a method for blindly estimating OFDM signal parameters. The main parameters which drive OFDM demodulators are: symbol length in number of samples, CP length in number of samples, the number of subcarriers and information about pilot tones. To accomplish this goal, an in-depth study of OFDM was performed which is presented in chapter 2, and a theoretical solution was devised which is presented in chapter 3. This technique shows how OFDM parameters can be blindly obtained and the research goal is achieved. Due to introduced practical challenges such as noise and filtering, alterations to this technique were needed. The result is a new autocorrelation based technique, as described in chapter 4.

6.2 Theoretical Solution

The main difference between OFDM and single carrier modulation schemes is the parallelisation of data. Where single carrier modulation schemes produce a relatively smooth signal, OFDM produces a series of symbols resulting in block waveforms where discontinuities can occur between blocks. This inherent feature can be exploited, which is what is done in the first approach. Each symbol has an arbitrary staring point, which means consecutive symbols do not necessarily line up. The step size between consecutive samples becomes clear. The large spikes clearly show discontinuities in the OFDM signal, caused by the steep step size between symbols. Using this information, symbol time and CP time parameters can be obtained. Applying an FFT to the obtained symbol, with some limitations regarding samples size etc., results in the subcarrier symbols. Finally, calculating the variance among the subcarriers over several symbols result in the location and value of the pilot tones.

6.3 Practical Considerations

The discontinuity based technique shows promising simulation results. However, due to filtering, windowing and other real world effects, practical alterations were needed. The result is a new, autocorrelation based approach. Calculating the correlation function on an OFDM signal with itself, shows clear spikes on each side of the energy peak. These peaks correspond the to overlapping of the CP with its image at the end of the symbol. The shift needed to accomplish this overlapping, is exactly the symbol time T_{SS}. Next, the correlation function is taken with a variable number of samples, but only observed at a distance T_{SS} as described in equation 4.4. During the CP, the amplitude increased due to the correlated samples. After the CP, the correlation between the sample of a symbol and a sample of the next CP, which are likely uncorrelated. By using multiple symbols, this decrease in correlation can be calculated. The length taken is therefore the length of the CP, T_{cp}. Further parameters are obtained in the same way as in the theoretical solution.

6.4 Conclusion

The discontinuity based technique performs very well on noiseless, pure OFDM waveforms as presented in chapter 5. The autocorrelation based technique performs well on most, but has difficulties with the simulated extremes like 0% or 3100% CP. This is an expected result, as the autocorrelation based technique heavily relies on the CP and its image. Regarding the filtered and noisy signals, the autocorrelation based technique performs better, but both perform poor when the relative CP length $\frac{1}{8}$ or smaller.

If discontinuities can be identified from a signal, the discontinuity based approach shows less error margins which makes it the preferred choice in the ideal case. The ideal case is, however, very unlikely. The conclusion drawn from the results is therefore that the best approach is to use both techniques in collaboration. Based on the measured SNR level or quality of the recorded signal, weights can be distributed between the results of both techniques. If discontinuities are clearly measurable or higher levels of SNR are obtained, the outcome of the discontinuity based approach can gain more weight in the final decision where lower levels of SNR put more weight on the autocorrelation based approach.

6.5 Limitations

Both techniques use the CP and its properties. Extreme cases have been tackled using the discontinuity approach, but the autocorrelation based technique is not able to handle either to large or to small CP lengths. Practical examples of OFDM using $\frac{1}{32}$ are being used but, if the FFT size is to

small, the autocorrelation based approach can not blindly obtain the signal parameters. This has been tested for an FFT size of 1024 and is shown in section 5.3.

6.6 Recommendations

The presented techniques are tested on simulated signals. Recorded signals are the next step in the research which is therefore the first recommendation for future work. Some clear challenges when working with real world recordings are discussed below and recommended to be tackled first.

Explore lower CP limit As shown in chapter 5, both presented solutions show a steep increase in error margin when the CP length is decreased. There are however standards which employ lower levels of CP like DVB.

Start of symbol So far, the start of the first CP has been at the start of the recording. Especially with live recordings, this will not always be the case. Multiple techniques exists, but have not been implemented together with the presented techniques due to timing limitations on the research. More research on this is needed to complete the presented techniques into a final toolbox.

Extend OFDM types The type of OFDM discussed in this thesis is the standard approach. It is used in multiple standards, but other techniques do exist. A few examples are:

- ZP-OFDM [33]
- OFDM without CP [34]

It is recommended to explore the use of these types of OFDM and, if deemed necessary, find a solution for blind parameter estimation of these OFDM types.

Appendices

A

Generating an OFDM signal

The first step is generating random data and assigning this data to the different subcarriers. Since 8-PSK is being used, the data is grouped per 3 binary values to form three bit symbols. For the centre subcarriers, f_{-2}, f_{-1}, f_1 and f_2, the random data generation appointed the following bit sequence: [111010100110]. Grouped per three bits, the result is [111], [010], [100] and [110] among the subcarriers respectively. Following a Gray coding scheme, this results in the waveforms for the real phasor rotations as shown in figure A.1. Summing these waveforms, and the other 20 subcarriers, results in the in-phase component of the OFDM symbol as shown in figure A.2. Mathematically, this step is described by

$$s_m(t) = \sum_{k=1}^{N} A_k cos(\omega_c t + \phi_k); \quad nT_S \leq t \leq (n+1)T_S \tag{A.1}$$

To verify the periodicity of the waveform, the same symbol has been plotted repeatedly in figure A.3. The plot clearly shows the two waveforms seamlessly connect. The next step is adding a CP to each symbol. This is illustrated for the symbol from figure A.2 in figure A.4. When the same waveform *with* CP is plotted repeatedly as shown in figure A.5, something inherent to OFDM is starting to show.

