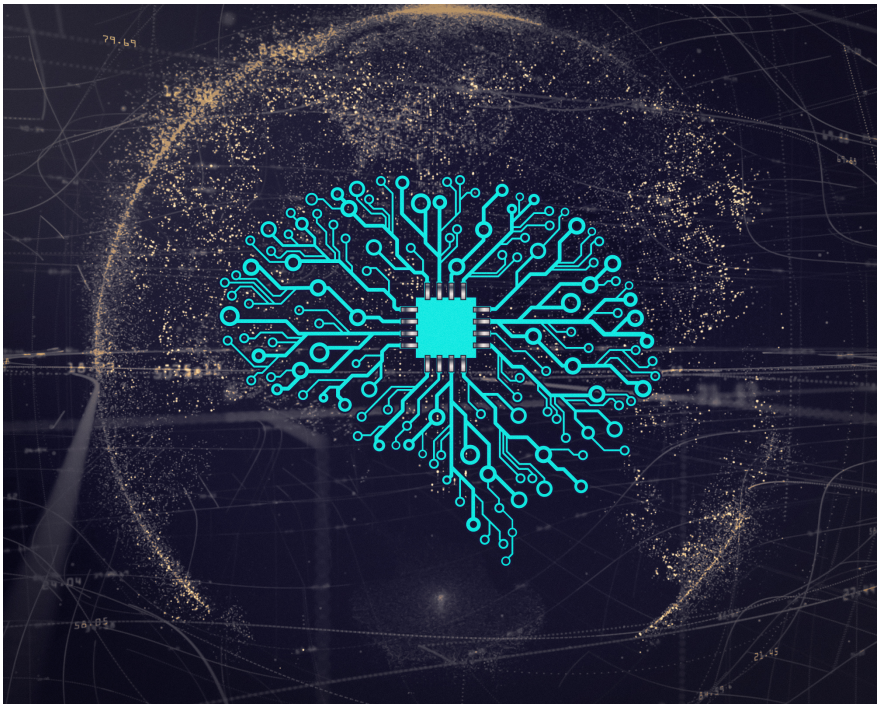
CHALMERS
UNIVERSITY OF TECHNOLOGY



Neural Network Based Receivers for Microwave Backhaul Links

Master's thesis in Communication Engineering

NAHID MOHEQ
JINGYI QIN

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020:

Neural Network Based Receivers for Microwave Backhaul Links

NAHID MOHEQ
JINGYI QIN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2020

Neural Network Based Receivers for Microwave Backhaul Links
NAHID MOHEQ
JINGYI QIN

Supervisors: Cristian Czegledi, Ericsson
Martin Sjödin, Ericsson
Ulf Gustavsson, Ericsson
Andreas Buchberger, Department of Electrical Engineering
Examiner: Thomas Ericsson, Department of Electrical Engineering

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: [1]

Typeset in LaTeX
Gothenburg, Sweden 2020

Neural Network Based Receivers for Microwave Backhaul Links
NAHID MOHEQ
JINGYI QIN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The report aims to investigate how the performance of a standard receiver compares to one implemented with machine learning (ML). This was done by modelling different impairments and then comparing the performance of the two receiver implementations. Three different types of impairments were implemented: phase noise (PN), in-phase (I) quadrature (Q) imbalance and non-linearities originated from the power amplifier.
The first part of the report aims to summarize the underlying theory of communication systems and machine learning. This theory is then used to derive and describe how to implement the transmitter, channel model, standard receiver and ML-based receiver. Furthermore, as to increase the transparency of the work it explains how the python libraries Keras and TensorFlow are used in the ML receiver.
To summarize the result, the ML receiver worked well when compensating for non-linearities and IQ imbalance impairments but the implementation failed to compensate for PN impairment. This could be due to the simple ML implementation and a further study could investigate the use of a neural network (NN).

Keywords: Communication system, machine learning, standard receiver, NN-based receiver, additive white Gaussian noise (AWGN) channel, IQ imbalance, non-linearities, PN.

Acknowledgements

In this section we would like to thank some people that helped us in completion of this report.
Firstly, our deepest thanks is given to our supervisors from Ericsson, Martin Sjödin, Cristian Czegledi, and Ulf Gustavsson. We are greatly thankful for your attentive engagement in every step of the way, weekly meetings and your insightful comments and ideas in every step of the way. We appreciate that you trust to select us to proceed with this project, I hope we full-filled your expectations.
We want to express our deepest gratitude to Martin Sjödin, whom we have learned a lot related to python and machine learning from. It was not easy to start a Master Thesis in a topic where you have no background in both of these area. Thank you for your patience toward our stupid mistakes and holding our hand in each step of the way. Cristian Czegledi, you were always around to help us about every small matter and your patience was really appreciative. You were there always whenever we got stuck and your encouragement and tips were the light to our path. Ulf Gustavsson, thank you for helping us with explaining more about the project and sharing the resources and your time.
Dear Andreas Buchberger, Thank you for your constructive inputs to the model and progress of our report and smart feed-backs through our report. Thomas Eriksson, thank you for your involvement throughout the project and positive feed-backs in our final report. Franserik Isaksson and Patrik Bannet, we appreciate your comments and feedback to improve our report and make it lucid to understand.
Yibo Wu, thank you for helping us in first steps of our thesis, even though you just started your PhD and you had a lot in your plate. You assisted us in time of our need where we were new and everything seem really hard.
Last, not least we would love to thank our family and all our friends, especially our parents whom support us spiritually in every step of the way, we hope that we have made you proud.

Nahid Moheq & Jingyi Qin, Gothenburg, October 2020

Contents

List of Figures

List of Tables

Acronyms

AI artificial intelligence
AWGN additive white Gaussian noise

BER bit error rate
BP back propagation
BPS blind phase search

CNN convolutional neural network
CRE constant radius estimation

DL deep learning
DSP digital signal processing

I in-phase
ISI inter symbol interference

MIMO multiple-input multiple-output
ML machine learning
MLE maximum likelihood estimation
MSE mean squared error

NN neural network

PN phase noise

Q quadrature
QAM quadrature amplitude modulation

RRC root-raised cosine

SNR signal-to-noise ratio

1
Introduction

This chapter aims to introduce the field of study by summarizing the background to communication systems and similar studies. Furthermore, it states the purpose of this master thesis and divides it into clearly described objectives while also noting on the points limiting the scope of the report. Lastly, it addresses the social, ethical and ecological impacts this study could imply.

1.1 Background

1.1.1 The structure of a communication system

Communication can be defined as the reliable transmission of a message from one point to another through a channel. In practice, the design and analysis of communication systems is an advanced field of modeling various channel types [2] and utilizing some techniques to compensate for its channel and hardware impairments [3]. Then, using digital signal processing (DSP) tools, the detection of a transmitted signal is optimized through this channel.

With the rapid development of communication systems, a major challenge in the fifth generation wireless systems arises with increased data-rates during the backhaul from the radio base station to the transport network. In order to cope with these demands, a point-to-point microwave link with dense modulation formats, large bandwidth, and multiple-input multiple-output (MIMO) is commonly used. This further impacts the demands on signal purity to a higher level. In order to compensate for this problem, the wireless communication network system needs to be constantly improved.

In the conventional communication system, the transmitter and receiver are divided into separate processing blocks, e.g., coding, modulation, demodulation, equalization, and decoding. Each of them is optimized individually based on mathematical modeling before merging to get the complete model. This type of approach has the advantages of being robust and efficient. However, in this conventional system, there exists a large number of configuration parameters in the mathematical models used to optimize every block, and each parameter may cause the accuracy of the demodulated signal to decrease [4].

Besides, optimizing each block separately and then combining them may not result in the optimal solution. By changing the channel environment, there must be done a large workload of parameter adjustment to match with the new channel environment, which this leads to poor robustness of the demodulation algorithm [5].

While if the whole system is employed as a whole unit and then optimized, the number of parameters to be adjusted may decrease. Therefore, due to uncertainty of the channel environment, it is necessary to design a flexible and adaptive demodulator that is able to adjust the parameters [6].

1.1.2 Conventional demodulation algorithms

In recent years, the rapid development of communication technology has also promoted the development of demodulation technology towards high accuracy. Experts and scholars have studied various linear and nonlinear signal demodulation and equalization algorithms to suppress inter symbol interference (ISI), and achieve a low bit error rate (BER) signal demodulation.

As early as in 1996, [7] proposed a nonlinear decision adaptive demodulation algorithm, which is compatible with differential coherent phase shift keying for frequency selective fading channels. This algorithm solves the problem that the traditional equalizers cannot track the phase change in the selective fading channels. Then, [8] proposed an improved maximum likelihood estimation (MLE) algorithm, which solves for the problem that the traditional MLE algorithm used in spaced multi-path channels results in high BER. This method not only reduces the computational complexity, but also has almost the same accuracy as the least mean square error estimation method.

A hybrid carrier modulation system is proposed in [9], which consists a discrete-time Fourier transform demodulation and low-complexity minimum mean squared error equalization. The proposed system can reduce the inter-carrier interference on the dual-select fading channel and has a lower complexity compared to the single carrier modulation and the orthogonal frequency division multiplexing system in high signal-to-noise ratio (SNR) region.

A new decoding scheme on dual-select channels is considered in [10]. This scheme first performs a channel estimation, and then performs signal demodulation based on the obtained channel estimation. The proposed algorithm has good demodulation performance for MIMO single-carrier frequency domain equalization systems on dual selective channels. A new combined MLE and maximum posterior estimation network is proposed in [11]. This kind of network can perform a channel estimation in the receiver. Although the estimation accuracy is significantly improved using this network, but the system is complex and it is difficult to implement.

In most of the above demodulation algorithms, the system requires to know the channel statistics, which in most cases it is assumed to be additive white Gaussian noise (AWGN). For example, in [12, 13, 14], they all assume that each receiver can accurately estimate the fading coefficient. However, actual wireless communication channels may suffer from other complex impairments such as multi-path fading, impulsive noise, and scattering or continuous interference [15].

Besides, the accuracy of channel estimation is limited due to the finite length of the training sequence [16]. In particular, for fast fading scenario, where the fading coefficient changes rapidly during the data transmission period, the system cannot obtain accurate channel information and noise distribution functions [16]. Since the receiver cannot accurately estimate the channel information, it is difficult to design

a low BER signal demodulator. However, [17] shows that the requirement of prior knowledge on the channel is not necessary for the demodulation based on machine learning (ML). This provides highly beneficial for designing a high-precision signal demodulator where accurate channel state information cannot be obtained.

1.1.3 Machine learning

Recently, due to the rapid development of computer hardware, ML has also developed accordingly. The purpose of ML is to enable the system learning from experience and improve performance autonomously [18]. ML has a strong learning ability, and can solve the problem of poor demodulation performance caused by a complex and changing channel environment by designing a ML model [19][20].

Deep learning (DL) is a branch of ML. The DL model is a multi-layer stacked neural network (NN) and the calculation result of the previous layer is the input of the next layer [21]. Through multi-layer processing, the original low-level feature representation is gradually converted into an abstract feature representation. This conversion can automatically discover the distribution patterns hidden in the data [22].

1.1.4 Demodulation algorithms based on ML

The prototype of ML is the "M-P neuron model" proposed by psychologist McCulloch and mathematical logician Pittis in the 1940s [17]. This model provides a basis for the subsequent development of ML. However, due to the limitation of computer hardware, ML has not been effectively developed in the last century.

In recent years, computers are able to support a large data throughput, which this leads to a rapid development in the ML research area. However, this leads to another problem, which is the error propagates backwards in the network and it sometimes disperse and cannot converge to a stable state [17]. One of the methods to solve the error divergence problem is weight sharing. In a general NN, the layers are fully connected, and the weight of each connection is an independent parameter [23]. By making a group of neurons use the same connection weights as neurons in another layer, weight sharing achieves the goal of reducing parameters and saving overhead [23]. Convolutional neural network (CNN) adopts this method [23]. Another method solving the error divergence problem is proposed by Professor Hinton of the University of Toronto in Canada in 2006. He proposed a deep belief network [17] which improved the algorithm of the model and break through the bottleneck of the back propagation neural network development.

The researches on the combination of ML and wireless communication are mostly in the application layer, network layer and physical layer. The research on the physical layer of communication systems related to ML can be divided into two categories.

First category uses ML to redesign or replace a module in a traditional system, such as channel estimation [24], media access control [25], automatic modulation classification [26], interference control [27], antenna power allocation [28] and channel decoding [29].

Another category applies the concept of interpreting the end-to-end communication system as an auto-encoder proposed in [19]. In an end-to-end system, multiple functional blocks of the communication system are replaced with neural networks, and the optimization of the entire system is obtained by training a large amount of data. For example, [30] proposed a wireless signal demodulation model based on DL in orthogonal frequency division multiplexing system. This model takes the entire receiver as a single block to be trained. This receiver block is directly connected with the radio frequency receiver through the NN. In [31] and [32], the end-to-end wireless communication system is completely replaced with a DL network. This network is implemented in order to optimize the entire wireless communication system in a global scale to minimize the loss.

Although scholars have made many attempts to apply ML to wireless signal demodulation, most of these methods are not tested in real-time system models [33]. Therefore, it is of great significance to study the performance of the ML based demodulation system in overcoming uncertain factor in practical. Due to the time limitation, this report focuses only on the simulated environment.

1.2 Aim

In this project, the aim is to investigate the impairments in a end-to-end microwave back-haul link using ML, where ML is applied to demodulate the noisy signal and compare the performance to a conventional method. In other words, we are challenging traditional signal processing tools which are used to compensate for various hardware-induced impairments such as additive noise, phase noise, or ripple, with ML.

This aims to give us a deeper insight into the communication algorithms, including how these impairments affect the quality of transmission. Also, it aims to verify whether independently optimized traditional DSP blocks can be outperformed by a NN-based black box. The results of this study are of interest to both, the academia and industry, as they explore a new possible branch of communication systems.

1.3 Problem Description and Objectives

The aim of the report largely revolves around demodulating signals using ML in order to compare the performance to a conventional receiver and in order to achieve this aim the following objectives were derived:

- Design and implement a conventional communication systems including: transmitter, channel and receiver.
- Simulate the channel impairments: AWGN, non-linearity, phase noise, and I-Q imbalance.
- Design and implement a NN-based receiver.
- Compensate for the simulated impairment using the NN-based receiver.
- Compensate for the simulated impairment using the conventional receiver, excluding I-Q imbalance and non-linearities.

- Evaluate the performance of the NN-based receiver by comparing it to the conventional receiver.

1.4 Limitations

In this project all the results are based on a simulated environment instead of the real-time transmission, including the impairments like the AWGN, in-phase (I)quadrature (Q) imbalance, non-linearities, phase noise (PN) and phase offset. Also, this report only replaces the receiver by a NN, not the whole system.

1.5 Social, Ethical and Ecological Aspects

There are enormous advantages in applying ML in many applications and systems but few scientists and artificial intelligence (AI) researchers are concerned about the potential drawbacks of AI [34]. Since AI algorithms are a result of human intelligence, it might be faulty and not robust and causes some problems. Therefore it is very important that researches besides focusing on enhancement of different applications using AI, focus also on the social and security and ethical aspect of AI.

Since the thesis is based around NN and only works in a simulated environment the impact on social and ethical issues is very limited. Regarding ecological aspects, the ML could provide with a more efficient way to design communication systems and thus it can be argued that it potentially has a positive effect.

1.6 Structure of This Report

The following chapters of the report are organized as follows. Section 2 is the introduction to the basic structure of the communication systems which is consisting of a transmitter, a channel, and a receiver. This section explains the theory behind each system blocks, fundamentals of NN and related topics used in this project. Section 3, explains in detail how each block of the system is constructed and how each step of the experiment is carried out. Section 4, presents the results which are different BER curves corresponding to a given impairment which depicts the performance of each section. Besides, this section gives a discussion and explanation on the the comparison between the NN-based receiver and the standard receiver. Section 6 concludes all the results and discussion and gives a brief summary of the whole project, achievements, limitations and future investigation.

2

Proposed Communication System

This chapter aims to describe the theory behind the communication system which is designed and implemented in this thesis. A simple wireless communication system comprises a transmitter, a channel and a receiver. These components will be discussed in the following sections.

2.1 Transmitter

The transmitter modifies the source message for efficient transmission [35]. A block diagram of the transmitter implemented in this report is shown in Figure 2.1. The figure illustrates two blocks: symbol generation and pulse-shaping. The generated input binary bit stream is mapped into M-quadrature amplitude modulation (QAM) symbols. Next the symbols are passed through the pulse-shaping filter where they are interpolated and convolved with a root-raised cosine (RRC) filter to obtain the signal for transmission.

2.1.1 Symbol generation

During the symbol generation procedure in Figure 2.1, a stream of random binary bits is generated as the communication resources. In the communication system, different modulation methods are adopted in order to make full use of communication resources and meet different needs of users. In this thesis, the bit stream is mapped into symbols by a certain modulation method called QAM modulation.

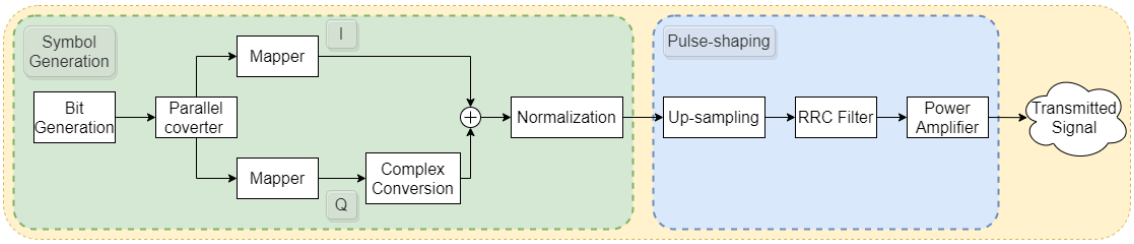


Figure 2.1: Block-diagram of the transmitter divided into symbol generation and pulse-shaping. The generated binary stream is divided into two sets of bits in the parallel converter before assigned by the corresponding amplitude level given to mapper. Then Q is converted to complex value and added by I to obtain the symbol $(I + jQ)$. Finally, the symbol sequence is padded with zero and convolved by the RRC to obtain the transmitted signal.

M-QAM modulation is used in all transmissions in this thesis, as it is a common standard and with high spectral efficiency [36]. QAM modulations make use of both phase and amplitude information, in comparison to the simpler PAM modulation which discards phase-information for a simpler system design.

In M-QAM modulation, each symbol represents $\log_2 M$ bits that the symbol maps to be transmitted and often a constellation diagram is used to visualise all of the QAM symbols. It is a signal vector diagram with each point in the diagram represents a QAM symbol, which is synthesized by two orthogonal vectors. For example, a signal modulated by 64-QAM can be mapped to 64 points in different states in the constellation diagram. A 64-QAM modulated signal can carry 6 bits of information, implying that transmitted bits are grouped six at a time with each 6-bit grouping being translated to a specific symbol. When $M = 16$, the modulation signal constellation is shown in Figure 2.2.

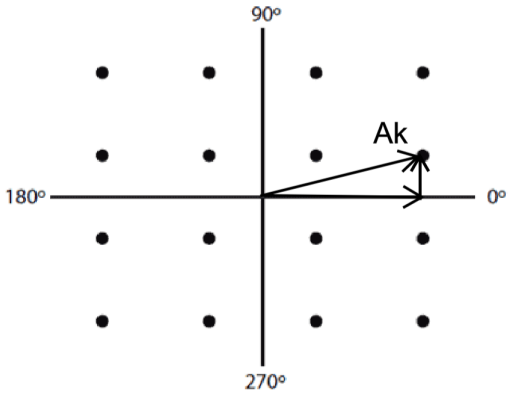


Figure 2.2: Constellation diagram of 16-QAM signals.

As shown in the symbol generation block in Figure 2.1, in this block, the generated binary bits are divided equally into two channels in the serial-to-parallel converter for level conversion, one is for I component and another is for Q component. These two components are independent. After passing through the level shift section, the different combinations of 0 and 1 in the binary stream is converted into different decimal numbers representing voltage levels of each combination. Then the Q component is multiplied by imaginary unit j to be phase-shifted 90 degrees such that it is orthogonal to I component. Finally I and Q are added and normalized, as shown in the following equation:

$$\boldsymbol{s} = \boldsymbol{s}' \frac{1}{\sqrt{\frac{1}{n}\sum |s'|^2}} \tag{2.1}$$

with

$$\boldsymbol{s}' \equiv \begin{bmatrix} s'_1 & s'_2 & \dots & s'_n \end{bmatrix} \tag{2.2}$$

$$\boldsymbol{s} \equiv \begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix} \tag{2.3}$$

where $\boldsymbol{s}'$ and $\boldsymbol{s}$ are the symbols before and after modification respectively, and n is the number of symbols.

2.1.2 Pulse-shaping

The M-QAM symbols obtained in the symbol generation block are then used for pulse-shaping. The goal for pulse-shaping is to make the signal ready to be transmitted through the communication channel within an effective bandwidth. During decoding procedure in the receiver, if the signal's bandwidth becomes larger than the channel bandwidth, it will interfere with the neighbouring channels and will cause ISI. Pulse-shaping controls the signal's spectrum to prevent any type of ISI[3].

The pulse-shaping block contains two sections: up-sampling and a RRC filter, as shown in Figure 2.1. Up-sampling is the process of increasing the symbol rate to get a higher sampling rate. During up-sampling, each QAM symbol is upsampled by a factor of L, creating a sequence $\boldsymbol{s_L}$ comprising the original samples $\boldsymbol{s}$, separated by $L-1$ zeros, and obtaining a higher sample-rate. This makes it easier to implement filters with a sharp cutoff, which are necessary to maximize the use of the available bandwidth. In the receiver side, the signal can be down-sampled to the original sampling frequency.

The sampling factor must be chosen carefully to avoid reducing SNR or getting ripples in the frequency domain. The SNR will increase with the factor L, but if the value of L is too large, the extended symbol sequence will be too long to transmit thus less efficient.

After up-sampling, the discontinuities of the expanded symbol sequence $\boldsymbol{s_L}$ is smooth out with a low-pass filter. In order to have the minimum ISI, the response of the transmit filter and the receive filter has to satisfy Nyquist ISI criterion. Raised-cosine filter is the most popular filter response satisfying this criterion [37]. Half of this filtering is done on the transmit side and half is done on the receive side, and the response of each half is a RRC filter. The RRC is the square root of the raised-cosine, and Compared to raised-cosine filter, the impulse response in RRC is not zero at the intervals of $\pm T_s$ where T_s is sampling time.

The roll-off factor β specifies the bandwidth of the filter with value range $0-1$. It controls two features: One is the rate at which the function's lobes decrease, another is the filter's bandwidth. With the roll-off factor set to 0, the side lobes decrease slowly and the bandwidth of the filter is equal to the signal's bandwidth B. When the roll-off factor is 1, the ripples decrease fast and the bandwidth becomes $2B$.

The RRC filter is characterised by two values: the roll-off factor β, and the sampling rate, T_s. The impulse response of the RRC filter $h(k)$ can be represented as:

$$h(k) = \begin{cases} \frac{1}{T_s}(1+\beta(\frac{4}{\pi}-1)), & k=0 \\ \frac{\beta}{T_s\sqrt{2}}\left[(1+\frac{2}{\pi}\sin(\frac{\pi}{4\beta})+(1-\frac{2}{\pi})\cos(\frac{\pi}{4\beta}))\right], & k=\pm\frac{T_s}{4\beta} \\ \frac{1}{T_s}\frac{\sin\left[\pi\frac{t}{T_s}(1-\beta)\right]+4\beta\frac{t}{T_s}\cos\left[\pi\frac{t}{T_s}(1+\beta)\right]}{\pi\frac{t}{T_s}\left[1-(4\beta\frac{t}{T_s})^2\right]}, & \text{otherwise} \end{cases} \tag{2.4}$$

.

The upsampled symbol $s_L(t)$ is convoluted with the RRC filter $h(k)$ with roll-off factor β and M (odd) taps, which is mathematically done by:

$$s_{trans}(k) = s_L(k) \circledast x(k) \tag{2.5}$$

where $\circledast$ denotes the convolution, and $s_{trans}(k)$ is the signal ready for transmission

that has a length of $NL + M - 1$ complex-valued samples, where N is the number of samples before pulse-shaping.

2.2 Channel Model and Impairments

Communication channel is the medium by which the signal travels through. In this thesis, the implemented stochastic channel model accounts for several important distortions and impairments, which includes IQ imbalance, PN, non-linearities and AWGN as follows.

2.2.1 IQ imbalance impairment

In real transmission, IQ imbalance is a hardware impairment caused by non-idealities in the oscillators, mixers and passive components, which cause the I and Q components to be non-orthogonal. As a result, the real and imaginary components of the complex signals interfere with each other which increases the BER.

In this thesis, the IQ imbalance is modeled on both the I and Q components of the modulated signal $s_{trans}(k)$ before transmission through the AWGN channel, as follows:

$$s_{iq}(k) = s_{trans}(k) + \alpha s^*_{trans}(k) \tag{2.6}$$

where α denotes the imbalance level and $s_{iq}(k)$ is the signal after adding the IQ imbalance.

2.2.2 Phase noise impairment

PN is a hardware effects from oscillators, which is one of the major hardware impairments affecting the performance of communication systems. Random rotation of the signal constellation is the main effects of phase noise in this thesis, and it may cause decision errors by moving the received signals to wrong decision regions.

The implementation of PN includes two parts, PN $n_{pn}(k)$ and the phase offset n_{po}. The value of the phase offset is a random number inside the uniform distribution of $[0, 2\pi)$. And the equation of the signal after adding PN is showing below:

$$s_{pn}(k) = s_{iq}(k)e^{-jn_{pn}(k)} \tag{2.7}$$

where $s_{pn}(k)$ denotes the signal after adding PN $n_{pn}(k)$, which is a random walk:

$$n_{pn}(k) = \begin{cases} n_{pn}(k)' + n_{po}, \ k = 1 \\ \sum_{k=1}^{k} n_{pn}(k)', \ others \end{cases} \tag{2.8}$$

where $n_{pn}(k)'$ is a set of k numbers of data with each value is randomly chosen from the normal distribution of $[0, pn^2)$, where pn^2 denotes the PN variance.

2.2.3 Non-linearities impairment

In practice, power amplifiers have non-linearity characteristics that make the output voltage a function of high order terms of the input voltage, resulting in a nonlinear input-output characteristic [38].

The non-linearities in this thesis are modeled as follows:

$$x_{pa}(k) = g(|x_{pn}(k)|)e^{j\angle x_{pn}(k)} \tag{2.9}$$

where

$$g(|x_{pn}(k)|) = \alpha_g \left|\max(x_{pn}(k))\right| \tanh\left|\frac{x_{pn}(k)}{\alpha_g \left|\max(x_{pn}(k))\right|}\right| \tag{2.10}$$

$x_{pa}(k)$ is the signal after adding nonlinear impairment, α_g denotes the saturation point percent of the input signal.

2.2.4 AWGN

AWGN is a basic noise model to mimic the effect of many random processes that occur in nature, which is chosen to give thermal noise in this thesis. In this thesis, the channel output $\boldsymbol{y_{ch}}$ is:

$$y_{ch}(k) = x(k) + n(k) \tag{2.11}$$

where $n(k)$ is independent, identically distributed, zero-mean and normal distributed with a variance $\boldsymbol{N}$, which represents the noise. In this thesis, the variance of the noise is set comparative to the energy level of the input signal.

2.3 Standard Receiver

The receiver aims to reconstruct the received signal which has been impaired by the channel. This thesis employs two types of receivers to recover the transmitted bit sequence, which are standard receiver and NN-based receiver. The standard receiver is used to compare and measure the performance of the NN-based receiver.

Figure 2.3 presents the block diagram of the standard receiver, which consists of two main blocks: signal to symbol converter and de-mapping.

2.3.1 Matched filtering and down-sampling

The received signal $y_{ch}(k)$ passes through a matched filter, which is obtained by correlating a template, with an unknown signal to detect the presence of the template in the unknown signal [39]. The matched filter is an optimal linear filter for maximizing the SNR in the presence of additive stochastic noise [39]. In this thesis, the template is the RRC filter in transmitter. Matched filtering is done by convolving a time-reversed conjugate format $x^*(-k)$ of the RRC filter, as illustrated in section 2.1.2. This is done mathematically by:

$$y_{rrc}(k) = y_{ch}(k) \circledast x^*(-k) \tag{2.12}$$

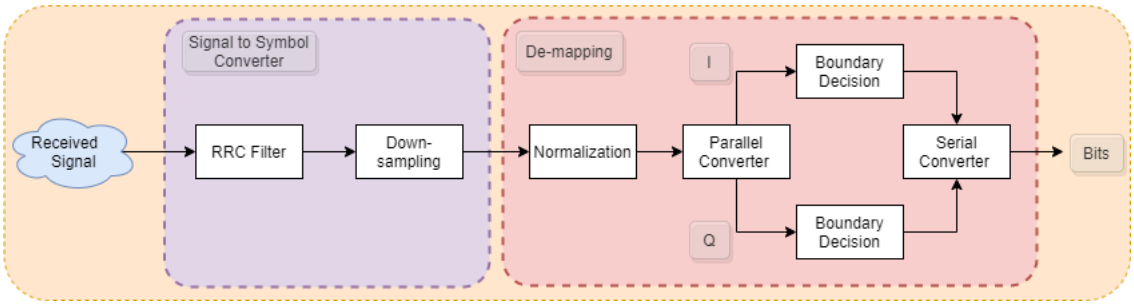


Figure 2.3: Explaining the structure of standard receiver consisting of signal to symbol converter and de-mapping. First the signal is convolved with the same filter as the transmitter block. Then it is down-sampled to retrieve the symbols back. After that, the received symbol is normalized, de-mapped and combined to get the bit sequence back.

where $y_{rrc}(k)$ is the signal after matched filter, and has a length of $N \times L$ complex-valued samples. N is the number of samples before pulse-shaping and L is the sampling factor. After that, the signal is down-sampled by a factor of L to retrieve N symbols back.

2.3.2 Demapping

After the received signal is converted to the symbol by matched filtering and down sampling, it needs to be normalized to adjust the values of the signal on their mean values of the energy scale. The normalization is done by transforming the signal so that its mean is 0 and standard deviation is 1. Signals in such form eases the process of comparisons as well as serving different needs in analysis, so the distinct is clearer and more apparent. The normalization process is the same as in transmitter.

After that, each normalized symbol $y(k)$ is mapped to one of the point $\hat{a}(k)$ on the constellation diagram shown in 2.2, denoted by A. This is done by maximum likelihood decision rule which chooses the minimum Euclidean distance between $y(k)$ and $\hat{a}(k)$. The signal $\hat{a}(k) \in A$ is chosen such that [40]

$$|y(k) - \hat{a}(k)| = |y(k) - \hat{a}(k)| = \min \{|y(k) - a(k)| \text{ for all } a(k) \in A\} \tag{2.13}$$

where $|y(k) - \hat{a}|$ denotes the Euclidean distance between the two points in the signal constellation, and $a(k)$ is the original transmitted symbol.

Next, each of the symbol points $\hat{a}(k)$ in the constellation diagram corresponds to a certain bit stream for both I and Q components, are combined in parallel converter to get the desired bit sequence back, which includes the transmitted information.

2.4 Neural Network-based Receiver

Figure 2.4 presents the block diagram of the receiver based on a NN. The receiver consists of a NN model and a demapping block.

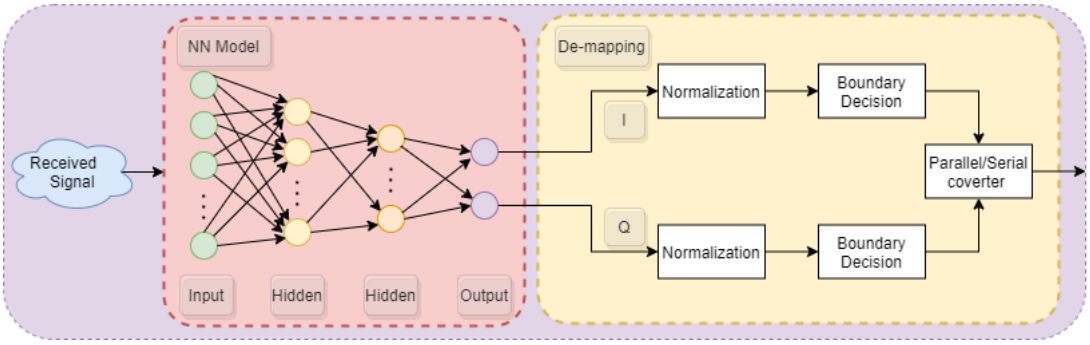


Figure 2.4: Presenting the block diagram of a NN-based receiver, consisting of NN model and de-mapping blocks. The received signal inputs to the NN model which is already trained to take the signals and outputs the corresponding I and Q components of symbol. Then each components is normalized and combined to get the bit sequence back.

During the training procedure, the parameters of the NN are optimized with the training data set which has been processed with the same transmitter and stochastic channel as the received signal. In the testing phase, after passing through the stochastic channel model, the received signals are sent into the NN. The NN outputs I and Q components of the data symbols. The NN replaces of the matched filter of the standard receiver. After passing through the same de-mapping process as illustrated before, the bit sequence is obtained.

In order to better explain the training procedure, some basic knowledge of the NN are introduced.

2.4.1 Basic structure and components of NN

The NN model block in Figure 2.4 has two hidden layers. The green nodes are neurons in the input layer, the yellow nodes represent neurons in the hidden layer while the purple nodes represent neurons in the output layer. The black arrows between layers represent the connections between neurons.

As Figure 2.5 shows, in NN layer block, each neuron calculates the output y_n from the output y_{n-1} of the previous layer by:

$$y = f(\boldsymbol{w}^T \boldsymbol{x} + b), \tag{2.14}$$

where $\boldsymbol{w}$ denotes the weights in the MatMul block in Figure 2.5, $\boldsymbol{x}$ is the input with

$$\boldsymbol{x} \stackrel{\text{def}}{=} \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \tag{2.15}$$

where $b \in \boldsymbol{R}$ is a bias added and $f(\cdot)$ is an activation function. The activation function is needed to introduce nonlinear properties or linear relationship to NN. After passing through $f(\cdot)$, y is the final output given to the regression loss block or the input to another layer if there is more than one layer.

2.4.1.1 Activation function

Activation functions can be divided into two types: linear and non-linear. Picking the simplest linear function, i.e., ($f(x) = x$ results in the network being linear and hence in a linear regression. A neural network with linear activation function will act as a linear regression with limited learning power [21]. Non-linear activation function is introduced to mimic non-linear effects that linear activation functions fails to compensate for. Table 2.1 lists some widely used activation functions.

Table 2.1: List of some widely used activation functions

Name	$f(x)$	Range
Identity	x	$(-\infty, \infty)$
Sigmoid	$\frac{1}{1+e^{-x}}$	$(0, 1)$
TanH	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(-1, 1)$
ReLU	$max(x, 0)$	$[0, \infty)$
Softmax	$\frac{e^{x}i}{\sum_{i} e^{x}i}$	$(0, 1]$

Activation functions play a key role in neural networks [21], so it is essential to understand the advantages and disadvantages to achieve better performance.

2.4.1.2 Cost function

After passing through the NN layer, the outputs together with the labels are sent into the regression loss block to calculate the loss, as Figure 2.4 illustrates. A loss function is needed to faithfully distill all aspects of the model down into a single number [21], in such a way that improvements in that number are a sign of a better model.

The loss is a function of the label (true output) and the output estimated and it is calculated through the loss function. $\hat{\boldsymbol{y}}$ is the NN's output and $\boldsymbol{y}$ is the label. The cost function is a non-negative real-valued function expressing the loss for the whole model, usually expressed by

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{N} \sum_{i=1}^{N} L(\hat{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{b}), \tag{2.16}$$

where $\boldsymbol{w}$ is the set of all weights of all layers and $\boldsymbol{b}$ is the set of all biases, $\boldsymbol{J}$ is the average loss of all the training examples. The loss is minimized by the model during the training process.

Mean squared error (MSE) and cross entropy are two commonly used loss functions in NN, where MSE is used to train the regression model and cross entropy is used to train the classification model. The MSE equation is shown below:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (\hat{\boldsymbol{y}} - \boldsymbol{y})^2 \tag{2.17}$$

In certain cases, a loss calculation formula that is not provided by the library may need to be used. In that case, people may consider defining and using their own loss function. This kind of user-defined loss function is called custom loss function.

A custom loss function is sometimes necessary to reach training objectives that cannot be satisfied with cost functions available in the ML libraries and can be very useful for solving specific problems more efficiently.

2.4.1.3 Optimizer

After obtaining the loss, the gradient of the loss is calculated given inputs data, outputs of Matmul, outputs of add bias, outputs of the NN layer and the calculated loss. This gradient is then fed back to the optimization method to update the weights to minimize the loss function.

In ML, the NN is trained using an optimization algorithm that requires a loss function to calculate the model error. In order to make the model output approach or reach the optimal value, various optimization strategies are needed to update and calculate network parameters ($\boldsymbol{w}$ and $\boldsymbol{b}$) that affect model training and model output [21]. All optimizers are based on gradient descent.

Adam optimizer [41] is a common-used algorithm proposed recently. This algorithm calculates the adaptive learning rate of each parameter. In addition to storing the average value of the exponential decay of the square of the past gradient like RMSprop, it also maintains the average value of the exponential decay of the past gradient like Momentum. Practice shows that Adam is better than other adaptive learning methods [41], for Adam adds bias-correction and momentum on the basis of RMSprop. As the gradient becomes sparse, Adam will have a better effect than RMSprop.

2.4.1.4 Backpropagation

The back propagation (BP) algorithm of the NN is applied to calculate the weight updates together with the optimizer.

The BP algorithm is a common method used in conjunction with optimization methods to train NN [42], which is given more detailed introduction in **??**. Figure 2.5 illustrates the training procedure of a NN model with one hidden layer in TensorFlow. This method consists of two main sections: forward and backward propagation.

Forward propagation is done by passing the data to the NN layer and the output is, together with the label, given to the regression loss block, to calculate the loss.In backward propagation, the gradients of the weights is calculated given a loss function. These gradient values are passed on to the optimizer block which updates the weights and tries to minimizes the loss. This process is done only for one batch. For the next batch, the updated values of weight and bias from the backward propagation is given to the NN layer in the forward propagation.

2.4.2 Modifications on the system model

Several modifications are introduced to the system model in this thesis.

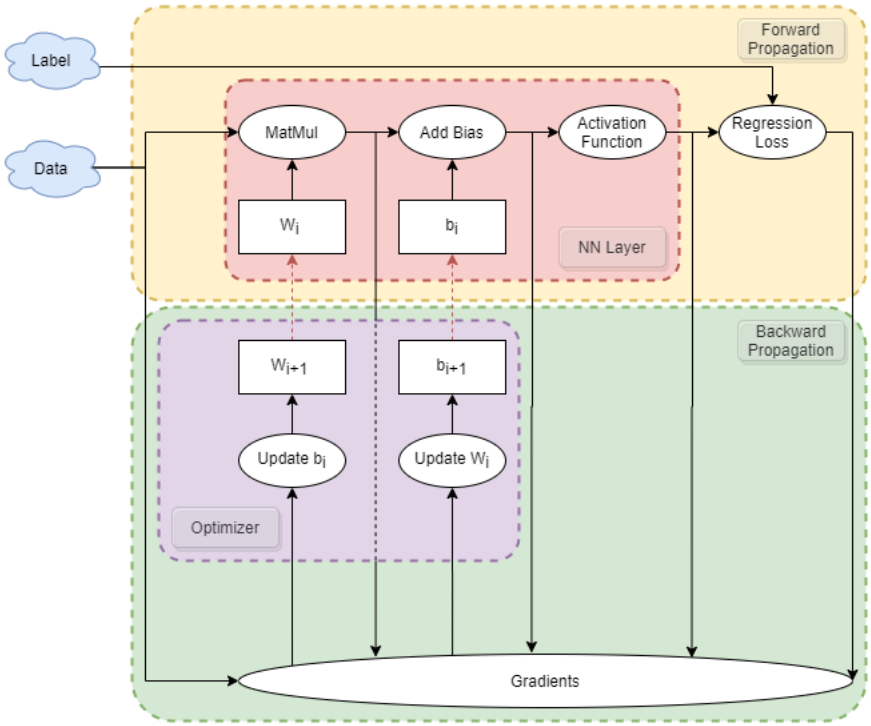


Figure 2.5: Illustrating training of a NN model with one hidden layer done in TensorFlow. Gradient of the loss function for the weight in the network is calculated. This gradient is then fed back to the optimization method to update the weights to minimize the loss function.

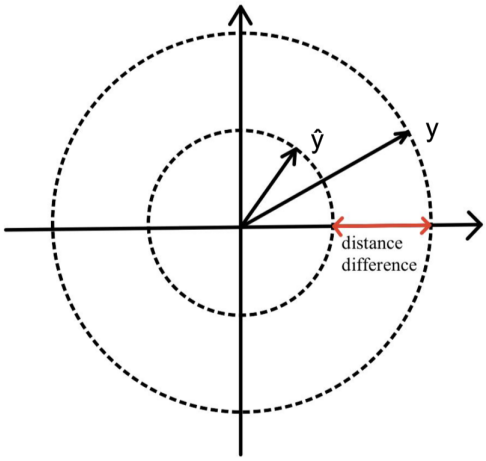


Figure 2.6: Illustrating the CRE function, denotes the difference between the distances of two points to the $(0, 0)$ point.

2.4.2.1 Custom loss function: constant radius estimation

In this thesis, the custom loss function is a constant radius estimation (CRE) function, as shown in the Figure 2.6. The loss function $L(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is defined as the difference between the distances of two points to the $(0, 0)$ point. The equation is showing below:

$$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \left| \sqrt{\sum_{i=1}^{N} \hat{y}_i^2} - \sqrt{\sum_{i=1}^{N} y_i^2} \right|^2 \tag{2.18}$$

where N is the number of samples.

As can be seen in the equation and the Figure 2.6, the function only considers the amplitude difference of the two comparing points but independent of the phase difference. This cost function is used when the channel model is with PN impairment and must be used with the blind phase search (BPS) algorithm [43] to correct the phase.

2.4.2.2 Blind phase search

A schematic structure of BPS [44] is shown in Figure 2.7. The algorithm is applied on the down-sampled signal where one sample represents one symbol. Each input symbol x_k is rotated with B number of test phases ϕ_b, spread over an angle of $\frac{\pi}{2}$ calculated as [44]:

$$\phi_b = \frac{b}{B} \cdot \frac{\pi}{2}, b \in \{0, 1, ...B - 1\} \tag{2.19}$$

Figure 2.8 illustrates the symbol rotation in the complex plane for 16 QAM and 8 test phases. The rotating track of the input sample is a part of a complete circle.

After that, the distance between each rotated symbol and its closest constellation point is calculated as

$$|d_{k,b}|^2 = \left| x_k e^{-j\phi_b} - \hat{X}_{k,b} \right|^2, \tag{2.20}$$

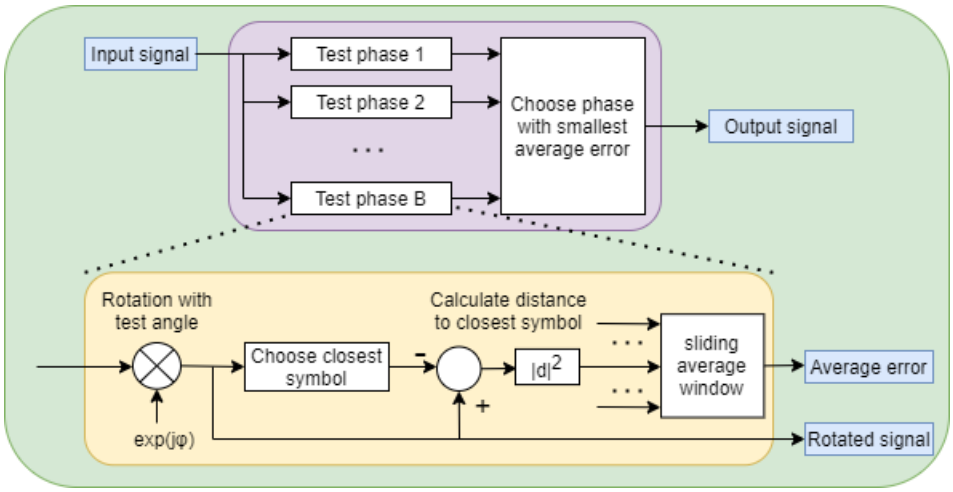


Figure 2.7: Illustrating the BPS algorithm. Each input symbol is rotated with B number of test phases, spread over an angle of $\frac{\pi}{2}$ to get the average error and the rotated signal. Next, the rotated signal with smallest average error is selected as the output signal.

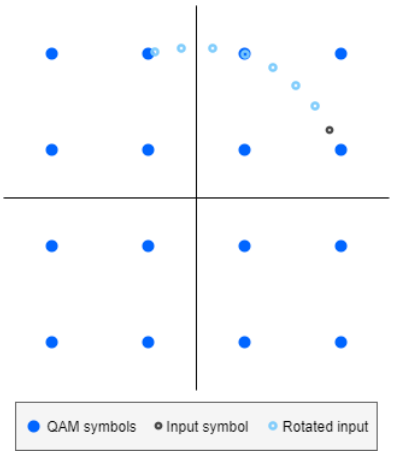


Figure 2.8: Illustrating the symbol rotation in the complex plane, for 16 QAM and 8 test phase angles.

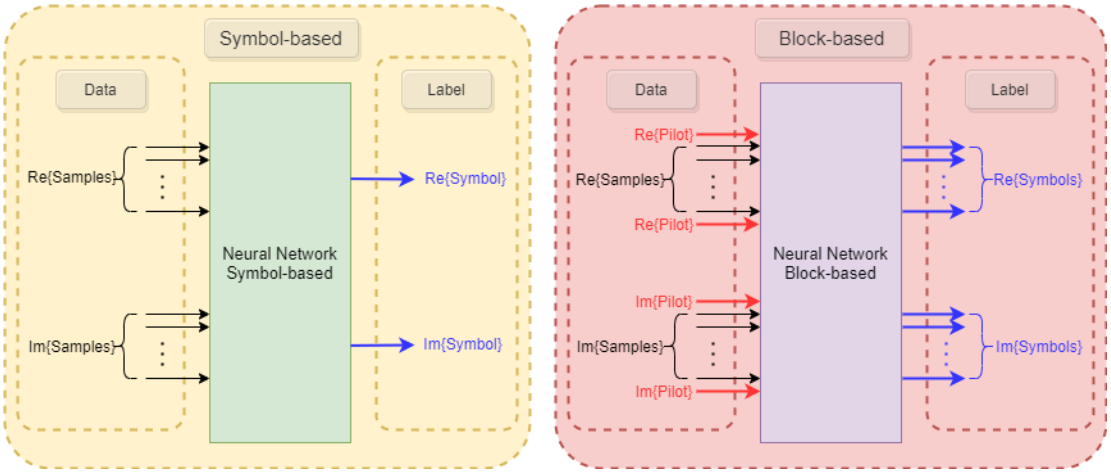


Figure 2.9: Showing the two approaches of selecting the NN examples. In the left, the symbol-based, the label is one symbol and the data is the corresponding received samples to that symbol. In the right, the block-based, the label is the block of symbols and the data consists of pilots and the received samples corresponding to that symbol-block.

where $\hat{X}_{k,b}$ is the closest symbol. The sliding average window in Figure 2.7 is used to reduce the impact of the noise on the result [44]:

$$e_{k,b} = \sum_{n=-N}^{N} \frac{|d_{k-n,b}|^2}{2N} \tag{2.21}$$

The sliding average window is applied to the calculated distances giving an average error for each test phase. N denotes the length of sliding average and it is chosen depending on the symbol rate [44]. The correct rotated input sample to be output signal can be selected by finding the minimum $e_{k,b}$.

2.4.2.3 Neural network training examples: data and labels

This thesis uses two different approaches when selecting training examples; symbol-based approach and block-based approach as is visualised in Figure 2.9. In order to train the NN, a number of training examples are given to NN to optimize the model. Each training example contains the data and labels. Data is the input given to machine learning and label is the true output corresponding to that data. In the symbol based approach, data corresponds to samples of the received signal and labels corresponds to the real and imaginary parts of a symbol. Similarly but slightly different, for the block-based approach data corresponds to a larger set of samples of the received signal together with pilot symbols, while the labels are a set of real and imaginary values of symbols.

Besides, since the NN can only process real values, that is why the real and the imaginary part of both samples from the received signal and symbols are separated nodes as shown in the Figure 2.9. It is clear that all nodes from the real part and imaginary parts are connected inside the NN model.

Looking closer, the pilot symbols are used as a part of training data in the block-based approach. These pilot symbols are known to both transmitter and receiver. In

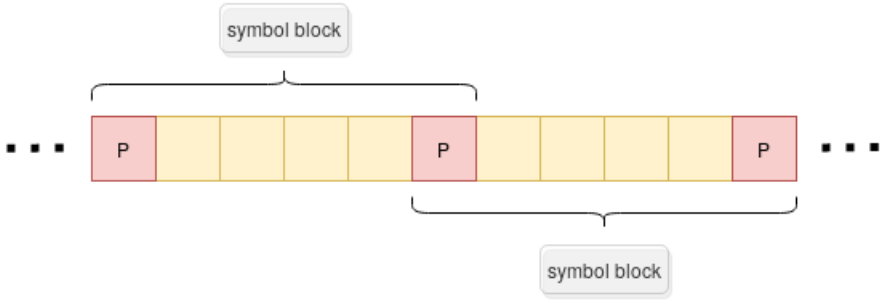


Figure 2.10: Showing the structures of the symbol block. Each block starts with a pilot-symbol followed up by few information symbols and ended by another pilot-symbol. Red squares present the pilot symbols and yellow squares present the information symbols.

a conventional receiver, they are used for various purposes such as synchronization and equalization while in this thesis they are used to compensate for PN, that is why they are serving as input data in the block-based approach. Pilot symbols are located in different parts of transmitted signal depending on the application. In this thesis, pilot symbols are with a regular spacing between information symbols as shown in Figure 2.10.

2.4.2.4 Series versus parallel structure of model

Although the non-linearities and IQ imbalance are introduced to the transmitted signal, the relationship between the input and output data is still mostly linear for the added non-linearities are quite weak [45]. In principle, the NN can learn this relationship, but this turns out to be difficult in practice [45]. As such, a parallel structure of the NN is introduced in this thesis that the input samples are also appended to the output samples from the final hidden layer before given to the output, as can be seen in Figure 2.11. In this way, the NN can focus on the nonlinear portion of the signal [45]. Figure 2.11 shows two different structures of the NN model.

2.5 Performance Metric

In analyzing the performance of a system, the quality of the output of the receiver in the presence of noise in the channel and transmitter is measured via a parameter known as SNR. Moreover, for a given transmitted power and noise in the system, such a quality depends on the type of modem used. A modem with higher resistance to noise allows more noise to be tolerated or less power to be transmitted, but still achieve the desired performance.

The average probability of bit-error is a measure of how well the received signal is de-mapped, which is the frequency of occurrence of errors in the decoded sequence. This measurement metric is BER. In this thesis, BER is expressed in terms of the SNR.

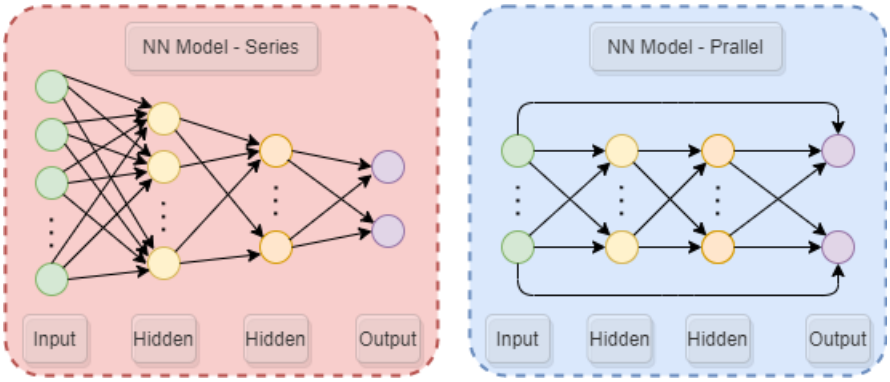


Figure 2.11: General structure of the NN. Left block shows the serial structure and right block shows the parallel structure. There are two output neurons for the real and imaginary part of the signal and 2 hidden layers. In the parallel structure, the input samples are also appended to the output samples from the final hidden layer before given to the output.

The error probability of M-QAM modulation over AWGN channel can approximately be calculated as:

$$P_s = \frac{2\sqrt{M}-1}{\sqrt{M}} Q\left(\sqrt{\frac{3\bar{r}_s}{M-1}}\right) \tag{2.22}$$

where $\bar{r}_s$ is the average energy per symbol.

The BER for Gray coded QAM over AWGN channel is shown in Figure 2.12.

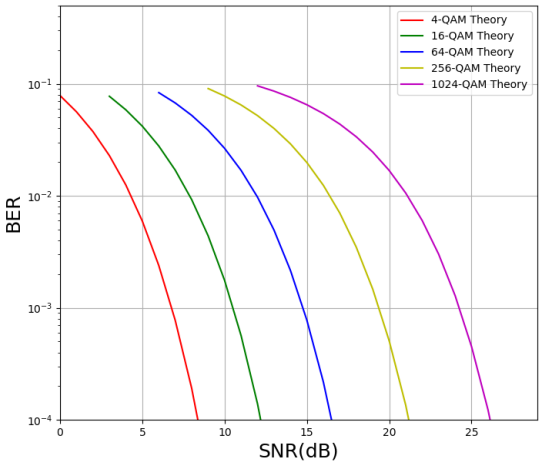


Figure 2.12: Theoretical BER for Gray coded QAM, for 4, 16, 64, 256 and 1024 bits per symbol.

3
Methodology

This chapter aims to describe the implemented communication system and the the libraries that were used for constructing the NN model. Figure 3.1 shows the block-diagram of the communication system which consists of three main parts: Transmitter, channel, and receiver. The simulation was performed using the Python programming language together with the Python libraries TensorFlow and Keras which where used to construct and train the NN model, refer to **??**. Worth noting is that the conventional and NN communication systems differ only in the receiver part, while the transmitter and channel blocks are the same.

3.1 Transmitter Implementation

The transmitter section in Figure 3.1 consists of three parts; bit generation, mapping and pulse-shaping. First, a random binary sequence is generated as a message to be transmitted. Next, this message is converted into complex symbols by the mapping block according to a gray-coded M-QAM constellation, more specifically any square QAM constellation, such as 4, 16, 64. Lastly, the complex symbols are converted to a signal in the pulse-shaping block by interpolating and then convolving the symbols with a RRC.

3.2 Simulation of Channel

The signal generated in the transmitter is passed through the simulated channel as shown in Figure 3.1. This channel-model has 4 simulated impairments which are: IQ imbalance, PN, non-linearities and AWGN. The impairments are implemented according to the models explained in Section 2.2 and the system is implemented such that the impairments can be enabled or disabled.

3.3 Receiver Implementation

The receiver has two different blocks as illustrated by the block diagram in Figure 3.1. The transmitted signal is transferred through the channel to the active receiver block; either the standard receiver, the NN-based receiver, or the NN-based receiver training.

The standard receiver first applies a matched filter, which is the same RRC filter that was used in the pulse-shaping block of the transmitter. Normally a matched

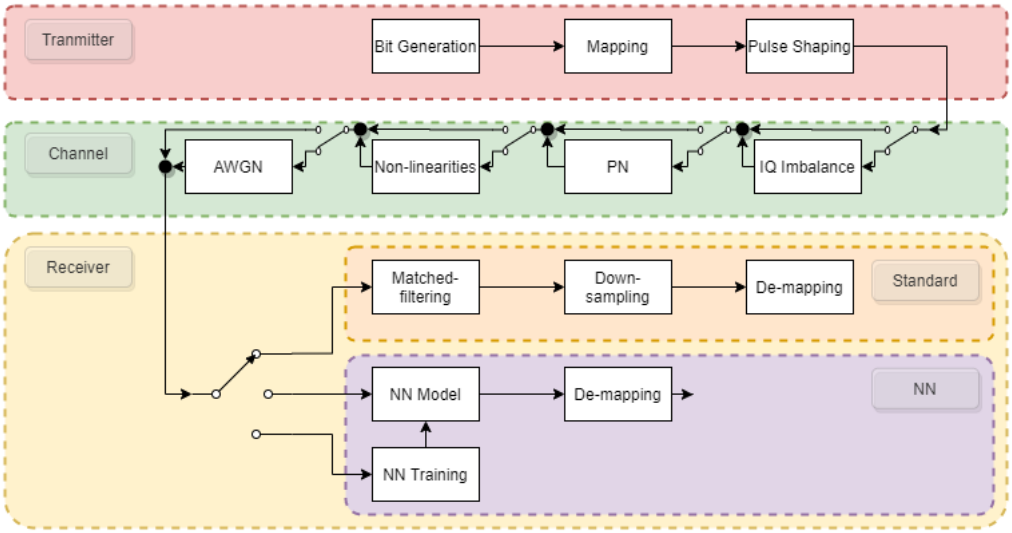


Figure 3.1: Showing the intended communication system, which includes the transmitter, channel, standard receiver and NN-based receiver. The simulated channel includes IQ imbalances, PN, non-linearities and AWGN. The NN-based receiver includes the trained NN model block which converts the received signals into symbols.

filter should be a time-reversed and complex conjugated version, but the RRC is time-symmetric and hence this is not necessary. Next, the signal is down-sampled in order to obtain the transmitted symbols. Lastly before converting the symbols to bits the signal is normalized in order to match the constellation points.

In the NN-based receiver, the matched filtering and down-sampling blocks are replaced with a NN model. The input to the NN model is samples from the received signal and in the block-based approach also the known pilot symbols. The NN output is the retrieved symbol or symbols depending on the approach. Next, the symbols are mapped into bits in the de-mapper block as shown in Figure 3.1.

Looking at Figure 3.1, the NN-based receiver also has the NN training block which is used to train the NN model. This model must be trained with sufficient number of training examples while also considering the number of epochs and batches, this is described in Chapter 4.

3.3.1 Neural network specifications

Also, it is worth mentioning that the the NN model used in this thesis was optimized using Adam optimizer which was explained in Section 2.4.1.3. Besides, two different types of cost function were used to train the NN model. The first cost function used was MSE because we want to get the amplitudes values of I and Q back, so it is a regression model. In an AWGN channel, the MSE cost function manages to optimize effectively the system in presence of AWGN channel.

However, when the PN impairment is added to the channel, then no longer the model with the MSE cost function manages to optimize the system, that is why the CRE cost function was introduced. This cost function which was explained in 2.4.2.1, is independent of phase value and corrects for only the amplitude value.

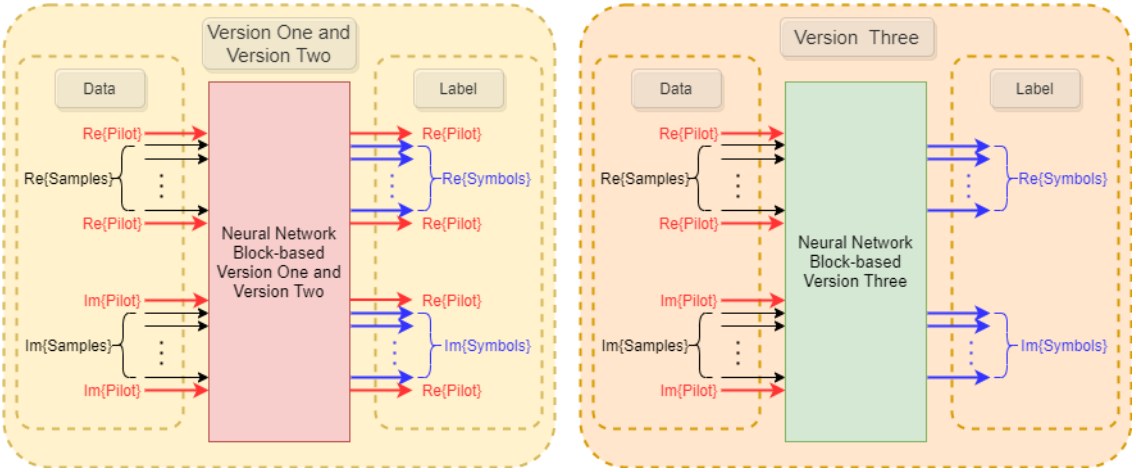


Figure 3.2: Showing the different implementations of block-based approach. The left figure depicts how the data and labels are given to train the NN in the first and second version. The right block-diagram shows the data and labels given to the NN in the third version.

To correct for phase value and estimate the symbol value, the CRE cost function is used together with the BPS algorithm, which corrects for the PN. So, using the CRE cost functions, first the model finds the best amplitude for the symbol and then the BPS algorithm is employed to estimate the phase of the symbol, ready to be given to the mapper.

3.3.2 Block-based approach

The block-based approach was implemented to compensate for the PN without the need of BPS algorithm. An important step to design a NN is that how the data, labels and costs functions are chosen to guide the NN to the wanted result. In this thesis work, three versions of the block-based approach are introduced, which are shown in Figure 3.2.

Firstly, considering the first version shown in Figure 3.2, a block of samples together with two pilot symbols are given as data to the NN. The labels correspond to the pilot symbols together with the information symbols. This version is updated to work with both MSE and CRE cost functions, one at a time. The primary aim of having the block-based approach was that, if the symbols are given as a block, even in the presence of PN, the NN still might learn how to correct for the information symbols since the pilots are known. Looking back to the block-based structure, each block start and end with a pilot, so the NN might pick up the relationship between the error of the first pilot and the last pilot, using linear optimization to correct for the information symbols between these 2 pilots.

Secondly, the block diagram of second version and how the data and labels are given to this model are the same as the first version as shown in Figure 3.2, with only difference in the cost function. In this version, the cost function applied the CRE cost function for the pilot symbols, and MSE for the information symbols. The idea behind this version comes from the results derived from the first version. In

the first version of structure, the NN model can not compensate for PN. By plotting the constellation diagram of the symbols out from the NN, it was observed that by plotting the received pilots and information symbols separately, some symbols are corrected perfectly with zero phase offset and others have a phase offset. Looking closer to this constellation diagram, it was clear that there were two set of constellation points, the pilot symbols were retrieved perfectly, while the information symbols were all shifted with a phase offset. This was the reason behind having separate cost functions for pilots which have a zero phase offset and information symbols with a fixed phase offset.

Lastly, the third version structure was distinctive to other two versions as shown in Figure 3.2. In this method, a block of samples together with two pilot symbols are given as data to the NN, similarly to version one and two. However, the labels differs from the version one and two and they correspond to only the information symbols. Since there are no pilot outputs, this version only works with MSE cost function. This idea behind this version was based on the constellation diagrams of previous version. In version two, by plotting the constellation diagrams of the symbols output from the NN, in case of having a pilot-spacing of two, there are two phase offsets. This is because the pilot symbols are given directly both as input and output to the NN and the model picks up that these two neurons are equal. So, the model fails to use the information in pilots to detect and correct for the phase offset and that is why the information symbols all are shifted with a constant angle, that is why the pilots were not given as output.

3.4 Validating System Performance

NN BER curve was compared to a reference standard BER curve. To verify the simulations, the resulting BER curves were compared to an additional reference, the theoretical BER for an AWGN channel.

A brief summery of all system tests are described as following:

- Verify if the standard and NN-based receivers perform as expected in the AWGN channel.
- Implement the customized cost function, CRE together with the BPS algorithm to compensate for PN impairment in an AWGN channel. Then, test and compare the standard receiver to the one of NN in presence of PN impairment.
- Test and compare the NN-based to the standard receiver in presence of non-linearities and IQ imbalance in an AWGN channel. Afterward, compensating for these impairment for NN receiver by adding a non-linear layer. The performance of system is tested given a variety of parameters, such as number of nodes and activation functions.
- implement the block-based approach to compensate for PN impairment without the BPS algorithm. Then, compare the resulting BER curve with the symbol-based approach BER curve, given different impairments.

4
Results and Discussion

This chapter aims to presents and briefly explains the results of this thesis. The chapter is divided into five sections, where the first section explains the parameter selections and the four following sections present and discuss the result given their respective impairments, as noted the AWGN test case is mainly used for system verification. The impairment for each section are; AWGN, PN, non-linearities together with IQ imbalance and lastly all three impairments are applied simultaneously.

In order to analyze the performance of the system, the BER curve of each test case was compared to the theoretical BER and the resulting standard receiver BER. The theoretical BER curve, is the curve given the AWGN impairment, which was explained in Section 2.5. The theoretical BER graph which can be regarded as an optimal BER curve derived from an AWGN channel as described in Section 2.5. The theoretical BER is the same for all of the test cases in order to make them comparable. It was also used to verify wherever the system was able to reach the optimal BER. Both the NN and the standard receivers, in most test cases applied the BPS algorithm when PN was present in the channel but they had no algorithm to compensate for IQ imbalance or non-linearities.

4.1 Parameter Selection

In order to make the results of the various tests comparable a set of standard values where chosen to the variables as shown in Table 4.1. The parameters are divided into four section: Training, demodulation, model and signal processing parameters. The following sections why parameters of each of these section are chosen in details.

4.1.1 Training parameters

The training parameters were selected by testing the NN model and observing its returning loss and capability of predicting the symbols from the noisy samples. In order to have a functioning NN model, it should be trained previously to use with sufficient amount of the training examples. In this thesis, the signal samples together with pilot symbols are the data while symbols are labels, therefore the number of training examples depends on the number of symbols.

The number of symbols used for training was decided to be $2.5 \cdot 10^6$ for the symbol-based model, and 10^7 for block-based model, because the loss reached to as low of $5 \cdot 10^{-4}$ after the first epoch. By increasing the training example, the loss did not improve notably, and thus this number was used due to the limited

Table 4.1: Values of constant parameters used in this thesis

Parameter type	Parameter	Standard value
Training	QAM Order	256
	SNR	24 dB
	No. of Symbols Symbol-based	$2.5 \cdot 10^6$
	No. of Symbols Block-based	10^7
Demodulation	QAM Order	16 and 64
	No. of Symbols	10^6
Model	No. of Epochs	10
	Batch Size	500
	Training/Validation Split	70/30
Signal Processing	Oversampling Factor	4
	Symbol Rate	$2.5 \cdot 10^{10}$
	Length of RRC (one-side)	20
	Roll-off Factor of RRC	0.1

computing power. Also, in order to use 16 and 64 QAMs for demodulation, the NN model is required to be trained with 64 QAM or any other higher modulations. 256 QAM was chosen, since it had a better performance compare to 64 and if a higher modulation was used, such as 1024, the results would not improve while the simulation time would increase. Besides, since the 256 QAM was chosen for training, the corresponding appropriate SNR should be selected. If the value of SNR for training is low, which means the noise level is high, the NN model cannot pick up how to give the corresponding symbols given the received samples. On the other hand, if the selected SNR value for training is high, the NN model converges fast for a low value of noise, so later on it will not be able to retrieve the symbols from received signal with a high value of noise. This value of SNR was selected by trial and error to be the best fit for 256 QAM order.

4.1.2 Model parameters

Before explaining on how these model parameters are chosen, two terms, epochs and batch size must be explained. The batch size determines how many training examples are given to the NN model simultaneously to find the loss and update the coefficients and biases. To test and train the model further, given the same data set, the NN can shuffle the training example and run it through the NN model again, each of these iterations, where the whole set of training examples have tested is called an epoch. Therefore, the whole training examples are divided into batches, where batch size of 500 is a suitable value. Given the sufficient number of batches into model, the model learned the pattern and reached to a very low loss after the first epoch, that is why we kept the number to 10. Besides, to have a reliable NN model, it needs to be tested and validated, this is why the training examples are split into two set: training and validation set. 70/30 percent ratio for training and validation sets is most commonly used and worked perfectly with our NN model.

4.1.3 Demodulation parameters

In order to compare the results given different impairments, 16 and 64 QAM BER curves were chosen. These two QAM modulations are, in comparison to 4 QAM, more affected by impairments and is therefore used to measure the obtained results. 4 QAM has only 4 decision regions and as far as the model is able predict the quadrant of the received symbol, the result is correct. The amplitude of symbol does not matter due to the geometry of the 4 QAM constellation. Besides the phase will be mapped correctly in a range of ± 45 degree, before it is detected as a different symbol. On the other hand, 16 and 64 QAM constellation points are placed closely. So, if the system is able to recover the symbols in presence of impairment for these two QAM orders, then most probably it manage to retrive the symbols for other QAMs too.

In order to get a reliable BER curve with a minimum BER of 10^{-4} at least 10^6 symbols are required, in order to have 100 error events. This to make it statistically unlikely that the BER deviates from the true average mean. For 16 and 64 QAM this corresponds to transmitting respectively $4 \cdot 10^6$ and $8 \cdot 10^6$ bits.

4.1.4 Signal processing parameters

As it was explained in Section 2.1.2, a number of suitable parameter should be selected for RRC filter. In this thesis, since an over-sampling of 4 and filter length of 20 is chosen, then a roll-off factor of 0.1 works very well with those values. Since, the smaller the number of roll-off factor, the sharper is the filter, so the higher must be the over-sampling and length of filter to avoid ISI. Also, 25 GHz is chosen as the Symbol-rate because the system is designed for a microwave link.

4.2 System Validation with Additive White Gaussian Noise Channel

Before using the designed communication system shown in Figure 3.1, it must be verified to check if it worked as intended. This verification was done by testing the system with the AWGN channel, given the simplest NN model with one linear hidden layer. If the model learns the optimal decisions boundaries then it will be the same as the theoretical BER curve.

A set of test cases were performed to find an appropriate number of nodes used for the linear layer which refers to a hidden layer with a linear cost function. That is why the NN-based receiver with linear layer was tested with 20, 50, 100, 200 and 322 neurons, where 322 is the number of samples in NN model. Figure 4.1 shows the simulated BER curve with different number of neurons in the linear hidden layer.

Figure 4.1 shows that BER curve for NN-based receiver given 20, 50, 100, 200 and 322 neurons are the same as the theory and standard BER curves. Meaning that the system is working properly and giving the reasonable result as expected. The number of linear nodes used in the following testing is set to be 20 since it takes lesser computational time compare to others.

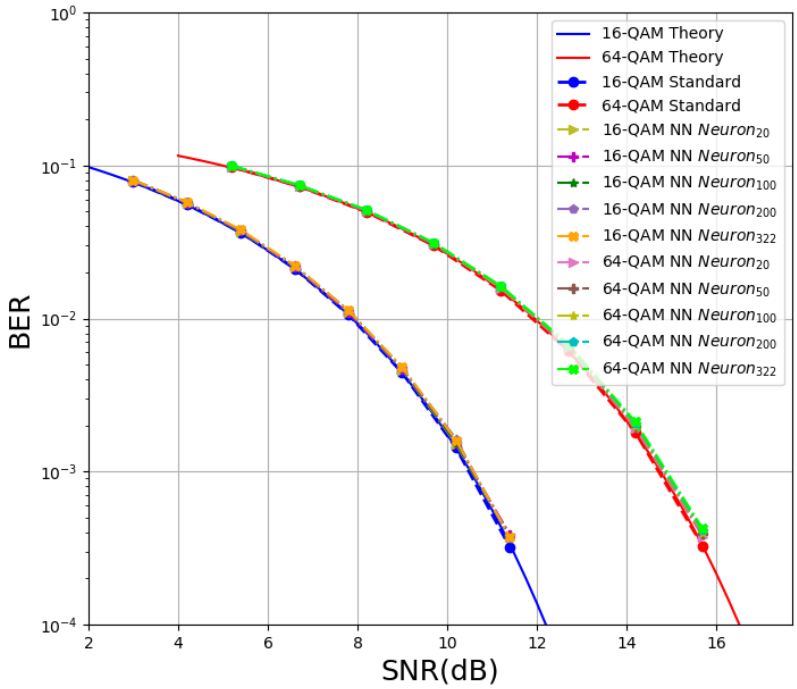


Figure 4.1: BER for theory, standard receiver, and NN receiver. The NN-based receiver was tested with 20, 50, 100, 20, and 322 neurons in the linear hidden layer.

4.3 Performance of the System in Presence of Phase Noise Impairment

In the following testings, the PN impairment consisted of two parts, the phase-offset and a PN. The PN was modeled as a zero-mean Gaussian random walk while the constant phase-offset in the interval 0 to 2π. An important note is that the same PN variance is used for both training the model and demodulating the signal.

4.3.1 Performance of symbol-based approach in presence of PN impairment

In order to compensate for Phase-noise, a number of setups where implemented and this section aims to cover those results. The verification tests carried out are summarized in 4.2 and Table 4.3.

The first verification test was intended to verify which cost function that could best compensate for PN. Furthermore, it also aimed to verify if the BPS was required to handle the PN. The various test cases used to verify this are summarized in Table 4.2, where there is both phase offset and PN of 10^{-7}.

Test cases C_{MSE} and C_{CRE} in Figure 4.2 clearly indicate that without the BPS algorithm, none of the NN models with MSE and CRE cost functions are capable of correcting for the current phase error. Hence, for our specific system the BPS is required for the system to correctly compensate for PN. Test case $C_{MSE}P_{BPS}$ displays that the NN model trained with the MSE cost function does not manage to recover the symbols effectively even when using the BPS algorithm together with closely placed pilots. This is more evident for higher modulation by comparing the BER curves of test case $C_{MSE}P_{BPS}$ corresponding to 16 and 64 QAMs. BER curve for test case $C_{CRE}P_{BPS}$ falls right on top of the theoretical BER curve, which demonstrate the model trained with CRE cost function performs satisfying for both QAM orders.

Also, it must be mentioned that the model trained with the CRE cost function requires to have the BPS. This is because as it was mentioned in Section 2.4.1.2, this cost function ignores the phase value and minimizes the cost by approximating the amplitude to the nearest radius to get the symbols back.

Therefore, it is concluded that the model trained with the MSE cost function cannot recover sufficiently the symbols transmitted through the AWGN channel in presence of PN impairment.

Next set of tests is performed to check the performance of the model trained with the CRE cost function, where the PN variance is increased and the pilot-spacing is reduced. To do so, the NN model is tested with only CRE cost function and in presence of phase offset and a variety of other parameters which are outlined in Table 4.3.

As it can be seen from Figure 4.3 that the NN model with CRE cost function can retrieve the symbols effectively even when both, the pilot-spacing and PN are increased. Therefore, in presence of PN noise in an AWGN, this model in combination with the BPS manages to recover the transmitted data.

Table 4.2: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for symbol-based method. All cases have both phase offset and PN of 10^{-7}.

Test Case	BPS	Pilot Spacing	Cost Function
C_{MSE}	Off	5	MSE
C_{CRE}	Off	5	CRE
$C_{MSE}P_{BPS}$	On	5	MSE
$C_{CRE}P_{BPS}$	On	5	CRE
Standard	On	5	None

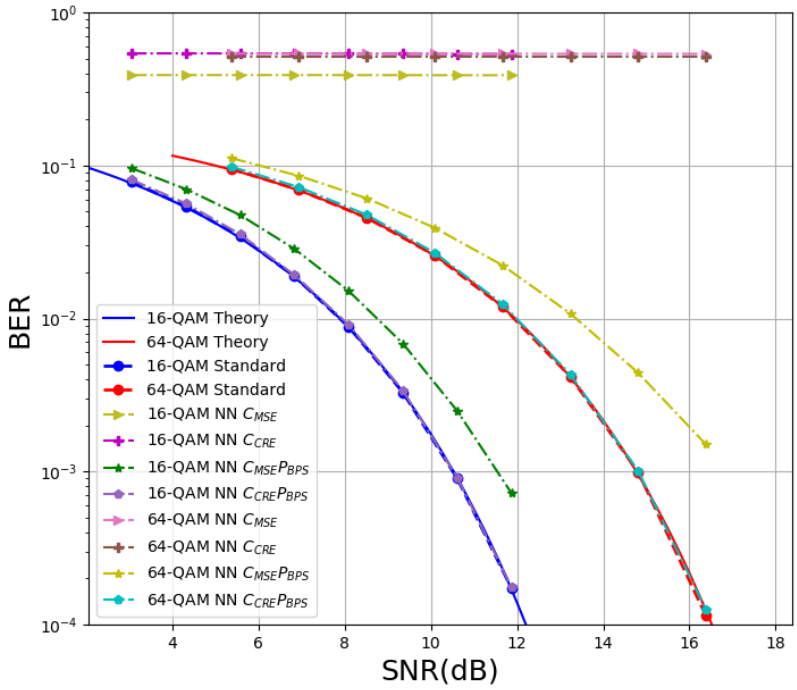


Figure 4.2: BER for theory, standard receiver, and NN receiver for symbol-based approach. The NN was tested with and without the BPS for cost functions MSE and CRE. All cases are performed in presence of both phase offset and PN of 10^{-7}.

Table 4.3: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for symbol-based method with CRE cost function.

Test Case	PN	Pilot Spacing
PN_7PS_5	10^{-7}	5
PN_5PS_5	10^{-5}	5
PN_7PS_{10}	10^{-7}	10
PN_5PS_{10}	10^{-5}	10
Standard	10^{-5}	10

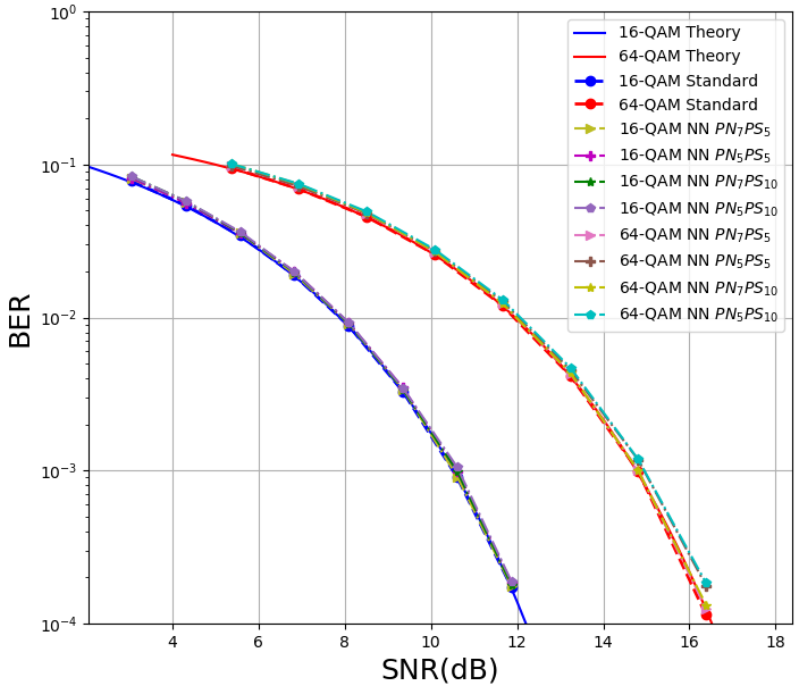


Figure 4.3: BER for theory, standard receiver, and NN receiver for symbol-based approach. The NN was tested given CRE cost function and BPS algorithm with different PN and Pilot spacing values. All cases are performed in presence of phase offset.

4.3.2 Performance of block-based approach in presence of PN impairment

The block-based approach which was explained in the Sections 2.4.2.3 and 3.3.2 is designed such that each block starts with a pilot and ends with a pilot. The distance between the two consecutive pilots is defined to be the pilot-spacing and is equal to the length of each block.

The concept behind applying the block-based was that if a block of symbols are given to the model instead of giving a single symbol, then the model can pick up and detect the PN and correct for it automatically. This is how version one was formed and was tested and later on based on its performance was adjusted to the version two and modified further after few more testings to the version three as it is already explained in the Section 3.3.2

4.3.2.1 Block-based approach first version

This section, investigate the block-based approach version one which the implementation was explained in Section 3.3.2. A set of test cases was carried out that are summarized in Table 4.4 and 4.5.

Table 4.4 compares the performance of block-based approach for two different cost functions. In order to validate the system performance an initial test with no impairments was performed, that is why the phase offset and PN are zero and the pilot spacing is 5 in Table 4.4. The BER curve for the test cases $C_{CRE}P_{BPS}$ and C_{CRE} in Figure 4.4 shows that the block-based approach does not function with the CRE cost function, given whether if the BPS is used or not. That is why the next test cases given in Table 4.5 are focused only around testing the system using MSE cost function.

The purpose of next attempt is to determine whether the version one of block-based approach with the MSE cost function can compensate for the PN. To do so, a number of different test cases given all with MSE cost function is performed which is outlined in Table 4.5.

The BER curves for test cases $P_{BPS}P_{off}PS_5$ and $P_{off}PS_5$ in Figure 4.5 show that version one model cannot compensate for the phase-offset parameter whether if the BPS algorithm is employed or not, which is why for the rest of test cases in the Table 4.5, no phase-offset is added to the system. Test cases $P_{BPS}PN_7PS_5$ and PN_7PS_5 display that PN can be corrected to some extend using this model with block length of 5 for 16 QAM modulation. Comparing the BER graphs of test cases $P_{BPS}PN_7PS_5$ and PN_7PS_2, shows that the BER curve improves if the BPS algorithm is applied.

Therefore, from the Figure 4.5 is concluded that the version one cannot recover the data transmitted effectively. By reducing the block length and pilot-spacing in test cases $P_{BPS}PN_7PS_2$ and PN_7PS_2, the BER graph is worsen. Compared to NN, the standard BER graph is closer to the theoretical BER for all the test cases, even though there exist no BPS. As a result, version one model does not perform even as good as a standard model. That is why to improve the block-based approach, the cost function was altered, resulting in version two block-based approach.

Table 4.4: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for the first version of block-based method. The NN was tested with and without BPS algorithm for cost functions MSE and CRE. In all test cases, the phase offset and PN are 0.

Test Case	BPS	Cost Function
$C_{MSE}P_{BPS}$	On	MSE
$C_{CRE}P_{BPS}$	On	CRE
C_{MSE}	Off	MSE
C_{CRE}	Off	CRE
Standard	Off	None

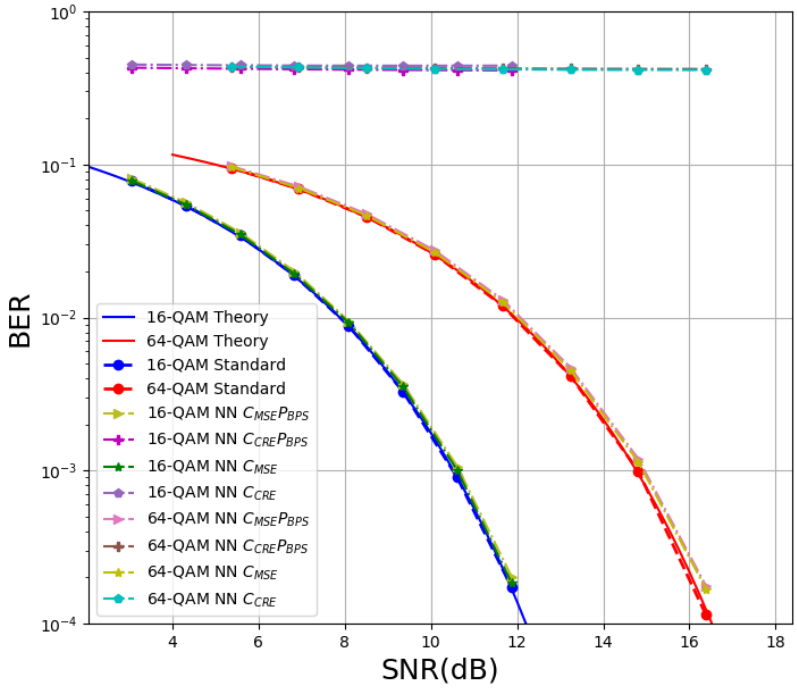


Figure 4.4: BER for theory, standard receiver, and NN receiver for first version of block-based approach. The NN was tested with and without BPS algorithm for cost functions MSE and CRE. In all test cases, the phase offset and PN are 0.

