Distributed Massive MIMO with Low-Resolution ADCs: Enhancing Efficiency through Deep Learning

Elina Amani

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Distributed Massive MIMO with Low-Resolution ADCs:
Enhancing Efficiency through Deep Learning

Elina Amani

Department of Electrical Engineering
Division of Communication Systems
Chalmers University of Technology
Gothenburg, Sweden 2024

Distributed Massive MIMO with Low-Resolution ADCs: Enhancing Efficiency through Deep Learning
ELINA AMANI

Supervisors: Giuseppe Durisi, Department of Electrical Engineering
Alireza Bordbar, Department of Electrical Engineering
Examiner: Giuseppe Durisi, Department of Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Division of Communication Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Distributed Massive MIMO with Low-Resolution ADCs: Enhancing Efficiency through Deep Learning
ELINA AMANI
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Distributed massive MIMO, including a central unit (CU) and a large number of spatially distributed antennas, provides more uniform quality of service (QoS) than co-located massive MIMO systems. One of the components used in distributed massive MIMO is the analog-to-digital converters (ADC). However, high-resolution ADCs consume a considerable power.

Having a simple structure and a very low power consumption, the low-resolution ADCs, can be used to decrease both the complexity and power consumption. However, using such ADCs, introduces non-linear distortions in the received signals, thus, complicating channel estimation and data detection at the receiver.

In this study, a distributed massive MIMO case with one-bit radio-over-fiber fronthaul has been studied where model-driven deep learning structures are used to compensate for the non-linear distortion caused by the low-resolution ADCs used in the communication system. The aim is to improve both channel estimation and data detection in the uplink phase.

Keywords: Distributed Massive MIMO, one-bit ADC, Deep Neural Network, Channel Estimation, Data Detection

Acknowledgements

This thesis was conducted in the Electrical Engineering Department at Chalmers University of Technology.

First, I would like to express my sincere appreciation to my examiner, Giuseppe Durisi, for proposing this engaging topic, and to my supervisor, Alireza Bordbar, for his patient guidance throughout this thesis. Additionally, I wish to acknowledge my husband for his steadfast support and encouragement during this process.

I would also like to acknowledge the computational and data handling resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS).

Finally, I acknowledge the use of OpenAI's language model, ChatGPT, for its assistance with grammar checking and enhancing the clarity of this report.

Elina Amani, Gothenburg, August 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
BER	Bit Error Rate
BPF	Band-Pass Filter
BS	Base Station
BLMMSE	Bussgang Linear Minimum Mean Square Error
CDF	Comulative Distributed Function
CSI	Channel State Information
CU	Central Unit
DAC	Digital-to-Analog Converter
OSR	Oversampling Ratio
E/O	Electrical-to-Optical
FBM-CENet	Few-Bit Massive MIMO Channel Estimation Network
FBM-DetNet	Few-Bit Massive MIMO Data Detection Network
LPF	Low-Pass Filter
MIMO	Multiple-Input, Multiple-Output
ML	Maximum Liklihood
MSE	Mean Squared Error
NMSE	Normalized Mean Squared Error
O/E	Optical-to-Electrical
PA	Power Amplifier
PB	Pass-Band
PDF	Probability Density Function
QoS	Quality of Service
RF	Radio Frequency
RRH	Remote Radio Head
SNR	Signal-to-Noise Ratio

Contents

List of Figures

1

Introduction

1.1 Background

One of the most significant challenges in distributed massive MIMO architecture is ensuring phase coherence across all Remote Radio Heads (RRHs) to maintain optimal performance.[1] In conventional distributed massive MIMO systems, the baseband signal is transmitted between the Central Unit (CU) and RRHs through fronthaul links, with upconversion and downconversion occurring at the RRHs. These systems achieve phase coherence by employing synchronized oscillators at the RRHs to facilitate upconversion and downconversion.[2] However, implementing this method, especially at high carrier frequencies, poses significant challenges due to increased phase noise and frequency stability issues.

Additionally, transferring analog signals over fronthaul optical fiber links is challenging due to the high linearity requirements. To address this issue, low-resolution analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) can be used to convert analog signals to low-resolution digital signals before sending them to the central unit (CU) through optical fronthaul links.[3] However, this process results in the signal received at the CU being non-linearly distorted. One way to improve the non-linear output of the low-resolution ADCs is to use a dither signal.[4]

In [5], an innovative distributed massive MIMO system has been studied through simulations and measurements on a test bed implemented at Chalmers University. The proposed transceiver architecture includes RRHs connected to the CU via optical fibers. To address the challenge of phase coherence, a direct RF sampling architecture is employed, where upconversion and downconversion are performed in the digital domain at the CU. Subsequently, the 1-bit quantized signal is transmitted from the CU to the RRHs via digital fronthaul optical fiber links.

In downlink mode, the sigma-delta modulator generates an oversampled 1-bit signal which then goes into a 1-bit DAC, from which it is led into an electrical-to-optical (E/O) converter. At the RRH, the radio-frequency (RF) signal is recovered from the 1-bit signal through a band-pass filter and a power amplifier, and then fed to an antenna.

In uplink mode, after the RF signal received by the antenna passes through a band-pass filter (BPF) and an automatic gain control (AGC), the RRH provides a dither

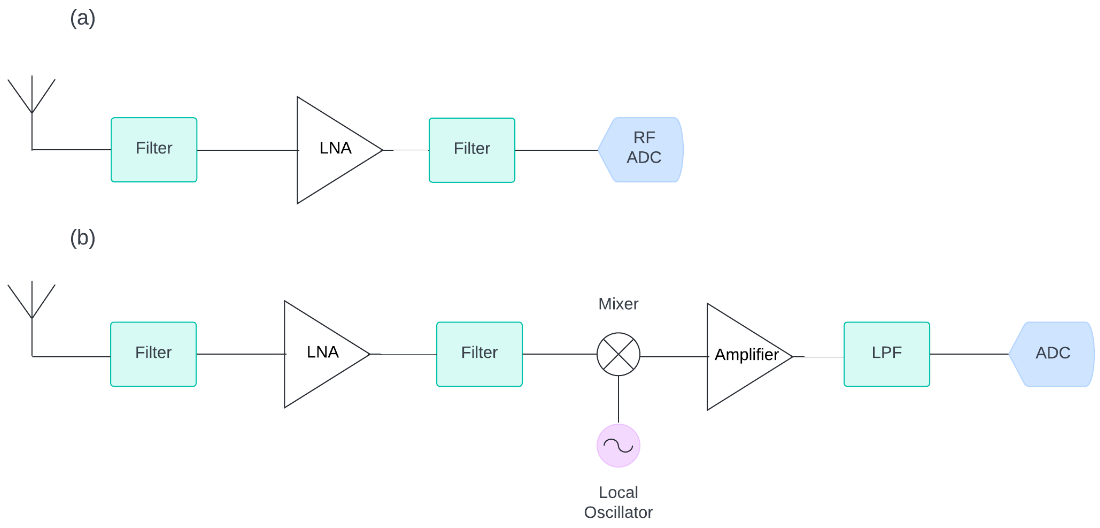

Figure 1.1: Sampling receiver systems from [6]: (a) Direct RF sampling, (b) Baseband sampling

signal to the comparator. This dither signal is generated at the CU and executed by the RRH using a low-pass filter (LPF). The output of the comparator is led to an optical-to-electrical (O/E) converter and sent to the CU through the optical fiber. In the CU, after 1-bit analog-to-digital conversion, digital signal processing (DSP) is applied to the converted signal.

1.2 Previous studies

In one-bit ADCs, only the sign of the received signals is retained. This causes severe non-linearity and, as a result, significant distortions in the received signals [7]. To address this problem, several studies have been conducted. For example, a one-bit maximum-likelihood (ML) detector was studied in [8]. However, because this method is not practical for large-scale systems, a near-ML data detection method was also proposed. These methods, however, require full prior knowledge of the channel state information to perform well at higher signal-to-noise ratios (SNRs). In another study, generalized approximate message passing (GAMP) and Bayesian inference methods were explored, but these methods are complex and expensive to implement.[9] Additionally, in [10], Bussgang decomposition was utilized to derive a one-bit Bussgang-based minimum mean-squared error (BMMSE) channel estimator.

Deep learning and machine learning are fast-growing tools in the field of telecommunication, with numerous applications such as channel estimation and data detection. In [11], support vector machine models were used to estimate massive MIMO channels and for data detection in systems with one-bit ADCs. A deep neural network (DNN) was proposed in [12] for data detection in systems with one-bit ADCs. However, the proposed network needs to be retrained every time the channel changes, which imposes a heavy computational load and consumes a significant amount of energy. Additionally, this results in lower spectral efficiency due to the increased pilot overhead.

1.3 Purpose

In [13], a few-bit massive MIMO case has been studied where deep learning has been used to compensate for the non-linear distortion caused by the low-resolution ADCs used in the communication system. In this study, deep unfolding of the first-order optimization iterations is used to derive channel estimation and data detection networks for one-bit and few-bit ADCs. The simulation results from this paper evaluating both the bit error rate (BER) for data detection, and normalized mean squared error (NMSE) for channel estimation in terms of signal-to-noise ratio (SNR), show that the proposed networks can successfully address this problem. Considering the results from [13], and the rapid growth of the usage of deep learning in a vast variety of fields, this master's thesis aims to design and implement a suitable deep learning network for the distributed massive MIMO system discussed in [5].

1.4 Thesis Outline

This thesis is structured as follows: In Chapter 2, the proposed distributed massive MIMO system in [5] is presented for both the uplink and downlink phases. Additionally, the derived deep unfolding networks introduced in [13] for channel estimation and data detection in a simple massive MIMO system are described in detail. In Chapter 3, the theory introduced in [13] is utilized to implement deep unfolding networks, which are applied to the received signals in the proposed system outlined in [5]. This approach aims to achieve superior performance in channel estimation and data detection. The results of these implementations are presented in Chapter 4, followed by the study's conclusions in Chapter 5.

2

Theory

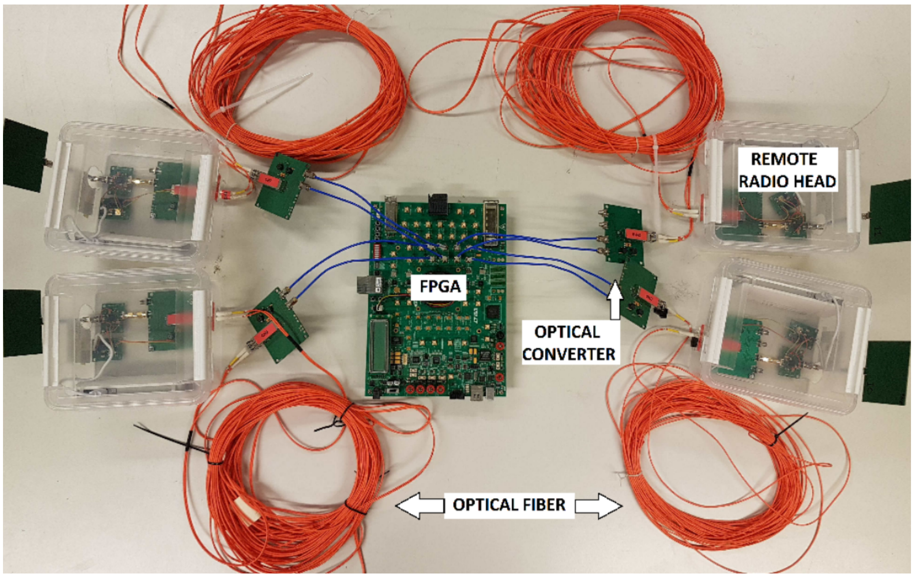

Figure 2.1: Downlink testbed hardware. Image from [5].

This chapter presents the theoretical background of the distributed massive MIMO test-bed, shown in Figure 2.1, which was presented by [5], along with two deep neural network (DNN) models presented by [13]: Few-Bit massive MIMO Channel Estimation Network (FBM-CENet) for channel estimation, and Few-Bit Massive MIMO Data Detection Network (FBM-DetNet) for data detection in a Massive MIMO system. Throughout this thesis, uppercase and lowercase boldface letters denote matrices and column vectors, respectively.

2.1 Distributed Massive MIMO with One-Bit Radio Over Fiber Fronthaul

2.1.1 Architecture

The proposed transceiver architecture, as illustrated in Figure 2.2, depicts the system in a scenario where the CU is connected to a single RRH, although the system can currently support up to three RRHs. In the downlink mode (depicted by blue traces and blocks), the digital pass-band (PB) signal undergoes sigma-delta modulation, resulting in an oversampled 1-bit signal. Following digital-to-analog conversion using a 1-bit DAC, the signal is converted to the optical domain via electrical-to-optical (E/O) conversion. At the RRH, the signal is converted back to the electrical domain, then passes through a BPF and a power amplifier (PA) before reaching the antenna.

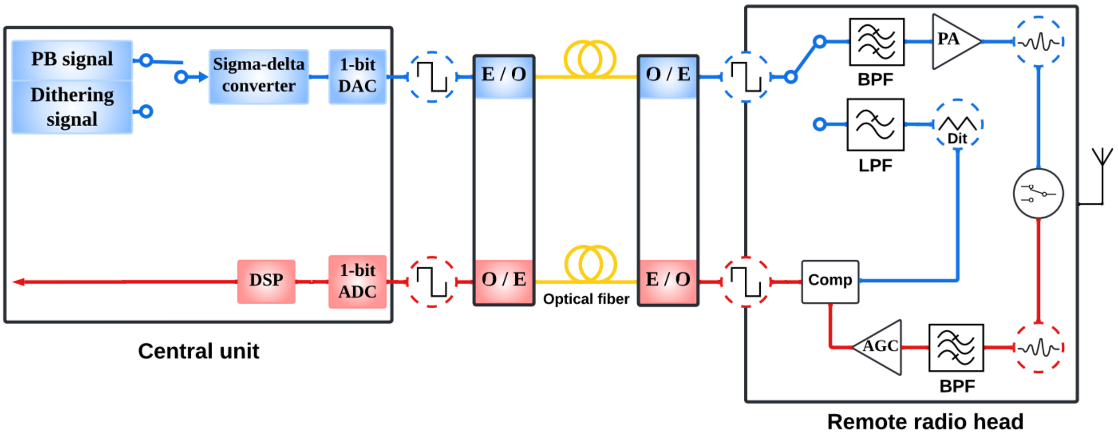

Figure 2.2: The transceiver architecture proposed in [5].

