Extraction of radio frequency quality metric from digital video broadcast streams by cable using software defined radio

Examensarbete utfört i Kommunikationssystem vid Tekniska högskolan vid Linköpings universitet
av
Viktor Eriksson

LiTH-ISY-EX--13/4676--SE
Linköping 2013
Extraction of radio frequency quality metric from digital video broadcast streams by cable using software defined radio

Examensarbete utfört i Kommunikationssystem vid Tekniska högskolan vid Linköpings universitet av

Viktor Eriksson

LiTH-ISY-EX--13/4676--SE

Handledare:  **Anton Blad**  
ISY, Linköpings universitet  
**Aner Gusic**  
Agama Technologies

Examinator:  **Mikael Olofsson**  
ISY, Linköpings universitet

Linköping, 16 juni 2013
Title

Extraction of radio frequency quality metric from digital video broadcast streams by cable using software defined radio

Författare

Viktor Eriksson

Sammanfattning

Abstract

The purpose of this master thesis was to investigate how efficient the extraction of radiofrequency quality metrics from digital video broadcast (DVB) streams can become using software defined radio. Software defined radio (SDR) is a fairly new technology that offers you the possibility of very flexible receivers and transmitters where it is possible to upgrade the modulation and demodulation over time.

Agama is interested in SDR for use in the Agama Analyzer, a widely deployed monitoring probe running on top of standard services. Using SDR, Agama could use that in all deployments, such as DVB by cable/terrestrial/satellite (DVB-C/T/S), which would simplify logistics.

This thesis is an implementation of a SDR to be able to receive DVB-C. The SDR must perform a number of adaptive algorithms in order to prevent the received symbols from being significantly different from the transmitted ones. The main parts of the SDR include timing recovery, carrier recovery and equalization. Timing recovery performs synchronization between the transmitted and received symbols and the carrier recovery performs synchronization between the carrier wave of the transmitter and the local oscillator in the receiver. The thesis discusses various methods to perform the different types of synchronizations and equalizations in order to find the most suitable methods.

Nyckelord

Keywords QAM, SDR, MER, Carrier Recovery, Timing Recovery
Abstract

The purpose of this master thesis was to investigate how efficient the extraction of radiofrequency quality metrics from digital video broadcast (DVB) streams can become using software defined radio. Software defined radio (SDR) is a fairly new technology that offers you the possibility of very flexible receivers and transmitters where it is possible to upgrade the modulation and demodulation over time.

Agama is interested in SDR for use in the Agama Analyzer, a widely deployed monitoring probe running on top of standard services. Using SDR, Agama could use that in all deployments, such as DVB by cable/terrestrial/satellite (DVB-C/T/S), which would simplify logistics.

This thesis is an implementation of a SDR to be able to receive DVB-C. The SDR must perform a number of adaptive algorithms in order to prevent the received symbols from being significantly different from the transmitted ones. The main parts of the SDR include timing recovery, carrier recovery and equalization. Timing recovery performs synchronization between the transmitted and received symbols and the carrier recovery performs synchronization between the carrier wave of the transmitter and the local oscillator in the receiver. The thesis discusses various methods to perform the different types of synchronizations and equalizations in order to find the most suitable methods.
Acknowledgments

Working on the thesis has been difficult but educational, fun but sometimes frustrating. Throughout the project it shifted between deep valleys and high mountains at regular intervals. During the periods where I was fumbling around in the deep valleys I got great help from some very kind individuals.

First of all, I would like to thank my two mentors, Anton Blad at ISY and Aner Gusic at Agama. Without your knowledge and support, the project would have been a lot harder to accomplish. Furthermore, I am very grateful to Erik Pettersson at Agama. In the beginning, when things were tough and the hardware and software from Dektec did not really cooperate, Erik were interested and helpful. In addition, I have been privileged to get in touch with some experts in specific areas during the thesis. Jan Zöllner from the Institut für Nachrichtentechnik at Technische Universität Braunschweig helped me a lot by carefully investigating the problems I had and then gave me a few new ideas and some theory lessons.

Linköping, June 2013

Viktor Eriksson
# Contents

## Notation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>xi</td>
<td>xi</td>
</tr>
</tbody>
</table>

## I Background and Theory

### 1 Introduction

1.1 Agama Technologies .................................. 3
1.2 Software Defined Radio ................................ 3
1.3 Digital Video Broadcasting by Cable .................. 4
1.4 Research Motivation .................................. 4
1.5 Research Objectives .................................. 4
1.6 Thesis Overview ...................................... 4

## 2 Communication Systems

2.1 Modulation Overview .................................. 8
2.1.1 Quadrature Modulation .............................. 10
2.1.2 Phase Shift Keying ................................ 11
2.2 Quadrature Amplitude Modulation ...................... 13
2.2.1 Multilevel Quadrature Amplitude Modulation ........ 15
2.3 Digital QAM Transmitter ................................ 16
2.4 Digital QAM Receiver .................................. 16
2.5 Noise and Offsets ..................................... 18
2.5.1 Additive White Gaussian Noise ...................... 18
2.5.2 Quantization Noise ................................ 19
2.5.3 Channel Distortions ................................ 20
2.5.4 Carrier Offset .................................... 21
2.5.5 Timing Offset ..................................... 22

## 3 QAM Demodulation

3.1 Automatic Gain Control ............................... 25
3.2 Matched Filtering ..................................... 27
3.2.1 Pulse Shaping ..................................... 27
3.2.2 Matched Filtering .................................. 27
CONTENTS

3.3 Symbol Timing Recovery ........................................... 29
  3.3.1 Feed-forward Methods ........................................... 29
  3.3.2 Feed-back Methods ............................................. 31
3.4 Carrier Recovery ..................................................... 35
  3.4.1 Non Data Aided .................................................. 35
  3.4.2 Decision Directed .............................................. 37
3.5 Equalization .......................................................... 37
  3.5.1 Least Mean Square Algorithm .................................... 38
  3.5.2 Constant Modulus Algorithm ..................................... 39
  3.5.3 Multi Modulus Algorithm ....................................... 39

II Implementation and Results

4 Implementation .................................................................. 43
  4.1 Automatic Gain Control ............................................. 44
  4.2 Multiply Filter Divide .............................................. 44
  4.3 Root Raised Cosine Filter .......................................... 45
  4.4 Modified Gardner Algorithm for Timing Recovery ............... 46
  4.5 Multi Modulus Algorithm for Equalization ....................... 50
  4.6 Decision Directed Phase Locked Loop for Carrier Recovery and the
      LMS Equalizer .......................................................... 51
  4.7 QAM-data .................................................................. 53
    4.7.1 Synthesized QAM-data .......................................... 53
    4.7.2 QAM Modulation ................................................. 53
    4.7.3 Root Raised Cosine Filter ...................................... 54
    4.7.4 Channel Errors ................................................... 54
    4.7.5 Carrier Offsets .................................................. 55
    4.7.6 Timing Error ...................................................... 56
  4.8 Transmitting and Receiving real QAM-data ......................... 56
    4.8.1 Transmitter ....................................................... 56
    4.8.2 Receiver .......................................................... 57
  4.9 Summation .................................................................. 57

5 Results ............................................................................ 59
  5.1 Measurements ............................................................ 59
    5.1.1 Modulation Error Ratio ......................................... 59
    5.1.2 Bit Error Rate ..................................................... 60
  5.2 Testing on Synthesized Data .......................................... 61
    5.2.1 Timing Offset ..................................................... 61
    5.2.2 Carrier Offset .................................................... 63
    5.2.3 Timing Offset and Carrier Offset .............................. 65
  5.3 Testing on Real Data ................................................... 70
    5.3.1 16-QAM ............................................................ 70
    5.3.2 64-QAM ............................................................ 71
    5.3.3 256-QAM .......................................................... 72
6 Summary and Conclusions 75

6.1 Summary .............................................. 75
  6.1.1 Objectives ........................................ 75
  6.1.2 Results ........................................... 75

6.2 Further Research .............................. 75
  6.2.1 DVB-S ........................................... 76
  6.2.2 Increasing the MER ........................ 76
  6.2.3 Source Decoder ............................... 76

Bibliography .............................. 77
### Notation

#### Equation Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{R} )</td>
<td>Real part</td>
</tr>
<tr>
<td>( \mathbb{I} )</td>
<td>Imaginary part</td>
</tr>
<tr>
<td>( \bar{X} )</td>
<td>Complex conjugate</td>
</tr>
<tr>
<td>( X^T )</td>
<td>Transpose</td>
</tr>
</tbody>
</table>
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/D</td>
<td>Analog to Digital</td>
</tr>
<tr>
<td>AGC</td>
<td>Automatic Gain Control</td>
</tr>
<tr>
<td>ASK</td>
<td>Amplitude Shift Keying</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase Shift Keying</td>
</tr>
<tr>
<td>CMA</td>
<td>Constant Modulus Algorithm</td>
</tr>
<tr>
<td>D/A</td>
<td>Digital to Analog</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DVB</td>
<td>Digital Video Broadcasting</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>FSK</td>
<td>Frequency Shift Keying</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Mean Square</td>
</tr>
<tr>
<td>MER</td>
<td>Modulation Error Ratio</td>
</tr>
<tr>
<td>MMA</td>
<td>Multi Modulus Algorithm</td>
</tr>
<tr>
<td>PSK</td>
<td>Phase Shift Keying</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>SDR</td>
<td>Software Defined Radio</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TED</td>
<td>Timing Error Detector</td>
</tr>
</tbody>
</table>
Part I

Background and Theory
1 Introduction

1.1 Agama Technologies

Agama Technologies [1], a Linköping based TV service quality assurance (QA) company, is a specialist in telco-grade video QA and monitoring solutions. Agama enables IP, cable, broadcast and over-the-top content TV operators to systematically manage service quality. In 2001, the expert team behind Agama saw an upcoming challenge for emerging new and more complex TV services and in 2004 decided to start working within the field of TV service QA by founding Agama Technologies. Today, Agama has a wide range of customers worldwide, including 15+ European tier one TV operators who currently rely on Agama Digital TV Monitoring Solution.