At the boundary between the two symbols, the waveform makes a jump. This is caused by the fact that the addition of the CP rotates the periodic properties of the OFDM waveform. The discontinuity that shows is illustrated in figure A.6. The same observation can be made if an OFDM waveform holds two different symbols, with or without CP. This is illustrated in figures A.7 and A.8 respectively. The next step is putting all the 20 symbols, including their CP, in series. The final waveform is shown in figure A.9 and contains 287 samples per symbol and 94 samples for each CP, which results in 7620 samples in total. The code for the OFDM waveform generator can be found in appendix B.4.

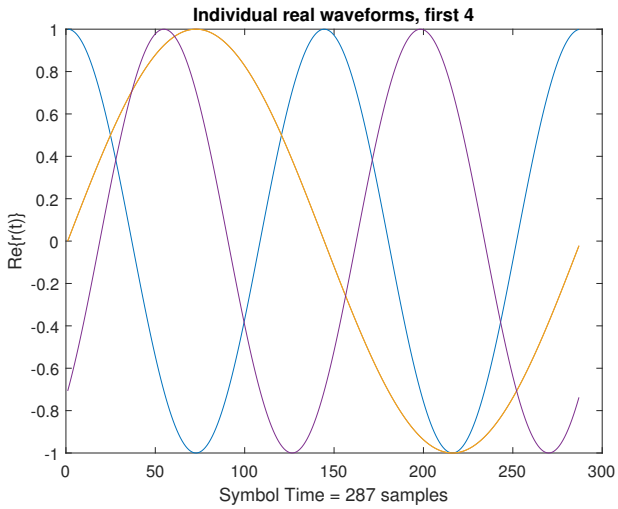


Figure A.1: Four centre individual real waveforms

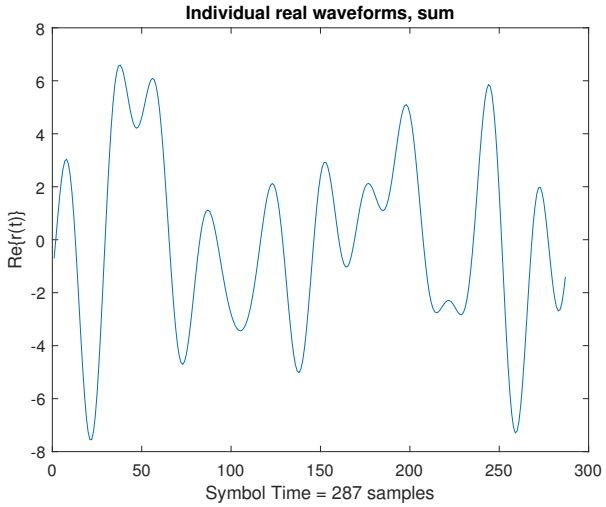


Figure A.2: In phase component of OFDM waveform of one symbol

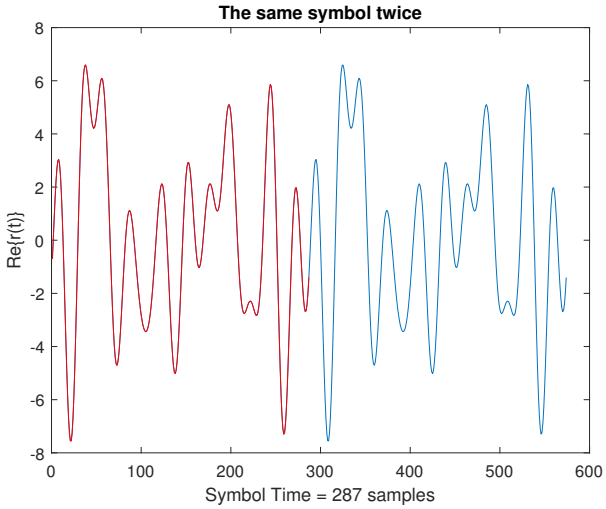


Figure A.3: Same OFDM symbol twice

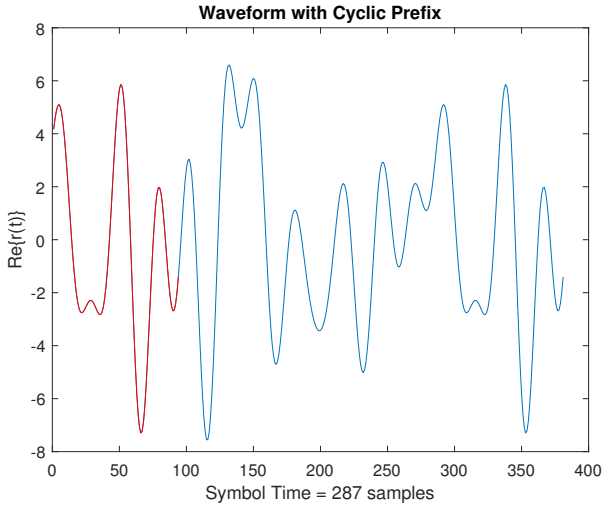


Figure A.4: OFDM waveform one symbol plus CP

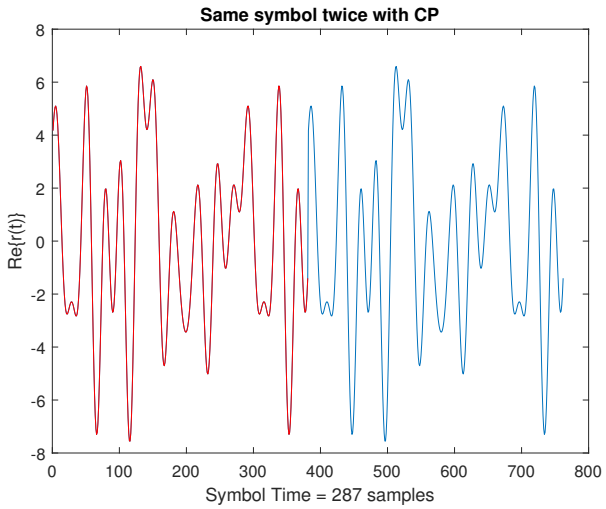


Figure A.5: Same OFDM waveform twice with CP

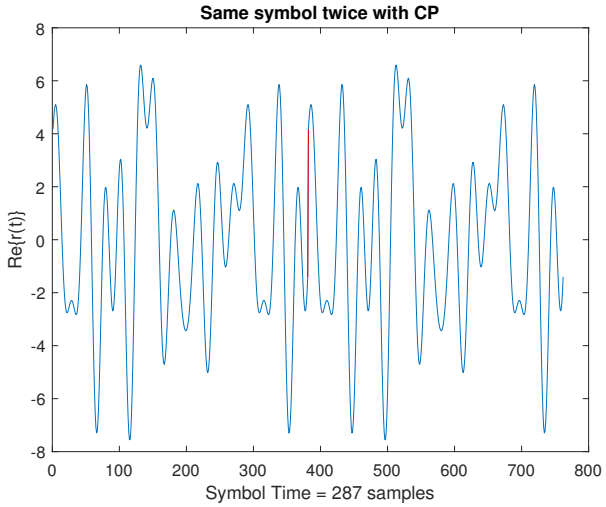


Figure A.6: Discontinuity between two copies of the same OFDM symbols with CP

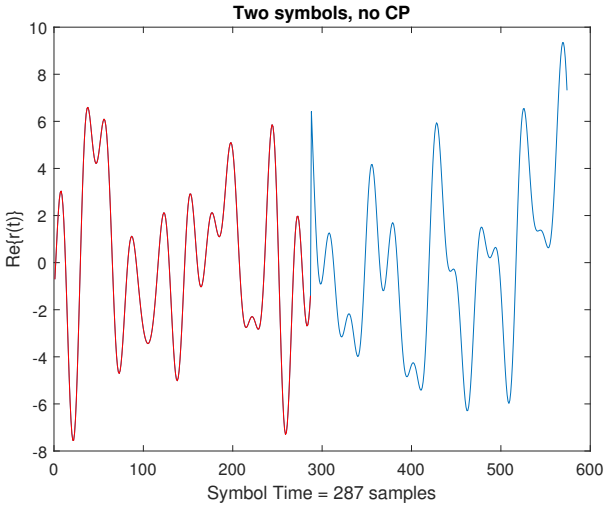


Figure A.7: Two OFDM symbols excluding CP

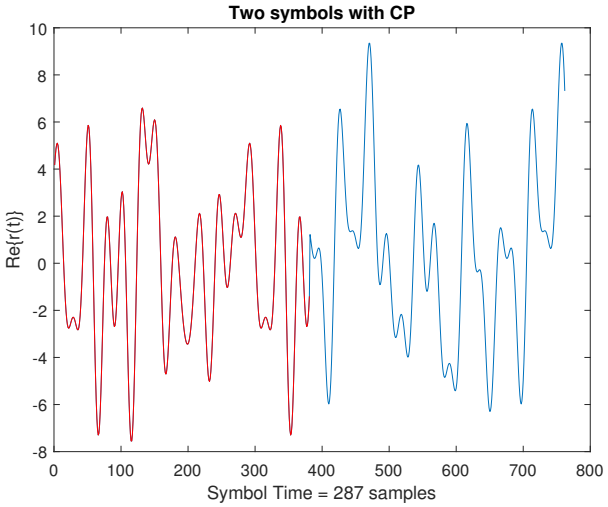


