University of BRISTOL

Department of Electrical and Electronic Engineering

MSc in Wireless Communications and Signal Processing

FlexRx: an end-to-end Deep Learning-Based OFDM Receiver

Yuzheng He

rc23017

September 2023

DECLARATION AND DISCLAIMER

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Postgraduate Programmes and that it has not been submitted for any other academic award.

Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted from the work of others, I have included the source in the references/bibliography.

Any views expressed in the dissertation are those of the author.

The author confirms that the printed copy and the electronic version of this thesis are identical.

Signed: Yuzheng He

Date: 2024.9.2

UNIVERSITY OF BRISTOL THESIS DEPOSIT AGREEMENT

Name : Yuzheng He

Title of Dissertation: FlexRx: an end-to-end Deep Learning-Based OFDM Receiver

Supervisor: Dr. Rasheed Hussain

Department: Department of Electrical and Electronic Engineering

I AGREE AS FOLLOWS

1. Library Access

a. I agree that the Thesis may be made available for consultation in the University of Bristol library and for inter-library lending for use in another library and may be copied in full or in part for any bona fide library or research worker, on the understanding that users shall be made aware of their obligations to me under copyright, i.e. that no quotations or significant extracts may be copied or used without sufficient acknowledgement.

2. Electronic Access

a. I agree that the Thesis may be made available for consultation in full without charge on the Internet via the University of Bristol's online digital repository of dissertations and with the British Library to be made available via the EThOS system. Rights granted to the University of Bristol and the British Library through this Agreement are non-exclusive. I remain free to publish the Thesis elsewhere in its present version or future versions.

b. I agree that the administrators of the University of Bristol Thesis Repository Service may, without changing content, digitise and migrate the Thesis to any medium or format for the purpose of future preservation and accessibility.

c. I warrant that the Thesis is original and that I am the author and owner of the copyright in the Thesis. I grant the University of Bristol and (where appropriate) the British Library a licence to make available the Thesis in digitised format through the University of Bristol repository and through the British Library via the EThOS system for the purposes of non-commercial research, private study, criticism, review and news reporting, illustration for teaching, and/or other educational purposes in electronic or print form.

d. I warrant that if the Thesis does include any substantial subsidiary material owned by third-party copyright holders, I have sought and obtained permission to include it in any version of my Thesis available in digital format via a stand-alone device or a communications network and that this permission encompasses the rights that I grant to the University of Bristol and to the British Library.

e. I understand and agree that neither the University of Bristol nor (where appropriate) the British Library have any obligation to take legal action on my behalf, or other rights holders, in the event of infringement of intellectual property rights, breach of contract or any other right in the Thesis.

Signed: Yuzheng He .... Dated: 2024.9.2 ....

Abstract

We explore a novel deep learning-based end-to-end Orthogonal Frequency Division Multiplexing (OFDM) receiver in this thesis, namely FlexRx, which is designed to optimize signal reception performance in modern wireless communication systems by reducing bit error rate (BER), enhancing system flexibility, and lowering computational complexity.

The proposed FlexRx includes four modules designed for channel estimation, noise estimation, channel equalization, and demapping. Our experiments show that, through the tight integration of four modules (ChanEstiNet, NoiseEstiNet, EqualizeNet, and DemappingNet), the FlexRx OFDM receiver exhibits superior signal recovery capabilities compared with other traditional receivers and existing deep learning-based receivers under various testing conditions, including different modulation orders, code rates, user speeds and wireless environments.

Despite excellent BER performance, our experiments also show that FlexRx has good flexibility, balancing minimal parameter size with fast processing time, ensuring high adaptability and sustained efficiency across diverse operating scenarios. These results underscore FlexRx's potential for advancing next-generation OFDM receiver designs, promoting its suitability for high-speed, high-reliability communication networks.

Content

List of Tables

List of Figures

Chapter 1. Introduction

Modern wireless communication systems require high data rates, robust transmission, and efficient spectrum use. Orthogonal Frequency Division Multiplexing (OFDM) [11] has become essential in these systems due to its ability to meet these demands, particularly in challenging environments with severe multipath propagation.

Despite their advantages, OFDM systems face challenges such as sensitivity to channel fading, inter-symbol interference (ISI), Doppler Shift, and the requirement for precise synchronization. Due to these challenges, several advanced signal processing techniques are usually employed particularly in the receiver design to ensure reliable communication.

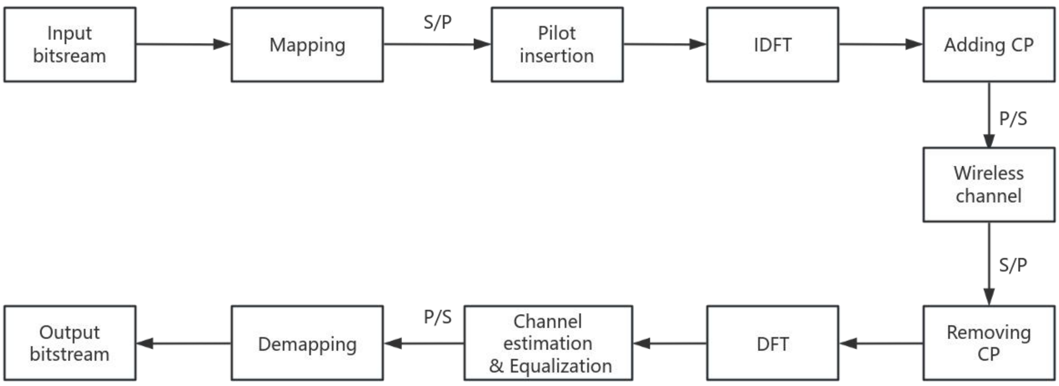

Figure. 1 A basic OFDM system

Figure 1 presents an overall OFDM system covering both signal transmission and reception processes. In addition to these basic steps provided in Figure 1, some other stages may be included in practical OFDM applications, such as channel coding and rate matching [8]. During signal reception, several algorithms are carefully designed to achieve channel estimation, equalization, and demapping.

In wireless communication systems, channel fading, caused by factors such as multipath propagation and environment interference, leads to signal distortion. As a result, estimating the characteristics of the channel for ensuring reliable communications becomes a crucial step. Traditionally, classical algorithms like Least Square (LS) and Minimum Mean Square Error (MMSE) [1] have been commonly employed for this purpose. In this step, the estimated channel response can be obtained, and is usually utilized by the subsequent equalization procedure.

After accurate channel estimation, equalization is usually employed to correct the distortions introduced by the channel, restoring the received signal to its original transmitted symbols. The channel matrix obtained in channel estimation, along with the received signal, are utilized together to restore the correct transmitted symbols in this step.

Finally, signal demapping is employed to translate the equalized symbol into its original bit form. This step can produce either hard bits, with definitive binary decisions, or soft bits, which carry additional reliability information, such as Log-Likelihood Ratios (LLRs).

Noise estimation is not an explicit step in this OFDM system, but this step can be very useful in some algorithms during signal reception, such as MMSE. These algorithms usually require accurate noise variance estimates as one of their inputs, helping to differentiate between signal and noise and thus obtaining accurate calculation results.