1.2 Software Defined Radio

SDR [11] is booming widespread within the communication sector and increases the flexibility of the receivers and transmitters. SDR makes it possible to upgrade the encoding and modulation in the transmitter and the demodulation and decoding in the receiver by reprogramming the software that handles the transmitting and receiving. Instead of using hardware designed for a specific task, using SDR and a SDR compatible transmitter or receiver it is possible to extend them to include multiple tasks. For example, a single processor in combination with different SDR could receive DVB-C, DVB-T and DVB-S. It simplifies logistics. Usually the SDR-code is implemented on a digital signal processor (DSP) or a Field Programmable Gate Array (FPGA). These can easily be reprogrammed.
1.3 Digital Video Broadcasting by Cable

DVB-C is the European consortium standard for the broadcast transmission of digital television over cable. The standard was published by the European Telecommunications Standards Institute (ETSI) [2] in 1994 in order to transmit digital audio/video streams using quadrature amplitude modulation (QAM) and source coding. DVB-C is today used widely over the world. For example, at Com Hem Sweden most of the television channels are transmitted and received using the DVB-C standard by ETSI.

1.4 Research Motivation

The thesis was initiated in order to investigate how efficient the extraction of radio frequency metrics from broadcast streams can become using SDR. Current available receiver cards are inflexible when it comes to retrieving, investigating and manipulating lower level metrics such as signal to noise ratio (SNR), modulation error ratio (MER) and channel impulse response. There is a possibility that this can be done in a more flexible way by using a receiver card in combination with SDR and this has to be investigated.

1.5 Research Objectives

The goal of the project was to develop a system that receives different quadrature amplitude modulated signals and investigates which radio frequency metrics that can be extracted and how efficient the extraction can become. The first objective was to investigate which different methods that could be used in order to receive, synchronize and equalize the data and compare the efficiency and quality of the methods. The second objective was to implement the most suitable methods to receive data and calculate the metrics.

1.6 Thesis Overview

Chapter 1 Provides an introduction to the thesis were the research motivation and objectives are introduces.

Chapter 2 Discusses the theory behind communication systems, for example how transmission and receiving works and what errors might occur are discussed.

Chapter 3 Presents the quadrature amplitude demodulation. The demodulation steps are discussed and different algorithms for each step are compared.

Chapter 4 Contains a presentation of the implementation of the methods that were chosen and how they were implemented.

Chapter 5 Presents the results of the thesis.
Chapter 6  Concludes with a summary of the thesis.
A communication system is a system for transmitting and receiving data in different forms. The basic elements of a communication system can be seen in figure 2.1 below.

Every communication system needs an information source which contains the data that are to be transmitted. The source encoder converts the digital input data into a binary sequence of data that is passed through the channel encoder which adds some redundancy to the data which is often required to overcome the noise added by the channel. The redundancy level in this case is the number of bits used to transmit the information in comparison with the number of bits that are actually carrying the information. The data is then mapped onto signal carrier waveforms by modulating the carrier wave before the signal is transmitted over the channel. The transmission can for example be over a coaxial cable, a telephone line or wirelessly. The receiver side receives the signal and samples it.
Using the samples in combination with certain methods and calculations, the de-
modulator decides which symbol that was transmitted by the transmitter. This
decision depends fully on which modulation type that was used by the trans-
mitter. After the demodulation, the symbol is decoded by the channel decoder
which corrects errors that are caused by the channel and the source decoder that
decodes the source encoding performed in the transmitter. Since the main ob-
jectives of this thesis are the modulation and demodulation, these parts are dis-
cussed in more detail.

2.1 Modulation Overview

Modulation is the process where the information from a source is encoded onto
signal carrier waves with a certain frequency. The information can be encoded
by changing the amplitude, frequency and phase of the signals. An example of a
modulated carrier signal is

\[ S(t) = A(t) \cos(F_c t + \phi(t)) \]  (2.1)

where \( A(t) \) is the amplitude, \( \phi(t) \) the phase and \( F_c \) is the carrier frequency. In-
formation is transmitted by modulating the sinus wave, i.e. changing \( A(t) \) and
\( \phi(t) \).

Some different ways to transmit data by modulating the carrier signal are:

**Amplitude shift keying**

The data to be transmitted are mapped onto a set of amplitude levels that are
used to modulate the amplitude of the carrier wave while the phase and fre-
quency are kept constant. In figure 2.2, an example of how the input bits can
be used to modulate the amplitude of a carrier wave.

![Figure 2.2: Amplitude and phase of the vector.](image)

The demodulator of an amplitude shift keying (ASK) signal is designed specifi-
cally for the symbols used by the modulator. It determines the amplitude of the
received signal and uses the value to recover the original data.
2.1 Modulation Overview

Phase shift keying
The data is mapped onto a set of phase levels that are used to modulate the phase of the carrier wave while the amplitude and frequency are kept constant. In figure 2.3, an example of how the input bits can be used to modulate the phase of a carrier wave is displayed.

![Figure 2.3: Amplitude and phase of the vector.](image)

The demodulator of a phase shift keying (PSK) signal is also designed specifically for the symbols used by the modulator. It determines the phase of the received signal and uses the value to recover the original data. This is often done by comparing the phase of the received signal to a reference signal in order to calculate the phase of the transmitted signal.

Frequency shift keying
The data is mapped onto a set of frequency levels that are used to modulate the frequency of the carrier wave while the amplitude and phase are kept constant. In figure 2.4, an example of how the input bits can be used to modulate the frequency of a carrier wave.

![Figure 2.4: Amplitude and phase of the vector.](image)

The demodulation of a frequency shift keying (FSK) signal can be done using different algorithms. For example, the binary FSK uses the Goertzel algorithm.
2.1.1 Quadrature Modulation

The waveform of a modulated signal can be geometrically represented as vectors in the complex plane. The length of the vectors corresponds to the amplitude of the signal and the direction of the vector corresponds to the phase. This way to represent the modulated signal is called quadrature modulation. The complex plane in figure 2.5 is referred to as the constellation diagram.

![Figure 2.5: Amplitude and phase of the vector.](image)

Quadrature modulation is commonly used in implementations. The data bits to be transmitted are used to modulate the amplitude of two carrier signals at the same frequency. The two carrier signals are out of phase by 90 degrees, hence the name of the modulation scheme. Usually, one of the carrier waves is mathematically described by a cosine wave and the other by a sine wave, they are called the in-phase signal and the quadrature-phase signal.

Instead of describing the modulated signal as in equation 2.1, the modulated signal can be represented using the amplitudes along the real and imaginary axis. This representation is

\[ S(t) = A_I(t) \cos(2\pi F_c t) - A_Q(t) \sin(2\pi F_c t) \]  \hspace{1cm} (2.2)

where \( A_I(t) \) is the in-phase amplitude (\( \Re e \)) and \( A_Q(t) \) is the quadrature-phase amplitude (\( \Im m \)). In the constellation diagram in figure 2.6, the in-phase amplitude and quadrature-phase amplitude are displayed.
2.1 Modulation Overview

Figure 2.6: In-phase and quadrature-phase amplitude displayed in QAM constellation diagram.

How many bits per symbol that can be transmitted depends on the number of different phase and amplitude levels that exist in the constellation diagram, i.e. the number of constellation points in the imaginary plane. For example, 16 constellation points in the diagram makes it possible to transmit \( \log_2(16) = 4 \) bits per point. The number of constellation points and bits per symbols for different types of phase shift keying techniques can be seen in table 2.1.

<table>
<thead>
<tr>
<th>Modulation Type</th>
<th>Constellation Points</th>
<th>Bits per Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary PSK</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Quadrature PSK</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8-PSK</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2.1: Bits per symbol for different modulation types.

2.1.2 Phase Shift Keying

In PSK, the phase of a constant amplitude and frequency carrier wave is switched between a finite number of values in order to transmit the data. The data bits to be transmitted are mapped onto the points in the constellation diagram from which the phase and constant amplitude values are retrieved. An example of a PSK constellation diagram can be seen in figure 2.7.
Binary Phase Shift Keying

An example of PSK modulation is the binary phase shift keying (BPSK). BPSK is a common modulation type where the phase of the carrier signal is switched between two values, usually separated by 180 degrees. In figure 2.8 below, an example of a BPSK constellation diagram is shown.

BPSK is the simplest form of phase shift keying but is the most robust modulation of all PSKs. This is because the large distance between the constellation points makes it robust against noise. However, BPSK is unsuitable for high data rates because there are only two points in the constellation diagram and thus only...
$\log_2(2) = 1$ bits per symbol can be transmitted. According to figure 2.7, depending on if the data bit to be transmitted is a 1 or a 0 the BPSK modulates the carrier signal with a phase shift of 0 degrees or 180 degrees.

**Quadrature Phase Shift Keying**

Quadrature phase shift keying (QPSK) is a little more advanced type of PSK-modulation. As modulation points, it uses four points on the constellation diagram placed on a circle with a certain amplitude. Since QPSK uses four different phases it can encode two bits per symbol. In figure 2.9, an example of how a constellation diagram for QPSK might look is displayed.

![QPSK constellation diagram](image)

*Figure 2.9: QPSK constellation diagram.*

The QPSK can be viewed as two independent BPSK signal. The disadvantages of the QPSK in comparison with the BPSK are that it is a bit more complicated and a bit more sensitive to noise. However, in modern electronics technology, the complexity penalty becomes quite moderate. The advantage is that it can either transmit twice the data rate in a given bandwidth as the BPSK or transmit at the same data rate but only use half the bandwidth as the BPSK.

### 2.2 Quadrature Amplitude Modulation

To obtain higher spectral efficiency, quadrature amplitude modulation (QAM) can be used. QAM is a modulation scheme that combines amplitude modulation and PSK by changing both the phase and amplitude of a carrier wave signal. The data bits to be transmitted are mapped onto a set of combination of amplitude levels. The amplitude values are then used to modulate the amplitude of two carrier signals. The two amplitude modulated and out of phase carrier signals are then summed to become a combination of both ASK and PSK.
As stated, the in-phase (I) and quadrature-phase (Q) amplitudes are taken from the real and imaginary values in the constellation diagram in figure 2.6 and the number of bits per symbol that are to be transmitted decides how many constellation points that are necessary to transmit the information. In table 2.2, the number of constellation points and bits per symbol for a number of QAM types are shown.

<table>
<thead>
<tr>
<th>Modulation Type</th>
<th>Constellation Points</th>
<th>Bits per Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-QAM</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>64-QAM</td>
<td>64</td>
<td>6</td>
</tr>
<tr>
<td>256-QAM</td>
<td>256</td>
<td>8</td>
</tr>
</tbody>
</table>

*Table 2.2: Bits per symbol for different modulation types.*

Based on the structure of the constellation diagram there are different types of QAM. There is the rectangular QAM where the constellation points in the constellation diagram form rectangles. There is also circular QAM where the constellation points are placed along a number of concentric circles. Example of a circular constellation and a rectangular constellation are shown in figure 2.10 and figure 2.11 below.