Figure A.8: Two OFDM symbols including CP

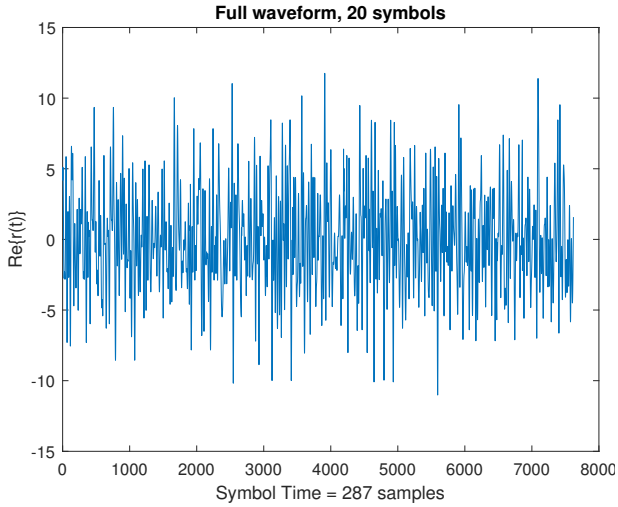


Figure A.9: OFDM waveform containing 20 symbols

B

Matlab Code

B.1 OFDM Parameter estimation - discontinuity based

```
function [constel,pilots, sub_N, symtime, cp, numberofsymbols] = OFDM_sep(sig)
    % Determine timings
    t=abs(xcorr(sig));
    tx=1:length(t);rt=1:length(t);
    TF = islocalmax(t,'MinSeparation',128);
    [a b]=maxk(t(TF),3);
    len = abs(tx(t==a(1))-min(tx(t==a(2))));
    %freq offset
    f_offset1 = angle(t(tx(t==a(1))));
    %fractional frequency offset
    f_frac_offset = angle(t(tx(t==a(2))));
    [y x] = max(abs(diff(abs(sig)))); % x is end of waveform
    r=1:length(sig);
    symtime = r(sig==sig(r==x));
    if length(symtime)>1
        symtime = symtime(2)-symtime(1); % what if no cp
    else
        y = maxk(abs(diff(sig)),floor(length(sig)/len));
        for i=1:length(y)
           x(i)=r(abs(diff(sig))==y(i));
           x = sort(x);
        end
        symtime=max(x);
        for i=1:(length(x)-1)
            if x(i+1)-x(i) < symtime && x(i+1)-x(i) > 10 && log2(x(i+1)-x(i))==round
(log2(x(i+1)-x(i)))
                symtime=x(i+1)-x(i);
            end
        end
        x=x(1);
    end
    %% Resamp to power of 2
    if log2(symtime)~=round(log2(symtime))
        sig = resample(sig,3,4);
```

```
    end
    tsamp=[];
    for val=2:len