However, compared with traditional methods, the integration of deep learning approaches into these signal reception processes represents a significant advancement in the field, offering improved performance and adaptability in complex communication environments. Deep learning-based receivers, such as our proposed FlexRx, are designed to enhance the BER performance as well as the flexibility of the receiver to adapt to diverse operating conditions.

In this thesis, we designed an end-to-end deep learning-based OFDM receiver, FlexRx, that integrates different modules designed for channel estimation, noise estimation, equalization, and demapping. The main objectives of our research are as follows:

1) Simulate a modern OFDM wireless communication system and generate OFDM signal datasets under the OFDM environment.

2) Develop sub-networks for each stage of signal reception—channel estimation, noise estimation, equalization, and demapping—to create the complete FlexRx.

3) Evaluate the Bit Error Rate (BER) performance of FlexRx under various conditions and compare it with both traditional and existing deep learning-based receivers.

4) Analyze the flexibility of FlexRx and compare its computational complexity with other traditional and existing deep learning-based receivers.

Chapter 2. Literature Review

2.1 Channel Estimation

2.1.1 Least Square

The Least Squares (LS) [1] estimation is known as a simple but widely used traditional channel estimation method. It's basic principle is to minimize the sum of the squared differences between the received signal and the estimated signal.

$$\widehat{H}_{LS} = X^{-1}Y \tag{1}$$

In Equation (1), Y is the received signal, while X is the known transmitted signal. In most cases, several interpolation schemes like Spline Interpolation are utilized instead of using the equation directly.

2.1.2 Linear Minimum Mean Square Error

The Minimum Mean Square Error (MMSE) [1] estimation aims to minimize the mean square error between the estimated and actual channel by considering both noise and channel statistics. The MMSE estimate for an OFDM system can be expressed as:

$$\widehat{H}_{MMSE} = R_{HH}[R_{HH} + (XX^H)^{-1}N{\delta_n}^2]^{-1}\widehat{H}_{LS} \tag{2}$$

$$R_{HH} = E[H_p H^{\dagger}] \tag{3}$$

In Equation (2), R_{HH} is the cross covariance and is given by Equation (3). Besides, δ_n is the noise variance and N is the number of subcarriers. This is much more complicated than LS, and an approximation form of the MMSE, namely Linear Minimum Mean Square Error (LMMSE) [2], is most commonly used.

$$\widehat{H}_{LMMSE} = R_{HH}(R_{HH} + \frac{\beta}{SNR}I)^{-1}\widehat{H}_{LS} \tag{4}$$

Equation (4) describes how LMMSE works and shows lower complexity mainly because the part of matrix inversion is optimized to reduce complexity.

2.1.3 ChannelNet

ChannelNet [3] was the first deep learning-based channel estimator utilizing super-resolution techniques. In this method, channel response, which is a matrix with complex values, was treated as two 2D images. Then a super-resolution convolutional neural network (SRCNN) [4] cascaded with a denoising convolutional neural network (DnCNN) [5] was utilized to process the input channel response. The results of their research showed that its performance is highly competitive with traditional algorithms such as MMSE.

2.2 Channel Equalization

2.2.1 Zero Forcing

Zero Forcing (ZF) [6] is a straightforward equalization method used to invert the channel effects by applying a filter that forces the combined channel and equalizer response to be as close as possible to the identity matrix. Equation (5) describes the ZF process where H is the channel matrix and H^H is the conjugate transpose.

$$X_{ZF} = (H^H H)^{-1}H^H Y \tag{5}$$

2.2.2 Linear Minimum Mean Square Error

Similarly, as a simplified version of MMSE concept, the Linear Minimum Mean Square Error (LMMSE) [6] equalization method combines considers both channel inversion and minimizing noise. The principal for the LMMSE equalizer can be generally expressed as:

$$X_{LMMSE} = (H^H H + {\delta_n}^2 I)^{-1} H^H Y \tag{6}$$

2.3 Demapping

2.3.1 Max-Log-MAP

The Max-Log-MAP (Max-Log) [6] demapping method is used to compute log-likelihood ratios (LLRs) after equalization, which simplifies the calculation by focusing on the most likely constellation points for each bit. Equation (7) describes how the LLR is calculated using this algorithm, where y is the received signal, δ_n is the noise variance, $C_{i,0}$ and $C_{i,1}$ are sets of constellation points for which the i-th bit is '1' and '0', respectively.

$$LLR(i) \approx max_{c \in C_{i,0}}\left(-\frac{|y-c|^2}{\delta_n}\right) - max_{c \in C_{i,1}}\left(-\frac{|y-c|^2}{\delta_n}\right) \tag{7}$$

2.3.2 A Posteriori Probability

A posteriori probability (APP) [6] is another demapping method that calculates the LLR for each bit based on the a posteriori probabilities of the transmitted bits given the received signal. Unlike the Max-Log method, which approximates the LLR using the most likely constellation points, the APP method considers all possible constellation points, weighted by their probabilities, to provide a more accurate calculation.

$$LLR(i) = ln\left(\frac{Pr(b_i=1|y,p)}{Pr(b_i=0|y,p)}\right) = ln\left(\frac{\sum_{c \in C_{i,1}} Pr(c|p)\, exp(-\frac{1}{\delta_n}|y-c|^2)}{\sum_{c \in C_{i,0}} Pr(c|p)\, exp(-\frac{1}{\delta_n}|y-c|^2)}\right) \tag{8}$$

In Equation (8), $p = [p_0, \ldots, p_{k-1}]$ is the vector of LLRs that represents prior knowledge on the K bits mapped to a constellation point, while $Pr(c|p)$ is the prior probability of the constellation symbol c given the prior knowledge p.

2.4 Neural Receiver

Neural Receiver is an OFDM system receiver implemented using neural networks, as demonstrated in the Sionna library tutorial [6]. It takes the received resource grid as its input and outputs LLRs of the transmitted coded bits. These LLRs are then used by the outer decoder to reconstruct the transmitted information bits. Figure 2 illustrates the architecture of the Neural Receiver, which includes convolutional and residual blocks [16] designed to process the input data, showing how the neural network layers are structured to perform the signal reception tasks.

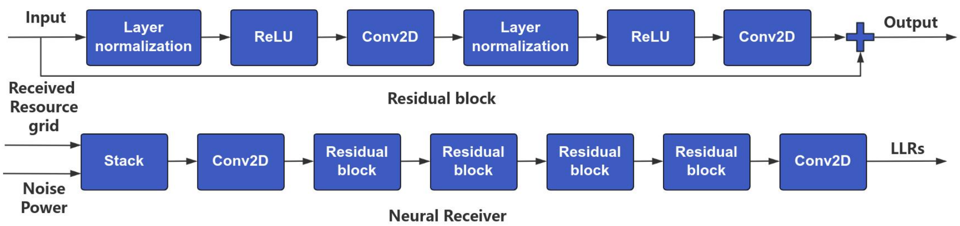

Figure. 2 Neural Receiver architecture

Chapter 3. FlexRx Architecture Design

3.1 FlexRx Overall Structural Framework

This section details the architecture design of the FlexRx, a complete OFDM receiver. Figure 3 illustrates the overall OFDM system, showcasing the end-to-end signal processing stages from transmission to reception. Within this framework, the proposed FlexRx is specifically designed to optimize the OFDM reception process by efficiently extracting soft information of the encoded bits from the received signal.