![Circular 16-QAM](image1)

*Figure 2.10: Circular 16-QAM.*

![Rectangular 16-QAM](image2)

*Figure 2.11: Rectangular 16-QAM.*

The constellation types perform differently under different channel conditions. Rectangular QAM is easier to modulate and demodulate because of its regular structure. However, the circular QAM performs better in channels affected by phase noise since the distance between the points in the constellation diagram is often larger than in the rectangular case.
2.2 Quadrature Amplitude Modulation

2.2.1 Multilevel Quadrature Amplitude Modulation

As stated above, the transmission of multiple bits per symbol requires a higher amount of amplitude levels for the I and Q amplitudes. However, the modulation schemes with higher amount of bits per symbol are more sensitive to noise which can cause high bit error rates (BER). In figure 2.12, the constellation diagrams of some quadrature amplitude modulation schemes for different amounts of amplitudes are shown when they have the same mean energy level. When using a large amount of amplitude levels, they are sometimes called multilevel quadrature amplitude modulation (M-QAM).

The noise sensitivity of a certain modulation type can be traced to the distance between the constellation points in the constellation diagram. The figures show that when the constellation diagrams are scaled such that the mean energy of the constellation points are the same, which is often the case, the distance between the constellation points decreases when the number of constellation points increases. Therefore, the 256-QAM is most sensitive to noise while 4-QAM is least sensitive. The distance between the amplitude levels used to modulate the carrier waves for the different modulation types are displayed in table 2.3. In this case, the mean energy of the constellation points is set to 1.
<table>
<thead>
<tr>
<th>Modulation Type</th>
<th>Distance Between Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-QAM</td>
<td>1.4142</td>
</tr>
<tr>
<td>16-QAM</td>
<td>0.6326</td>
</tr>
<tr>
<td>64-QAM</td>
<td>0.3086</td>
</tr>
<tr>
<td>256-QAM</td>
<td>0.1534</td>
</tr>
</tbody>
</table>

Table 2.3: Bits per symbol for different modulation types.

2.3 Digital QAM Transmitter

In figure 2.13, a typical structure of a digital QAM transmitter is displayed. The source data is split in two branches, one channel for the in-phase and one channel for the quadrature-phase.

![Figure 2.13: Example of the structure of a QAM transmitter.](image)

The splitting into two branches generates two independent signals to be transmitted. The signals are QAM and filtered by a transmit filter before they are multiplied with the carrier wave signals. If the symbols to be transmitted are generated using square pulses, which is usually the case, the signal power will be spread across a large bandwidth. This is why a transmit filter is used in order to limit the bandwidth of the signal without introducing a large amount of inter-symbol interference (ISI). The transmitter filter will be discussed more thoroughly in section 3.2. The I-channel is multiplied with a cosine while the Q-channel is multiplied with a sine. This causes the 90 degrees phase shift between them. The two signals are then summed to complete the QAM.

2.4 Digital QAM Receiver

The structure of the typical QAM receiver is shown in figure 2.14 below.
Figure 2.14: Example of the structure of a QAM receiver.

The automatic gain control (AGC) is an adaptive system that performs a scaling of the signal in order to utilize the dynamic range of the analog to digital (A/D) converter. The output signal level is fed back or fed forward, depending on the method, to adjust the gain to get appropriate amplitudes. This is important in channels where the attenuation of the channel varies with time. The samples are multiplied with a cosine and sine wave to extract the I or Q component of the transmitted signal.

In practice, there is an offset between the frequency and phase of the cosine and sine wave in the transmitter and the local oscillator in the receiver, the effects of this offset will be discussed more thoroughly in section 2.5.4. This multiplication also causes the signal to occur at two frequencies and the signal needs to be low-pass filtered. By excluding the transmit filter, this can be displayed using the equations 2.3 to 2.5 below.

\[
S(t) = A_I(t) \cos(2\pi F_c t) - A_Q(t) \sin(2\pi F_c t) \tag{2.3}
\]

\[
R(t) = S(t) \cos(2\pi F_c t) - jS(t) \sin(2\pi F_c t) = ... \\
= 0.5 \cdot A_I(t)(1 + (\cos(4\pi F_c t) - \sin(4\pi F_c t))) \\
+ j0.5 \cdot A_Q(t)(1 + (\cos(4\pi F_c t) - \sin(4\pi F_c t))) \tag{2.4}
\]

\[
R_{LP}(t) = 0.5 \cdot A_I(t) + j0.5 \cdot A_Q(t) \tag{2.5}
\]

\(S(t)\) is the signal prior to the multiplication with the cosine and sine waves in the receiver, \(R(t)\) is the signal after the multiplication and \(R_{LP}(t)\) is the signal after the low-pass filter.

After the multiplication with the sine and cosine waves and the low-pass filter, the A/D-converter samples the analog signal at a certain rate. The clock that performs the digital to analog (D/A) conversion in the transmitter and the clock that performs the A/D-conversion in the receiver are most certainly not synchronized in practice. This causes a timing offset which also will be discussed more thoroughly later in section 2.5.5. The I and Q samples are then fed to the matched filters where they are filtered to remove noise and ISI. Following the filtering, the
signal is demodulated, the errors are corrected and the samples are de-mapped before they are fed to the source decoder.

2.5 Noise and Offsets

In a communication system using quadrature amplitude modulation and demodulation, there are a number of noise sources and offsets that cause errors between the transmitted and received signal.

2.5.1 Additive White Gaussian Noise

This type of noise occurs in all types of communication channels. It is mainly caused by the electronic components in the receiver and transmitter. Additive white Gaussian noise (AWGN) can statistically be characterized by a Gaussian process, hence the name. The AWGN channel consists of a transmitted signal and the additive noise, this is typically structured and displayed as in figure 2.15.

![Figure 2.15: Structure of AWGN.](image)

In figure 2.15, \( S(t) \) is the transmitted signal, \( N(t) \) is the additive white Gaussian noise and \( R(t) \) is the received signal.

The amount of AWGN can be measured using the signal to noise ratio (SNR), usually displayed in decibel (dB). The more constellation points the modulator uses, the more sensitive to noise it gets and therefore needs a higher value of the SNR to get a good result. In relation to the satellite and terrestrial transmission, the transmission by cable has a quite low amount of AWGN. Examples of 16-QAM data with different amounts of AWGN and therefore different SNR are displayed in figure 2.16 below.
2.5 Noise and Offsets

2.5.2 Quantization Noise

In A/D-conversion, there is a difference between the actual analog value and the quantized digital value. This difference is called quantization error. The error is caused by rounding and truncation. The quantization error is related to the number of quantization points where fewer quantization levels leads to larger quantization error and vice versa. The number of quantization levels that are used is changed by representing the signal values using more bits, so if a large amount of bits are used, the quantization noise will be relatively small. The number of quantization levels used by the A/D-converter in relation to the number of bits used by the converter can be described by equation 2.6 below.

\[ B = 2^N \]  

(2.6)

\( B \) is the number of quantization levels and \( N \) is the number of bits used by the A/D-converter to describe the digital values.

An example of quantization error for an A/D-converter with a 2-bit resolution can be seen in figure 2.17.

*Figure 2.16: 16-QAM data with different SNRs.*
2.5.3 Channel Distortions

In certain channels, the amplitude of the received signals can vary by time. This causes a signal where some parts of it may have much higher amplitude levels than other parts, even though they should be approximately the same. 16-QAM data put through different types of channels can be seen in figure 2.18.

The channel distortions are often corrected using different equalizers. Various methods to perform equalization will be discussed more in section 3.5.
2.5 Noise and Offsets

2.5.4 Carrier Offset

The oscillator in the transmitter and receiver are often running free. This causes a frequency and phase mismatch between the frequency and phase of the carrier wave in the transmitter and the frequency and phase of the local oscillator in the receiver. This error misplaces the received symbol. The misplacement can be visualized as a rotation in the constellation diagram consisting of the received symbols. In figure 2.19, the rotation caused by a phase offset is displayed and in figure 2.20, the rotation caused by a frequency offset is displayed.

![Figure 2.19: 16-QAM data with carrier phase offset.](image1)

![Figure 2.20: 16-QAM data with carrier frequency offset.](image2)

This offset can be mathematically described using equations 2.7 and 2.8.

\[
S(t) = A_I(t) \cos(2\pi F_c t + \phi_c(t)) - A_Q(t) \sin(2\pi F_c t + \phi_c(t)) \quad (2.7)
\]

\[
R(t) = S(t) \cos(2\pi F_d t + \phi_d(t)) - jS(t) \sin(2\pi F_d t + \phi_d(t)) \quad (2.8)
\]
$F_c$ and $\phi_c$ is the frequency and phase of the carrier wave in the transmitter and $F_d$ and $\phi_d$ are the frequency and phase of the local oscillator in the receiver.

Using Euler in combination with the procedure in equation 2.7 and 2.8, the received signal with a carrier frequency and phase offset after the low-pass filtering can be described as in equation 2.9.

$$R_{LP}(t) = (A_I(t) + jA_Q(t)) \cdot e^{-i2\pi(F_c-F_d)t + (\phi_c-\phi_d)}$$ (2.9)

$F_c - F_d$ is the frequency offset and $\phi_c - \phi_d$ is the phase offset.

### 2.5.5 Timing Offset

In order for the receiver to receive the correct symbol, the D/A-converter clock in the transmitter and the A/D-converter clock in the receiver must be synchronized and running at the same frequency. This is most often not the case. Instead, the symbols in the receiver are taken at the wrong instance. A simple example visualization of this error is shown in figure 2.21 below.

---

**Figure 2.21: Timing offset example.**

---

In the figure above:

- The synchronized clock samples the signal at the correct places, in this case at the peaks of the sine wave.

- The phase unsynchronized clock starts sampling at the incorrect place and then samples at the same frequency as the synchronized one causing the signal to be sampled at the incorrect places.

- The frequency unsynchronized one starts sampling at the correct place but then samples at the wrong frequency causing it to sample at other places.
than at the peaks.

The timing recovery is extremely important and without it, the received samples in the constellation diagram will basically just look like noise. In figure 2.22, the plot to the left represents the modulated transmitted data and the plot to the right represents the received I and Q-samples when both fixed and variable timing error were present between the D/A and A/D-converters.

\textbf{Figure 2.22:} Left: Transmitted signal. Right: Received signal with fixed and variable timing error.
It is in the QAM demodulator where the receiver matched filtering is completed, the errors are corrected, the synchronization is performed and the de-mapping is executed. The received signal is filtered to reduce noise and ISI, the errors and synchronization offsets caused by the channel and oscillators are corrected as much as possible and the de-mappings of the received signals on to the constellation diagram are all parts of the demodulation. The steps of the demodulator are discussed in separate sections. An example of the structure of a QAM demodulator can be seen in figure 3.1.

