Channel Estimation and Equalisation for DVB-T

by

Junzhe Zhu

Supervisor:
Prof. Dr.-Ing. Thorsten Herfet

Advisor:
M.Sc. Kelvin Chelli

Reviewers:
Prof. Dr.-Ing. Thorsten Herfet
Prof. Dr.-Ing. Michael Moeller

in the

Telecommunications Lab

Department of Computer Science

May 2017

Master Thesis

Junzhe Zhu

2556095

Channel Estimation and Equalisation for DVB-T

Telecommunications Lab

Department of
Computer Science

Prof. Dr.-Ing. Thorsten Herfet

Universität des Saarlandes
Campus Saarbrücken
C6.3, 10. OG
66123 Saarbrücken

Phone: +49 681 302-70852
Fax: +49 681 302-70857

www.nt.uni-saarland.de

Digital Video Broadcast-Terrestrial (DVB-T) is a digital television standard for the transmission of digital terrestrial television. While originally being designed and developed for fixed receivers, reception of broadcast content while in motion has become an important requirement in the connected world of today. The second generation of digital terrestrial broadcast (DVB-T2) makes several improvements with respect to the transmission capacity, integration of IPv4 and the ability to receive content under low mobilty in certain modes. However, the core problem of mobility is not directly addressed. In this context, studying the effects of mobility and developing algorithms that are able to estimate the channel accurately with a complexity that is relevant for consumer hardware is addressed in this thesis. The algorithms for the compensation of high mobility are implemented as a generic block in GNU Radio that can be configured to be used in different communication systems. The goal is to enable high mobility in current as well as future DVB systems.

In particular the thesis includes the following tasks:

- Evaluate the DVB-T transceiver chain and make sure that it conforms to the DVB-T standard.
- Study the working of the transceiver in the presence of a doubly selective channel
- Study the effects of a doubly selective channel.
- Implement the Matching Pursuit algorithm for the DVB-T standard.
- Implement a generic block for channel estimation that can be configured to be used in different standards.
- Evaluate the performance of the developed schemes in terms of Bit Error Rate (BER) and complexity for different channel conditions.

Advisor:

Kelvin Chelli, M.Sc.

Supervisor:

Prof. Dr.-Ing. Thorsten Herfet

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis. I hereby confirm the congruence of the contents of the printed data and the electronic version of the thesis.

May 2017

Junzhe Zhu

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

May 2017

Junzhe Zhu

Abstract

Multipath propagation and Doppler shift mainly affect the signal transmission through the wireless channel. Digital Video Broadcast-Terrestrial (DVB-T) uses Orthogonal Frequency-Division multiplexing (OFDM) modulation to transmit compressed digital audio, digital video and other data in a MPEG transport stream. Single-Frequency Network (SFN) which allows two or more transmitters to carry the same data operating on the same frequency at the same time is also used in DVB-T. The application of the OFDM modulation and the operation SFN can bring some issues (e.g. multipath propagation) associated with the signal receiving at the same time. Due to the multipath propagation, the receiver always obtains a number of signals containing various amount of delay compared to the first arrived signal. The Doppler shift also appears when transmitter or receiver is in relative motion of each other.

The goal of this thesis is to estimate the effect of the delays and Doppler effect separately by using different methods and to equalise the received signal. The main adopted method is applying matching pursuit algorithm in the frequency domain to estimate delays and Doppler shift. Thereafter, the received data is equalised. The simulation is performed on GNU Radio DVB-T project. The performance is analysed on Matlab.

Acknowledgements

I would first like to express my sincere gratitude to my supervisor Prof. Dr.-Ing. Thorsten Herfet. He has guided me through all the stages of my master study, from the lectures studying to the thesis writing. I would also like to thank Prof. Dr.-Ing Michael Moeller for reviewing my thesis.

Second, I would like to express my heartfelt appreciation to my advisor Kelvin Chelli. He has helped me a lot for the project research and the thesis writing. I also want to thank other group members (especially, Yongtao Shuai, Praharsha Sirsi) in the Telecommunications Lab. They encourage and help me a lot for my work and life.

In addition, I have to thank my parents for their continuous support and encouragement. Finally, I would like to thank my wife specially. She lets me have family during the maser study.

Contents

List of Figures

List of Tables

Abbreviations

DVB-T/T2	**D**igital **V**ideo **B**roadcast-**T**errestrial/**T**errestrial 2
DVB-C/C2	**D**igital **V**ideo **B**roadcast-**C**able/**C**able 2
DVB-S/S2	**D**igital **V**ideo **B**roadcast-**S**atellite /**S**atellite 2
DVB-H	**D**igital **V**ideo **B**roadcasting- **H**andheld
ETSI	**E**uropean **T**elecommunications **S**tandards **I**nstitute
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
ISI	**I**nter **S**ymbol **I**nterference
FFT	**F**ast **F**ourier **T**ransform
SFN	**S**ingle **F**requency **N**etwork
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
TPS	**T**ransmission **P**arameter **S**ignalling
DFT	**D**iscrete **F**ourier **T**ransform
DBPSK	**D**ifferential **B**inary **P**hase **S**hift **K**eying
GI	**G**uard **I**nterval
CP	**C**yclic **P**refix
SNR	**S**ignal to **N**oise **R**atio
TU6	**T**ypical **U**rban **6**-tap
RIP	**R**estricted **I**sometry **P**roperty
LS	**L**east **S**quare
MSE	**M**ean **S**quare **E**rror
MMSE	**M**inimum **M**ean **S**quare **E**rror
QAM	**Q**uadrature **A**mplitude **M**odulation
QPSK	**Q**uadrature **P**hase **S**hift **K**eying
PRBS	**P**seudo **R**andom **B**inary **Sequence**
MPEG-2	**M**oving **P**ictures **E**xpert **G**roup-2

AWGN	**A**dditive **W**hite **G**aussian **N**oise
BMP	**B**asic **M**atching **P**ursuit
OMP	**O**rthogonal **M**atching **P**ursuit
BER	**B**it **E**rror **R**ate
RS	**R**eed **S**olomon
DAC	**D**igital **A**nalog **C**onverter

Symbols

$\mathbf{C}$	circular matrices
Λ	diagonal matrices
$\mathbb{C}$	complex space
dB	Decibel
$< \cdot, \cdot >$	inner product
$\lvert b \rvert$	absolute value
$\lVert \cdot \rVert$	L^2 norm
$\mathbb{R}$	real space

Chapter 1

Introduction

1.1 Motivation

Since Television (TV) has been used to share and obtain information, it brings great convenience and lots of entertainment. In modern life, people hope to receive television signal through different devices in different circumstances, such as TV in the front of a bus. In order to timely meet the customers' demand for digital TV, different transmission standards have been issued by the Digital Video Broadcasting (DVB) project, including DVB-C/-C2 for the cable network, DVB-S/S2 for satellite broadcasting, DVB-H for handheld devices, DVB-T/T2 for terrestrial broadcasting. Among those, Digital Video Broadcasting-Terrestrial (DVB-T) is the basic standard and has been widely used for television signal transmission. This thesis analyses the DVB-T standard and investigates the strategies to improve the performance of the received signal.

1.2 Basic definitions

1.2.1 Digital Video Broadcasting (DVB)

Digital Video Broadcast-Terrestrial (DVB-T) is a digital television standard for broadcast transmission of digital terrestrial television [1], which is published as EN 300 744. According to [1], 166 countries have adopted or deployed either DVB-T or DVB-T2. The below figure shows the adoption of the DVB-T/T2 standard.

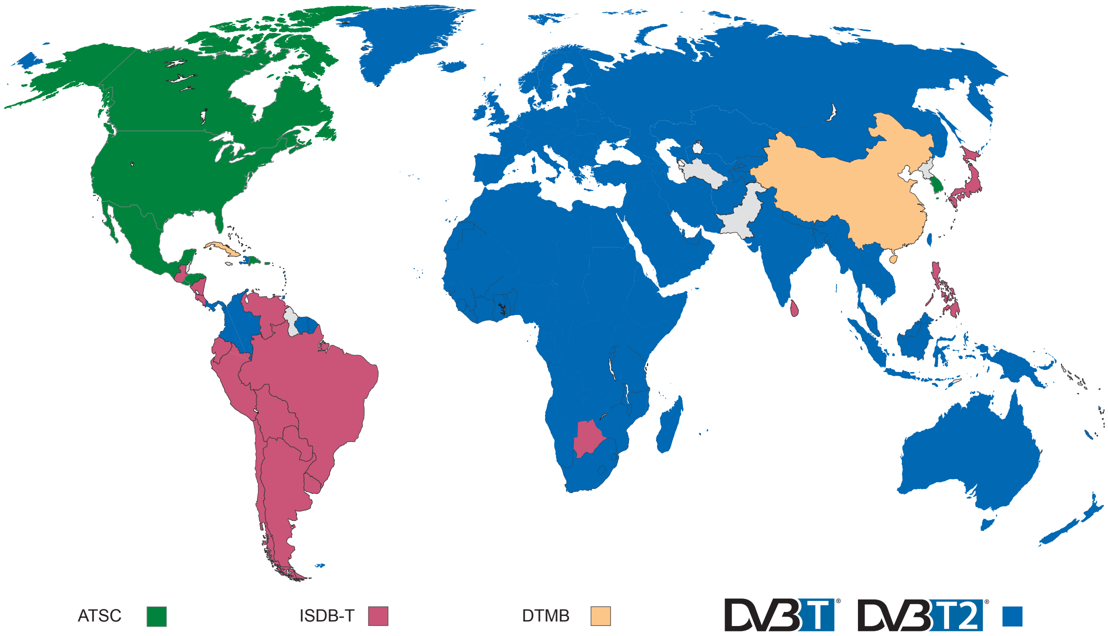

FIGURE 1.1: Digital terrestrial television systems worldwide.
[2]

The DVB-T system transmits compressed digital audio, digital video and other data in an MPEG transport stream, using coded orthogonal frequency-division multiplexing modulation.
DVB-T provides choices of three different modulation schemes (QPSK, 16QAM, 64QAM). Several newer standards have been developed based on DVB-T, such as DVB-H (Handheld), which was a commercial failure and no longer in operation [2], and DVB-T2, the second generation of DVB-T.

1.2.2 OFDM

Orthogonal Frequency-Division Multiplexing (OFDM) [3] has been used in a number of wireless communication systems, such as Wireless LAN (WLAN) and DVB system. The main idea of OFDM modulation is using multiple sub-carriers within the same channel to transform digital data. Each sub-carrier can be modulated with Quadrature Amplitude Modulation (QAM) , Quadrature Phase-Shift Keying (QPSK), or other schemes.

1.2.3 Single-frequency network

Single-Frequency Network (SFN) [4] is a broadcast network including several transmitters simultaneously transmitting the same signal over the same frequency channel and the same time [5]. The picture below illustrates the simplified SFN model.

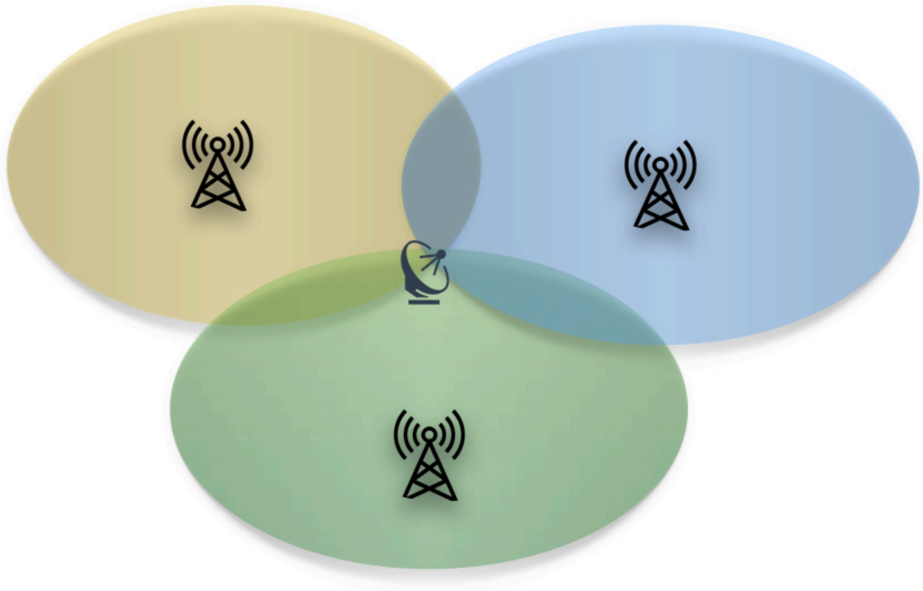

Figure 1.2: Single-Frequency Network Model

In a SFN, the signal at the receiver is the superposition of all signals with different delays coming from all the transmitters. In DVB-T, SFN is used among main transmitter towers.
According to [1], the "2K mode" is suitable for single transmitter operation and for small SFN networks with limited transmitter distances. The "8K mode" can be used both for single transmitter operation and for small as well as large SFN networks.

1.3 Structure of the thesis

The structure is organised as follows:
Chapter 2 is the theoretical part. This chapter reviews the OFDM modulation, DVB-T standard and data frame, channel model, GNU Radio and compressive sensing.
Chapter 3 firstly analyses the channel characters and mathematically expresses the main factors, then describes different channel estimation methods and focus on the matching pursuit estimation method, which contains the delay and Doppler effect elements estimation. The channel estimation is a key part, the accuracy of it directly determines the quality of the system transmission.
In chapter 4, previous to equalisation of the signal, the channel response is reconstructed

according to the estimated results. And different channel equalisation methods are discussed.

The last task is the implementation of the theory. The performance of channel tracking and equalisation are also analysed in chapter 5.

Chapter 6 is the conclusion part.

Chapter 2

Theoretical overview

In this chapter, the theoretical part will be reviewed, including OFDM, DVB-T, GNU Radio, channel model and compressive sensing.

2.1 Orthogonal frequency-division multiplexing

In OFDM, a series of sub-carriers are used to carry signal data. Those sub-carriers are orthogonal to each other in spacing. The use of OFDM which allows more sub-carriers per bandwidth resulting in an increase in spectral efficiency [3]. The OFDM signal can be mathematically expressed as [6],

$$g_t = \sum_{n=0}^{N-1} \left[a_n(t) + j \cdot b_n(t)\right] \cdot e^{j\pi \cdot \frac{n}{T} t} \quad \underbrace{e^{-j\pi \cdot \frac{(N-1)}{T} t}}_{center\ signal\ at\ zero} \tag{2.1}$$

The complex signal data $a_n(t) + j \cdot b_n(t)$ can be written as $a_n(\frac{iT}{N}) + j \cdot b_n(\frac{iT}{N})$, T is the time duration of the OFDM symbol. N is the FFT length. The above equation can be rewritten as:

$$g_i = \underbrace{\sum_{n=0}^{N-1} \left[a_n(\frac{iT}{N}) + j \cdot b_n(\frac{iT}{N})\right] \cdot e^{j\pi \cdot \frac{n}{T} t}}_{N-IDFT} \cdot \underbrace{e^{-j\pi \cdot \frac{(N-1)}{T} t}}_{center\ signal\ at\ zero} \tag{2.2}$$

The first part of Eq.2.2 is the inverse discrete Fourier transform over N samples. The function of second part is to center the signal at zero.