During uplink transmission (illustrated by red traces and blocks), the RF signal received by the antenna undergoes amplification by a low-noise amplifier (LNA), followed by filtering with a BPF, and adjustment of signal gain through AGC. Subsequently, a comparator (Comp) further processes the signal. In uplink mode, the transmitter introduces a dither signal (Dit) to the comparator at the RRH. This dither signal, generated at the CU, passes through an LPF at the RRH before being input to the comparator. The comparator's output is converted to the optical domain and transmitted to the CU via optical fiber. After 1-bit ADC, the signal is routed to the DSP unit for further processing.

The subsequent sections provide detailed explanations for each aspect of the transceiver architecture described above.

2.1.1.1 Downlink

Quantization introduces non-linear distortion to the signal, complicating its recovery at the RRH. Therefore, sigma-delta modulation is employed at the CU to produce a binary output while minimizing the distortion caused by quantization. The oversampling ratio (OSR), and the order of the sigma-delta modulator are two key factors that determine the noise shaping properties of these modulators. The OSR is defined as (2.1).

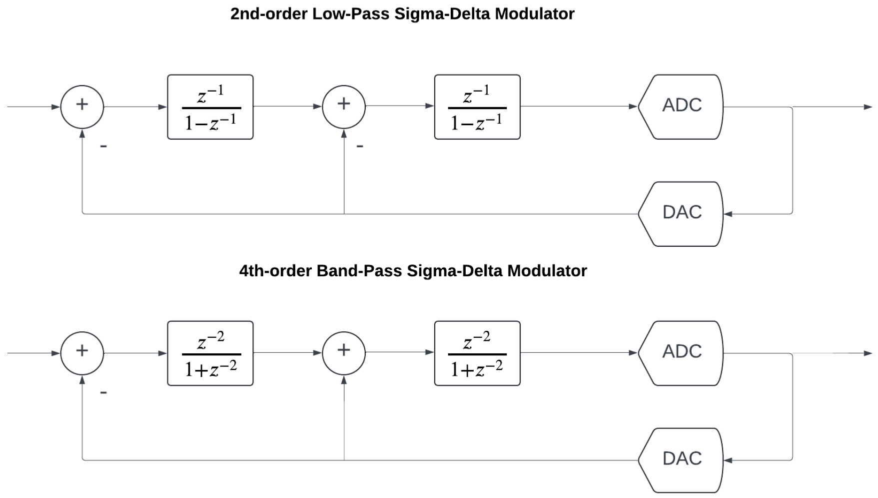

Figure 2.3: LP-to-BP $z \rightarrow -z^2$ transformation. Image from [14].

$$OSR = \frac{f_s}{2B} \tag{2.1}$$

where f_s is the sampling frequency and B denotes the bandwidth of the input signal.

Sigma-delta modulation employs feedback loops to introduce distortion error to the signal before quantization, thereby mitigating the non-linear effects caused by quantization. This process results in the shifting of quantization noise frequencies to higher frequencies in the frequency domain, reducing their presence within the bandwidth of interest.

Furthermore, the order of the sigma-delta modulator determines the number of feedback loops used to compensate for quantization noise, thereby reducing the non-linearity induced by the quantization process. In other words, as the order of the sigma-delta converter increases, the noise produced during sigma-delta conversion is effectively pushed farther away from the desired frequency, which makes it easier to convert the signal back from digital to analog using a pass-band filter.

The model employs a fourth-order band-pass sigma-delta converter in the downlink to convert a pass-band digital signal into a high-resolution one-bit digital signal. This one-bit signal is then converted into an optical signal and transmitted through an optical fiber.

Despite the sigma-delta modulator outputting a single-bit stream, the high oversampling ratio and noise shaping help to push quantization noise out of the band of interest, achieving high resolution during the analog-to-digital conversion process and minimizing distortion during subsequent digital-to-analog conversion.

Figure 2.3 from [14], shows a 4^{th}-order band-pass and a 2^{nd}-order low-pass sigma-delta modulator loop in the Z domain. Considering these two modulators, we can see that by applying the transformation $z^{-1} \rightarrow -z^{-2}$ to a 2^{nd}-order low-pass sigma-delta modulator, we can achieve a 4^{th}-order band-pass sigma-delta modulator with a frequency centered at one quarter of the sampling frequency ($f_s/4$).

2.1.1.2 Uplink

In the uplink, a comparator within the remote radio head is utilized to convert the received signal into a 1-bit quantized signal, which is then transmitted to the central unit via an optical fiber. In order to reduce the quantization error, a dither signal is added to the received signal. The dither signal is employed to introduce randomness into the signal, thereby decorrelating the quantization error from the received signal. This addition ensures a more uniform distribution of quantization errors.

As the dither signal is generated at the CU, its frequency and power can be designed to achieve maximum enhancement in reducing quantization error. Therefore, an AGC should be employed in the uplink at the RRH to scale up the received signal to a power level that allows for the optimal design of the dither signal with respect to it.

Finally, the received signal at the CU after ADC is as shown in (2.2) [15]

$$\boldsymbol{z}_n^{RF} = \operatorname{sgn}(\boldsymbol{y}_n^{RF} + \boldsymbol{d}_n), \qquad n = 0, 1, 2, ..., N-1 \tag{2.2}$$

where $\boldsymbol{d}_n \sim \mathcal{N}(0_B, \frac{E_d}{2}\boldsymbol{I}_B)$ is the dither signal, which is independent of $\boldsymbol{y}_n^{RF}$, and $\boldsymbol{y}_n^{RF} \in \mathbb{R}^{B\times T}$ is the n-th sample of the received signal as defined in (2.3), after band-pass (BP) filtering, and before 1-bit quantization, where B is the number of RRHs and T is the length of the signal for each user and E_d is the power of the received signal. We can express $\boldsymbol{y}_n^{RF}$ as

$$\boldsymbol{y}_n^{RF} = \sqrt{2}\Re\{\boldsymbol{y}_n^{BB} e^{j2\pi\frac{f_c}{f_s}n}\}. \tag{2.3}$$

Here, f_c is the carrier frequency, f_s is the sampling frequency, and $\boldsymbol{y}_n^{BB} \in \mathbb{C}^{B\times T}$ is the base-band oversampled received signal, which can be modeled as

$$\boldsymbol{y}_n^{BB} = \frac{1}{\sqrt{N}} \sum_{k\in\mathcal{S}} (\boldsymbol{H}_k \boldsymbol{s}_k + \boldsymbol{\omega}_k) e^{j2\pi\frac{k}{N}n} \tag{2.4}$$

where $\boldsymbol{s}_k$ is the signal expressed in frequency domain, $\boldsymbol{H}_k \in \mathbb{C}^{B\times U}$ is the channel frequency response, where U is the number of users, and $\boldsymbol{\omega}_k \sim \mathcal{CN}(0_B, N_0\boldsymbol{I}_B)$ is the additive Gaussian noise at APs, where N_0 where N_0 represents the variance of each entry of the noise vector $\boldsymbol{\omega}$.

In (2.4), N represents the number of samples after upsampling, and k belongs to the set $\mathcal{S}$ defined as

$$\mathcal{S} = \{0, 1, ..., (S-1)/2, N-(S-1)/2+1, ..., N-1\} \tag{2.5}$$

which is a symmetric set for sampling. Utilizing symmetric sampling is essential for accurately representing the frequency spectrum of the signal and preventing aliasing.

In 2.5, $S = WT$, where W is the bandwidth, and T is the time interval in which the received signal is considered for further processing.

It is also worth noting that the term $e^{j2\pi \frac{k}{N} n}$ in (2.4) serves as the inverse Fourier transform factor, facilitating the conversion of the received signal to the time domain during the process of upsampling.

2.2 Deep Learning for Channel Estimation and Data Detection in 1-Bit Massive MIMO

2.2.1 Introduction

The quantization distortion caused by one-bit DAC and ADC makes channel estimation and data detection challenging. However, deep neural networks can learn this non-linear behavior and adapt their weights to achieve optimal channel estimation and data detection using gradient ascent or descent.

In this section, the massive MIMO system proposed in [13] is considered, and two different deep unfolding networks are employed for channel estimation and data detection in the uplink scenario. These results are compared to Bussgang-based channel estimation and data detection to evaluate the performance of the networks.

2.2.2 System Model

In this section, a massive MIMO system is considered, comprising U single-antenna users, each transmitting a message of length T, and a base station (BS) with B antennas. The received analog signal $\boldsymbol{Y}$, prior to quantization, is expressed as

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{X} + \boldsymbol{W}. \tag{2.6}$$

Here, $\boldsymbol{X} \in \mathbb{C}^{U \times T}$ is the complex transmitted matrix, $\boldsymbol{H} \in \mathbb{C}^{B \times U}$ is the channel matrix, and $\boldsymbol{W} \in \mathbb{C}^{B \times T}$ is the noise matrix which has independent and identically distributed (i.i.d.) entries drawn from the complex Gaussian distribution $\mathcal{CN}(0, N_0)$.

After passing through the 1-bit ADC, the received digital signal $\boldsymbol{Z}$ given by

$$\boldsymbol{Z} = \operatorname{sgn}(\Re\{\boldsymbol{Y}\}) + \operatorname{sgn}(j\Im\{\boldsymbol{Y}\}). \tag{2.7}$$

2.2.3 Channel Estimation

In order to perform channel estimation, a pilot sequence $\boldsymbol{X_t} \in \mathbb{C}^{U \times T}$ is generated for use in the channel estimation process. Here, U represents the number of users, T indicates the length of the pilot sequence transmitted by each user, and the subscript t denotes the training phase.

Now, (2.6) and (2.7) can be rewritten as

$$\boldsymbol{Z_t} = \operatorname{sgn}(\Re\{\boldsymbol{Y_t}\}) + \operatorname{sgn}(j\Im\{\boldsymbol{Y_t}\}) \tag{2.8}$$

$$s.t.\ \boldsymbol{Y_t} = \boldsymbol{H}\boldsymbol{X_t} + \boldsymbol{W_t} \tag{2.9}$$

where $\boldsymbol{Z_t} \in \mathbb{C}^{B \times T}$, and $\boldsymbol{W_t} \in \mathbb{C}^{B \times T}$.

To simplify processing and ensure data compatibility with the neural network, it is recommended to vectorize the received signal and split all vectors and matrices in (2.8) and (2.9) into their real and imaginary parts. This enables handling real values during the training process. Consequently, the received signal is transformed into

$$\boldsymbol{z_t} = \operatorname{sgn}(\boldsymbol{P}\boldsymbol{h} + \boldsymbol{\omega_t}) \tag{2.10}$$

where

$\boldsymbol{z_t} = \begin{bmatrix} \Re\{vec(\boldsymbol{Z_t})\} \\ \Im\{vec(\boldsymbol{Z_t})\} \end{bmatrix}$, $\boldsymbol{h} = \begin{bmatrix} \Re\{vec(\boldsymbol{H})\} \\ \Im\{vec(\boldsymbol{H})\} \end{bmatrix}$, and $\boldsymbol{P} = \begin{bmatrix} \Re\{\bar{\boldsymbol{P}}\} & -\Im\{\bar{\boldsymbol{P}}\} \\ \Im\{\bar{\boldsymbol{P}}\} & \Re\{\bar{\boldsymbol{P}}\} \end{bmatrix}$, with $\bar{\boldsymbol{P}} = \boldsymbol{X_t} \otimes \boldsymbol{I_B}$. The operator '$\otimes$' denotes Kronecker product.

As a result, the inputs to the network will be: $\boldsymbol{z_t} \in \mathbb{R}^{2BT \times 1}$, $\boldsymbol{h} \in \mathbb{R}^{2BT \times 2BU}$, $\boldsymbol{P} \in \mathbb{R}^{2BT \times 2BU}$, and $\boldsymbol{\omega_t} \in \mathbb{R}^{2BT \times 1}$.

2.2.3.1 Bussgang-Based Linear Channel Estimator

The Bussgang theorem is a fundamental concept that provides a mathematical framework for modeling a random input signal passing through a nonlinear system followed by a linear system. The theorem states that the output can be expressed as a linear combination of the input signal and a noise term, which is uncorrelated with the input signal. Consequently, the Bussgang theorem is useful for nonlinear distortion analysis, particularly in cases involving nonlinearities caused by low-resolution ADCs.

In this part, the Bussgang Linear Minimum Mean Square Error (BLMMSE) is studied on the model described in (2.10) as a baseline for linear channel estimation to be compared to the results from the deep unfolding network proposed for channel estimation [13].

The expression in (2.10) can be linearized using the Bussgang decomposition as (2.11).

$$\boldsymbol{z_t} = \boldsymbol{V_t P h} + \boldsymbol{V_t \omega_t} + \boldsymbol{n_t} \tag{2.11}$$

$$= \boldsymbol{A_t h} + \boldsymbol{e_t} \tag{2.12}$$

where $\boldsymbol{A_t} = \boldsymbol{V_t P} \in \mathbb{R}^{2BT \times 2BU}$, $\boldsymbol{V_t} \in \mathbb{R}^{2BT \times 2BT}$, and $\boldsymbol{e_t} = \boldsymbol{V_t \omega_t} + \boldsymbol{n_t}$.

In a one-bit ADC, $\boldsymbol{V_t}$ can be expressed as

$$\boldsymbol{V_t} = \frac{\Delta}{\sqrt{2\pi}} \mathrm{diag}(\boldsymbol{\Sigma_{y_t}})^{-\frac{1}{2}}. \tag{2.13}$$

Here, Δ represents the step size between the two quantization levels, and the covariance matrix $\boldsymbol{\Sigma_{y_t}} = \boldsymbol{P \Sigma_h P^T} + \frac{N_0}{2}\boldsymbol{I} \in \mathbb{R}^{2BT \times 2BT}$ characterizes $\boldsymbol{y_t} = \boldsymbol{Ph} + \boldsymbol{\omega_t}$.

Now the Bussgang-based linear channel estimation can be defined as

$$\hat{\boldsymbol{h}}_{BLMMSE} = \boldsymbol{\Sigma_{h z_t} \Sigma_{z_t}^{-1} z_t} \tag{2.14}$$

$$= \boldsymbol{\Sigma_h A_t^T \Sigma_{z_t}^{-1} z_t} \tag{2.15}$$

where $\boldsymbol{\Sigma_{h z_t}} \in \mathbb{R}^{2BU \times 2BT}$, $\boldsymbol{\Sigma_{z_t}} \in \mathbb{R}^{2BT \times 2BT}$, and

$$\boldsymbol{\Sigma_{z_t}} = \frac{\Delta^2}{2\pi} \arcsin\Big(\mathrm{diag}(\boldsymbol{\Sigma_{y_t}})^{-\frac{1}{2}} \boldsymbol{\Sigma_{y_t}} \mathrm{diag}(\boldsymbol{\Sigma_{y_t}})^{-\frac{1}{2}}\Big). \tag{2.16}$$

2.2.3.2 FBM-CENet

In this section, the application of deep unfolding in channel estimation is studied. Generally, deep unfolding is a model-based optimization algorithm, that combines traditional optimization algorithms and neural network architectures.[16] This purpose is achieved by unfolding each iteration of an optimization process, like gradient decent, into a neural network layer. In this architecture, each layer corresponds to one iteration of the optimization process.