        samp1 = sig(1,1:val);
        samp2 = sig(1,1025:1025+val);
        t=xcorr(samp1,samp2);
        tsamp(val)=abs((t(ceil(length(t)/2))));
    end
    [c cp] = min(diff(diff(tsamp(2:length(tsamp)))));
    if c>-0.02
        cp = 0
    else
        cp=cp+2
    end
    wave = symtime+cp;
    %% Separate subcarriers and remove cp
    numberofsymbols = fix(length(sig(1:x))/wave) +fix(length(sig(x+1:end))/wave);
    symbefore = floor(x/wave);
    start =1;
    sig(1:start-1)=[]; % remove partial symbol at start of signal
    data_demod = [];rx_perwave=[];constel=[];
    for i=1:numberofsymbols
        rx_perwave(i,:)=sig(1,(i-1)*wave+1:wave*i);
    end
    rx_perwave = flip(rx_perwave);
    rx_persymbol = rx_perwave(:,cp +1:end);
    clear i
    %% Determine pilot tones
    rxt_tot = [];
    for n_sym=1:numberofsymbols
        rxt=fftshift(fft(rx_persymbol(n_sym,:)));
        rxt = rxt(abs(rxt)>(max((abs(rxt)))*0.05));
        rxt_tot(n_sym,:) = rxt;
    end
    sub_N = length(rxt_tot(1,:));
    [a, pilots_n]=max(diff(mink(var(imag(rxt_tot)),sub_N))); % pilots_n = number of
    pilots, based on varriance among symbols, spike in differential between random
    subcarriers and constant pilots
    [a, pilots] = mink(var(imag(rxt_tot)),pilots_n); % since number of pilots is
    known, take this number of minimum varriance in subcarriers among symbols
    constel = rxt_tot;
end
```