2.1.1 OFDM system implementation

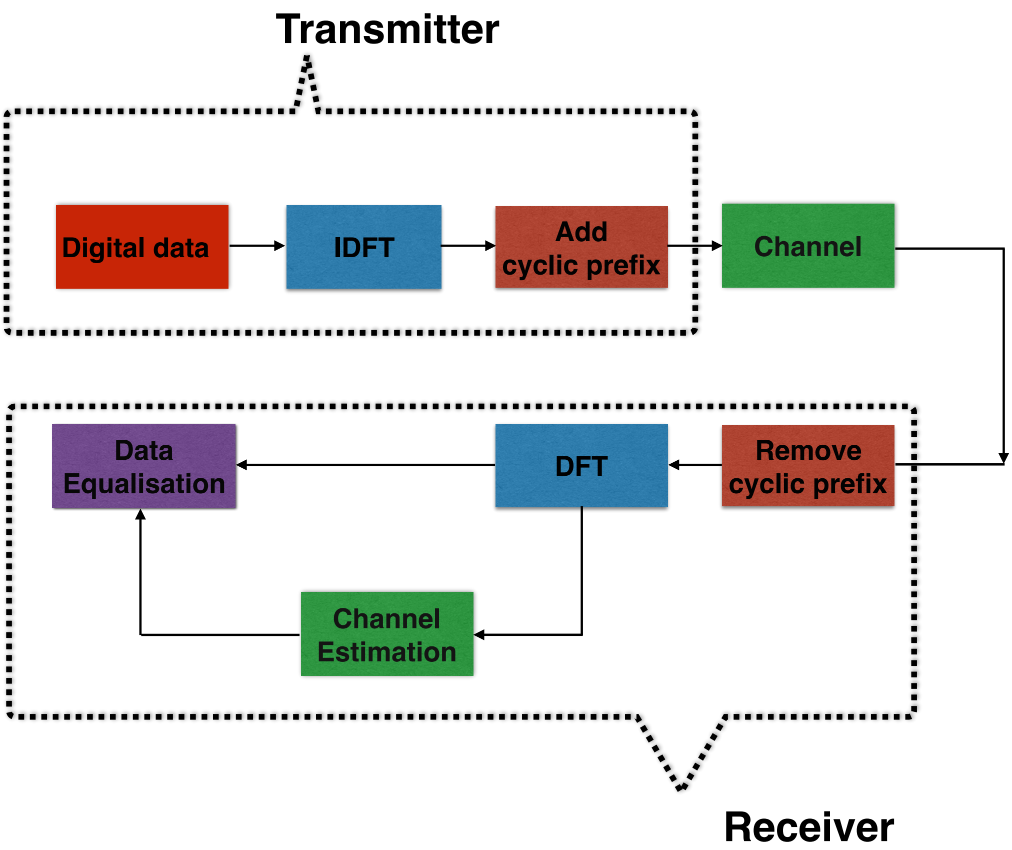

FIGURE 2.1: OFDM system model

Fig. 2.1 shows the operating principle of OFDM system. Digital signal passes through the serial-to-parallel converter by performing an Inverse Discrete Fourier Transform (IDFT). The cyclic prefix is added in the front of each symbol. After passing through the channel, the receiver first removes the cyclic prefix, then performs the Discrete Fourier Transform (DFT), and finally equalise the received data according to the channel estimation.

- **Guard Interval:** The guard interval is the space between the symbols used to eliminate Inter-Symbol Interference (ISI). ISI happens when echoes or reflections from one symbol interfere with another. Adding time between symbol transmission allows these echos and reflections to gathered before the next symbol is transmitted [7].

- **Cyclic Prefix:** Cyclic Prefix (CP) is a repetition of the end of the symbol which is placed at the start of the symbol. As similar to guard interval, the CP is used to eliminate ISI from the previous symbol. Fig. 2.2 shows cyclic prefix in the OFDM symbol.

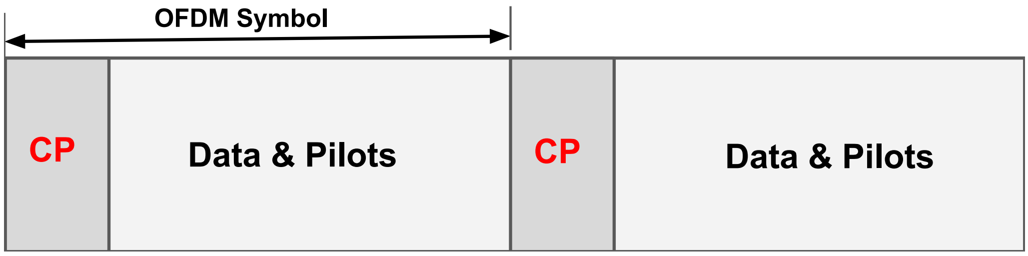

FIGURE 2.2: Insertion of Cyclic Prefix

The below figure shows the theory of an OFDM symbol in the frequency domain and time domain. In the frequency domain, sub-carriers are each independently modulated and orthogonal to each other. In the time domain, in order to prevent inter-symbol interference, guard intervals are inserted between each of the symbols.

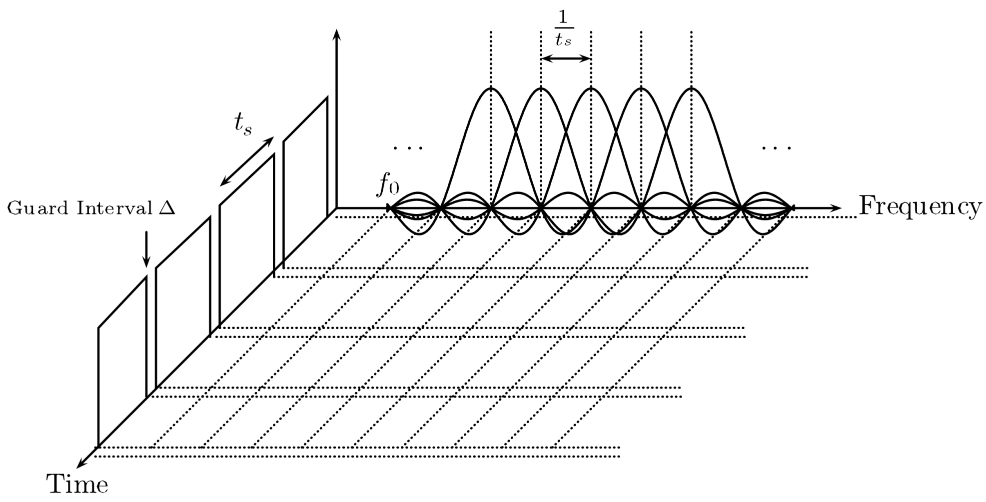

FIGURE 2.3: Orthogonal frequency-division multiplexing
[6]

2.1.2 Advantages of OFDM

- By using multiple orthogonal sub-carriers, the spectral use efficiency is increased[8].
- The Inter-Symbol Interference (ISI) can be eliminated by inserting Cyclic prefix in front of each OFDM symbol.

- OFDM is computationally efficient by using FFT techniques to implement the modulation and demodulation functions.

2.1.3 Disadvantage of OFDM

- OFDM is more sensitive to carrier frequency offset than single carrier systems.
- High peak to average power. This is due to OFDM signal has a noise like amplitude with a very large dynamic range, which requires RF power amplifiers with a high peak to average power ratio.
- OFDM is sensitive to Doppler shift.

2.2 The DVB-T standard

The MPEG-2 is a data compression method for the video data standand, which was developed by the Moving Pictures Expert Group (MPEG). As shown in Fig.2.4, the following processes are applied to the data stream from output of the MPEG-2 transport multiplexer[1, 9, 10].

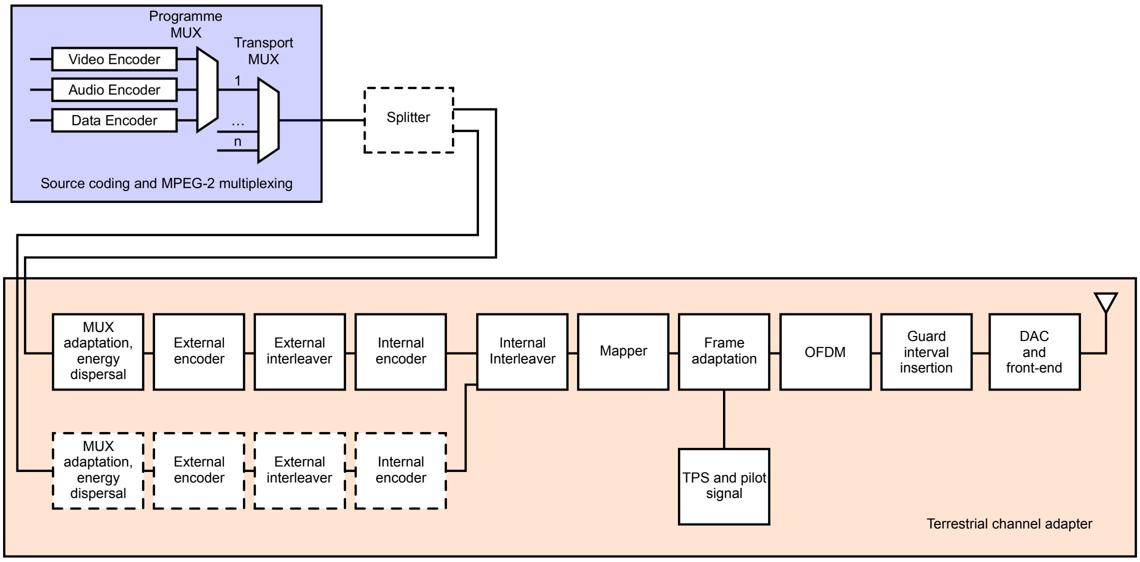

FIGURE 2.4: Functional block diagram of the DVB-T transmitter [1]

- **Transport multiplex adaptation and randomization for energy dispersal** The input stream shall be organized in fixed length packets, following the MPEG-2 transport multiplexer, as Fig.2.5. The total packet length of the MPEG-2 transport multiplex (MUX) packet is 188 bytes, including 1 sync-word byte. In order to ensure adequate binary transitions, the data of the input MPEG-2 multiplex shall be randomized in accordance with the PRBS generator.

FIGURE 2.5: MPEG-2 transport MUX packet

- **External coder**

  The external coder implements Reed-Solomon code, RS(204, 188) encoding with length 204 bytes, dimension 188 bytes, t=8 bytes error correcting capability.

- **External interleaver**

  The convolutional interleaving process should be based on the Forney approach which is compatible with the Ramsey type III approach, with I = 12. The interleaved data bytes shall be composed of error protected packets and shall be delimited by inverted or non-inverted MPEG-2 sync bytes.

- **Internal coder**

  The internal coder implements punctured convolutional encoding with mother code 1/2. The allowed puncturing rates cab be 2/3, 3/4, 5/6, 7/8. The puncturing pattern and transmitted sequence after parallel-to-serial conversion for the possible code rates are shown as Fig.2.6, where X and Y refer to the two outputs of the convolutional encoder.

Code Rates r	Puncturing pattern	Transmitted sequence (after parallel-to-serial conversion)
1/2	X: 1 Y: 1	$X_1 Y_1$
2/3	X: 1 0 Y: 1 1	$X_1 Y_1 Y_2$
3/4	X: 1 0 1 Y: 1 1 0	$X_1 Y_1 Y_2 X_3$
5/6	X: 1 0 1 0 1 Y: 1 1 0 1 0	$X_1 Y_1 Y_2 X_3 Y_4 X_5$
7/8	X: 1 0 0 0 1 0 1 Y: 1 1 1 1 0 1 0	$X_1 Y_1 Y_2 Y_3 Y_4 X_5 Y_6 X_7$

FIGURE 2.6: The puncturing pattern and transmitted sequence after parallel-to-serial conversion for the possible code rates

[9]

- **Internal interleaver**

  The Internal interleaver performs symbol interleaver and bitwise interleaver. The result should be prepared for mapping of the number of bits to the constellation according to the used modulation type.

The Bit-wise interleaving, the input, which consists of up to two bit streams, is demultiplexed into v sub-streams, where $v = 2$ for QPSK, $v = 4$ for 16-QAM, and $v = 6$ for 64-QAM. In the non-hierarchical mode, the single input stream is demultiplexed into v sub-streams. In the hierarchical mode the high priority stream is demultiplexed into two sub-streams and the low priority stream is demultiplexed into v-2 sub-streams. This applies in both uniform and non-uniform QAM modes. The purpose of the symbol interleaver is to map v bit words onto the 1512 (2K mode) or 6048 (8K mode) active carriers per OFDM symbol. The symbol interleaver acts on blocks of 1512 (2K mode) or 6048 (8K mode) data symbols.

- **Mapper**
  The DVB-T standard uses Orthogonal Frequency Division Multiplex (OFDM) transmission. All data carriers in one OFDM frame are modulated using either QPSK, 16-QAM, 64-QAM, non-uniform 16-QAM or non-uniform 64-QAM constellations. The Gray mapping is applied.

- **Pilot and Transmission Parameter Signalling (TPS) insertion**
  In order to achieve the synchronization and equalisation at the receiver, the standard adds three types of pilots: continual pilots, scattered pilots and TPS to the data frame. The data frame will be discussed in detail later.

- **Frame adaptation**
  The function of frame adaptation is to combine the signal data and the reference signal together.

- **Orthogonal Frequency Division Multiplexing (OFDM) transmission**
  The OFDM modulation is used in DVB-T for creating frequency domain bins and perform an IFFT to convert them to time domain.

- **Guard interval insertion**
  In order to prevent ISI, the guard interval is inserted in front of each symbol. The length of guard interval can be 1/32, 1/16, 1/8, or 1/4 of the original block length.

- **DAC and front-end**
  In order to transform the signal, firstly, the digital signal need to be transformed into an analogue signal by a digital-to-analog converter (DAC), and then modulated to radio frequency by the RF front end.

2.2.1 DVB-T data frame

In DVB-T, transmitted signal is organized in frames and super frames. Each frame has a duration of T_F, and consists of 68 OFDM symbols. Four frames constitute one super-frame [1].