In the case of channel estimation, the goal is to maximize the likelihood of the received signal given the channel. Hence, the first step is to define the maximum likelihood (ML) channel estimator.

The expression (2.10) can be rewritten as (2.17), where $\boldsymbol{P} = [\boldsymbol{p_1}, \boldsymbol{p_2}, ..., \boldsymbol{p_{2BT}}]^T$, $\boldsymbol{z_t} = [\mathrm{z}_{t,1}, \mathrm{z}_{t,2}, ..., \mathrm{z}_{t,2BT}]^T$, and $\boldsymbol{\omega_t} = [\mathrm{w}_{t,1}, \mathrm{w}_{t,2}, ..., \mathrm{w}_{t,2BT}]^T$.

$$z_{t,i} = \mathrm{sgn}(\boldsymbol{p_i^T h} + \omega_{t,i}) \qquad i = 1, 2, ..., 2BT. \tag{2.17}$$

The received signal is a one-bit quantized signal with upper and lower quantization thresholds of the bin to which $z_{t,i}$ belongs defined as

$$q_{t,i}^{up} = \begin{cases} z_{t,i} + \frac{\Delta}{2}, & if \;\; z_{t,i} < \tau \\ \infty & otherwise \end{cases} \tag{2.18}$$

$$q_{t,i}^{low} = \begin{cases} z_{t,i} - \frac{\Delta}{2}, & if \;\; z_{t,i} > \tau \\ -\infty & otherwise \end{cases} \tag{2.19}$$

where τ represents the quantization threshold. (Here, $\tau = 0$).

Utilizing these upper and lower thresholds, $s_{t,i}^{up}$ and $s_{t,i}^{low}$ can be defined as

$$s_{t,i}^{up} = \sqrt{2\rho}(q_{t,i}^{up} - \boldsymbol{p_i^T h}) \tag{2.20}$$

$$s_{t,i}^{low} = \sqrt{2\rho}(q_{t,i}^{low} - \boldsymbol{p_i^T h}) \tag{2.21}$$

where, $\rho = \frac{1}{N_0}$.
These values represent the deviation of the received signal from the expected signal which is a signal-to-noise ratio (SNR)-related quantity.

Using $s_{t,i}^{up}$, and $s_{t,i}^{low}$, the ML channel estimator is given as

$$\hat{\boldsymbol{h}}_{ML} = \arg\max_{\boldsymbol{h}} f(\boldsymbol{z_t}|\boldsymbol{h}) \tag{2.22}$$

$$= \arg\max_{\boldsymbol{h}} \sum_{i=1}^{2BT} \log[\Phi(s_{t,i}^{up}) - \Phi(s_{t,i}^{low})] \tag{2.23}$$

where, Φ represents the cumulative distributed function (CDF) of the standard normal distribution, defined as $\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{\tau^2}{2}} d\tau$.

To have a convex optimization problem, both the objective function and the feasible region corresponding to the constraint should be convex. This means that the feasible region needs to include all points within any intervals between its elements.[17] Now, considering $\mathcal{P}_t(\boldsymbol{h}) = \sum_{i=1}^{2BT} \log[\Phi(s_{t,i}^{up}) - \Phi(s_{t,i}^{low})]$ as the cost function, the optimization problem in (2.23) can be solved by utilizing an iterative gradient ascent algorithm, due to the fact that the expression in (2.23) is convex [13].

The gradient of the cost function $\mathcal{P}_t(\boldsymbol{h})$ is equivalent to

$$\nabla \mathcal{P}_t(\boldsymbol{h}) = \sum_{i=1}^{2BT} \frac{-\sqrt{2\rho} p_i \left(\phi(s_{t,i}^{up}) - \phi(s_{t,i}^{low})\right)}{\Phi(s_{t,i}^{up}) - \Phi(s_{t,i}^{low})} \tag{2.24}$$

where $\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$ is the power density function (PDF) to the standard normal random variable. Considering that the denominator of the gradient in (2.24) includes the subtraction of two CDFs of the standard normal distribution, and knowing that this CDF approaches zero and one rapidly, a concern is to have equal values for $\Phi(s_{t,i}^{up})$ and $\Phi(s_{t,i}^{low})$ in some steps of the iterative gradient descent, potentially

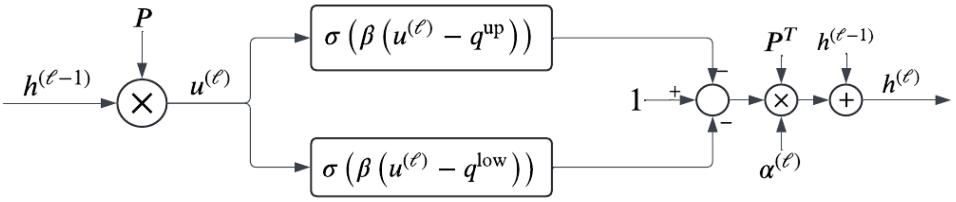

Figure 2.4: The structure of layer l of FBM-CENet from [13].

resulting in a zero denominator in (2.24).

To address this problem, the cost function $\mathcal{P}_t(\boldsymbol{h})$ can be approximated by the Sigmoid function as follows

$$\Phi(t) \approx \sigma(ct) = \frac{1}{1+e^{-ct}}. \tag{2.25}$$

If we set $c = 1.702$, it has been shown in [44], that the approximation error is $|\Phi(t) - \sigma(ct)| \leq 0.0095, \forall t \in \mathbb{R}$.
Now, the cost function $\mathcal{P}_t(\boldsymbol{h})$ can be rewritten as

$$\mathcal{P}_t(h) \approx \sum_{i=1}^{2BT} \log\left[\frac{1}{1+e^{-cs^{up}_{t,i}}} - \frac{1}{1+e^{-cs^{low}_{t,i}}}\right]. \tag{2.26}$$

As a result, the gradient of the cost function can be approximated as

$$\nabla\mathcal{P}_t(\boldsymbol{h}) \approx \sum_{i=1}^{2BT} c\sqrt{2\rho}p_i\left(1 - \frac{1}{1+e^{cs^{up}_{t,i}}} - \frac{1}{1+e^{cs^{low}_{t,i}}}\right) \tag{2.27}$$

$$= c\sqrt{2\rho}\boldsymbol{P^T}\left[1 - \sigma\left(c\sqrt{2\rho}(\boldsymbol{Ph} - \boldsymbol{q_t^{up}})\right) - \sigma\left(c\sqrt{2\rho}(\boldsymbol{Ph} - \boldsymbol{q_t^{low}})\right)\right]. \tag{2.28}$$

Now, defining each layer as

$$\boldsymbol{h^{(l)}} = \boldsymbol{h^{(l-1)}} + \alpha_t^{(l)}\nabla\mathcal{P}_t\left(\boldsymbol{h^{(l-1)}}\right) \tag{2.29}$$

where l is the iteration index, we are able to solve the optimization problem in (2.22) by applying iterative gradient descent.

In Figure 2.4, the structure of the derived deep unfolding network used for channel estimation is depicted.

The trainable parameters in this network are $\alpha_t^{(l)}$, for each layer l, and β_t. The constant $c\sqrt{2\rho}$ is absorbed into these parameters.

Considering Figure 2.4, and the expressions (2.28) and (2.29), $\alpha_t^{(l)}$ can be interpreted as the step size when moving along the gradient direction, while β_t, decides on the optimal direction to move along with step size $\alpha_t^{(l)}$ to get the best estimation of the channel. As a result, in this network we are jointly training the optimal directions and step sizes.

2.2.4 Data Detection

In this section, the same approach is used to build a deep unfolding network for data detection in one-bit massive MIMO.

To start, the expressions (2.3) and (2.4) are converted into the real domain as

$$\boldsymbol{z} = \operatorname{sgn}(\bar{\boldsymbol{H}}\boldsymbol{x} + \boldsymbol{\omega}) \tag{2.30}$$

where,

$$\boldsymbol{z} = \begin{bmatrix} \Re\{\boldsymbol{Z}_t\} \\ \Im\{\boldsymbol{Z}_t\} \end{bmatrix}, \ \boldsymbol{x} = \begin{bmatrix} \Re\{\boldsymbol{X}\} \\ \Im\{\boldsymbol{X}\} \end{bmatrix}, \text{ and } \bar{\boldsymbol{H}} = \begin{bmatrix} \Re\{\boldsymbol{H}\} & -\Im\{\boldsymbol{H}\} \\ \Im\{\boldsymbol{H}\} & \Re\{\boldsymbol{H}\} \end{bmatrix}.$$

As a result, $\boldsymbol{z} \in \mathbb{R}^{2B}$, $\boldsymbol{x} \in \mathbb{R}^{2U}$, $\boldsymbol{\omega} \in \mathbb{R}^{2B}$, and $\bar{\boldsymbol{H}} \in \mathbb{R}^{2B\times 2U}$.

2.2.4.1 Bussgang-Based Linear Data Detection

In this section, BLMMSE is going to be studied on the model described in (2.30) as a baseline for linear data detection to be compared to the results from the deep unfolding network.

By applying the Bussgang decomposition to (2.30), $\boldsymbol{z}$ can be written as

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{V}\bar{\boldsymbol{H}}\boldsymbol{x} + \boldsymbol{V}\boldsymbol{\omega} + \boldsymbol{e} & (2.31) \\ &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} & (2.32) \end{aligned}$$

where $\boldsymbol{A} = \boldsymbol{V}\bar{\boldsymbol{H}}$ and $\boldsymbol{V} \in \mathbb{R}^{2B\times 2B}$ is a diagonal matrix as

$$\boldsymbol{V} = \frac{\Delta}{\sqrt{2\pi}} \operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-\frac{1}{2}} \tag{2.33}$$

with $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \bar{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\bar{\boldsymbol{H}} + \frac{N_0}{2}\boldsymbol{I} \in \mathbb{R}^{2B\times 2B}$, and $\boldsymbol{n}$ is modeled as $\mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{n}})$, with

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{n}} = \frac{\Delta^2}{2\pi} \Big[& \arcsin\Big(\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-\frac{1}{2}}\boldsymbol{\Sigma}_{\boldsymbol{y}}\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-\frac{1}{2}}\Big) \\ & -\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-\frac{1}{2}}\boldsymbol{\Sigma}_{\boldsymbol{y}}\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-\frac{1}{2}} + \frac{N_0}{2}\operatorname{diag}(\boldsymbol{\Sigma}_{\boldsymbol{y}})^{-1}\Big]. \end{aligned} \tag{2.34}$$

Knowing this, $\boldsymbol{\Sigma}_{\boldsymbol{z}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{xz}}$ can be expressed as

$$\boldsymbol{\Sigma}_{\boldsymbol{z}} = \boldsymbol{V}\bar{\boldsymbol{H}}\boldsymbol{\Sigma}_{\boldsymbol{x}}\bar{\boldsymbol{H}}^{\boldsymbol{H}}\boldsymbol{V}^{\boldsymbol{H}} + \boldsymbol{\Sigma}_{\boldsymbol{n}} = \boldsymbol{A}\boldsymbol{A}^{\boldsymbol{H}} + \boldsymbol{\Sigma}_{\boldsymbol{n}} \tag{2.35}$$

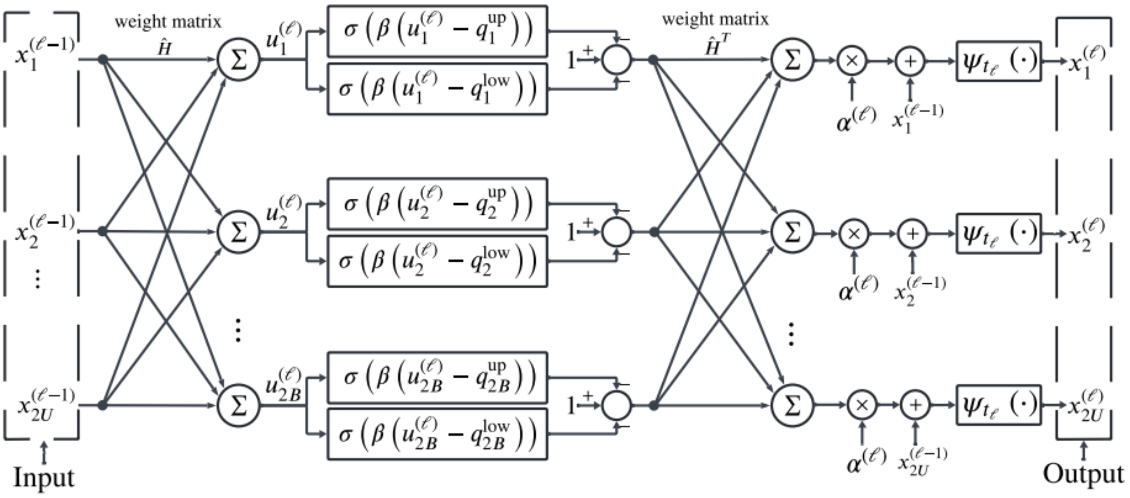

Figure 2.5: The structure of layer l of FBM-DetNet from [13].

$$\mathbf{\Sigma_{xz}} = \mathbf{\Sigma_x}\bar{\boldsymbol{H}}^{\boldsymbol{H}}\boldsymbol{V}^{\boldsymbol{H}} = \boldsymbol{A}^{\boldsymbol{H}} \tag{2.36}$$

respectively.