![Figure 3.1: Basic structure of a QAM demodulator.](image)

### 3.1 Automatic Gain Control

The AGC can be implemented in several ways. There are both feed-back and feed-forward methods to correct the variations between amplitudes in different sequences of the signal.

The structure of a feed-forward system is displayed in figure 3.2.
Feed-forward systems calculate a gain that is used to scale the received signal. The blocks in figure 3.2 perform the following tasks:

- **Detector** Detects the amplitude of the received signal.
- **Comparator** Compares the amplitude of the received signal with a reference amplitude.
- **Gain** The gain factor is adjusted using the value calculated by the comparator.

The structure of a feed-back system is displayed in figure 3.3.

Feed-back systems utilize a tracking loop that tracks the values of the signals in order to scale them properly. Typically a new gain is calculated for every symbol. The blocks in figure 3.3 perform the same tasks as in the feed-forward case in figure 3.2.
3.2 Matched Filtering

3.2.1 Pulse Shaping

When the modulation rate increases, the signal’s bandwidth increases. If the signal bandwidth becomes larger than the channel bandwidth, the channel introduces distortions to the signal called ISI. In order to reduce the bandwidth of the transmitted signal the waveform of the transmitted pulse is changed. This is done using pulse shaping filters. The filter to be used as a pulse shaping filter must satisfy a certain criteria, called the Nyquist ISI criterion. The Nyquist ISI criterion states that the signal value at a certain sampling instance must only depend on one symbol. Theoretically, to minimize the ISI it is ideal to use a sinc shaped filter. But since its tails are infinitely long, it is not suitable to implement or use in practice. Therefore, a widely used pulse shaping filter is the raised cosine filter which is practical to implement and gives the communication system the opportunity of setting the excess bandwidth using the roll-off factor, which is a constant between 0 and 1.

The impulse response of the sinc filter and a number of raised cosine filters for different roll-off factors can be seen in figures 3.4. \( T \) is the symbol time and is set to 10\( ms \).

![Sinc and raised cosine impulse responses.](image)

The bandwidth of the signal depends on the roll-off factor. To get the bandwidth of the signal, the symbol rate is multiplied with \((1 + \text{roll-off factor})\). In order to prevent ISI, this value must be lower than the channel bandwidth.

3.2.2 Matched Filtering

A common way to implement the pulse shaping filter is to split the pulse shaping filter into two parts, one at the transmitter side and one at the receiver side.
The one at the receiver side is a complex conjugated and time reversed version of the transmitter side filter. The splitting of the filter is to achieve optimum tolerance for noise in the system. In the raised cosine case, the filter is split into two root raised cosine (RRC) filters where the root raised cosine filters’ amplitude response is point wise the square roots of the raised cosine filter. Since the RRC-filter is symmetrical, the conjugated and time reversed version of the filter is equal to the original. The frequency response of a raised cosine filter is described by its amplitude response which is displayed in figure 3.5 [6].

\[
X_{rc}(\tilde{f}) = \begin{cases} 
T & \text{for } 0 \leq |\tilde{f}| \leq \frac{1-\beta}{2T} \\
\frac{T}{2} \left( 1 + \cos \left( \frac{\pi T}{\beta} \left( |\tilde{f}| - \frac{1-\beta}{2T} \right) \right) \right) & \text{for } \frac{1-\beta}{2T} \leq |\tilde{f}| \leq \frac{1+\beta}{2T} \\
0 & \text{for } |\tilde{f}| > \frac{1+\beta}{2T}
\end{cases}
\]

\textit{Figure 3.5: Amplitude response of a RC filter.}

where \( R \) is the roll-off factor, \( f \) is the frequency, \( T \) is the symbol period and \( \beta \) is the roll-off factor.

The “root” aspect of a root-raised cosine filter is in the frequency domain. So, the frequency response of the root raised cosine filter is simply the square root of the raised cosine filter frequency response.

The root raised cosine filter for some roll-off factors frequently used in demodulators are displayed in figure 3.6.

\textit{Figure 3.6: Root raised cosine impulse responses.}
Images of how the signal to be transmitted can look before and after the root raised cosine filtering at the transmitter side are displayed in figure 3.7.

![Image: Transmitted I-channel and Root Raised Filtered I-channel](image)

*Figure 3.7: I-channel, prior to and post root raised cosine filtering.*

Note that the RRC-filtered signal is up-sampled by a factor 2. This up-sampling is required in order to fulfill the Nyquist sampling theorem.

### 3.3 Symbol Timing Recovery

The task of the symbol timing recovery is to take care of the timing offset, i.e. the problem caused by the fact that the D/A-converter in the transmitter and the A/D-converter in the receiver are not synchronized. There are several ways of performing symbol timing recovery. The ideal is if the transmitter and receiver were running of the same clock. However, this is typically impossible in most of the communication systems. Another way is to transmit some kind of symbol clock information along with the signal. A problem is that it requires that the transmitter allocates some of its bandwidth to transmit the symbol clock information. There are also a number of self-synchronizing methods that works well with random data with no knowledge about the timing. Some of these are discussed more thoroughly.

#### 3.3.1 Feed-forward Methods

The basic structure of a feed-forward timing recovery method can be seen in figure 3.8.
The input signal is usually the output from the receiver matched filter used to perform the pulse shaping of the signal, the timing estimator calculates an estimate of the input signals phase offset and the timing corrector interpolates the signal using the calculated timing phase offset in order to receive the correct symbols from the signal.

**Square Timing Recovery**

The square timing recovery recovers the fixed symbol timing phase using a squaring method. This algorithm works on blocks containing a sequence of the input signal. The sequences will contain a symbol frequency at half the symbol rate and if the signal is squared, the frequency component at half the symbol rate will be doubled. This results in a frequency component at $1/T$ where $T$ is the symbol period. Using a narrow band-pass filter, the sample clock can be extracted from the squared signal. The timing estimator used by the square timing recovery scheme can be seen in equation 3.1.

$$- \frac{1}{2\pi} \arg \left( \sum_{n=0}^{LN-1} |x_{m+1}|^2 e^{-j2\pi n/N} \right)$$  \hspace{1cm} (3.1)

Where $x$ is a block containing some of the samples from the input signal, $N$ the number of samples per symbol and $L$ the length of the input block.

The square timing recovery method is suitable for linear baseband modulation types, such as PSK and QAM where the signals are likely to consist of sequences with symbols that alternate between positive and negative values with approximately the same amplitude. This probability decreases when the number of constellation points and level of amplitudes increases which causes the square timing recovery to be less effective. Another disadvantage is that it requires that the phase offset is constant for all the input samples in the block which means it does not work well when there is a variable offset between the clocks.
3.3 Symbol Timing Recovery

3.3.2 Feed-back Methods

The basic structure of a feed-back timing recovery method can be seen in figure 3.9.

![Figure 3.9: Feed-back timing recovery scheme.](image)

The input signal is usually the output from the receiver matched filter used to perform the pulse shaping of the signal. The interpolator interpolates using the interpolation factor provided by the controller to receive the correct symbols from the input samples and the timing error detector (TED) calculates timing error for each symbol. Which TED that is most suitable to use depends on which modulation type that is used. The loop filter updates the phase estimate of the symbol using the timing error and the previous phase estimate. The structure of the loop filter varies depending on the need of simplicity, speed and performance. The controller then uses the phase estimate to determine the interpolation factor, i.e. the interpolating instants between the input samples.

Some different algorithms that are popular to use as timing error detectors in feed-back timing recovery methods will now be discussed in more detail.

**Early-Late Gate Symbol Timing Recovery**

The early-late gate method [7] relies on the property that the absolute value of the signal amplitude is at its highest at the optimum sample point when averaged over a number of symbols. The early late gate algorithm then generates its error by using samples that are early and late compared to the ideal sampling point, i.e. at the symbol. This technique requires 3 samples per symbol, which is impractical in systems with high symbol rates.

An example where the receiver samples later than at the optimum offset is shown in figure 3.10 [5].
If the receiver samples at the optimum time, the average of the absolute value of the early and late samples will be equal, otherwise they will differ. Therefore, the difference between the average early and late samples can be used as a timing error for the feed-back loop. The algorithm used to calculate the timing error is displayed in equation 3.2 to 3.4.

\[
e[k] = e_1[k] + e_Q[k] \tag{3.2}
\]

\[
e_1[k] = y_1[kT + d_k] \cdot (y_1[kT + \frac{T}{2} + d_k] - y_1[kT - \frac{T}{2} + d_{k-1}]) \tag{3.3}
\]

\[
e_Q[k] = y_Q[kT + d_k] \cdot (y_Q[kT + \frac{T}{2} + d_k] - y_Q[kT - \frac{T}{2} + d_{k-1}]) \tag{3.4}
\]

\(y_1\) and \(y_Q\) are the in-phase and quadrature components of the input signal, \(T\) is the symbol period and \(d_k\) is the phase estimate for the \(k_{th}\) symbol.

This algorithm is suitable for PAM and QAM and is quite similar to the Gardner algorithm. However, the Early-late gate method introduces higher self-noise than the Gardner method and performs worse in systems with high signal to noise ratio.

**Mueller Muller Symbol Timing Error Detector**

The Mueller Muller method [9] just requires one sample per symbol. The timing error is calculated using equation 3.5.

\[
e[k] = \Re(c'_{k-1} \ast y[kT + d_k] - c'_{k} \ast y[(k - 1) \ast T + d_{k-1}]) \tag{3.5}
\]

\(y\) is the input signal, \(c'_k\) is the decision based on the sample value \(y[kT + d_k]\), \(T\) is the symbol period and \(d_k\) is the phase offset for the \(k_{th}\) symbol.
The decision value \( c'_k \) is calculated by making a decision of the sample value \( y[kT + d_k] \). This decision is the closest point in the constellation diagram for the modulation type used by the transmitter. The need for a decision on each sample makes the method very sensitive to carrier offsets and the carrier recovery must be performed prior to the Mueller Muller timing error detector. This makes it unsuitable in many cases.

**Gardner Symbol Timing Error Detector**

One of the most widely used timing error detectors (TED) is the Gardner TED [4]. It uses a zero-crossing synchronization technique that relies on the property that, on average, zero-crossings occur half a symbol before the optimal time to sample. The method uses two samples per symbol and calculates the timing error using equation 3.6 to 3.8 below.

\[
\begin{align*}
    e[k] &= e_I[k] + e_Q[k] \quad (3.6) \\
    e_I[k] &= (y_I[(k - 1) \cdot T + d_{k-1}] - y_I[kT + d_k])y_I[kT - \frac{T}{2} + d_{k-1}] \quad (3.7) \\
    e_Q[k] &= (y_Q[(k - 1) \cdot T + d_{k-1}] - y_Q[kT + d_k]) \cdot y_Q[kT - \frac{T}{2} + d_{k-1}] \quad (3.8)
\end{align*}
\]

\( y_I \) and \( y_Q \) are the in-phase and quadrature components of the input signal, \( T \) is the symbol period and \( d_k \) is the phase estimate for the \( k \)th symbol.