TABLE 2.1: Main parameters of DVB-T system

Paramenters	8K mode	2K mode
Number of carriers K	8192	2048
Value of carrier number Kmin	0	0
Value of carrier number Kmax	6 816	1 704
Duration T_u	896 μs	224 μs
Carrier spacing $1/T_u$	1 116 Hz	4 464 Hz
Spacing between carriers Kmin and Kmax (K-1)/T_u	7,61 MHz	7,61 MHz

As shown in Tab.2.1, in the 2K mode, there are 1705 useful carriers; In the 8K mode, each OFDM symbol is constituted by 6817 useful carriers. Each symbol has a duration of T_s. It contains two parts: a useful part with a duration T_u and a guard interval with a duration Δ.

In addition to the transmitted data, an OFDM frame also contains: scattered pilot cells, continual pilot carriers, and Transmission Parameters Signalling (TPS) pilot carriers. The pilots can be used for frame synchronization, frequency synchronization, time synchronization, channel estimation, transmission mode identification. These three kinds of pilots are as follow:

- **Scattered pilot cells:**
  For the symbol index l (the symbols in an OFDM frame are numbered from 0 to 67), the carriers are indexed by $k \in [K_{min}, K_{max}]$ which belong to the subset

  $$k = K_{min} + 3 \cdot (l \ mod \ 4) + 12p, \quad p \in Integer, \quad p \geq 0 \tag{2.3}$$

  are scattered pilots.
  Fig. 2.7 shows the scattered pilots organization in DVB-T data frame.
  The scattered pilots are modulated according to a PRBS sequence (see Appendix B).

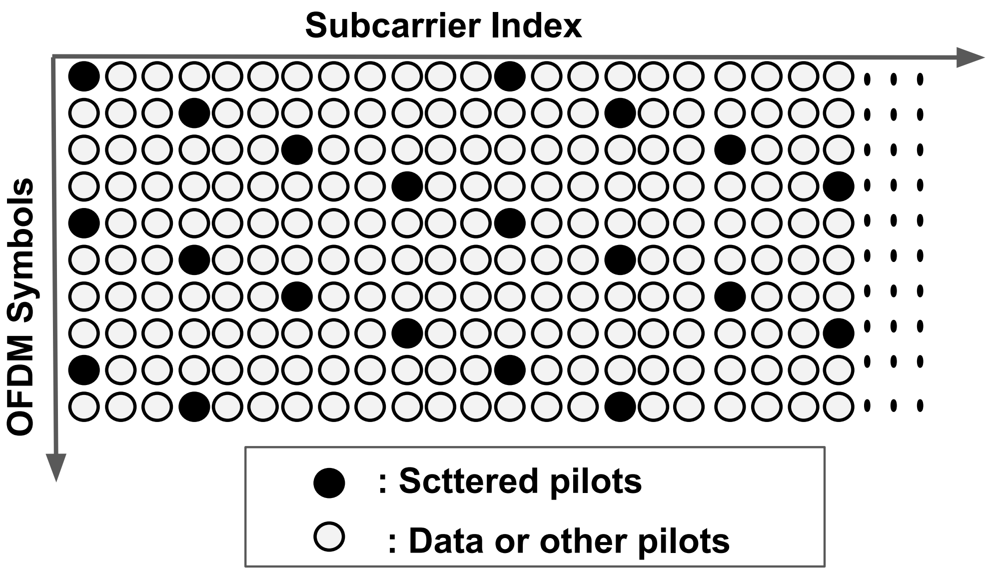

FIGURE 2.7: Scattered pilots organization

- **Continual pilot cells:**
  In addition to the scattered pilots, continual pilots are also inserted. These pilots are on the same position in every symbol. All the continual pilots are modulated according to reference sequence. The corresponding modulation is given by:

$$Re\{c_{m,l,k}\} = 4/3 * 2 * (1/2 - W_k) \tag{2.4}$$

$$Im\{c_{m,l,k}\} = 0 \tag{2.5}$$

  Where W_K is Pseudo Random Binary Sequence (PRBS) value [1].
  The locations of continual pilots can be found in appendix B.

- **TPS carriers:**
  Transmission Parameter Signalling (TPS) carriers are used for the purpose of signalling parameters related to the transmission scheme. The TPS carriers are transmitted at a normal level. The TPS data itself is Differential Binary Phase Shift Keying (DBPSK) modulated. The DBPSK is initialized at the beginning of each TPS block.
  DBPSK is a modulation with one bit per symbol. The modulation scheme: a logical one is represented by a change of the phase, a logical zero is represented by the same phase. The locations of TPS pilots can be found in appendix B.

2.2.2 The second generation DVB-T2

DVB-T2 is the extension of the television standard DVB-T, which was built based on DVB-T. This thesis will focus on the research of DVB-T [11].

2.3 GNU Radio

GNU Radio is an open source project that provides a collection of software modules to perform the operations of signal processing [12]. GNU Radio is widely used in hobbyist, academic and commercial environments to support both wireless communications research and possible implementations tests.
The software in GNU Radio is divided into modules. The modules that perform the signal processing are programmed in C++, e.g. signal filters, FFT modules. The other group of modules include the software needed to interconnect these signal processing modules and configure them according to our needs. Those modules are programmed in Python and act as some kind of glue that makes the whole system one unit [13].
The main reason to choose the GNU Radio framework as a software solution results from its efficiency and high flexibility [14]. The steps of building a new GNU Radio model is shown in appendix A.

2.3.1 GNU Radio DVB-T

The GNU Radio DVB-T project (gr-dvbt) [15] is an out-of-tree modules project (A GNU Radio component that does not exist in the GNU Radio original source tree, developed by B. Diaconescu), which implements DVB-T transmitter and receiver [16].
Fig. 2.8 shows the GNU Radio DVB-T receiver. As we can see, after the signal is received, the first step is to detect the start of the OFDM symbol, which is achieved by OFDM symbol acquisition block. And the cyclic prefix is taken out. The signal is changed to the frequency domain by FFT implementation. In order to extract the useful data, the pilots need to be cut out.
The functions of the following blocks, including demapping, decoding and deinterleaving, are the inverse action of the transmitter blocks.

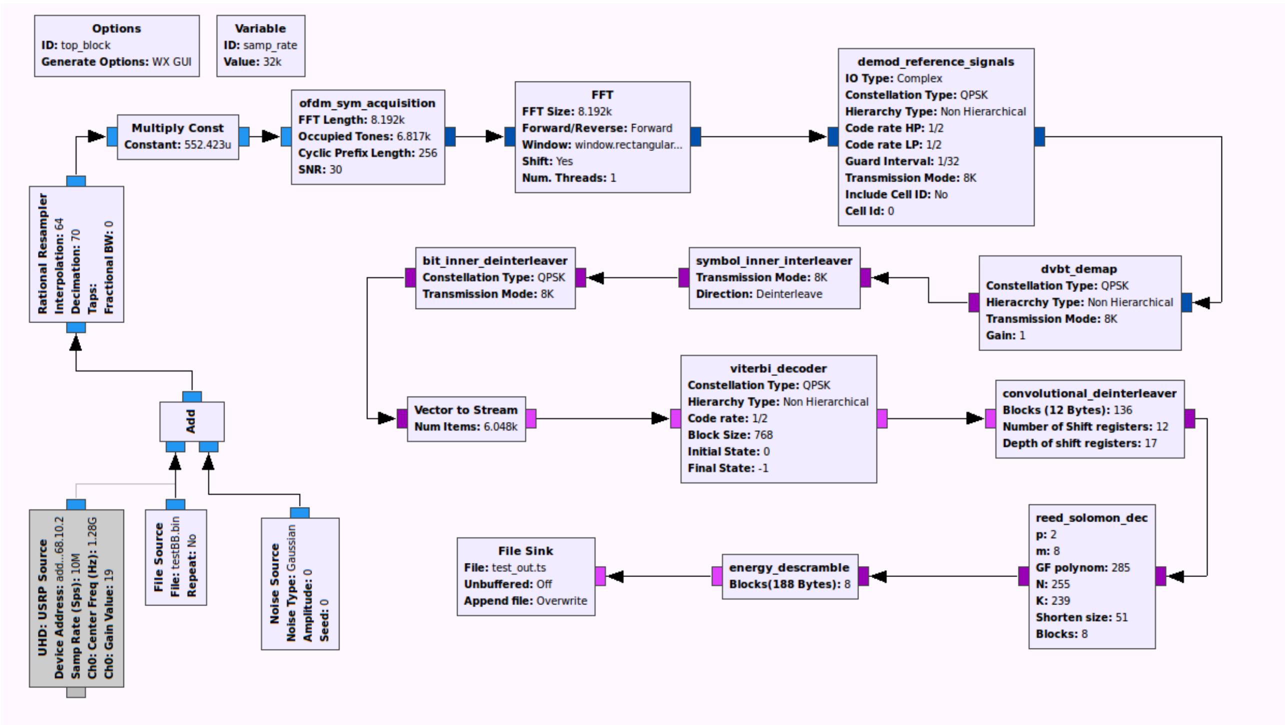

FIGURE 2.8: GNU Radio DVB-T receiver

2.4 Channel model

2.4.1 AWGN channel

Additive White Gaussian Noise (AWGN) channel model is a simple channel model which only linearly adds white noise with a constant spectral density and a Gaussian distribution of amplitude.

$$y(t) = x(t) * h(t) + n(t) \tag{2.6}$$

where $h(t)$ is the signal magnitude change, $n(t)$ is white Gaussian noise. The AWGN channel is widely used to simulate the background noise.

2.4.2 Time selective channel

The channel characteristics of the time selective channel depend on the varying time. This could be caused by the moving transmitter or receiver. The complex gain of the channel can be written as [8]:

$$\widetilde{\alpha}_n(t) = \alpha_n(t) \cdot e^{j\theta_n(t)} \tag{2.7}$$

Where α and θ are the channel attenuation and the phase rotation respectively. For the multipath condition, the overall complex gain can be written as:

$$\widetilde{\alpha}(t) = \sum_{n=1}^{N} \alpha_n(t) \cdot e^{j\theta_n(t)} \tag{2.8}$$

Then the channel response will be:

$$\widetilde{h}(t,\tau) = \widetilde{\alpha}(t) \cdot \delta(\tau) \tag{2.9}$$

where $\delta(\tau)$ is the unit-impulse function.

2.4.3 Frequency selective channel

The channel characteristics of the frequency selective channel are rely on the frequency. The channel complex gain is assumed to be constant. The channel response of the frequency selective channel can be written as:

$$\widetilde{h}(t,\tau) = \sum_{n=1}^{N} \widetilde{\alpha}_n(t) \cdot \delta(t-\tau_n) \tag{2.10}$$

The response of frequency selective channel mainly effects by different path delays.

2.4.4 Rayleigh multipath channel model

Rayleigh multipath channel model is a doubly selective channel model (both time and frequency selective) when adding Doppler effect. When the signal pass through Rayleigh channel model, the signal's magnitude will fade according to Rayleigh distribution. The arriving time of different path is various. The channel impulse response of Rayleigh multipath channel could be expressed as:

$$h(t) = \sum_{n=1}^{N} \alpha_n(t) \cdot e^{j\theta_n(t)}\delta(t-\tau_n) \tag{2.11}$$

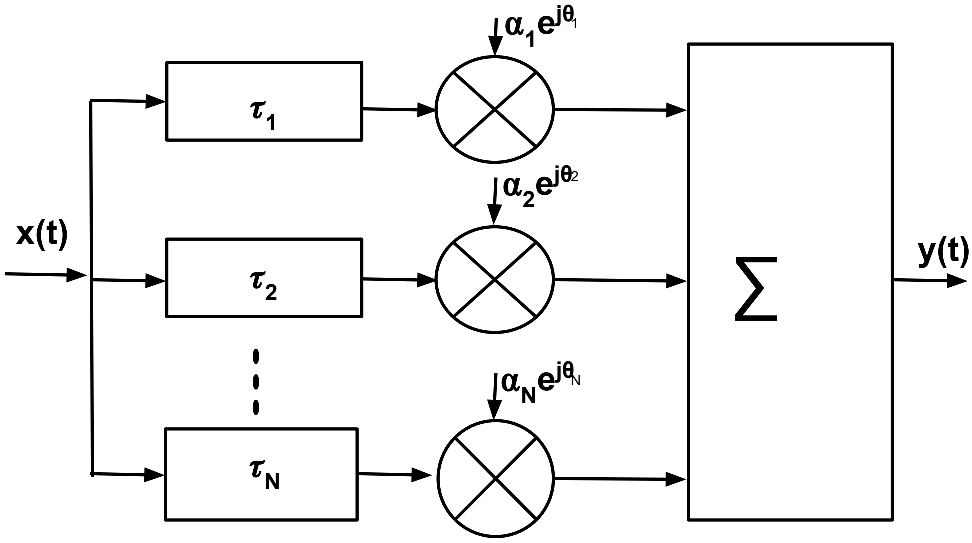

FIGURE 2.9: Rayleigh channel model

The Rayleigh channel model is showed as Fig. 2.9. After signal $x(t)$ passes through different path with a certain delay τ, different phase rotation and attenuation are added to each path.

2.5 Compressive Sensing

The compressive sensing is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solution to underdetermined linear systems [17, 18]. This is based on the principle which is through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than the Nyquist rate. There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals [19].

2.5.1 Problem statement

Given a vector $y \in \mathbb{C}^m$, and a matrix $A \in \mathbb{C}^{m \times n}$, find a vector $x \in \mathbb{C}^n$

$$y = Ax \tag{2.12}$$

In the case data is sparse or compressible, the matrices A with $m \ll n$.

2.5.2 Coherence

Definition *The coherence of a matrix A, $\mu(A)$ is the largest absolute inner product between any two columns a_i, a_j of A [17]:*

$$\mu(A) = \max_{1 \leq i \leq n} \frac{| \langle a_i, a_j \rangle |}{\| a_i \|_2 \| a_j \|_2} \tag{2.13}$$

Theorem *If the sparsity k with:*

$$k < \frac{1}{2}\left(1 + \frac{1}{\mu(A)}\right) \tag{2.14}$$

then for each measurement vector $y \in \mathbb{C}^m$ there exists at most one signal $x \in \sum_k$ such that $y = Ax$.

This theorem provides an upper bound on the level of sparsity k.

2.5.3 Signal Recovery Algorithms

The l_1-minimization and greedy methods are two major approaches to sparse recovery.

2.5.3.1 Basis pursuit

The idea of Basis Pursuit is to replace the difficult sparse problem with an easier optimization problem (l_1 minimization algorithms).

$$\widehat{x} = \arg \min_{x:Ax=y} \| x \|_1 = \sum_{i=1}^{n} | x_i | \tag{2.15}$$

The l_1 minimization can approach an exactly recovery for k-sparse signals and closely approximate compressible signals with higher probability. However, l_1 minimization approach is too slow for large scale application.