Now the Bussgang-based linear data detection can be defined as

$$\begin{aligned}\hat{\boldsymbol{x}}_{BLMMSE} &= \mathbf{\Sigma_{xz}}\mathbf{\Sigma_z^{-1}}\boldsymbol{z} \\ &= \boldsymbol{A}^{\boldsymbol{H}}(\boldsymbol{A}\boldsymbol{A}^{\boldsymbol{H}} + \mathbf{\Sigma_n})^{-1}\boldsymbol{z}.\end{aligned} \tag{2.37}$$

2.2.4.2 FBM-DetNet

In this section, the application of deep unfolding networks in data detection is going to be studied using the ML detection problem as

$$\hat{\boldsymbol{x}}_{\boldsymbol{ML}} = \arg\max_{\boldsymbol{X}\in\mathcal{M}^k} f(\boldsymbol{z}|\boldsymbol{x}) = \arg\max_{\boldsymbol{X}\in\mathcal{M}^k} \mathcal{P}(\boldsymbol{x}) \tag{2.38}$$

where $\mathcal{M}^k$, is a set of symbols used for mapping k bits of information to, and $\mathcal{P}(\boldsymbol{x})$ is approximated as shown in (2.26), such that $s_{t,i}^{up}$ and $s_{t,i}^{low}$ represent the deviation of the received signal from the expected signal defined as

$$s_{t,i}^{up} = \sqrt{2\rho}(q_i^{up} - \bar{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{T}}\boldsymbol{x}) \tag{2.39}$$

$$s_{t,i}^{low} = \sqrt{2\rho}(q_i^{low} - \bar{\boldsymbol{h}}_{\boldsymbol{i}}^{\boldsymbol{T}}\boldsymbol{x}) \tag{2.40}$$

with q_i^{up} and q_i^{low} defined in (2.18) and (2.19). The gradient of $\mathtt{P}(\bar{x})$ is equal to

$$\nabla \mathcal{P}(\boldsymbol{x}) \approx \sum_{i=1}^{2B} c\sqrt{2\rho}\bar{h}_i \left(1 - \frac{1}{1+e^{cs_i^{up}}} - \frac{1}{1+e^{cs_i^{low}}}\right) \tag{2.41}$$

$$= c\sqrt{2\rho}\bar{\boldsymbol{H}}^{\boldsymbol{T}} \left[1 - \sigma\left(c\sqrt{2\rho}(\bar{\boldsymbol{H}}\boldsymbol{x} - \boldsymbol{q}^{\boldsymbol{up}})\right) - \sigma\left(c\sqrt{2\rho}(\bar{\boldsymbol{H}}\boldsymbol{x} - \boldsymbol{q}^{\boldsymbol{low}})\right)\right]. \tag{2.42}$$

Considering the constraint $\boldsymbol{X} \in \mathcal{M}^k$ in (2.38), it is clear that that the optimization problem is not convex due to the discrete feasibility constraint.

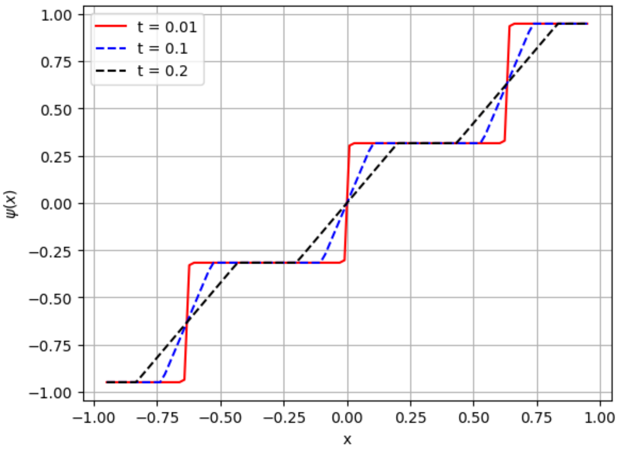

Figure 2.6: Projection function $\Psi_t(\boldsymbol{x})$ with different values of t from [13].

To address this problem, an iterative projected gradient descent method is used to train the network as shown below

$$\boldsymbol{x}^{(l)} = \Psi_{t_l}\left(\boldsymbol{x}^{(l-1)} - \alpha^{(l)}\nabla\mathcal{P}(\boldsymbol{x}^{(l-1)})\right). \tag{2.43}$$

Here, l is the iteration index, $\alpha^{(l)}$ is the step size, and $\Psi_{t_l}(.)$, is a non-linear trainable projector as

$$\Psi_{t_l}(x) = -(2^{b'}-1)\frac{\Delta'}{2} + \frac{\Delta'}{2t_l}\sum_{i=-B'}^{B'}\left[f_{relu}(x+i\Delta+t_l) - f_{relu}(x+i\Delta-t_l)\right] \tag{2.44}$$

that forces the signal to the nearest constellation point.

In (2.44), f_{relu} is the *ReLU* activation function, $B' = 2^{b'-1}-1$, and for 16-QAM signaling, $\{b', \Delta'\} = \{2, \frac{2}{\sqrt{10}}\}$.

The structure of the FBM-DetNet network is demonstrated in Figure 2.5, and the effect of the trainable parameter t_l is illustrated in Figure(2.6). It can be seen that a smaller t_l, makes the projector sharper.

3

Methods

In this chapter, we are going to use FBM-CENet and FBM-DetNet to estimate the channel and perform data detection in the distributed massive MIMO test-bed proposed in [5] in uplink mode. The BLMMSE method will also be studied on this test-bed and considered as a benchmark to evaluate the performance of the two proposed networks in [13].

3.1 Introduction

Considering the test-bed shown in Figure 2.2, we can clearly see that, compared to the massive MIMO model studied in Section 2.3, this test-bed uses the real part of the up-sampled and up-converted signal to transmit data over the air. Additionally, each AP includes an AGC and a comparator, which should be considered when modeling the test-bed.

3.2 System Model

In this section, we aim to define a suitable system model for the test-bed described in Section 2.1.2, considering only the case with one user ($U = 1$), and one AP ($B = 1$) to prevent complexity in the notation.

Assuming each user sending pilot signals of bandwidth W centered at a carrier frequency such that $f_c >> W$, and the sampling frequency $f_s \geq 2f_c + W$, the received 1-bit signal at CU can be described as

$$\boldsymbol{Z}^{RF} = \operatorname{sgn}\Big(\boldsymbol{Y}^{RF} + \boldsymbol{D}\Big) \tag{3.1}$$

where $\boldsymbol{D}$ with a size of $N \times N_P$, is the dither signal matrix, with independent entries drawn from the normal distribution $\mathcal{N}(0, E_d/2)$, and $\boldsymbol{Y}^{RF}$ with a size of $N \times N_P$, is defined as

$$\boldsymbol{Y}^{RF} = \sqrt{2}\Re\{\operatorname{diag}(\boldsymbol{u})\boldsymbol{Y}^{BB}\}. \tag{3.2}$$

Here u is an N-dimensional vector used to model the up-conversion, with elements defined as $u_n = e^{j2\pi(f_c/f_s)n}$, where $n = 0, 1, \ldots, N-1$ and N is the length of the observed signal at the ADC.

In order to prevent aliasing, a symmetric set $\mathcal{S}$ is defined as (2.5), where N, is the length of the up-sampled signal, and S is the length of the channel.
Now, To incorporate the effect of oversampling into the model, $\boldsymbol{Y}^{BB}$ can be defined as (3.3).

$$\boldsymbol{Y}^{BB} = \boldsymbol{F}^{inv}(\boldsymbol{H}\boldsymbol{X} + \boldsymbol{W}) \tag{3.3}$$

Here, the channel matrix $\boldsymbol{H}$ has a size of $S \times U$ with its elements in the frequency domain. The matrix $\boldsymbol{X}$ contains the transmitted symbols, also in the frequency domain, with a size of $U \times N_P$, where N_p is the length of the transmitted symbols before up-sampling. The noise matrix $\boldsymbol{W}$ has a size of $S \times N_P$, and is generated from a complex normal distribution $\mathcal{CN}(0, N_0)$.

The matrix $\boldsymbol{F}^{inv}$ with a size of $N \times S$, is generated by selecting the rows corresponding to the indices in the set $\mathcal{S}$ from the inverse discrete Fourier transform (IDFT) of size N. Based on this, $\boldsymbol{Y}^{BB}$ will be a matrix of oversampled received symbols in the time domain.

3.3 Channel Estimation

This section presents the theoretical background from [18] including the utilization of the FBM-CENet proposed in [13] for channel estimation within the one-bit radio-over-fiber distributed massive MIMO model discussed in [5]. To achieve this, the model shown in Section 3.1.2 will be vectorized for use in the channel estimation process. For this purpose, the matrix $\boldsymbol{Y}^{BB}$ expressed in (3.3) is rewritten as

$$\boldsymbol{Y}^{BB} = \boldsymbol{F}^{inv}(\boldsymbol{H}\boldsymbol{P} + \boldsymbol{W}) \tag{3.4}$$

where matrix $\boldsymbol{P}$ contains the pilot symbols, with the same size as matrix $\boldsymbol{X}$.
Now, we vectorize the matrix $\boldsymbol{Z}$ to obtain $\boldsymbol{z} = vec(\boldsymbol{Z}) \in \mathbb{R}^{NN_P \times 1}$, and perform the same operation for matrices $\boldsymbol{Y}^{RF}, \boldsymbol{D}, \boldsymbol{Y}^{BB}, \boldsymbol{H}$, and $\boldsymbol{W}$, resulting in $\boldsymbol{y}^{RF}, \boldsymbol{d}, \boldsymbol{y}^{BB}$, and $\boldsymbol{w}$, respectively. This transformation changes expressions (3.1), (3.2), and (3.4) to

$$\boldsymbol{z}_n^{RF} = \operatorname{sgn}\left(\sqrt{2}\Re\{\boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}(\tilde{\boldsymbol{P}}\boldsymbol{h}_k + \boldsymbol{w}_k)\} + \boldsymbol{d}_k\right), \qquad n = 0, 1, 2, ..., N-1 \tag{3.5}$$

$$\boldsymbol{y}_n^{RF} = \sqrt{2}\Re\{\boldsymbol{U}\boldsymbol{y}_n^{BB}\} \tag{3.6}$$

$$\boldsymbol{y}_n^{BB} = \tilde{\boldsymbol{F}}^{inv}(\tilde{\boldsymbol{P}}\boldsymbol{h}_k + \boldsymbol{w}_k) \tag{3.7}$$

where $\boldsymbol{U} = \boldsymbol{I_{N_P}} \otimes \operatorname{diag}(\boldsymbol{u})$, $\tilde{\boldsymbol{F}}^{inv} = \boldsymbol{I_{N_P}} \otimes \boldsymbol{F}^{inv}$, and $\tilde{\boldsymbol{P}} = \boldsymbol{P^T} \otimes \boldsymbol{I_S}$.

3.3.1 Bussgang-Based Linear Channel Estimator

In this section, the BLMMSE channel estimation will be studied and used as a benchmark. It will be compared with the results from the deep unfolding network presented in the next section, for performing channel estimation in the system described in Section 3.2.1.

Using the Bussgang Theorem, the non-linear 1-bit received signal $\boldsymbol{z}_n^{RF} = \operatorname{sgn}(\boldsymbol{y}_n^{RF} + \boldsymbol{d}_n)$, shown in (3.5), can be linearized as

$$\boldsymbol{z}_n^{RF} = \boldsymbol{V}\boldsymbol{q}_n + \boldsymbol{e}_n \tag{3.8}$$

where $\boldsymbol{q}_n = \boldsymbol{y}_n^{RF} + \boldsymbol{d}_n$, $\boldsymbol{V} = \sqrt{\frac{2}{\pi}}\operatorname{diag}(\boldsymbol{\Sigma_q})^{-1/2}$, and $\boldsymbol{\Sigma_q}$ is the auto-covariance of $\boldsymbol{q}_n$. Additionally, $\boldsymbol{e}_n$ represents the quantization distortion, which is uncorrelated with $\boldsymbol{q}_n$.

Since the dither signal is white and independent of the received signal $\boldsymbol{y}_n^{RF}$, $\boldsymbol{\Sigma_q}$ can be calculated as (3.9).

$$\boldsymbol{\Sigma_q}[m] = \boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m] + \frac{E_d}{2}\boldsymbol{I_B}\delta[m], \qquad m = 0, 1, 2, ..., N-1 \tag{3.9}$$

where, $\boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m] = \mathbb{E}[\boldsymbol{y}_n^{RF}(\boldsymbol{y}_{n-m}^{RF})]$, can be calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m] = \frac{1}{N}\Re\{\sum_{k\in\mathcal{S}}(\tilde{\boldsymbol{P}}\boldsymbol{C}_{\boldsymbol{h}_k}\tilde{\boldsymbol{P}}^H + N_0\boldsymbol{I_B})e^{j2\pi(k/N+f_c/f_s)m}\} \tag{3.10}$$

where N_0 is the variance of the $AWGN$ noise $\boldsymbol{\omega}$, and $\boldsymbol{C}_{\boldsymbol{h}_k}$ represents the prior covariance matrix of the channel $\boldsymbol{h}_k$. If the components of $\boldsymbol{h}_k$ are i.i.d. random variables, the covariance matrix simplifies to $\sigma_{\boldsymbol{h}_k}^2\boldsymbol{I}_B$, where $\sigma_{\boldsymbol{h}_k}^2$ represents the common variance.

Now, the auto-covariance matrix of the quantization error $\boldsymbol{e}_n$ can be written as

$$\boldsymbol{\Sigma_e}[m] = \boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}[m] - \boldsymbol{V}\boldsymbol{\Sigma_q}[m]\boldsymbol{V}. \tag{3.11}$$

Using the arcsine-law [19], $\boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}$ can be expressed as

$$\boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}[m] = \frac{2}{\pi}\arcsin\left(\operatorname{diag}(\boldsymbol{\Sigma_q}[0])^{-1/2}\boldsymbol{\Sigma_q}[m]\operatorname{diag}(\boldsymbol{\Sigma_q}[0])^{-1/2}\right). \tag{3.12}$$

In the receiver, the received up-sampled and up-converted signal is down-converted and down-sampled for further processes and data extraction. As a result, the received signal $\boldsymbol{z}_n^{RF}$, is converted to $\boldsymbol{z}_k^{BB}$, as follows

$$\boldsymbol{z}_k^{BB} = \sqrt{\frac{2}{N}}\sum_{n=0}^{N-1}\boldsymbol{z}_n^{RF}e^{-j2\pi(k/N+f_c/f_s)n}. \tag{3.13}$$

More compactly,

$$\boldsymbol{z}_k^{BB} = \boldsymbol{V}\left(\tilde{\boldsymbol{P}}\boldsymbol{h}_k + \hat{\boldsymbol{\omega}}_k + \hat{\boldsymbol{d}}_k\right) + \hat{\boldsymbol{e}}_k \tag{3.14}$$

where, $\hat{\boldsymbol{\omega}}_k$, $\hat{\boldsymbol{d}}_k$, and $\hat{\boldsymbol{e}}_k$ are given by

$$\hat{\boldsymbol{\omega}}_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \boldsymbol{\omega}_n e^{-j2\pi(k/N+f_c/f_s)n} \tag{3.15}$$

$$\hat{\boldsymbol{d}}_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \boldsymbol{d}_n e^{-j2\pi(k/N+f_c/f_s)n} \tag{3.16}$$

$$\hat{\boldsymbol{e}}_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \boldsymbol{e}_n e^{-j2\pi(k/N+f_c/f_s)n}. \tag{3.17}$$

To calculate the covariance matrix $\boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}_k} = \mathbb{E}\left[\hat{\boldsymbol{e}}_k(\hat{\boldsymbol{e}}_k)^H\right]$, we can use $\boldsymbol{\Sigma}_{\boldsymbol{e}}[m]$ from (3.11) and obtain

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}k} = 2 \sum_{m=0}^{N-1} \boldsymbol{\Sigma}_{\boldsymbol{e}}[m] e^{-j2\pi(k/N+f_c/f_s)m}. \tag{3.18}$$

The BMMSE-based channel estimation can be written as

$$\hat{\boldsymbol{h}}_{BMMSE} = \boldsymbol{A}_k \hat{\boldsymbol{z}}_k^{BB} \tag{3.19}$$

where $\boldsymbol{A}_k = \boldsymbol{\Sigma}_{\boldsymbol{h}_k \hat{\boldsymbol{z}}_k^{BB}} \boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}_k^{BB}}^{-1} \boldsymbol{V}^{-1}$. The terms $\boldsymbol{\Sigma}_{\boldsymbol{h}_k \hat{\boldsymbol{z}}_k^{BB}}$ and $\boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}_k^{BB}}$ can be calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{h}_k \hat{\boldsymbol{z}}_k^{BB}} = \boldsymbol{C}_{\boldsymbol{h}_k} \tilde{\boldsymbol{P}}^H \tag{3.20}$$

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}_k^{BB}} = \tilde{\boldsymbol{P}} \boldsymbol{C}_{\boldsymbol{h}_k} \tilde{\boldsymbol{P}}^H + (N_0 + E_d)\boldsymbol{I}_{\boldsymbol{B}} + \boldsymbol{V}^{-1} \boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}_k} V^{-1}. \tag{3.21}$$

3.3.2 Deep Unfolding Channel Estimation Network

In this section, the deep unfolding network discussed in section 2.2.3.2 for channel estimation is further extended to be utilized in the system described in section 3.2.1.