Table 4.5: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for first version of block-based method. The NN was tested only for MSE cost function, with and without BPS algorithm, PN, phase offset, and pilot spacing 2 and 5.

Test Case	BPS	Phase Offset	PN	Pilot Spacing
$P_{BPS}PN_7PS_5$	On	Off	10^{-7}	5
$P_{BPS}P_{off}PS_5$	On	On	0	5
PN_7PS_5	Off	Off	10^{-7}	5
$P_{off}PS_5$	Off	On	0	5
$P_{BPS}PN_7PS_2$	On	Off	10^{-7}	2
PN_7PS_2	Off	Off	10^{-7}	2
Standard	Off	Off	10^{-7}	2

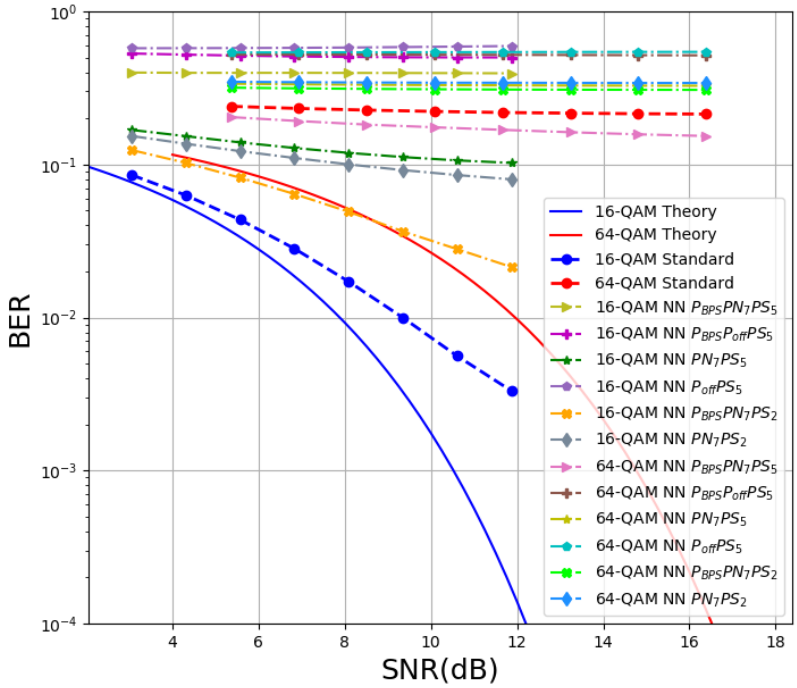


Figure 4.5: BER for theory, standard receiver, and NN receiver for the first version of block-based approach. The NN was tested only for MSE cost function, with and without BPS algorithm, PN, phase offset, and pilot spacing 2 and 5.

4.3.2.2 Block-based approach second version

As it was explained in Section 3.3.2, block-based approach version two is an extension of version one where its cost function is a combination of MSE cost function for data and CRE cost function for pilots and which is the same for all the test cases in Table 4.6. To evaluate the performance of this version and its response to the PN, a combination of test cases were done which are summarized in table 4.6.

In order to evaluate the performance of the system designed, the test cases $V_2P_{BPS}PN_0$ and V_2PN_0 were performed with no PN impairments. results from test cases $V_2P_{BPS}PN_0$ and V_2PN_0 in Figure4.6 show that version two model performs as expected if there is no PN and phase-offset. Next, the response of the version two model to phase-offset was examined given test cases $V_2PN_0P_{off}$ and $V_2PN_7P_{off}$. The results show clearly that this system cannot compensate at all for the phase-offset. test cases. However, the result of other tests shows that if there is some phase-noise or offset, the model cannot correct for it. It seems that the NN model cannot learn in presence of PN. Besides, The standard receiver outperforms all the NN models without any kind of BPS. Therefore, we changed this model slightly to V3 and reinspect it.

4.3.2.3 Block-based approach third version

In the V3, since the output symbols are only the information symbols and the pilots are not output, the BPS cannot be used, therefore only the MSE cost function is applied. to assess the performance of this version, a series of tests were completed, which are recapped in Table 4.7. In all these testings the BPS algorithm was not used, because if the model's output does not include the pilots, then are no pilots to be given to BPS to correct for the phase.

Figure 4.7 presents the BER curves corresponding to the Table 4.7. The test case PN_0PS_5 shows that the model is performing well, since its BER graph is top of the theoretical graph. However, test cases $P_{off}PN_0PS_5$, PN_7PS_5, and PN_8PS_5, it can be concluded that this version cannot perform well in presence of the phase noise and cannot compensate for it. Test case PN_7PS_5 BER curves are a bit better than the test $P_{off}PN_0PS_5$, but still they are very far from the theoretical BER graph. Therefore, the block-based version cannot detect the transmitted signal if there is a phase-offset. However, it might be able to correct if there is a small PN variance, or pilot-spacing.

Test case PN_8PS_5 shows that the performance of receiver improves a little bit for 16 QAM, while it does not change much for the 64 QAM. Test case PN_7PS_2 displays that if the pilot-spacing reduces to 2, which in other words, one information symbol and the next one is a pilot, this worsen the results. This might be because there is no BPS so it cannot use the pilots to correct for it. Finally, by comparing the BER curve for the test case PN_7PS_5, PN_7PS_2 and PN_7PS_{10}, where the only different factor is pilot-spacing, it is wrapped up that the pilot-spacing 5 has the best results. The first hunch was that it might get better if the pilot-spacing increases from 5 to 10, because the model might learn somehow the PN variance from the first few examples, and correct it for the next few symbols in the same block. However, it can be seen that the model cannot learn how to correct for PN using this model.

Table 4.6: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for the second version of block-based method. The NN was tested with and without BPS algorithm, PN, and phase offset for cost functions MSE and CRE.

Test Case	BPS	Phase Offset	PN	Pilot Spacing
$V_2P_{BPS}PN_0$	On	Off	0	5
V_2PN_0	Off	Off	0	5
$V_2PN_0P_{off}$	Off	On	0	5
V_2PN_7	Off	Off	10^{-7}	5
$V_2P_{BPS}P_{off}$	On	On	0	5
$V_2P_{BPS}PN_7$	On	Off	10^{-7}	5
Standard	On	Off	10^{-7}	5

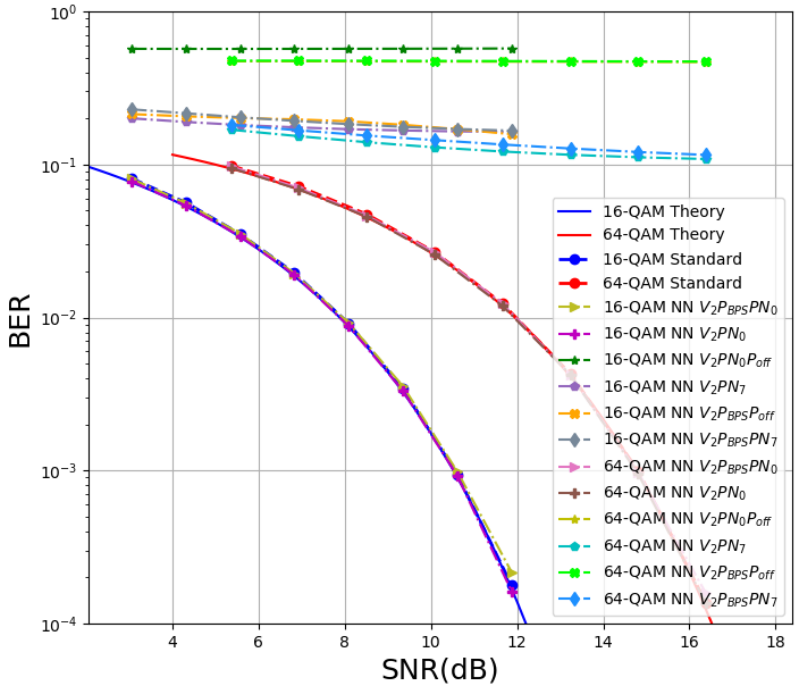


Figure 4.6: BER for theory, standard receiver, and NN receiver for second version of block-based approach. The NN was tested with and without BPS algorithm, PN, and phase offset for cost functions MSE and CRE.

Table 4.7: The various test cases implemented to investigate the impact of adding PN to the AWGN channel for the third version of block-based method. The NN was tested with and without PN, and phase offset for pilot spacing 2, 5 and 10. All test cases are performed without the BPS algorithm and with only the MSE cost function.

Test Case	Phase Offset	PN	Pilot Spacing
PN_0PS_5	Off	0	5
$P_{off}PN_0PS_5$	On	0	5
PN_7PS_5	Off	10^{-7}	5
PN_8PS_5	Off	10^{-8}	5
PN_7PS_2	Off	10^{-7}	2
PN_7PS_{10}	Off	10^{-7}	10
Standard	Off	10^{-7}	5

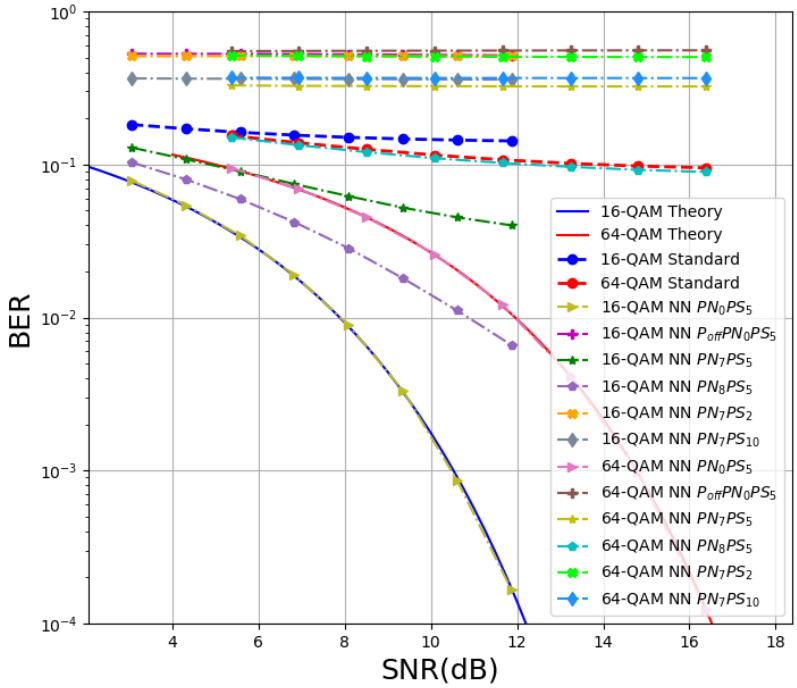


Figure 4.7: BER for theory, standard receiver, and NN receiver for third version of block-based approach. The NN was tested with and without PN, and phase offset for pilot spacing 2, 5 and 10. All test cases are performed without the BPS algorithm and with only the MSE cost function.

Therefore, we can conclude that block-based algorithm can cover for very low value of PN, and when the block length is not too long or short.

From all the above experiments, it can be derived that if the QAM modulation increases, the PN affects the BER curve more, which is completely expected. Since the decision boundaries are closer to each other for higher modulations, with same PN for both 16 and 64 QAM causing the same rotation, some symbols might be detected in the right boundaries for 16 QAM, while it will not be for 64 QAM. Finally, comparing all the BER curves to the one of standard receiver, it is obvious that standard receiver performs better than or the same as any other NN models.

4.3.3 Comparison between symbol- and block-based approach

The last inquiry is to compare the performance of symbol-based versus block-based approach. Results obtained in section 4.3.2, shows that the block-based versions cannot compensate for phase-offset under any kind of testing. Besides, in all the testing when there was no phase-offset and phase noise was 10^{-7}, the standard receiver outperform all of the models while it does not have any kind of phase-correction. This shows that the block-based version cannot perform as good as standard version under any circumstances.

Besides, the results obtained in section 4.3.1 display clearly that the NN model performs as good or better than the standard version, especially if the QAM constellation is increased.

Therefore, the symbol-based version with the BPS works much better than the block-based version. Also, the purpose of using block-based version is not to use the BPS but by comparing the block-based version without the phase-correction and symbol-based with BPS algorithm that, We can conclude that we need an algorithm to correct for phase or a more complicated NN, such as recurrent neural network which has a memory.

4.4 Testing the System in Presence of Non-linearities, In-phase Quadrature Imbalance and AWGN

The non-linearities, IQ imbalance and AWGN are implemented as impairments to the stochastic channel. A non-linear layer is added as one of the hidden layers in the NN model to compensate the non-linear impairments. Different arrangements of nonlinear activation functions and neurons are tested in order to give the comparison and seek for the best case. The evaluations of the non-linearities done in this section are divided into two aspects, Symbol-mode and block-based.

4.4.1 Performance of symbol-based approach

In this section, evaluations of adding non-linear hidden layer, using different activation functions and different NN model structures are done, as Table 4.8 - 4.11 showed, with non-linearities, IQ imbalance and AWGN. Alpha and sat-point in the table denotes the IQ imbalance and non-linearities level implemented to the channel

respectively, which is α and α_g in equation 2.2.1 and 2.2.3, as illustrated in section 2.2. The simulated BER curve of these evaluations are compared and discussed.

4.4.1.1 Adding a non-linear layer

To discuss the improvements of adding non-linear hidden layers, to start with, tanh, relu, softmax and sigmoid are used as activation functions, and with $4-60$ neurons are tested respectively in just one non-linear layer. Table 4.8 outlines one of the evaluations done in this section, which is relu function and 20 neurons used in this non-linear layer. And this is the arrangement that can give the best fit of BER curve among all the evaluations, which will be discussed in the next section.

Table 4.8: Comparison between with and without non-linear hidden layer in the presence of IQ-imbalance, nonlinearities and AWGN impairments in the channel

Impairments		Non-linear Layer		
α	Sat_point	Activation Function	Number of Nodes	Series/Parallel
0.1	0.95	None	None	None
0.1	0.95	Relu	20	Series

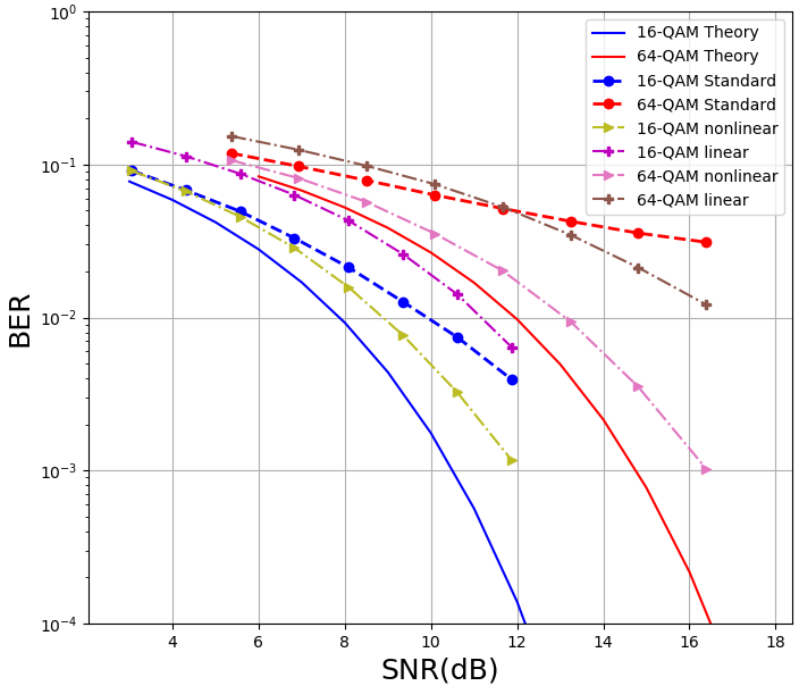


Figure 4.8: BER curve of theory, standard receiver, NN-based receiver with only one hidden linear layer, and NN-based receiver with one nonlinear layer (the activation function is relu, and with 20 neurons) and one linear layer in the hidden layers, for 16 and 64-QAM.

The improvements of adding a non-linear layer is seen in Figure 4.8. As can be seen in the figure, by adding non-linear layer, the NN curve is better more

than the standard and the NN model without non-linear layer. There still exists margin between the NN results and the theory one. When increasing the training parameters, for example, the number of training examples, epochs, and QAM order, there is no obvious improvements. This may because the error is not corrected in the receiver side, and the NN cannot compensate the impairments totally, but still we can give the conclusion that by adding one non-linear layer, the non-linearities, IQ imbalance and AWGN can be compensated properly.

Then if two or more non-linear layers are added to the NN model, for example, as shown in Table 4.9. Relu activation function with 20 neurons is chose since it appears better than other arrangements. The simulated results are showed in Figure 4.8.

Table 4.9: Comparison between one and more nonlinear layers in hidden layers in the presence of IQ-imbalance, nonlinearities and AWGN impairments in the channel.

No. of Hidden Layers	Activation Functions	No. of Nodes	Series/Parallel
2	Relu-Linear	20-322	Series
3	Relu-Relu-Linear	20-20-322	Series
4	Relu-Relu-Relu-Linear	20-20-322	Series

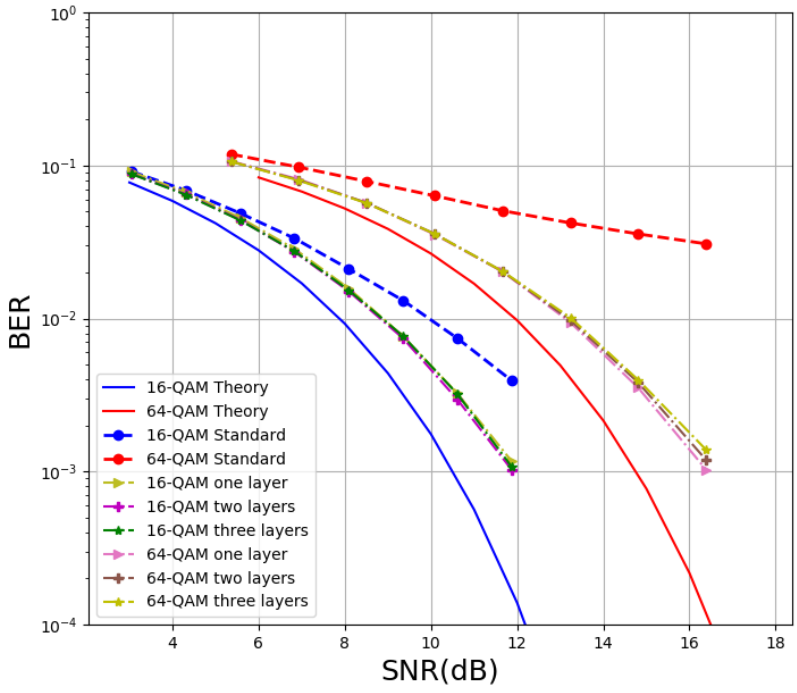


Figure 4.9: BER curve of theory, standard receiver, NN-based receiver with one, two and three nonlinear layers (the activation function is relu, and with 20 neurons) and one linear layer in the hidden layers, for 16 and 64-QAM.

As can be seen in the Figure 4.9, there is no improvements by adding more layers, even the results for adding just one nonlinear layer is slightly better than adding

more layers in high SNR. Therefore not only the non-linearities can be compensated properly by adding just one nonlinear layer.

4.4.1.2 Activation functions

The effects of different activation functions in the added non-linear layer are compared in this section. Table 4.10 shows part of the evaluations. The number of nodes of in certain layer is dependent of the activation function in this layer. After testing with $4 - 60$ neurons for each activation function, the number of nonlinear nodes in table 4.10 is the best arrangement with this function. Alpha and sat-point in the table denotes the IQ imbalance and non-linearities level implemented to the channel respectively, which is α and α_g in equation 2.2.1 and 2.2.3, as illustrated in section 2.2.

The simulated BER curve is showed in Figure 4.10 . The results are almost the same for different activation functions, this may because the number of QAM modulation order used for training examples is larger than the number of the QAM demodulation order, and the number of training examples is large enough, so that the NN model can learn well from more computational complex training data whatever using any activation function. When changing the parameters for training, for example, decreasing the number of training examples from $2,000,000$ to $1,000,000$, and decreasing the number of QAM modulation order for training from 256 to 64, the results is showed in Figure 4.11. In this case, the relu and softmax functions appears better than others.

As illustrated in **??**, for sigmoid function, it exists between 0 to 1. Therefore, it is especially used for models where there is a need to predict the probability as an output. As for tanh function, it is like a sigmoidal function but with range -1 to 1. Compared to sigmoid and tanh functions, softmax function is a generalization of logistic function that maps a K-dimensional vector of arbitrary real values to a K-dimensional vector of real values in the range $(0, 1)$ that add up to 1. It is more suitable when there are more than one output neurons in the system, so maybe that is why it appears better. For the relu function, it is much simpler and computationally more efficient compared with sigmoid and tanh, maybe that's the reason why it appears better than other functions in this system.

4.4.1.3 Series versus parallel structure

The comparisons of different structures of NN model are given in this section. Table 4.11 outlines part of these evaluations, where relu with 20 neurons and softmax with 6 neurons are the two combinations give better results. Series means the input data is given into the NN model layer by layer, and parallel means the input samples are also appended to the output samples from the final hidden layer before given to the output, as illustrated in chapter 2.4.2.4.

The simulated BER curves are showed in Figure 4.12.

As introduced before in chapter 2.4.2.4, by using the parallel structure, the NN can focus more on the nonlinear portion in the transmitted signals so that it may give better results than series in theory. But in Figure 4.12, the results for the series and parallel structure are almost the same. This may be because the non-linearities

Table 4.10: Comparison among different activation functions, for adding one non-linear layer to the hidden layer of the NN model, with IQ imbalance, non-linearities and AWGN added as impairments in the stochastic channel.

Impairments		Non-linear Layer		
Alpha	Sat_point	Activation Function	Nonlinear Nodes	Series/Parallel
0.1	0.95	Tanh	16	Series
0.1	0.95	Relu	20	Series
0.1	0.95	Sigmoid	18	Series
0.1	0.95	Softmax	6	Series

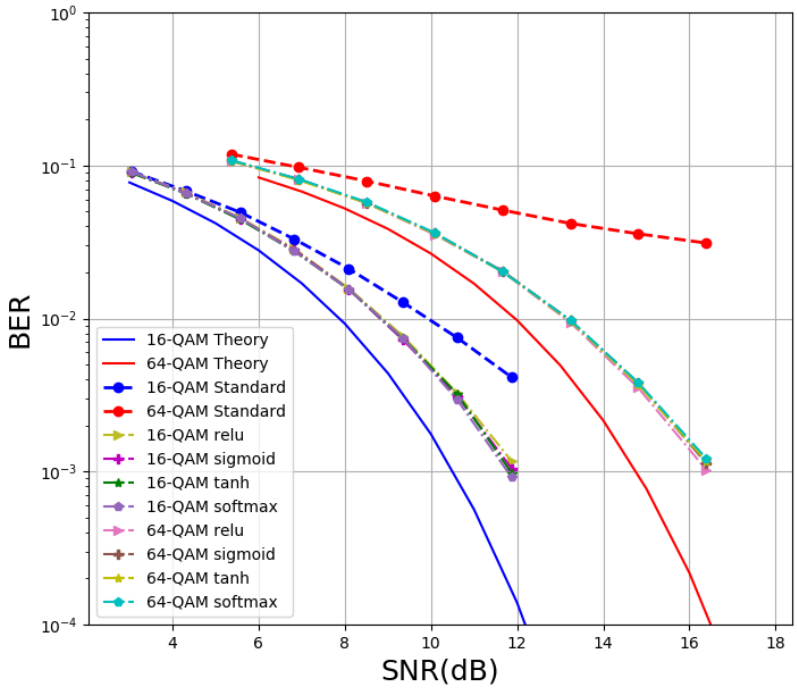


Figure 4.10: BER curve of theory, standard receiver, NN-based receiver with different activation functions and neurons in the nonlinear hidden layer, for 16 and 64-QAM.

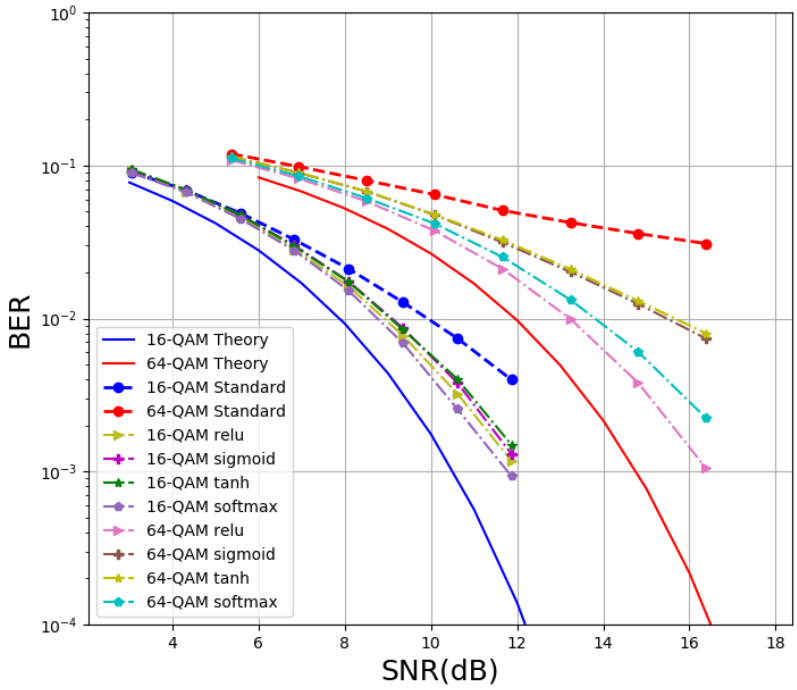


Figure 4.11: BER curve of theory, standard receiver, NN-based receiver with different activation functions and neurons in the nonlinear hidden layer, for 16 and 64-QAM. The training parameters are set to be 64-QAM and 100,000 examples.

Table 4.11: Comparison between the series and parallel structure of the model, with IQ imbalance, non-linearities and AWGN added as impairments in the stochastic channel.

Impairments		Non-linear Layer		
Alpha	Sat_point	Activation Function	Nonlinear Nodes	Series/Parallel
0.1	0.95	Relu	20	Series
0.1	0.95	Relu	20	Parallel
0.1	0.95	Softmax	6	Series
0.1	0.95	Softmax	6	Parallel

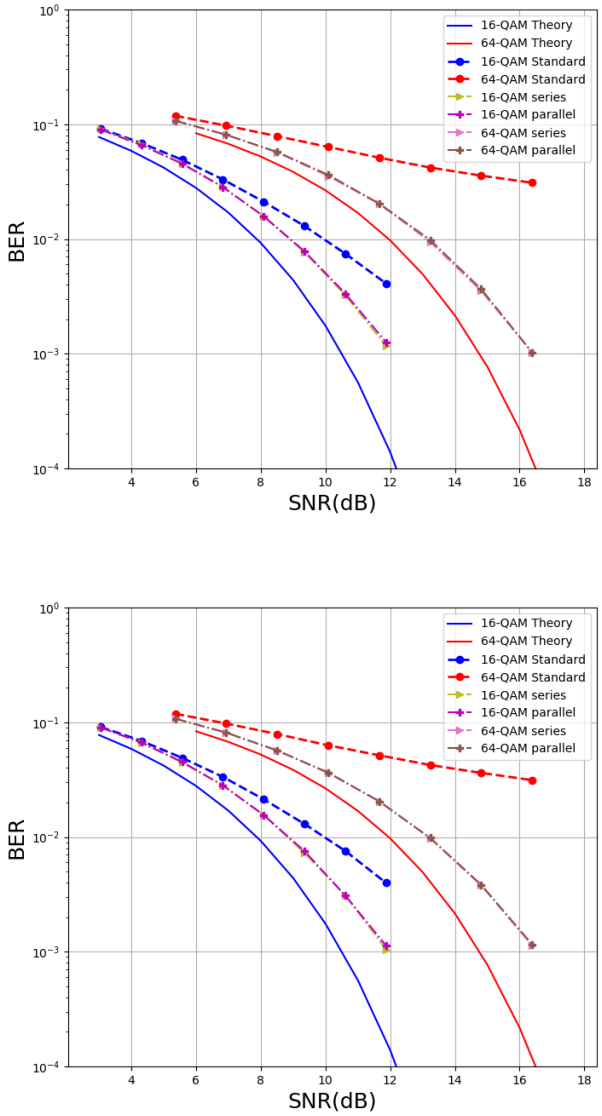


Figure 4.12: BER curve of theory, standard receiver, NN-based receiver with series and parallel structure in the networks, for 16 and 64-QAM. The upper figure is with relu function and 20 neurons in the nonlinear hidden layer, and the lower figure is with softmax function and 6 neurons in the nonlinear hidden layer.

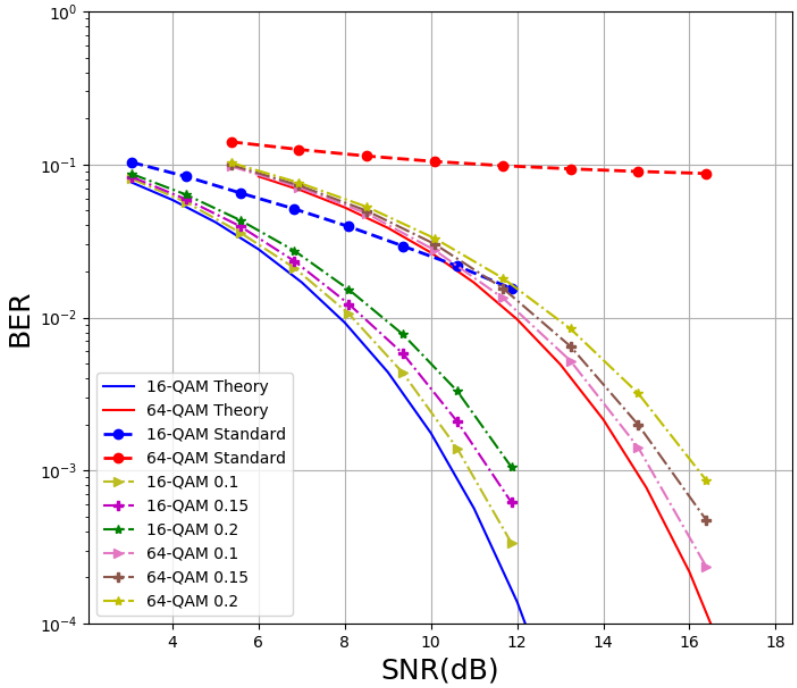


Figure 4.13: BER curve of theory, standard receiver, NN-based receiver with added IQ imbalance in the channel of different alpha values, for 16 and 64-QAM, with IQ imbalance and AWGN added as impairments to the stochastic channel.

added in the transmitted signal are small so that the NN can compensate for them properly.

4.4.1.4 Stability of the NN model

This section aims to study the stability of the model when the level of added IQ imbalance and non-linearities changed. The evaluation is done with a better case of the model that using relu function and 20 neurons in the nonlinear hidden layer, as illustrated in the previous section.

First, the evaluation of changing the value of alpha for IQ imbalance is done, with series structure and relu activation function in the model, and with. The results with alpha equals to 0.1, 0.15 and 0.2 are showed in Figure 4.13. It can be seen that the results did not become much worse when increasing the IQ imbalance level, while the results of the standard receiver became very worse, almost a straight line in 64-QAM. Therefore, one can infer that the selected NN model is stable with this IQ imbalance impairment.

Then the evaluation of increasing the non-linearities level by changing the value of saturation percent from 0.9 to 0.85 and 0.95 is done. The corresponding results are showed in Figure 4.14.

When introducing more non-linearities to the channel, as can be seen in the Figure 4.14, the results did not become much worse and the curves move less than

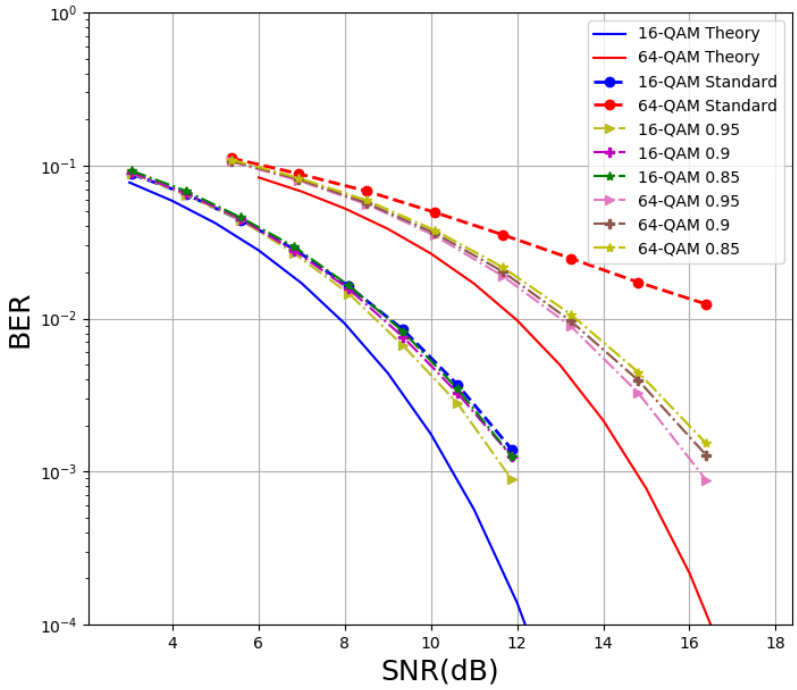


Figure 4.14: BER curve of theory, standard receiver, NN-based receiver with added non-linearities in the channel of different saturation percent, for 16 and 64-QAM, with IQ imbalance and AWGN added as impairments to the stochastic channel.

the conventional receiver.

4.4.2 Block-based compared to symbol-based approach

In this section, the performance of symbol-based versus block-based approach is investigated in presence of non-linearity and IQ imbalance in an AWGN channel. Therefore different tests are performed which are summarized in Table 4.12. In all the testing there is no PN, phase-offset and BPS and the NN model has 322 linear node, 20 non-linear node and series structure. Also, for all the block-based versions the pilot spacing or the block-length is chosen to be 5.

Figure 4.15 displays the results of test cases done in table 4.12. As it can be seen that block-based version performs the worth compare to the standard and symbol-based approach. Therefore, block-based approach is not fit in presence of IQ imbalance and non-linearities.

4.5 Testing the System in Presence of Phase Noise, Non-linearities, In-phase Quadrature Imbalance and Additive White Gaussian Noise

This section tests the performance of the overall system in presence of all the impairments, PN, non-linearities, IQ imbalance and AWGN.

Results obtained in section 4.4.2 suggest that the block-based approach does not perform adequately for PN impairment. Results from Section 4.3.3 show similar results in presence of non-linearities and IQ imbalance. Therefore, this section focuses merely on the performance of the symbol-based approach when all the impairments are added to the channel.

A set of different test cases are performed to evaluate the performance of the system given symbol-based approach and are summarized in Table 4.13. In this test, The alpha for non-linearity impairment is set to 0.1, PN is set to 10^{-7} and the saturation point for IQ imbalance impairment is set to be 0.95. Also, there exist the BPS algorithm with a pilot-spacing of 5 which corresponds to 20 percent overhead.

To test the performance of the model with and without adding a non-linear layer, a non-linear layer is added in some cases. This non-linear layer is chosen to use the relu activation function with 20 nodes with a series structure, which were the best results obtained in Section 4.4.1.3.

The BER curves corresponding to test cases $C_{CRE}P_{BPS}L_{Nonlin}$ and $C_{CRE}P_{BPS}$ in Figure 4.16 demonstrate that the model trained with CRE cost function is not able to retrieve any data in presence of all the impairments at the same time. Test cases $C_{MSE}P_{BPS}L_{Nonlin}$, $C_{MSE}P_{BPS}$, and $C_{MSE}L_{Nonlin}$ corresponds to the MSE cost functions. By comparing these three cases, it is obvious that if there exist a phase-correction, then the system performs better. However, Figure 4.16 displays that none of the NN models performs as good as standard receiver with a BPS.

Table 4.12: Comparison of the results between using symbol-based approach and different block-based approaches, for MSE cost function and without PN in the channel. The various test cases implemented to compare the performance of the symbol-based versus block-based method after adding non-linearities IQ imbalance, and AWGN to the channel.

Test Case	Alpha	Sat_point	Activation Func.	Cost Func.	Symbol/Block
S_{based}	0.1	0.95	Relu	MSE	Symbol-based
$B_{based}V_1$	0.1	0.95	Relu	MSE	Block-based V1
$B_{based}V_2$	0.1	0.95	Relu	MSE	Block-based V2
$B_{based}V_3$	0.1	0.95	Relu	MSE	Block-based V3
Standard	0.1	0.95	None	None	None

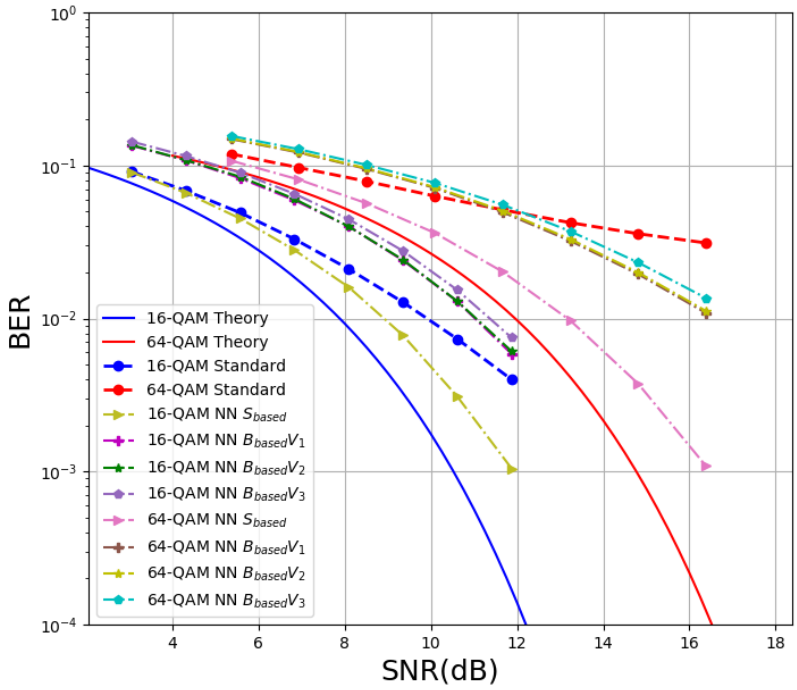


Figure 4.15: BER for theory, standard receiver, and NN-based receiver. The NN was tested with symbol-based approach and three different block-based approaches in presence of non-linearities and IQ imbalance.

Table 4.13: The various test cases implemented to investigate the impact of adding all impairments to the channel for symbol-based method. The NN was tested with and without BPS algorithm and non-linear layer given different cost functions.

Test Case	BPS	Cost Function	Non-linear Layer
$C_{CRE}P_{BPS}L_{Nonlin}$	On	CRE	On
$C_{CRE}P_{BPS}$	On	CRE	Off
$C_{MSE}P_{BPS}L_{Nonlin}$	On	MSE	On
$C_{MSE}P_{BPS}$	On	MSE	Off
$C_{MSE}L_{Nonlin}$	Off	MSE	On
Standard	Off	None	None

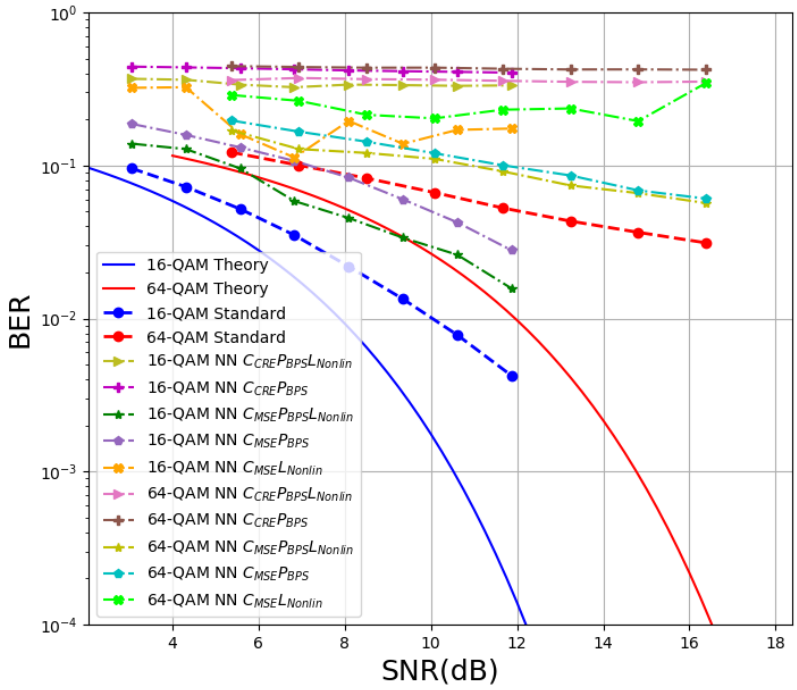


Figure 4.16: BER for theory, standard receiver, and NN-based receiver in presence of all three impairments (PN, IQ imbalance and non-linearities) in the AWGN channel. The NN was tested with and without BPS algorithm and non-linear layer given different cost functions.

5
Conclusion

In this thesis, we have implemented a communication system and applied a NN model in the receiver to demodulate the noisy signal and compare the performance with the conventional communication system. The channel model is an AWGN channel with three impairments added: IQ imbalance, non-linearities and PN. The IQ imbalance and non-linearities can be compensated by the NN with one nonlinear-hidden layer in the model. For the PN impairment, there were two models suggested, which were symbol-based and block-based structure. The symbol-based structure NN model with a customized function, could compensate the PN impairment if it was used together with a phase-correction algorithm. However, this model could not compensate for PN by itself without this algorithm. Therefore, a block-based structure was introduced which tried to compensate for the PN, but it could not compensate for the PN.

However, there are some aspects where more studies potentially could further this field of study. First, the NN model implemented in this thesis can just compensate the impairments separately by using different cost functions and model structures. When adding all the three impairments (IQ imbalance, non-linearities and PN) in the channel, the NN model cannot learn well and compensate for them that the model used in this thesis could be improved. Second, the AWGN channel model used in this report only considers three impairments. This leads to two future implementations which could largely future this study: one is that there are more impairments can be added to the channel such as phase offset and frequency offset. Also the non-linearities implemented in this thesis are based on the theoretical equation. In order to simulate actual measurements of the non-linearities, the Chalmers Weblab is a good tool to perform real high frequency measurements without having to purchase or manage complicated high frequency instruments such as signal generator, oscilloscope and amplifiers. Another limitation is that the simulated system is not a real time system and all the impairments are not provided by actual components. It is more valuable to implement and study the performance of the NN based on real time communication system. Third, the channel model used in this thesis is an AWGN channel which is a basic noise model to mimic the effect of many random processes that occur in nature. This can be extended to a more complicated channel such as MIMO where complicated signal processing algorithms can be replaced with a NN model. Last, the NN model in this thesis is replaced instead of the down-sampling and matched filtering blocks in the receiver side. It can be extended further to also perform the de-mapping in the receiver.

Bibliography

[1] Wikimedia Commons. File:artificial neural network with chip.jpg — wikimedia commons, the free media repository, 2020. [Online; accessed 21-September-2020].

[2] T. S. Rappapor. *Wireless Communications: Principles and Practice, 2nd ed.* Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[3] T. Schenk. *RF Imperfections in High-rate Wireless Systems: Impact and Digital Compensation*. Springer Netherlands, 2008.

[4] J. Ryckaert, P. De Doncker, R. Meys, A. de Le Hoye, and S. Donnay. Channel model for wireless communication around human body. *Electronics Letters*, 40(9):543–544, 2004.

[5] K. Vasudevan. Coherent detection of turbo coded ofdm signals transmitted through frequency selective rayleigh fading channels. In *2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 1–6, 2013.

[6] S. Ali, E. Hossain, and D. I. Kim. Non-orthogonal multiple access (noma) for downlink multiuser mimo systems: User clustering, beamforming, and power allocation. *IEEE Access*, 5:565–577, 2017.

[7] A. Masoomzadeh-Fard and S. Pasupathy. Nonlinear equalization of multipath fading channels with noncoherent demodulation. *IEEE Journal on Selected Areas in Communications*, 14(3):512–520, 1996.

[8] A. Trivedi and R. Gupta. Improved ml channel estimation for uplink mc-cdma systems in closely spaced multipath channels. *Wireless Personal Communications*, 52(2):341–357, 2008.

[9] Y. Li, X. Sha, F. Zheng, and K. Wang. Low complexity equalization of hcm systems with dpfft demodulation over doubly-selective channels. *IEEE Signal Processing Letters*, 21(7):862–865, 2014.

[10] Z. Xie, X. Chen, and X. Liu. Joint channel estimation and equalization for mimo-scfde systems over doubly selective channels. *Journal of Communications and Networks*, 19(6):627–636, 2017.

[11] Y. Gao and Y. Chen. Channel estimation for af relaying using ml and map. *Wireless Networks*, 24(8):3161–3170, 2018.

[12] T. Fath and H. Haas. Performance comparison of mimo techniques for optical wireless communications in indoor environments. *IEEE Transactions on Communications*, 61(2):733–742, 2013.

[13] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui. Deep learning for super-resolution channel estimation and doa estimation based massive mimo system. *IEEE Transactions on Vehicular Technology*, 67(9):8549–8560, 2018.

[14] J. Wang, J. Yang, J. Xiong, H. Sari, and G. Gui. Shafa: sparse hybrid adaptive filtering algorithm to estimate channels in various snr environments. *IET Communications*, 12(16):1963–1967, 2018.

[15] V. Raj and S. Kalyani. Backpropagating through the air: Deep learning at physical layer without channel models. *IEEE Communications Letters*, 22(11):2278–2281, 2018.

[16] S. Schiessl, H. Al-Zubaidy, M. Skoglund, and J. Gross. Delay performance of wireless communications with imperfect csi and finite-length coding. *IEEE Transactions on Communications*, 66(12):6527–6541, 2018.

[17] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

[18] Michael Jordan and T.M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science (New York, N.Y.)*, 349:255–60, 07 2015.

[19] T. O'Shea and J. Hoydis. An introduction to deep learning for the physical layer. *IEEE transactions on cognitive communications and networking*, 3(4):563–575, Dec. 2017.

[20] M. Zhang, Z. Liu, L. Li, and H. Wang. Enhanced efficiency bpsk demodulator based on one-dimensional convolutional neural network. *IEEE Access*, 6:26939–26948, 2018.

[21] Z. Zhou. *Machine Learning*. Tsinghua University Press, 2016.

[22] Jürgen Schmidhuber. Review. *Neural Networks*, 61(Complete):85–117, 2015.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] Muhammet Nuri Seyman and Necmi Taşpınar. Channel estimation based on neural network in space time block coded mimo–ofdm system. *Digital Signal Processing*, 23(1):275 – 280, 2013.

[25] S. Hu, Y. Yao, and Z. Yang. Mac protocol identification using support vector machines for cognitive radio networks. *IEEE Wireless Communications*, 21(1):52–60, 2014.

[26] Ai sheng LIU and Qi ZHU. Automatic modulation classification based on the combination of clustering and neural network. *The Journal of China Universities of Posts and Telecommunications*, 18(4):13 – 38, 2011.

[27] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018.

[28] K. Kim, J. Lee, and J. Choi. Deep learning based pilot allocation scheme (dl-pas) for 5g massive mimo system. *IEEE Communications Letters*, 22(4):828–831, 2018.

[29] F. Liang, C. Shen, and F. Wu. An iterative bp-cnn architecture for channel decoding. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):144–159, 2018.

[30] H. Ye, G. Y. Li, and B. Juang. Power of deep learning for channel estimation and signal detection in ofdm systems. *IEEE Wireless Communications Letters*, 7(1):114–117, 2018.

[31] Timothy J. O'Shea, Tugba Erpek, and T. Charles Clancy. Deep learning based mimo communications, 2017.

[32] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan. Channel agnostic end-to-end learning based communication systems with conditional gan. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–5, 2018.

[33] Meng Fan and Lenan Wu. Demodulator based on deep belief networks in communication system. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pages 1–5, 2017.

[34] Stuart; University of California Russell, Tom; Oregon State University Dietterich, Eric; Microsoft Horvitz, Bart; Cornell University Selman, Francesca; University of Padova Rossi, Demis; DeepMind Hassabis, Shane; DeepMind Legg, Mustafa; DeepMind Suleyman, Dileep; Vicarious George, and Scott; Vicarious Phoenix. Letter to the editor: Research priorities for robust and beneficial artificial intelligence: An open letter. 2015.

[35] Wikipedia contributors. Communications system — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Communications_system&oldid=967003659`, 2020. [Online; accessed 20-July-2020].

[36] F. Yu and P. Zhang. All-digitized implementation of qam modulation and demodulation. *Modern Electronic Technique*, 28:53–55, 2015.

[37] Wikipedia contributors. Root-raised-cosine filter — Wikipedia, the free encyclopedia, 2017. [Online; accessed 22-July-2020].

[38] Daniel Silveira, Michael Gadringer, M. Mayer, and Gottfried Magerl. Analysis of rf-power amplifier modeling performance using a 16-qam modulation over awgn channels. pages 164 – 167, 03 2006.

[39] Wikipedia contributors. Matched filter — Wikipedia, the free encyclopedia, 2020. [Online; accessed 9-September-2020].

[40] A.V. Oppenheim, R.W. Schafer, Massachusetts Institute of Technology. Center for Advanced Engineering Study, Institute of Electrical, and Electronics Engineers. Educational Activities Board. *Digital Signal Processing*. MIT video course. Prentice-Hall, 1975.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014.

[42] David E. Rumelhart and Geoffrey E. Hinton. Learning representations by back-propagating errors. 10 1986.

[43] Timo Pfau, Sebastian Hoffmann, and Reinhold Noé. Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for m-qam constellations. *J. Lightwave Technol.*, 27(8):989–999, Apr 2009.

[44] T. Pfau, S. Hoffmann, and R. Noe. Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for m -qam constellations. *Journal of Lightwave Technology*, 27(8):989–999, 2009.

[45] Chance Tarver, Alexios Balatsoukas-Stimming, and Joseph R. Cavallaro. Design and implementation of a neural network based predistorter for enhanced mobile broadband, 2019.