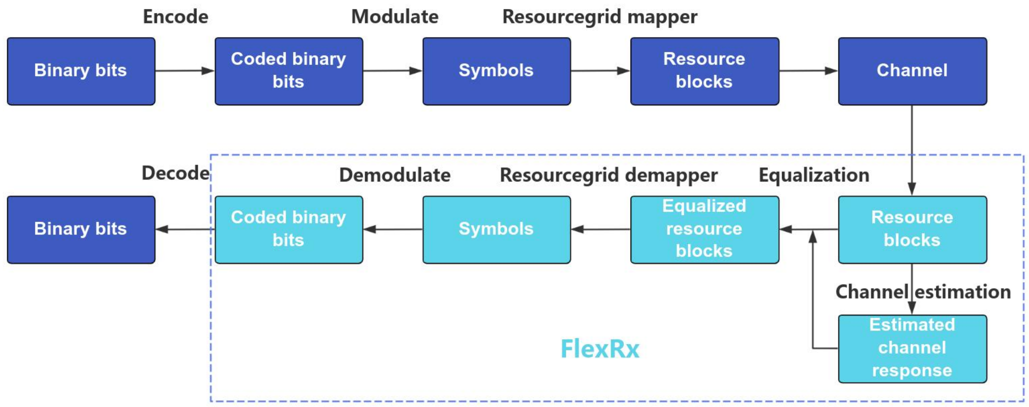

Figure. 3 End-to-end signal processing in an OFDM system with FlexRx

Figure 4 shows the overall structure of FlexRx, which integrates four modules to form a comprehensive OFDM receiver. These modules are ChanEstiNet, NoiseEstiNet, EqualizeNet, and DemappingNet, each playing a crucial role in the signal processing chain. If the noise variance is known, NoiseEstiNet can be bypassed. Additionally, the output of each module is valuable, allowing parts of FlexRx to be selectively utilized depending on specific needs, thereby demonstrating the receiver's flexibility to adapt to different situations.

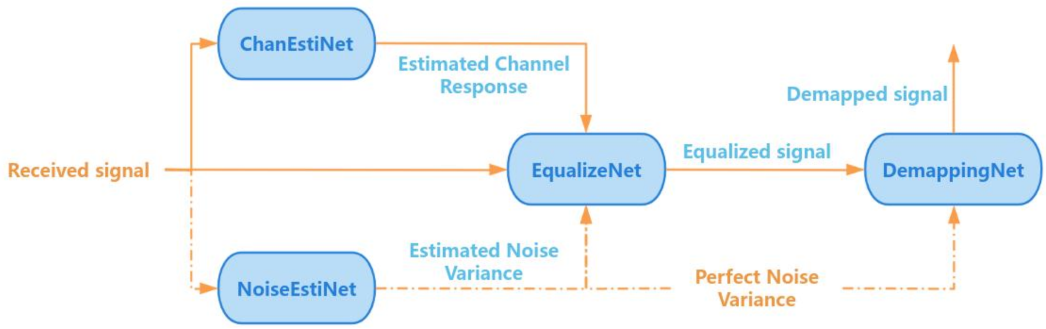

Figure. 4 FlexRx network structure

3.2 FlexRx Modular Design

3.2.1 ChanEstiNet

ChanEstiNet is designed for channel estimation in the FlexRx OFDM receiver, focusing on accurately estimating the channel response from the received signals. Inspired by ChannelNet, ChanEstiNet uses a super-resolution approach to enhance the precision of the channel estimation, treating the real and imaginary parts of the estimated channel response matrix as separate 2D images.

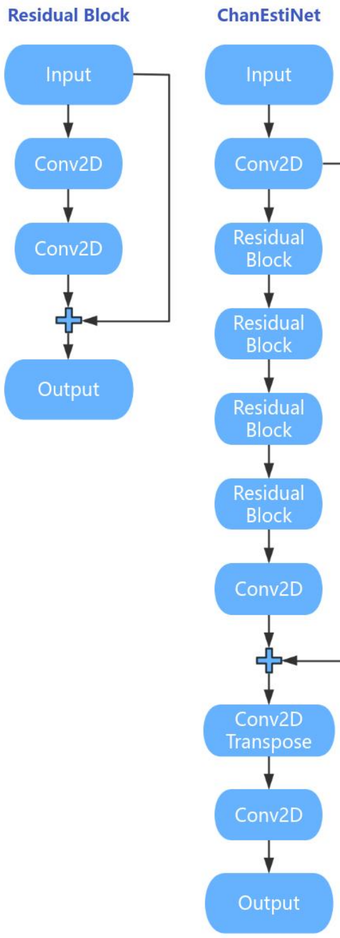

Figure. 5 ChanEstiNet structure

For the implementation of ChanEstiNet, we employ the Enhanced Deep Super-Resolution Network (EDSR) [7], a well-established super-resolution network known for its effectiveness in image processing tasks. Figure 5 shows that the EDSR structure used in ChanEstiNet consists of a series of convolutional layers and residual blocks for feature extraction. ReLU activation function is used after each convolution operation but is omitted from the figure. Moreover, a convolutional transpose layer is utilized for upsampling, allowing the model to predict a higher resolution channel response from the low-resolution inputs obtained at pilot locations.

By applying this super-resolution technique, ChanEstiNet outputs more accurate channel estimates, achieving a low mean squared error (MSE) result in different testing conditions. This low MSE value indicates that ChanEstiNet can provide channel estimates that are very close to the perfect channel response, which is essential for the subsequent equalization processing step in FlexRx.

3.2.2 NoiseEstiNet

NoiseEstiNet is designed to estimate the noise variance from the received signal. Accurate noise variance estimation is important for the subsequent stages of equalization and demapping, serving as one of the inputs to the equalization and demapping networks. As seen in Figure 6, the architecture of NoiseEstiNet primarily consists of five convolutional layers, followed by a global average pooling layer and a repeat layer. This design allows the network to effectively extract and generalize noise characteristics from the input signal.

As for the activation function, NoiseEstiNet uses the Softplus activation function [12] instead of the more commonly used ReLU function. The Softplus function is defined as:

$$Softplus(x) = ln\,(1 + e^{x}) \tag{9}$$

By using Softplus, NoiseEstiNet ensures that even the smallest levels of noise are processed appropriately, avoiding any unrealistic zero or negative noise variance outputs.

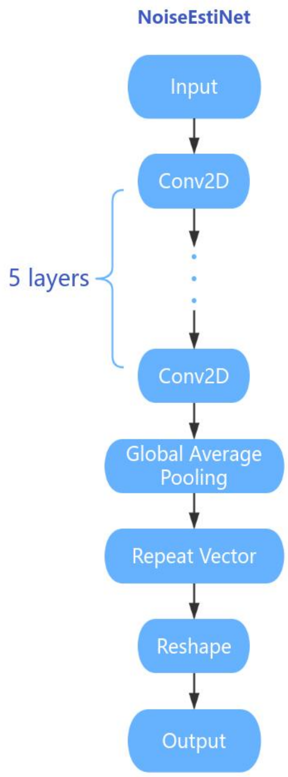

Figure. 6 NoiseEstiNet structure

The network further employs a global average pooling layer, which is used to average the features across the entire spatial dimensions of the output from the convolutional layers. This is particularly important in the context of noise variance estimation because, within a signal resource block, it is generally assumed that each signal point is affected by noise to the same extent. After the global average pooling, a repeat layer is utilized to replicate the noise variance across all signal points within a block to achieve the correct shape for the noise variance matrix. After undergoing global pooling and repetition, the estimated noise variance values become identical across all signal points.

It is also worth noting that NoiseEstiNet is an optional component in the FlexRx receiver architecture. If the ideal noise variance is known a priori, the network can be bypassed, allowing the receiver to operate with this perfect knowledge. This flexibility highlights the adaptability of the FlexRx receiver, making it suitable for various operating conditions where noise characteristics may or may not be precisely known.

3.2.3 EqualizeNet

EqualizeNet is the module responsible for signal equalization within the FlexRx OFDM receiver. Its primary function is to compensate for the distortions introduced by the wireless channel, thereby restoring the transmitted signal as accurately as possible. The inputs to EqualizeNet include the received signal, the estimated channel response, and the noise variance (which can be either an estimated or true value). The output of this network is the equalized signal, which will be subsequently used for demapping.

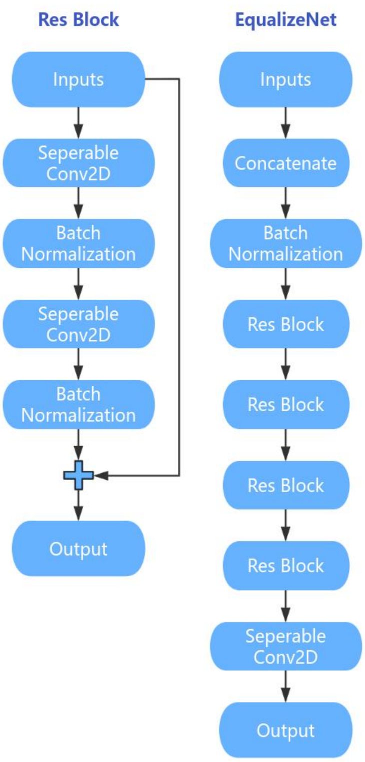

Figure. 7 EqualizeNet structure

From Figure 7, it can be observed that EqualizeNet is mainly composed of several residual blocks designed for effective feature extraction and signal recovery. Each residual block employs depthwise separable convolutional layers [7], a choice motivated by the specific nature of the equalization process. In equalization, the real and imaginary parts of the signal are treated independently, as they undergo separate equalization processes. There is no interdependency between these components across different channels. Luckily, depthwise separable convolutions are well-suited for this task as they allow the network to learn spatial features independently for each channel, enhancing its capacity to generalize while significantly reducing the number of trainable parameters.

In addition to employing the depthwise separable convolutional layers, each residual block in EqualizeNet also includes batch normalization layers. These layers standardize the inputs to each layer, which stabilizes the learning process and speeds up convergence during training. Batch normalization is especially important in deep networks, as it mitigates the internal covariate shift problem and allows for faster training and more robust performance. After processing by EqualizeNet, the signal is effectively equalized to correct for channel-induced distortions and noise effects, and this equalized signal is then passed on to the demapping stage, where it is converted back into the transmitted bits.

3.2.4 DemappingNet

DemappingNet serves as the final module in the FlexRx architecture, which is responsible for converting the equalized signal and noise variance into soft information of the encoded bits in the form of LLRs. The network structure of DemappingNet primarily consists of layer normalization layers [13] and multi-head convolutional layers [14], with Leaky ReLU [15] used as the activation function. These layers are carefully chosen to optimize the demapping process, thereby ensuring a more efficient and accurate conversion from signal space to bit space.

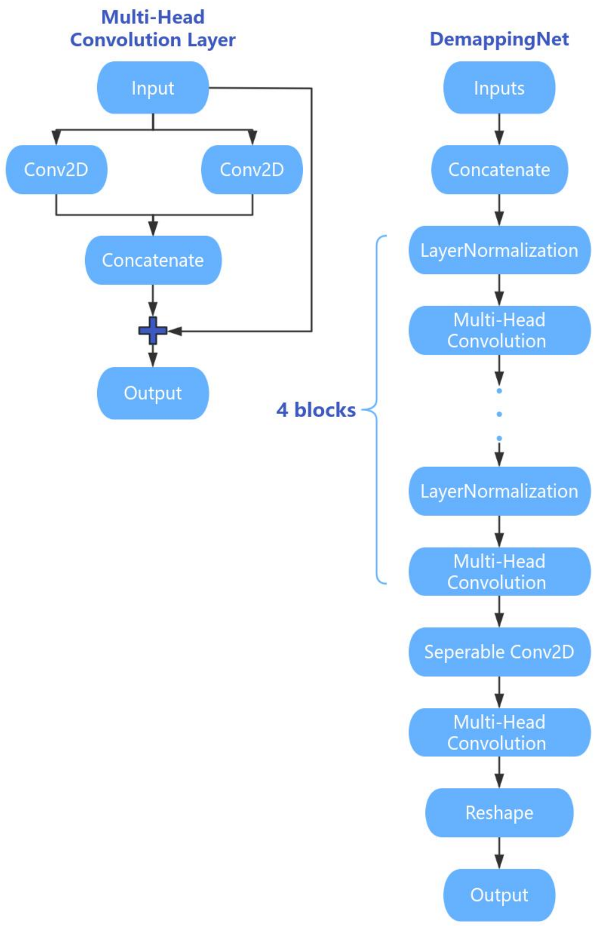

Figure. 8 DemappingNet structure

As shown in Figure 8, layer normalization stabilizes the training process by normalizing the inputs across the features for each data point, rather than across the batch. This normalization technique is particularly beneficial for demapping because it helps in mitigating the internal covariate shift, ensuring that the network learns effectively regardless of the input variations due to channel effects and noise.

Additionally, we utilize the self-attention technique by introducing multi-head convolutional layers into DemappingNet. Considering every OFDM signal point value is in complex form, the multi-head convolutional layer is designed to have two heads focusing separately on the real and imaginary parts of the signal, improving DemappingNet's ability to capture complex patterns. Moreover, the Leaky ReLU activation function further adds non-linearity to the network while allowing a small, non-zero gradient when the input is negative. This choice helps prevent dead neurons, which are neurons that output zero and stop learning, and encourages the network to learn more complex, non-linear relationships in the input data.

Finally, the LLRs of the encoded bits are obtained after processing through DemappingNet. These LLRs provide a measure of the confidence for each bit being a '0' or '1', which will be used by the decoder to reconstruct the transmitted information.

Chapter 4. Simulation Experiments under Varying Conditions

4.1 Experimental and Training Condition Setup

This section describes the setup of conditions where the simulation experiments were conducted. The experimental framework for this thesis was constructed on the foundation of the Channel Estimation Test Bed (CeBed) [17], a comprehensive suite of implementations and benchmarks specifically designed for OFDM channel estimation using TensorFlow. Building upon the fundamental OFDM wireless communication environment provided by CeBed, we further developed and integrated additional modules for coding, decoding, equalization, and demapping. The main training and testing methodologies were also redesigned to accommodate the modern OFDM system requirements.

To optimize the training process of the FlexRx receiver, an adaptive learning rate decay strategy was implemented using an exponential decay schedule. The learning rate started at an initial value of 0.001 and was reduced every 1,000 steps by a decay rate of 0.97, following a staircase decay pattern. Adam optimizer was used for all training stages. The training process was divided into three distinct phases:

1) Training ChanEstiNet and NoiseEstiNet: In the first phase, ChanEstiNet and NoiseEstiNet were trained using the MSE as the loss function. MSE is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{10}$$

In Equation (10), y_i represents the true values and $\hat{y}_i$ represents the predicted values. MSE measures the average squared difference between the estimated values and the actual values, minimizing the prediction error.

2) Training EqualizeNet: In the second phase, EqualizeNet was trained, also using MSE as the loss function. This phase focused on minimizing the error between the equalized signal produced by the network and the ideal equalized signal.

3) Training DemappingNet: In the final phase, DemappingNet was trained using Binary Cross Entropy (BCE) as the loss function. BCE is defined as:

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} [y_i \, log\,(\hat{y}_i) + (1 - y_i)\, log\,(1 - \hat{y}_i)] \tag{11}$$

where y_i represents the true binary labels and $\hat{y}_i$ represents the predicted probabilities. BCE measures the difference between the actual binary outcome and the predicted probability, making it particularly suitable for tasks that involve binary classification, such as demapping where the network predicts the likelihood of each bit being '0' or '1'.

All experiments were conducted on an NVIDIA GeForce RTX 3090 GPU, and after each training phase, the model weights were frozen at the point where the lowest validation loss was achieved.

4.2 Testing Condition Setup

To evaluate the effectiveness of FlexRx under various conditions, we conducted performance tests across different OFDM receivers using a range of transmission parameters and channel conditions. These tests aimed to assess the BER performance of FlexRx in comparison to other traditional or deep learning-based receivers, under diverse scenarios.

<table>
<tr><td rowspan="3">Receiver Configurations</td><td rowspan="3">FlexRx</td><td rowspan="3" colspan="2">Neural Receiver</td><td colspan="2">ChannelNet</td><td colspan="2">LS</td><td>LMMSE</td></tr>
<tr><td colspan="2">Zero Forcing</td><td colspan="2">Zero Forcing</td><td>Zero Forcing</td></tr>
<tr><td colspan="2">Max-Log</td><td colspan="2">Max-Log</td><td>Max-Log</td></tr>
<tr><td>Test Conditions</td><td colspan="2">Code Rate:
0.75 & 0.90</td><td colspan="2">Modulation Order:
16 QAM & 64 QAM</td><td colspan="2">Wireless Environment:
UMi & RMa</td><td colspan="2">User Speed:
static & 120 km/h</td></tr>
</table>

Table. 1 Testing condition setup

Table 1 shows the detailed testing conditions for different OFDM receivers. Among these receivers, FlexRx and Neural Receiver are complete OFDM receivers designed to directly recover soft bits from received signals using advanced neural network-based architectures. As for the remaining ChannelNet, LS, and LMMSE, they perform channel estimation using different techniques. These estimators are combined with traditional ZF equalizers and Max-Log demappers to form complete receivers.

The testing conditions were varied systematically to cover a comprehensive set of scenarios:

1) Code rate: Low-Density Parity-Check (LDPC) code is considered in our OFDM system due to its excellent error-correcting ability. Two different code rates, 0.75 and 0.90, were also tested to analyze the receivers' performance. In principle, a lower code rate, such as 0.75, introduces more redundancy and enhances error correction, while a higher code rate, like 0.90, has less redundancy, which may reduce error correction capability but increases data throughput.

2) Modulation order: The receivers were tested with two modulation orders, 16QAM and 64QAM. These modulation schemes represent different levels of spectral efficiency and robustness. In principle, 64QAM provides higher data rates but can be more susceptible to noise and interference compared to 16QAM.

3) Wireless environments: Two different wireless channels, urban microcell (UMi) and rural macrocell (RMa), are considered in this testing stage to examine the impact of different propagation conditions on receiver performance. Being affected by more complex conditions due to factors such as multipath fading, shadowing from buildings as well as other urban obstacles, the UMi channel is usually considered to be a “worse” wireless channel, making signal reception more challenging than in RMa environments.

4) User Speed: Both static and mobility scenarios are tested in this stage. Specifically, a high speed of 120 km/h is chosen to analyze the impact of Doppler shift on the communication system. The tests of all OFDM receivers are conducted in the RMa environment since 120 km/h reflects the typical speed of a vehicle on a highway, which is more common in rural macrocell areas than in urban microcell scenarios.

Chapter 5. Testing Results and Discussion

This chapter presents the performance testing results of various OFDM receivers under different conditions and provides a thorough analysis of these results. In addition, the flexibility and computational complexity of FlexRx are analyzed to demonstrate its adaptability and efficiency compared to other receivers.

5.1 Baseline BER Performance

In this section, we present the baseline BER performance comparison of different OFDM receivers without changing any test conditions. All experiments are conducted under the same standard setup with a code rate of 0.75, a modulation scheme of 16QAM, a static user scenario, and an UMi wireless channel.

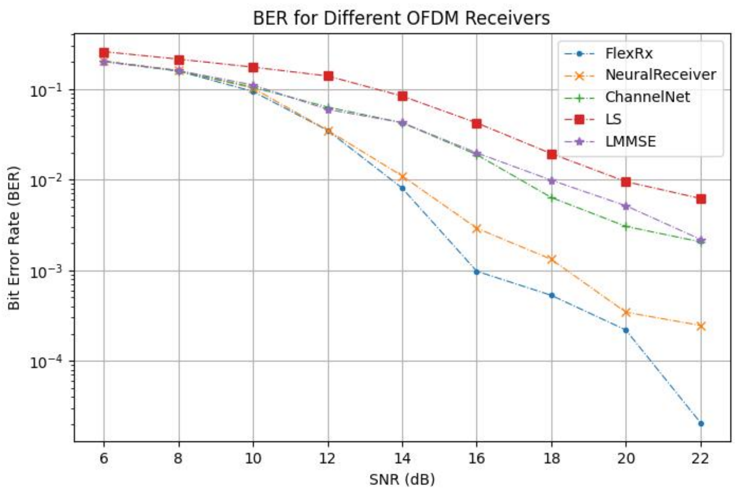

Figure. 9 Baseline BER performance

As shown in Figure 9, the BER versus Signal-to-Noise Ratio (SNR) curves indicate that FlexRx consistently outperforms the other receivers across all SNR levels. Specifically, FlexRx demonstrates the lowest BER at each tested SNR point, indicating superior error-correcting performance in this baseline scenario. Quantitatively, to achieve a target BER of 10^{-3}, FlexRx requires lower SNR than the other receivers. For instance, the next best-performing receiver, Neural Receiver, requires approximately 2.5 dB higher SNR to achieve the same BER level.

5.2 BER Performance under Different Code Rates

This section explores the BER performance of various OFDM receivers under different code rates. The experiments were conducted using the same settings as the baseline test, with 16QAM modulation, an UMi wireless channel, and a static user scenario, except for varying the code rate. The primary objective was to assess whether FlexRx maintains its effectiveness across different code rates and to evaluate how different code rates affect the BER performance of each receiver.

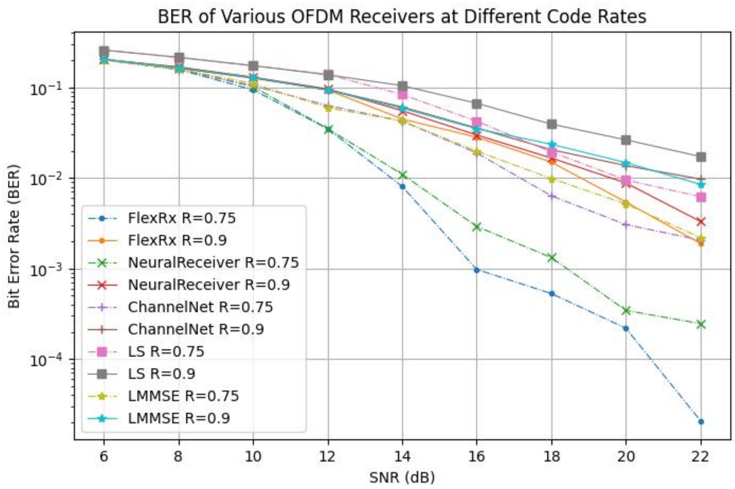

Figure. 10 BER performance at different code rates

As depicted in Figure 10, the BER performance was evaluated for each receiver at different code rates. The results show that FlexRx with a code rate of 0.75 exhibits the best performance, achieving the lowest BER across all SNR levels compared to the other receivers. When the code rate is increased to 0.9, all receivers' BER curves shift towards the upper right corner, indicating that a higher SNR is required to achieve the same BER. This shift implies that higher code rates, which have less redundancy for error correction, necessitate more transmission power to maintain the same level of reliability.

Quantitatively, to achieve a BER of 10^{-2}, FlexRx with a code rate of 0.9 requires approximately 5 dB more SNR than FlexRx with a code rate of 0.75. Additionally, comparing the same code rate of 0.9, Neural Receiver needs about 1 dB more SNR than FlexRx to achieve the same BER level. These results highlight the superior performance of FlexRx, especially under lower code rates.

5.3 BER Performance under Different Modulation Orders

This section explores the BER performance of various OFDM receivers under different modulation orders. The experiments were conducted by varying the modulation order from 16QAM to 64QAM while keeping the other settings the same as the baseline test, including a code rate of 0.75, an UMi wireless channel, and a static user scenario. The primary objective was to assess whether FlexRx maintains its effectiveness across different modulation orders and to evaluate how different modulation orders affect the BER performance of each receiver.

As illustrated in Figure 11 below, the BER performance was evaluated for each receiver using two modulation schemes: 16QAM and 64QAM. The results show clear differences in performance between the two modulation orders across the receivers.

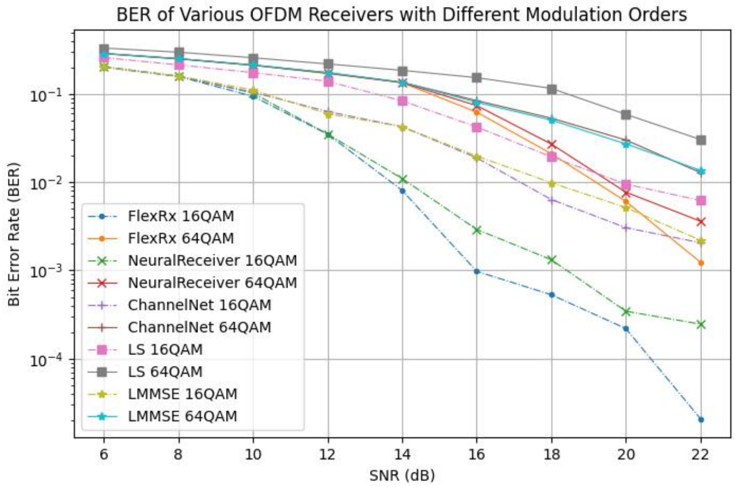

Figure. 11 BER performance at different modulation orders

The FlexRx OFDM receiver with 16QAM modulation exhibits the best performance, achieving the lowest BER across all SNR levels compared to the other receivers. As the modulation order increases to 64QAM, all receivers' BER curves shift toward the upper right, indicating that a higher SNR is required to achieve the same BER. This shift implies that higher modulation orders, which have more bits per symbol and are thus more spectrally efficient, require a stronger signal to maintain the same level of reliability due to increased susceptibility to noise and interference.

Quantitatively, to achieve a BER of 10^{-3}, FlexRx with 64QAM requires approximately 5.5 dB more SNR than FlexRx with 16QAM. Furthermore, within the same modulation order of 64QAM, Neural Receiver needs about 0.4 dB more SNR than FlexRx to achieve the same BER level. These results highlight the superior performance of FlexRx, especially under lower modulation orders.

5.3 BER Performance under Different Wireless Environments

This section explores the BER performance of various OFDM receivers under different wireless environments. The experiments were conducted by varying the wireless channel conditions while keeping the other settings the same as the baseline test, including a code rate of 0.75, 16QAM modulation, and a static user scenario.

Figure 12 shows the BER performance of each receiver tested under two types of wireless channels: UMi and RMa. From Figure 12, it can be observed that FlexRx operating in the RMa channel exhibits the best BER performance, achieving the lowest BER across all SNR levels compared to other receivers. As the wireless environment shifts from UMi to RMa, there is a noticeable improvement in BER performance for all receivers, indicated by the BER curves moving toward the lower left, suggesting that better channel conditions require less SNR to achieve the same BER.

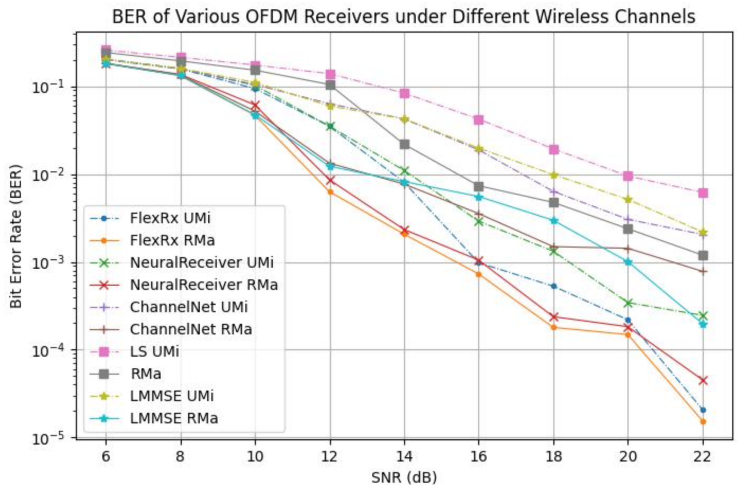

Figure. 12 BER performance under different wireless channels

Quantitatively, to achieve a BER of 10^{-3}, FlexRx in the UMi channel requires approximately 0.8 dB more SNR than FlexRx in the RMa channel. Additionally, under RMa conditions, Neural Receiver needs about 0.8 dB more SNR than FlexRx to achieve the same BER level. These results demonstrate the superior performance of FlexRx in different wireless channels.

5.4 BER Performance under Different User Speeds

This section explores the BER performance of various OFDM receivers under different user speeds. The experiments were conducted by varying the user speed while keeping the other settings the same, including a code rate of 0.75, 16QAM modulation, and a RMa wireless channel.

Figure 13 shows the BER performance of each receiver tested under two user speed conditions: static (0 km/h) and high speed (120 km/h). The results show significant variations in performance due to changes in user speed.

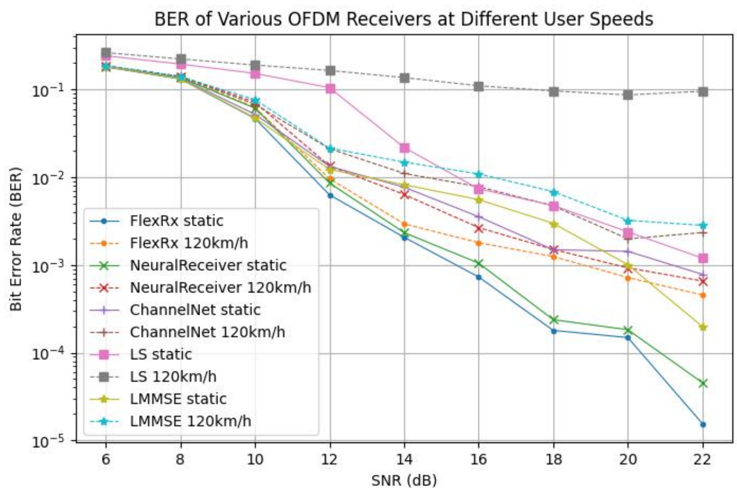

Figure. 13 BER performance at different user speeds

The FlexRx receiver under static conditions exhibits the best BER performance, achieving the lowest BER across all SNR levels compared to other receivers. From Figure 13, it can be observed that the Doppler effect becomes more pronounced, causing all receivers' BER curves to shift towards the upper right as user speed increases. This shift indicates that a higher SNR is required to achieve the same BER, as the increased speed introduces more Doppler shifts, challenging the receivers' ability to maintain signal integrity.

Quantitatively, to achieve a BER of 10^{-3}, FlexRx at 120 km/h requires approximately 3.5 dB more SNR than FlexRx under static conditions. Additionally, at 120 km/h, Neural Receiver requires about 1 dB more SNR than FlexRx to reach the same BER level. These results underscore the superior signal reception performance of FlexRx in both static and mobility scenarios.

5.5 Flexibility Analysis

This section analyzes the flexibility and complexity of the FlexRx OFDM receiver by evaluating its modular design and the resulting performance across different configurations. Due to its modular architecture, our proposed FlexRx allows various combinations of its modules to create distinct receiver setups, each suited for specific conditions or requirements. Here we evaluated four configurations of FlexRx:

1) FlexRx with estimated noise: utilizes all FlexRx modules, including ChanEstiNet, NoiseEstiNet, EqualizeNet, and DemappingNet.

2) FlexRx with perfect noise: utilizes ChanEstiNet, EqualizeNet, and DemappingNet, assuming perfect knowledge of the noise variance, bypassing the need for NoiseEstiNet.

3) FlexRx with equalized signal: utilizes ChanEstiNet, NoiseEstiNet, and EqualizeNet while incorporating the traditional Max-Log demapping method instead of DemappingNet.

4) FlexRx with channel response: combines ChanEstiNet with conventional ZF equalization and Max-Log demapping instead of NoiseEstiNet and DemappingNet.

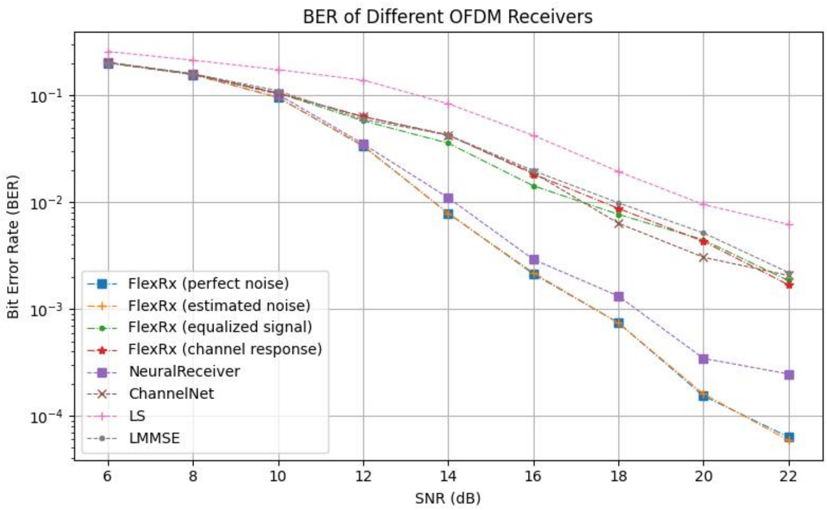

Figure. 14 BER performance with different FlexRx configurations

Figure 14 shows the BER-SNR curves of these four FlexRx configurations alongside other OFDM receivers. The results show that FlexRx, whether using estimated or perfect noise variance, achieves the lowest BER. However, this comes at the cost of increased training time, particularly for configurations utilizing more network modules. When using predicted equalized signals or channel responses, FlexRx shows a slight increase in BER but remains within acceptable limits, demonstrating its adaptability.

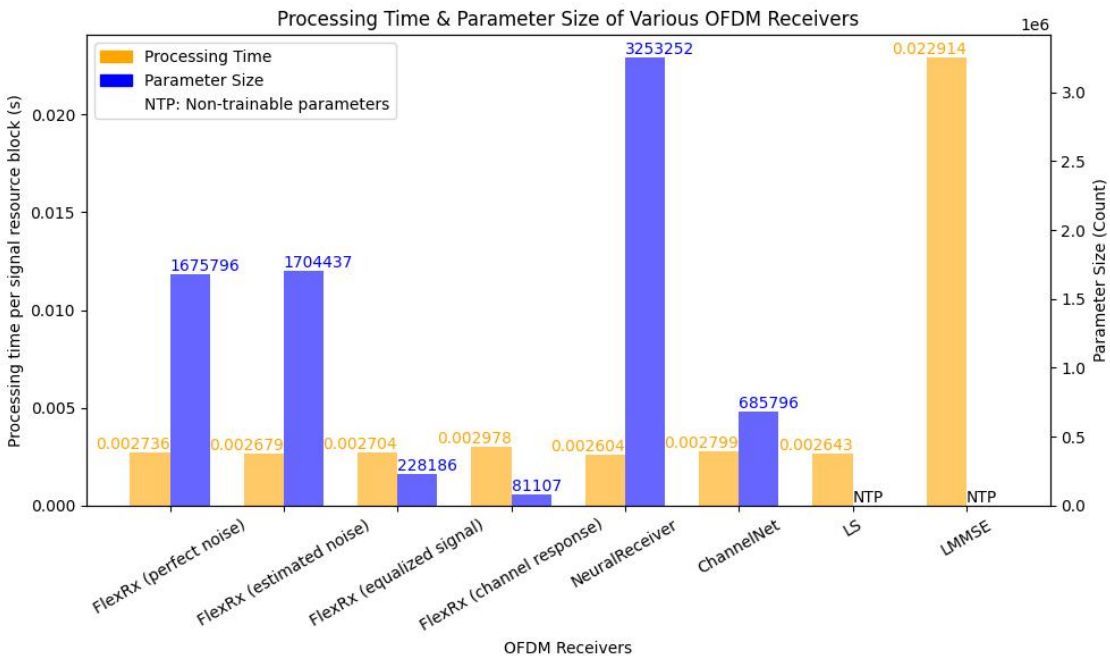

Figure. 15 Processing time and parameter size of various OFDM receivers

Figure 15 provides a comparative analysis of parameter size and processing time for different receiver configurations. The Neural Receiver, with its performance achieved primarily through extensive parameter stacking rather than a principled design, exhibits the largest parameter size. FlexRx configurations generally have moderate parameter sizes, with fewer modules leading to smaller sizes. In terms of processing time, LMMSE exhibits the longest due to its high algorithmic complexity, while Neural Receiver processes signals faster than FlexRx configurations. This is attributed to the time consumed by data transfer between FlexRx modules, although this remains within acceptable limits.

The flexibility of FlexRx lies in its ability to freely combine its modules with traditional algorithms, allowing users to tailor the receiver to specific performance and complexity requirements. Using more network modules typically improves BER performance but also increases training time. The complexity analysis shows that FlexRx achieves a balance between minimal parameter size and fast processing time. Higher network usage results in greater memory requirements (higher spatial complexity) due to more parameters, but since all computations are handled internally without involving traditional algorithms, the temporal complexity is often reduced. This balance between spatial and temporal complexity makes FlexRx a versatile and efficient choice for varying operational environments.

Chapter 6. Conclusion and Future Work

6.1 Conclusion

In this thesis, we present the development of FlexRx, an end-to-end multi-module deep learning-based OFDM receiver designed to effectively recover soft bits from received signals. By integrating several specialized neural network modules, specifically ChanEstiNet, NoiseEstiNet, EqualizeNet, and DemappingNet, our proposed FlexRx can handle various stages of the signal reception chain, from channel estimation to demapping, ensuring a robust and accurate data recovery process. Several performance tests are then conducted and the results show FlexRx's excellent BER performance compared to traditional and other neural network-based receivers across a range of operating conditions, including different code rates, modulation orders, wireless environments, and user speeds. Furthermore, since its modular design allows for different configurations, FlexRx offers high flexibility with relatively low complexity. This adaptability enables efficient and versatile processing, catering to various scenarios and requirements. Overall, FlexRx represents a significant advancement in OFDM receiver design, combining the strengths of deep learning with practical considerations of flexibility and computational efficiency.

6.2 Future Work

This section presents some future work that can be done to improve our work. Firstly, the performance testing method can be refined. In this thesis, BER performance was evaluated using a limited number of bits (e.g., one million) due to time constraints, which resulted in less smooth BER curves. Future work could involve increasing the number of test bits or conducting simultaneous testing over extended periods, such as a day or a week, to achieve more accurate and reliable results across all receivers. Moreover, extending the FlexRx architecture to support MIMO systems is a crucial next step. This would involve adapting the receiver to handle the complexities of multiple antennas, making it much more applicable to real-world scenarios.

Reference

[1] M. B. Sutar and V. S. Patil, "LS and MMSE estimation with different fading channels for OFDM system," in 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2017, pp. 740–745, doi: 10.1109/ICECA.2017.8203641.

[2] A. Khlifi and R. Bouallegue, "Performance Analysis of LS and LMMSE Channel Estimation Techniques for LTE Downlink Systems," International Journal of Wireless & Mobile Networks, vol. 3, no. 5, pp. 141–149, Oct. 2011, doi: 10.5121/ijwmn.2011.3511.

[3] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," arXiv, 2019. [Online]. Available: https://arxiv.org/abs/1810.05893.

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," arXiv, 2015. [Online]. Available: https://arxiv.org/abs/1501.00092.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, Jul. 2017, doi: 10.1109/TIP.2017.2662206.

[6] J. Hoydis, S. Cammerer, F. Ait Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, "Sionna: An Open-Source Library for Next-Generation Physical Layer Research," arXiv preprint, Mar. 2022. [Online]. Available: https://arxiv.org/abs/2203.11854.

[7] F. Chollet, "Xception: Deep Learning With Depthwise Separable Convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017.

[8] M. Honkala, D. Korpi, and J. M. J. Huttunen, "DeepRx: Fully Convolutional Deep Learning Receiver," IEEE Transactions on Wireless Communications, vol. 20, no. 6, pp. 3925-3940, Jun. 2021, doi: 10.1109/TWC.2021.3054520.

[9] R. Gallager, "Low-density parity-check codes," IRE Transactions on Information Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962, doi: 10.1109/TIT.1962.1057683.

[10] ETSI TR 138 901 V14.3.0 (2018-01), "5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 14.3.0 Release 14)," Technical Report, Sophia Antipolis, France: ETSI, Jan. 2018. [Online]. Available:https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/14.03.00_60/tr_138901v140300p.pdf

[11] A. F. Molisch, "Orthogonal Frequency Division Multiplexing (OFDM)," in Wireless Communications, IEEE, 2011, pp. 417–443, doi: 10.1002/9781119992806.ch19.

[12] D. C. Marcu and C. Grava, "The impact of activation functions on training and performance of a deep neural network," in Proceedings of the 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, Romania, 2021, pp. 1–4, doi: 10.1109/EMES52337.2021.9484108.

[13] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," arXiv, 2016. [Online]. Available: https://arxiv.org/abs/1607.06450.

[14] Y. Zhang, B. Xu, and T. Zhao, "Convolutional multi-head self-attention on memory for aspect sentiment classification," IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 4, pp. 1038–1044, Jul. 2020, doi: 10.1109/JAS.2020.1003243.

[15] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, "Reluplex made more practical: Leaky ReLU," in Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 2020, pp. 1–7, doi: 10.1109/ISCC50000.2020.9219587.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385.

[17] A. Feriani, D. Wu, S. Liu, and G. Dudek, "CeBed: A Benchmark for Deep Data-Driven OFDM Channel Estimation," 2023. [Online]. Available: https://github.com/SAIC-MONTREAL/cebed.git.

Supplement

All the experiments in this thesis were conducted under the CeBed framework, as described by Feriani et al (2023). The original code can be found at: https://github.com/SAIC-MONTREAL/CeBed.

The following modifications and implementations were made by the author:

cebed/datasets/sionna.py line 204-223, line 303-496

cebed/models/mine.py

cebed/models/NR.py

cebed/envs.py line 197-215, line 291-354

cebed/evaluation.py line 15-67

cebed/trainer.py line 20-162, line 171-803

hyperparams/MyNet.yaml

hyperparams/NeuralReceiver.yaml

PerformanceTest/Compexity.py

PerformanceTest/MyNet_test.py

PerformanceTest/MyNetAll_test.py

PerformanceTest/NeuralReceiver_test.py

PerformanceTest/Othernet_test.py

PerformanceTest/Replot_MyNetall.py

PerformanceTest/Replot.py

PerformanceTest/Traditional_baseline_test.py

scripts/generate_datasets_from_sionna.py line 20-72

scripts/train.py line 14-35