Listing B.1: Discontinuity based technique

B.2 OFDM Parameter estimation - autocorrelation based

```
function [constel,pilots_n, sub_N, symtime, cp, numberofsymbols] = OFDM_sep_resamp(
```

```
    sig)
t=abs(xcorr2(sig));
tx=1:length(t);rt=1:length(t);
TF = islocalmax(t,'MinSeparation',31);
[a b]=maxk(t(TF),3);
len = abs(tx(t==a(1))-min(tx(t==a(2))));
while log2(len)~=round(log2(len))
    lennew = 2^ceil(log2(len));
    sig = resample(sig,lennew,len);
    t=abs(xcorr(sig));
    tx=1:length(t);rt=1:length(t);
    TF = islocalmax(t,'MinSeparation',32);
    [a b]=maxk(t(TF),3);
    len = abs(tx(t==a(1))-min(tx(t==a(2))));
end
[y, x] = max(abs(diff(abs(sig))));
tsamp=[];
for val=2:(0.5*len)
    samp1 = sig(1,1:val);
    samp2 = sig(1,len+1:len+1+val);
    t=xcorr(samp1,samp2);
    tsamp(val)=abs((t(ceil(length(t)/2))));
end
[c, cp] = min(diff(diff(tsamp(2:length(tsamp)))));
if c>-0.02
    cp = 0;
else
    cp=cp+2;
end
wave = len+cp;
symtime=len;
%% Separate subcarriers and remove cp
numberofsymbols = length(sig)/wave;%
symbefore = floor(x/wave);
start =1;%
data_demod = [];rx_perwave=[];constel=[];
for i=1:numberofsymbols
    rx_perwave(i,:)=sig(1,(i-1)*wave+1:wave*i);
end
rx_perwave = flip(rx_perwave);
rx_persymbol = rx_perwave(:,cp +1:end);
%% Determine pilot tones based
for n_sym=1:numberofsymbols
rxt = (fftshift(fft(rx_persymbol(n_sym,:))));
dbdeb = 10^(-abs(12)/20)*max(abs(rxt)); % side lobe suppresion is -15dB from max
r1 = find(abs(rxt)>dbdeb,1,'first');
r2 = find(abs(rxt)>dbdeb,1,'last');
sub_N(n_sym)=r2-r1;
```

```
end
sub_N=round(mean(sub_N))-1;
mid = floor(r1+sub_N/2);
rxt_tot = [];
for n_sym=1:numberofsymbols
    rxt = (fftshift(fft(rx_persymbol(n_sym,:))));
    rxt_t2 = rxt([r1:mid,mid+2:r2]);
    rxt_tot(n_sym,:)=rxt_t2;
end
sub_N = 2^round(log2(length(rxt_tot(1,:))));
[a, pilots_n]=max(diff(mink(var(imag(rxt_tot)),sub_N))); % pilots_n = number of
    pilots, based on varriance among symbols, spike in differential between random
    subcarriers and constant pilots
[a, pilots] = mink(var(imag(rxt_tot)),pilots_n); % since number of pilots is known,
    take this number of minimum varriance in subcarriers among symbols
constel = rxt_tot;
end
```

Listing B.2: Autocorrelation based technique

B.3 Tukey window

```
function waveout = tukeyfilt(wavein, Fs, BW, iBW, stopbanddB)
% TUKEYFILT Apply Tukey window to frequency response of input signal
% waveout = TUKEYFILT(wavein, bw, fclk) applies the Tukey window to the
% signal wavein of occupied bandwidth bw and sampling rate of fclk, and
% returns the filtered signal waveout
%     wavein:      input waveform to be filtered.  Complex valued column
%                  vector
%     Fs:          clock or sampling frequency of wavein in Hz
%     BW:          channel bandwidth of wavein in Hz
%     iBW:         occupied bandwidth of wavein in Hz.  Tip: over spec this
%                  value
%     stopbanddB:  rejection of stop band in dB


grdbnd=(BW-iBW)/2e6;
BW = BW/1e6; % normalize to MHz
stopbandmag = 10^(-abs(stopbanddB)/20); % linear rejection magnitude

t_vec = numel(wavein)/Fs;
dt = 1/Fs;
Df = 1/dt;
fspc = linspace(-Fs/(2*1e6),Fs/(2*1e6),numel(wavein));
tvec = linspace(0,t_vec,numel(wavein));

linpsd=fftshift(fft(wavein));
figure;plot(abs(linpsd))
```

```
psd=abs(linpsd);

df = fspc(2) - fspc(1);
fxtn = grdbnd;      % MHz transition for the Tukey window
osr = round(fxtn/df);   tt=0:osr;

%Tukey window profiles: RC, DZ3, DZ4
% filtype='RC';
% filtype='DZ3';
filtype='DZ4';

switch filtype
    case 'RC'
        grc = sin(pi*tt/(osr));
        xtn_rc = cumsum(grc)./sum(grc)+.0001;
        filtstr='FRC';
        n=-osr/2:osr/2;
    case 'DZ3'
        n=-osr/2:osr/2;
        gdz3=1+(4/3)*cos(2*n*pi/osr)+(1/3)*cos(4*n*pi/osr);
        xtn_dz3 = cumsum(gdz3)./sum(gdz3)+.0001;
        filtstr='FDZ3';
    case 'DZ4'
        n=-osr/2:osr/2;
        gdz4=1+1.5*cos(2*n*pi/osr)+0.6*cos(4*n*pi/osr)+0.1*cos(6*n*pi/osr);
        xtn_dz4 = cumsum(gdz4)./sum(gdz4)+.0001;
        filtstr='FDZ4';
end

%search for the stopband edges
kk=1;
while fspc(kk)<-BW/2
    kk=kk+1;
end
stopbandL = kk-1;
while fspc(kk)<BW/2
    kk=kk+1;
end
stopbandH = kk;

%build the Tukey window with proper frequency scaling, then use it
passband=ones(1,stopbandH-stopbandL-2*length(n));

switch filtype
    case 'RC'
        tw_xtn_rc = (1-stopbandmag)*xtn_rc + stopbandmag;
tukeywindow=[stopbandmag*ones(1,stopbandL) tw_xtn_rc passband fliplr(tw_xtn_rc)
    stopbandmag*ones(1,length(psd)-stopbandH)];
```

```
    case 'DZ3'
        tw_xtn_dz3 = (1-stopbandmag)*xtn_dz3 + stopbandmag;
tukeywindow=[stopbandmag*ones(1,stopbandL) tw_xtn_dz3 passband fliplr(tw_xtn_dz3)
    stopbandmag*ones(1,length(psd)-stopbandH)];
    case 'DZ4'
        tw_xtn_dz4 = (1-stopbandmag)*xtn_dz4 + stopbandmag;
tukeywindow=[stopbandmag*ones(1,stopbandL) tw_xtn_dz4 passband fliplr(tw_xtn_dz4)
    stopbandmag*ones(1,length(psd)-stopbandH)];
end

newlinpsd = linpsd.*tukeywindow';
waveout = ifft(fftshift(newlinpsd));
% newpsd=abs(newlinpsd);
%% Tukey window overlay
% figure(51)
% plot(fspc,20*log10(psd/max(psd)),'k-'); grid on; hold on;
% plot(fspc,tukeywindow,'b');hold off;
%%axis([-2*bw 2*bw -20 2]);
% axis([-bw bw -15 5]);
%
%% PSD before and after
% figure(52);
% plot(fspc,20*log10(psd/max(psd)),'b-'); hold on;
% plot(fspc,20*log10(newpsd/max(newpsd)),'k-'); hold off; grid on;
% axis([-fclk/2e6 fclk/2e6 -100 0]);
```