If \( e[k] \) is negative, the sampling occurs too early and if \( e[k] \) is positive, the sampling occurs too late. The value of \( e[k] \) is then used as timing error to adjust the timing offset estimate via the loop filter and controller. An example of each case is shown in figure 3.11.

![Figure 3.11: Early, late and perfect timing for Gardner TED.](image-url)
Some advantages of the Gardner TED are that it is insensitive to carrier offset which makes it possible to recover the symbol timing prior to the carrier recovery. It is also quite easy to implement and works well for BPSK and QPSK. Some disadvantages is that for the Gardner TED to have a good precision when it determines if it is sampling too early or too late, it needs a transition from positive to negative or vice versa. The magnitude of the two points on each side of the middle point also needs to have approximately the same amplitude for the zero-crossing to occur at half a symbol before the optimal time to sample. This is not always the case for received M-QAM modulated signals which makes the Gardner TED less suitable for M-QAM \( (M \geq 16) \). For these M-QAM signals, the probability for there to be no zero-crossing also increases.

**Modified Gardner Symbol Timing Error Detector**

The modified Gardner symbol TED is exactly what it sounds like, a modified version of the previously discussed Gardner symbol TED. The zero-crossing technique is extended to operate on larger constellations, i.e. modulation schemes with more constellation points by using the crossing of a predicted edge amplitude instead of the zero-crossing. This technique takes care of the problems that the magnitude of the two points on each side of the middle sample needs to have approximately the same amplitude and the fact that sometimes no zero-crossing occur. This is because transitions that do not cross zero will in this case instead cross the predicted edge amplitude.

The equation used to calculate the timing error for the modified Gardner TED is displayed in equation 3.9 to 3.13.

\[
e[k] = e_I[k] + e_Q[k] \tag{3.9}
\]

where, \( e_I[k] \) and \( e_Q[k] \) are,

\[
e_I[k] = (y_I[(k-1)T + d_{k-1}] - y_I[kT + d_k])\delta_I[k] \tag{3.10}
\]

\[
e_Q[k] = (y_Q[(k-1)T + d_{k-1}] - y_Q[kT + d_k])\delta_Q[k] \tag{3.11}
\]

and \( \delta_I[k] \) and \( \delta_Q[k] \) are,

\[
\delta_I[k] = (y_I[kT - \frac{T}{2} + d_k] - \frac{y_I[(k-1)T + d_{k-1}] + y_I[kT + d_k]}{2}) \tag{3.12}
\]

\[
\delta_Q[k] = (y_Q[kT - \frac{T}{2} + d_k] - \frac{y_Q[(k-1)T + d_{k-1}] + y_Q[kT + d_k]}{2}) \tag{3.13}
\]

In the equations above, \( y_I \) and \( y_Q \) are the in-phase and quadrature components of the input signal, \( T \) is the symbol period and \( d_k \) is the phase estimate for the \( k_{th} \) symbol.
A simplified example of how the modified Gardner algorithm for perfect timing can look can be seen in figure 3.12.

![Figure 3.12: Modified Gardner algorithm example.](image)

### 3.4 Carrier Recovery

The carrier recovery has the task of tracking the phase and frequency offsets between the frequency and phase of the carrier wave in the transmitter and the frequency and phase of the local oscillator in the receiver. In the ideal system, the carrier frequency oscillators in the transmitter and receiver would have the exact same frequency and phase which means there is no need for a carrier recovery. However, this is rarely the case.

Depending on the modulation the carrier recovery can be performed in two different ways.

- Non-data aided, where no information about the received signal is known.
- Decision directed, where information about the received signal is either predicted or known.

#### 3.4.1 Non Data Aided

Non data aided carrier recovery does not rely on any information about the modulated symbols. This method is usually used for simple carrier recovery schemes.

**Multiply Filter Divide**

The multiply filter divide method [12] is performed by applying a non-linear operation to the modulated signal in order to create harmonics of the carrier frequency. The harmonic is band-pass filtered and frequency divided to recover the carrier frequency. Generally, the number of constellation points matches the
order of the nonlinear operator required to produce a relatively clean carrier harmonic. Therefore, a BPSK-signal must be squared and a QPSK-signal must be squared twice.

An advantage of the method is that it is non-data aided, which means that it can be performed before the symbol timing recovery without any problem. It works well for BPSK and QPSK, but also quite well for M-QAM (M \geq 16). The disadvantage with the method is that it is not able to recover the phase offset.

In equation 3.14 and 3.15, the multiply filter divide method in the power of four case when a frequency offset \( F_o \) is present is shown.

\[
R[k] = A[k] \cos(F_o t + n\frac{\pi}{2}), \quad n = 0, 1, 2, 3
\]

\[
R[k]^4 = \frac{A[k]^4}{8} (3 + 4 \cos(2f_o t + n\pi) + \cos(4F_o t + n\pi2))
\]

\( R[k] \) is the received signal.

Taking the signal to the power of four produces a signal at four times the carrier offset with no phase modulation since the phase is \( n2\pi \). Using an appropriate filter around \( 4F_o \), the frequency offset can be retrieved.

**Phase Locked Loops**

The task of the phase-locked loop is to generate an output signal with a phase that is related to the phase of the reference signal, which in most cases is the input signal. The circuit compares the phase of the input signal with the phase in the local oscillator at the receiver in order to adjust it to match the input phase. The phase offset is used in a feed-back loop in order to keep track of the phase over time. Since the frequency is the time derivate of the phase, keeping the phase of the input and output locked also causes the frequency between the input and output to be locked. A widely used loop that is based on the phase locked loop is the Costas loop [3].

A typical structure of the Costas loop can be seen in figure 3.13.

![Costas loop](image)
3.4.2 Decision Directed

If a successful timing recovery has been performed prior to the carrier recovery, a decision directed method can be used to recover the carrier frequency. In a decision directed method, the symbol outputs from the timing recovery are fed to a comparison circuit where the phase and frequency errors between the synchronized symbols and the closest constellation points in the constellation diagram used by the modulation type currently used. The errors between the symbols and their closest constellation points are calculated using arc tangent. These errors are then tracked and updated using a phase locked loop.

The decision directed method is suitable to track frequency errors that are less than the symbol rate because the comparisons are performed one time per symbol. Another problem is that the arc tangent is limited to compute a phase correction between 0 and $\frac{\pi}{2}$. This phase offset is usually solved by using differential coding in the source encoder and decoder. An example of how a decision directed phase locked loop can look like is displayed in figure 3.14.

![Figure 3.14: Decision directed phase locked loop.](image)

3.5 Equalization

As stated before, ISI is a common problem in communication systems that cause the received symbol to depend on several transmitted symbols. The ISI is caused due to the bandlimiting of the channel and also echo effects in the channel. To help the matched filtering to prevent ISI, equalizers are commonly used. Another important task of the equalizer is to prevent the channel distortions of distorting the signal to much.

The equalizer is a linear filter that provides the approximate inverse of the channel response. Usually, the attributes of the channel are unknown or changes over time. This requires an adaptive equalizer structure that adapts the filter coefficients by using some adaptive algorithm. This could either be done using a training sequence as in the least mean square (LMS) method or without a training sequence as in the constant modulus and multi modulus algorithm (CMA,
3.5.1 Least Mean Square Algorithm

The LMS algorithm [8] is a decision feed-back method that uses a training sequence of known length. The training sequence consists of a finite amount of transmitted symbols. The transmitted symbols in the training sequence are then compared to the received symbols and the error between them are calculated. A number of filter weights are updated using the error in a manner to converge to the optimum filter weights for the received symbols. The number of transmitted symbols that are needed in the training sequence depends on how fast the filter weights converge to the optimum weights.

The algorithm starts with small weights stored in a block, usually the middle weight is one and the rest is zero. The input symbols are read in blocks of a predetermined length and compared to the training sequence. After each calculation of the error between the transmitted symbol and the received symbol, the weights are updated by finding the gradient of the mean square error. For example, if the gradient is positive it means that the error will increase at the next iteration, this means that the weights are reduced to keep the amplitudes level as constant as possible. The equations used by the least mean square equalizer to update the filter weights can be seen in equation 3.16 to 3.20.

\[
\begin{align*}
\text{Weights} : \mathbf{w}[n] &= [w_0 w_1 \ldots w_{n-1}]^T \\
\text{Signal Input} : \mathbf{x} &= [x_n x_{n-1} \ldots x_{n-N+1}]^T \\
\text{Filter Output} : \mathbf{y}[n] &= \mathbf{w}^T \mathbf{x} \\
\text{Error Signal} : \mathbf{e}[n] &= \mathbf{d}[n] - \mathbf{y}[n] \\
\text{Weight Update} : \mathbf{w}[n+1] &= \mathbf{w}[n] + \mu \mathbf{e}[n] \mathbf{x}
\end{align*}
\]

In the equations above,

- \( N \) is the input block length.
- \( \mathbf{w}[n] \) stores the filter weights.
- \( \mathbf{x}[n] \) stores the currently read symbols.
- \( \mu \) is the step size of the algorithm.
- \( \mathbf{d}[n] \) stores the training sequence for \( \mathbf{x}[n] \).

\( \mathbf{d}[n] \) can either be transmitted along with the signal or calculated as the closest constellation point to the received symbol \( \mathbf{x}[n] \).

The advantages of the LMS-method are that it is simple to implement and that it has a stable and robust performance against different signal conditions. The disadvantages are that is that it converges slowly and that a training sequence is needed which in many cases is not provided.
3.5.2 Constant Modulus Algorithm

CMA [10] is a method that improves system bandwidth efficiency by avoiding the usage of a training sequence. The objective of this algorithm is just as the LMS-method to update a number of filter weights. The algorithm works exactly like the LMS-method except that the calculation of the error signal has been replaced. Instead of trying to achieve that $y[n]$ is equal to $d[n]$ to minimize the error, the goal is now to update the weights so that $|y[n]| = R$ for all $n$. The calculation of the error signal used by the CMA can be seen in equation 3.21.

$$e[n] = y[n](R - |y[n]|^2)$$ (3.21)

$R$ is a constant adapted to the modulated type used. $R$ is calculated using equation 3.22 below.

$$R = \frac{\text{mean}(\Re(C)^4)}{\text{mean}(\Re(C)^2)}$$ (3.22)

$C$ is a vector containing all constellation points used by the modulation type.