2.5.3.2 Greedy algorithms

Greedy algorithms rely on the iterative approximation of signal coefficients and support, either by iteratively identifying the support of the signal until a convergence criterion is met, or alternatively by obtaining an improved estimation of the sparse signal at each iteration that attempts to account for the mismatch to the measured data. One of the widely used greedy approaches is Orthogonal Matching Pursuit (OMP)[20, 21].

OMP constructs a sparse solution to a given problem by iteratively building up an approximation; the vector y is approximated as a linear combination of a few columns of A, where the active set of columns to be used is built column by column [22]. At each iteration the locally optimum solution is calculated. This is done by finding the column vector in A which most closely resembles to a residual vector r. The residual vector starts being equal to the vector that is required to be approximated i.e $r = y$ and is adjusted at each iteration to take the previous vector into consideration.
The following notation is used: $<\ ,\ >$ denotes the inner product. The basic idea of OMP is as follow:

Algorithm : Orthogonal Matching Pursuit [23]
Input:

- CS matrix/dictionary A, data y
- Stopping criterion

Output:

- Sparse representation $\widehat{x}$

Procedure:

1. Initialize the residual $r_0 = y$, the index set $V_0 = \emptyset$, and the iteration counter $t = 1$.
2. Let $v_t = i$, where a_i gives the solution of $max < r_t, a_k >$ where a_k are the column vectors of A.
3. Update the set V_t with v_t : $V_t = V_{t-1} \bigcup \{v_t\}$.
4. Solve the least-squares problem:

$$\underset{\widehat{x} \in \mathbb{C}^{V_t}}{min} \parallel y - \sum_{j=1}^{t} \widehat{x}_{(v_j)} a_{v_j} \parallel_2^2 \tag{2.16}$$

5. Calculate the new residual using $\widehat{x}$.

$$r_t = r_{t-1} - \sum_{j=1}^{t} \widehat{x}_{(v_j)} a_{v_j} \tag{2.17}$$

6. Set $t = t + 1$

7. Check stopping criterion; if the criterion has not been satisfied then return to step 2, while the choose $i \notin V_t$.

The next chapter will detail discuss how to use OMP for channel estimation.

Chapter 3

Channel estimation

The channel estimation is based on the channel characteristics. In this chapter, the channel response is mathematically analysed. Then different estimation algorithms will be discussed.

3.1 Channel characters analyse

3.1.1 Multipath Propagation

When a signal is transmitted, the path is not single rather than a number of different paths with different delay elements. The receiver picks up a superposition of multiple copies of the attenuated transmit signal. The delay time for a corresponding path will be donated with τ.

Suppose that the signal $x(t)$ is transmitted through a channel, there are n propagation paths, the received signal is given by:

$$y(t) = \rho_1 x(t-\tau_1) + \rho_2 x(t-\tau_2) + ... + \rho_n x(t-\tau_n) \tag{3.1}$$

where ρ_n represents the corresponding path attenuation factor. τ_n represents the delay for the corresponding path. After Fourier transform into frequency domain, the received signal can be expressed as :

$$Y(f) = X(f)(\rho_1 e^{-j2\pi f\tau_1} + \rho_2 e^{-j2\pi f\tau_2} + ... + \rho_n e^{-j2\pi f\tau_n}) \tag{3.2}$$

where $X(f)$ is the transmitted data in the frequency domain. In the frequency domain, the linear phase progression is dependent on the delay and the carrier frequency. It also has a circular repetition of 2π.

3.1.2 Doppler shift

The moving of the transmitter, receiver and/or scatters cause Doppler effect on the transmitted signal, which results in frequency shift. The Doppler shift will be denoted by ν_L. The corresponding signal in time domain can be expressed as:

$$X(f \pm \nu_L) \bullet\!\!-\!\!\circ\, x(t)e^{\mp j2\pi\nu_L t} \tag{3.3}$$

The Doppler shift in frequency domain results in linear phase progression in the time domain. In other words, Doppler shift effect causes time-dependant phase shift in the time domain.

3.1.3 Delay-Doppler spreading function

In this section, $\mathbf{C}$ and Λ denote the circular and diagonal matrices respectively. $\mathbb{C}$ denotes the complex space. $sinc(x)$ is defined as $\frac{sin(\pi x)}{\pi x}$. As mentioned before, the main effects underlying dynamic channel are Doppler effect and multipath propagation. The channel transfer function [24] is formulated for i-th sample and m-th delay bin as follows:

$$h[i,m] = \sum_{k=1}^{K-1}\sum_{l=0}^{L-1}(\mathbf{U}_{k,l})e^{j2\pi\nu_l i}sinc(m - \frac{\tau_k}{T_s}) \tag{3.4}$$

where T_s is the system sampling period. K and L are the numbers of the maximum delays and the Doppler shifts respectively. $\mathbf{U}_{k,l}$ is the channel gain matrix. The channel response matrix in the time domain can be expressed as [25]:

$$H_T = \sum_{k=1}^{K-1}\sum_{l=0}^{L-1}(\mathbf{U}_{k,l}\Lambda_{T,l}\mathbf{C}_{T,k}) \tag{3.5}$$

where $\Lambda_{T,l} \in \mathbb{C}^{N\times N}$ is the Doppler shift matrix in the time domain. If only consider about Doppler shift component, $\Lambda_{T,l} \in \mathbb{C}^{N\times N}$ will be a a diagonal matrix, which can be expressed as [26] :

$$\Lambda_{T,l} = diag(e^{j2\pi\nu_l t_0}, e^{j2\pi\nu_l t_1}, ..., e^{j2\pi\nu_l t_{N-1}}) \tag{3.6}$$

N is the length of OFDM samples. $\mathbf{C}_{T,k} \in \mathbb{C}^{N\times N}$ is the delay matrix in time domain. Because the received signal is a weighted superposition of the different echoes. So when only the delay component τ_k is considered, the corresponding channel matrix in the time domain will be a circular matrix, which can be expressed as:

$$\mathbf{C}_k = \begin{pmatrix} c_{k,0} & 0 & \cdots & c_{k,M} & \cdots \\ c_{k,1} & c_{k,0} & \cdots & 0 & c_{k,M-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & c_{k,M-1} & \cdots & c_{k,M-1} \end{pmatrix} \tag{3.7}$$

where $c_{k,m} = sinc(m - \frac{\tau_k}{T_s})$. After Discrete Fourier transform, the channel response in the frequency domain can be expressed as:

$$\begin{aligned} H_F &= \mathbf{F}\Big(\sum_{k=1}^{K-1}\sum_{l=0}^{L-1} \mathbf{U}_{k,l}\Lambda_{T,l}\mathbf{C}_{T,k}\Big)\mathbf{F}^H \\ &= \Big(\sum_{k=1}^{K-1}\sum_{l=0}^{L-1} \mathbf{U}_{k,l}\mathbf{F}\Lambda_{T,l}\mathbf{F}^H\mathbf{F}\mathbf{C}_{T,k}\mathbf{F}^H\Big) \end{aligned} \tag{3.8}$$

where H_F is the channel response in the frequency domain with $\mathbf{F} \in \mathbb{C}^{N\times N} \implies \mathbf{F}_{n,k} = \frac{1}{\sqrt{N}}exp(\frac{-j2\pi nk}{N})$. The delay matrix in the frequency domain will be:

$$\mathbf{F}\mathbf{C}_{T,k}\mathbf{F}^H = \Lambda_{F,k} \tag{3.9}$$

As Eq.3.9, delay matrix in the frequency domain is a diagonal matrix.

3.2 Traditional channel estimation methods

The channel estimation [27] steps are as follow. The first step is to estimate the channel response of pilot locations. Least Squares (LS) estimation method [28] is the easiest method. This method estimates the effects (including Doppler effect, path delays, and noise) together. The carrier complex gain and the phase rotation will not exactly equalised. In order to more clearly estimate the Doppler shift and multipath propagation effects, matching pursuit method is one of the best choices for the channel estimation. After getting the channel characteristics of pilots carriers, the following step is to get all of the sub-carrier channel characteristics.

3.2.1 The reference carriers

The scattered pilots are used as channel taps for channel estimation. The scattered pilots are placed according to the formula below:

$$k = K_{min} + 3 \cdot (l\ mod\ 4) + 12p, p \in integer, p \geq 0, k \in [K_{min}, K_{max}] \tag{3.10}$$

Where k represents the sub-carrier index where k=0 means the first used carrier (excluding the guard band) and l is the symbol index. K_{min} and K_{max} are the first data carrier index and the last data carrier index separately.

3.2.2 LS Estimation

The Least Squares (LS) estimation is the simplest channel [29]. The estimation of the channel at scattered pilot sub-carriers based on LS estimation can be written as:

$$H_{LS} = \frac{Y_p(k)}{X_p(k)} \tag{3.11}$$

where $X_p(k)$ and $Y_p(k)$ are input and output at the k^{th} scattered pilot sub-carrier respectively.
After the frequency response H_p of the pilots is obtained, the total channel frequency response H_k can be obtained by H_p interpolation. This part will be discussed in the next chapter.

3.3 Matching pursuit estimation methods

The basic theory of compressive sensing has been discussed in chapter 2. The concept of using Matching Pursuit to estimate the channel response by matching the received signal with the dictionary which builds on the transmitted references signal under the effects of different delays or Doppler shifts [26, 30–32]. Two popular adopted algorithms are Basic Matching Pursuit(BMP) and Orthogonal Matching Pursuit(OMP) algorithm.

3.3.1 Delays estimation

The character of the delays' effect has been analysed in section 1. In the frequency domain, delay causes the linear phase progression with a circular repetition of 2π. The different arrived OFDM signal has a different phase progression pattern. The idea of the delays estimation is comparing the phase progression pattern of the received signal and the dictionary, which contains all phase progression patterns corresponding to all possible delays.

3.3.1.1 Dictionary construction

When using matching pursuit algorithm to estimate the delays happen in the time domain channel, the essential part is the construction of the dictionary. The transmitted signal value and positions $X(f)$ of the scattered pilots have been defined by the DVB-T standard. The scattered pilots' positions have been defined by Eq.3.10, the corresponding values are also defined (Appendices B). The value of received scattered pilots can be easily extracted from the received symbol if their positions is known. When the signal passes through the channel, the receiver gets a number of paths signal. Others paths' signal has certain time delays compared to the first arrived path signal. The received path signal $Y(f)$ in the frequency domain with the delays τ can be expressed as:

$$Y(f) = X(f) * (e^{-j2\pi f\tau}) \tag{3.12}$$

The received data is the discrete data. So f in the Eq.3.12 can be written as:

$$f = \left(k - \frac{N}{2}\right) \times \frac{1}{T_u} \tag{3.13}$$

Where k is scattered pilots position index, which was defined according to Eq.3.10. N is the total number of the carrier index, $\frac{1}{T_u}$ is the carrier spacing which has been defined in Tab.2.1.

The dictionary construction steps can be showed as Fig.3.1. The general steps are:

- Extract the reference signal from the OFDM symbol. Here reference signal is the scattered pilots carriers. In the figure, the black color represents the reference signal carriers.
- Adding the possible delay range to the reference signal. The delay range width should not over the length of the guard band.

The dictionary is a combination of the reference signal (scattered pilots) under the effect of all possible delays (from 0 to K times delay scale).

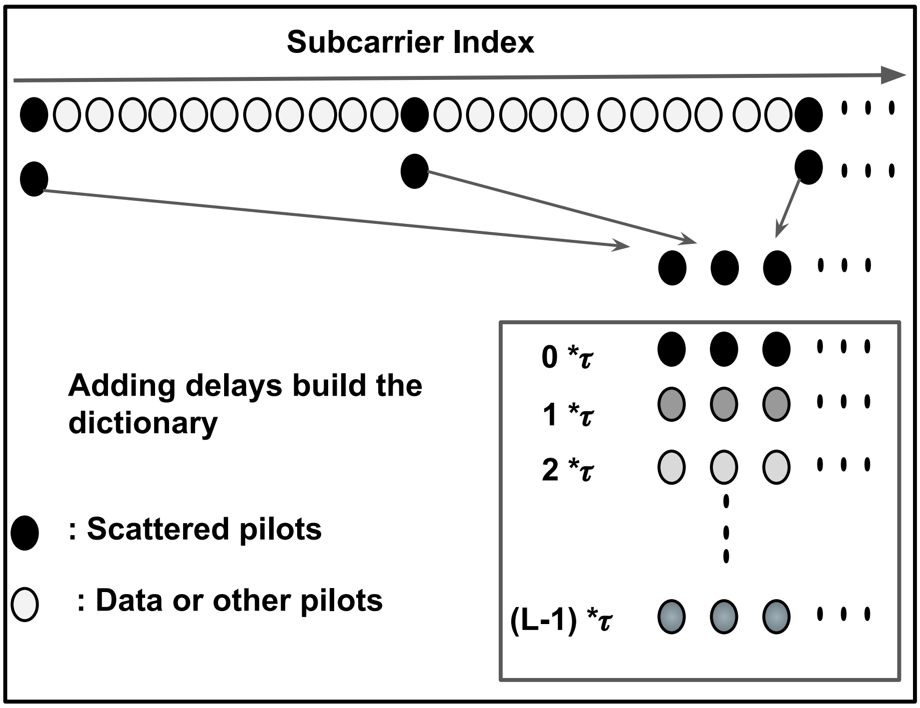

Figure 3.1: Delay dictionary construction

3.3.1.2 Basic matching pursuit

The basic MP algorithm is an iterative procedure which can sequentially identify the dominant channel taps and estimate the associated tap coefficients. At each iteration, it selects one column that correlates best to the approximation residual from the previous iteration [30, 33].
After the dictionary construction, as Tab.3.1, the dictionary, the received symbol and the stopping criterion will be inputs. The first step is to store the received scattered pilots' value as residual vector r_0. The following step is the calculation of the rank one projection r_0 on all the columns $\mathbf{d}_j$ of the dictionary $\mathbf{D}$. The column s_p (first iteration $p = 1$) of D which has maximum rank-one projection with r_0 is calculated by:

$$s_p = arg_{j=1,\cdots,K, j \notin I_{p-1}} \max \frac{|\mathbf{d}_j^h r_{p-1}|^2}{\| d_j \|^2} \tag{3.14}$$

where I_{p-1} is the index set of all the previous iterations.
The $\widehat{x}$, which is associated with $\mathbf{d}_{s_p}$, is calculated as:

$$\widehat{x}_p = \frac{\mathbf{d}_{s_p}^h r_{p-1}}{\| \mathbf{d}_{s_p} \|^2} \tag{3.15}$$

Then the residual vector need to be updated.

$$r_p = r_{p-1} - \widehat{x}_p \mathbf{d}_{s_p} \tag{3.16}$$

with $r_0 = y$. The Eq.3.14 and Eq.3.15 with $\mathbf{d}_j^h r_p = b_{p-1,j}$. So the $b_{p,j}$ can be calculated as:

$$b_{p,j} = b_{p-1,j} - \frac{\mathbf{d}_j^h \mathbf{d}_{s_p}}{\| \mathbf{d}_{s_p} \|^2} b_{p-1,s_p} = b_{p-1,j} - \widehat{x}_p \mathbf{d}_j^h \mathbf{d}_{s_p} \tag{3.17}$$

The stopping criterion can be the number of iteration times or other complicated method calculating results [34].

Table 3.1: Basic matching pursuit Algorithm

Input: Dictionary D, received signal y, stopping criterion	
$\mathbf{r}_0 = \frac{\mathbf{y}}{\|\mathbf{y}\|}$	1
$b_{0,j} = \mathbf{d}_j^h \mathbf{r}_0$, for$j = 1, \cdots, K$	2
$s_1 = arg\max_{j=1,\cdots,K} \frac{\|b_{0,j}\|^2}{\|\mathbf{d}_j\|^2}$	3
$I_1 = \{s_1\}$	4
$\widehat{x}_1 = \frac{b_{0,s_1}}{\|\mathbf{d}_{s_1}\|^2}$	5
$b_{1,j} = b_{0,j} - \widehat{x}_1 d_j^h \mathbf{d}_{s_1}$ for $j = 1, \cdots, K, j \notin I_1$	6
the pth iteration, $p > 1$	7
$s_p = arg\max_{j=1,\cdots,K, j\notin I_{p-1}} \frac{\|b_{p-1,j}\|^2}{\|\mathbf{d}_j\|^2}$	8
$I_p = \{I_{p-1}, s_p\}$	9
$\widehat{x}_p = \frac{b_{p-1,s_p}}{\|\mathbf{d}_{s_p}\|^2}$	10
$b_{p,j} = b_{p-1,j} - \widehat{x}_p \mathbf{d}_j^h \mathbf{d}_{s_p}$ for $j = 1, \cdots, K, j \notin I_p$	11
Output:$I_p, \widehat{x}_p$	

The outputs include the selected columns number and the corresponding $\widehat{x}$.

3.3.1.3 Orthogonal Matching Pursuit

Another widely used matching pursuit method is Orthogonal Matching Pursuit(OMP). When using Orthogonal Matching Pursuit(OMP) algorithm, the dictionary structure is the same as the Basic Matching Pursuit(BMP) algorithm. The basic matching pursuit algorithm is based on the assumption that different columns of the dictionary are orthogonal to each other. While this orthogonality between the selected columns can not always be guaranteed. When the selected columns are not orthogonal, the value calculated by the Eq.3.15 will not be correct. Consequently the residual vector obtained form the Eq.3.16 will not be minimum. In order to solve this issue of BMP, the OMP algorithm has been proposed [30].

$$\begin{aligned}\widehat{x}_p &= \arg\min \parallel y - \mathbf{D}_{s_p} x \parallel^2 \\ &= \left[\mathbf{D}_{s_p}^h \mathbf{D}_{s_p}\right]^{-1} \mathbf{D}_{s_p}^h y\end{aligned} \tag{3.18}$$

the Eq.3.18 is the LS minimization based on the atoms selected form each iteration. Where $\mathbf{D}_{s_p} = \left[\mathbf{d}_{s_1} \cdots \mathbf{d}_{s_p}\right]$. The main difference between BMP and OMP is the difference of their $\widehat{x}$ calculation algorithms are different. The implementation part will apply the OMP algorithm.

3.3.1.4 Stopping Criteria

When applying matching pursuit algorithm, the stopping criterion is critical. The stopping criterion decides the iteration times and then decides the accuracy of the searching result. It can be a limit to the number of iterations or the requirement that $y \approx Dx$ [17]. During this thesis implementation, we assume the number of paths is known, and the iterative threshold is set based on the number of paths.

3.3.2 Doppler estimation

The channel characters have been analysed at the beginning of this chapter. It is known that Doppler shift could cause time-dependant phase shift in the time domain. The theory of the Doppler estimation in the time domain is based on this fact. While lacking reference signal in the time domain, it brings great challenge when trying to estimate Doppler shift in the time domain.

3.3.2.1 Doppler estimation in time domain

After the delay elements are found, the delays caused channel matrix can be constructed and then converted to the time domain through IFFT transformation. The Doppler effect dictionary can be build in a similar way to the delay elements dictionary.
The disadvantages of Doppler estimation in the time domain are:

- Without reference signal in the time domain. The three kinds of pilots (scattered pilots, continual pilots and TPS pilots) are inserted in the frequency domain.
- When estimating Doppler shift in time domain, the delay matrix should also be considered. After finishing the delay elements search, the delays channel matrix

in frequency needs to be changed to time domain through IFFT firstly, which increases the calculation complexity.

- If using IFFT of frequency domain pilots as the reference signal, which means use all the carriers, the dimension of the dictionary would be too high.
- The effect of Inter Carrier Interference (ICI) can not be avoided.
- The estimated Doppler effect result is not accurate, which is a combination of multipath Doppler shift.
- The equalisation needs to be finished in the frequency domain, which means the result need to be recovered to frequency again through FFT transform.

Due to the lake of pilots in the time domain for DVB-T data frame and the high complexity of this approach, this scheme is not suitable for DVB-T standard channel Doppler shift estimation.

3.3.2.2 Doppler estimation in frequency domain

It is impractical to figure out the Doppler effect in the time domain. Another option is to evaluate the Doppler effect in the frequency domain.

Doppler effects in frequency domain characters

The Delay-Doppler spreading function in the time domain can be expressed as Eq.3.4. After IFFT, it will be as Eq.3.8. The part of Doppler shift diagonal in the frequency domain is:

$$\mathbf{F}\Lambda_{T,l}\mathbf{F}^H = \mathbf{C}_{F,l} \tag{3.19}$$

The Doppler shift diagonal in the frequency domain will be a circular matrix.

We assume that the delays are estimated exactly through matching pursuit method. Thus phase progression model of each path can be reconstructed. The basic idea of estimation Doppler in the frequency is to estimate this matrix path by path. For each path, the value of weights within a short length is assumed to be constant. Each short length is a piece. The next step is to use matching pursuit algorithm to estimate each piece pilots.

Doppler dictionary

Without considering the effects of the noise, one of the main two factors has been estimated through delays estimation. And the delay effects of each path can be reconstructed. The atom Doppler dictionary is a piece of the path with different phase mode. The dimension of the Doppler dictionary is dependent on the scattered pilots

piece length and the number of the previous searched number delay elements.
For one path, the search time is:

$$t_s = \left\lceil \frac{N_s}{L_p} \right\rceil \tag{3.20}$$

where N_s is the total number of scattered pilots in the symbol. L_p is the length of the search piece.
The total search times which also depends on the number of delay elements for one OFDM symbol is:

$$t_{total} = t_s \times L \tag{3.21}$$

where L is the number of paths, which is equal to the number of founded delay elements.

Doppler estimation

For each piece, the weight can be estimated by calculating the $\hat{x}_d$ through matching pursuit algorithm.

- **BMP Doppler weights estimation**

TABLE 3.2: BMP Doppler weights estimation

Input: Dictionary D, received signal y, stopping criterion	
$\mathbf{r}_0 = \mathbf{y}$	1
$b_{0,j} = \mathbf{d}_j^h \mathbf{r}_0$, for$j = 1, \cdots, K$	2
$I_1 = \{1\}$	3
$\widehat{x}_1 = \frac{b_{0,1}}{\Vert\mathbf{d}_1\Vert^2}$	4
$b_{1,j} = b_{0,j} - \widehat{x}_1 d_j^h \mathbf{d}_1$ for $j = 1, \cdots, K, j \notin I_1$	5
the pth iteration, $p > 1$	
$I_p = \{I_{p-1}, p\}$	6
$\widehat{x}_p = \frac{b_{p-1,p}}{\Vert\mathbf{d}_p\Vert^2}$	7
$b_{p,j} = b_{p-1,j} - \widehat{x}_p \mathbf{d}_j^h \mathbf{d}_p$ for $j = 1, \cdots, K, j \notin I_p$	8
Output:$I_p, \widehat{x}$	

Unlike delay elements estimation, the Doppler estimation do not need to choose the column with maximum rank-one projection.

- **OMP Doppler weights estimation**

$$\widehat{x}_p = \left[\mathbf{D}_p^h \mathbf{D}_p\right]^{-1} \mathbf{D}_p^h y \tag{3.22}$$

When using OMP to estimate the Doppler weights, it is much more efficient.

Doppler weights estimation happens in the frequency domain. The Doppler weights are estimated piece by piece. Each piece is a certain width of carriers.
The matching pursuit delays estimation combined with Doppler weights estimation in the frequency domain was proposed in [35] and named as Rake-MP (RMP) algorithm.

Chapter 4

Channel response reconstruction and equalisation

The channel characteristics of the estimation scattered pilots sub-carriers are obtained through channel estimation by using different algorithms. The following step is to build the channel response matrix by using the channel estimation result. Finally, use the channel response matrix to equalise the received signal data.

4.1 LS estimation channel response reconstruction

By using Least Squares (LS) estimation method, the channel characters scattered pilots are calculated by Eq 3.11. The follow steps involve getting all sub-carriers channel characters by using interpolation algorithm. When the pilot interval is shorter than coherent bandwidth, after the frequency response of pilot sub-channel is estimated, interpolation is used in the frequency domain to get the channel estimation [36]. Different interpolation methods will yield different accuracy.

4.1.1 Interpolation algorithm

The widely used channel interpolation algorithms are piecewise constant interpolation, the first order linear interpolation, second order interpolation and cubic spline interpolation [37][28] .

- **Piecewise Constant Interpolation**

  The simplest estimation method is the piecewise constant interpolation. The channel is estimated by the previous pilot. And the channel estimation is given by;

$$H(k) = H(mL + l) = H(m) \qquad 0 \leq l < L, m = 0, 1, ...N_p - 1 \tag{4.1}$$

- **Linear Interpolation**

  Linear interpolation performs better than the piecewise constant interpolation. The channel estimation at the data sub-carrier is obtained by estimation of the response of two adjacent pilot sub-channels. But the precondition is the linearity of the transmitted functions of adjacent sub-channels. The linear interpolation algorithm is shown as below:

$$H(k) = H(mL+l) = \big(H(m+1) - H(m)\big)\frac{l}{L} + H(m) \qquad 0 \leq l < L, m = 0, 1, ...N_p - 1 \tag{4.2}$$

- **Second Order Interpolation**

  The second order interpolation is a high order interpolation method. The channel response at the data sub-carrier is calculated by using linear combination of three adjacent pilots:

$$\begin{aligned} &H(k) = H(mL + l) \\ &= C_1 H(m-1) + C_0 H(m) + C_{-1} H(m+1) \\ &where \begin{cases} c_1 = \frac{\alpha(\alpha-1)}{2} \\ c_0 = -(\alpha - 1)(\alpha + 1), \alpha = \frac{l}{L} \\ c_{-1} = \frac{\alpha(\alpha+1)}{2} \end{cases} \end{aligned} \tag{4.3}$$

  Theoretically, the high order interpolation should give better channel estimation because of using more pilots. However, the computational complexity is increased with the increasing of order.

- **Cubic interpolation algorithm**

  The cubic spline interpolation method produces a smooth and continuous polynomial fitted to the given data points, which is given by

$$\begin{aligned} &H(k) = H(mL + l) = \\ &\alpha_1 H(m+1) + \alpha_0 H(m) + L\alpha_1 H^{'}(m+1) - L\alpha_0 H^{'}(m) \\ &\qquad\qquad 0 \leq l < L, m = 0, 1, ...N_p - 1 \end{aligned} \tag{4.4}$$

$H^{'}(m)$ is the first order derivative of $H(m)$, and

$$\begin{cases} \alpha_1 = \frac{3(L-l))^2}{L^2} - \frac{2(L-l)^3}{L^3} \\ \alpha_0 = \frac{3l^2}{L^2} - \frac{2l^3}{L^3} \end{cases} \tag{4.5}$$

4.2 Matching pursuit channel response reconstruction

The Delay-Doppler spreading function Eq.3.4 has been discussed in Chapter 3. The Doppler effect and multipath propagation have been mathematically analysed. Because there are only reference signal in the frequency domain. The channel estimation is performed in the frequency. So the channel response matrix will be constructed in the frequency domain.

4.2.1 Delays channel matrix

Assume the channel estimator got n paths signal, the corresponding delays and $\hat{X}$ are $\tau_1 \cdots \tau_n$ and $\hat{X}_1 \cdots \hat{X}_n$. Without considering other effects, the channel matrix can be built as following. The first step is to build the phase index for each path, as similar as the delay elements dictionary columns construction, which can be written as:

$$P_n(i) = e^{-j2\pi\left(i-\frac{N_{fft}}{2}\right)\times carrierspacing\times\tau_n} \tag{4.6}$$

Where i is the channel index, the range is $1 \cdots N_{fft}$ The final channel matrix is the combination of the each path phase index after multiplying the corresponding $\hat{x}$, which will be :

$$H(i) = \sum_{1}^{n} \hat{X}_n * e^{-j2\pi\left(i-\frac{N}{2}\right)\times carrierspacing\times\tau_n} \tag{4.7}$$

4.2.2 RMP channel matrix

The Rake-MP (RMP) algorithm performs Doppler scales estimation in the frequency domain after getting the delay taps. Applying the Doppler scales estimation method, the $\hat{X}$ for each path was estimated piece by piece, for each path the channel response is:

$$H(i) = \sum_{1}^{M} \hat{X}_m * e^{-j2\pi\left(i-\frac{N}{2}\right)\times carrierspacing\times\tau_n} \tag{4.8}$$

Where $\hat{X}_m$ is the Doppler scale weight of m^{th} piece. The final channel response is the combination of all the paths, which will be:

$$H(i) = \sum_{1}^{n} \sum_{1}^{M} \hat{X}_{n_m} * e^{-j2\pi\left(i-\frac{N}{2}\right)\times carrierspacing \times \tau_n} \tag{4.9}$$

where $\hat{X}_{n_m}$ represents the $\hat{X}$ for n^{th} path m^{th} piece.

4.3 Channel equalisation

In order to improve the quality of the received signal , it is necessary to perform the equalisation. In this section, the widely used equalisation methods, one tap equaliser and Minimum Mean Square Error (MMSE) equaliser, will be discussed [38].

4.3.1 One tap Equaliser

The one tap equaliser applies the inverse of the frequency response of the channel [29]. For a channel with frequency response $H(f)$, the zero forcing equaliser $E(f)$ can be constructed as:

$$E(f) = \frac{1}{H(f)} \tag{4.10}$$

The one tap equaliser is simple and can be easily adopted.