Listing B.3: Tukey window

B.4 OFDM Waveform Generator

```
function [sig_cp data_in] = OFDMsiggen(fc,mod_method,sub,cp,sym_n,noise,snr,samp)
% OFDM modulator by K.P. van der Mark figure;plot(fftshift(abs(fft(sig_cp))))
%% Determine pilots
pil = max(floor(log2(sub/32)),-1);
pil = pil*2+4;
if rem(ceil(sub/pil),2)==0 %even sub over pilots, so no clear middle
    mid = (sub-pil)/pil-1;
    space = linspace(1,pil+1,pil+1);
    space(ceil(length(space)/2)) = mid;
    for s=2:(ceil(length(space)/2)-1)
        space(s)=round(sub/pil-1);
        space(length(space)-s+1)=sub/pil-1;
    end
    out = sub-pil-sum(space(2:length(space)-1));
    space(1)=out/2;
    space(length(space))=out/2;
else % Uneven, so clear middle
    mid = round(sub/pil-1);
```

```
    space = linspace(1,pil+1,pil+1);
    space(ceil(length(space)/2)) = mid;
    for s=2:(ceil(length(space)/2)-1)
        space(s)=ceil(sub/pil);
        space(length(space)-s+1)=ceil(sub/pil);
    end
    out = sub-pil-sum(space(2:length(space)-1));
    space(1)=out/2;
    space(length(space))=out/2;
end
space(length(space))=[];
pilots = linspace(0,pil-1,pil);
for q=1:length(pilots)
    pilots(q)=floor(space(q)+1 + pilots(max(q-1,1)));
end
%% Input data
mod_methods = {'BPSK', 'QPSK', '8PSK', '16QAM', '32QAM', '64QAM'};
mod_order = find (ismember (mod_methods, mod_method));
b=num2str(randi([0 1],sym_n*sub*mod_order,1,'single')); % omzetten randi 0 - 255 en
    dan omzetten naar bits
% calculate remainder
sym_rem = mod( mod_order-mod( length( b), mod_order), mod_order);
% Pad with zeroes if needed
padding = repmat ( '0', sym_rem, 1);
b_padded = [b; padding];
sym_send = reshape( b_padded, mod_order, length(b_padded)/mod_order)';

%% symbol modulation
%BPSK
sym_bookb = [1,-1];
symbolsb = {'1','0'};
if mod_order == 1
    symbols = {'1','0'};
    sym_book = [1,-1];
end
% QPSK
if mod_order == 2
    symbols = {'11','01','00','10'};
    sym_book = [sqrt(2)/2-sqrt(2)/2*1i,-sqrt(2)/2-sqrt(2)/2*1i,-sqrt(2)/2+sqrt(2)
    /2*1i,sqrt(2)/2+sqrt(2)/2*1i];
end
% 8PSK - Gray coding 8PSK from wikipedia
if mod_order == 3
    symbols = {'101','100','000','001','011','010','110','111'};
    bps = 2^(mod_order-1);
  n = 0: pi/bps: 2*pi-pi/bps;
  in_phase = cos(n+pi/4);
  quadrature = sin(n+pi/4);
```

```
65    sym_book = (in_phase + quadrature*1i)';
66  end
67
68  %16QAM - Gray coding from slides Earl
69  if mod_order == 4
70      symbols = {'0000','0001','0011','0010','0100','0101','0111','0110','1100','1101'
        ,'1111','1110','1000','1001','1011','1010'};
71    bps = sqrt(2^mod_order);
72    in_phase = repmat(linspace(-sqrt(2)/2, sqrt(2)/2, bps), bps, 1);
73    quadrature = repmat(linspace(-sqrt(2)/2, sqrt(2)/2, bps)', 1, bps);
74    sym_book = in_phase(:) + quadrature(:)*1i;
75  end
76
77  %32QAM - Quasi Gray coding
78  if mod_order == 5
79    symbols = {'00110','00111','00101','00100','00010','01110','01111','01101','01100'
        ,'00000','00011','01010','01011','01001','01000','00001','10011','11010','11011',
        '11001','11000','10001','10010','11110','11111','11101','11100','10000','10110','
        10111','10101','10100'};
80    bps = 6;
81    in_phase = repmat(linspace(-.8334, .8334, bps), bps, 1);