The advantages of the CMA-method are that it uses adaptive tracking of the sources and is still quite easy to implement. Some disadvantages are that it is noisy, needs a small step size which means it is slow and has the possibility of convergence to local minimum.

3.5.3 Multi Modulus Algorithm

The MMA [13] uses the same filter weight update as the LMS and CMA except that the calculation of the error signal has been replaced. In the MMA, the error estimation for the real and imaginary parts are separately compared to $R$ which makes the MMA more suitable for M-QAM where $M >= 16$, i.e. modulation types where many different amplitudes occurs. See equation 3.23 for the error calculations used by the MMA.

$$e[n] = \Re(y[n])(R - \Re(y[n]^2)) + \Im(y[n])(R - \Im(y[n]^2))$$ (3.23)

$R$ still is the calculated value from equation 3.22.

The advantages of the MMA in comparison to the CMA is that it works better for M-QAM where $M >= 16$ and that is has less possibility of convergence to local minimum than the CMA.

This algorithm can be structured and described as in figure 3.15.
Figure 3.15: Multi modulus algorithm structure.
Part II

Implementation and Results
In this chapter, the implementation of the QAM receiver is discussed. The most suitable algorithms of the previously presented have been picked out and implemented.

The structure of the implemented QAM receiver is displayed in figure 4.1.

![Figure 4.1: Structure of the implemented QAM receiver.](image)

The various blocks in the QAM receiver were implemented in Matlab. The implemented algorithms were then tested using both synthesized and real data. The synthesized data was created in Matlab according to the ETSI DVB-C standard [2]. Using the synthesized data, the amount of timing offset, carrier offset and AWGN could be adjusted in order to test the implemented algorithms’ performance.

The real data was created and transmitted by the DTU-215 from Dektec which modulate signals according to the DVB-C standard and then received using the DTA-2131 from Dektec. The DTU-215 is a USB-2 VHF/UHF (Very high frequency / Ultra high frequency) modulator and the DTA-2131 is a VHF/UHF receiver. The received data was recorded and retrieved by Matlab. This transmitting and receiving of real data will be furtherly discussed in section 4.8.
The rest of this chapter will describe the implementation of the various blocks in the QAM receiver.

## 4.1 Automatic Gain Control

An automatic gain control is already implemented in the DTA-2131 which makes it unnecessary to implement, test and analyze an automatic gain control in Matlab.

## 4.2 Multiply Filter Divide

To recover most of the carrier frequency offset, a version of the multiply filter divide method described in section 3.4.1 is used. Since the goal is to receive 64-QAM modulation data and demodulate it, the squaring method will not work. Therefore, the power of four method will be used. This means that input signal is squared twice in order to get an amplitude peak at four times the carrier frequency offset in the frequency domain. By using the fourier transform on $|S[n]|^4$ where $S$ is the input signal, the carrier frequency offset can be retrieved by calculating the frequency where the maximum amplitude occurs. This carrier frequency offset is then divided by four and also divided by the number of input samples in order to calculate the carrier frequency offset between each sample. $|S[n]|^4$ when it has been centered and normalized can be seen in figure 4.2.

![Figure 4.2: Peak at four times the relative carrier offset. The x-axis shows the frequency divided by the symbol frequency.](image)

The calculated frequency offset is used to de-rotate the constellation diagram by multiplying the signal as displayed in equation 4.1.

$$R[n] = S[n] \cdot e^{(i2\pi - \theta_{rel} T)}$$

(4.1)
Where,

- \( S[n] \) is the incoming signal, i.e. the signal from the AGC.
- \( \theta_{rel} \) is the frequency offset
- \( T \) is the sample period
- \( R[n] \) is the de-rotated and carrier frequency semi-synchronized signal.

This step is necessary to perform since if the carrier frequency offset is too large in the decision directed carrier recovery, it will not be able to recover the carrier frequency and phase offsets.

### 4.3 Root Raised Cosine Filter

Since the ETSI DVB-C standard [2] uses the root raised cosine filter as matched filter described in section 3.2.2, a root raised cosine filter was implemented in Matlab. According to that standard, there are a number of requirements for the implementation of the filter. The theoretical function of the root raised cosine filter is described in the DVB-C standard and is defined by the following equation.

\[
H(f) = \begin{cases} 
1 & \text{for } |f| < f_n(1 - \alpha) \\
\sqrt{\frac{1}{2} + \frac{1}{2} \sin\left(\frac{n}{2f_n} \frac{(f_n - |f|)}{\alpha}\right)} & \text{for } f_n(1 - \alpha) < |f| < f_n(1 + \alpha) \\
0 & \text{for } |f| > f_n(1 + \alpha)
\end{cases}
\] (4.2)

\( \alpha \) is the roll-off factor, which in this case is 0.15. \( f_n = \frac{R_{s}}{2} \) where \( R_{s} \) is the symbol rate. The filter also requires a pass-band up to 0.85 * \( f_n \) and that the out-band rejection outside of 1.15 * \( f_n \) must be greater than 43\( dB \).

To design the filter, the following parameters need to be set.

**Roll-off Factor** The roll-off factor \( \alpha \) is set to 0.15 according to the standard.

**Oversampling Factor** The oversampling factor is set to the number of samples per symbol at the input of the filter. In this case the over-sampling factor is 2.

**Filter Length** Using a filter design function in Matlab it is possible to calculate the number of filter taps required to fulfill the requirements. Using these parameters and the theoretical function of the filter, it is implemented in Matlab.

The shape of the filter can be seen in figure 4.3.
4.4 Modified Gardner Algorithm for Timing Recovery

The modulation type usually used by DVB-C is the 64-QAM. Therefore, the modified Gardner algorithm described in section 3.3.2 was chosen to perform the timing recovery. The method is implemented using a second order phase locked loop and a TED. The structure of the method can be seen in figure 4.4.

Between the root raised cosine filter output and the input to the timing recovery scheme, the signal is up-sampled from 2 samples per symbol to 8 samples per symbol using a fractional interpolator. This is done because the interpolator in the recovery scheme uses a linear interpolator and without a large number of samples per symbol, a linear interpolator will not provide sufficient accuracy.

**Interpolator** The interpolator uses the timing offset provided by the controller to calculate the correct symbol by interpolating between the 8 samples read from the signal. The integer part of the timing offset decides between which 2 of the
8 read samples to perform the interpolation and the fractional part decides were between the two selected samples the symbol is located. In figure 4.5 and 4.6 the basic idea of the interpolator is shown.

The symbol is then calculated by first deciding which two samples to be used and then interpolating between these samples using the fractional part of the timing offset $\mu$. See equation 4.3.

$$R = (1 - \mu)X(\lfloor \mu \rfloor) + \mu X(\lceil \mu \rceil)$$  \hspace{1cm} (4.3)

$R$ is the received symbol, $X$ are the current 8 read samples and $\mu$ is the timing offset.

In figure 4.7, an example where the timing offset is set to 2.3 is shown.
In the example, the timing offset is 2.3. The integer part tells us that the 2 samples to be used are $S_3$ and $S_4$. The symbol is calculated using the fractional part, 0.3, of the timing offset in combination with the two samples using equation 4.4.

$$R = S_3(1 - 0.3) + S_40.3 = S_30.7 + S_40.3$$  \hspace{1cm} (4.4)

**Timing Error Detector** The TED is as stated the modified version of the Gardner algorithm. During the start of a new iteration, 8 samples from $t - \frac{T}{2}$ to $t + \frac{T}{2} - 1$ and 8 samples from $t$ to $t + T - 1$ are read. The first 8 samples are used together with the previously calculated timing offset by the interpolator to calculate the sample between the symbols and the second 8 samples are used together with the current timing offset to calculate the symbol.

The last symbol, the middle sample and the new symbol are then used by the TED to calculate the next timing error using equation 4.5 below.

$$e = \Re \{(S_{\text{last}} - S_{\text{curr}})(S_{\text{mid}} - \frac{S_{\text{last}} + S_{\text{curr}}}{2})\}$$  \hspace{1cm} (4.5)

Where $e$ is the current error, $S_{\text{last}}$ is the last sample, $S_{\text{mid}}$ is the middle sample and $S_{\text{curr}}$ is the current sample.

In figure 4.8, an example of how the reading of the samples are performed and how they are used by the interpolator to calculate the symbols are shown.
4.4 Modified Gardner Algorithm for Timing Recovery

Figure 4.8: Reading of the samples for the implemented modified Gardner TED.

Loop Filter  The loop filter is a second order phase locked loop described by figure 4.9.

![Second order phase locked loop diagram]

Figure 4.9: Second order phase locked loop.

It consists of an integral part and a proportional part. The integral part tracks the variation between the errors by using the previously calculated error and a gain factor. The proportional part tracks the fixed error using the error and a gain factor. The loop in the integral part must be faster than the main loop in order to provide the correct value to the correct offset.

The filter gains decide how fast and robust the loop is. A good relation between the loop gains is

\[ K_i = \frac{K_p K_p}{4} \] (4.6)

Using these relations the following equations show the loop filter in the recovery scheme.

\[ \mu = \mu + \omega + K_p e \] (4.7)
\[ \omega = \omega + K_i e \] (4.8)
Implementation

\( \mu \) is the timing offset, \( \omega \) is the value from the integral path and \( e \) is the previously calculated error.

**Controller** The controller makes sure the offset does not go outside the boundaries. The timing offset between the 8 read samples can only lie between 0 and 7, i.e. if the offset is larger than 7, 8 new samples are read and the offset is adjusted.

### 4.5 Multi Modulus Algorithm for Equalization

Since no training sequence is available the chosen equalizer is the multi modulus algorithm equalizer described in section 3.5.3. The multi modulus is a version of the constant modulus algorithm that is more suitable for QAM-constellations with many constellation points. And since the goal is to demodulate 64-QAM data, multi modulus was seen as the most suitable to use.

The input to the equalizer is the output from the timing recovery scheme. The structure of the equalizer can be seen in figure 4.10 below.