We start by considering $\boldsymbol{M} = \boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}\tilde{\boldsymbol{P}} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{NN_P}]^T$, which allows us to rewrite (3.7) as

$$\boldsymbol{z}^{RF} = \operatorname{sgn}(\sqrt{2}\Re\{\boldsymbol{M}\boldsymbol{h} + \boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}\boldsymbol{\omega}\} + \boldsymbol{d}). \tag{3.22}$$

The entries of the term $\boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}\boldsymbol{\omega}$ in (3.22) are correlated, complicating the calculations for the new channel estimation network. To address this problem, the additive noise $\boldsymbol{\omega}$ is ignored in the mathematical derivations of the deep unfolding network. As a result, (3.22) can be rewritten as

$$\boldsymbol{z}^{RF} = \operatorname{sgn}(\sqrt{2}\Re\{\boldsymbol{M}\boldsymbol{h}\} + \boldsymbol{d}). \tag{3.23}$$

The ML channel estimation of $\boldsymbol{h}$ is given as

$$\begin{aligned}\hat{\boldsymbol{h}}_{ML} &= \arg\max_{\boldsymbol{h}} f(\boldsymbol{z}|\boldsymbol{h}) \\ &= \arg\max_{\boldsymbol{h}} \sum_{i=0}^{NN_P} \log\Big[\Phi\left(\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right) \\ &\quad -\Phi\left(\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right)\Big] \qquad (3.24)\end{aligned}$$

where $\rho = \sqrt{2/E_d}$.

Considering $\mathcal{P}_t(\boldsymbol{h}) = \sum_{i=1}^{NN_P} \log[\Phi(s_{t,i}^{up}) - \Phi(s_{t,i}^{low})]$ as the cost function we want to maximize, the optimization problem in (3.24) can be solved via an iterative gradient ascent due to the fact that the expression in (3.24) is convex.

Using the approximation in (2.25), we can rewrite $\mathcal{P}_t(\boldsymbol{h})$ as

$$\begin{aligned}\tilde{\mathcal{P}}_t(\boldsymbol{h}) = \sum_{i=0}^{NN_P} \log\Big[\sigma\left(c\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right) \\ -\sigma\left(c\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right)\Big]. \qquad (3.25)\end{aligned}$$

Since we are using the real part of the up-converted signal in the cost function $\mathcal{P}_t(\boldsymbol{h})$, to calculate the gradient of $\mathcal{P}_t(\boldsymbol{h})$, we need to compute the vector-valued Wirtinger derivative. The Wirtinger derivative of $\mathcal{P}_t(\boldsymbol{h})$ is given by

$$\begin{aligned}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{h})}{\partial\boldsymbol{h}^*} = \sum_{i=0}^{NN_P} c\sqrt{\rho}\Big[\sigma\left(c\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right) \\ -\sigma\left(c\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{h}\})\right)\Big]\frac{\partial\Re\{\boldsymbol{m}_i^T\boldsymbol{h}\}}{\partial\boldsymbol{h}^*} \qquad (3.26)\end{aligned}$$

where, $\frac{\partial\Re\{\boldsymbol{m}_i^T\boldsymbol{h}\}}{\partial\boldsymbol{h}^*} = \frac{\partial}{\partial\boldsymbol{h}^*}\left(\frac{1}{2}\left[(\boldsymbol{m}_i^H\boldsymbol{h}^*)^T + \boldsymbol{m}_i^T\boldsymbol{h}\right]\right) = \frac{\boldsymbol{m}_i^*}{2}$.

Now, $\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{h})}{\partial\boldsymbol{h}^*}$ can be written as

$$\begin{aligned}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{h})}{\partial\boldsymbol{h}^*} = \frac{c\sqrt{\rho}}{2}\boldsymbol{M}^*\left[1 - \sigma\left(c\sqrt{\rho}(\boldsymbol{q}^{up} - \Re\{\boldsymbol{M}\boldsymbol{h}\}\right)\right) \\ -\sigma\left(c\sqrt{\rho}(\boldsymbol{q}^{low} - \Re\{\boldsymbol{M}\boldsymbol{h}\})\right)\Big]. \qquad (3.27)\end{aligned}$$

Now, using (3.27), we can write the iterative gradient ascent algorithm to estimate the channel by maximizing the cost function $\tilde{\mathcal{P}}_t(\boldsymbol{h})$ as

$$\boldsymbol{h}^{(l)} = \boldsymbol{h}^{(l-1)} + \alpha^{(l)}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{h})}{\partial\boldsymbol{h}^*}\bigg|_{\boldsymbol{h}=\boldsymbol{h}^{(l-1)}} \qquad (3.28)$$

where l is the iteration index, which in a deep unfolding network corresponds to the number of the layer as $l \in \{1, \ldots, L\}$, and $\alpha^{(l)}$ is the step size. Now, each layer of the deep unfolding network implements

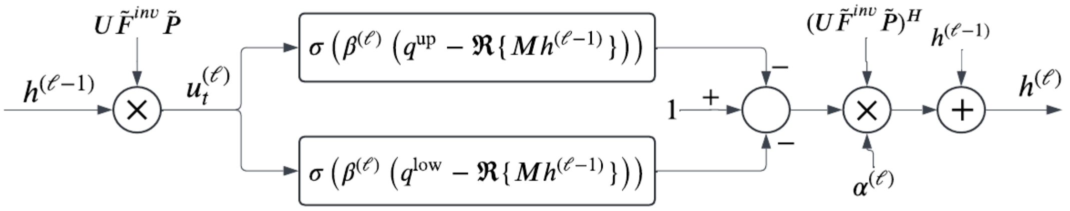

Figure 3.1: The structure of layer l of the proposed DNN for channel estimation in [18].

$$\begin{aligned}\boldsymbol{h}^{(l)} = \boldsymbol{h}^{(l-1)} + \alpha^{(l)}\boldsymbol{M}^{*}\Big[1 - \sigma\left(\beta^{(l)}(\boldsymbol{q}^{up} - \Re\{\boldsymbol{M}\boldsymbol{h}\})\right) \\ -\sigma\left(\beta^{(l)}(\boldsymbol{q}^{low} - \Re\{\boldsymbol{M}\boldsymbol{h}\})\right)\Big]. \qquad (3.29)\end{aligned}$$

Here, $\alpha^{(l)}$ and $\beta^{(l)}$ are the trainable parameters, interpreted as the step size and the optimal direction to move along for obtaining the best estimation of the channel, respectively.

In Figure 3.1, the structure of layer l of the channel estimation network is illustrated.

3.4 Data Detection

In this section, the model described in Section 3.1.2 will be vectorized for use in the data detection process. For this purpose, matrices $\boldsymbol{Z}$, $\boldsymbol{Y}^{RF}$, $\boldsymbol{D}$, $\boldsymbol{Y}^{BB}$, $\boldsymbol{X}$, and $\boldsymbol{W}$ should be vectorized. This transformation changes expressions (3.1), (3.2), and (3.6) to

$$\boldsymbol{z}_n^{RF} = \mathrm{sgn}\left(\sqrt{2}\Re\{\boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}(\tilde{\boldsymbol{H}}\boldsymbol{x} + \boldsymbol{w}_k)\} + \boldsymbol{d}_k\right), \qquad n = 0, 1, 2, ..., N-1 \qquad (3.30)$$

$$\boldsymbol{y}_n^{RF} = \sqrt{2}\Re\{\boldsymbol{U}\boldsymbol{y}_n^{BB}\} \qquad (3.31)$$

$$\boldsymbol{y}_n^{BB} = \tilde{\boldsymbol{F}}^{inv}(\tilde{\boldsymbol{H}}\boldsymbol{x} + \boldsymbol{w}_k) \qquad (3.32)$$

where $\tilde{\boldsymbol{H}} = \mathrm{diag}(\boldsymbol{h})$, and $\boldsymbol{x} = vec(\boldsymbol{X}) \in \mathbb{R}^{UN_P \times 1}$. $\tilde{\boldsymbol{F}}^{inv}$ and $\boldsymbol{U}$ are defined as in Section 3.2.1.

3.4.1 Bussgang-Based Linear Data Detection

Considering the system model in Section 3.3, in this section, the BMMSE-based data detection is studied.

Using the Bussgang Theorem, the one-bit received signal $\boldsymbol{z}_n^{RF}$ can be written as

$$\boldsymbol{z}_n^{RF} = \boldsymbol{V}\boldsymbol{q}_n + \boldsymbol{e}_n \tag{3.33}$$

where $\boldsymbol{q}_n = \boldsymbol{y}_n^{RF} + \boldsymbol{d}_n$, $\boldsymbol{V} = \sqrt{\frac{2}{\pi}}\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{q}})^{-1/2}$. Additionally, $\boldsymbol{e}_n$ represents the quantization distortion, which is uncorrelated with $\boldsymbol{q}_n$.

Now, $\boldsymbol{\Sigma}_{\boldsymbol{q}}$ can be calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{q}}[m] = \boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m] + \frac{E_d}{2}\boldsymbol{I}_{\boldsymbol{B}}\delta[m], \qquad m = 0, 1, 2, ..., N-1 \tag{3.34}$$

where, $\boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m]$, can be calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{y}^{RF}}[m] = \frac{1}{N}\Re\{\sum_{k\in\mathcal{S}}(\tilde{\boldsymbol{H}}\boldsymbol{C}_{\boldsymbol{x}}\tilde{\boldsymbol{H}}^{H} + N_0\boldsymbol{I}_{\boldsymbol{B}})e^{j2\pi(k/N+f_c/f_s)m}\} \tag{3.35}$$

where $\boldsymbol{C}_{\boldsymbol{x}}$, represents the auto-covariance matrix of the transmitted signal.

Now, the auto-covariance matrix of the quantization error $\boldsymbol{e}_n$ can be written as

$$\boldsymbol{\Sigma}_{\boldsymbol{e}}[m] = \boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}[m] - \boldsymbol{V}\boldsymbol{\Sigma}_{\boldsymbol{q}}[m]\boldsymbol{V}. \tag{3.36}$$

Using the arcsine-law, $\boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}$ can be expressed as

$$\boldsymbol{\Sigma}_{\boldsymbol{z}^{RF}}[m] = \frac{2}{\pi}\arcsin\left(\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{q}}[0])^{-1/2}\Sigma_q[m]\text{diag}(\boldsymbol{\Sigma}_{\boldsymbol{q}}[0])^{-1/2}\right). \tag{3.37}$$

In the receiver, the received up-sampled and up-converted signal is down-converted and down-sampled for further processes and data extraction. As a result, the received signal $\boldsymbol{z}_n^{RF}$, is converted to $\boldsymbol{z}_k^{BB}$. The same applies to $\boldsymbol{\omega}_n$, $\boldsymbol{d}_n$, and $\boldsymbol{e}_n$ which all get down-converted and down-sampled to $\hat{\boldsymbol{\omega}}_k$, $\hat{\boldsymbol{d}}_k$,and $\hat{\boldsymbol{e}}_k$ as shown in (3.15), (3.16), and (3.17), respectively.

To calculate the covariance matrix $\boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}_k}$, we can use $\boldsymbol{\Sigma}_{\boldsymbol{e}}[m]$ from (3.36)and obtain

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}_k} = 2\sum_{m=0}^{N-1}\boldsymbol{\Sigma}_{\boldsymbol{e}}[m]e^{-j2\pi(k/N+f_c/f_s)m}. \tag{3.38}$$

The BMMSE-based data detection can be written as

$$\hat{\boldsymbol{x}}_{BMMSE} = \boldsymbol{A}_k\hat{\boldsymbol{z}}_k^{BB} \tag{3.39}$$

where $\boldsymbol{A}_k = \boldsymbol{\Sigma}_{\boldsymbol{x}\hat{\boldsymbol{z}}^{BB}}\boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}^{BB}}^{-1}\boldsymbol{V}^{-1}$. The terms $\boldsymbol{\Sigma}_{\boldsymbol{x}\hat{\boldsymbol{z}}^{BB}}$ and $\boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}^{BB}}$ can be calculated as

$$\boldsymbol{\Sigma}_{\boldsymbol{x}\hat{\boldsymbol{z}}^{BB}} = \boldsymbol{C}_{\boldsymbol{x}}\tilde{\boldsymbol{H}}^{H} \tag{3.40}$$

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{z}}_k^{BB}} = \tilde{\boldsymbol{H}}\boldsymbol{C}_{\boldsymbol{x}}\tilde{\boldsymbol{H}}^{H} + (N_0 + E_d)\boldsymbol{I}_{\boldsymbol{B}} + \boldsymbol{V}^{-1}\boldsymbol{\Sigma}_{\hat{\boldsymbol{e}}_k}\boldsymbol{V}^{-1}. \tag{3.41}$$

3.4.2 Deep Unfolding Data Detection Network

In this section, the deep unfolding network discussed in Section 3.2.3 for channel estimation is adapted to be utilized in data detection within the system described in Section 3.3.