82    quadrature = repmat(linspace(-.8334, .8334, bps)', 1, bps);
83    sym_book = in_phase(:) + quadrature(:)*1i;
84    sym_book = sym_book([2:5  7:30  32:35]);
85  end
86  % 64QAM
87  if mod_order == 6
88      symbols = {'000000','000001','000011','000010','000110','000111','000101','
        000100','001000','001001','001011','001010','001110','001111','001101','001100','
        011000','011001','011011','011010','011110','011111','011101','011100','010000','
        010001','010011','010010','010110','010111','010101','010100','110000','110001','
        110011','110010','110110','110111','110101','110100','111000','111001','111011','
        111010','111110','111111','111101','111100','101000','101001','101011','101010','
        101110','101111','101101','101100','100000','100001','100011','100010','100110','
        100111','100101','100100'};
89    bps = sqrt(2^mod_order);
90    in_phase = repmat(linspace(-sqrt(2)/2, sqrt(2)/2, bps), bps, 1);
91    quadrature = repmat(linspace(-sqrt(2)/2, sqrt(2)/2, bps)', 1, bps);
92    sym_book = in_phase(:) + quadrature(:)*1i;
93  end
94
95  for c=1:length(sym_send(:,mod_order))
96      loc(c) = find(ismember(symbols,sym_send(c,:)));
97  end
98  X = sym_book(loc);
99  sym_send=cellstr(sym_send);
100 for r=1:sym_n
101     X(pilots+sub*(r-1))=1+0i;
```

```
    sym_send(pilots+sub*(r-1))=cellstr('1');
end


%% move to time domain - manual
fft_rem = mod(sub-mod(length(X), sub), sub); % remainder to be padded
X_padded = [X; zeros(fft_rem,1)];
% sym_n = length(X_padded)/sub; % calculate numer of symbols
n = linspace(0,2*pi,sub*samp);n(length(n))=[];
re_sum = []; im_sum = []; re_cp=[];im_cp=[];fcreim=[];
cp=floor(length(n)*cp/100); % Convert to %, rounded down
for k=1:sym_n
    temp = ([1:sub]);
    temp = temp+(sub*(k-1)); % per loop, which part of input data is one full symbol
    Xin=X_padded(temp);
    t1=linspace(-sub/2,sub/2,sub+1);
    t1(sub/2+1)=[]; % Define a space for each subcarrier, e.g for 4: -2 -1 1 2, as
    in literature, to define rotation
    for c=1:sub
        re(c,:)=abs(Xin(c))*cos(t1(c)*n+angle(Xin(c)));
        im(c,:)=abs(Xin(c))*sin(t1(c)*n+angle(Xin(c)));
    end
    re_cp = [re(:,end-cp+1:end) re];
    re_sum(k,:) = sum(re);
    im_sum(k,:) = sum(im);
end
fcreim = [fcreim re*cos(2*pi*fc)+im*sin(2*pi*fc)];
re_cp = [re_sum(:,end-cp:end) re_sum]; % Add cyclip prefix for Real part
im_cp = [im_sum(:,end-cp:end) im_sum]; % Add cyclip prefix for Imag part
comb=(re_cp+im_cp*1i); % Combined im and re parts for final signal
sig_cp = [];
for c=1:sym_n
    sig_cp = [sig_cp comb(c,:)];
end

sigfc = real(sig_cp)*cos(2*pi*fc)+imag(sig_cp)*sin(2*pi*fc);

%% Add noise
if noise == 1
    pwr = mean(abs(sig_cp.^2));
    noise = pwr/10^(snr/10); % Determine noise power based on input snr
    noise = normrnd(0,sqrt(noise/2),size(sig_cp)) + normrnd(0, sqrt(noise/2), size(
    sig_cp))*1i;
    sig_cp = sig_cp + noise; % Add noise to signal
    snr_meas = 10*log10(mean(abs(sig_cp.^2))/mean(abs(noise.^2))); % Verify SNR
    level
end
I = real(sig_cp);
```

```
147 Q = imag(sig_cp);
148 t=length(im);
149 cpsamp = cp;
150 data_in = cellstr(sym_send);
151 end
```

Listing B.4: OFDM Waveform Generator

Bibliography

[1] International Energy Agency, "More Data Less Energy," in , Jul 2014.

[2] P. Gandotra and R. K. Jha and S. Jain, "Green Communication in Next Generation Cellular Networks: A Survey," *IEEE Access*, vol. 5, no. , pp. 11 727–11 758, 2017.

[3] S. B. Weinstein, "The history of orthogonal frequency-division multiplexing [History of Communications]," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 26–35, November 2009.

[4] David L. Adamy, *EW 104: EW Against a New Generation of Threats*. Artech House, 2015.

[5] D. Curtis Schleher, *Electronic Warfare in the Information Age*. Artech House, 1999.

[6] Roger N. McDermott, *Russia's Electronic Warfare Capabilities to 2025*. International Centre for Defence and Security, 2017.

[7] Malte von Spreckelsen, *Electronic Warfare – The Forgotten Discipline*. JAPCC Journal 27, 2018.

[8] David Adamy, *EW 101: A First Course in Electronic Warfare*. Artech House, 2001.

[9] (2019) Electronic Defence. [Online]. Available: https://www.tno.nl/nl/aandachtsgebieden/defensie-veiligheid/expertisegroepen/electronic-defence/

[10] (2019) Plane almost had 'potentially fatal' collision with drone near UK airport. [Online]. Available: https://www.independent.co.uk/news/uk/home-news/plane-drone-london-oxford-airport-a9023636.html

[11] MyDefence. (2019) Protecting airports against drones White Paper. Nørresundby, Denmark. [Online]. Available: https://mydefence.dk/wp-content/uploads/2019/02/White-Paper-Protecting-airports-against-drones.pdf

[12] (2019) Drones at Prisons: The Results of This 9-Month Study Show Exactly Why we Need Counter Drone Tech. [Online]. Available: https://dronelife.com/2019/03/15/drones-at-prisons-the-results-of-this-9-month-study-show-exactly-why-we-need-counter-drone-technology/

[13] (2019) DJI- Mavic 2. [Online]. Available: https://www.dji.com/nl/mavic-2?site=brandsite&from=nav

[14] (2019) DJI OCUSYNC 2.0: WHAT YOU NEED TO KNOW ABOUT THIS FPV TRANSMISSION SYSTEM. [Online]. Available: http://djibestdrones.com/dji-ocusync-2-0/

[15] A. Marwanto, m. a. Sarijari, N. Fisal, S. Kamilah, and R. A Rashid, "Experimental study of ofdm implementation utilizing gnu radio and usrp - sdr," in , 01 2010, pp. 132 – 135.

[16] Chadha, Ankit and Satam, Neha and R. Ballal, Beena, "Orthogonal Frequency Division Multiplexing and its Applications," *International Journal of Science and Research*, vol. 2, p. 325, 01 2013.

[17] Dr.Ir. G.M.J. Janssen, *ET-4358 Wireless Communications.* Delft University of Technology, 2018.

[18] K. van der Mark, *Reverse Wireless Signal Engineering Survey.* Delft University of Technolog, 2018.

[19] Mathworks. (2014) comm.ofdmdemodulator. [Online]. Available: https://nl.mathworks.com/help/comm/ref/comm.ofdmdemodulator-system-object.html

[20] Ishii and Wornell, "OFDM Blind Parameter Identification in Cognitive Radios," in *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, no. , Sep. 2005, pp. 700–705.

[21] A. Al-Habashna and O. A. Dobre and R. Venkatesan and D. C. Popescu, "Second-Order Cyclostationarity of Mobile WiMAX and LTE OFDM Signals and Application to Spectrum Awareness in Cognitive Radio Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 1, pp. 26–42, Feb 2012.

[22] O. Ali, F. Nasir, and A. A. Tahir, "Analysis of OFDM parameters using Cyclostationary spectrum sensing in cognitive radio," in *2011 IEEE 14th International Multitopic Conference*, Dec 2011, pp. 301–305.

[23] Z. Sun, Q. Wang, and C. Che, "Study of Cognitive Radio Spectrum Detection in OFDM System," in *2010 Asia-Pacific Conference on Wearable Computing Systems*, April 2010, pp. 235–238.

[24] leopedrini. (2014) What is ISI (Inter Symbol Interference) in LTE? [Online]. Available: http://www.telecomhall.com/what-is-isi-inter-symbol-interference-in-lte.aspx

[25] Chang, Robert W., "Synthesis of Band-Limited Orthogonal Signals for Multichannel Data Transmission," *Bell System Technical Journal*, vol. 45, no. 10, pp. 1775–1796, 1966. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1966.tb02435.x

[26] Leon W. Couch, *Digital and Analog Communication Systems.* Pearson, 2013.

[27] (2019) Using OFDM signals in wireless communication. [Online]. Available: https://nl.mathworks.com/discovery/ofdm.html

[28] E. McCune, *Practical Digital Wireless Signals.* Cambridge: Cambridge University Press., 2010.

[29] Y. du and K. Tai Chan, "System performance of concatenated stbc and block turbo codes in dispersive fading channels," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, 05 2005.

[30] H. Zhou, A. V. Malipatil, and Y. Huang, "Synchronization issues in ofdm systems," in *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*, Dec 2006, pp. 988–991.

[31] Tektronix, *Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements*. Tektronix, 2018.

[32] C. D. Parekha and J. M. Patel, "Overview on synchronization in ofdm systems," in *2016 International Conference on Advances in Computing, Communication, Automation (ICACCA) (Spring)*, April 2016, pp. 1–6.

[33] B. Muquet, Zhengdao Wang, G. B. Giannakis, M. de Courville, and P. Duhamel, "Cyclic prefixing or zero padding for wireless multicarrier transmissions?" *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 2136–2148, Dec 2002.

[34] M. Toeltsch and A. F. Molisch, "Efficient ofdm transmission without cyclic prefix over frequency-selective channels," in *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. PIMRC 2000. Proceedings (Cat. No.00TH8525)*, vol. 2, Sep. 2000, pp. 1363–1367 vol.2.

[35] The MathWorks, Inc., *MATLAB Release 2018b*. Natick, Massachusetts, United States, 2018.

[36] Federal Comn1unications Commission, "Spectrum policy task force report," *"ET Docket No. 02-135"*, 2002.

[37] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan 1978.

[38] T. Sultana, M. S. A. Showkat, M. S. Alam, D. Hossain, and A. K. Mandal, "Reducing peak to average power ratio of ofdm signals using tukey window technique," 2013.