![Figure 4.10: Structure of the MMA equalizer.](image)

The number of filter weights and the block length used is 18. The input signal is read to by the equalizer, block by block and multiplied with the current 18 filter weights to get \( y[n] \). \( y[n] \) is then used to update the filter weights according to equation 4.9 and 4.10.

\[
e[n] = \Re(y[n])(R - \Re(y[n]^2)) + \Im(y[n])(R - \Im(y[n]^2))
\]

\[
w[n + 1] = w[n] + \mu e[n]x
\]

Where \( w[n] \) is the vector containing the 18 filter weights, \( \mu \) is the step size, \( e[n] \) is the calculated error and \( x \) are the 18 currently read input symbols.
When all of the symbols in the signal have been fed to the circuit, the filter weights have hopefully converged to the correct values and the weights are used to filter the signal in order to get a properly equalized output signal.

### 4.6 Decision Directed Phase Locked Loop for Carrier Recovery and the LMS Equalizer

For the carrier recovery, the decision directed phase locked loop described in section 3.4.2 was used. Since the timing recovery is performed prior to the carrier recovery, decision directed phase locked loop is a good way to perform the carrier recovery. In order to use a decision directed phase locked loop, the closest constellation point to each symbol must be calculated. These decisions can also be seen as a training sequence which makes it possible to include a LMS-equalizer described in section 3.5.1 in the loop. The structure of the loop containing both decision directed carrier recovery and a LMS equalizer is shown in figure 4.11.

![Figure 4.11: Structure of the decision directed carrier recovery and LMS phase locked loop.](image)

The loop reads one symbol at a time from the signal and calculates the closest constellation point using an algorithm that first controls which quadrant the symbol belongs to and then, using mean square error, calculates the closest constellation point in that quadrant. The following equations are then used to calculate the offset between the constellation points and the symbol.
The calculated error is fed to the loop filter that tracks the frequency and phase offsets using

\[
\theta[n + 1] = \theta[n] + \beta S_{error} \tag{4.14}
\]

\[
\phi[n + 1] = \phi[n] + \theta[n] + \alpha S_{error} \tag{4.15}
\]

\(\theta\) and \(\phi\) are the frequency and phase offsets and \(\alpha\) and \(\beta\) are the gain factors of the loop filter. They are calculated using:

\[
\alpha = \frac{4d b_w}{1 + 2d b_w + b_w^2} \tag{4.16}
\]

\[
\beta = \frac{4b_w^3 b_w}{1 + 2d b_w + b_w^2} \tag{4.17}
\]

\(b_w\) and \(d\) are the bandwidth and damping factor of the phase locked loop.

The new calculated frequency and phase offsets are then used to de-rotate the next read symbol by multiplying the symbol with

\[
\text{Rot}[n] = \cos(\phi + \theta) + i \sin(\phi + \theta) \tag{4.18}
\]

\[
S_{rotated}[n] = S[n]\text{Rot}[n - 1] \tag{4.19}
\]

\(S_{rotated}[n]\) is the rotated symbol, \(S[n]\) is the read symbol and \(\text{Rot}[n-1]\) is the previously calculated rotation.

During these calculations, the LMS equalization is also performed. The LMS uses the current symbol and the 15 previous symbols in combination with the decisions for each of these symbols to update filter taps.

\[
\text{LMS}[n]_{error} = S[n] - S[n]_{decision} \tag{4.20}
\]

\[
f_{real} = \Re(f) - \mu \Re(\text{LMS}[n]_{error}) \tag{4.21}
\]

\[
f_{imag} = \Im(f) - \Im(i\mu \Im(\text{LMS}[n]_{error})) \tag{4.22}
\]

\[
f = f_{real} + f_{imag} \tag{4.23}
\]
In order to be able to test the implemented algorithms, both synthesized and real M-QAM data were used. The synthesized data were created in Matlab and the real data were transmitted and recorded using Dektec devices.

4.7.1 Synthesized QAM-data

For the testing with synthesized data to be performed properly, the synthesized data is put through an AWGN-channel and includes both fixed and variable timing offsets as well as phase and frequency carrier offsets. The structure of the synthesized data is displayed in figure 4.12 below.

![Figure 4.12: Structure of the implemented data synthesizer.](image)

4.7.2 QAM Modulation

First, a random bit stream is created. The bits are then, depending on the modulation type paired in K-bit sequences, where K is the number of bits per symbol. They are then mapped on the constellation diagram in order to get the proper I and Q symbols to transmit. Figure 4.13 shows 64-QAM data prior to root raised cosine filtering.
4.7.3 Root Raised Cosine Filter

To simulate a real DVB-C signal, the I and Q samples are put through a root raised cosine filter with a roll-off factor of 0.15 and stop band attenuation of 43 dB. Figure 4.14 shows 64-QAM data after the data has been root raised cosine filtered.

![Figure 4.13: 64-QAM data.](image1)

![Figure 4.14: Root raised filtered 64-QAM data.](image2)

4.7.4 Channel Errors

After the root raised cosine filtering, the I and Q symbols are fed through the AWGN-channel with tunable SNR to simulate a real channel. In this case, the SNR (in dB) is in this case the ratio of the signal power to the noise power. In figure 4.15, the signal after the AWGN channel with SNR set to 30dB is shown.
4.7.5 Carrier Offsets

Carrier offsets are applied to the signal by multiplying the signal with

$$e^{i(2\pi F t + \phi)}$$

(4.24)

Where $F$ is the frequency offset that varies with time and $\phi$ is the phase offset.

In figure 4.16 the signal after the carrier frequency and phase offsets have been added is displayed.

Figure 4.15: RRC-filtered 64-QAM data put through an AWGN-channel with SNR 30dB.

Figure 4.16: 64-QAM data with carrier frequency and phase offset.
4.7.6 Timing Error

Timing offset is applied by delaying the signal with a variable delay. For example:

- $TO_{fixed} = 1.3$
- $TO_{variable} = 0.001$

Where $TO_{fixed}$ and $TO_{variable}$ are the fixed and variable timing offset.

The first sample in the stream is delayed by 1.3 samples, the second one is delayed with $1.3 + 0.001$ etc. The delay is applied to the signal by using a variable fractional delay channel from the digital signal processing toolbox in Matlab.

When all of the offsets and errors have been added, the output is a synthesized version of the real data and is used to test the various methods and algorithms implemented. In figure 4.17 the signal with all offsets and errors added to it is displayed.

![Figure 4.17: 64-QAM data with all errors and offsets applied.](image)

4.8 Transmitting and Receiving real QAM-data

The transmitting and receiving of real DVB-C data were made using two different Dektec devices.

4.8.1 Transmitter

The transmission of DVB-C signals was performed using Dektec DTU-215. The DTU-215 was connected to a laptop where the Dektec software called StreamXpress was installed. The parameters to be set in the StreamXpress:

**Modulation Type** The modulation type the data is modulated with. Tests are performed with 16-QAM, 64-QAM and 256-QAM.
**Bit Sampling Rate**  This value represents the number of bits transmitted per second.

**Baud Rate**  Set to the number of symbols per second. The number of symbols per second depends on the modulation type and the bit sampling rate.

**Carrier Frequency**  The carrier frequency sets the frequency of the carrier waves in the transmitter.

### 4.8.2 Receiver

The receiving of the real DVB-C signals was performed using Dektec DTA-2131. The Dektec DTA-2131 was connected to the computer provided by Agama. The Linux based software DtRecord was used to record the incoming signal. The parameters to be set in order to receive the correct signal:

- **Modulation Type**  Set to IQ, since the demodulation needs IQ-samples to work.
- **Modulation Carrier Frequency**  This is set to the carrier frequency of the transmitter. They must match in order to receive the correct signal.
- **IQ Bandwidth**  The bandwidth needed to receive the signal depends on the baud-rate in the transmitter. Since the root raised cosine filter has a roll-off factor set to 0.15, the bandwidth needed to receive a transmitted signal with baud-rate $X$ is $X \times 1.15$.
- **IQ Sample Rate**  Depending on how many samples per symbol that is needed, this rate is set to an integer multiple of the symbol rate, i.e. the baud-rate of the transmitter. For example, if the baud-rate in the transmitter is set to 5 Msymbols per second and the demodulator needs 2 samples per symbol, the IQ sample rate is set to 10 Msymbols per second.

### 4.9 Summation

In this section, the implementation of all the algorithms and methods used by the demodulator have been discussed. The data that were used to test the implemented algorithms were also introduced. So, the demodulator is implemented and both synthesized IQ-samples and real IQ-samples are created and recorded.
Using the synthesized and real data, the implementations discussed in chapter 4 was tested. The testing starts with the synthesized data where it is possible to add and remove the different errors and offsets. The final test is then to demodulate the real QAM data recorded using the DTA-2131 from Dektec.

5.1 Measurements

In order to determine the performance of the algorithms for the various errors and offsets, some measurement levels are needed. The modulation error ratio (MER) and bit error rate (BER) are two common measurements.

5.1.1 Modulation Error Ratio

The modulation error ratio is a measurement used to quantify the performance of a digital communication system. In the ideal case, the transmitted IQ-samples are the ones that are received in the receiver. This would mean that the received IQ-samples are located exactly at the constellation points used in the transmitter and would cause the MER to go towards infinity.

Synthesized Data

For the synthesized data, the MER (in dB) can be calculated simply by comparing the transmitted symbols with the received symbol.

For synthesized IQ-samples, the MER is calculated using equation 5.1.
\[ MER_{dB} = 10 \log_{10} \frac{\sum_{i=1}^{N} (I_{received}^2 + Q_{received}^2)}{\sum_{i=1}^{N} [(I_{transmitted} - I_{received})^2 + (Q_{transmitted} - Q_{received})^2]} \] (5.1)

\( N \) is the length of the signal, \( I_{received} \) and \( Q_{received} \) are the received IQ-symbols and \( I_{transmitted} \) and \( Q_{transmitted} \) are the transmitted IQ-symbols.

**Real Data**

For real data, the transmitted symbols are not available. In this case, a decision is made for each received symbol. The decision is the closest constellation point in the constellation diagram used by the transmitter. For real IQ-samples, the MER is calculated using equation 5.2.

\[ MER_{dB} = 10 \log_{10} \frac{\sum_{i=1}^{N} (I_{received}^2 + Q_{received}^2)}{\sum_{i=1}^{N} [(I_{decision} - I_{received})^2 + (Q_{decision} - Q_{received})^2]} \] (5.2)

\( N \) is the length of the signal, \( I_{received} \) and \( Q_{received} \) are the received IQ-symbols and \( I_{decision} \) and \( Q_{decision} \) are the decision made on \( I_{received} \) and \( Q_{received} \).

**5.1.2 Bit Error Rate**

The bit error rate is a measurement that compares each bit transmitted with the bit received. To calculate the bit error rate, the number of bit errors is divided by the number of bits transmitted.

\[ BER = \frac{N_{err}}{N_{bit}} \] (5.3)

Where \( N_{err} \) is the number of bit errors and \( N_{bit} \) is the number of bits.

Since the decoding of the encoding performed in the source encoder is not performed in this thesis, the real bit error rate will not be calculated. However, if the decision, i.e. the closest constellation points, of the received constellation point is not the transmitted constellation point, the symbol will be de-mapped to the wrong bits. This causes a bit error that can be measured. Since the decision for the real data is also used as the transmitted symbol, the bit error rate cannot be calculated for the real data. In the synthesized data case, the closest constellation point in the constellation diagram used by the transmitter for each received symbol is the decision. The bits that represent the decision is then compared to the bits that represent the transmitted symbol to calculate the BER.
5.2 Testing on Synthesized Data

Testing on the synthesized QAM-data is performed in several steps. The offsets and errors are added one by one in order to investigate their impact on the MER and BER.

5.2.1 Timing Offset

When no carrier offset is present, the constellation diagram of the received signal looks like a square.

Fixed Timing Offset

When a fixed timing offset is present, the timing recovery circuit in the receiver must lock onto the phase offset in order to sample the signal at the correct instances.

The tracking of a fixed timing offset for a 64-QAM signal can be seen in figure 5.1.

\[\text{Fixed Timing Offset}\]

\[\text{Tracking of a fixed timing offset.}\]

The amount of fixed timing offset does not affect the MER or BER significantly.

Variable Timing Offset

The presence of a variable timing offset causes drifting between the clocks in the receiver and the transmitted, and the receiver samples the signal at the wrong instances. Therefore, the timing recovery circuit in the receiver must lock onto the variable offset in order to sample the signal at the correct instances.

The tracking of a variable timing offset can be seen in figure 5.2.
The amount of variable timing offset affects the MER and BER quite much. When the drifting between the clocks is too large, the phase locked loop cannot lock onto the variable timing offset properly which causes the system to fail and the MER to go towards zero. In figure 5.3, the relation between the amount of variable timing offset and the MER can be seen.

**Fixed and Variable Timing Offsets**

When both a fixed and variable timing offset is present, the first symbol is taken at the wrong instance and the sampling frequency of the receiver is incorrect. The summation of these problems forces the phase locked loop to track both the phase and frequency offsets.
The tracking of the fixed and variable timing offset can be seen in figure 5.4. The variable offset is quite large in this case.

**Figure 5.4:** 64-QAM demodulation: Tracking of both fixed and variable timing offset.

Since it is only the amount of variable timing offset that cause the MER to drop, the relation between the summation of the fixed and variable timing offset and the MER will be exactly as the relation between the variable timing offset and the MER.

### 5.2.2 Carrier Offset

When no timing offset is present but carrier offsets are, the received signal looks like a number of concentric circles. The number of concentric circles depends on the modulation type. See table 5.1.

<table>
<thead>
<tr>
<th>Modulation Type</th>
<th>Concentric Circles</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-QAM</td>
<td>3</td>
</tr>
<tr>
<td>64-QAM</td>
<td>9</td>
</tr>
<tr>
<td>256-QAM</td>
<td>32</td>
</tr>
</tbody>
</table>

*Table 5.1:* Number of concentric circles for the different modulation types.

**Carrier Phase Offset**

When only a carrier phase offset is present, the received constellation points will be a rotated version of the transmitted constellation points. The decision directed phase locked loop must lock onto the phase offset in order to de-rotate the received constellation points.

The decision directed phase locked loop’s tracking of the carrier phase offset can be seen in figure 5.5.
The amount of phase offset does not affect the MER significantly.

**Carrier Frequency Offset**

When only a carrier frequency offset is present, the received constellation points will look like concentric circles. The carrier frequency recovery is performed in two steps. First, the multiply filter divide method is used to recover most of the carrier frequency offsets and then the rest of it is recovered in the decision directed phase locked loop. The multiply filter divide method does not take care of any carrier phase offsets. The decision directed phase locked loop’s tracking of the carrier frequency offset can be seen in figure 5.6. Note that the frequency offset is almost zero since the multiply filter divide has taken care of most of the carrier frequency offset.

**Figure 5.5:** 64-QAM demodulation: Tracking of carrier phase offset.

**Figure 5.6:** 64-QAM demodulation: Tracking of carrier frequency offset.
The amount of carrier frequency offset affects the MER and BER quite much. When the frequency difference between the carrier wave in the transmitter and the local oscillator is too large, the MER will drop. In figure 5.7, the relation between the amount of carrier frequency offset and the MER can be seen.

**Figure 5.7: 64-QAM demodulation: The carrier frequency offset’s impact on the MER.**

**Carrier Frequency and Phase Offsets**

Since the tracking of the phase and frequency are kept separate, the tracking of the phase and frequency will look like figure 5.5 and 5.6 even though both offsets are present. It is only the carrier frequency offset that affects the MER and BER, the presence of both carrier phase and frequency offsets will not change the behavior significantly from figure 5.7.

**5.2.3 Timing Offset and Carrier Offset**

When all of the offsets are applied to the signal, the received signal will basically just look like noise. The recovery of a signal with all offset applied will be displayed for different QAM types.

**16-QAM**

A recovered 16-QAM signal for different amount of noise are displayed in figure 5.8.
Since the 16-QAM has the largest distance between the constellation points of the three modulation types investigated, it is least sensitive to noise. The relation between the SNR and the MER for 16-QAM data can be seen in figure 5.9. The MER is set to zero when the demodulator completely fails to recover the symbols to clearly show at which SNR the demodulator stops working for the modulation type.

Note that the MER follows the SNR quite well except for when the SNR becomes smaller than 14dB. For real DVB-C signals this is not a problem since the SNR for a DVB-C signal is usually around 30 – 35dB. When the SNR is smaller than 14dB, the distance between the constellation points is not large enough and the...
decision for the received symbols will be interpreted as the wrong point. This causes the demodulation to fail completely.

The relation between the SNR and the BER for 16-QAM data can be seen in figure 5.10.

![BER vs SNR](image)

**Figure 5.10:** Synthesized 16-QAM: SNRs impact on the BER.

### 64-QAM

A recovered 64-QAM signal for different amount of noise are displayed in figure 5.11.

![64-QAM Signal](image)

**Figure 5.11:** Recovered 64-QAM for some different SNRs.

The 64-QAM has a smaller distance between the constellation points which makes
it more sensitive to noise. The relation between the SNR and the MER for 64-QAM data can be seen in figure 5.12.

![MER vs SNR graph](image)

*Figure 5.12: Synthesized 64-QAM: SNRs impact on the MER.*

Note that the demodulator fails already at a SNR of 18dB instead of a SNR of 14dB as in the 16-QAM case. The MER peak lies around 30dB.

The relation between the SNR and the BER for 64-QAM data can be seen in figure 5.13.

![BER vs SNR graph](image)

*Figure 5.13: 64-QAM demodulation: SNRs impact on the BER.*

### 256-QAM

A recovered 256-QAM signal for different amount of noise are displayed in figure 5.14.
5.2 Testing on Synthesized Data

Figure 5.14: Recovered 64-QAM for some different SNRs.

The 256-QAM has the smallest distance between the constellation points of the three types. In figure 5.15, the relation between the MER and SNR for 256-QAM is displayed. The demodulator fails when the SNR is smaller than 27dB. The MER peak at around 32dB is probably caused by the linear interpolator in the timing recovery phase locked loop. With a better interpolator this can be improved.

Figure 5.15: Synthesized 256-QAM: SNRs impact on the MER.

The relation between the SNR and the BER for 256-QAM data can be seen in figure 5.16.
5.3 Testing on Real Data

For testing on the recorded real DVB-C data, a recording was performed for each of the three modulation types. The received symbols basically look like rotated noise, see figure 5.17 for the received 16-QAM data.

5.3.1 16-QAM

The demodulated data, timing offset tracking, phase offset tracking and frequency offset tracking can be seen in figure 5.18 to 5.21.
5.3 Testing on Real Data

The MER of the received 16-QAM signal, calculated using the decision for each symbol is

\[ MER = 29.5\text{dB} \]

5.3.2 64-QAM

The demodulated data, timing offset tracking, phase offset tracking and frequency offset tracking for 64-QAM data can be seen in figure 5.22 to 5.25.
The MER of the received 64-QAM signal is

\[ \text{MER} = 29.6 \text{dB} \]

5.3.3 256-QAM

The demodulated data, timing offset tracking, phase offset tracking and frequency offset tracking for 64-QAM data can be seen in figure 5.26 to 5.29.
5.3 Testing on Real Data

Figure 5.26: Real data 256-QAM: Recovered constellation diagram.

Figure 5.27: Real data 256-QAM: Timing offset tracking.

Figure 5.28: Real data 256-QAM: Carrier phase offset tracking.

Figure 5.29: Real data 256-QAM: Carrier frequency offset tracking.

The MER of the received 256-QAM signal is

\[ \text{MER} = 28.7 \text{dB} \]
6.1 Summary

6.1.1 Objectives

The main goal of the research was to develop, implement and test a DVB-C demodulator. To achieve this goal, a large number of adaptive algorithms to perform timing recovery, carrier recovery and equalization were investigated and some of them tested. All of the implemented methods operate to quickly synchronize the received symbols without any knowledge about the clocks in the transmitter or the channel. The implemented methods are described in detail and the performances of the different methods were tested.

6.1.2 Results

The results were quite satisfying since the main goal were successfully achieved. However, the extractions of some metrics were not quite so successful since the knowledge about the transmitted signal was almost zero. This made it impossible to calculate metrics such as the impulse response and the BER for the real data.

With a bit more time, the MER of the real data would be improved by implementing a better interpolator in the timing recovery phase locked loop. The decoding of the source encoding would also be performed in order to be able to calculate the real BER value.

6.2 Further Research

During the thesis, a number of possible opportunities for further study came up.
6.2.1 DVB-S

It would be quite easy to expand the receiver to demodulate DVB-S signals as well. Since the DVB-S standard uses the 8-PSK modulation type, which is a modulation type that uses eight constellation points along a circle with a certain amplitude it is quite easy to recover the timing, phase and frequency offsets.

6.2.2 Increasing the MER

Using the implemented algorithms, the MER has a peak around 32 dB. This is probably because of the linear interpolator in the timing recovery phase locked loop. Using a better interpolator, for example a polyphase interpolator, the MER could be increased.

6.2.3 Source Decoder

In order to calculate the transmitted bits, a source decoder must be implemented. This decoder consists of for example a Reed-Solomon decoder, a De-interleaver and a differential decoder. With a proper implemented source decoder, the real BER can be calculated.
Bibliography


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnt som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a period of 25 years from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Viktor Eriksson