We start by considering $\boldsymbol{M} = \boldsymbol{U}\tilde{\boldsymbol{F}}^{inv}\tilde{\boldsymbol{H}} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{NN_P}]^T$, which allows us to rewrite (3.30) as

$$\boldsymbol{z}^{RF} = \operatorname{sgn}(\sqrt{2}\Re\{\boldsymbol{M}\boldsymbol{x}\} + \boldsymbol{d}). \tag{3.42}$$

Here, again the additive noise $\boldsymbol{\omega}$ is neglected due to the correlation between it's elements which makes the calculations of the data detection network complicated.

The ML data detection of $\boldsymbol{x}$ is given as

$$\begin{aligned}\hat{\boldsymbol{x}}_{ML} &= \arg\max_{\boldsymbol{x}} f(\boldsymbol{z}|\boldsymbol{x})\\ &= \arg\max_{\boldsymbol{x}} \sum_{i=0}^{NN_P} \log\Big[\Phi\left(\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\\ &\quad -\Phi\left(\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\Big]\end{aligned} \tag{3.43}$$

where $\rho = \sqrt{2/E_d}$.
Considering $\mathcal{P}_t(\boldsymbol{x}) = \sum_{i=1}^{NN_P} \log[\Phi(s_{t,i}^{up}) - \Phi(s_{t,i}^{low})]$ as the cost function we want to maximize, the optimization problem in (3.43) can be solved via an iterative gradient ascent due to the fact that the expression in (3.43) is convex.

Using the approximation in (2.25), we can rewrite $\mathcal{P}_t(\boldsymbol{x})$ as

$$\begin{aligned}\tilde{\mathcal{P}}_t(\boldsymbol{x}) = &\sum_{i=0}^{NN_P} \log\Big[\sigma\left(c\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\\ &-\sigma\left(c\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\Big].\end{aligned} \tag{3.44}$$

Since we are using the real part of the up-converted signal in the cost function $\mathcal{P}_t(\boldsymbol{x})$, to calculate the gradient of $\mathcal{P}_t(\boldsymbol{x})$, we need to compute the vector-valued Wirtinger derivative. The Wirtinger derivative of $\mathcal{P}_t(\boldsymbol{x})$ is given by

$$\begin{aligned}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{x})}{\partial\boldsymbol{x}^*} = &\sum_{i=0}^{NN_P} c\sqrt{\rho}\Big[\sigma\left(c\sqrt{\rho}(q_i^{up} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\\ &-\sigma\left(c\sqrt{\rho}(q_i^{low} - \Re\{\boldsymbol{m}_i^T\boldsymbol{x}\})\right)\Big]\frac{\partial\Re\{\boldsymbol{m}_i^T\boldsymbol{x}\}}{\partial\boldsymbol{x}^*}\end{aligned} \tag{3.45}$$

where, $\frac{\partial\Re\{\boldsymbol{m}_i^T\boldsymbol{x}\}}{\partial\boldsymbol{x}^*} = \frac{\partial}{\partial\boldsymbol{x}^*}\left(\frac{1}{2}\left[(\boldsymbol{m}_i^H\boldsymbol{x}^*)^T + \boldsymbol{m}_i^T\boldsymbol{x}\right]\right) = \frac{\boldsymbol{m}_i^*}{2}$.
Now, $\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{x})}{\partial\boldsymbol{x}^*}$ can be written as

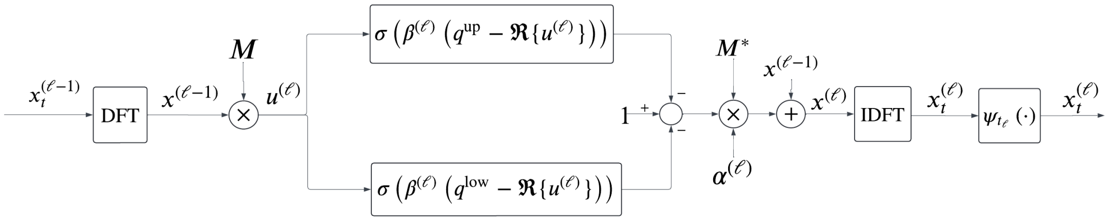

Figure 3.2: The structure of layer l of the proposed DNN for data detection.

$$\begin{aligned}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{x})}{\partial\boldsymbol{x}^*} = \frac{c\sqrt{\rho}}{2}\boldsymbol{M}^*\,&[1 - \sigma\left(c\sqrt{\rho}(\boldsymbol{q}^{up} - \Re\{\boldsymbol{M}\boldsymbol{x}\})\right)\\ &-\sigma\left(c\sqrt{\rho}(\boldsymbol{q}^{low} - \Re\{\boldsymbol{M}x\})\right)\Big]\,. \end{aligned} \tag{3.46}$$

Now, using (3.46), we can write the iterative gradient ascent algorithm to estimate the channel by maximizing the cost function $\tilde{\mathcal{P}}_t(\boldsymbol{x})$ as

$$\boldsymbol{x}^{(l)} = \boldsymbol{x}^{(l-1)} + \alpha^{(l)}\frac{\partial\tilde{\mathcal{P}}_t(\boldsymbol{x})}{\partial\boldsymbol{x}^*}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(l-1)}} \tag{3.47}$$

where $\alpha^{(l)}$ is the step size. Now, each layer of the deep unfolding network is as bellow

$$\begin{aligned}\boldsymbol{x}^{(l)} = \boldsymbol{x}^{(l-1)} + \alpha^{(l)}\boldsymbol{M}^*\,&\left[1 - \sigma\left(\beta^{(l)}(\boldsymbol{q}^{up} - \Re\{\boldsymbol{M}\boldsymbol{x}\})\right)\right.\\ &\left.-\sigma\left(\beta^{(l)}(\boldsymbol{q}^{low} - \Re\{\boldsymbol{M}\boldsymbol{x}\})\right)\right]\,. \end{aligned} \tag{3.48}$$

Here, $\alpha^{(l)}$ and $\beta^{(l)}$ are the trainable parameters, interpreted as the step size and the optimal direction to move along for obtaining the best estimation of the channel. Considering the constraint $\boldsymbol{x} \in \mathcal{M}^k$ in demapping, it is clear that the optimization problem is not convex due to the discrete feasibility constraint. To address this issue, a non-linear projection, as shown in (2.44), is applied to the inverse discrete Fourier transform (IDFT) of $\boldsymbol{x}^{(l)}$ in each iteration, as

$$\boldsymbol{x}_t = \Psi\left(\text{IDFT}(\boldsymbol{x}^{(l)})\right) \tag{3.49}$$

where $\boldsymbol{x}_t$ is the matrix of estimated symbols in the time domain, and $\boldsymbol{x}$ is the matrix of estimated symbols in the frequency domain. This transformed signal is then converted back to the frequency domain to pass through the network again. The structure of this DNN is shown in Figure 3.2.

4

Results

In this chapter, the performance of the systems and theories explained in Chapter 2 and Chapter 3 are examined, and their results are analyzed and compared.

4.1 The Results of the Distributed Massive MIMO Test-bed

In this section, the results of the distributed massive MIMO system described in Section 2.1 are depicted, including the results from both downlink and uplink.

4.1.1 Downlink

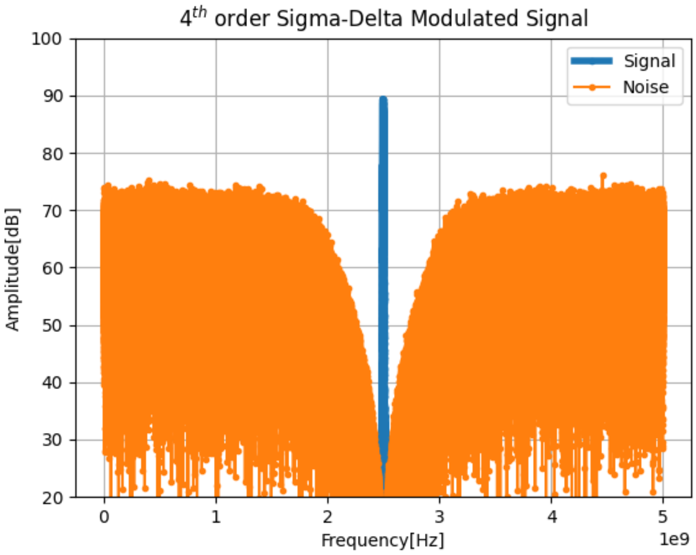

Figure 4.1: Frequency-domain representation of the 4^{th}-order sigma-delta modulated band-pass signal, along with the quantization noise.

To demonstrate the effect of using a sigma-delta modulator in downlink, as shown in Figure 2.3, we consider a 16-QAM pass-band signal with a $10MHz$ bandwidth. The signal is centered at a frequency $f_c = \frac{f_s}{4}Hz$, where $f_s = 10GHz$ is the sampling frequency utilized by the sigma-delta modulator. Additionally, a root-raised cosine

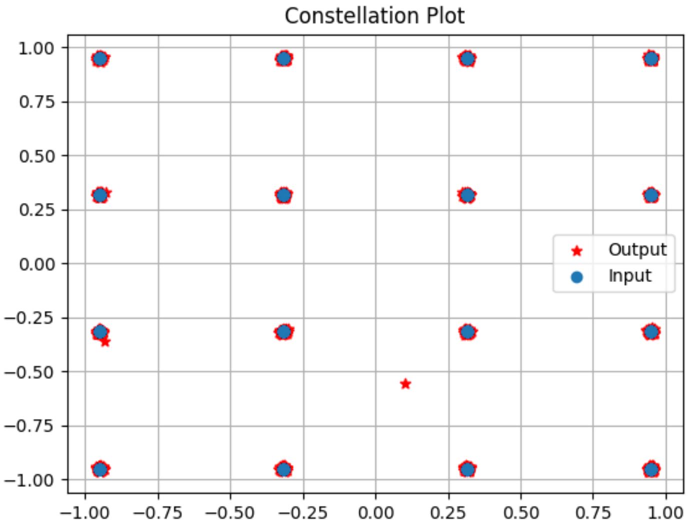

Figure 4.2: Constellation plot showing the input symbols to the sigma-delta modulator and the sigma-delta modulated symbols after passing through the BPF at the RRH.

waveform with roll-off factor 0.2 is used for pulse shaping.

Figure 4.1 illustrates the sigma-delta modulated band-pass signal in the frequency domain, as well as the quantization noise generated by the ADC.

After sigma-delta modulation, the band-pass signal and quantization noise in the frequency domain demonstrate that the 4^{th}-order sigma-delta modulator effectively shifts the quantization noise to higher frequencies. Applying a band-pass filter to this modulated signal helps mitigate the non-linearity introduced by the 1-bit DAC.

The constellation diagram shown in Figure 4.2 illustrates the 16-QAM symbols plotted after sigma-delta conversion at CU and band-pass filtering at the RRH. The per-symbol MSE achieved is $-37dB$. indicating marginal in-band distortion caused by the sigma-delta modulator. This demonstrates that it is possible to recover the symbols from the two-level received signal through the optical fiber at the RRH. Consequently, the use of optical fibers is feasible with minimal issues, ensuring phase coherence across all RRHs.

4.1.2 Uplink

To study the uplink scenario, a single-carrier RF signal centered at $f_c = 2.365GHz$, with a bandwidth of $BW = 10MHz$ is used to transmit 16-QAM symbols. A root-raised cosine waveform with roll-off factor 0.2 is applied for pulse shaping. The RF signal is then passed through an AWGN channel with $SNR = 10dB$.

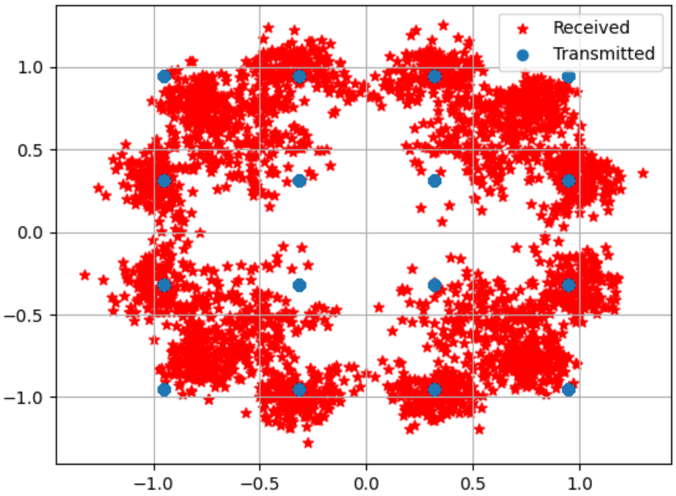

Figure 4.3: Constellation plot showing the transmitted symbols and the received symbols after passing through an AWGN channel with an SNR of $10dB$ and one-bit quantization without dithering.

In the first case, the received signal at the RRH, after passing through a band-pass filter and AGC, is fed into the 1-bit ADC without adding a dither signal. This processed signal is then ready to be transmitted through the optical fiber to the CU.

Figure 4.3 shows the constellation plot of the received symbols at the CU after undergoing digital signal processing, as well as the transmitted symbols.

It can be observed that the quantization distortion caused by the comperator in the RRH noticeably degrades the system's performance. The per-symbol MSE achieved in this case is $-12dB$.

In the second case, a dither signal is added to the RF signal after it passes through the AGC and before it reaches the comparator. The dither signal, a triangular waveform with a fundamental frequency of $20MHz$, is generated at the CU. After passing through the sigma-delta modulator, the dither signal is transmitted over the optical fiber and filtered at the RRH using a low-pass filter.

In Figure 4.4, the constellation of the demodulated symbols at the CU is plotted. The ratio between the power of the dither signal and the power of the RF signal is set to $2dB$. As it can be seen, the dither signal added to the RF signal has successfully mitigated the effect of the quantization error caused by the comparator, and the per-symbol MSE is improved to $-28dB$.