4.3.2 MMSE equaliser

The Minimum Mean Square Error (MMSE) equaliser is a method based on the minimization Mean Square Error (MSE) $E[|e|^2]$, where e is the error signal, which is calculated by subtracting the transmitted signal from the received signal [39]. The MSE can be written as:

$$\begin{aligned} MME = &\|Y - H_T X\|^2 = (Y - H_T X)^H (Y - H_T X) \\ &= Y^H Y - Y^H H_T X - H_T^H X^H Y + H_T^H X^H H_T X \end{aligned} \tag{4.11}$$

In order to minimize the MSE, the Eq. 4.11 take the derivative with respect to H_T, which can be written as:

$$\frac{\partial MSE}{\partial H_T} = -2H_T^H (Y - H_T X) = 0 \tag{4.12}$$

So the X can be written as:

$$X = H_T^H (H_T H)^{-1} Y \tag{4.13}$$

After adding a small ridge, Eq. 4.13 will be as:

$$X = H_T^H (H_T H + \Lambda(\frac{N}{SNR}))^{-1} Y \tag{4.14}$$

Where $\Lambda(\frac{N}{SNR})$denotes a diagonal matrix with all the diagonal elements equal to SNR. The complexity of MMSE equaliser is much higher than the Zero Forcing Equaliser.

4.3.3 Equalisation algorithm selection

Comparing to one tap equaliser, MMSE equaliser is more complex. It also requires the value of SNR. It is hard to exactly estimate the SNR value in DVB-T receiver. Another important reason is that the final channel response is a vector which can be viewed as a diagonal matrix. So when applying MMSE equaliser, it has the same performance with the one tap equaliser. During the simulation section, one tap equaliser will be applied.

Chapter 5

Implementation and performance analysis

In this chapter, the channel estimation and equalisation algorithms are implemented in the GNU Radio DVB-T. The performance of different algorithms will be analysed. When applying to the hardware, another essential factor is the complexity. The complexity of different methods will be also discussed in this chapter.

5.1 Implementation

The Tab.5.1 shows the simulation parameters, the applying mode is $8K$. There are 8192 carriers, which contains 6818 useful data carriers, others are guard band.

Table 5.1: GNU Radio DVB-T simulation parameters

Transmission mode	8K mode
NFFT length	8192
Guard Interval	1/32
Number of scattered pilots	525
Duration time T_u	896us
Carrier spacing $1/T_u$	1116 Hz
Modulation scheme	QPSK

The choosing modulation scheme is QPSK. The carrier spacing is 1116 Hz. The reference signal is the scattered pilots carried data. There are 525 scattered pilots in each OFDM symbol.

5.1.1 GNU Radio DVB-T

The Fig.5.1 show the GNU Radio DVB-T transmitter. In chapter 2, the Fig.2.4 has shown the functional block diagram of the DVB-T transmitter. The below figure is the implementation of the functional block diagram.

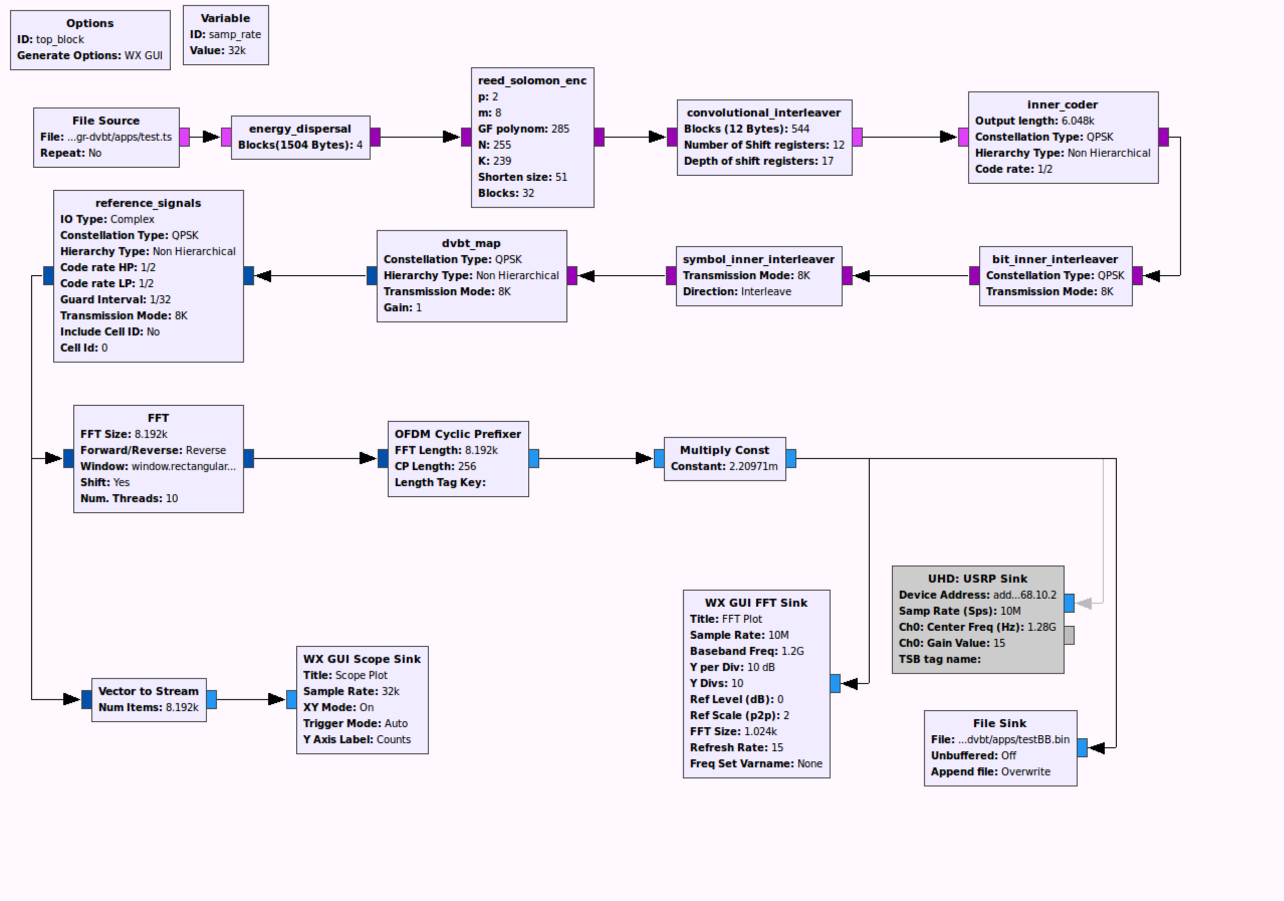

FIGURE 5.1: GNU Radio DVB-T transmitter chain

The GNU Radio DVB-T receiver is shown as Fig.5.2. The new block, **omp_estimator** is added between FFT block and de_mod block. The steps can be found in Appendix A. The **omp_estimator** block contains different channel estimation and equalisation algorithms, which can be chosen manually.

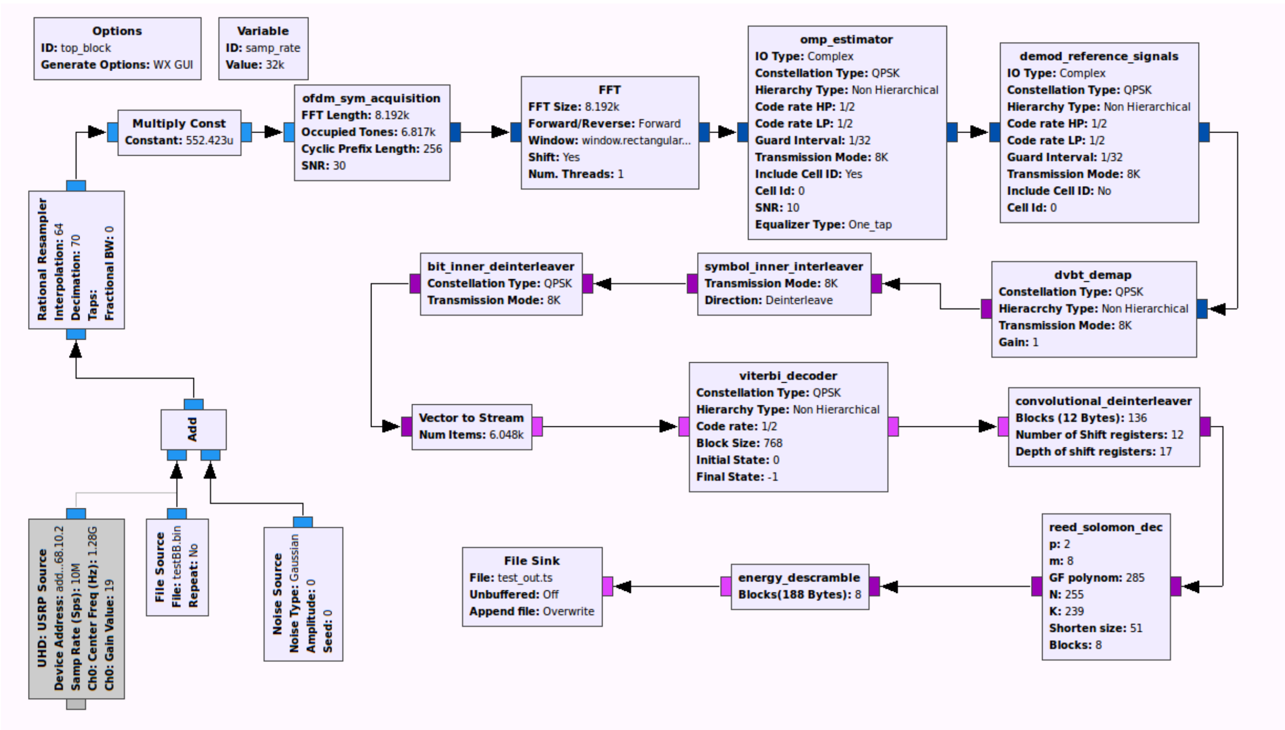

FIGURE 5.2: GNU Radio DVB-T receiver chain

The estimation and equalisation implement after FFT. The equalisation performance will be evaluated after **demap** block.

5.2 Channel tracking performance

The channel tracking performance will be analysed by comparing the channel responses constructed by different estimation algorithms. The channel tracking performance shows the performance of the quality of channel estimation. The performance is analysed based on Matlab platform. The channel is simulated by Matlab Rayleigh channel model since it is easy to analyse. The performance of LS channel estimation combined with four different interpolation algorithms are analysed separately. Then the performance of LS channel estimation is compared with matching pursuit method and RMP estimation. If channel response curve close to the Rayleigh channel model response curve, the performance is better.

5.2.1 LS estimation

The Fig.5.3 shows the channel tracking performance of LS estimation combined with four different interpolation algorithms (Piecewise Constant Interpolation, Linear Interpolation, Second Order Interpolation, Cubic Interpolation).

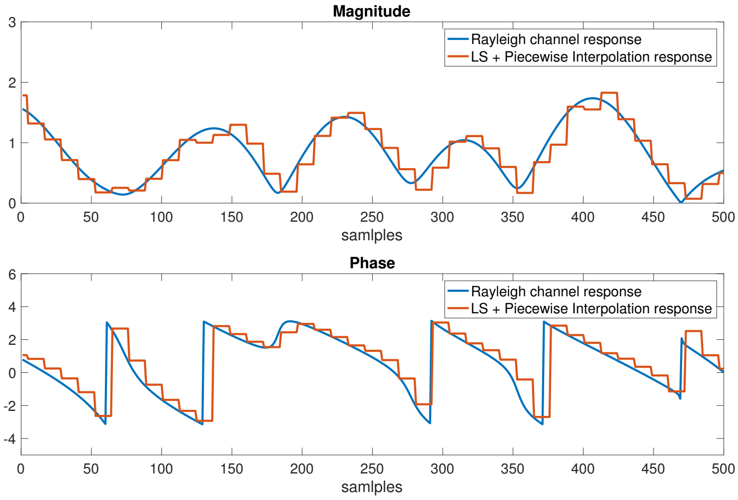

FIGURE 5.3: LS + Piecewise Constant Interpolation

As Fig.5.3 shows, the performance of LS channel estimation combined with piecewise constant interpolation algorithm can not fit the actual channel response tightly. Because it assumes that the channel gains of the 11 carriers after scattered pilot are constant. This interpolation will not be applied for the later equalisation section.

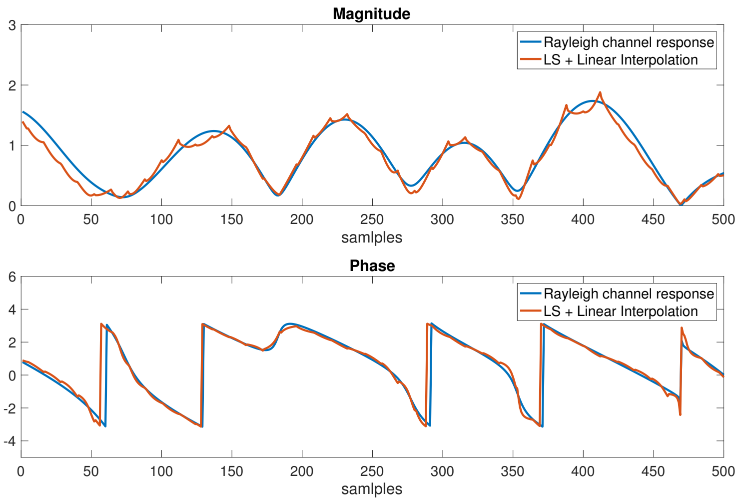

FIGURE 5.4: LS + Linear Interpolation

The Fig.5.4 is the results of LS channel estimation combined with linear interpolation. The performance is much better than piecewise constant interpolation algorithm, in terms of both the phase and the magnitude change.

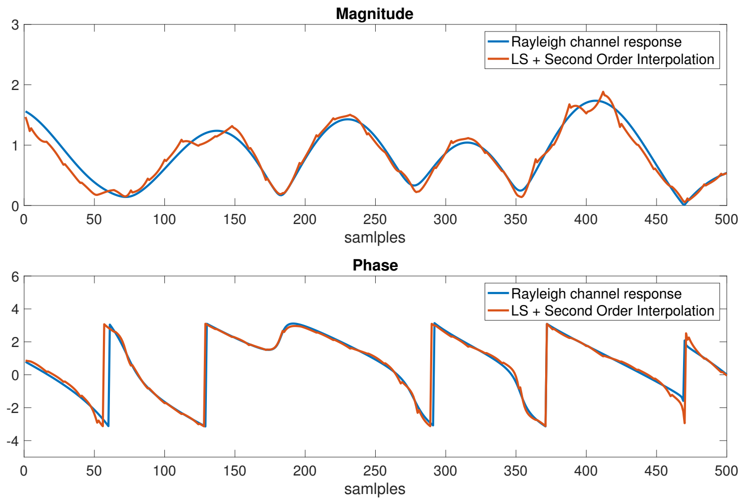

FIGURE 5.5: LS + Second Order Interpolation

As Fig.5.5 shows, the performance of LS channel estimation combined with second order interpolation algorithm. The performance is a little better. But the advantage is not obvious.

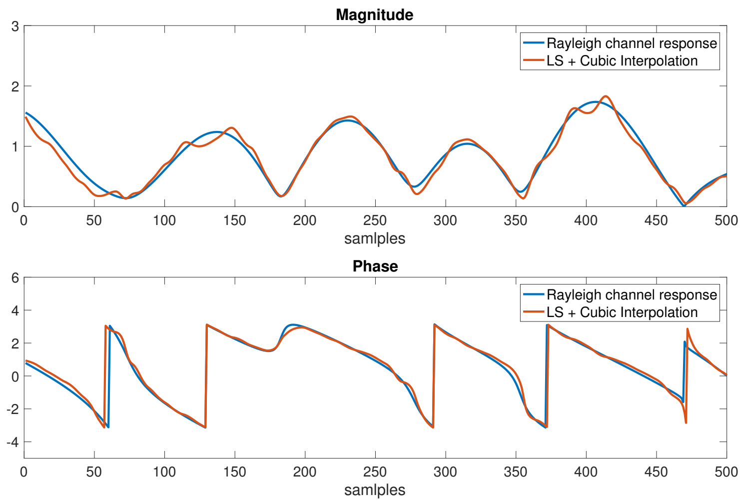

FIGURE 5.6: LS + Cubic Interpolation

The Fig.5.6 shows the performance of LS channel estimation combined with Cubic Interpolation interpolation algorithm. The performance is improved on the peaks. While the computational complexity is much more high.

After comparing the performance of different interpolation algorithms, we found linear interpolation can fit the Rayleigh channel response closely and its computational complexity is not high. The linear interpolation algorithm will be adopted for the channel response construction after LS estimation.

5.2.2 Matching pursuit methods

After confirming the interpolation algorithm for LS estimation, the performance of the matching pursuit algorithm will be analysed and compared with that of LS estimation method.

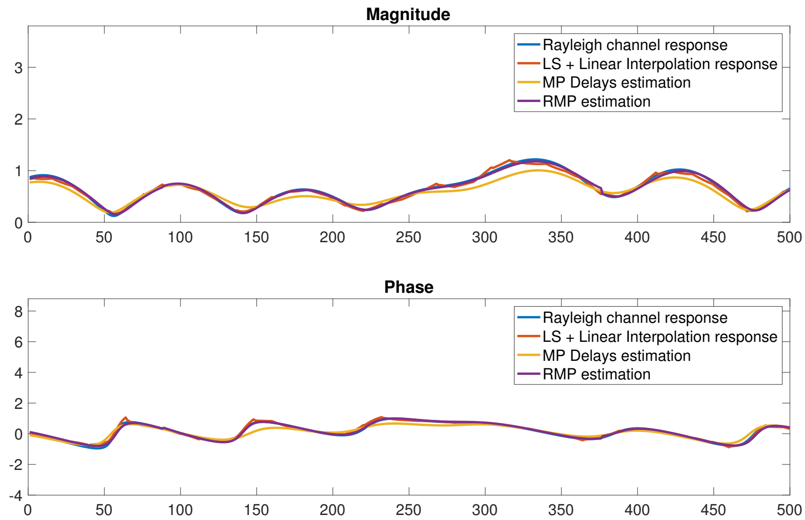

FIGURE 5.7: Channel tracking performance, 6 paths, $maximumDoppler = 0.001\% * carrierspacing$

The Fig.5.7 shows different algorithms' channel tracking performance under 6 path and very low (almost without) Doppler shift. For the phase and magnitude, the gap among three methods is very small.

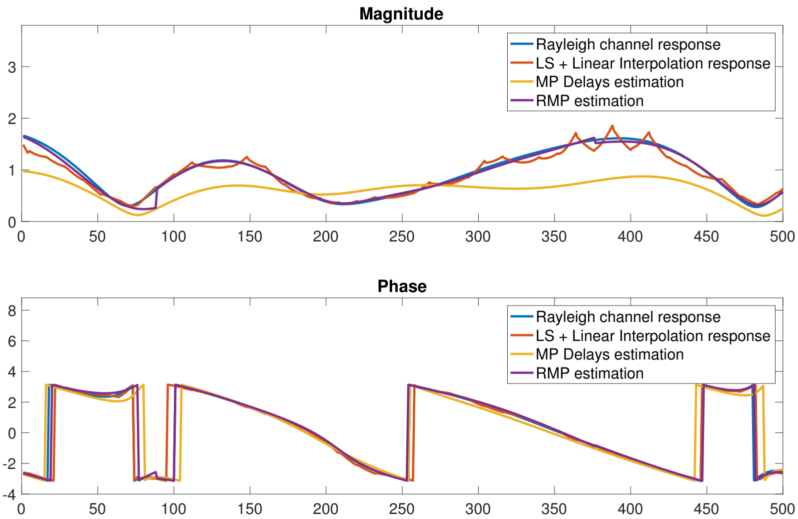

FIGURE 5.8: Channel tracking performance, 6 paths, $maximumDoppler = 1.0\% * carrierspacing$

As Fig.5.8, under 6 paths and $maximumDoppler = 1.0\% * carrierspacing$ circumstance, RMP estimation has significantly better performance. The Doppler equalisation is necessary.

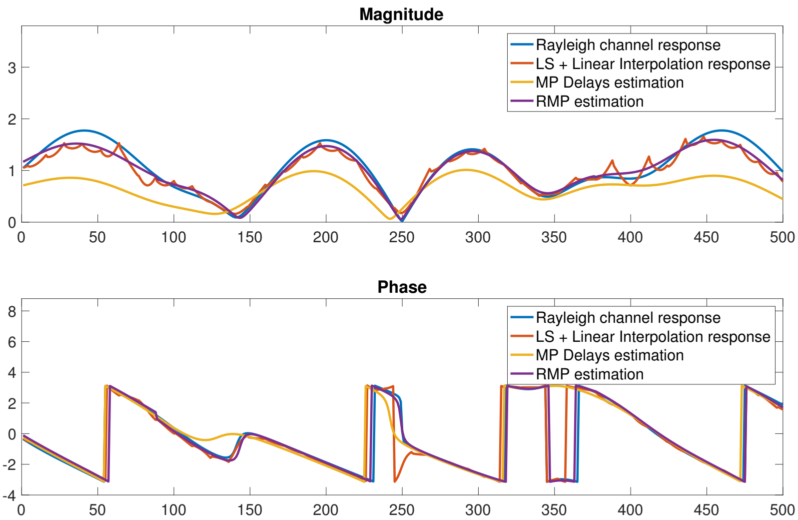

FIGURE 5.9: Channel tracking performance, 6 paths, $maximumDoppler = 10.0\% * carrierspacing$

If the Doppler shift is increased to higher, $maximumDoppler = 10.0\% * carrierspacing$, the RMP also shows better performance in terms of both the phase and the magnitude tracking.

5.3 Equalisation performance

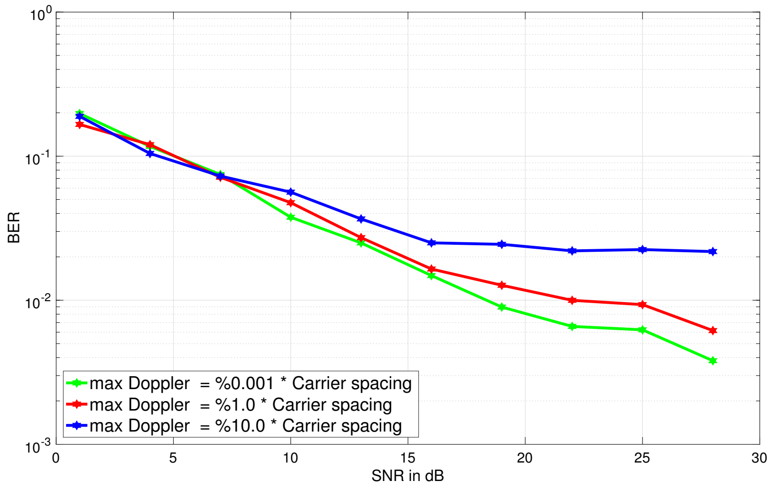

FIGURE 5.10: Equalisation performance, 6 paths, LS algorithm under different Doppler shift

The Fig.5.10 shows the equalisation performance of LS algorithm under different Doppler shift. As the increasing of the Doppler shift, the bit error rate (BER) becomes higher.

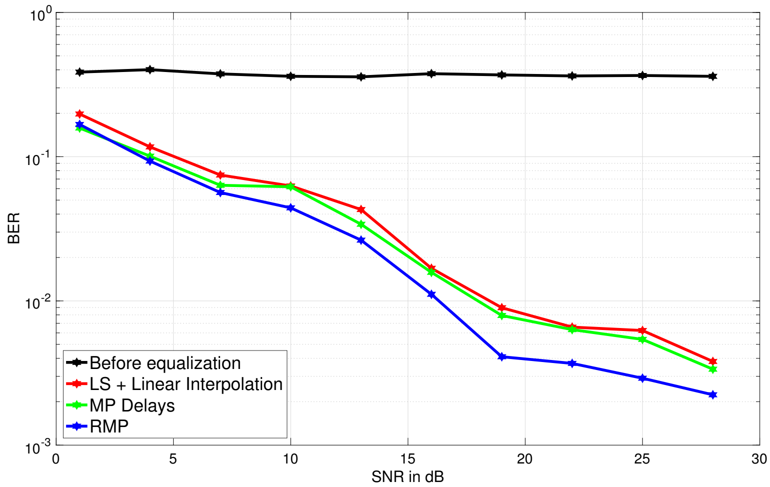

FIGURE 5.11: Equalisation performance, 6 paths, $maximumDoppler = 0.001\% * carrierspacing$

Fig.5.11 shows the equalisation performance of different algorithms under 6 path and very low Doppler shift. Due to the very low Doppler shift, close to zero, the matching pursuit delays estimation method show a little better performance than LS (almost close), but still slightly worse than RMP estimation performance.

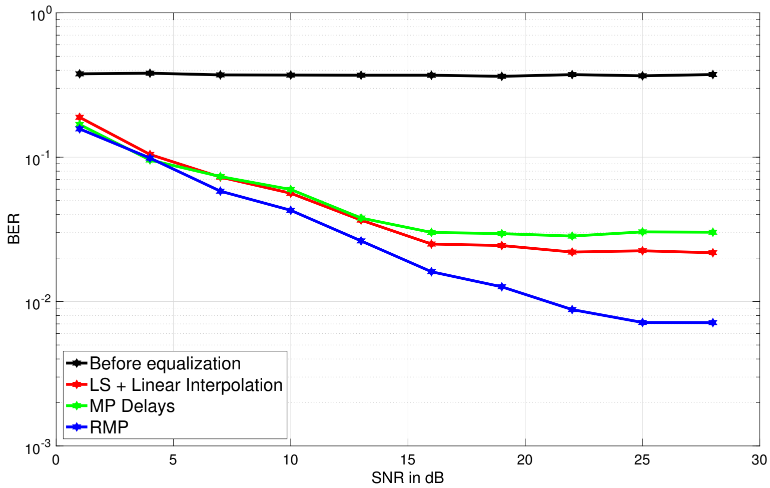

FIGURE 5.12: Equalisation performance, 6 paths, $maximumDoppler = 10\% * carrierspacing$

Fig.5.12 shows the BER curves of under high Doppler shift and 6 path circumstance. Because the Doppler effect is not taken into consideration, the matching pursuit delays estimation shows the worst performance among these three methods. The Doppler effect estimation is necessary. The RMP method shows obvious advantage.

After comparing performance under multi-paths and different Doppler shift range, we found that the gap among different algorithms is small if the Doppler is very low. As the increasing of the Doppler, the advantages of the RMP method are more obvious.

5.4 Complexity analysis

The complexity of the estimation algorithm is another important issue. In order to guarantee the efficiency and the fluency signal of the signal receiving, the complexity of estimation must be lower when applying to the hardware.
The LS estimation needs to estimate each OFDM symbol channel response, the computational complexity of the channel estimation for each OFDM symbol will be $\mathcal{O}(N)$. N is the number of carriers in each OFDM symbol. In DVB-T, an OFDM frame contains 68 OFDM symbol. The computational complexity for each OFDM frame will be $\mathcal{O}(MN)$. M is the number of OFDM symbol in each OFDM frame.

If only estimate the delays by using orthogonal matching pursuit algorithm, we assume that in each OFDM frame the delays are constant. The computational complexity will be $\mathcal{O}\left(KLN\right)$. K is the delay range. L is the number of paths. Although it involves dictionary building and its dimension may be high, which depends on the delay range. The delays just need to be estimated once. For each OFDM frame, the computational complexity still is $\mathcal{O}\left(KLN\right)$.

If the Doppler shift is high, the RMP estimation has to be applied. For each OFDM symbol, the Doppler estimation computational complexity will be $\mathcal{O}\left(LN\right)$. For each OFDM frame, the computational complexity will be $\mathcal{O}\left(KLN + LNM\right)$.

Compared to the Doppler estimation in the time domain, the complexity of Doppler estimation in the frequency domain is much lower. First, the Doppler estimation in the frequency domain will not perform IFFT. In addition, the complexity of OMP in the time domain will be $\mathcal{O}(QLN)$, where Q is the total number of possible Doppler shifts.

Chapter 6

Conclusion

The Doppler shift and the Multipath propagation are the main factors affecting the signal transmission through wireless channel. In order to clearly estimate the Doppler shift and multipath propagation effect, the channel characteristics are mathematically analysed.

The Digital Video Broadcasting-Terrestrial (DVB-T) standard is the basic standard and has been widely used for television signal transmission. The different channel estimation algorithms (LS estimation, matching pursuit delays estimation and RMP estimation) based on DVB-T standard are discussed. The matching pursuit delays combined with Doppler estimation in the frequency domain (RMP) is mainly analysed. Then different equalisation algorithms are discussed and confirm that the one tap equaliser is adopted.

The simulation is based on GNU Radio DVB-T project, which implements the DVB-T transmitter and receiver. A new block, which contains the channel estimation algorithms and equalisation algorithms, is added to the GNU Radio DVB-T.

The performance is analysed on Matlab. The channel is simulated by Matlab Rayleigh channel. Because it is easy to build the real channel response and then convenient to track the channel and compare the performance. Compared to other interpolation algorithms, LS estimation combined with linear interpolation has a good performance in terms of both for the phase and the magnitude tracking. Its computational complexity is lower compared to second order and Cubic interpolation algorithms.

Then received data under different circumstance is equalised by using different algorithms. Under multi-paths and very low Doppler shift condition, we found that matching pursuit delays estimation almost has a slightly better performance than LS estimation but worse than that of RMP estimation in the frequency domain. If increasing the Doppler shift, the advantages of the RMP method are more obvious. Finally, the complexity of different methods is analysed. The RMP in the frequency domain has a better performance and its complexity is not high compared to the traditional method (the Doppler shift is estimated in the time domain).

Appendix A

Creating a new GNU Radio module

An out-of-tree (OOT) module is a GNU Radio component that does not exist within the GNU Radio source tree [12].

- The first step is to create a new module,which is name as modname.
  gr_modtool newmod modname
  There will be a new directory called gr-modname.
  Structure of a module. All files written in C++ is put into lib/. Python stuff goes into the python/ directory.

  For documentation, docs/ contains some instructions on how to extract documentation from the C++ files and Python files.

  The apps/ subdir contains any complete applications (both for GRC and standalone executables) which are installed to the system alongside with the blocks.

  The directory, examples/ can be used to save (guess what) examples, which are a great addendum to documentation, because other developers can simply look straight at the code to see how your blocks are used.

  The build system brings some baggage along, as well: the CMakeLists.txt file (one of which is present in every subdirectory) and the cmake/ folder. You can ignore the latter for now, as it brings along mainly instructions for CMake on how to find GNU Radio libraries etc. The CMakeLists.txt files need to be edited a lot in order to make sure your module builds correctly.

- Writing a block in C++
  $\$gr\_modtool\ add - t\ generalfunction1$

On the command line, a block is adding, its type is 'general' and it is called function1. $gr_modtool$ then asks you if your block takes any arguments (it doesn't, so we leave that empty), whether or not we want QA code for Python (yes, we do) and for C++ (no, we don't right now).

- Flesh out the XML file. The .xml provides the user interface between the OOT module displayed in the GRC and the source code. Moreover, the XML file defines an interface to pass the parameters specific for the module. Hence, to access the module inside GRC, it is important to modify the .xml files manually.

- Install the block in grc.
  After finished the implementation of our block, now it's important use its functionality under GRC. So we build our OOT and install the underlying blocks. To do so, we need to execute the following commands:
  $*mkdir build.*
  $*cd build.*
  $*cmake.*

The creating a new GNU Radio module steps are from GNU Radio website [12].

Appendix B

Signals in DVB-T

B.1 Duration of symbol part for the allowed guard intervals

Mode	8K mode				2K mode			
Guard interval Δ / T_U	1/4	1/8	1/16	1/32	1/4	1/8	1/16	1/32
Duration of symbol part T_U	8 192 × T 896 μs (see note)				2 048 × T 224 μs (see note)			
Duration of guard interval Δ	2 048 × T 224 μs	1 024 × T 112 μs	512 × T 56 μs	256 × T 28 μs	512 × T 56 μs	256 × T 28 μs	128 × T 14 μs	64 × T 7 μs
Symbol duration $T_S = \Delta + T_U$	10 240 × T 1 120 μs	9 216 × T 1 008 μs	8 704 × T 952 μs	8 448 × T 924 μs	2 560 × T 280 μs	2 304 × T 252 μs	2 176 × T 238 μs	2 112 × T 231 μs

Figure B.1: Duration of symbol part for the allowed guard intervals for 8 MHz channels

[9]

B.2 PBRS sequence generation

The PRBS sequence is generated according to Fig.B.2. The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carrier.

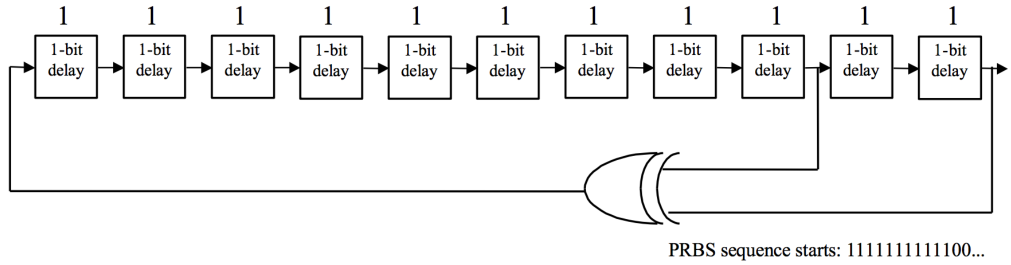

FIGURE B.2: Generation of PRBS sequence
[1]

The polynomial for the Pseudo Random Binary Sequence (PRBS) generator shall be:

$$x^{11} + x^2 + 1 \tag{B.1}$$

The PRBS is initialized so that the first output bit from the PRBS coincides with the first active carrier. A new value is generated by the PRBS on every used carrier (whether or not it is a pilot) [9].

B.3 Continual pilot carriers

There are 177 continual pilots in the 8K mode and 45 in the 2K mode, are inserted according to blow table.

Continual pilot carrier positions (index number k)													
2K mode							8K mode						
0	48	54	87	141	156	192	0	48	54	87	141	156	192
201	255	279	282	333	432	450	201	255	279	282	333	432	450
483	525	531	618	636	714	759	483	525	531	618	636	714	759
765	780	804	873	888	918	939	765	780	804	873	888	918	939
942	969	984	1 050	1 101	1 107	1 110	942	969	984	1 050	1 101	1 107	1 110
1 137	1 140	1 146	1 206	1 269	1 323	1 377	1 137	1 140	1 146	1 206	1 269	1 323	1 377
1 491	1 683	1 704					1 491	1 683	1 704	1 752	1 758	1 791	1 845
							1 860	1 896	1 905	1 959	1 983	1 986	2 037
							2 136	2 154	2 187	2 229	2 235	2 322	2 340
							2 418	2 463	2 469	2 484	2 508	2 577	2 592
							2 622	2 643	2 646	2 673	2 688	2 754	2 805
							2 811	2 814	2 841	2 844	2 850	2 910	2 973
							3 027	3 081	3 195	3 387	3 408	3 456	3 462
							3 495	3 549	3 564	3 600	3 609	3 663	3 687
							3 690	3 741	3 840	3 858	3 891	3 933	3 939
							4 026	4 044	4 122	4 167	4 173	4 188	4 212
							4 281	4 296	4 326	4 347	4 350	4 377	4 392
							4 458	4 509	4 515	4 518	4 545	4 548	4 554
							4 614	4 677	4 731	4 785	4 899	5 091	5 112
							5 160	5 166	5 199	5 253	5 268	5 304	5 313
							5 367	5 391	5 394	5 445	5 544	5 562	5 595
							5 637	5 643	5 730	5 748	5 826	5 871	5 877
							5 892	5 916	5 985	6 000	6 030	6 051	6 054
							6 081	6 096	6 162	6 213	6 219	6 222	6 249
							6 252	6 258	6 318	6 381	6 435	6 489	6 603
							6 795	6 816					

FIGURE B.3: Carrier indices for continual pilot carriers
[1]

B.4 TPS pilot carriers

The TPS is transmitted in parallel on 17 TPS carriers for the 2K mode and on 68 carriers for the 8K mode. Every TPS carrier in the same symbol conveys the same differentially encoded information bit. The following carrier indices contain TPS carriers:

2K mode					8K mode							
34	50	209	346	413	34	50	209	346	413	569	595	688
569	595	688	790	901	790	901	1 073	1 219	1 262	1 286	1 469	1 594
1 073	1 219	1 262	1 286	1 469	1 687	1 738	1 754	1 913	2 050	2 117	2 273	2 299
1 594	1 687				2 392	2 494	2 605	2 777	2 923	2 966	2 990	3 173
					3 298	3 391	3 442	3 458	3 617	3 754	3 821	3 977
					4 003	4 096	4 198	4 309	4 481	4 627	4 670	4 694
					4 877	5 002	5 095	5 146	5 162	5 321	5 458	5 525
					5 681	5 707	5 800	5 902	6 013	6 185	6 331	6 374
					6 398	6 581	6 706	6 799				

FIGURE B.4: Carrier indices for TPS carriers
[1]

References

[1] European Telecommunications Standards Institute. “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television”. In: *European Standard (Telecommunications series)* ETSI EN 300 744 V1.6.1 (2009). URL: http://www.dvb.org/.

[2] DVB Project. URL: https://www.dvb.org.

[3] Richard van Nee and Ramjee Prasad. *OFDM for Wireless Multimedia Communications*. 1st. Norwood, MA, USA: Artech House, Inc., 2000. ISBN: 0890065306.

[4] A. Mattsson. “Single frequency networks in DTV”. In: *IEEE Transactions on Broadcasting* 51.4 (2005), pp. 413–422. ISSN: 0018-9316. DOI: 10.1109/TBC.2005.858419.

[5] Jie Huang, Letizia Lo Presti, and Roberto Garello. “Digital video broadcast-terrestrial (DVB-T) single frequency networks positioning in dynamic scenarios”. In: *Sensors* 13.8 (2013), pp. 10191–10218.

[6] Thorsten Herfet. *Telecommunications 1: Digital Transmission and Signal Processing*. Telecommunications Lab, Universitat des Saarlandes, 2016.

[7] Douglas Haider. URL: https://wifijedi.com/tag/intersymbol-interference/.

[8] Thorsten Herfet. *Telecommunications 2: Audio/Visual Communications and Networks*. Telecommunications Lab, Universitat des Saarlandes, 2016.

[9] European Telecommunications Standards Institute. “Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T)”. In: *European Standard (Telecommunications series)* DVB BlueBook A012 (2015). URL: http://www.dvb.org/.

[10] *DVB-T implementation in GNUradio*. URL: https://yo3iiu.ro/blog/?p=1220.

[11] *DVB-T2 - 2nd Generation Terrestrial Broadcasting*. European Telecommunications Standards Institute, 2016.

[12] GNU Radio Website. URL: http://www.gnuradio.org.

[13] Marcos Majo Boter. "Design and implementation of an OFDM-based communication system for the GNU Radio platform". In: (2011).

[14] V Pellegrini, G Bacci, and M Luise. "Soft-DVB: A Fully-Software GNURadio-Based ETSI DVB-T Modulator, Dip". In: *Ingegneria dellInformazione, University of Pisa* (2008).

[15] B. Diaconescu. *gr-dvbt. @ONLINE*. URL: https://github.com/BogdanDIA/gr-dvbt.

[16] B. Diaconescu. *DVB-T implementation in GNUradio*. URL: https://yo3iiu.ro/blog/?p=1191.

[17] Mark A. Davenport, Marco F. Duarte, Yonina C. Eldar, and Gitta Kutyniok. *Introduction to Compressed Sensing*.

[18] Richard G Baraniuk. "Compressive sensing". In: *IEEE signal processing magazine* 24.4 (2007).

[19] M Davenport. "The fundamentals of compressive sensing". In: *IEEE Signal Processing Society Online Tutorial Library* (2013).

[20] Joel A Tropp and Anna C Gilbert. "Signal recovery from random measurements via orthogonal matching pursuit". In: *IEEE Transactions on information theory* 53.12 (2007), pp. 4655–4666.

[21] Tong Zhang. "Sparse recovery with orthogonal matching pursuit under RIP". In: *IEEE Transactions on Information Theory* 57.9 (2011), pp. 6215–6221.

[22] D Donoho and Y Tsaig. "Fast solution of l1-norm minimization problems when the solution may be sparse, 2006". In: *Preprint* 1.2 ().

[23] Philip Breen. "Algorithms for sparse approximation". In: *School of Mathematics, University of Edinburgh, Year* 4 (2009).

[24] Kelvin Chelli. "Doppler Shift Compensation in Vehicular Communication Systems". In: *Master Thesis* (2015).

[25] Nasimi Eldarov, Guoping Tan, and Thorsten Herfet. "Delay-Doppler search for matching pursuit algorithms in time-variant channels". In: *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE. 2015, pp. 1–5.

[26] Kelvin Chelli. "Doppler Shift Compensation in Vehicular Communication Systems". In: *International Conference on Computer and Communications (ICCC)* (2016).

[27] Shiou-Hong Chen, Way-Hong He, Hou-Shin Chen, and Yumin Lee. "Mode detection, synchronization, and channel estimation for DVB-T OFDM receiver". In: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. Vol. 5. IEEE. 2003, pp. 2416–2420.

[28] Fei Wang. "Pilot-based channel estimation in OFDM system". In: (2011).

[29] A. M. Khan, V. Jeoti, and M. A. Zakariya. "Improved pilot-based LS and MMSE channel estimation using DFT for DVB-T OFDM systems". In: *2013 IEEE Symposium on Wireless Technology Applications (ISWTA)*. 2013, pp. 120–124. DOI: 10.1109/ISWTA.2013.6688752.

[30] W. Li and J. C. Preisig. "Estimation of Rapidly Time-Varying Sparse Channels". In: *IEEE Journal of Oceanic Engineering* 32.4 (2007), pp. 927–939. ISSN: 0364-9059. DOI: 10.1109/JOE.2007.906409.

[31] N. Eldarov, G. Tan, and T. Herfet. "Delay-Doppler search for matching pursuit algorithms in time-variant channels". In: *2015 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. 2015, pp. 1–5. DOI: 10.1109/BMSB.2015.7177226.

[32] C. R. Berger, Z. Wang, J. Huang, and S. Zhou. "Application of compressive sensing to sparse channel estimation". In: *IEEE Communications Magazine* 48.11 (2010), pp. 164–174. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5621984.

[33] Stéphane G Mallat and Zhifeng Zhang. "Matching pursuits with time-frequency dictionaries". In: *IEEE Transactions on signal processing* 41.12 (1993), pp. 3397–3415.

[34] Laura Rebollo-Neira and David Lowe. "Optimized orthogonal matching pursuit approach". In: *IEEE Signal Processing Letters* 9.4 (2002), pp. 137–140.

[35] Kelvin Chelli, Praharsha Sirsi, and Thorsten Herfet. "Cognitive Framework for the Estimation of Doubly Selective Channels". In: *2017 IEEE 86th Vehicular Technology Conference (VTC2017-Fall)*. Toronto, Canada, Sept. 2017.

[36] Leonard Cimini. "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing". In: *IEEE transactions on communications* 33.7 (1985), pp. 665–675.

[37] S. Coleri, M. Ergen, A. Puri, and A. Bahai. "Channel estimation techniques based on pilot arrangement in OFDM systems". In: *IEEE Transactions on Broadcasting* 48.3 (2002), pp. 223–229. ISSN: 0018-9316. DOI: 10.1109/TBC.2002.804034.

[38] AM Khan, Varun Jeoti, and MA Zakariya. "Improved pilot-based LS and MMSE channel estimation using DFT for DVB-T OFDM systems". In: *Wireless Technology and Applications (ISWTA), 2013 IEEE Symposium on*. IEEE. 2013, pp. 120–124.

[39] T. Hrycak and G. Matz. "Low-Complexity Time-Domain ICI Equalization for OFDM Communications Over Rapidly Varying Channels". In: *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*. 2006, pp. 1767–1771. DOI: 10.1109/ACSSC.2006.355065.