In Figure 4.5, the per-symbol MSE values are shown as a function of the ratio between the power of the dither signal and the power of the RF signal. It is clear from the results that the minimum MSE value is achieved when the ratio between the

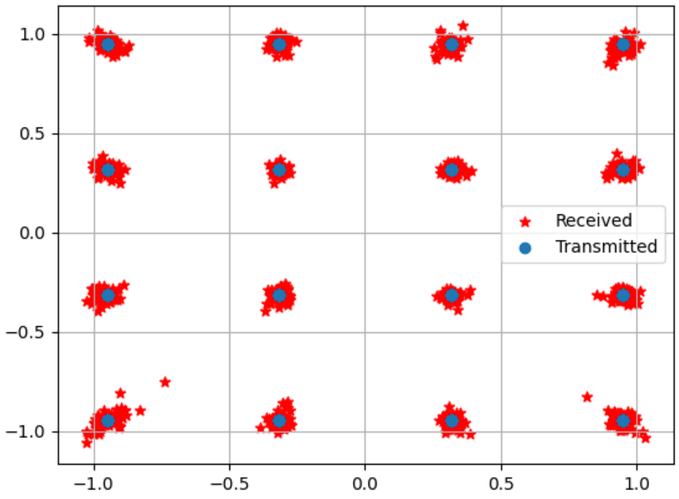

Figure 4.4: Constellation plot showing transmitted and received symbols after passing through an AWGN channel with SNR $= 10dB$ and one-bit quantization with dithering.

power of the dither signal and the power of the RF signal is $P_D/P_{RF} = 5dB$.

4.2 The Results of the FBM-CENet and FBM-DetNet

4.2.1 Channel Estimation

In this section, the results of using the deep neural network described in Section 2.2.3.2 for channel estimation are compared to the Bussgang-based channel estimation calculated in Section 2.2.3.1. The comparison is made in terms of normalized mean squared error (NMSE), which is defined as NMSE $= \mathbb{E}\left[||\boldsymbol{H} - \hat{\boldsymbol{H}}||^2\right] / \mathbb{E}\left[||\boldsymbol{H}||^2\right]$, where $\hat{\boldsymbol{H}}$ is the estimation of the channel $\boldsymbol{H}$.

The estimated channel $\hat{\boldsymbol{h}} = \boldsymbol{h}^{(L)}$ is the output of the network's last layer and represents the vectorized form of $\hat{\boldsymbol{H}}$. The loss function used to train the network is $||\boldsymbol{h} - \hat{\boldsymbol{h}}||^2$, where $||.||$ denotes the l_2-norm.

To train the network, a pilot matrix is used which is assumed to contain U columns of a DFT matrix of size $T \times T$. The channel $\boldsymbol{H}$ is considered to be a Rayleigh fading channel defined as $\boldsymbol{H}_k \sim \mathcal{CN}(0, \boldsymbol{I_B})$. It is assumed that there is a perfect transmit power control at users, ensuring that the received signal power of all users is the same and equal to one. In this case, the SNR is given by $\rho = 1/N_0$, where N_0 is the noise power.

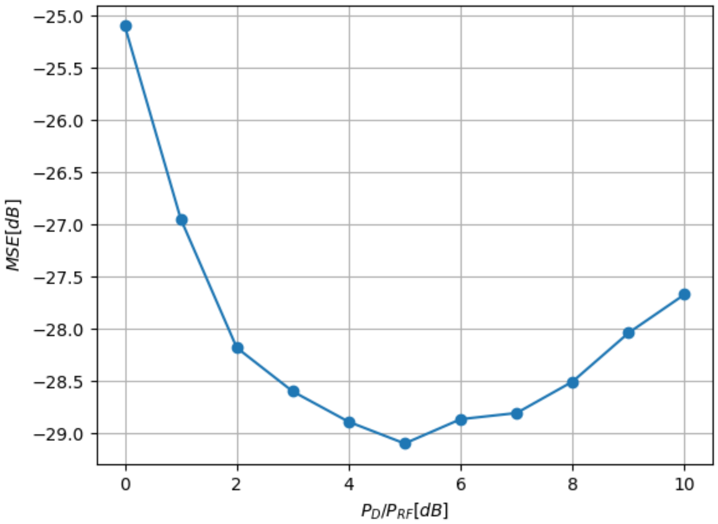

Figure 4.5: The per-symbol MSE values as a function of the ratio between the power of the dither signal and the power of the RF signal.

To train the network, we use the Adam optimizer with an initial learning rate of 0.002, which decays by a factor of 0.97 every 100 epochs. The network's first layer input is initialized to zero, and the pilot matrix length is set to be five times the number of users ($T = 5U$). Initial values for α_1 to α_8 are set to 0.01, and β is set to 5. The training is conducted with a batch size of 1000, utilizing mini-batches of 200 samples. The process runs for 300 epochs and is carried out using TensorFlow.

In Figure 4.6, the obtained NMSE values is shown for both the Bussgang-based and deep neural network channel estimation as a function of the SNR. The network is considered to have $L = 8$ layers, and the number of transmitting users and receiving antennas are set to $U = 4$ and $B = 32$, respectively.

It can be seen from figure 4.6 that the FBM-CENet outperforms the BMMSE method. Additionally, in the BMMSE method, a perfect knowledge of the channel correlation is used, which in practice may not be available. However, the FBM-DetNet gives very good results without relying on this correlation information.

Considering that the Bussgang decomposition method models a non-linear system using a Gaussian distribution, the BMMSE method is more accurate when the received signal is Gaussian, implying high noise power and low SNR. However, as SNR increases, BMMSE's performance becomes more limited. In contrast, FBM-DETNet has no such limitations and learns the non-linear model by extracting its features for each SNR, thereby achieving superior performance compared to BMMSE.

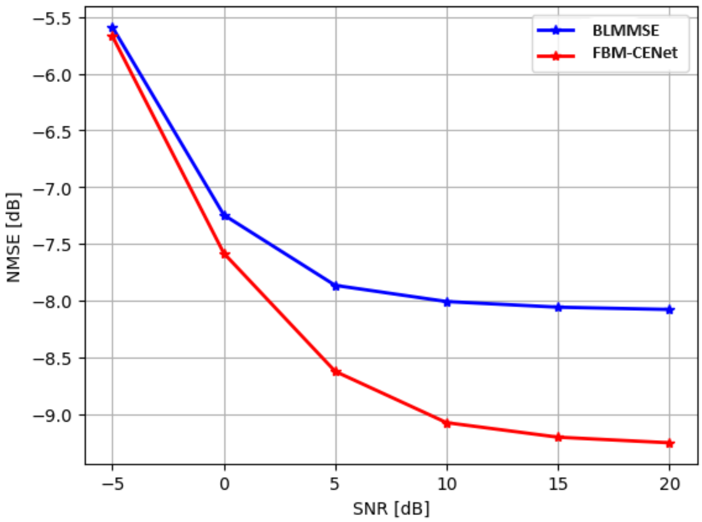

Figure 4.6: The NMSE as a function of the SNR.

4.2.2 Data Detection

To compare the performance of the Bussgang-based data detector and FBM-DetNet, each user sends 16-QAM signals. Here, $U = 4$ users and $B = 64$ receiving antennas are considered, and the $L = 8$ layers are used in the FBM-DetNet structure. The comparison is made in terms of the bit error rate (BER), and the loss function used to train the network is $||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$, where $\hat{\boldsymbol{x}} = \boldsymbol{x}^{(L)}$ is the detected data in the receiver and $\boldsymbol{x}$ is the transmitted data.

After generating the channel matrix $\boldsymbol{H}$ from a Rayleigh distribution defined as $\boldsymbol{H} \sim \mathcal{CN}(0, \boldsymbol{I_B})$, the estimated channel $\hat{\boldsymbol{H}}$ is obtained using the FBM-CENet. This estimated channel $\hat{\boldsymbol{H}}$ is then used as channel state information (CSI) for data detection.

The initial learning rate is set to 0.002, which decays by a factor of 0.97 every 100 epochs. The network's first layer input is initialized to zero. Initial values for α_1 to α_8 are set to 0.1, t_1 to t_8 are set to 1, and β is set to 5. The training is conducted with a batch size of 1000, and the number of epochs is 1000.

In Figure 4.7, the obtained BER is depicted for both Bussgang-based data detection and the FBM-DetNet structure described in Section 2.2.4.2 as a function of the SNR. As illustrated, the deep learning network achieves a significantly lower BER at higher SNRs, outperforming the BMMSE data detection method. Additionally, it can be observed that when the signal exhibits more Gaussian behavior, which occurs at low SNRs, the BMMSE method yields slightly better results due to its modeling of non-linear transforms using a Gaussian distribution.

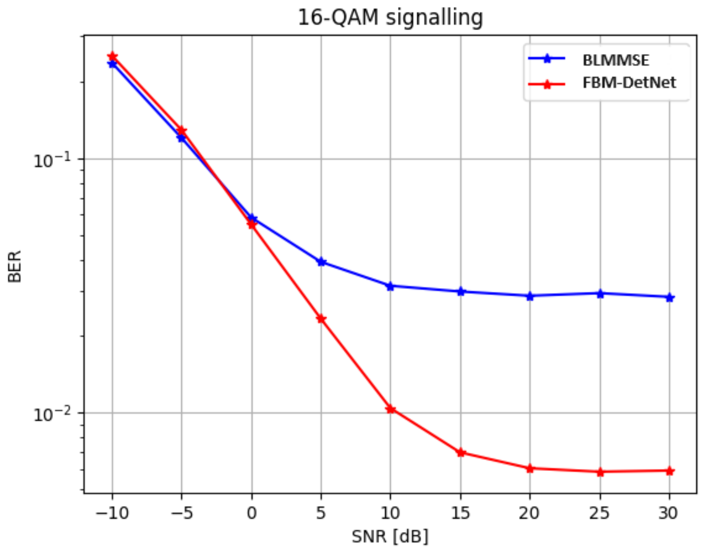

Figure 4.7: BER with as a function of SNR for data detection.

4.3 Deep-Learning-Based Channel Estimation and Data Detection for Distributed MIMO With One-Bit Radio-Over-Fiber Fronthaul

4.3.1 Channel Estimation

In this section, we present the results of the channel estimation network, as detailed in Section 3.3, with $U = 1$ and $B = 1$, for the distributed MIMO model depicted in Figure 2.2 during the uplink phase. These results are compared to those obtained using the Bussgang-based channel estimation method.

A single-carrier pilot signal centered at $f_c = 2.4GHz$, consisting of $N_p = 10$ symbols with power E_s, is transmitted to the AP with a bandwidth of $BW = 240MHz$, and a sampling frequency of $f_s = 10GHz$. The channel length in the frequency domain is considered to be $S = 9$ and is generated from the distribution $\boldsymbol{h} \sim \mathcal{CN}(0, I_S)$. The length of the upsampled signal is $N = 189$.

The DNN used for channel estimation consists of $L = 5$ layers. The loss function used for channel estimation the network is $||\boldsymbol{h} - \hat{\boldsymbol{h}}||^2$, where $\hat{\boldsymbol{h}} = \boldsymbol{h}^{(L)}$. The initial values for the parameters α_1 to α_5 are set to 0.01, and β_1 to β_5 are set to 5. Additionally, the initial input to the DNN is set to zero.

The network is trained using the Adam optimizer with an initial learning rate of 0.002, which decays by a factor of 0.97 every 100 epochs. The training process consists of 1000 epochs, and the size of each training batch is set to 1000. To evaluate the performance of the network, we use NMSE averaged over 5000 channel realizations.

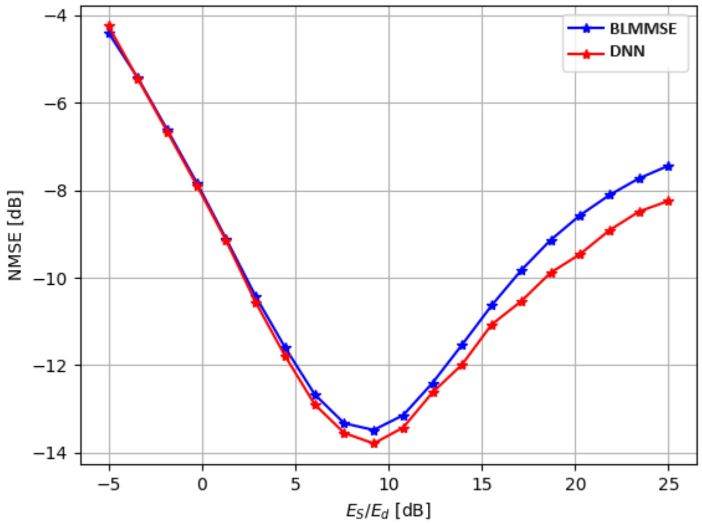

Figure 4.8: The changes in NMSE with respect to the ratio of the RF signal power to the dither signal power.

In Figure 4.8, the NMSE is shown as a function of the ratio of the power of the RF signal to the power of the dither signal, denoted as E_s/E_d. It is evident from the results that both channel estimation methods perform almost equally well when the E_s/E_d ratio is smaller, indicating more Gaussian behavior. However, as this ratio increases, the deep neural network (DNN) estimator performs more successfully compared to the BMMSE-based estimator, resulting in a smaller NMSE.

Furthermore, the results indicate that the optimal E_s/E_d is around $E_s/E_d = 9.2dB$. At this ratio of RF signal power to dither signal power, both methods have similar performance, with the DNN estimator performing slightly better. However, the DNN estimator does not require prior knowledge about the distribution of the channel realizations.

In Figure 4.9, NMSE is depicted as a function of the SNR using the optimal value for E_s/E_d shown in Figure 4.8. When the SNR is low, both DNN and Bussgang-based estimators perform equally well.

Although the model-driven DNN estimator does not account for the effect of noise, as illustrated in Figure 4.9, at larger SNRs, the results from the DNN estimator match the results in Figure 4.8, where the network is trained in absence of noise. This indicates that in presence of small noise, the performance of the DNN estimator does not degrade and, as illustrated in Figure 4.9, it slightly outperforms the Bussgang-based estimator. Again, it is noteworthy to mention that the DNN estimator learns the channel behavior without requiring any prior information about its statistical characteristics.

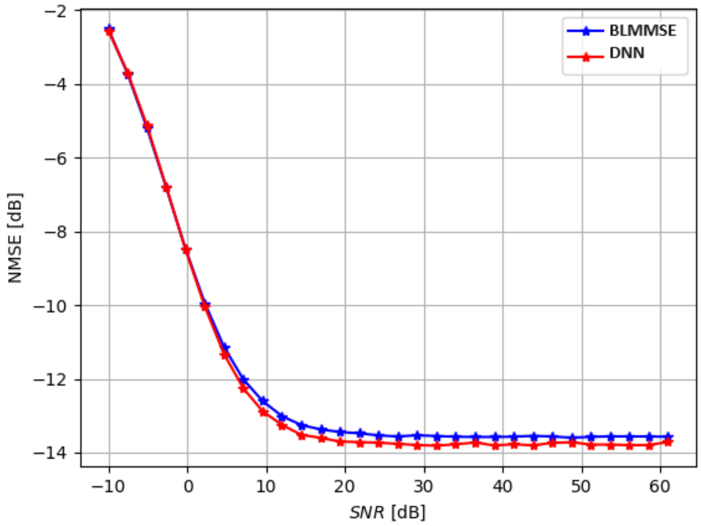

Figure 4.9: Changes in NMSE with respect to the SNR using the optimal $E_s/E_d = 9.2dB$.

4.3.2 Data Detection

In this section, we present the results of the data detection network, as detailed in Section 3.4, with $U = 1$ and $B = 1$, for the distributed MIMO model depicted in Figure 2.2 during the uplink phase. These results are compared to those obtained using the Bussgang-based channel estimation method.

First, using the channel estimation neural network, we estimate the channel with the same setup as in Section 4.0.3.1, which is then used for data detection.

The DNN used for data detection consists of $L = 5$ layers. The loss function is $||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2$, where $\hat{\boldsymbol{x}} = \boldsymbol{x}^{(L)}$ and $\boldsymbol{x} \in \mathcal{M}$, where $\mathcal{M}$ representing a 16-QAM constellation. The initial values for the parameters α_1 to α_5 are set to 1, β_1 to β_5 are set to 5, and t_1 to t_5 are set to 1. Additionally, the initial input to the DNN is set to zero.

The network is trained using the Adam optimizer with an initial learning rate of 0.002, which decays by a factor of 0.97 every 100 epochs. The training process consists of 2000 epochs, and the size of each training batch is set to 1000. To evaluate the performance of the network, we use mean squared error (MSE) and BER averaged over 5000 channel realizations.

The DNN is trained twice. First, we consider the perfect CSI case. In this case, we assume that the receiver has complete knowledge of the channel, and the channel matrix is directly used for data detection. In the second scenario, the receiver uses an estimation of the channel to detect the received data. Figure 4.10 illustrates the BER as a function of E_s/E_d for these cases.

Considering the results, it is evident that the DNN detector outperforms the BLMMSE method in both cases. However, the advantage of using the DNN is more significant when the estimated channel is used in the receiver for data detection This scenario is closer to real-world cases where the channel is unknown to the receiver.

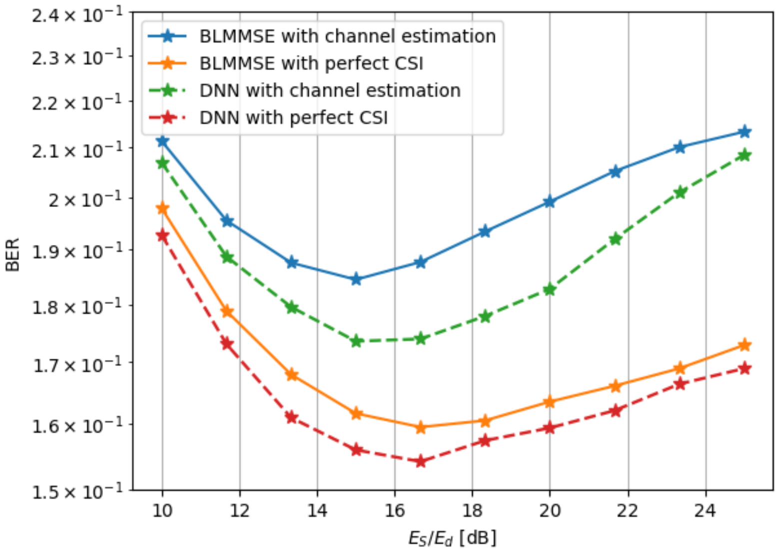

Figure 4.10: BER as a function of the ratio of the RF signal power to the dither signal power.

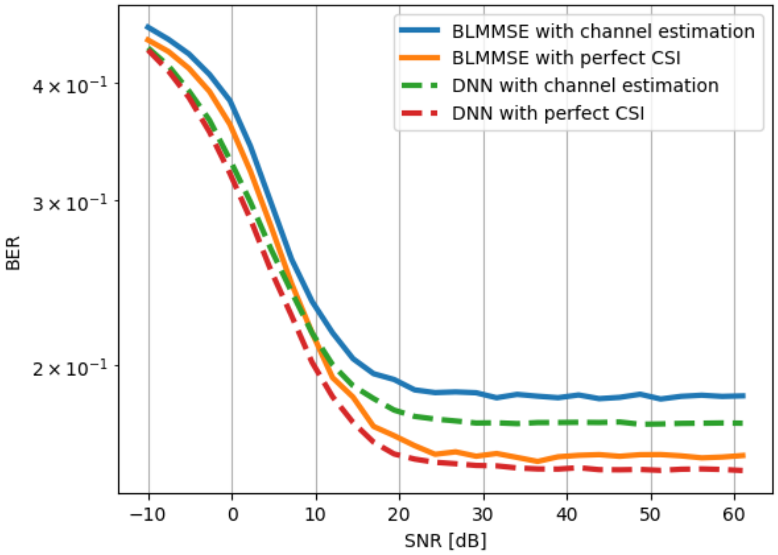

Figure 4.11: BER as a function of the SNR using the optimal value for E_s/E_d.

In Figure 4.11, the BER is illustrated as a function of the SNR, using the optimal value for the power ratio E_s/E_d as shown in Figure 4.10. It is evident that the DNN outperforms the BLMMSE detectors in both cases, and again, a larger gap can be seen between the DNN and BLMMSE detectors in the case where the channel is unknown to the receiver. Considering that the DNN does not account for the effect of the AWGN noise, at higher SNRs, the same BER is achieved as in Figure 4.10 with the optimal power ratio values when the noise is absent. This indicates that a low amount of noise does not degrade the performance of the data detection network.

5

Conclusion

In this thesis, we began by studying a transceiver architecture for distributed massive MIMO systems proposed in [5], with the primary goal of ensuring phase coherence by applying direct RF sampling at the CU instead of using local oscillators at each RRH. This architecture utilizes low-resolution ADCs/DACs to transmit 1-bit digital signals over optical fibers between each AP and the CU. In the downlink, a fourth-order pass-band sigma-delta modulator is employed to shift the quantization error to higher frequencies, thereby reducing the quantization error within the bandwidth of interest. In the uplink, a dither signal is added to the received RF signal at the AP to evenly distribute the quantization error and decorrelate it from the received signal by introducing randomness. Based on the simulation results, we observed that both techniques used to mitigate the quantization error caused by the low-resolution ADCs/DACs enhance the system's performance in both uplink and downlink.

We continued by studying two deep neural networks proposed in [13] used for channel estimation and data detection by unfolding each iteration in the process of ML estimation. For this purpose, the gradient of the ML optimizer is calculated to be used in the gradient ascent/descent algorithm. The channel estimation network, FBM-CENet, and the data detection network, FBM-DetNet, were compared to the results from the Bussgang-based channel estimation and data detection methods in terms of NMSE and BER, respectively. From this comparison, we observed that the DNN estimators noticeably outperforms the results from the BLMMSE estimators. Additionally, BLMMSE estimators require prior information about the distribution from which the channel and data are generated. In contrast, DNN estimators can operate without any pre-existing information, learning the patterns and characteristics directly from the data during the training process.

In the next step, a DNN derived from [18] for channel estimation in the uplink phase of the system described in [5] was studied in the presence of a dither signal randomly generated from a normal distribution. In this DNN, the effect of AWGN noise was not considered to avoid complexity. The network was tested with various ratios between the RF signal power and the dither signal power, and these results were compared to those of the BLMMSE estimator. The optimal power ratio was selected for testing the DNN channel estimator in the presence of AWGN noise. The results indicated that, despite the AWGN not being considered in the network computations, the channel estimation performance was not adversely affected by the AWGN. The DNN channel estimator demonstrated the same performance as the BLMMSE estimator in terms of NMSE at lower SNRs, but slightly outperformed the BLMMSE estimator at higher SNRs.

As a final step, a DNN was developed using the same method as in [13] and [18] for data detection in the uplink phase of the system described in [5]. The network was trained for two different scenarios. In the first scenario, perfect CSI was assumed, meaning the channel is completely known to the receiver and directly used for data detection. In the second scenario, the estimated channel from the channel estimator was used for data detection. The results demonstrate that the DNN data detector outperforms the BLMMSE detector in both cases. However, the advantage of using the DNN was more significant when the estimated channel was used for data detection.

Using the optimal power ratios in each case, the BLMMSE and DNN detectors were tested in the presence of AWGN noise. The results showed that the DNN data detector exhibited superior performance in both cases, with an even greater advantage when the estimated channel was used for data detection. Additionally, the results at higher SNRs matched those obtained in the absence of noise, confirming the effectiveness of the DNN detector even without accounting for noise in its network structure when exposed to a small amount of noise.

In this study, we considered a random dither signal generated from a normal distribution. Given the significant role of dithering in the performance of the distributed massive MIMO system described in [5], future work should evaluate different types of dither signals. Based on the statistical characteristics of the chosen dither signal, the channel estimation and data detection DNNs should be updated and further analyzed.

Bibliography

[1] M. Reza Khanzadi, Giuseppe Durisi, and Thomas Eriksson. “Capacity of SIMO and MISO Phase-Noise Channels With Common/Separate Oscillators”. In: *IEEE Transactions on Communications* 63.9 (2015), pp. 3218–3231. DOI: `10.1109/TCOMM.2015.2408605`.

[2] Emil Björnson and Luca Sanguinetti. “Making Cell-Free Massive MIMO Competitive With MMSE Processing and Centralized Implementation”. In: *IEEE Transactions on Wireless Communications* 19.1 (2020), pp. 77–90. DOI: `10.1109/TWC.2019.2941478`.

[3] Laurens Breyne et al. “Comparison Between Analog Radio-Over-Fiber and Sigma Delta Modulated Radio-Over-Fiber”. In: *IEEE Photonics Technology Letters* 29.21 (2017), pp. 1808–1811. DOI: `10.1109/LPT.2017.2752284`.

[4] Morriel Kasher, Predrag Spasojevic, and Michael Tinston. “Linearization of Non-Uniform Quantizers via Adaptive Non-Subtractive Dithering”. In: *2023 57th Annual Conference on Information Sciences and Systems (CISS)*. 2023, pp. 1–6. DOI: `10.1109/CISS56502.2023.10089625`.

[5] Lise Aabel et al. “Distributed Massive MIMO via all-Digital Radio Over Fiber”. In: *2020 54th Asilomar Conference on Signals, Systems, and Computers*. 2020, pp. 319–323. DOI: `10.1109/IEEECONF51394.2020.9443347`.

[6] Mingqiang Guo et al. “A 5 GS/s 29 mW Interleaved SAR ADC With 48.5 dB SNDR Using Digital-Mixing Background Timing-Skew Calibration for Direct Sampling Applications”. In: *IEEE Access* 8 (2020), pp. 138944–138954. DOI: `10.1109/ACCESS.2020.3012699`.

[7] Ly V. Nguyen, A. Lee Swindlehurst, and Duy H. N. Nguyen. “Linear and Deep Neural Network-Based Receivers for Massive MIMO Systems With One-Bit ADCs”. In: *IEEE Transactions on Wireless Communications* 20.11 (2021), pp. 7333–7345. DOI: `10.1109/TWC.2021.3082844`.

[8] Junil Choi, Jianhua Mo, and Robert W. Heath. “Near Maximum-Likelihood Detector and Channel Estimator for Uplink Multiuser Massive MIMO Systems With One-Bit ADCs”. In: *IEEE Transactions on Communications* 64.5 (2016), pp. 2005–2018. DOI: `10.1109/TCOMM.2016.2545666`.

[9] Chao-Kai Wen et al. “Bayes-Optimal Joint Channel-and-Data Estimation for Massive MIMO With Low-Precision ADCs”. In: *IEEE Transactions on Signal Processing* 64.10 (2016), pp. 2541–2556. DOI: `10.1109/TSP.2015.2508786`.

[10] Yongzhi Li et al. “Channel Estimation and Performance Analysis of One-Bit Massive MIMO Systems”. In: *IEEE Transactions on Signal Processing* 65.15 (2017), pp. 4075–4089. DOI: `10.1109/TSP.2017.2706179`.

[11] Ly V. Nguyen, Duy H. N. Nguyen, and A. Lee Swindlehurs. "SVM-based Channel Estimation and Data Detection for Massive MIMO Systems with One-Bit ADCs". In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC).* 2020, pp. 1–6. DOI: `10.1109/ICC40277.2020.9148630`.

[12] Shahin Khobahi et al. "LoRD-Net: Unfolded Deep Detection Network With Low-Resolution Receivers". In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 5651–5664. DOI: `10.1109/TSP.2021.3117503`.

[13] Ly V. Nguyen, Duy H. N. Nguyen, and A. Lee Swindlehurst. "Deep Learning for Estimation and Pilot Signal Design in Few-Bit Massive MIMO Systems". In: *IEEE Transactions on Wireless Communications* 22.1 (2023), pp. 379–392. DOI: `10.1109/TWC.2022.3193885`.

[14] Jose M. de la Rosa. "Bandpass Sigma–Delta Modulation: The Path toward RF-to-Digital Conversion in Software-Defined Radio". In: *Chips* 2.1 (2023), pp. 44–69. ISSN: 2674-0729. DOI: `10.3390/chips2010004`. URL: `https://www.mdpi.com/2674-0729/2/1/4`.

[15] Anzhong Hu et al. "EVM Analysis of Distributed Massive MIMO with 1-Bit Radio-Over-Fiber Fronthaul". In: *IEEE Transactions on Communications* (2024), pp. 1–1. DOI: `10.1109/TCOMM.2024.3412769`.

[16] Kai Zhang, Luc Van Gool, and Radu Timofte. "Deep Unfolding Network for Image Super-Resolution". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 3214–3223. DOI: `10.1109/CVPR42600.2020.00328`.

[17] Akshay Agrawal, Shane Barratt, and Stephen Boyd. "Learning Convex Optimization Models". In: *IEEE/CAA Journal of Automatica Sinica* 8.8 (2021), pp. 1355–1364. DOI: `10.1109/JAS.2021.1004075`.

[18] Alireza Bordbar et al. "Deep-Learning-Based Channel Estimation for Distributed MIMO with 1-bit Radio-Over-Fiber Fronthaul". In: (2024). Available at `https://doi.org/10.48550/arXiv.2406.11325`. arXiv: `2406.11325 [eess.SP]`.

[19] J.H. Van Vleck and D. Middleton. "The spectrum of clipped noise". In: *Proceedings of the IEEE* 54.1 (1966), pp. 2–19. DOI: `10.1109/PROC.1966.4567`.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY