UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RAFAEL MENDES CAMPELLO

MACHINE LEARNING APPLICATIONS IN COMMUNICATION SYSTEMS DECODING

Recife

2022

RAFAEL MENDES CAMPELLO

MACHINE LEARNING APPLICATIONS IN COMMUNICATION SYSTEMS DECODING

Dissertation presented to the Graduate Program in Electrical Engineering at the Federal University of Pernambuco, Centro de Tecnologia e Geociências, as a partial requirement for obtaining a master's degree in Electrical Engineering.

Concentration Area: Communications.

Supervisor: Prof. Dr. Cecilio José Lins Pimentel

Co-supervisor: Prof. Dr. Daniel Pedro Bezerra Chaves

Recife

2022

Catalogação na fonte:
Bibliotecária Sandra Maria Neri Santiago, CRB-4 / 1267

C193m Campello, Rafael Mendes.
Machine learning applications in communication systems decoding / Rafael Mendes Campello. – 2022.
78 f.: il., figs., tabs., abrev. e sigl.

Orientador: Prof. Dr. Cecilio José Lins Pimentel.
Coorientador: Prof. Dr. Daniel Pedro Bezerra Chaves.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, Recife, 2022.
Inclui referências.

1. Engenharia elétrica. 2. Aprendizagem de máquina. 3. Aprendizagem profunda. 4. Comunicação. 5. Caótica. 6. Códigos corretores de erro. 7. Acesso múltiplo não-ortogonal. I. Pimentel, Cecilio José Lins (Orientador). II. Chaves, Daniel Pedro Bezerra (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2022-156

RAFAEL MENDES CAMPELLO

MACHINE LEARNING APPLICATIONS IN COMMUNICATION SYSTEMS DECODING

Dissertation presented to the Graduate Program in Electrical Engineering at the Federal University of Pernambuco, Centro de Tecnologia e Geociências, as a partial requirement for obtaining a master's degree in Electrical Engineering.

Concentration Area: Communications.

Approved in: February 11th, 2022.

EXAMINATION BOARD

Prof. Dr. Cecilio José Lins Pimentel (Supervisor and Internal Examiner)
Universidade Federal de Pernambuco

Prof. Dr. Daniel Pedro Bezerra Chaves (Co-supervisor and Internal Examiner)
Universidade Federal de Pernambuco

Prof. Dr. José Sampaio de Lemos Neto (External Examiner)
Universidade Federal de Pernambuco

Prof. Dr. Tarcisio Ferreira Maciel (External Examiner)
Universidade Federal do Ceará

ACKNOWLEDGEMENTS

During the development of this research I was fortunate to had assistance from professors and colleagues, and I would like to acknowledge them here. I would like to thank my professors and supervisors, Prof. Cecilio Pimentel and Prof. Daniel Chaves, for the frequent guidance during the whole period of my M. Sc. course, including their help in project ideas and papers writing. Thanks to Dr. Carlos Souza and Gabriel Carlini for the help during the realization of different parts of this research. Thanks to my family, for the daily support during my whole life. Finally, thanks to the National Council for Scientific and Technological Development (CNPq) for the financial support throughout the development of this research.

ABSTRACT

The usage of machine learning (ML) techniques in different academic and professional fields confirms its theoretical and practical utility. The communications field is no exception. In fact, models that learn from data were already in use prior to the recent advancement in the ML field. This research investigates different kinds of usage that can be done with ML models in three different problems, seeking to show their high flexibility and to present alternative ways of obtaining classical results which employ well established algorithms, or even outperform them in some scenarios. The first problem discusses the so-called Markov-Gaussian channels and compares an ML model with the already common hidden Markov models approach. The second problem deals with non-orthogonal multiple access transmissions and compares an ML model with the usually employed decoding algorithm. The third presents a chaos-based communication system and compares the maximum likelihood decoding to a neural network-based one.

Keywords: machine learning; deep learning; chaos communication; error correcting codes; non-orthogonal multiple access.

RESUMO

O uso de técnicas de aprendizagem de máquina em diferentes campos acadêmicos e profissionais confirma sua utilidade teórica e prática. O campo de comunicações não é exceção, possuindo diversas aplicações em problemas estabelecidos. Este trabalho faz uma investigação de diferentes formas de utilizar modelos baseados em aprendizagem de máquina em três problemas distintos envolvendo decodificação em sistemas de comunicação, buscando demonstrar sua flexibilidade e apresentar formas alternativas de obter resultados clássicos que empregam algoritmos estabelecidos, ou até mesmo obter desempenhos melhores em situações específicas. O primeiro problema trata de canais Markov-Gauss e compara um modelo de aprendizagem de máquina com o modelo oculto de Markov usualmente empregado, o segundo trata de sistemas de comunicação baseados em acesso múltiplo não-ortogonal e compara um modelo de aprendizagem com o algoritmo de decodificação usualmente empregado e o terceiro trata de um sistema de comunicação baseado em teoria do caos em que uma rede neural é utilizada na decodificação em comparação com a decodificação por máxima verossimilhança.

Palavras-chave: aprendizagem de máquina; aprendizagem profunda; comunicação caótica; códigos corretores de erro; acesso múltiplo não-ortogonal.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv Algorithm
BS	Base Station
BWA	Baum-Welch Algorithm
CB-STTC	Chaos-Based Space-Time Trellis Code
CEM	Channel Estimation Module
CSI	Channel State Information
DL	Deep Learning
FBA	Forward-Backward Algorithm
FDMA	Frequency-Division Multiple Access
FER	Frame Error Rate
FSE	Finite-State Encoder
GRU	Gate Recurrent Unit
HMM	Hidden Markov Model
KSMLD	Known-State Maximum Likelihood Decoder
LDPC	Low-Density Parity-Check
LLR	Log-Likelihood Ratio
MLD	Maximum Likelihood Decoding
ML	Machine Learning
MSE	Mean Squared Error
NN	Neural Network
NOMA	Non-Orthogonal Multiple Access
OFDMA	Orthogonal Frequency-Division Multiple Access
OMA	Orthogonal Multiple Access
PDF	Probability Density Function
QPSK	Quadrature Phase Shift Keying
RNN	Recurrent Neural Network
SER	Symbol Error Rate
SIC	Successive Interference Cancellation
SISO	Soft-Input Soft-Output

SNR	Signal to Noise Ratio
TDMA	Time-Division Multiple Access

LIST OF SYMBOLS

Γ	Markov-Gaussian channel parameters
$\boldsymbol{\Omega}$	Hidden Markov model parameters
P	Markov chain state transition probability matrix
μ	Markov-Gaussian channel memory
κ	Markov-Gaussian variance relation between states
$\gamma_i(p)$	Probability of being in state p given channel noise realization
σ_s^2	State s variance of conditional Gaussian PDF
ζ	Chapter 2 signal to noise ratio
I	Interleaver depth
l_r	NN learning rate
n_c	Codeword length
k_c	Number of information bits
z_c	Protograph-LDPC expansion factor
$\boldsymbol{n}$	Noise vector
$\boldsymbol{c}$	Codeword
$\boldsymbol{S}$	Markov-Gauss state set
$b_p(n_i)$	Markov-Gauss conditional PDF (2)
L	Log-likelihood ratio
N_0	Spectral noise density
U_i	User i
Λ	Batch size
K	Total energy restriction used in NOMA SER analysis
h_i	NOMA channel gain multiplicative term for user i
ξ_i	Spacing between users multiplicative terms: $h_i^2 = h_{i+1}^2 + \xi_i$ (dB)
B	Scale factor for NOMA symbols axes, $B = \frac{\sqrt{2}}{2}$
Q	Tail distribution function of the standard normal distribution. $Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-t^2/2}\, dt$
R	Code rate
J	Cost function
ρ	Chapter 4 signal to noise ratio

N_T	Number of transmitter antennas in Chapter 4
N_R	Number of receive antennas in Chapter 4
$\boldsymbol{v}$	Target vector for Chapter 4 NN training
ℓ	Length of vector $\boldsymbol{v}$ in Chapter 4
$\omega_{\beta\gamma}^{\alpha}$	Chaotic waveform associated with region α of the Poincaré section, $\beta\gamma$ indicates subsequent visited regions

CONTENTS

1 INTRODUCTION

Although an old subject, the machine learning (ML) field has been revitalized by the recent deep learning (DL) improvements. Newly developed methods have brought significant advances to a wide range of fields and tasks, such as, computer vision and natural language processing (LECUN; BENGIO; HINTON, 2015), mastering games of go and chess (SILVER et al., 2016; SILVER et al., 2018), predicting diseases (SIDEY-GIBBONS; SIDEY-GIBBONS, 2019). In the communications field, important tasks have already been tackled by recent works (HE et al., 2018; NEUMANN; WIESE; UTSCHICK, 2018; SOLTANI et al., 2019; JIANG; STRUFE; SCHOTTEN, 2020; NACHMANI et al., 2017; BE'ERY et al., 2020; BENNATAN; CHOUKROUN; KISILEV, 2018; KIM et al., 2018; HUANG et al., 2020).

ML applications in the communication field include channel estimation (HE et al., 2018; NEUMANN; WIESE; UTSCHICK, 2018; SOLTANI et al., 2019; JIANG; STRUFE; SCHOTTEN, 2020), channel decoding (NACHMANI et al., 2017; BE'ERY et al., 2020; BENNATAN; CHOUKROUN; KISILEV, 2018; KIM et al., 2018), fifth generation (5G) applications on the physical layer (HUANG et al., 2020), discovery of new error correcting codes and decoding algorithms (JIANG et al., 2019b). In (KIM et al., 2018), ML is used seeking to automate the discovery of decoding algorithms for error correcting codes, for such, recurrent neural network (RNN) architecture is used to decode convolutional and turbo codes. Similar works focusing on channel decoding are done in (LIANG; SHEN; WU, 2018; BALATSOUKAS-STIMMING; STUDER, 2019). In (WU et al., 2019), ML is used as an end-to-end autoencoder architecture capable of jointly performing the tasks of encoding, modulation, demodulation and decoding under various system conditions. The results suggests that the model can match classical modulations and encoding designs for additive white Gaussian noise (AWGN) channels, while outperforming existing schemes for non-standard bursty channels. Similar works on autoencoder are done in (JIANG et al., 2019c; DÖRNER et al., 2018; O'SHEA; KARRA; CLANCY, 2016; JIANG et al., 2019a; LETIZIA; TONELLO, 2020).

ML techniques can also be employed in specialized problems. In (CHU et al., 2021; WU; JIANG; ZHAO, 2018; LUGOSCH; GROSS, 2017), low-density parity-check (LDPC) decoding is optimized for certain scenarios aided by learning techniques. In (DOAN et al., 2019), a neural belief propagation algorithm is proposed for polar concatenated codes. In (CHIEN; BJORNSON; LARSSON, 2019), benefits from ML are investigated in the task of spectral efficiency maximization in MIMO systems.

The variety of applications shows that the data learning approach is highly flexible, even within the communication domain, and that different techniques can be used for each setting. In this work, three different scenarios are presented, each employing some sort of learning approach to tackle its problems. Therefore, the following chapters can be viewed as a grouping of three unrelated problems, bounded by the presence of ML techniques in the communication field. This choice of presentation seeks to evaluate different possibilities of

application of the ML toolset on different scenarios.

The first presented scenario deals with LDPC decoding under a channel with memory, the Markov-Gauss channel. In this setting, a bursty noise channel is modelled by a Markov process in which the noise variance is determined by the state of a Markov chain. Being a variant of the Gilbert-Elliot channel (GILBERT, 1960; ELLIOTT, 1963) in the literature, this scenario traditionally employs a hidden Markov model (HMM) approach in its decoding (FERTONANI; COLAVOLPE, 2009). In this work, we also investigate the usage of a more modern approach with neural networks (NN) aiming to replicate and analyze the impact of this choice in this problem setting.

The second presented scenario deals with an important technique for future communications technologies, the non-orthogonal multiple access (NOMA) transmissions, in which multiple users share the same channel resources, accepting controlled inter-user interference while seeking to obtain higher data rates and acceptable error performance. In this work, we analyze a possible modification to the traditionally employed successive interference cancellation decoding, which can be obtained by the usage of decision tree models, and also compare it to the NN-based decoding present in the literature.

The third presented scenario presents a chaos-based communication system. In this setting a chaotic attractor is used to define a communication system in which chaotic waveforms are employed in transmission, additionally the attractor possesses dynamics restrictions which are employed to devise a transmitter encoder. Due to its chaotic nature, traditional decoding strategies might be difficult or inadequate in this setting, which motivates the use of ML techniques aimed at jointly adapting to attractor waveforms and dynamics.

For all presented results in this research, we employ Python 3.8, Tensorflow 2.1 and Google Colab as computational tools.

1.1 ORGANIZATION

The remaining of the dissertation is organized in the following chapters. The chapters are organized as self-contained works which do not necessarily share connections with each other. Therefore each one starts with an introduction of its own topics and ends with a discussion of its contents.

- Chapter 2 discusses the Markov-Gaussian channel, a type of channel in which the noise process can be modelled by a Markov chain and its well-suited to ML models assisted decoding, such as HMM and NN. In this scenario, the noise process memory is captured by the ML models and aids the decoding by combining the memory information with error correcting code capabilities. In practice, the NN model performs at least as well as the traditionally employed HMM for presented tasks.

- Chapter 3 discusses NOMA, a transmission scheme that has gained attention for use in the next generation of communication systems. In this scheme multiple users share the same resources without the need to maintain an orthogonality condition between them. A decoding algorithm is proposed for high interference use cases. We provide a theoretical analysis and compare the results to the standard employed decoding and ML methods.
- Chapter 4 presents a chaos-based space-time trellis code (CB-STTC) scheme based on chaotic attractors. This code is defined by a finite state encoder that maps information sequences to restricted sequences satisfying the dynamics of the attactor. We also propose a NN architecture capable of learning how to decode the CB-STTC.

1.2 PUBLISHED RESEARCH

Some results obtained during the course of this research were published in co-authorship:

- Journal
  - SOUZA, CARLOS E. C.; CAMPELLO, RAFAEL; PIMENTEL, CECILIO; CHAVES, DANIEL P. B., Chaos-based space-time trellis codes with deep learning decoding. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-EXPRESS BRIEFS, v. 68, p. 1472-1476, apr. 2021. DOI: 10.1109/TCSII.2020.3038481.
  - CAMPELLO, RAFAEL; CARLINI, GABRIEL; SOUZA, CARLOS E. C.; PIMENTEL, CECILIO; CHAVES, DANIEL P. B., Successive interference cancellation decoding with adaptable decision regions for NOMA schemes. IEEE Access, v. 10, p. 2051-2062, dec. 2021. DOI: 10.1109/ACCESS.2021.3137994.
- Conference
  - SOUZA, CARLOS; CAMPELLO, RAFAEL; PIMENTEL, CECILIO; CHAVES, DANIEL. Esquemas de codificação espácio-temporal baseados em caos com decodificação por aprendizagem profunda. In: XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais. nov. 2020. DOI: 10.14209/SBRT.2020.1570655166.
  - CAMPELLO, RAFAEL; CARLINI, GABRIEL; SOUZA, CARLOS EDUARDO; PIMENTEL, CECILIO; CHAVES, DANIEL P B. Decodificação via cancelamento sucessivo de interferência generalizado para sistemas de acesso múltiplo não ortogonal. In: XXXIX Simpósio Brasileiro de Telecomunicações e Processamento de Sinais. pp. 1-5, oct. 2021. DOI: 10.14209/sbrt.2021.1570726340.

2 MARKOV-GAUSSIAN CHANNEL

The Markov-Gaussian channel has been used to model impulsive interfence channels with memory (FERTONANI; COLAVOLPE, 2009; MITRA; LAMPE, 2009; MITRA; LAMPE, 2010; AHMED et al., 2020). The noise process is described by a mixture-Gaussian process where the conditional probability density function (PDF) of each noise sample is Gaussian with variance that depends on the states of a two-state Markov chain. In this chapter, we analyze the decoding of LDPC codes over variations of this channel.

Temporal correlated noise behavior is a common consideration in wireless communications (BLACKARD; RAPPAPORT; BOSTIAN, 1993; FERTONANI; COLAVOLPE, 2009). While this characteristic has been studied in classical models such as the Gilbert-Elliot model (ELLIOTT, 1963; GILBERT, 1960; MUSHKIN; BAR-DAVID, 1989; ECKFORD; KSCHISCHANG; PASUPATHY, 2005), the most common approach is the channel modeling as a two-state Markov model where channel parameters or topology are assumed to be partially or fully known at the decoder (FERTONANI; COLAVOLPE, 2009; MOREIRA; PIMENTEL, 2017; MITRA; LAMPE, 2010).

Previous works in this setting mainly include an estimation module that provides log-likelihood ratio (LLR) updates to the conventional LDPC decoder (that implements the standard belief propagation algorithm). This module is based on the forward-backward algorithm (FBA) (FERTONANI; COLAVOLPE, 2009; MOREIRA; PIMENTEL, 2017; MITRA; LAMPE, 2010) where the channel parameters are assumed to be known or estimated using the Baum-Welch algorithm (BWA) (BAUM, 1972), or evenly jointly estimated in the decoding process (MITRA; LAMPE, 2009). These works consider different assumptions of what is considered to be known at the receiver, such as channel parameters or signal to noise ratio (SNR), which implies multiple scenarios in the receiver implementation. It is commonly considered that at least the channel model topology is known, such as the presence of a Markovian channel with a fixed number of states, which is of central importance in the FBA and BWA derivation, which has as a predefined input the number of states to be estimated. The problem of estimating a Gaussian source corrupted by a Markov-Gaussian channel is addressed in (ALAM; KADDOUM; AGBA, 2018; AHMED et al., 2020), where a Bayesian estimator is derived.

We consider a more general approach to the LLR updates. The proposed scheme employs a NN architecture for time series prediction based on gate recurrent units (GRUs) which aims to estimate the real-time channel variances without explicitly considering the channel assumptions or topology. For such, the NN must learn exclusively from the training data which reflects the considered channel characteristics.

By considering a real-time channel estimation in the form of an instantaneous variance prediction, the scenario is decoupled from the traditional Markovian state classification and can be used in an LLR update step. The presence of NN in multiple tasks in the com-

munication field (DÖRNER et al., 2018; SHLEZINGER et al., 2020; JIANG; SCHOTTEN, 2020) indicates that this toolset can provide new capabilities which improves performance or flexibility of existing schemes.

It is also considered a generalization of the Markov-Gaussian channel, in which the channel parameters are assumed to vary according to a Gaussian PDF, or by assuming the number of states as a random variable, which could reflect different mismatches between the modeled scenario and the deployed one in a practical setting. It is shown that the proposed scheme can be better suited to exploit noise correlations on generalized settings as its lack of direct assumptions is beneficial while the traditional methods are not well-suited when their assumptions are violated.

The rest of this chapter is organized as follows. In Section 2.1 we give an overview of the considered communication system. In Section 2.2 we present the classical HMM based approach to tackle the problem. Section 2.3 presents the LLR update schemes based on FBA/BWA. In Section 2.4 we introduce the NN-based model for the same task. Section 2.5 presents simulation results and performance comparisons. Final remarks are presented in Section 2.6.

2.1 COMMUNICATION SYSTEM

The components of the coded communication system considered in this chapter are shown in Figure 1. Each block is briefly described in the sequel. It is composed of an LDPC encoder and decoder, an interleaver and deinterleaver, a BPSK modulator, a Markov-Gaussian channel M realizing the noise $\boldsymbol{n}$, and a channel memory estimation module (CEM) that provides the LLR of each coded bit to the LDPC decoder.

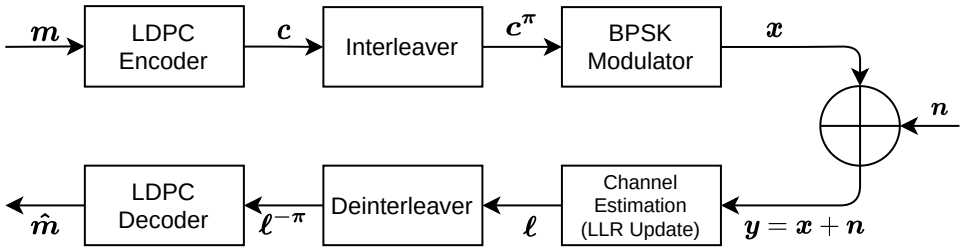
We consider a binary protograph-based LDPC code from the 5G celular system standard (BAE et al., 2019) with parameters (n_c, k_c, z_c), where n_c is the codeword length, k_c is the number of information bits, and z_c is an expansion factor. A codeword is denoted as $\boldsymbol{c} = (c_1, \ldots, c_{n_c})$.

The interleaver is known to be an important component to disperse noise bursts between different codewords. It is assumed here a regular block interleaver, as in (MITRA; LAMPE, 2010), with I rows and n_c columns such that the interleaved codeword $\boldsymbol{c}^\pi = (c_1^\pi, \ldots, c_{n_c}^\pi)$ is found as

$$c^\pi_{(jI+i)} = c_{(in_c+j)}, \quad 0 \leq i < I, 0 \leq j < n_c. \tag{1}$$

The interleaver depth I should be selected according to the channel memory. The modulator maps the interleaved binary-valued input c_i^π to BPSK symbols x_i according to the relation, $x_i = 1 - 2c_i^\pi$, where $x_i \in \{1, -1\}$, $1 \leq i \leq n_c$. Each entry of the received sequence $\boldsymbol{y} = (y_1, \ldots, y_{n_c})$ is given by $y_i = x_i + n_i$, where n_i is the i-th sample of the noise pro-

Figure 1 – Block diagram of the communication system.



Source: The author (2021).

cess (described in Section 2.1.1). The noise and modulated sequences are statistically independent.

The received sequence $\boldsymbol{y}$ is the input to the CEM that provides the LLR of each coded bit to the LDPC decoder in order to exploit the channel memory in the decoding process (the CEM approaches considered in this chapter are described in Section 2.3). The deinterleaved LLR sequence is the input to the LDPC decoder. For the LDPC decoding, we employ the *minsum* layered belief propagation (HOCEVAR, 2004) with number of iterations set to 8. This unusually low number is due to the decoder implementation which reduces the usual number of required iterations (BAE et al., 2019; HOCEVAR, 2004), while preserving sufficient accuracy in a lower simulation time.

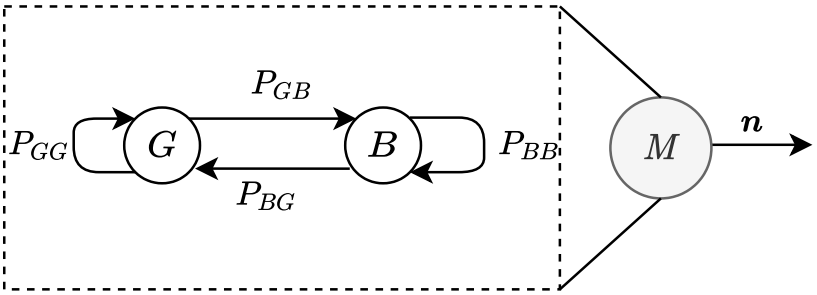
2.1.1 Channel Model

The Markov-Gaussian channel (FERTONANI; COLAVOLPE, 2009; MUSHKIN; BAR-DAVID, 1989; MITRA; LAMPE, 2010) consists of a stationary two-state Markov chain with state set $\boldsymbol{S} = \{G, B\}$, where the noise sample n_i generated from state G (termed good state) represents the background Gaussian noise (modeled by a zero-mean Gaussian random variable with variance σ_G^2) and from state B (termed bad state) represents the impulsive interference (zero-mean Gaussian with variance σ_B^2), where $\sigma_G^2 \leq \sigma_B^2$. The parameter $\kappa = \sigma_B^2/\sigma_G^2$ indicates the strength of the impulsive noise compared to the background noise. Figure 2 presents a two-state noise model diagram. The conditional PDF of n_i is:

$$b_p(n_i) \triangleq p_{n_i|s_i=p}(n_i) = \frac{1}{\sqrt{2\pi\sigma_p^2}} e^{\frac{-n_i^2}{2\sigma_p^2}}, \quad p \in \{G, B\}. \tag{2}$$

The (p, q)-th entry of the Markov chain transition probability matrix P is $P_{pq} = P(s_{i+1} = q \mid s_i = p), \quad p, q \in \boldsymbol{S}$, and the stationary vector is $[P_G, P_B] = [P_{BG}/(P_{GB} + P_{BG}), P_{GB}/(P_{GB} + P_{BG})], P_B + P_G = 1$. According to (MUSHKIN; BAR-DAVID, 1989), the parameter $\mu \triangleq 1 - P_{GB} - P_{BG}, 0 \leq \mu \leq 1$ represents the channel memory. Higher values of μ indicate greater temporal correlation between the channel states. The channel model is completely specified by the set $\Gamma = \{\sigma_G^2, \kappa, P_B, \mu\}$. The SNR per bit, denoted ζ, is defined

Figure 2 – Two-state Markov-Gaussian noise model diagram.



Source: The author (2021)

as

$$\zeta = \frac{1}{2\,R\,\sigma_G^2}, \tag{3}$$

where $R = k_c/n_c$ is the code rate.

The natural n-state, $n \geq 3$, channel generalization is defined by the state set $\mathbf{S} = \{G, B, B_1, \ldots, B_{n-2}\}$ where the state G defines the base channel configuration in which the definition of ζ is set, P is the $n \times n$ transition probability matrix, the variance of remaining states $\{B, B_1, \ldots, B_{n-2}\}$ are derived from G. For $s_i \in \mathbf{S} \setminus \{G\}$, σ_{s_i} is obtained from σ_G by definition of a second set $\mathbf{K} = \{\kappa_B, \kappa_{B_1}, \ldots, \kappa_{B_{n-2}}\}$ which provides the relations

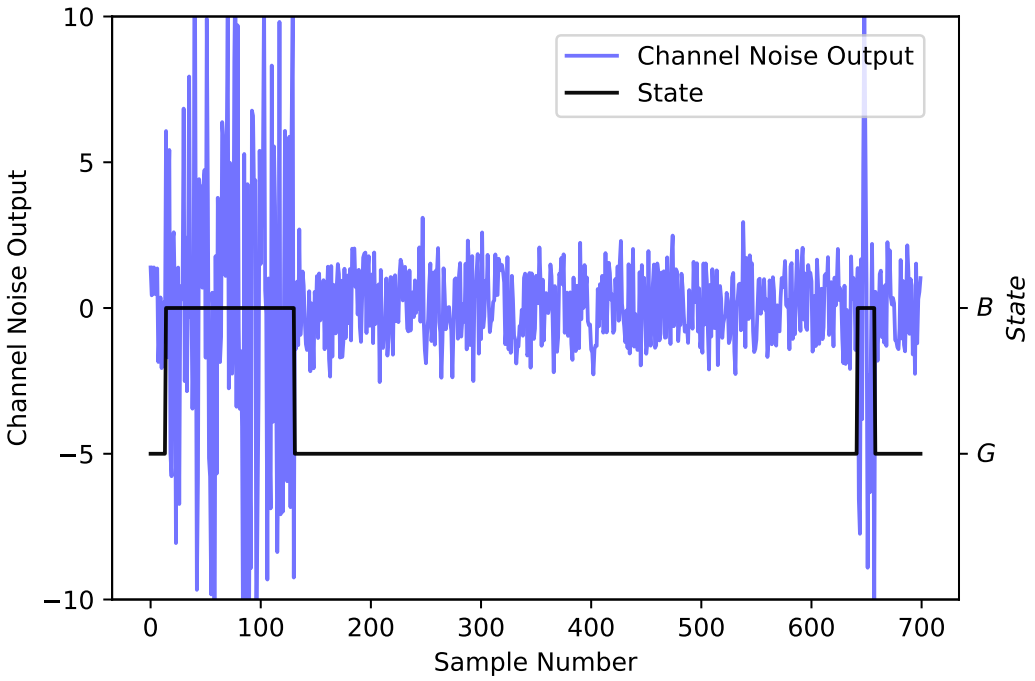
$$\sigma_{s_i}^2 = \kappa_{s_i}\sigma_G^2, \quad s_i \in \mathbf{S} \setminus \{G\}. \tag{4}$$

Figures 3 and 4 provide two noise realizations for a two-state channel with parameters $P_B = 0.1, \kappa = 100$, ζ = 4 dB and different μ values. Figure 3 employs $\mu = 0.97$ and Figure 4, $\mu = 0.70$. The parameter P_B indicates that on average 10% of samples correspond to state B, and μ controls how these samples are distributed. For high μ values (as in Figure 3) the noise realization is deeply correlated, meaning that if the current state is s_i, there is a high probability that the next sample also belongs to the same state, which translates to few state transitions. For lower μ values (as in Figure 4), the temporal dependency is weaker, meaning that two adjacent samples hold less information about its neighbors, that directly translates to reduced channel capacity (MUSHKIN; BAR-DAVID, 1989). This last channel is closer to a memoryless one.

2.2 HIDDEN MARKOV MODELS

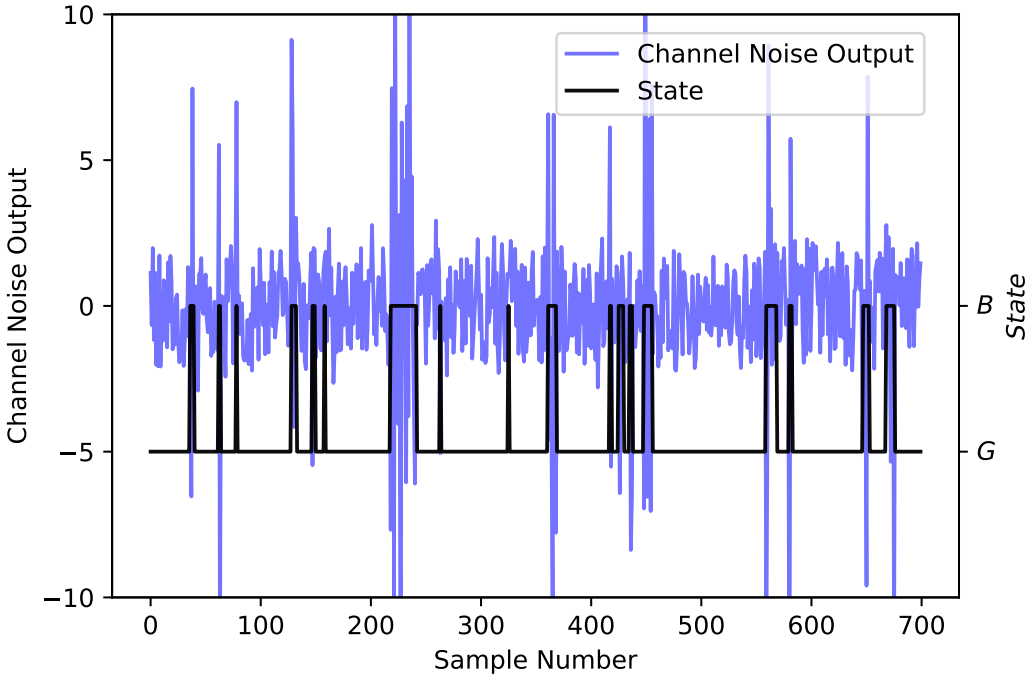
The task addressed in this section is the retrieval of the transmitted sequence $\boldsymbol{x}$ given the received sequence $\boldsymbol{y}$, that is, the computation of the *a posteriori probability* $P(x_i|\boldsymbol{y})$ for each timestep i, which can be directly used by the LDPC decoder. For such, it is expected that any model devised to this task must use the channel memory extracted in some form from $\boldsymbol{y}$, aided by usage of the channel model parameters Γ. The closely related task of estimating $P(s_i|\boldsymbol{y})$, the probability that the channel is at state s_i at timestep i given the

Figure 3 – Two-state Markov-Gaussian channel noise realization for $P_B = 0.1, P_G = 0.9, \kappa = 100, \mu = 0.97$, ζ = 4 dB, $P_{GG} = 0.997, P_{GB} = 0.003, P_{BG} = 0.027, P_{BB} = 0.973, R = 1/2, \sigma_G \approx 0.63, \sigma_B \approx 6.31$. The solid black line indicates the state $s_i \in \boldsymbol{S}$.



Source: The author (2021)

Figure 4 – Two-state Markov-Gaussian channel noise realization for $P_B = 0.1, P_G = 0.9, \kappa = 100, \mu = 0.70$, ζ = 4 dB, $P_{GG} = 0.97, P_{GB} = 0.03, P_{BG} = 0.27, P_{BB} = 0.73, R = 1/2, \sigma_G \approx 0.63, \sigma_B \approx 6.31$. The solid black line indicates the state $s_i \in \boldsymbol{S}$.



Source: The author (2021)

received sequence $\boldsymbol{y}$, can be seen as a direct instance of a HMM and FBA algorithm (BAHL et al., 1974). Adaptations of this method have been discussed in (FERTONANI; COLAVOLPE, 2009; MOREIRA; PIMENTEL, 2017; CHEN; TANNER, 2006; ALAM; KADDOUM; AGBA, 2018) and are compared to the proposed NN approach in following sections.

We consider the HMM task as the estimation of $P(s_i|\boldsymbol{n})$, the probability of being in state s_i at timestep i given the actual noise realization of the channel. It is clear that we do not observe the noise directly in a practical setting, nevertheless, it is possible to consider a training phase scenario where pilot symbols are transmitted aiming at the retrieval of $\boldsymbol{n}$. In the deployment scenario, $P(s_i|\boldsymbol{n})$ is used in the computation of the LLR (FERTONANI; COLAVOLPE, 2009; CHEN; TANNER, 2006), defined in Section 2.3.

The training phase scenario is used here instead of the joint channel state estimation and decoding procedures, such as in (CHEN; TANNER, 2006) (MITRA; LAMPE, 2009) for fairness comparison with the NN approach, since our proposed NN-based model requires a training phase. It should be noted that, for the HMM case, the consideration of a training phase scenario where the pilot symbols are transmitted means in practice that the iterative joint estimation and decoding provides no performance gain compared to a non-iterative one, as stated in (FERTONANI; COLAVOLPE, 2009). The performance gain is possible if no training phase or channel considerations are assumed such as in (MITRA; LAMPE, 2009). Therefore, our HMM benchmark can be viewed as a close to maximum performance achievable by the FBA approach.

We denote the n-state HMM as the set of parameters $\boldsymbol{\Omega} = \{P, \boldsymbol{b}\}$, where P is the $n \times n$ transition matrix as before and $\boldsymbol{b}$ is a n-dimensional vector of conditional PDFs, where $b_p(n_i)$ indicates the conditional PDF of obtaining n_i when in state p at timestep i. Usually, HMMs are defined in terms of observations in the discrete case, and the parameter $\boldsymbol{b}$ denotes an output probability matrix. For the considered case, although the set of states $\boldsymbol{S}$ is finite, the observable values are continuous. Therefore we use PDFs to represent $\boldsymbol{b}$. This can be done in multiple ways. Here, $b_p(n_i)$ is considered as a Gaussian PDF with mean value $\mu_p = 0$ and variance σ_p^2 (JURAFSKY; MARTIN, 2008) (HUANG; REDDY; ACERO, 2001) as in (2). The definition of $\boldsymbol{b}$ in (2) means in practice that the standard BWA is slightly modified when compared to the usual discrete implementation.

An important consideration to be done is whether the set $\boldsymbol{\Omega}$ is assumed to be known or must be estimated, and under which assumptions. It is frequently assumed some form of *maximum a posteriori* (MAP) based state detection using the BCJR algorithm. For such, the set of parameters $\boldsymbol{\Omega}$ is necessary, which can be assumed or estimated in an interactive way by different formulations (FERTONANI; COLAVOLPE, 2009; MOREIRA; PIMENTEL, 2017; CHEN; TANNER, 2006; MITRA; LAMPE, 2009; MOON; WONG; SHEA, 2006; ALAM; KADDOUM; AGBA, 2018). These results presents a suitable approach in a practical setting, although there are still strong implicit considerations being done about the channel, such

as the model being trained for each considered ζ, which can be viewed as a practical consideration since (2) depends on ζ, or is at least considered that the channel is a two-state Markov-Gaussian channel itself since the definition of the number of states is critical to the algorithmic implementation. An alternative way of approaching the task with different formulation and assumptions is discussed in Section 2.4. Results for different models and considerations are presented in Section 2.5.

In this chapter, if no prior assumption is made about $\mathbf{\Omega}$, the first step of the HMM approach is the estimation of $\mathbf{\Omega}$ by the BWA algorithm during training phase by pilot symbols, which can be done by the following brief formulation (FERTONANI; COLAVOLPE, 2009; JURAFSKY; MARTIN, 2008; RABINER, 1989) of the forward-backward recursion. We define $\alpha_i(p)$, the probability of seeing observations $y_1, y_2, \ldots, y_i$ being in state p at timestep i, and $\beta_i(p)$, the probability of the ending sequence $y_{i+1}, \ldots, y_T$ given that the channel is in state p at timestep i, and we initialize (FERTONANI; COLAVOLPE, 2009)

$$\alpha_1(p) = \pi_p, \quad p \in \boldsymbol{S} \tag{5}$$

$$\beta_T(G) = 1, \tag{6}$$

$$\beta_T(p) = 0, \quad p \in \boldsymbol{S} \setminus \{G\}, \tag{7}$$

where π_p refers to the stationary probability of state p, which can be initialized to plausible values if assumed to be unknown, or simply $\pi_G < \pi_B < \ldots < \pi_{B_{n-2}}$ with evenly spaced values. At each executed step of the algorithm we normalize α and β values to the $[0, 1]$ range. We recursively compute

$$\alpha_i(q) = \sum_{p \in \boldsymbol{S}} \alpha_{i-1}(p) P_{pq} b_q(n_i), \quad 1 < i \leq T \tag{8}$$

$$\beta_i(p) = \sum_{q \in \boldsymbol{S}} \beta_{i+1}(q) P_{pq} b_q(n_{i+1}), \quad 1 \leq i < T. \tag{9}$$

After each iteration we compute $\xi_i(p, q) \triangleq P(s_i = p, s_{i+1} = q)$ as

$$\xi_i(p, q) = \frac{\alpha_i(p) P_{pq} \beta_{i+1}(q) b_q(n_{i+1})}{\sum_{k \in \boldsymbol{S}} \sum_{w \in \boldsymbol{S}} \alpha_i(k) P_{kw} \beta_{i+1}(w) b_w(n_{i+1})}, \tag{10}$$

and $\gamma_i(p) \triangleq P(s_i = p | \boldsymbol{n})$ as

$$\gamma_i(p) = \frac{\alpha_i(p) \beta_i(p)}{\sum_{s \in \boldsymbol{S}} \alpha_i(s) \beta_i(s)}, \tag{11}$$

which, after training, is considered as the final output of the model after the last iteration. The transition matrix P is updated as

$$P_{pq} = \frac{\sum_{i=1}^{T} \xi_i(p,q)}{\sum_{i=1}^{T} \gamma_i(p)}. \tag{12}$$

We use the following re-estimation for (2) at each iteration (JURAFSKY; MARTIN, 2008) (HUANG; REDDY; ACERO, 2001):

$$\sigma_p^2 = \frac{\sum_{i=1}^{T} \gamma_i(p)(n_i - \mu_p)^2}{\sum_{i=1}^{T} \gamma_i(p)}. \tag{13}$$

Additionally, parameters μ_p are considered zero since the noise is assumed to have zero mean.

After the last iteration of the algorithm the set $\mathbf{\Omega}$ is obtained, and the training phase is taken as complete. The $\gamma_i(p)$ values are employed in decoding process, as described in Section 2.3.

2.3 CHANNEL ESTIMATION

In this section, we consider different strategies for CEM in computing the LLRs which accomplish the mapping defined as (CHEN; TANNER, 2006)

$$L(y_i|c_i^\pi) = \log\left(\frac{p_{y_i|c_i^\pi=0}(y_i)}{p_{y_i|c_i^\pi=1}(y_i)}\right) \tag{14}$$

$$= \log\left(\frac{\sum_{p\in\boldsymbol{S}} p_{y_i|c_i^\pi=0,s_i=p}(y_i)\gamma_i(p)}{\sum_{p\in\boldsymbol{S}} p_{y_i|c_i^\pi=1,s_i=p}(y_i)\gamma_i(p)}\right) \tag{15}$$

$$= \log\left(\frac{\sum_{p\in\boldsymbol{S}} \frac{\gamma_i(p)}{\sqrt{2\pi\sigma_p^2}}\, e^{\frac{-(y_i-1)^2}{2\sigma_p^2}}}{\sum_{p\in\boldsymbol{S}} \frac{\gamma_i(p)}{\sqrt{2\pi\sigma_p^2}} e^{\frac{-(y_i+1)^2}{2\sigma_p^2}}}\right), \tag{16}$$

where $\gamma_i(p)$ is the probability that the channel is in state p at timestep i, and σ_p is the estimated standard deviation from state p. Note that (14) computes the LLR by conditioning the transmition hypothesis of c_i^π and consequently $p_{y_i|c_i^\pi}(y_i) = p(n_i = y_i - x_i)$ (FERTONANI; COLAVOLPE, 2009), and (15) expands it in terms of the state hypothesis and incorporates the probabilities $\gamma_i(p), \quad p \in \boldsymbol{S}$.

2.3.1 KSMLD (MITRA; LAMPE, 2010)

The best possible considered CEM is the *known-state maximum-likelihood decoder* (KSMLD), adapted from (MITRA; LAMPE, 2010), in which the decoder has perfect knowledge

of the state s_i and the corresponding variance $\sigma^2_{s_i}$ for each timestep i. In practice, this means that $\gamma_i(p) = 1$ for some $p \in \boldsymbol{S}$, and (15) simplifies to

$$L(y_i|c_i) = \frac{2y_i}{\sigma^2_{s_i}}, \tag{17}$$

where the state and standard deviation σ_{s_i} is assumed to be known. This model provides the maximum possible performance benchmark considered in this chapter, since the proposed models make use of (15).

2.3.2 HMM FBA (FERTONANI; COLAVOLPE, 2009)

In this model, it is assumed that the channel parameters $\boldsymbol{\Omega}$ are known and the model is used as a soft-input soft-output (SISO) module, employing the FBA. The HMM training phase and BWA are not done, although the forward-backward recursion in (8), (9) and (11) are still used in obtaining $\gamma_i(s)$. The knowledge assumption of PDFs b_p indicates that σ_p are known for $p \in \boldsymbol{S}$ and, therefore, we use (16) directly. It is argued in (FERTONANI; COLAVOLPE, 2009) that an iterative decoder that exchanges soft-information between CEM and LDPC decoder provides incremental performance gains and this strategy is not considered here.

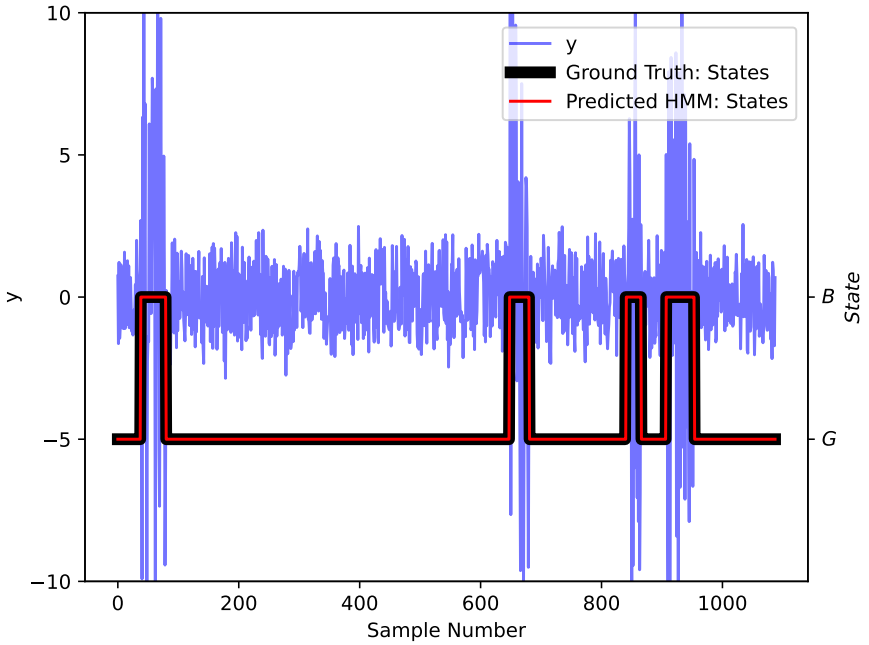
Figures 5 and 6 present state predictions done by the HMM FBA model over channels with different μ values. Figure 5 refers to $\mu = 0.97$, in which it can be seen that the long state persistence results in a high state prediction accuracy. Figure 6 refers to $\mu = 0.70$, in which the lower temporal correlation results in more frequent state transitions which results in lower state prediction accuracy.

2.3.3 HMM BWA

In this model, the BWA training phase is done as in Section 2.2, and (16) is applied just as before. The values used for $\boldsymbol{\Omega}$ are the ones estimated in the training phase employing the BWA. The training phase dataset simulates a possible ζ mismatch by considering training sequences $\boldsymbol{n}$ obtained by randomly selecting ζ in an interval [2.5dB, 5dB], chosen as a plausible use case where the expected bit error rate is not too low or high for the given code. The training set is considered as 1000 sequences of size n_c, where each sequence has ζ uniformly randomly selected in the given range.

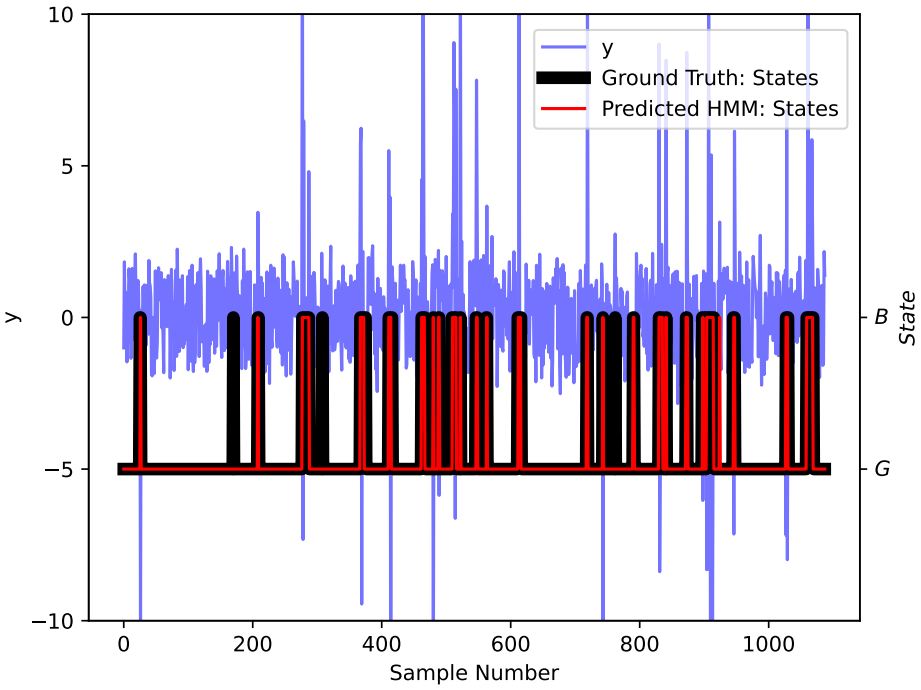
It is empirically seen that this assumption translates to well estimated transition matrix P and an estimated ζ as the mean of the ζ range considered. In practice, this mismatch is robust for most scenarios, as indicated in (FERTONANI; COLAVOLPE, 2009).

Figure 5 – State prediction by HMM FBA over a two-state Markov-Gaussian channel, $P_B = 0.1, \kappa = 100, \mu = 0.97, \zeta = 4.5$ dB. The solid black line indicates the state $s_i \in \boldsymbol{S}$, and the solid red line indicates the HMM model state prediction ($\text{argmax}_p \gamma_i(p)$).



Source: The author (2021)

Figure 6 – State prediction by HMM FBA over a two-state Markov-Gaussian channel, $P_B = 0.1, \kappa = 100, \mu = 0.75, \zeta = 4.5$ dB. The solid black line indicates the state $s_i \in \boldsymbol{S}$, and the solid red line indicates the HMM model state prediction ($\text{argmax}_p \gamma_i(p)$).



Source: The author (2021)

2.4 NEURAL NETWORK BASED DECODING

The NN module is designed to estimate instantaneous values of the noise process standard deviation σ_{s_i} for each timestep i given the received sequence $\boldsymbol{y}$ from Figure 1 as input, that is $\mathcal{N}(\boldsymbol{y}) = \hat{\boldsymbol{\sigma}}$, where $\mathcal{N}(\cdot)$ denotes the NN mapping, $\boldsymbol{y}$ is defined in interval $1 \leq i \leq T$ and $\hat{\boldsymbol{\sigma}}$ the estimated standard deviation vector, defined in the same interval. The NN based CEM does not estimate the state probabilities $\gamma_i(p)$ as the HMM, since the NN does not classify between states. Instead, the estimated noise standard deviation $\hat{\sigma}_{s_i}$ is used directly in the LLR computation for each timestep as in (17). This definition is motivated intuitively by two facts: the consistency between (15) and (17) means that the direct normalization procedure in (17) can be used if we have high confidence that the current state is known. This assumption is true for the majority of timesteps since erroneous state prediction comes almost exclusively when state transitions are occurring which are not frequent on most relevant settings.

The NN must learn channel characteristics directly from the provided dataset, and can provide more flexibility if more general assumptions about the channel are used, which could be inadequate to HMM training. In the canonical fixed two-state Markov-Gaussian setting traditionally considered in (FERTONANI; COLAVOLPE, 2009) (MITRA; LAMPE, 2010) (GILBERT, 1960) it is expected that the proposed model performs very near to established methods, even when no explicit channel assumptions are directly designed in the NN architecture.

We employ a bidirectional RNN structure (GOODFELLOW; BENGIO; COURVILLE, 2016) (SCHUSTER; PALIWAL, 1997) based on architectures presented in (KIM et al., 2018) (CELIK et al., 2020). The architecture employs GRU units (CHO et al., 2014) as memory learning structures. The bidirectional RNN formulation permits the training to be done simultaneously in positive and negative time direction (SCHUSTER; PALIWAL, 1997) which holds similarity with the forward-backward HMM algorithms and allows the NN to learn about channel memory from past and future timesteps. Convolutional and fully connected layers are also present which is typical for NNs, the full architecture is shown in Table 1.

Since we are dealing with a regression task, we employ the mean squared error (MSE) loss function

$$J(\hat{\boldsymbol{\sigma}}, \boldsymbol{\sigma}) = \frac{1}{\Lambda} \sum_{k=1}^{\Lambda} \sum_{i=1}^{t_s} (\hat{\sigma}_{s_i} - \sigma_{s_i})^2, \tag{18}$$

where Λ is the batch size, chosen as 32, and t_s is the number of considered timesteps per input frame, chosen as 736. The choice of t_s has a practical importance for the model. It must be noted that although one may argue that a more convenient choice of t_s is the input frame size itself, the length of $\boldsymbol{y}$ ($T = I \times I_c$ = 14720). This choice would result in an unnecessarily

Table 1 – Employed NN architecture.

Layer Type	Nodes/Activation	Output Dimension (Λ = 32, t_s = 736)
Input Layer	-	(Λ, t_s, 1)
Convolutional 1-Dimensional	32/*relu*	(Λ, t_s, 32)
Convolutional 1-Dimensional	32/*linear*	(Λ, t_s, 32)
Batch Normalization	-	(Λ, t_s, 32)
Bidirectional GRU	70×2/*tanh*	(Λ, t_s, 140)
Batch Normalization	-	(Λ, t_s, 140)
Bidirectional GRU	40×2/*tanh*	(Λ, t_s, 80)
Batch Normalization	-	(Λ, t_s, 80)
Bidirectional GRU	30×2/*tanh*	(Λ, t_s, 60)
Time Distributed Fully Connected	30/*relu*	(Λ, t_s, 30)
Time Distributed Fully Connected	10/*relu*	(Λ, t_s, 10)
Time Distributed Fully Connected	1/*linear*	(Λ, t_s, 1)

long architecture size since the number of nodes should be chosen accordingly. Therefore, we make the practical choice of using $t_s = \frac{T}{20} = 736$ which is long enough to capture channel memory while preserving a practical network size. We then employ the model repeatedly for 20 contiguous subsequences $\boldsymbol{y}^q$ of $\boldsymbol{y}$, $\quad 1 \leq q \leq 20$, with

$$y_i^q = y_i, \quad 1 + (q-1)t_s \leq i \leq qt_s, \tag{19}$$

and take $\boldsymbol{y}^q$ as input to the NN applying

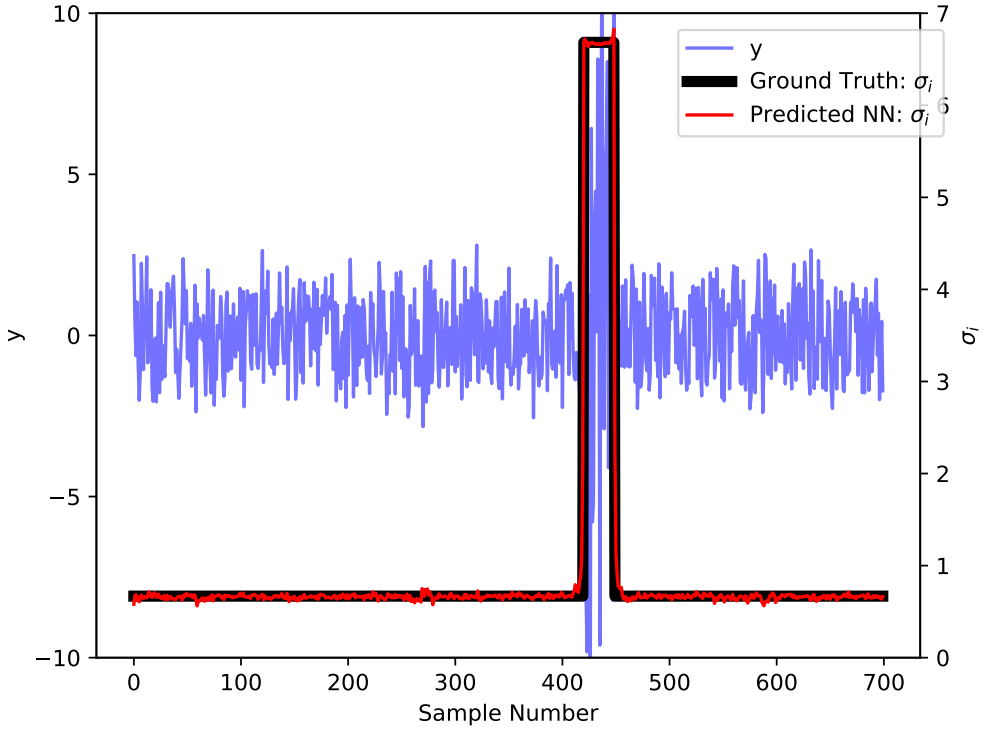
$$\hat{\boldsymbol{\sigma}}_q = \mathcal{N}(\boldsymbol{y}^q), \tag{20}$$

and obtain the final $\hat{\boldsymbol{\sigma}}$, of length T, by concatenation of $\hat{\boldsymbol{\sigma}}_q$, $\quad 1 \leq q \leq 20$.

The training phase is done using Adam optimizer (KINGMA; BA, 2015) with learning rate $l_r = 10^{-3}$. Two heuristics are used to control training convergence: training iterations are only stopped when the cost function stagnates for 10 iterations on a hold-out validation set and the learning rate l_r is reduced by a factor of 10 if the cost function stagnates for 4 iterations on the same validation set (RASKUTTI; WAINWRIGHT; YU, 2011). For each training iteration, a new dataset batch is created by simulating Figure 1 diagram with a random message $\boldsymbol{m}$, and we take $\boldsymbol{y}$ to construct the NN input $\boldsymbol{y^q}$ as in (19). The target sequence $\boldsymbol{\sigma}$, which holds the actual value of standard deviation realized by the channel model M at each timestep i, that is $\boldsymbol{\sigma} = \sigma_{s_i}, 1 \leq i \leq T$ is obtained by the NN mapping (20).

An important practical issue is the choice of the ζ value for the training set (KIM et al., 2018) (GRUBER et al., 2017) (GEORGE; HUERTA, 2018) (TANDLER et al., 2019). If the NN is trained on high ζ, it is not suited to lower ζ as the NN tends to learn suboptimal representations that are insufficient to make extrapolations (BJORNSON; GISELSSON, 2020), while if it is trained on low ζ, the model might not learn at all because the training data may be overly corrupted to represent the system in a reliable way.

Figure 7 – Standard deviation σ_{s_i} prediction by NN over a two-state Markov-Gaussian channel, $P_B = 0.1, \kappa = 100, \mu = 0.97, \zeta = 3.5$ dB. The solid black line indicates the noise standard deviation σ_{s_i} at each timestep, the solid red line indicates the NN model noise standard deviation prediction $\hat{\sigma}_{s_i}$.



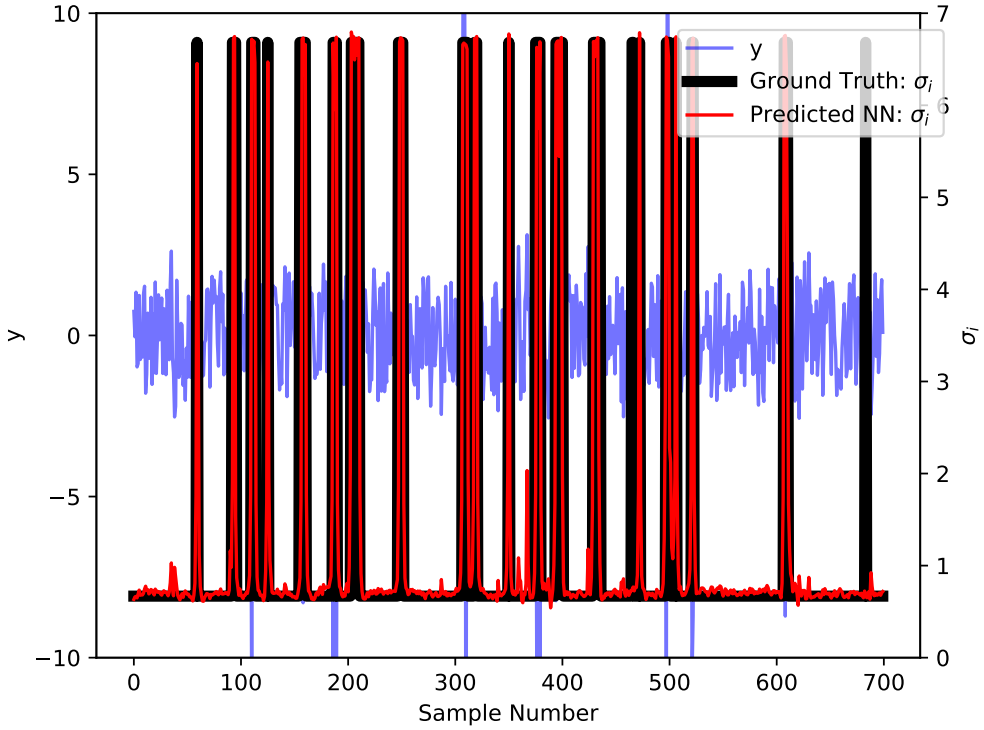
Source: The author (2021)

The approach in (KIM et al., 2018) chooses the training ζ value as $\min\{\zeta_{test}, 0dB\}$. A specific metric (normalized validation error) is proposed in (GRUBER et al., 2017) to measure the accuracy of the training ζ. In (TANDLER et al., 2019), a priori ramp-up scheme modifies the probability distribution of the input sequence during training.

The chosen heuristic for the choice of training ζ is adapted from (KIM et al., 2018), where some guidelines are proposed. In this chapter, for each training batch the data ζ is chosen by sampling a continuous uniform distribution on an interval defined according to the desired use case scenario, which depends on the considered simulation. A plausible range was empirically determined as [2.5dB, 5dB]. The validation dataset ζ is made higher than the training case, which will select models that are able to generalize well for higher ζ values, tipically it is used [3.75dB, 5dB]. Deviations from these intervals are stated in Section 2.5 if needed.

Similarly to Figures 5 and 6 in the HMM case, the instantaneous standard deviation prediction done by the NN are shown in Figures 7 and 8. As before, Figure 7 shows the $\mu = 0.97$ channel, in which it can be seen a high model prediction accuracy. Figure 8 shows a similar phenomenon to Figure 6, as lower channel memory directly translates to lower model accuracy, in particular around state transitions samples.

Figure 8 – Standard deviation σ_{s_i} prediction by NN over a two-state Markov-Gaussian channel, $P_B = 0.1, \kappa = 100, \mu = 0.70, \zeta = 3.5$ dB. The solid black line indicates the noise standard deviation σ_{s_i} at each timestep, the solid red line indicates the NN model noise standard deviation prediction $\hat{\sigma}_{s_i}$.



Source: The author (2021)

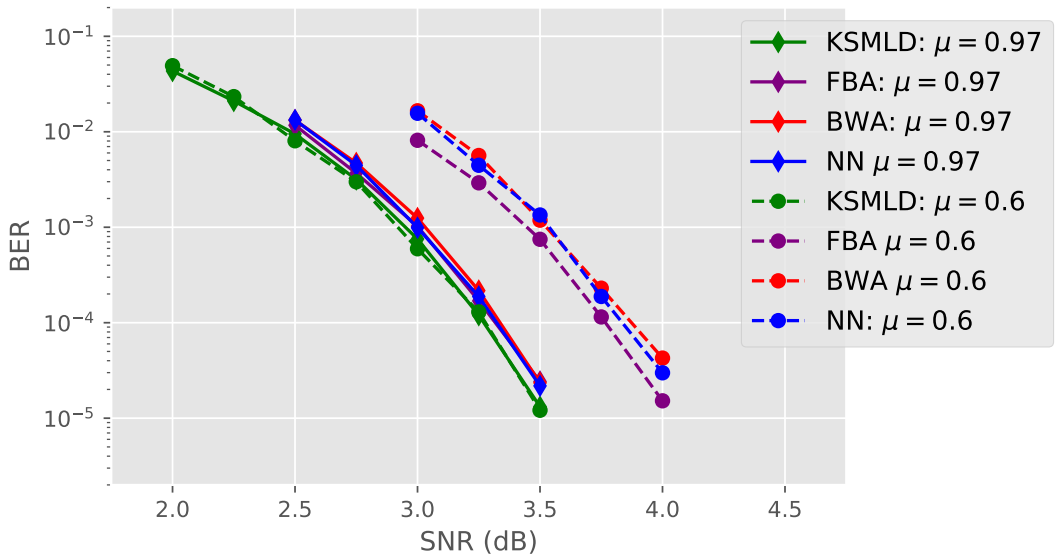
The NN is trained under the same considerations as in Section 2.3.3. Although differently from HMM training, the NN is trained directly with received sequences $\boldsymbol{y}$, which aims to decouple the noise state and standard deviation predictions for more general settings.

2.5 RESULTS

We present simulation results considering three different scenarios, in which we compare the performance of the CEM presented in the previous sections (KSMLD, FBA, BWA, NN). The parameters of the LDPC code are (1472, 704, 32). The interleaver depth is fixed to $I = 10$ for consistence across the simulated scenarios, which performs well in various settings. The performance curves are given in terms of the bit error rate (BER) versus the ζ defined in (3).

It should be noted that under LDPC coding and burst error hyphotesis, it is known that the error floor phenomenon should occur for some ζ (MITRA; LAMPE, 2010) (NURELLARI; İNCE, 2012) which affects the BER curve's slope. Nevertheless, the performance comparisons are made relatively to the KSMLD metric since the channel estimation procedure targets recovery of the channel memory, which is assumed perfect in KSMLD.

Figure 9 – BER versus ζ for the two-state Markov-Gaussian channel with $\kappa = 100$, $P_B = 0.1$, and two values of channel memory: $\mu = 0.97$ (solid lines) and $\mu = 0.6$ (dashed lines).



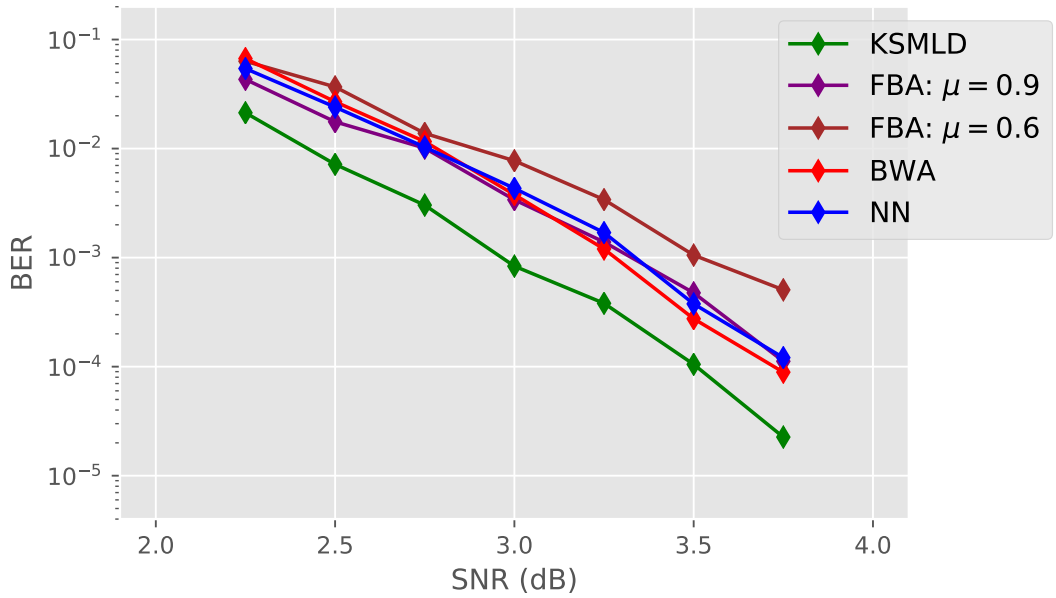
Source: The author (2021).

Figure 9 presents performance comparisons for two values of μ, indicating high channel memory ($\mu = 0.97$, in continuous line) and low channel memory ($\mu = 0.6$, in dashed lines), $\kappa = 100$, $P_B = 0.1$. The performance at high memory configuration is close to that of the KSMLD for all analyzed CEM scenarios. For the lower memory scenario, it is expected that performance is reduced due to lower noise temporal correlation and more frequent transitions between states. The results indicate that the proposed NN is able to match traditional CEM strategies even when relaxing the channel model assumptions and provides an alternative method to the frequently considered FBA and BWA schemes.

The second scenario investigates the effect of channel parameter mismatch, which represents not only training imperfections, but also a possible deviation from the assumed channel model. Here we assume that for each interleaver block of size $I \times n_c$ the parameters μ and κ of the two-state channel are drawn from a truncated Gaussian PDF centered at 0.9 (with standard deviation 0.1) and 60 (with standard deviation 20), respectively (the actual values of these parameters are $\mu \triangleq \min(0.99, x)$ and $\kappa \triangleq \max(10, x)$, where x is a sample from the corresponding Gaussian PDF). The training dataset for the BWA and NN considers the variation of these two channel parameters and the FBA considers fixed parameters, the average values ($\mu = 0.9$ and $\kappa = 60$) as well as an underestimation of channel memory ($\mu = 0.6$ and $\kappa = 60$).

The curves in Figure 10 indicate that considered CEM are robust to small deviations of the channel parameters. The robustness of the HMM FBA for a variant of this scenario is discussed in (FERTONANI; COLAVOLPE, 2009). It is empirically seem that the BWA estimates its channel parameters Ω based on the mean of both μ and κ, which translates

Figure 10 – BER versus ζ for the two-state Markov-Gaussian channel. The channel parameters μ and κ are drawn from a Gaussian PDF with mean 0.9 and 60, respectively. $P_B = 0.1$.



Source: The author (2021).

to a minor performance impact, since the FBA with $\mu = 0.9$ performs very similarly to BWA. A performance degradation is observed for the FBA $\mu = 0.6$, since the low channel memory hypothesis is unable to fully utilize the actual channel memory, which is higher than the assumed $\mu = 0.6$. This means that the underestimation of the channel memory can provoke performance degradation. The NN is able to replicate the performance of other CEM strategies, as in the previous scenario.

The third scenario is a generalization of the two-state Markov-Gaussian channel where the number of states is allowed to vary. Three possible channel configurations are selected with equal probability in each interleaver block of size $I \times n_c$. This approach seeks to evaluate the flexibility of the employed CEM under deviation from the traditional setting. The channel parameters (with 1 good and $(n-1)$ bad states) for each configuration are:

$$\boldsymbol{P}_1 = \begin{bmatrix} 0.997 & 0.003 \\ 0.027 & 0.973 \end{bmatrix}, \quad \boldsymbol{\kappa}_1 = 100, \tag{21}$$

$$\boldsymbol{P}_2 = \begin{bmatrix} 0.990 & 0.008 & 0.002 \\ 0.026 & 0.960 & 0.014 \\ 0.010 & 0.030 & 0.960 \end{bmatrix}, \quad \boldsymbol{\kappa}_2 = [10, 100], \tag{22}$$

$$\boldsymbol{P}_3 = \begin{bmatrix} 0.950 & 0.040 & 0.010 & 0.000 \\ 0.300 & 0.500 & 0.150 & 0.050 \\ 0.100 & 0.300 & 0.500 & 0.100 \\ 0.000 & 0.100 & 0.400 & 0.500 \end{bmatrix}, \boldsymbol{\kappa}_3 = [10, 60, 100], \tag{23}$$

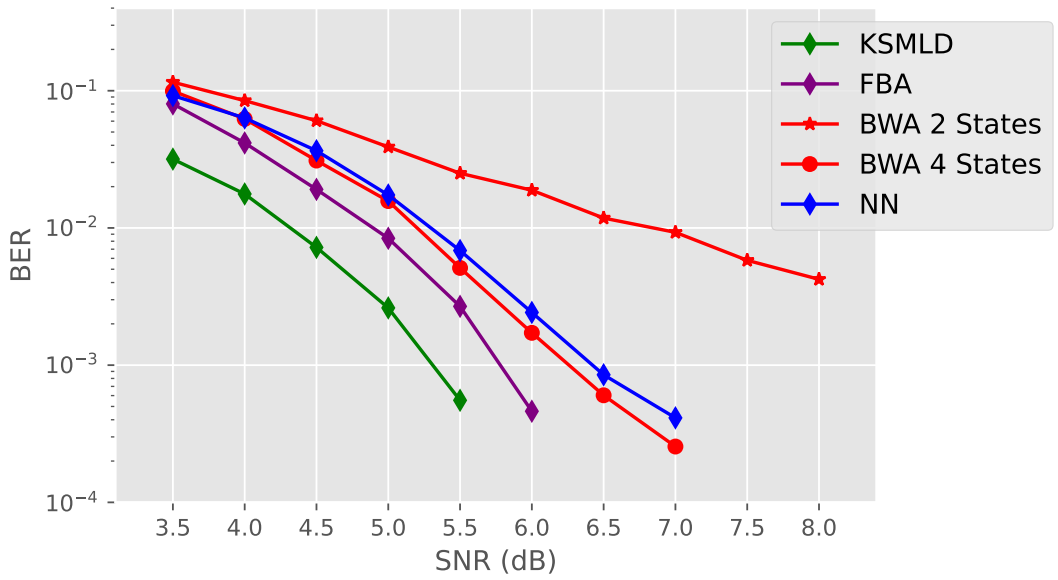
where each entry of the vector $\boldsymbol{\kappa}$ is the ratio of the variance of the corresponding bad state to σ_G^2.

One difficulty that must be addressed is how to properly train the Markov-Gaussian channel in this scenario, since the BWA assumes that the number of states is known. We present BWA training using the assumption that the channel has either 2 or 4 states. The first case indicates a scenario where an insufficient number of states is assumed, and the second one represents a better channel configuration guess. Figure 11 shows the BER for this scenario. Since this scenario has lower expected performance, indicated by the KSMLD reference curve, it is required higher ζ for similar BER performance compared to that obtained in previous scenarios. BWA and NN are trained using a different ζ interval [5 dB, 7.5 dB] and validation interval [6.0 dB, 7.0 dB]. The FBA is also taken as a benchmark, since the channel parameters are assumed to be known in each interleaver block, and therefore, the randomness of the channel configuration is not relevant for this model. The results indicates that the NN model is able to learn a suitable channel representation solely from the provided dataset, while the BWA is susceptible to the number of states mismatches, although it is still possible to maintain a good performance if some channel knowledge is assumed, as indicated by the BWA four-state model. The BWA two-state model suffers severely in performance, as it is unable to represent the channel behavior.

It is noticed that the two CEM that estimate the channel configuration (BWA and NN) suffer performance degradation for high ζ since their BER curves do not follow the slope of the KSMLD and FBA benchmarks. This effect can be explained by two factors: (i) a wider range of ζ is used in the training phase, and the ζ mismatch affects the BER for high ζ values (GRUBER et al., 2017); (ii) since in this scenario the channel estimation task is harder, the CEM has greater difficulty in accurately estimating the channel characteristics, resulting in a worse performance for low BER since higher estimation precision is required in this BER range.

The NN based model can be seen as a more flexible approach to the traditional CEM for the Markov-Gaussian channel. Nevertheless the best possible HMM usage where some channel consideration is assumed, such as the knowledge of the ζ or the number of states, is unbeatable in the presented simulations.

Figure 11 – BER versus ζ curves for the Markov-Gaussian channel with 2, 3 or 4 states. One of three presented channel configurations is selected with equal probability in each interleaver block.



Source: The author (2021).

2.6 FINAL CONSIDERATIONS

This chapter discussed different CEM strategies for LDPC decoding over bursty channels modeled by the Markov-Gaussian channel. The traditional CEM approaches were compared to a proposed alternative based on NN. The NN is able to match established performance results in settings previously considered in literature, which indicates that it is a suitable alternative to the proposed task, while being entirely data-driven. In the HMM usage, some channel consideration is assumed, such as the knowledge of the SNR or the number of states. It is also important to note that the HMM performs its forward-backward algorithm in $\mathcal{O}(N^2T)$ time complexity (N being the number of states and T the number of observations). While the neural network complexity cannot be calculated in the same manner, we employ a network architecture which is linear in the number of observations.

In some channel mismatch scenarios, it is possible that the lack of NN assumptions about the channel model directly translates to a performance gain if the assumptions adopted by either FBA or BWA are violated. The NN is a flexible model designed to work in a wider variety of settings while no additional modification in its architecture is required, apart from the training dataset.

3 NON-ORTHOGONAL MULTIPLE ACCESS

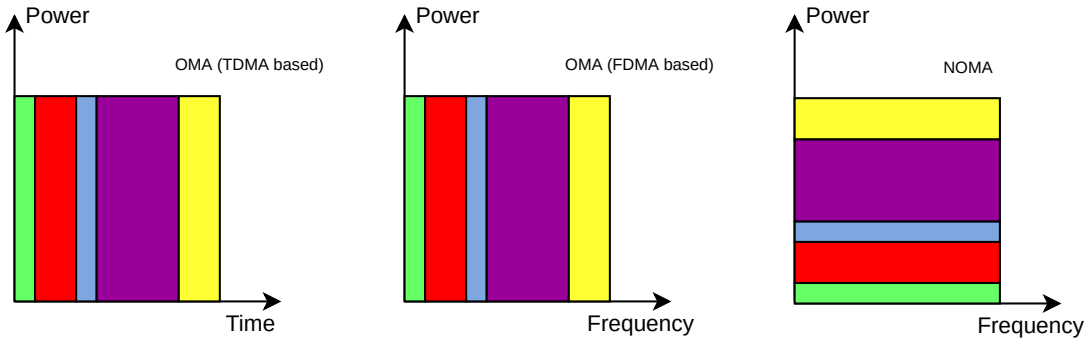
Channel resource sharing techniques have been gaining importance due to the necessity of achieving higher spectral efficiency in the next generations of wireless communications systems (SAAD; BENNIS; CHEN, 2020). Among them, power domain non-orthogonal multiple access (NOMA) is an important technique in accomplishing this task (DING et al., 2017), which is also frequently linked to 5G technologies (BUDHIRAJA et al., 2021). In this scheme, signals from multiple users with different power levels share the same channel resources, and the successive interference cancellation (SIC) is the usual decoding method, in which the decoding at each user is performed sequentially according to the ordering of their power levels.

A pictorial comparison between orthogonal multiple access (OMA) and NOMA is shown in Figure 12, where OMA is represented by time-division multiple access (TDMA) and frequency-division multiple access (FDMA), in which the users can be separated by their time or frequency resources, respectively. On the other hand, in the NOMA scheme the users share the same frequency resources.

Two subdivisions are typically considered in the NOMA case: uplink and downlink NOMA, which are illustrated in Figures 13 and 14, respectively. In the uplink scenario, multiple users, denoted by u_i, transmit their data to a base station (BS) using the same channel resource. In the downlink scenario, the BS transmits samples which holds informations pertinent to all users in the same channel resource. More precise formulations are shown along the chapter.

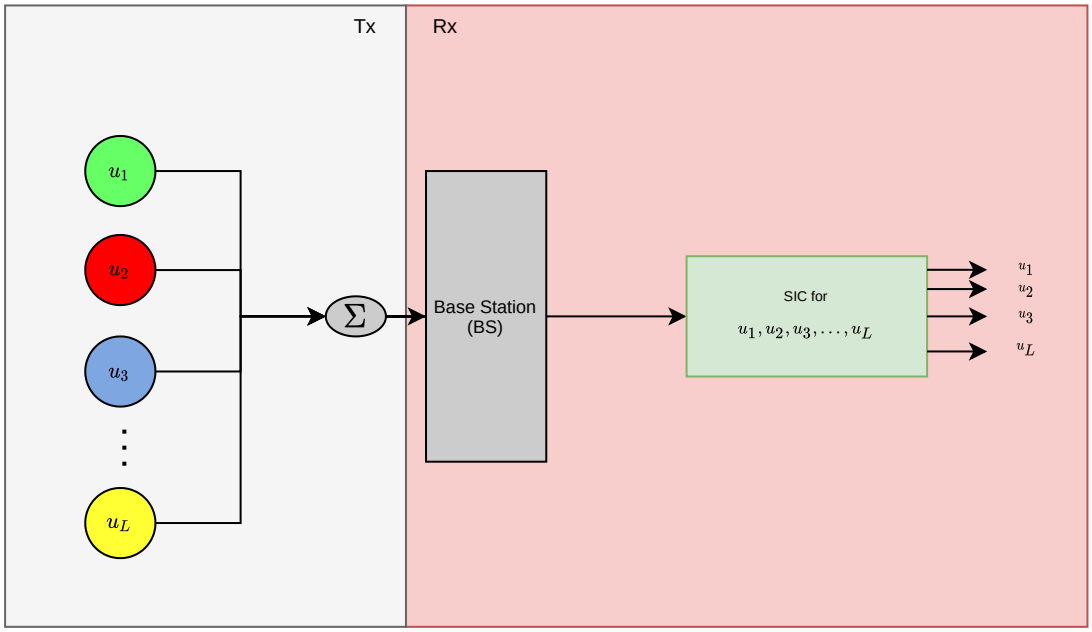
Multiple analytical expressions for NOMA system performance have been derived under different assumptions and scenarios (WANG; LABEAU; MEI, 2017; WEI et al., 2020; ASSAF et al., 2020b; LEE; KIM, 2019; HE; HU; SCHMEINK, 2019; ALMOHIMMAH; ALRESHEEDI, 2020; KARA; KAYA, 2020b; ASSAF et al., 2019; KARA; KAYA, 2019; GARNIER et al., 2020; LUO et al., 2021; ALDABABSA et al., 2020; HAN et al., 2021; WANG et al.,

Figure 12 – Comparison OMA vs NOMA. Colors indicate how different users share channel resources.



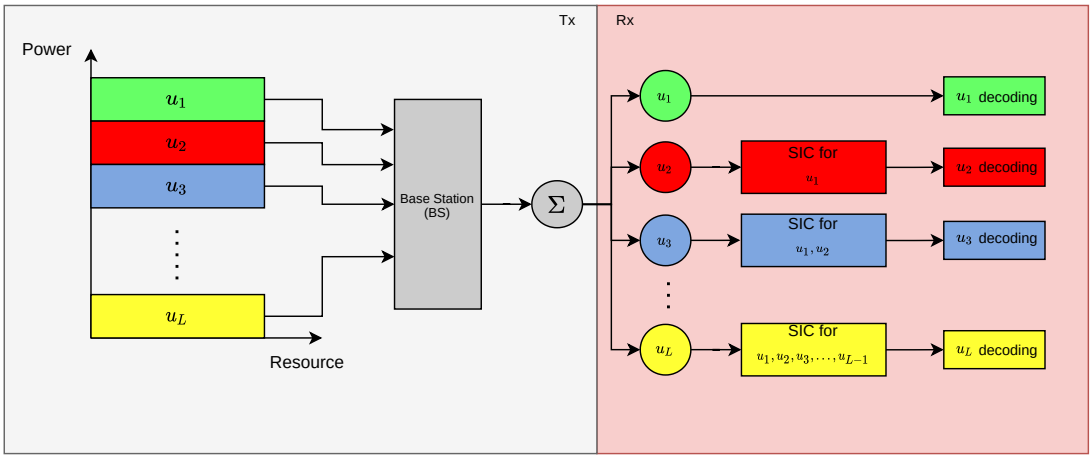
Source: The author (2021).

Figure 13 – Uplink NOMA scheme.



Source: The author (2021).

Figure 14 – Downlink NOMA scheme.



Source: The author (2021).

2021; YEOM et al., 2019; KARA; KAYA, 2018; KARA; KAYA, 2020a). A two-user uplink scheme with QPSK is analyzed in (WANG; LABEAU; MEI, 2017) and with BPSK/QPSK in (WEI et al., 2020). In (HE; HU; SCHMEINK, 2019; LEE; KIM, 2019; ASSAF et al., 2020b; ALMOHIMMAH; ALRESHEEDI, 2020), two downlink users with arbitrary modulation order are considered. In (KARA; KAYA, 2020b), the same scenario is analyzed under imperfect channel state information assumption. The three-user scenario is considered for downlink with QPSK (ASSAF et al., 2019), and space-shift keying (KARA; KAYA, 2019). In (GARNIER et al., 2020), theoretical analysis and experimental validation are done for multi-user downlink. Arbitrary users analysis is considered in (LUO et al., 2021; ALDABABSA et al., 2020; EMIR et al., 2021; HAN et al., 2021). Imperfect SIC modeling is done in (ALDABABSA et al., 2020; CHEN; JIA; NG, 2019; WANG et al., 2021; KARA; KAYA, 2020b) by considering

different approaches for the error propagation phenomenon. In (ALDABABSA et al., 2020), the interference caused by SIC decoding is modeled by a zero-mean Gaussian random variable. A linear model to represent imperfect SIC is employed in (CHEN; JIA; NG, 2019). In (WANG et al., 2021), a precise analysis of imperfect SIC is considered for the two-user downlink case.

To the best of our knowledge, sufficient power spacing between users is generally assumed, to avoid unnecessary interference. In (IRAQI; AL-DWEIK, 2021), power allocation conditions are presented for QAM-based systems while requiring low interference between users. Traditionally, the usage of SIC in scenarios where interference is significant results in error floor (EMIR et al., 2021; ALDABABSA et al., 2020; KARA; KAYA, 2020a), in which inadequate power allocation between users can increase the error propagation in SIC decoding (LUO et al., 2021). In particular, results from (EMIR et al., 2021) suggest that the usage of a NN based decoding scheme is capable of decisively outperforming the traditional SIC decoding. Therefore, we also consider in this chapter the scenario where arbitrary interference is present and an SIC modification is employed to handle decoding in high interference imperfect SIC cases by exploiting channel gain knowledge.

We consider an uplink/downlink scenario where three users using QPSK modulation communicate with a BS. There is no restriction under the channel gain parameters (apart from the ordering restriction). The number of users is chosen aiming to generalize the usual two-user scenario, enabling the presence of more subtle phenomenona, while also preserving lower complexity than the arbitrary number of users case and allowing better results visualization.

We propose a decision region modification for the first step in SIC decoding that is able to justify and replicate the improved performance in DL based methods (EMIR et al., 2021) in situations where channel gain parameters are assumed to be known at the decoder.

The proposed method enables classical techniques to be used in the analysis of the system performance and decoder implementation, which contributes to results interpretability and possible extensions to more generalized scenarios. We also present an analytical expression for the symbol error rate (SER) of the first decoding step where the modified decision region is used, and approximate formulas for the SER of the other users. Comparisons between simulated and analytical results are conducted and we employ our results to justify a power allocation analysis where users are constrained to transmit a limited amount of power. We also present a joint parameter analysis using heatmaps which enables a more general overview and intuition about system performance. Although the analysis is initially done focusing on the uplink scenario, we utilize an theoretical adaptation which enables the extension of the uplink scenario to the downlink one.

The rest of this chapter is structured as follows. We define the uplink communication system and present the proposed decoder in Section 3.1. In Section 3.2, we present an

analytical SER analysis for the proposed decoder. In Section 3.3, we present various SER results for the considered scenarios. We extend previous results to downlink scenario in Section 3.4 and also present simulation results. Finally, in Section 3.5 we discuss final considerations for this chapter.

3.1 COMMUNICATION SYSTEM

We consider a power-domain uplink NOMA system consisting of a single-antenna BS and three single-antenna users. We consider that the complex channel coefficients between the user i and the BS, denoted by h'_i, $i \in \{1, 2, 3\}$, are perfectly known at the BS (ASSAF et al., 2020a). For coherent detection, given that the phase of h'_i is compensated perfectly, the received signal $\boldsymbol{y} = (y_I, y_Q)$ is given by

$$\boldsymbol{y} = \sum_{i=1}^{3} |h'_i| \sqrt{P_i} \boldsymbol{x}_i + \boldsymbol{n}, \tag{24}$$

where P_i is the transmitted power of the user i, $\boldsymbol{x}_i$ is a unit energy QPSK symbol transmitted by user i, $\boldsymbol{x}_i \in \mathcal{A} = \{(\pm B, \pm B)\}$, $B = \sqrt{2}/2$, and $\boldsymbol{n} = (n_I, n_Q)$ is the AWGN noise modeled as a zero-mean Gaussian random variable with variance $N_0/2$ per dimension. We define $h_i \triangleq |h'_i|\sqrt{P_i}$ as the multiplicative term of the symbol $\boldsymbol{x}_i$ and consider h_i fixed during transmission. Users are labeled according to $h_1 > h_2 > h_3$.

The received signal is decoded at the BS using the SIC technique, following the order indicated by the multiplicative term h_i. In the considered scenario, user 1 is decoded first, followed by user 2 and user 3. The estimated symbols for each user $\hat{\boldsymbol{x}}_i$, $i \in \{1, 2, 3\}$, are obtained as follows. The user 1 symbol is directly decoded by treating the symbols of the remaining users as interference

$$\hat{\boldsymbol{x}}_1 = \underset{j}{\operatorname{argmin}} |\boldsymbol{y} - h_1 \boldsymbol{x}_j|^2, \tag{25}$$

where $\boldsymbol{x}_j \in \mathcal{A}$, and the argmin_j notation indicates that we take the corresponding $\boldsymbol{x}_j$ which minimizes the evaluated expression. In the sequel, the interference from user 1 is subtracted from the received vector yielding $\boldsymbol{y}_2 \triangleq \boldsymbol{y} - h_1 \hat{\boldsymbol{x}}_1$. Thus

$$\hat{\boldsymbol{x}}_2 = \underset{j}{\operatorname{argmin}} |\boldsymbol{y}_2 - h_2 \boldsymbol{x}_j|^2. \tag{26}$$

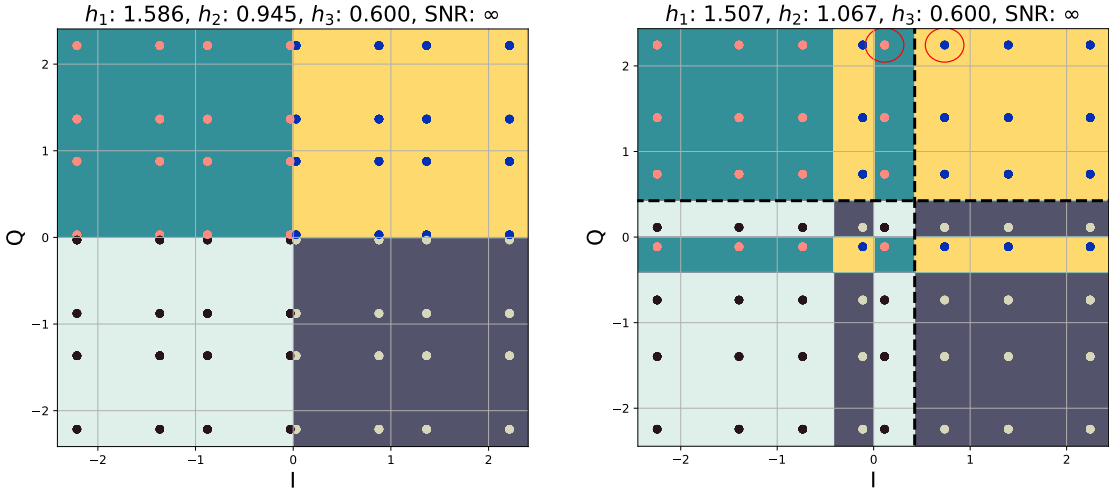
Finally, the estimation of $\boldsymbol{x}_3$ is achieved from $\boldsymbol{y}_3 \triangleq \boldsymbol{y}_2 - h_2 \hat{\boldsymbol{x}}_2$ as

$$\hat{\boldsymbol{x}}_3 = \underset{j}{\operatorname{argmin}} |\boldsymbol{y}_3 - h_3 \boldsymbol{x}_j|^2. \tag{27}$$

3.1.1 Proposed SIC Decoder

We consider the 64 possible signals $\boldsymbol{y}$ in absence of AWGN noise in (24) for different configurations of the parameters h_i (the combination of three QPSK users creates a 64-NOMA constellation). In the three-user case, the inequality $\mathcal{D} : h_1 > h_2 + h_3$ determines the

Figure 15 – Received constellation for the 64 possible combinations of transmitted symbols, for a given h_1, h_2 and h_3, for $\mathscr{D}$ (Figure 15a) and $\overline{\mathscr{D}}$ cases (Figure 15b). Background colors indicate the decision regions.



(a) $\mathscr{D}$: $h_1 > h_2 + h_3$ (b) $\overline{\mathscr{D}}$: $h_1 < h_2 + h_3$

Source: The author (2021).

distribution of the signals in the I/Q plane. If $\mathscr{D}$ is satisfied, all transmitted symbols from user 1 remain in their respective quadrant for any combination of the transmitted symbols from users 2 and 3. If $\mathscr{D}$ is not satisfied, denoted by $\overline{\mathscr{D}}$, the interference from users 2 and 3 is able to modify the quadrant of the symbol transmitted from user 1.

Figure 15 shows, for a given h_1, h_2 and h_3, the received constellations (in the absence of noise) for the cases $\mathscr{D}$ (Figure 15a) and $\overline{\mathscr{D}}$ (Figure 15b). For both figures, the colors of each signal indicate the transmitted symbol from user 1. The background colors indicate the decision regions for the cases $\mathscr{D}$ and $\overline{\mathscr{D}}$ (using the minimum euclidean distance criterion). Figure 15b provides an intuitive explanation to the error floor phenomenon for the case $\overline{\mathscr{D}}$ when the traditional SIC is used. In this scenario, 28 out of 64 symbols are moved outside of their respective quadrants, which results in decoding errors in the traditional SIC algorithm for user 1 (even in the noiseless scenario), and additionally, causes error propagation at the decoding of users 2 and 3.

Therefore, the first step of the proposed decoder is the evaluation of the inequality $\mathscr{D}$. If it is satisfied, we employ the traditional SIC decoding for user 1. Otherwise, we employ the customized decision regions in the SIC decoding of this user, following Figure 15b, and the traditional SIC algorithm for the remaining users, since the correct estimation of $\boldsymbol{x}_1$, $\hat{\boldsymbol{x}}_1$, is sufficient to remove the error floor phenomenon in the decoding of users 2 and 3 (due to $h_2 > h_3$). We do not consider the situation $h_1 = h_2 + h_3$, since this equality means that there exist combinations of transmitted symbols that are mapped into the same received constellation symbol.

It is necessary to determine in Figure 15b the threshold that separates decision

regions with different background colors. For instance, we consider two symbols indicated by red circles in Figure 15b, which are associated with different transmitted symbols from user 1. The in-phase (I) components of these symbols are $Bh_1 - Bh_2 + Bh_3$ and $-Bh_1 + Bh_2 + Bh_3$, which indicates that the in-phase threshold $L = Bh_3$ separates the corresponding yellow and dark cyan regions. Due to the constellation symmetry, analogous thresholds separate the other regions at $\pm Bh_3$ at both in-phase and quadrature (Q) components, beyond the trivial threshold 0 which separates the quadrants. The thresholds at $I = L$ and $Q = L$ are shown in black dashed lines in Figure 15b.

The procedure `Modified-SIC` is given in Algorithm 1. The inputs are the parameters h_1, h_2, h_3, and $\boldsymbol{y} = (y_I, y_Q)$, and the outputs are the decoded symbols $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3$. At the given pseudocode, line 1 evaluates the inequality $\mathscr{D}$ and decides which decision regions are adopted for user 1. The loop in line 4 seeks to obtain $\hat{\boldsymbol{x}}_1$ by determining in which of the 16 disjoint decision regions in Figure 15b $\boldsymbol{y}$ lies. For such, the **If** in line 7 performs reflections in I/Q components enabling the evaluation of boolean variables reg_1, reg_2, reg_3, reg_4 in lines 14, 15, 16 and 17. The variable reg_k indicates that $\boldsymbol{y}$ belongs to the k-th region of the same background color in Figure 15b (there are four disjoint regions of the same background color, which is always true for the $\overline{\mathscr{D}}$ case). The rest of the procedure from line 20 is identical to the traditional SIC.

Since Algorithm 1 fixes the number of users as three, the application of the proposed technique in scenarios with more users may be achieved by employing auxiliary techniques such as pairing algorithms (RAUNIYAR; ENGELSTAD; ØSTERBØ, 2020). In a typical scenario, N users are split into clusters, each one with a subset of users. The proposed method allows the formation of clusters of three users with high interference.

3.2 SER ANALYSIS

In this section, we present a closed-form SER expression, with respect to the parameters h_i and the SNR, for user 1, and SER approximations for users 2 and 3 (following some simplifications presented in (KARA; KAYA, 2020b)). As shown in Section 3.3, the performance of user 1 has high influence on the performance of the other users due to SIC error propagation. Thus, the SER for user 1 serves as an indicator to the overall performance.

We consider different expressions conditioned to the veracity of the inequality $\mathscr{D}$. Let $P(c_1)$ be the probability of decoding a correct symbol for user 1. We first compute the probability of a correct decoding for user 1 for the case $\overline{\mathscr{D}}$, by initially fixing the transmitted symbol, $P(c_1 \mid \overline{\mathscr{D}}, \boldsymbol{x}_1 = (B, B))$. Then, we divide the 16 symbols of the received constellation (considering $\boldsymbol{x}_1 = (B, B)$) in ten cases, as indicated in Figure 16. We assume that all symbols have equal probability of being transmitted. By symmetry, $P(c_1 \mid \overline{\mathscr{D}}) = P(c_1 \mid \overline{\mathscr{D}}, \boldsymbol{x}_1 = (B, B))$. Then,

Algorithm 1: Modified-SIC

Input: $\boldsymbol{y}, h_1, h_2, h_3$
Output: $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3$

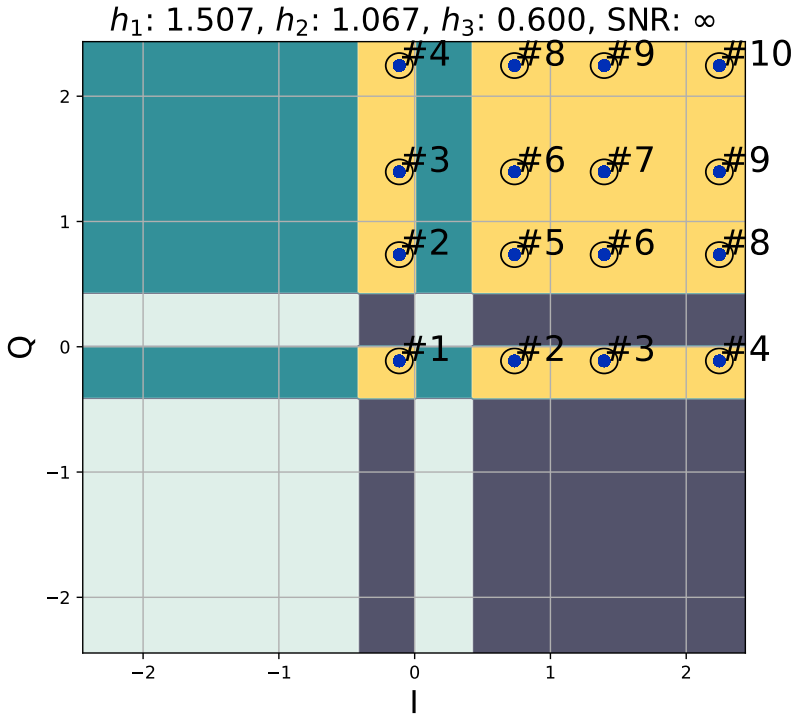
if $\mathscr{D} : h_1 > h_2 + h_3$ **then**
  $\hat{\boldsymbol{x}}_1 \leftarrow \text{argmin}_j \, |\boldsymbol{y} - h_1 \boldsymbol{x}_j|^2$ // $x_j \in \mathscr{A}$
else
  for j = 1 to 4 do
    $y'_I \leftarrow y_I$ // in-phase component
    $y'_Q \leftarrow y_Q$ // quadrature component
    if j = 2 then
      $y'_I \leftarrow -y'_I$
    else if j = 3 then
      $y'_Q \leftarrow -y'_Q$
    else if j = 4 then
      $y'_I \leftarrow -y'_I$
      $y'_Q \leftarrow -y'_Q$
    // reg_k is a boolean variable which indicates if (y'_I, y'_Q) belongs to region k.
    $reg_1 \leftarrow (y'_I > L$ **and** $y'_Q > L)$
    $reg_2 \leftarrow (y'_I < 0$ **and** $y'_I > -L$ **and** $y'_Q > L)$
    $reg_3 \leftarrow (y'_I > L$ **and** $y'_Q < 0$ **and** $y'_Q > -L)$
    $reg_4 \leftarrow (y'_I < 0$ **and** $y'_I > -L$ **and** $y'_Q < 0$ **and** $y'_Q > -L)$
    if reg_1 **or** reg_2 **or** reg_3 **or** reg_4 **then**
      $\hat{\boldsymbol{x}}_1 \leftarrow \boldsymbol{x}_j$
$\boldsymbol{y}_2 \leftarrow \boldsymbol{y} - h_1 \hat{\boldsymbol{x}}_1$
$\hat{\boldsymbol{x}}_2 \leftarrow \text{argmin}_j \, |\boldsymbol{y}_2 - h_2 \boldsymbol{x}_j|^2$
$\boldsymbol{y}_3 \leftarrow \boldsymbol{y}_2 - h_2 \hat{\boldsymbol{x}}_2$
$\hat{\boldsymbol{x}}_3 \leftarrow \text{argmin}_j \, |\boldsymbol{y}_3 - h_3 \boldsymbol{x}_j|^2$

$$P(c_1 \mid \overline{\mathscr{D}}) = \frac{1}{16} \sum_{\boldsymbol{x}_a \in \mathscr{A}} \sum_{\boldsymbol{x}_b \in \mathscr{A}} P(c_1 | \overline{\mathscr{D}}, \boldsymbol{x}_1 = (B, B), \boldsymbol{x}_2 = \boldsymbol{x}_a, \boldsymbol{x}_3 = \boldsymbol{x}_b) \tag{28}$$

$$\begin{aligned} &= \frac{1}{16} \{ P(c_1 | \#\mathbf{1}) + P(c_1 | \#\mathbf{5}) + P(c_1 | \#\mathbf{7}) + P(c_1 | \#\mathbf{10}) \\ &+ 2[P(c_1 | \#\mathbf{2}) + P(c_1 | \#\mathbf{3}) + P(c_1 | \#\mathbf{4}) + P(c_1 | \#\mathbf{6}) \\ &\quad + P(c_1 | \#\mathbf{8}) + P(c_1 | \#\mathbf{9})] \}. \end{aligned} \tag{29}$$

Expressions for each probability in (29) are shown in (30)-(39), at the top of next page, with respect to h_1, h_2, h_3 and the SNR, defined as SNR $\triangleq \rho \triangleq 1/N_0$. For each of the ten cases, we disregard the probability of the AWGN noise moving a symbol between disjoint regions with the same background color. This is an adequate consideration for practical SNR situations, as is shown in Section 3.3, in which we compare analytical and simulation results.

Figure 16 – Resulting constellation by fixing $\boldsymbol{x}_1 = (B, B)$, case $\overline{\mathcal{D}}$. The ten cases considered in derivation of the SER expression are indicated.



Source: The author (2021)

$$P(c_1|\#\mathbf{1}) = P(-Bh_3 < B(h_1 - h_2 - h_3) + n_I < 0)\, P(-Bh_3 < B(h_1 - h_2 - h_3) + n_Q < 0) \tag{30}$$
$$= \left(Q(-B\sqrt{2\rho}(h_1 - h_2)) - Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3))\right)^2$$

$$P(c_1|\#\mathbf{2}) = P(Bh_3 < B(h_1 - h_2 + h_3) + n_I < \infty)\, P(-Bh_3 < B(h_1 - h_2 - h_3) + n_Q < 0) \tag{31}$$
$$= (Q(-B\sqrt{2\rho}(h_1 - h_2)))(Q(-B\sqrt{2\rho}(h_1 - h_2)) - Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3)))$$

$$P(c_1|\#\mathbf{3}) = P(Bh_3 < B(h_1 + h_2 - h_3) + n_I < \infty)\, P(-Bh_3 < B(h_1 - h_2 - h_3) + n_Q < 0) \tag{32}$$
$$= (Q(B\sqrt{2\rho}(2h_3 - h_1 - h_2)))\,(Q(-B\sqrt{2\rho}(h_1 - h_2)) - Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3)))$$

$$P(c_1|\#\mathbf{4}) = P(Bh_3 < B\,(h_1 + h_2 + h_3) + n_I < \infty)\, P(-Bh_3 < B(h_1 - h_2 - h_3) + n_Q < 0) \tag{33}$$
$$= (Q(-B\sqrt{2\rho}(h_1 + h_2)))\,(Q(-B\sqrt{2\rho}(h_1 - h_2)) - Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3)))$$

$$P(c_1|\#\mathbf{5}) = P(Bh_3 < B(h_1 - h_2 + h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 - h_2 + h_3) + n_Q < \infty) \tag{34}$$
$$= \left(Q(-B\sqrt{2\rho}(h_1 - h_2))\right)^2$$

$$P(c_1|\#\mathbf{6}) = P(Bh_3 < B(h_1 + h_2 - h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 - h_2 + h_3) + n_Q < \infty) \tag{35}$$
$$= (Q(B\sqrt{2\rho}(2h_3 - h_1 - h_2)))(Q(-B\sqrt{2\rho}(h_1 - h_2)))$$

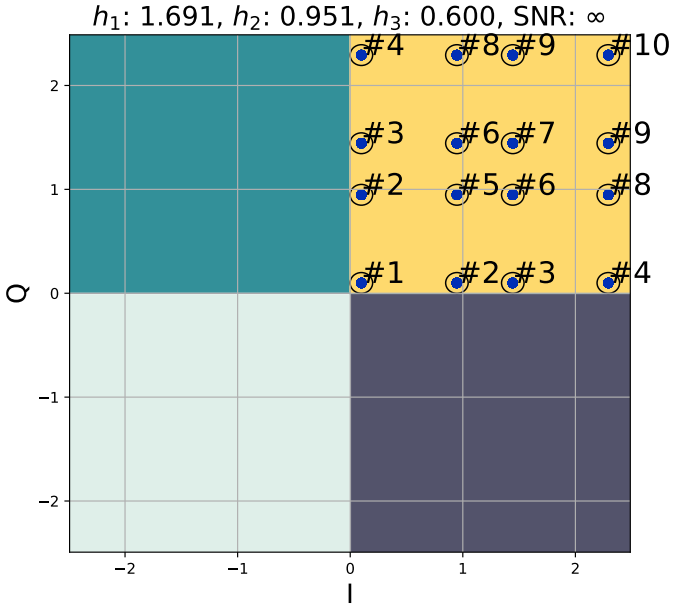
$$P(c_1|\#\mathbf{7}) = P(Bh_3 < B(h_1 + h_2 - h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 + h_2 - h_3) + n_Q < \infty) \tag{36}$$
$$= \left(Q(B\sqrt{2\rho}(2h_3 - h_1 - h_2))\right)^2$$

$$P(c_1|\#\mathbf{8}) = P(Bh_3 < B(h_1 + h_2 + h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 - h_2 + h_3) + n_Q < \infty) \tag{37}$$
$$= (Q(-B\sqrt{2\rho}(h_1 + h_2)))(Q(-B\sqrt{2\rho}(h_1 - h_2)))$$

$$P(c_1|\#\mathbf{9}) = P(Bh_3 < B(h_1 + h_2 + h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 + h_2 - h_3) + n_Q < \infty) \tag{38}$$
$$= (Q(-B\sqrt{2\rho}(h_1 + h_2)))(Q(B\sqrt{2\rho}(2h_3 - h_1 - h_2)))$$

$$P(c_1|\#\mathbf{10}) = P(Bh_3 < B(h_1 + h_2 + h_3) + n_I < \infty)\, P(Bh_3 < B(h_1 + h_2 + h_3) + n_Q < \infty) \tag{39}$$
$$= \left(Q(-B\sqrt{2\rho}(h_1 + h_2))\right)^2.$$

Figure 17 – Resulting constellation by fixing $\boldsymbol{x}_1 = (B, B)$, case $\mathscr{D}$. The ten cases considered in derivation of the SER expression are indicated.



Source: The author (2021)

Substituting (30)-(39), into (29) yields

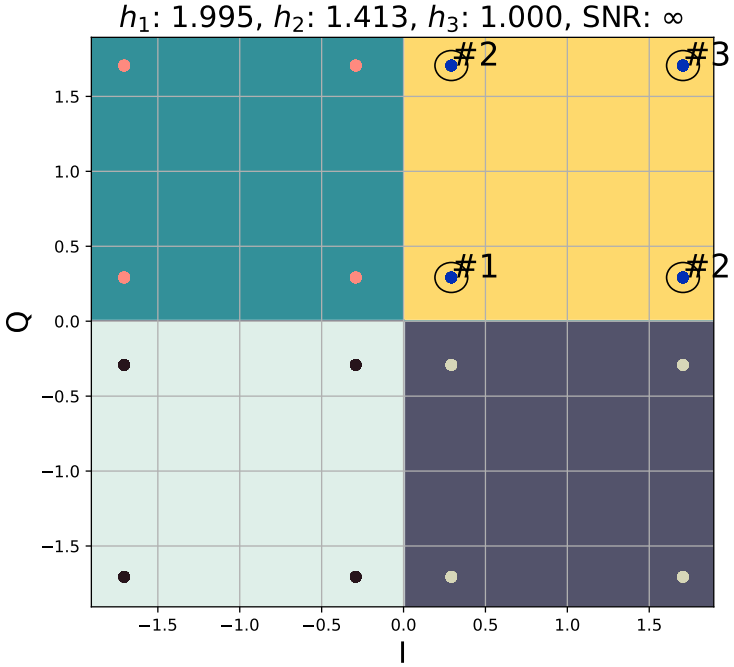
$$\begin{aligned} P(c_1|\overline{\mathscr{D}}) = \frac{1}{16}[&2Q(-B\sqrt{2\rho}(h_1 - h_2)) \\ &- Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3)) \\ &+ Q(-B\sqrt{2\rho}(h_1 + h_2 - 2h_3)) \\ &+ Q(-B\sqrt{2\rho}(h_1 + h_2))]^2. \end{aligned} \tag{40}$$

An analogous procedure is applied to obtain $P(c_1 \mid \mathscr{D})$, where the probabilities in (29) are indicated in Figure 17. This calculation is performed in (HAN et al., 2021, Eq. (58)) yielding

$$\begin{aligned} P(c_1|\mathscr{D}) = \frac{1}{16}[&Q(-B\sqrt{2\rho}(h_1 + h_2 + h_3)) \\ &+ Q(-B\sqrt{2\rho}(h_1 - h_2 + h_3)) \\ &+ Q(-B\sqrt{2\rho}(h_1 + h_2 - h_3)) \\ &+ Q(-B\sqrt{2\rho}(h_1 - h_2 - h_3))]^2. \end{aligned} \tag{41}$$

Expressions (40) and (41) provide a SER expression which covers all inputs for Algorithm 1. We now procede to compute approximate analytical expressions for users 2 and 3. Let $P(e_1)$ be the probability of decoding an erroneous symbol for user 1. The conditional SER for user 1 is $P(e_1 \mid X) = 1 - P(c_1 \mid X)$, $X \in \{\mathscr{D}, \overline{\mathscr{D}}\}$. To derive an approximate expression for the conditional probability of a correct decoding for user 2, we consider as a necessary condition

Figure 18 – Received constellation $\boldsymbol{y}_2$ for all 16 combinations of transmitted symbols from users 2 and 3. We consider that the transmitted symbols from user 1 are correctly decoded and subtracted from the received signal.



Source: The author (2021)

the correct decoding for user 1. This consideration is done in (KARA; KAYA, 2020b) in a two-user scenario in which it is argued that by considering $\hat{\boldsymbol{x}}_1 \neq \boldsymbol{x}_1$, the subtraction of vector $h_1\hat{\boldsymbol{x}}_1$ in line (20) of a procedure analogous to Algorithm 1, the energy of the subtracted vector is bigger than the energy of the user 2, causing a decoding error. Therefore,

$$P(c_2|X) \approx P(c_2|c_1)P(c_1|X), \quad X \in \{\mathcal{D}, \overline{\mathcal{D}}\}, \tag{42}$$

where $P(c_1|X)$ is given either by (40) or (41), for cases $\overline{\mathcal{D}}$ and $\mathcal{D}$, respectively, and $P(c_2|c_1)$ refers to the conditional probability of a correct decoding for user 2, given a correct decoding for user 1. This probability corresponds to the same uplink NOMA scenario, but considering only two users, since

$$\boldsymbol{y} - h_1\boldsymbol{x}_1 \triangleq \boldsymbol{y}_2 = h_2\boldsymbol{x}_2 + h_3\boldsymbol{x}_3 + \boldsymbol{n}. \tag{43}$$

The signal constellation from (43) is shown in Figure 18. The calculation of $P(c_2|c_1)$ is a simplified version of $P(c_1|\mathcal{D})$. This is in (HAN et al., 2021, Eq. (14)):

$$\begin{aligned} P(c_2 \mid c_1) =& \frac{1}{4}\sum_{\boldsymbol{x}_a \in \mathcal{A}} P(c_2 \mid c_1, \boldsymbol{x}_2 = (B,B), \boldsymbol{x}_3 = \boldsymbol{x}_a) \\ =& \frac{1}{4}\{P(c_2|c_1, \#\mathbf{1}) + 2P(c_2|c_1, \#\mathbf{2}) + P(c_2|c_1, \#\mathbf{3})\} \\ =& \frac{1}{4}[Q(-B\sqrt{2\rho}(h_2 - h_3)) + Q(-B\sqrt{2\rho}(h_2 + h_3))]^2. \end{aligned} \tag{44}$$

Expressions for probabilities indicated in (44) are

$$P(c_2|c_1,\#\mathbf{1}) = P(0 < B(h_2-h_3)+n_I < \infty)P(0 < B(h_2-h_3)+n_Q < \infty) \quad (45)$$
$$= \left(Q(-B\sqrt{2\rho}(h_2-h_3))\right)^2$$
$$P(c_2|c_1,\#\mathbf{2}) = P(0 < B(h_2-h_3)+n_I < \infty)P(0 < B(h_2+h_3)+n_Q < \infty) \quad (46)$$
$$= Q(-B\sqrt{2\rho}(h_2-h_3))Q(-B\sqrt{2\rho}(h_2+h_3))$$
$$P(c_2|c_1,\#\mathbf{3}) = P(0 < B(h_2+h_3)+n_I < \infty)P(0 < B(h_2+h_3)+n_Q < \infty) \quad (47)$$
$$= Q(-B\sqrt{2\rho}(h_2+h_3))^2.$$

For user 3, we apply again the consideration of correct decoding for users 1 and 2 as a necessary condition to the correcting decoding of user 3. Therefore

$$P(c_3|X) \approx P(c_3|(c_2 \cap c_1))P(c_2 \cap c_1|X), \quad X \in \{\mathcal{D}, \overline{\mathcal{D}}\}. \quad (48)$$

This step is not correct in general, although we will empirically verify it for most h_i in Section 3.3. It is possible that decoding of users 1 and 2 are incorrect and user 3 correct. In some cases, SER of user 3 is much lower than users 1 and 2, as shown in Section 3.3. Nevertheless, this consideration simplifies greatly the SER computation and is correct for the majority of cases, thus it is employed here. In the literature, it is commonly considered different restrictions in the problem formulation which simplify the problem. For example, in (HAN et al., 2021), a spacing restriction under the values of h_i is imposed resulting in the above argument correct for all valid h_i.

We consider that $P(c_2 \cap c_1|X) = P(c_2|X)$ and $P(c_3|(c_2 \cap c_1)) = Q^2(-Bh_3\sqrt{2\rho})$ is the SER of the QPSK constellation with symbols at $(\pm Bh_3, \pm Bh_3)$, since:

$$\boldsymbol{y}_2 - h_2\boldsymbol{x}_2 \triangleq \boldsymbol{y}_3 = h_3\boldsymbol{x}_3 + \boldsymbol{n}, \quad (49)$$

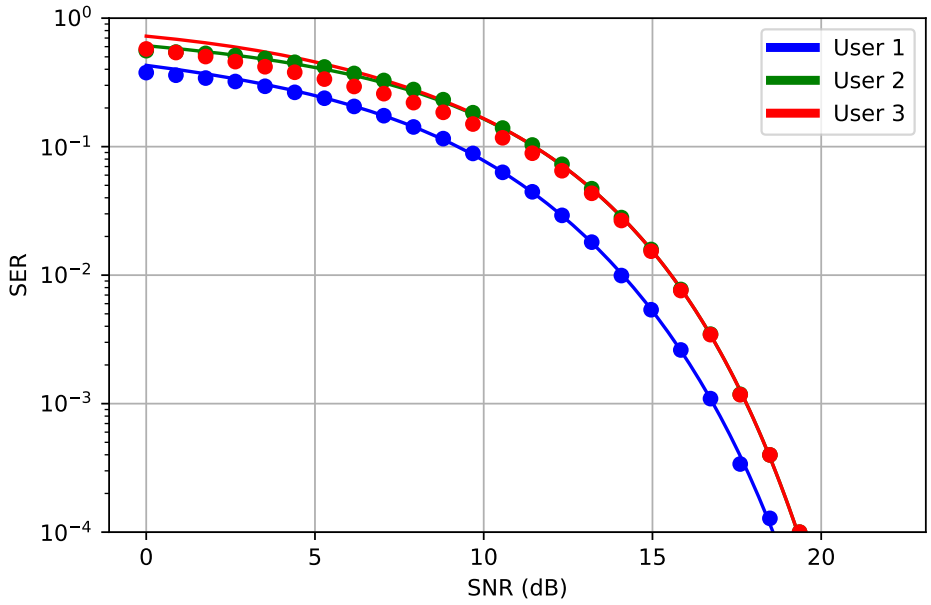
and

$$P(c_3|(c_2 \cap c_1)) = P(0 < Bh_3 + n_I < \infty)P(0 < Bh_3 + n_Q < \infty) \quad (50)$$
$$= \left(Q(-Bh_3\sqrt{2\rho})\right)^2.$$

3.3 RESULTS

In the analyses conducted in this section we consider $h_3 = 1$ and vary the parameters h_1, h_2, and SNR. Different values for h_3 yield similar results. We utilize the convention to define the parameters h_i by the relation $h_i^2 = h_{i+1}^2 + \xi_i$ (dB), $i = 1, 2$, and $h_3^2 = 0$ (dB).

Figure 19 – Simulated and analytical SER comparison for each user considering $\xi_1 = \xi_2 = 3$ dB (case $\overline{\mathcal{D}}$). The solid lines represent the analytical solution, whereas the circled markers indicate the simulation results.



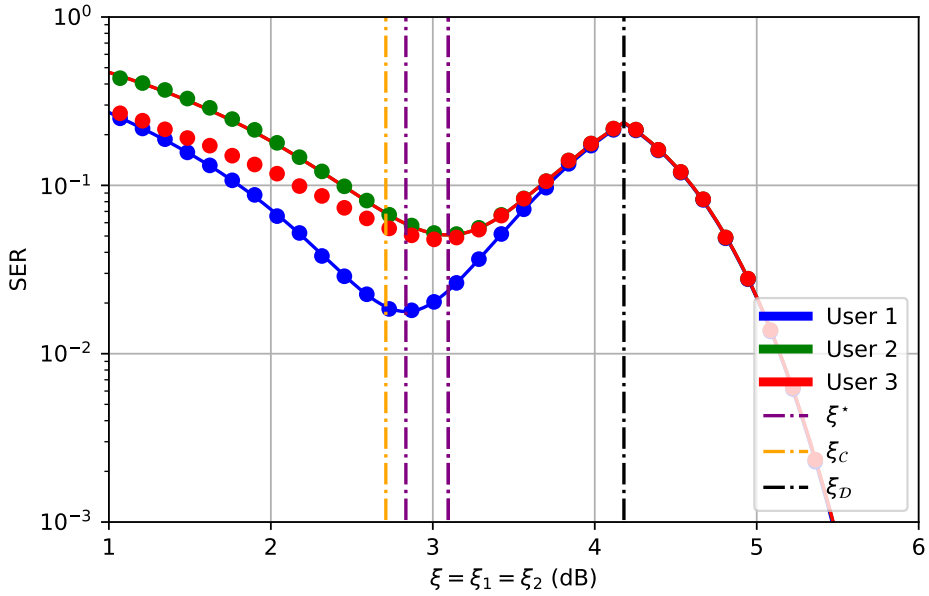
Source: The author (2021).

Therefore, we consider the SER system performance with respect to ξ_1, ξ_2, and SNR, and fix one or two variables resulting in, respectively, a two-dimensional or a one-dimensional plot. For all figures presented in this section, we denote by colors blue, green and red the users 1, 2 and 3, respectively. Circle markers indicate simulated results and solid lines indicate analytical ones presented in Section 3.2, unless otherwise stated.

We consider some scenarios to compare simulated and analytical SER for each user. In the first one, shown in Figure 19, we fix $\xi_1 = \xi_2 = 3$ dB (case $\overline{\mathcal{D}}$) and vary the SNR. In another one, the SNR is fixed at 13 dB and 18 dB, and we consider the restriction $\xi = \xi_1 = \xi_2$ and vary ξ, as shown in Figures 20 and 21, respectively. These figures show a good agreement between analytical and simulation results for users 1 and 2 and some disagreement for user 3 for some choices of ξ in the high SER/low SNR region. As already discussed in Section 3.2, the derived analytical SER considers the correct decoding of the previous users, and therefore, $P(e_1) \leq P(e_2) \leq P(e_3)$, which is not correct for all values of ξ_1 and ξ_2, as shown by the simulated results in Figures 20 and 21. The analytical curves for users 2 and 3 are almost identical.

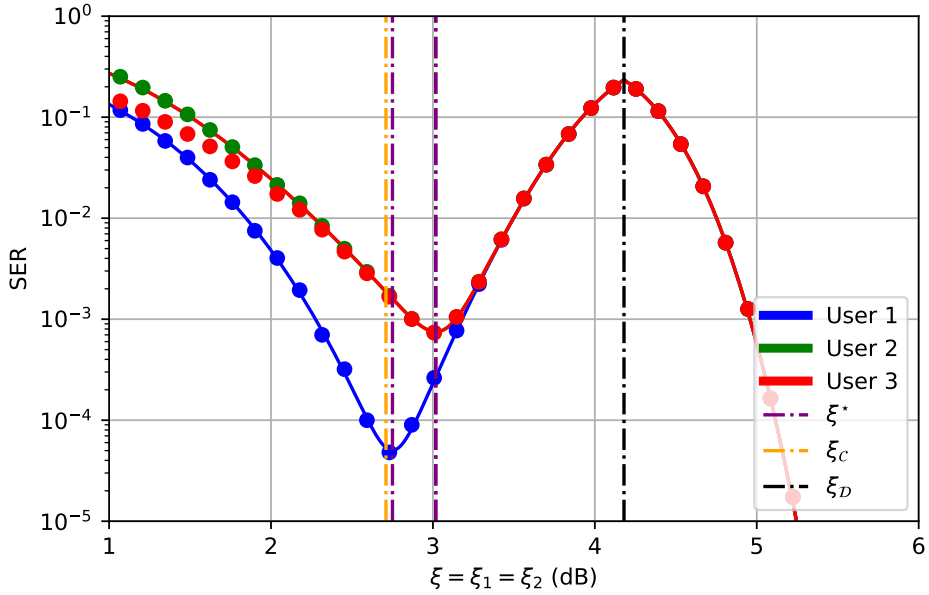
We focus our analysis on Figures 20 and 21, where it is indicated special values of ξ with dashed vertical lines. These values are obtained by employing the restriction $\xi = \xi_1 = \xi_2$ in combination with the specified scenario in each case. The threshold for the inequality $\mathcal{D}$ (indicated in the figure by $\xi_{\mathcal{D}}$) corresponds to $\xi_{\mathcal{D}} = 20\log_{10}\left(\frac{1+\sqrt{5}}{2}\right) \approx 4.18$ dB, which is obtained by combining the inequality $\mathcal{D}$ with the ξ_i definition, under the restriction

Figure 20 – Simulated and analytical SER comparison for each user under the restriction $\xi = \xi_1 = \xi_2$. SNR = 13 dB. The solid lines represent the analytical solution, whereas the circled markers indicate the simulation results.



Source: The author (2021).

Figure 21 – Simulated and analytical SER comparison for each user under the restriction $\xi = \xi_1 = \xi_2$. SNR = 18 dB. The solid lines represent the analytical solution, whereas the circled markers indicate the simulation results.



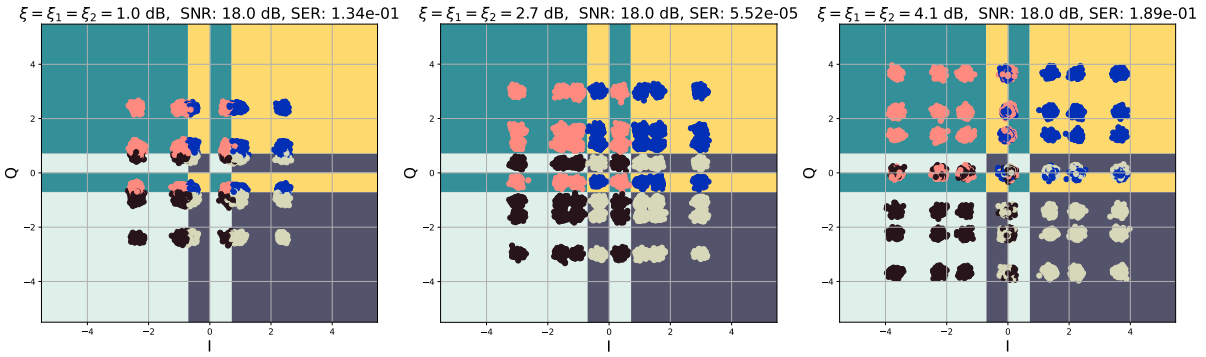
Source: The author (2021).

$\xi = \xi_1 = \xi_2$. The SER for the case $\mathcal{D}$ (region to the right of $\xi_{\mathcal{D}}$) decreases monotonically with increasing ξ. For the case $\overline{\mathcal{D}}$ (region to the left of $\xi_{\mathcal{D}}$), the SER dependence on ξ is not monotonic, with an optimal spacing, denoted by $\xi^\star$, for each user. It is also important to note that the optimal spacing $\xi^\star$ for user 1 does not always correspond to optimal $\xi^\star$ for other users. A meaningful result obtained by comparing Figures 20 and 21 is the dependence of the optimal spacing with the SNR. For user 1 and SNR = 13 dB (Figure 20), $\xi^\star = 2.83$ dB, while for SNR = 18 dB (Figure 21), $\xi^\star = 2.75$ dB. Both values agree with the simulations and can be obtained numerically from the provided SER expression and employing the restriction $\xi = \xi_1 = \xi_2$.

We also show in Figures 20 and 21 a special value $\xi_{\mathcal{C}} = 20\log_{10}\left(\frac{1+\sqrt{3}}{2}\right) \approx 2.71$ dB, which refers to a choice of h_i such that the points indicated by $\#\mathbf{1}$, $\#\mathbf{2}$, $\#\mathbf{3}$ and $\#\mathbf{4}$ in Figure 17 are symmetrically positioned in their respective regions, under the restriction $\xi = \xi_1 = \xi_2$. In other words, if we choose ξ such that the point indicated by $\#\mathbf{1}$ in Figure 17 is at the coordinate $(\frac{-L}{2}, \frac{-L}{2})$, the center of its respective square, we obtain a suboptimal SER. This can be explained by the fact that the placement of a constellation point must be done by adjusting some h_i, which also displaces other constellation points to unfavorable positions. The h_i adjustment can be done in a practical setting if the channel gains h_i' are assumed to be known, since it is possible to select the transmitted power for each user and employ $h_i \triangleq |h_i'|\sqrt{P_i}$. This result shows that the positioning of the constellation points in Figure 15b should be done according to the SNR. The disjoint decision regions add complexity to the optimal constellation design.

It should be noted that under case $\overline{\mathcal{D}}$, we have a non-monotonic behaviour. This can be explained by the atypical disjoint decision regions. Figure 22 shows three receiving constellations at the base station, similar to those shown in Figure 15, for extreme ξ_i values. In Figure 22a, $\xi = 1$ dB, results in different user 1 symbols mapped to overlapping positions under the specified SNR, at regions near $|I| = L$ or $|Q| = L$, resulting in poor performance. In Figure 22b, $\xi = 2.7$ dB, which is close to the optimal $\xi^\star$ value, resulting in well-spaced constellation points and good performance. In Figure 22c, we consider a value of ξ close to the threshold $\xi_{\mathcal{D}}$ ($\xi = 4.1$ dB), resulting in poor performance due to different symbols being mapped to overlapping positions near $I = 0$ or $Q = 0$.

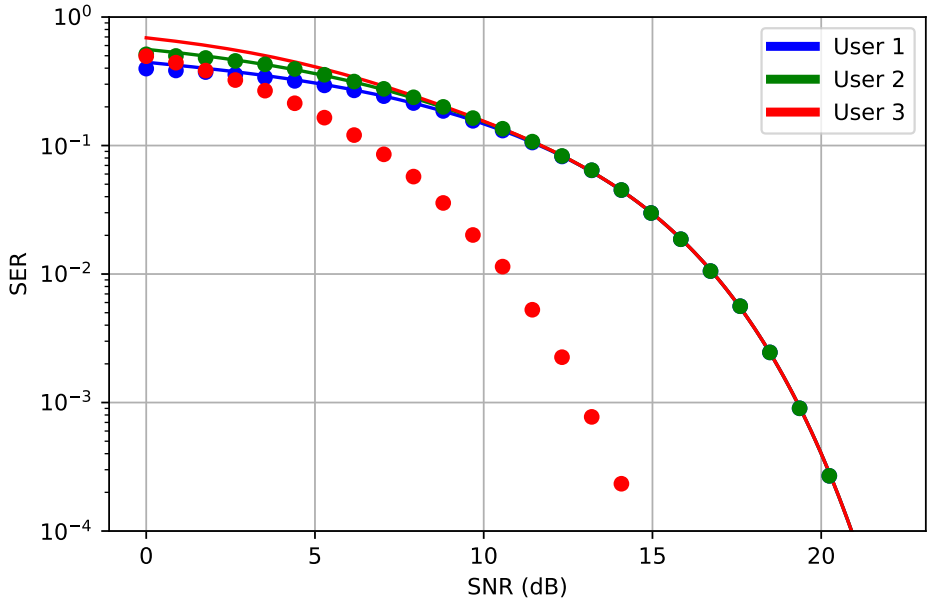
We now consider a scenario in which the SER of user 3 is significantly lower than that of users 1 and 2, which occurs when $h_1 \approx h_2$ and h_2 is sufficiently larger than h_3. Figure 23 shows the SER versus SNR of each user for $\xi_1 = 1.5$ dB and $\xi_2 = 5$ dB (case $\overline{\mathcal{D}}$). A possible explanation for this phenomenon is that a decoding error at user 1 corresponds to a vector addition of $-h_1\hat{\boldsymbol{x}}_1$, $\hat{\boldsymbol{x}}_1 \neq \boldsymbol{x}_1$, which can provoke an error at user 2, $\hat{\boldsymbol{x}}_2 \neq \boldsymbol{x}_2$, and subsequent addition of $-h_2\hat{\boldsymbol{x}}_2$. It is also important to note that the errors at users 1 and 2 are correlated due to the nature of the sequential SIC decoding, which can result in a cancellation of the added error vectors, which in practice means that the symbol of user 3 is corrupted by



(a) $\xi = 1.0$ dB. (b) $\xi = 2.7$ dB. (c) $\xi = 4.1$ dB.

Figure 22 – Received constellation for the 64 possible combinations of transmitted symbols under constraint $\xi = \xi_1 = \xi_2$, and $\xi < 4.18$ dB, reflecting case $\overline{\mathcal{D}}$. SNR = 18 dB. Refer to Figure 21 for SER curves for the same scenario.

Figure 23 – Simulation and analytical SER comparison for $\xi_1 = 1.5$ dB and $\xi_2 = 5$ dB, and therefore case $\overline{\mathcal{D}}$. The solid lines represent the analytical solution, whereas the circled markers indicate the simulation results.

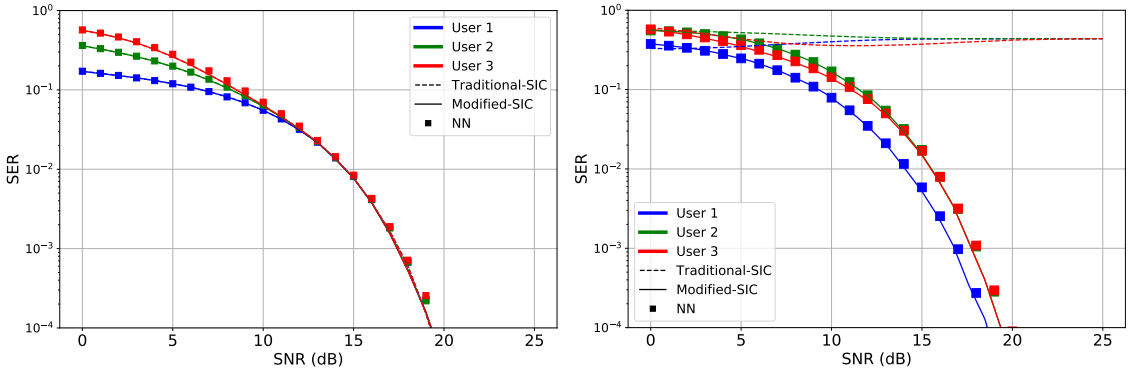


Source: The author (2021).

an error propagation factor proportional to $B(h_1 - h_2)$, which is not necessarily sufficient to cause an error at user 3. The exact scenarios where the analytical expression for user 3 is inadequate are discussed later in this section. Recall that the discrepancy between analytical and simulations results for user 3 is due to the assumption of correct decoding of the previous users.

We show next the SER comparisons for three decoder strategies: Traditional-SIC, `Modified-SIC`, and a NN similar to the presented in (EMIR et al., 2021). Figure 24 shows the SER versus SNR for cases $\mathcal{D}$ and $\overline{\mathcal{D}}$. For case $\mathcal{D}$, shown in Figure 24a for $\xi_1 = \xi_2 = 5$ dB, all decoders have very similar performance. Figure 24b uses $\xi_1 = \xi_2 = 3$ dB and shows

Figure 24 – SER comparison between Traditional-SIC, `Modified-SIC`, and NN decoders. Figure 24a shows the scenario for $\xi_1 = \xi_2 = 5$ dB, case $\mathscr{D}$. Figure 24b shows the scenario for $\xi_1 = \xi_2 = 3$ dB, case $\overline{\mathscr{D}}$.



(a) $\mathscr{D} : h_1 = 3.162, h_2 = 1.778, h_3 = 1.000$ (b) $\overline{\mathscr{D}} : h_1 = 1.995, h_2 = 1.413, h_3 = 1.000$

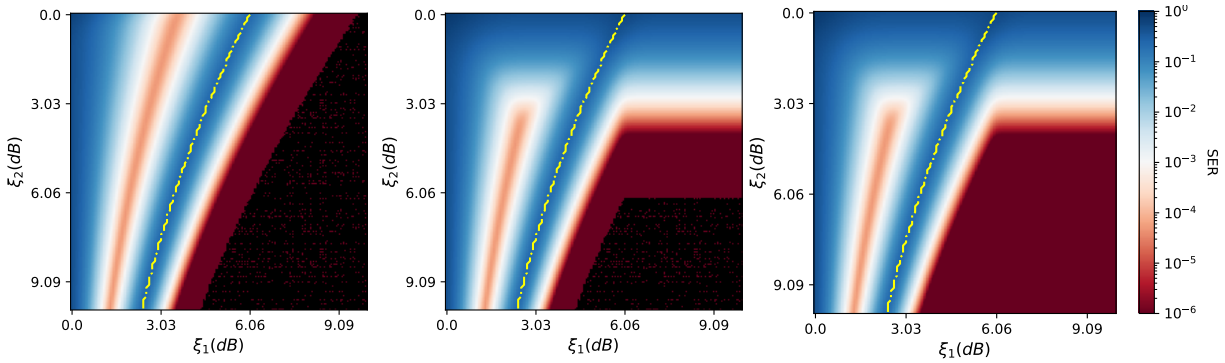
Source: The author (2021).

that the `Modified-SIC` decisively outperforms the Traditional-SIC for case $\overline{\mathscr{D}}$, which confirms that the proposed customized decision regions enable better performance in some cases. The NN model performs almost identically to the `Modified-SIC` decoder, which suggests that the NN boost in performance over the Traditional-SIC reported in (EMIR et al., 2021) occurs due to the learning of decision regions similar to those shown in Figure 15b.

Since the model in (EMIR et al., 2021) deals with NOMA decoding in scenarios in which ξ_i is low (high user interference case), we compare the proposed decoder with the NN and argue that it is an alternative to the NN approach for low modulation orders M, in the three-user scenario. According to (EMIR et al., 2021), the NN scheme has a computational complexity which is empirically determined, given as $\mathcal{O}(60 \times 80)$, and is essentially fixed for the scenario investigated therein. The proposed decoder has complexity $\mathcal{O}(M^2)$, valid for the three-user scenario, since for user 1 decoding we have to loop over its M possible transmission hypotheses (line 4 of Algorithm 1) and perform M operations to check if (y'_I, y'_Q) belongs to each one of the possible M regions (lines 14-17), in the worst case. Our decoder also provides theoretically guaranteed performance and a better understanding of the performed decoding steps. As a consequence of the similar performance in the proposed scenario, it could also be argued that such decoder can be viewed as a theoretical estimate or reference to the expected performance of the NN model.

The heatmaps for the analytical SER expressions for users 1, 2, and 3 are shown in Figures 25a, 25b, and 25c, respectively. The heatmaps aim to show the overall performance behavior by considering joint changes in parameters ξ_1 and ξ_2, given a fixed SNR = 18 dB. It is also indicated (in dashed line) the threshold for the inequality $\mathscr{D}$. For all users, these figures indicate that the performance is poor for h_i configurations close to the threshold ($h_1 \approx h_2 + h_3$) and it is possible to obtain better results by moving away from the threshold. It can be seen that low ξ_1 values correspond to poor performance, for any ξ_2, since if $h_1 \approx h_2$

Figure 25 – Analytical SER heatmaps for each user. The dashed line indicates the inequality $\mathscr{D}$. SNR = 18 dB. Black color indicates SER $< 10^{-6}$.



(a) User 1. (b) User 2. (c) User 3.

Source: The author (2021).

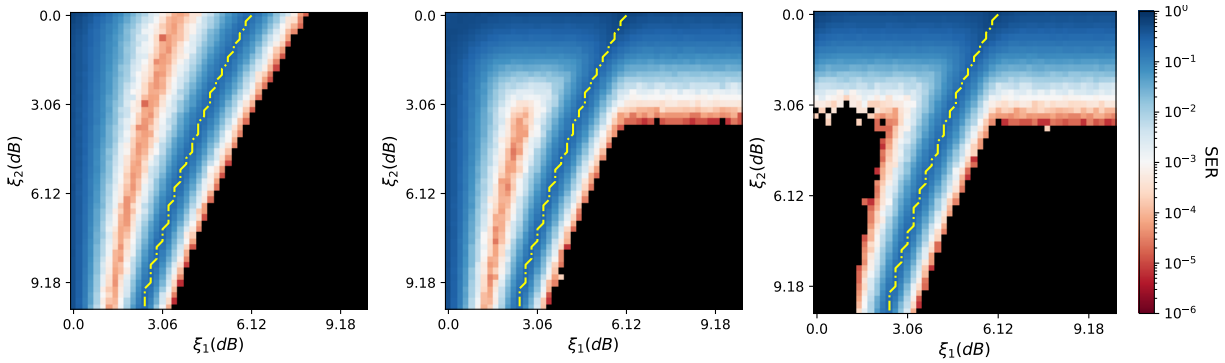
the interference in the decoding of user 1 is high, and errors in first step of SIC decoding affect the remaining users (users 2 and 3). In Figures 25b and 25c, a cut-off value for performance of users 2 and 3 is seen at around $\xi_2 \approx 3$ dB for any ξ_1 used, which can be explained by considering that if correct user 1 decoding occurs, the ξ_2 parameter controls the interference between the remaining users. Due to the employed derivation hypothesis $P(e1) \leq P(e2) \leq P(e3)$, the regions where good performance for users 2 and 3 occur are necessarily a subset of the regions where good performance occurred for user 1. This behavior can be seen in the theoretical heatmaps.

For validation purposes, we also present simulated heatmaps, which confirm the accuracy of the SER expressions for users 1 and 2 (Figures 26a and 26b), and outline where disagreements occur for user 3 (Figure 26c). The simulations are performed for a low number of bits (10^5) per cell, on a 50×50 grid due to hardware limitations. As discussed before, it is possible that user 3 has lower SER (for a given SNR) than users 1 and 2, which is not considered in the analytical expressions. This is shown in the simulated heatmap for user 3 (Figure 25c) for $\xi_1 < 2$ dB and $\xi_2 > 3$ dB. For these values of ξ_1 and ξ_2, the proposed analytical expression for user 3 is invalid. Figures 26a, 26b and 26c confirms the overall behaviour of the theoretical heatmaps shown in Figures 25a, 25b and 25c. Users 1 and 2 have excellent agreement between simulation and theoretical results.

The average SER heatmaps ($\frac{1}{3}\sum_{i=1}^{3} P(e_i \mid X)$, $X \in \{\mathscr{D}, \overline{\mathscr{D}}\}$), are shown in Figure 27 for SNR = 18 dB. Figure 27a presents the analytical result and Figure 27b the simulated one. It is seen that both versions agree for all considered ξ_i. Thus, the average SER can be expressed with the provided analytical expressions. Similar experiments, not shown here, were also conducted for different SNRs to confirm the result.

The analytical expressions can be employed by the users to adjust their own transmitted power P_i aiming to minimize the SER, given that the channel gain of each user is known. This situation is plausible in a scenario where the base station communicates with

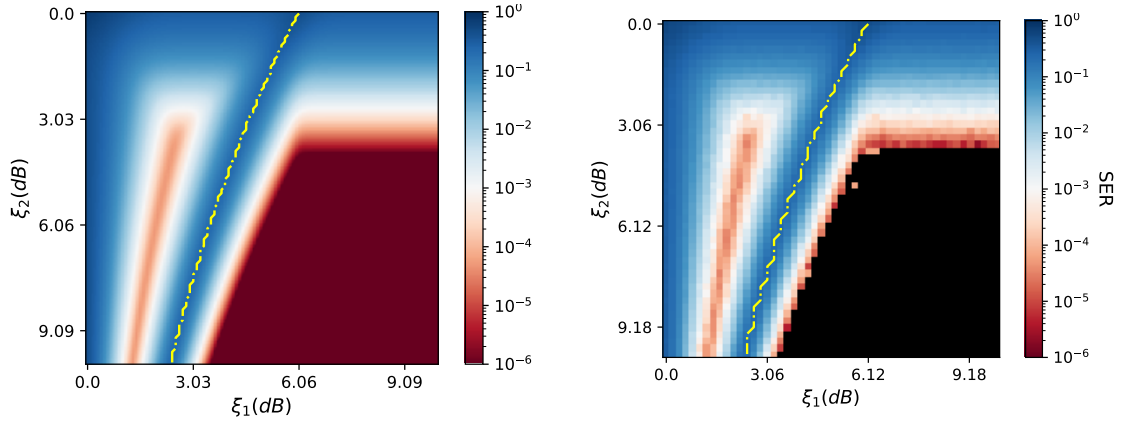
Figure 26 – Simulated SER heatmaps for each user. The dashed line indicates the inequality $\mathscr{D}$. SNR = 18 dB. Black color indicates SER $< 10^{-6}$.



(a) User 1. (b) User 2. (c) User 3.

Source: The author (2021).

Figure 27 – Average SER heatmap comparison between analytical and simulation. SNR = 18 dB.



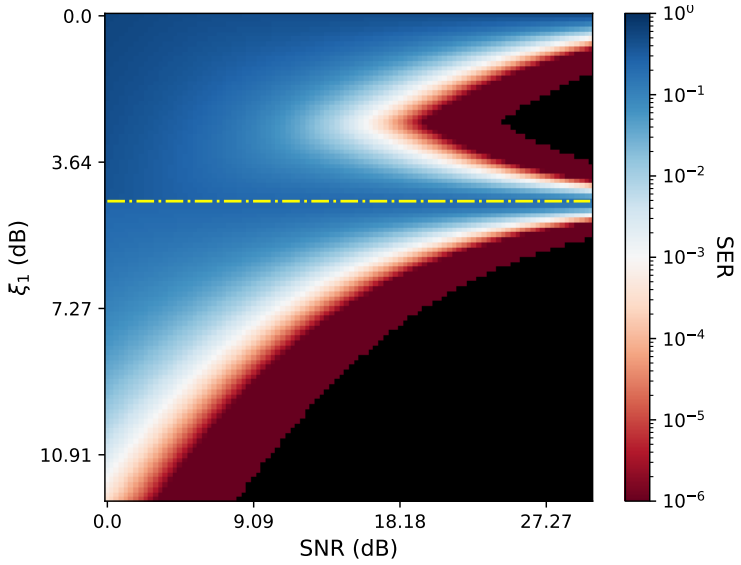
(a) Analytical. (b) Simulation.

Source: The author (2021).

the users informing the current channel gain obtained from pilot symbol transmissions. In the following, we present an analysis targeting user 1 SER, which is chosen due to the strong influence on the overall system performance, although any user could have been chosen. Figure 28 shows the heatmap for user 1 with respect to ξ_1 and SNR, in which ξ_2 is fixed at 3 dB. This scenario seeks to reflect the situation where h_2, h_3 are known by user 1 and this user seeks to adjust P_1 and consequently ξ_1 to obtain optimal SER for a given SNR (under the assumption that h_1' is known). By inspecting Figure 28 we conclude that different scenarios may be employed according to the SNR. By considering SNR = 25 dB, increasing or decreasing P_1 (and consequently ξ_1) around a reference value, $\xi_1 \approx 4.65$ dB (dashed line) reduce the SER. The reduction of P_1 under SNR = 9 dB does not provide the same gains.

We now consider a restriction on the total transmitted power by considering $\sum_{i=1}^{3} h_i^2 \leq K$, under fixed SNR assumption. We consider a scenario where we seek to optimize the SER of user 1 under the constraint that users 2 and 3 must have SER $\leq 10^{-3}$ in addition

Figure 28 – Analytical SER heatmap for user 1. $\xi_2 = 3$ dB. The dashed line $\xi_1 \approx 4.65$ dB indicates inequality $\mathscr{D}$.



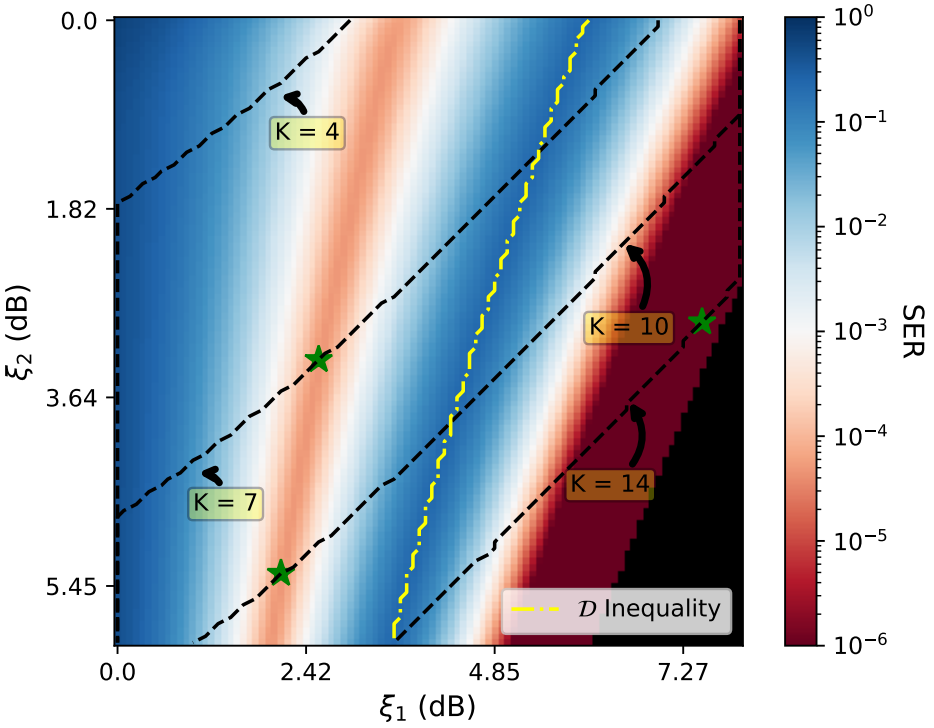
Source: The author (2021).

to the transmitted power restriction. The provided theoretical formulas (40, 41, 44, 50) can be used to quickly find the optimal points under the constraints. We delimit four regions in the SER heatmap of user 1, for $K \in \{4, 7, 10, 14\}$, indicated by black dashed lines in Figure 29, meaning that the power constraint limits the analyzed region to the upper side of the black dashed lines. The green stars in Figure 29 indicate the optimal values for each considered restriction. The yellow dashed line indicates the $\mathscr{D}$ inequality, in which the right side represents case $\mathscr{D}$, and the left side, $\overline{\mathscr{D}}$.

- For $K = 14$, the optimal values are $\xi_1 \approx 7.5$ dB and $\xi_2 \approx 2.9$ dB, which indicates case $\mathscr{D}$.
- For $K = 10$, the optimal SER for user 1 is achieved for $\xi_1 \approx 2.1$ dB and $\xi_2 \approx 5.3$ dB, which indicates $\overline{\mathscr{D}}$. In this case, it is not possible to achieve better performance using the right side of the yellow line, since the performance for users 2 and 3 does not satisfy the constraints.
- For $K = 7$, optimal SER for user 1 is achieved for $\xi_1 \approx 2.5$ dB and $\xi_2 \approx 3.3$ dB, which indicates $\overline{\mathscr{D}}$.
- For $K = 4$, it is impossible to satisfy the SER restriction for users 2 and 3 in the considered scenario.

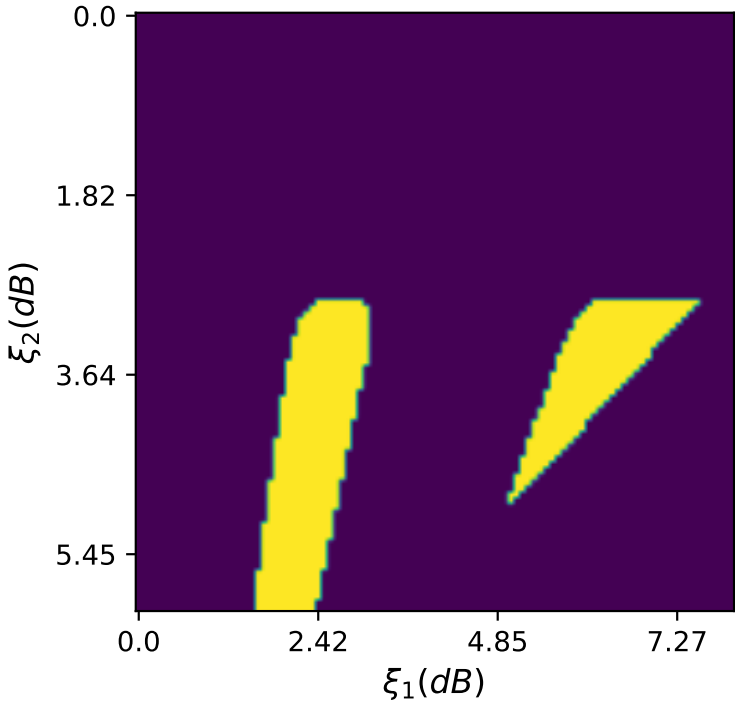
For illustration purposes, we provide the valid search space for ξ_1 and ξ_2 in Figure 30, for the $K = 14$ case. Recall that the search space for optimal user 1 SER is delimited by both the energy restriction (indicated by the black dashed lines) and the minimal performance requirement for users 2 and 3 (SER $\leq 10^{-3}$).

Figure 29 – Analytical SER heatmap for user 1. We fix SNR = 18 dB and employ the restriction $\sum_{i=1}^{3} h_i^2 \leq K$. Four regions are shown for $K \in \{4, 7, 10, 14\}$.



Source: The author (2021).

Figure 30 – Valid ξ_1 and ξ_2 search space for optimal SER of user 1 shown in yellow after power constraint $K = 14$ and SER $\leq 10^{-3}$ for users 2 and 3.



Source: The author (2021).

3.4 DOWNLINK

We now consider the three-user single-antenna NOMA downlink scenario. The received signal for user u, $u \in \{1,2,3\}$, is written as (KARA; KAYA, 2020b)

$$\boldsymbol{y}_u = \sqrt{P_s}|h'_u| \sum_{i=1}^{3} (\sqrt{\alpha_i}\boldsymbol{x}_i) + \boldsymbol{n}_u, \tag{51}$$

where P_s is the transmit power of the base station, h'_u is the channel gain for user u, which is perfectly phase compensated, α_i is the power allocation coefficient for the transmitted signal corresponding to user i, which satisfies $\sum_{i=1}^{3} \alpha_i = 1$, $\boldsymbol{x}_i$ is the unit energy QPSK symbol transmitted to user i and $\boldsymbol{n}_u = (n_I, n_Q)$ is the AWGN noise. We assume that $\alpha_1 > \alpha_2 > \alpha_3$, and the SIC decoding follows this ordering. We define $h_u \triangleq |h'_u|\sqrt{P_s}$ as the multiplicative term of user u.

By analogy to the uplink scenario, we interpret (51) as distinct realizations of (24) for each user u. Therefore, we set

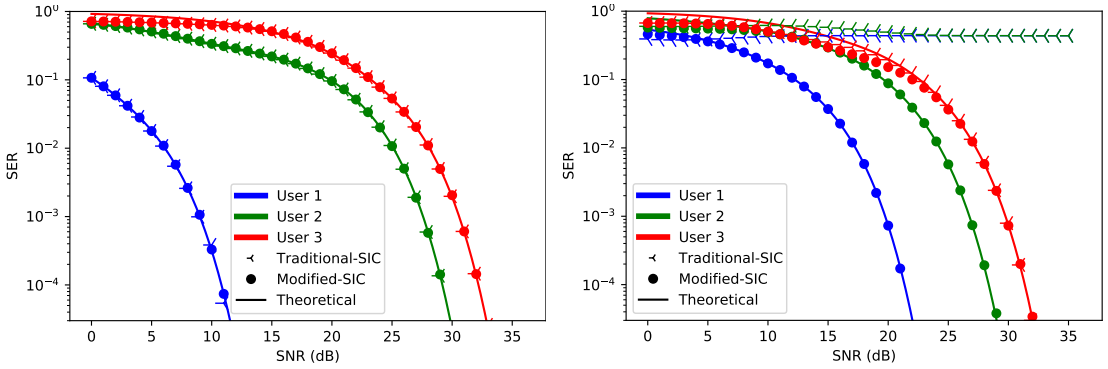
$$\tilde{h_{ui}} \triangleq h_u\sqrt{\alpha_i}, \tag{52}$$

and employ the SER analysis for the uplink scenario shown in Section 3.2 using the resulting nine $\tilde{h_{ui}}$. In the downlink scenario, we employ for user 1 the uplink result $P(e_1|X)$ (40)-(41), $X \in \{\mathscr{D}, \overline{\mathscr{D}}\}$ by setting $h_i = \tilde{h_{1i}}$, for $i = \{1,2,3\}$. For user 2 we employ (42) by setting $h_i = \tilde{h_{2i}}$, and for user 3 we employ (48) by setting $h_i = \tilde{h_{3i}}$.

We verify the adequacy of the conversion given in (52) by comparing the analytical and simulation results under two α_i configurations. We utilize the convention to define the parameters h_u by the relation $h_u^2 = h_{u+1}^2 + \gamma_u$ (dB), $u = \{1,2\}$, and $h_3^2 = 0$ (dB). The power allocation coefficients α_i determine if the corresponding scenario corresponds to $\mathscr{D}$ or $\overline{\mathscr{D}}$ according to inequality $\mathscr{E} : \sqrt{\alpha_1} > \sqrt{\alpha_2} + \sqrt{\alpha_3}$, which follows immediately from inequality $\mathscr{D}$ and (52). Figure 31 shows the comparison results for two scenarios $\mathscr{E}$ and $\overline{\mathscr{E}}$ and confirms that the analysis from Section 3.2 can be employed by converting downlink-uplink gain parameters according to (52). We use in this figure $\gamma_1 = \gamma_2 = 3$ dB.

We next present heatmaps for the downlink scenario. We fix γ_u and present the result in terms of α_i. We set $\gamma_1 = \gamma_2 = 3$ dB and consider the previously presented restrictions for α_i: $\sum_{i=1}^{3} \alpha_i = 1$ and $\alpha_1 > \alpha_2 > \alpha_3$. The analytical and simulated heatmaps are presented in Figures 32 and 33, respectively, where invalid choices of α_i are shown in green. It is shown the inequality $\mathscr{E}$ and the corresponding usage of the `Modified-SIC` decoder. Similar to the uplink scenario, regions closer to the inequality threshold result in poor performances. These figures show adequate agreement between analytical expressions and simulated results for users 1 and 2. User 3 continues to have a significant mismatch between analytical and simulated results for a restricted region in the `Modified-SIC` side of the inequality $\mathscr{E}$.

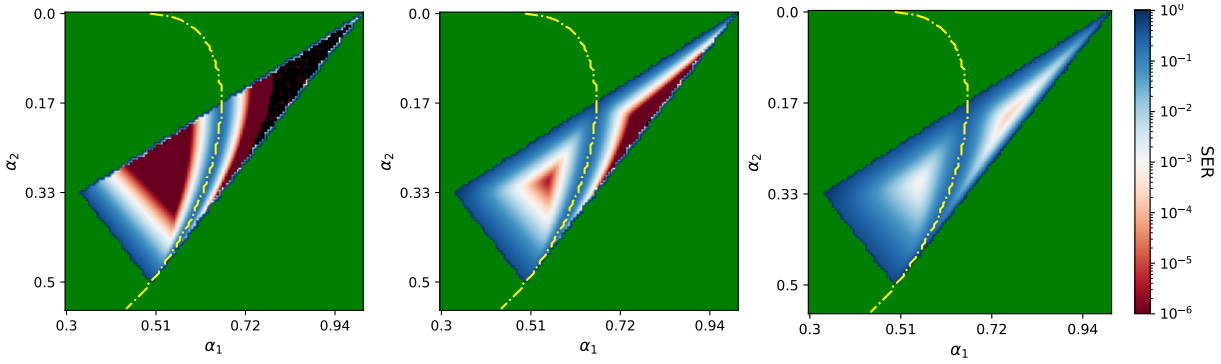
Figure 31 – SER comparison between Traditional-SIC and `Modified-SIC` for the downlink scenario. Figure 31a shows a power allocation coefficient satisfying $\mathscr{E}$ and Figure 31b considers the case $\overline{\mathscr{E}}$. For both figures, $\gamma_1 = \gamma_2 = 3$ dB.



(a) $\mathscr{E} : \alpha_1 = 0.9, \alpha_2 = 0.07, \alpha_3 = 0.03$ (b) $\overline{\mathscr{E}} : \alpha_1 = 0.5, \alpha_2 = 0.3, \alpha_3 = 0.2$

Source: The author (2021).

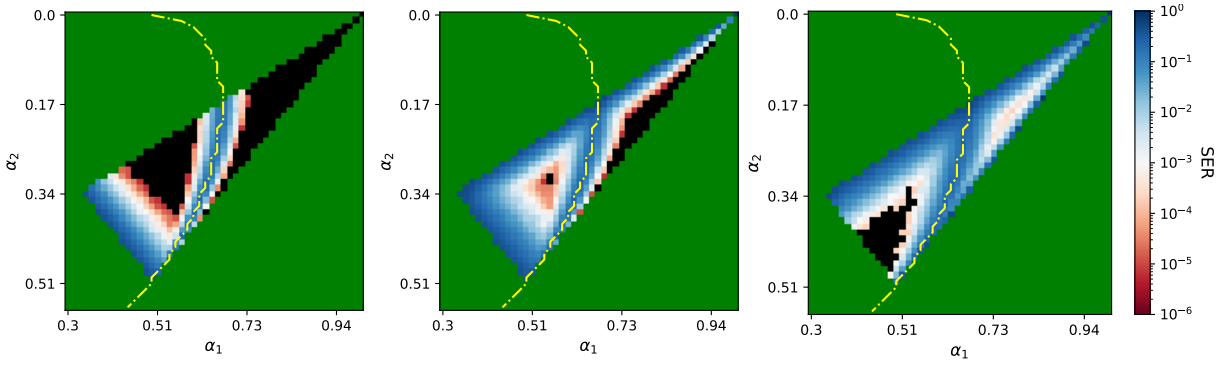
Figure 32 – Analytical downlink SER heatmaps for each user. $\gamma_1 = \gamma_2 = 3$ dB. The dashed line indicates the inequality $\mathscr{E}$. SNR = 25 dB.



(a) User 1. (b) User 2. (c) User 3.

Source: The author (2021).

Figure 33 – Simulated downlink SER heatmaps for each user. $\gamma_1 = \gamma_2 = 3$ dB. The dashed line indicates the inequality $\mathscr{E}$. SNR = 25 dB.



(a) User 1. (b) User 2. (c) User 3.

Source: The author (2021).

3.5 FINAL CONSIDERATIONS

In this chapter, we proposed a modified SIC decoding for the three-user NOMA scenario by assuming channel gain knowledge. We investigated the SER performance for different decoders under both scenarios $\mathscr{D}$ and $\overline{\mathscr{D}}$ (or $\mathscr{E}$ and $\overline{\mathscr{E}}$), in which results holds for both uplink and downlink scenarios, with a minor necessary adaptation. It was shown that the proposed decoder can be seen as an alternative way to obtain previously encountered results in literature using DL techniques (EMIR et al., 2021), while maintaining a quadratic time complexity with regard to the modulation order. Furthermore, the joint bidimensional analysis enables a power allocation approach and shows that the proposed decoder can outperform the traditional one under specific power allocation and channel gain constraints.

4 CHAOS-BASED COMMUNICATION SYSTEM

Chaotic dynamical systems have been explored in the literature to design wireless communication systems (KADDOUM, 2016; CHEN et al., 2017; YAO et al., 2019; CAI et al., 2021). It is shown in (REN; BAPTISTA; GREBOGI, 2013) that the Lyapunov exponent of modulated chaotic signals generated by three-dimensional chaotic attractors is invariant under the effect of multipath propagation. This implies that there is no information loss due to the channel effect (REN; BAPTISTA; GREBOGI, 2013). Moreover, chaotic signals are topologically invariant under the effect of multipath fading (REN; BAPTISTA; GREBOGI, 2013; YAO et al., 2017), hence the chaotic attractor can be reconstructed in the decoder using the received chaotic signals. This topological robustness in multipath channels is an evidence that chaotic signals have peculiar properties to be incorporated in the design of chaos-based wireless communication schemes. When properly designed, these schemes can compensate the typical imperfections of wireless channels employing strategies to benefit from the properties of chaotic signals (YAO et al., 2017).

Space-time trellis codes (STTCs) are proposed in (TAROKH; SESHADRI; CALDERBANK, 1998) in order to increase the performance and transmission rate in wireless communication systems. In this scheme, the utilization of multiple transmit and receive antennas aggregates diversity to the system (TAROKH; SESHADRI; CALDERBANK, 1998; BLUM, 2002; YUAN et al., 2003). Moreover, there is a joint design of channel coding, modulation, transmit diversity and receive diversity. STTCs can achieve the best compromise between diversity gain, data rate, and trellis complexity (TAROKH; SESHADRI; CALDERBANK, 1998).

A trellis-coded modulation system based on three-dimensional chaotic attractors is proposed in (SOUZA; PIMENTEL; CHAVES, 2020) with a detailed case study for the Sprott D attractor (SPROTT, 1994). The dynamical evolution of the chaotic trajectories within the attractor is represented by the symbolic dynamics generated by a labeled partition of a Poincaré section. A finite-state encoder (FSE) is designed in order to encode information sequences into the restricted sequences generated by the symbolic dynamics of the chaotic trajectories. However, there are particular aspects regarding the implementation of chaos-based communication systems such as the lack of a matched filter, the irregular dynamics, and the infinity dimensional signal constellation due to the non-periodic behavior of the chaotic dynamics. These characteristics require new methodologies to design chaos-based communication systems and DL decoding can be an approach to deal with these irregularities.

This chapter analyzes the network capability of capturing the irregular dynamics generated by the chaotic system and the memory structure that emerges from the dynamical properties of the chaotic flow by using modulated chaotic waveforms.

The decoding of linear block codes using DL techniques has been considered in the literature (GRUBER et al., 2017; KIM et al., 2018; BE'ERY et al., 2020; LIANG; SHEN; WU,

2018). The results presented in (GRUBER et al., 2017) show evidence that NNs are able to learn some sort of decoding algorithm rather than just being a classifier between possible codewords. A feed-forward network structure is employed in (GRUBER et al., 2017) to decode polar and random codes. The approach in (KIM et al., 2018) employs a bidirectional-GRU recurrent network for the task of decoding convolutional codes. It is shown in (LIANG; SHEN; WU, 2018) a convolutional based network used in conjunction with traditional belief propagation decoding of LDPC codes. A bidirectional recurrent structure is employed to decode turbo codes in (JIANG et al., 2019a). As far as we know, decoding strategies based on DL have not been considered for chaos-based trellis codes.

The contributions of this chapter are twofold. The first one is the incorporation of the chaotic dynamics, represented by the symbolic dynamics of the chaotic trajectories, in chaos-based STTC systems which we denote as CB-STTC (*chaos-based space-time trellis code*). In this case, the restrictions derived from the dynamical and topological properties of the chaotic flow are exploited in the design of a communication system with multiple antennas. In particular, the methodology proposed in (SOUZA; PIMENTEL; CHAVES, 2020) is generalized to design a CB-STTC based on the Sprott D chaotic attractor. The second contribution is the proposition of a DL architecture that combines convolutional and RNNs to analyze the capability of the network to learn how to decode the proposed CB-STTC. The objective is to design a framework that captures the irregular dynamics of the chaotic signals and the memory structure that emerges from the dynamical properties of the modulated chaotic waveforms. These characteristics are incorporated in the decoder in the learning process. The training strategy of the network is discussed and its performance is analyzed.

This chapter is divided into five sections. In Section 4.1 the Sprott D attractor is presented and we discuss its symbolic dynamics and the associated FSE. The communication system and the proposed CB-STTC are detailed in Section 4.2. The NN architecture is discussed in Section 4.3. In Section 4.4 the performance of the proposed CB-STTC is presented with computer simulations. Finally, in Section 4.5 we present the final considerations.

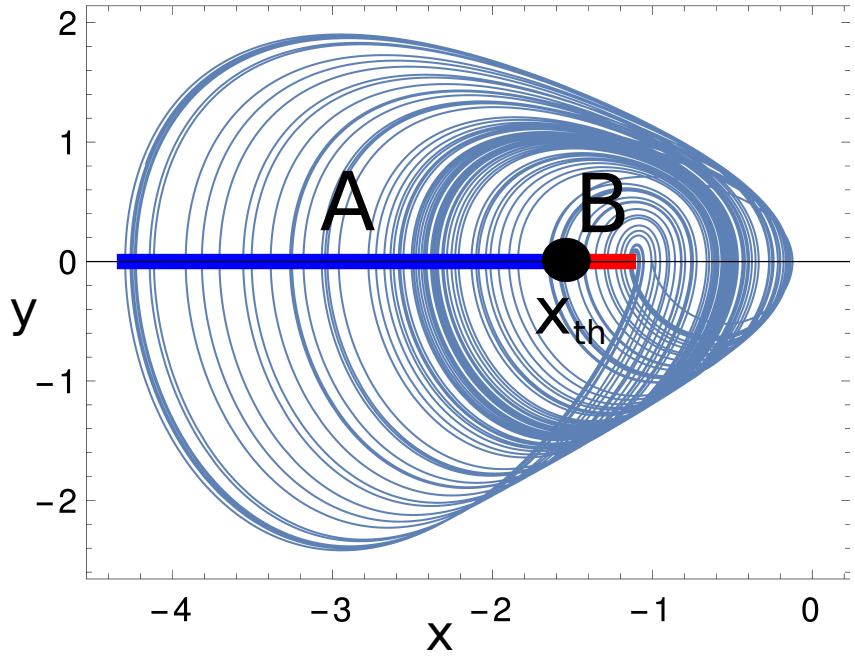
4.1 PRELIMINARIES

The Sprott D system is defined in Table 1 at (SPROTT, 1994)

$$\begin{cases} \dot{x}(t) &= -y(t) \\ \dot{y}(t) &= x(t) + z(t) \\ \dot{z}(t) &= x(t)z(t) + ay^2(t), \end{cases} \tag{53}$$

where $a = 2.3$ is the control parameter. The dynamical evolution of the chaotic trajectories within the attractor is simply described by using a labeled partition of a Poincaré section. A convenient section is a plane perpendicular to the plane xy, located at $y = 0$. Figure 34 shows the Sprott D attractor and the considered Poincaré section projected in the plane

Figure 34 – Sprott D attractor and its Poincaré projection in the plane xy with the considered binary partition.



Source: (SOUZA et al., 2021) (2021).

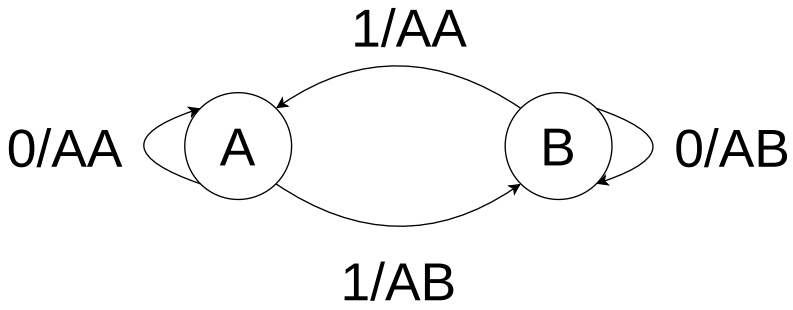
xy. It is also shown a binary labeling over the alphabet $\mathcal{A}_2 = \{\mathrm{A}, \mathrm{B}\}$. Each symbol in $\mathcal{A}_2$ indicates if the crossing is on the left or on the right of the threshold x_{th}, where x_{th} is the minimum of the Poincaré map (GILMORE; LEFRANC, 2012).

The Poincaré map $P : X \to X$ for a three-dimensional chaotic attractor X is defined by $p_{i+1} = P(p_i)$, where $p_i = (x_i, y_i, z_i)$ are the coordinates of the i-th crossing between a trajectory within X and the Poincaré section. Let $p = (p_0, p_1, p_2, \ldots, p_{N-1})$ be a sequence of N crossings and let $s_N = s_0 s_1 s_2 \ldots s_{N-1}, s_i \in \mathcal{A}_2$ be a symbolic sequence, where s_i is the label of the region of the Poincaré section in p_i. In the following we consider this representation for the communication system.

4.1.1 Finite-State Encoders

The existence of restrictions in the occurrence of adjacent symbols in s_N for the Sprott D attractor is detailed in (SOUZA; PIMENTEL; CHAVES, 2020). These restrictions are specified by the set of forbidden sequences $\mathcal{F} = \{\mathrm{BB}, \mathrm{BAAB}, \mathrm{BAAAAB}\}$. This means that s_N contains no sequences belonging to $\mathcal{F}$ in the dynamical evolution of the Sprott system. From $\mathcal{F}$ we design the FSE with rate $1/2$ illustrated in Figure 35 (SOUZA; PIMENTEL; CHAVES, 2020). This FSE is a symbolic representation of the dynamical behavior of the Sprott D system, derived from its topological organization. The sequences generated by paths in the FSE are equivalent to the symbolic sequences generated by successive crossings between the chaotic trajectories and the Poincaré section, satisfying the restrictions specified by $\mathcal{F}$. The FSE maps any binary information sequence into the restricted symbolic sequences

Figure 35 – Finite-state encoder for the Sprott D attractor to map arbitrary binary information sequences into the restricted symbolic sequences specified by $\mathscr{F}$.



Source: (SOUZA et al., 2021) (2021).

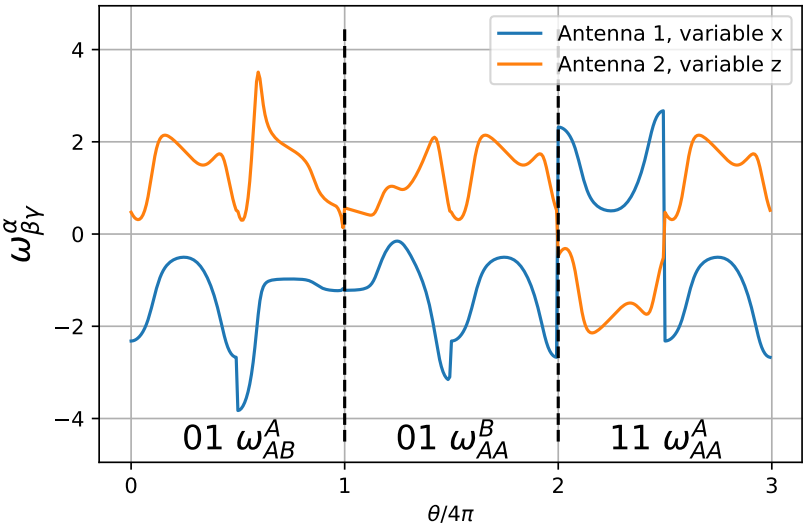
over the alphabet $\mathscr{A}_2$ generated by the symbolic dynamics of the chaotic trajectories. Each bit is mapped into the symbolic sequence $s_1 s_2$. This procedure is equivalent to associate each bit with two crossings in the Poincaré section illustrated in Figure 35. This encoder emerges from the dynamical constraints in the symbolic dynamics of the chaotic trajectories and it forms the basis of the CB-STTC proposed in the next section. In order to transmit information employing the FSE, it is necessary to define the chaotic waveforms associated with the state transitions in the FSE.

4.1.2 Chaotic Waveforms

The chaotic waveforms are the segments of trajectories within the attractor connecting the corresponding regions of the Poincaré section for each possible transition in the FSE. As depicted in (SOUZA et al., 2021), the time interval between the crossings is not uniform due to the aperiodicity of the chaotic behavior. Hence, the chaotic waveforms are parameterized by the angular variable θ such that for successive crossings in the Poincaré section the variation of θ is 2π. For example, the chaotic waveforms associated with the transition from state A to state B in the FSE are the segments of the trajectory $x(\theta), y(\theta)$ or $z(\theta)$ connecting the regions A to A and in the following connecting the regions A to B, which we denote as $w_{\mathrm{AB}}^{\mathrm{A}}(\theta),\ w = x, y, z$, with angular period 4π. The superscript in $w_{\beta\gamma}^{\alpha}(\theta)$ indicates the current region of Poincaré section and the subscript indicates the subsequent visited regions. Figure 36 illustrates the chaotic trajectories that encode the information sequence 010111 starting from state A.

The chaotic waveforms associated with each determinate transition are slightly different for each crossing due to the aperiodicity of the chaotic behavior. Therefore, we associate an average curve for each possible transition constructed as the mean curve generated by several initial conditions for (53) in the same region for each region. We denominate the average curve as *typical transition curve* and employ this curve in the demodulator as detailed in Section 4.2.

Figure 36 – Chaotic trajectories $x(\theta), z(\theta)$ encoding the information sequence 010111 starting from state A in the FSE, or region A in the Poincaré section.



Source: Adapted from (SOUZA et al., 2021) (2021).

4.2 COMMUNICATION SYSTEM

In this section we consider the FSE in Figure 35 to design a communication system with two states, N_T transmit antennas and N_R receive antennas. The channel has quasi-static fading with additive Gaussian white noise. The fading coefficients h_{ij} from transmit antenna i to receive antenna j are complex Gaussian random variables, constant in each frame of length ℓ and change independently in consecutive frames. In each signaling interval, the chaotic signal $w^{\alpha}_{\beta\gamma}(\theta), 0 \leq \theta \leq 4\pi$, is mapped into the transmitted signal $s(t)$. The received signal at the j-th receive antenna is the linear superposition

$$r_j(t) = \sum_{i=1}^{N_T} |h_{ij}| s_i(t) + n(t),\ j = 1, 2, \ldots, N_R, \tag{54}$$

where $n(t)$ is an additive white Gaussian noise with power spectral density $N_0/2$ and $|h_{ij}|$ has Rayleigh PDF with unit second moment. The maximum likelihood decoding (MLD) is performed by the Viterbi algorithm (MOON, 2005) with a trellis derived from the FSE. Assuming that the decoder perfectly knows the fading coefficients h_{ij}, the decoding metric is given by

$$\sum_{j=1}^{N_R} \int \left(r_j(t) - \sum_{i=1}^{N_T} h_{ij} \bar{s}_i(t) \right)^2 dt, \tag{55}$$

where $\bar{s}(t)$ denotes a typical transition curve. The SNR per receive antenna is defined as $\sum_{i=1}^{N_T} E_i/N_0$, where E_i is the average energy of the transmitted signal through the i-th antenna.

4.2.1 Design Criterion of the CB-STTC

For the quasi-static fading channel, the design of the CB-STTC follows the criteria proposed in (TAROKH; SESHADRI; CALDERBANK, 1998). Let $\mathbf{c} = (s_1^1(t) \dots s_1^{N_T}(t) s_2^1(t) \dots s_2^{N_T}(t) \dots s_\ell^1(t) \dots s_\ell^{N_T}(t))$ be a codeword in a frame, where $s_k^i(t)$ is the chaotic signal transmitted in the k-th signaling interval through antenna i. Consider another codeword $\mathbf{e} = (e_1^1(t) \dots e_1^{N_T}(t) e_2^1(t) \dots e_2^{N_T}(t) \dots e_\ell^1(t) \dots e_\ell^{N_T}(t))$. The matrix of distances $\mathbf{A}(\mathbf{c}, \mathbf{e})$ is an $N_T \times N_T$ square matrix with the entries $a_{pq}, 1 \leq p, q \leq N_T$, defined as (TAROKH; SESHADRI; CALDERBANK, 1998)

$$a_{pq} = \sum_{k=1}^{\ell} \int \Big(s_k^p(t) - e_k^p(t)\Big)\Big(s_k^q(t) - e_k^q(t)\Big)^* dt, \tag{56}$$

where $*$ denote the complex conjugate operation. The criteria to design an STTC as function of $\mathbf{A}(\mathbf{c}, \mathbf{e})$ are (TAROKH; SESHADRI; CALDERBANK, 1998):

- Maximize the minimum rank of $\mathbf{A}(\mathbf{c}, \mathbf{e})$ for all possible pairs of codewords.
- Maximize the minimum determinant of $\mathbf{A}(\mathbf{c}, \mathbf{e})$ for all possible distinct pairs of codewords with minimum rank.

The first criterion is associated with the diversity gain and the second criterion with the coding gain. Let r be the rank of $\mathbf{A}(\mathbf{c}, \mathbf{e})$, hence the diversity gain of the STTC is rN_R (TAROKH; SESHADRI; CALDERBANK, 1998). Ideally, the STTC should be designed such that $\mathbf{A}(\mathbf{c}, \mathbf{e})$ is a full rank matrix. As depicted in (TAROKH; SESHADRI; CALDERBANK, 1998; BLUM, 2002; YUAN et al., 2003), we consider $N_T = 2$ and rate 2 bits/s/Hz, therefore there are four divergent edges from each state in the FSE. Each edge is labeled with two information bits. The two-state FSE is designed such that the matrices $\mathbf{A}(\mathbf{c}, \mathbf{e})$ associated with pairs of codewords have minimum rank $r = N_T = 2$ for any pair of codewords. Consequently, the diversity gain of the STTC is $2N_R$.

The chaotic waveforms transmitted through each antenna follow the scheme shown in Figure 35. Computer simulations show that transmitting the variables x and z with the inversion of signals associated with bit 1 increases the minimum determinant of $\mathbf{A}(\mathbf{e}, \mathbf{c})$. Hence, antenna 1 transmits the signal $\pm x(t)$ and antenna 2 transmits $\pm z(t)$. The FSE for the proposed CB-STTC is detailed in Table 2.

4.3 DEEP LEARNING DECODING

In this section we detail the architecture and training of the NN employed to decode the CB-STTC.

Let $\mathbf{v}$ be a vector of length ℓ composed of a quaternary sequence over the alphabet $\{0, 1, 2, 3\}$ where each element of this vector is a class representing a pair of information bits

Table 2 – Two-state FSE for the proposed CB-STTC.

FSE Transition	Transmitted pair	Antenna 1	Antenna 2
$A \to A$	00	$x_{AA}^{A}(t)$	$z_{AA}^{A}(t)$
$A \to A$	11	$-x_{AA}^{A}(t)$	$-z_{AA}^{A}(t)$
$A \to B$	01	$x_{AB}^{A}(t)$	$-z_{AB}^{A}(t)$
$A \to B$	10	$-x_{AB}^{A}(t)$	$z_{AB}^{A}(t)$
$B \to A$	01	$x_{AA}^{B}(t)$	$-z_{AA}^{B}(t)$
$B \to A$	10	$-x_{AA}^{B}(t)$	$z_{AA}^{B}(t)$
$B \to B$	00	$x_{AB}^{B}(t)$	$z_{AB}^{B}(t)$
$B \to B$	11	$-x_{AB}^{B}(t)$	$-z_{AB}^{B}(t)$

$\{00, 01, 10, 11\}$. We generate the codeword $\mathbf{c}$ associated with $\mathbf{v}$ and calculate $r_j(t)$ as in (54) for a given channel realization. We then calculate the metric defined in (55) for all possible typical transition curves. There are eight possible pairs of signals transmitted simultaneously by the two antennas, as detailed in the FSE in Table 2. We construct an $(8 \times \ell)$-dimensional matrix $\mathbf{X}$ with all metrics in each frame. The pair ($\mathbf{X}$, $\mathbf{v}$) represents the communication system and all information provided to the NN is restricted to this set. It is expected that the network learns how to map $\mathbf{X}$ to $\mathbf{v}$ and be capable of generalizing for SNR values not used in the training step.

The NN is composed of three types of layers: convolutional, recurrent and totally connected (GOODFELLOW; BENGIO; COURVILLE, 2016). The simultaneous utilization of these layers allows to extract attributes provided by the convolutional layer while preserving the sequential dynamics of the data set due to the recurrent layer. The totally connected layer is employed for the classification of the input sequence.

The initial layers employ one-dimensional causal convolutional filters with dilatation rate duplicated at each step (OORD et al., 2016). This structure can be employed to obtain attributes in sequential data. The parameters in this layer are the quadruple (*number of filters, kernel size, stride, dilatation rate*). The subsequent layers implement the recurrent structure of the network by using bidirectional GRU layers with activation function *tanh*. This structure is successful employed in the decoding of convolutional codes in AWGN channels (KIM et al., 2018; TANDLER et al., 2019). The recurrent architecture is employed for the sequential processing of the received data and it performs estimation of the data at each time step. Batch Normalization layers are interleaved along the layers to provide stability to the network (IOFFE; SZEGEDY, 2015). The totally connected layer estimates the final classification of $\mathbf{X}$ according to the possible transmitted symbols using the softmax

Table 3 – Architecture of the NN. The batch size is $\Lambda = 32$.

Layer Type	Parameters	Output Dimension
Causal Conv1D	(32, 3, 1, 1)	$(\Lambda, \ell, 32)$
Causal Conv1D	(32, 3, 1, 2)	$(\Lambda, \ell, 32)$
Causal Conv1D	(32, 3, 1, 4)	$(\Lambda, \ell, 32)$
Causal Conv1D	(32, 3, 1, 8)	$(\Lambda, \ell, 32)$
Batch Normalization	No Parameters	$(\Lambda, \ell, 32)$
GRU Bidirecional	200 nodes, *tanh*	$(\Lambda, \ell, 400)$
Batch Normalization	No Parameters	$(\Lambda, \ell, 400)$
GRU Bidirecional	80 nodes, *tanh*	$(\Lambda, \ell, 160)$
Batch Normalization	No Parameters	$(\Lambda, \ell, 160)$
GRU Bidirecional	40 nodes, *tanh*	$(\Lambda, \ell, 80)$
Batch Normalization	No Parameters	$(\Lambda, \ell, 80)$
Totally Connected	4 nodes, *softmax*	$(\Lambda, \ell, 1)$

layer. This layer normalizes the activation functions of the previous layer in order to obtain the probability distribution of each class

$$S(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, i = 1, ..., C, \tag{57}$$

where $C = 4$ since there are four classes and the entries of the vector $\mathbf{z}$ are the values returned by the activation functions of the previous layer. The output of the network is the index associated with the greatest argument of $S(\mathbf{z})$. The complete architecture of the network is detailed in Table 3. It is worth pointing out that a reduction in the number of layers implies in a degradation of the system performance for the 1 receiving antenna case. On the other side, when we consider a network with more layers we observe no meaningful performance gain.

4.3.1 Training

The training step employs two datasets: the training set and the validation set. The training set is composed by 10000 pairs $(\mathbf{X}, \mathbf{v})$. Each element in $\mathbf{X}$ is normalized by subtracting the mean of the elements in $\mathbf{X}$ and dividing by the standard deviation. We consider that the network perfectly knows the fading coefficients h_{ij} (a similar assumption is considered, for example, in (WU et al., 2019)).

As discussed in Section 2.4, the network performance depends on the SNR used for training (GRUBER et al., 2017). The validation set iteratively evaluates the performance of the network at each training step. The SNR employed for the validation set must be distinct of the SNR in the training set because we need to verify the generalization capacity of the network. We consider that these SNR are fixed and are empirically estimated depending on the number of receive antennas since the diversity gain affects the system performance. For a single antenna (resp. two antennas) the training and validation SNR are 9 dB and 16 dB (resp. 4.5 dB and 7.5 dB), respectively. The objective in training is to minimize the cost function (GOODFELLOW; BENGIO; COURVILLE, 2016)

$$J(\boldsymbol{\Delta}) = \frac{1}{\Lambda} \sum_{k=1}^{\Lambda} \sum_{i=1}^{\ell} l(\hat{v}_i^{(k)}, v_i^{(k)}), \tag{58}$$

where $\boldsymbol{\Delta}$ are the training parameters of the network, the *batch size* Λ is the number of tuples $(\mathbf{X}, \mathbf{v})$ employed in the evaluation of $J(\boldsymbol{\Delta})$, $\hat{v}_i^{(k)}$ is the estimate of the network for $v_i^{(k)}$ corresponding to the tuple k at time i and l is the cross entropy function per sample (GOODFELLOW; BENGIO; COURVILLE, 2016)

$$l(\hat{v}_i^{(k)}, v_i^{(k)}) = -\sum_{c=1}^{C} \mathbb{1}_{(v_i^{(k)}=c)} \log P(\hat{v}_i^{(k)} = c), \tag{59}$$

where $\mathbb{1}_{(v_i^{(k)}=c)}$ is the binary indicator function that returns 1 if $v_i^{(k)} = c$, or 0 otherwise. We employ same training heuristics as in Section 2.4.

The set of metrics for each frame $\mathbf{X}$ provided to the network in the training step considers the small variations of the chaotic waveforms in the dynamical evolution of the chaotic system.

4.4 SIMULATION RESULTS

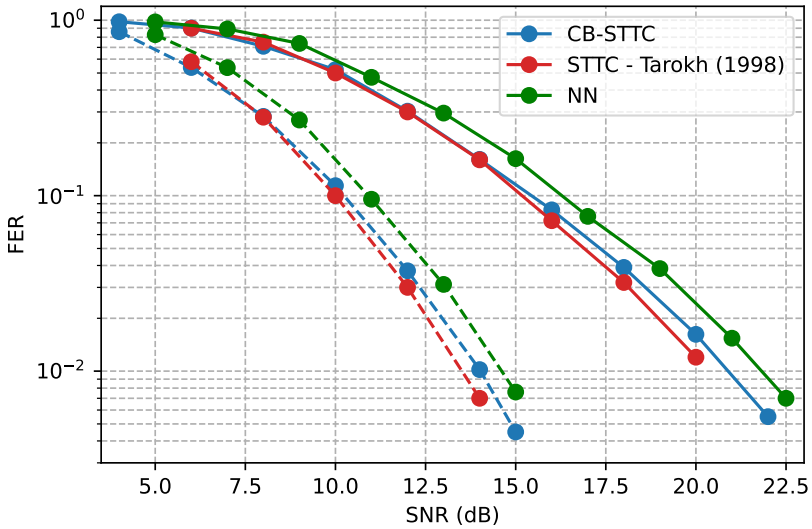
The performance of the proposed CB-STTC is analyzed by computer simulations. The MLD is performed by the Viterbi algorithm (MOON, 2005) with the two-state trellis derived from the FSE in Table 2 with metric increments defined in (55). We consider that h_{ij} coefficients are known by the decoding algorithms.

Figure 37 shows the frame error rate (FER) versus SNR where each frame has length $\ell = 130$ for $N_T = 2$ transmit antennas and $N_R = 1$ and 2 receive antennas. We observe that the diversity gain of the CB-STTC (this is obtained by the slope of the curve at a particular SNR) increases with the number of receive antennas since the matrices $\mathbf{A}(\mathbf{c}, \mathbf{e})$ have full rank. For comparison purposes, it is also shown the performance of the four-state STTC proposed in (TAROKH; SESHADRI; CALDERBANK, 1998). Using the variables x and z we are able to label the CB-STTC trellis (Table 2) in such a way to achieve a coding gain very close to the conventional STTC. The performance of the two schemes is very similar. The performance obtained by the NN has the same behavior of the other schemes and therefore the proposed architecture is capable of learning the decoder structure of the proposed CB-STTC, however with a gap of approximately 1 dB to the optimal performance.

4.5 FINAL CONSIDERATIONS

In this chapter we proposed a CB-STTC scheme with $N_T = 2$ transmit antennas and N_R receive antennas with maximum diversity gain $2N_R$. We also proposed a NN architecture to decode the proposed CB-STTC that is capable of learning the CB-STTC. Hence, the

Figure 37 – FER versus SNR of the proposed two-state CB-STTC and the four-state STTC in (TAROKH; SESHADRI; CALDERBANK, 1998) with $N_T = 2$, $N_R = 1, 2$ and $\ell = 130$. Solid lines indicates $N_R = 1$, dashed lines, $N_R = 2$.



Source: (SOUZA et al., 2021).

proposal is viable to design chaos-based wireless systems. As explained in (SHLEZINGER et al., 2020), NN models are efficient in scenarios where the channel model is unknown or cannot be accurately estimated in such a way that the MLD is unfeasible or extremely complex. The analysis of the CB-STTC in these scenarios are left for future work.

5 CONCLUSION

The increasing presence of ML on an wide range of applications inspired the study of these techniques in the communications field. This research proposed to present and analyze three different problems which can be in some way or form approached by ML methods. Since each one is fundamentally different in nature, there is no singular approach to all of them, for each one it was presented a self-contained problem formulation and problem solving approach, including ML based ones.

In Chapter 2, the Markov-Gaussian channel problem was presented, two different ML models were discussed representing the traditional approach based on HMM, as well as a proposed alternative based on a NN. Comparisons were made between both approaches and it was shown that the two have similar performance, and the NN can potentially outperform the HMM if more flexible and data driven environments are considered. Future works can include different scenarios and channel models in which the DL approach can more reliably outperform conventional methods. Further investigation of training and validation SNRs are also possible.

In Chapter 3, the NOMA problem was presented, in which a modified version of the traditionally employed SIC is shown. A NN based decoder was employed to validate that the obtained performance of the proposed decoder matches the NN one. Therefore, the ML method can also be used as an auxiliary tool in development of traditional algorithms. Future work can investigate generalization to a higher number of users and the decision regions for each user. A possible approach to achieve this generalization is to employ a decision tree (DT) or a support vector machine (SVM) model as part of the decoding algorithm, which will correlate the necessary decision region to the set of channel gains. Given the channel gain parameters, or at least estimated median values, the model can be trained to learn the constellation mapping for the first step of decoding. For further steps, one can employ different models trained under the assumption that previous users have been correctly decoded. Both these learning models have better interpretability than NNs and we expect that the performance can match NNs in most of the simpler conventionally considered scenarios such as AWGN or Rayleigh fading channels. This approach should not scale as well as NNs for increasing number of users, but still can be employed as analysis tool and modulation and decoding design. DTs could be used alternatively to NNs since the decision regions obtained after training can be extracted directly from the trees. This model typically provides a better understanding about the task than NNs since the tree can be effectively visualized, and any model prediction can be explained using boolean logic. We also argue that the DT, although less powerful, is enough to tackle the low number of users scenario while using a white box model, in contrast to the NN's black box. Investigation of imperfect knowledge of channel gains can also be considered for a better understanding of the proposed method's practical usefulness, since the decision regions depend on these

variables. Another line of investigation is the analysis of related techniques for N users generalization, such as the influence of the proposed algorithm in pairing schemes or beamforming.

In Chapter 4, a chaos-based communication system was presented, in which a NN based decoder was employed and compared to the MLD performed by the Viterbi algorithm. The results show that the NN decoder is able to maintain the same diversity gain as the MLD while a small constant degradation is observed. Future work can include ML decoding strategies seeking to obtain equal performance to the MLD. Another possibility is to integrate LDPC codes (FANG et al., 2019) with the proposed CB-STTC in order to improve the system performance. An NN architecture to decode this combined scheme can also be proposed in the future.

The discussed problems show the high flexibility of the ML approach in solving problems, while, in some cases, obtaining maximum or near maximum performances, thus motivating its use as an alternative to known methods or at least as benchmark.

REFERENCES

AHMED, I. et al. Deep learning for MMSE estimation of a Gaussian source in the presence of bursty impulsive noise. **IEEE Communications Letters**, p. 1–1, 2020.

ALAM, M. S.; KADDOUM, G.; AGBA, B. L. Bayesian MMSE estimation of a Gaussian source in the presence of bursty impulsive noise. **IEEE Communications Letters**, v. 22, n. 9, p. 1846–1849, sep. 2018.

ALDABABSA, M. et al. Bit error rate for NOMA network. **IEEE Communications Letters**, v. 24, n. 6, p. 1188–1191, jun. 2020.

ALMOHIMMAH, E. M.; ALRESHEEDI, M. T. Error analysis of NOMA-based VLC systems with higher order modulation schemes. **IEEE Access**, v. 8, p. 2792–2803, 2020.

ASSAF, T. et al. Exact BER performance analysis for downlink NOMA systems over Nakagami-m fading channels. **IEEE Access**, v. 7, p. 134539–134555, 2019.

ASSAF, T. et al. NOMA receiver design for delay-sensitive systems. **IEEE Systems Journal**, p. 1–12, 2020.

ASSAF, T. et al. Exact bit error-rate analysis of two-user NOMA using QAM with arbitrary modulation orders. **IEEE Communications Letters**, v. 24, n. 12, p. 2705–2709, dec. 2020.

BAE, J. H. et al. An overview of channel coding for 5G NR cellular communications. **APSIPA Transactions on Signal and Information Processing**, vol. 8, e17, pp. 1-14, jun. 2019.

BAHL, L. et al. Optimal decoding of linear codes for minimizing symbol error rate. **IEEE Transactions on Information Theory**, v. 20, n. 2, p. 284–287, mar. 1974.

BALATSOUKAS-STIMMING, A.; STUDER, C. Deep unfolding for communications systems: A survey and some new directions. In: **IEEE International Workshop on Signal Processing Systems (SiPS)**. 2019. p. 266–271.

BAUM, L. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. **Inequalities, vol. 3, no. 1, pp. 1-8**, jan. 1972.

BENNATAN, A.; CHOUKROUN, Y.; KISILEV, P. Deep learning for decoding of linear codes - A syndrome-based approach. **arXiv:1802.04741**, feb. 2018.

BE'ERY, I. et al. Active deep decoding of linear codes. **IEEE Transactions on Communications**, v. 68, n. 2, p. 728–736, feb. 2020. ISSN 1558-0857.

BJORNSON, E.; GISELSSON, P. Two applications of deep learning in the physical layer of communication systems [Lecture Notes]. **IEEE Signal Processing Magazine**, v. 37, n. 5, p. 134–140, sep. 2020.

BLACKARD, K.; RAPPAPORT, T.; BOSTIAN, C. Measurements and models of radio frequency impulsive noise for indoor wireless communications. **IEEE Journal on Selected Areas in Communications**, v. 11, n. 7, p. 991–1001, sep. 1993.

BLUM, R. Some analytical tools for the design of space-time convolutional codes. **IEEE Transactions on Communications**, v. 50, n. 10, p. 1593–1599, oct. 2002.

BUDHIRAJA, I. et al. A systematic review on NOMA variants for 5G and beyond. **IEEE Access**, v. 9, p. 85573–85644, 2021.

CAI, G. et al. Design of an MISO-SWIPT-aided code-index modulated multi-carrier M-DCSK system for e-Health IoT. **IEEE Journal on Selected Areas in Communications**, v. 39, n. 2, p. 311–324, feb. 2021.

CELIK, N. et al. Deep-Channel uses deep neural networks to detect single-molecule events from patch-clamp data. **Communications Biology**, v. 3, n. 1, p. 1–10, jan. 2020.

CHEN, J.; TANNER, R. A hybrid coding scheme for the Gilbert-Elliott channel. **IEEE Transactions on Communications**, v. 54, n. 10, p. 1787–1796, oct. 2006.

CHEN, P. et al. Design of a capacity-approaching chaos-based multiaccess transmission system. **IEEE Transactions on Vehicular Technology**, v. 66, n. 12, p. 10806–10816, dec. 2017.

CHEN, X.; JIA, R.; NG, D. W. K. On the design of massive non-orthogonal multiple access with imperfect successive interference cancellation. **IEEE Transactions on Communications**, v. 67, n. 3, p. 2539–2551, mar. 2019.

CHIEN, T. V.; BJORNSON, E.; LARSSON, E. G. Sum spectral efficiency maximization in massive MIMO systems: Benefits from deep learning. In: **IEEE International Conference on Communications (ICC)**. 2019. p. 1–6.

CHO, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: **Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)**. 2014.

CHU, L. et al. NOLD: A neural-network optimized low-resolution decoder for LDPC codes. **Journal of Communications and Networks**, v. 23, n. 3, p. 159–170, jun. 2021.

DING, Z. et al. Application of non-orthogonal multiple access in LTE and 5G networks. **IEEE Communications Magazine**, v. 55, n. 2, p. 185–191, feb. 2017.

DOAN, N. et al. Neural belief propagation decoding of CRC-polar concatenated codes. In: **IEEE International Conference on Communications (ICC)**. 2019. p. 1–6.

DÖRNER, S. et al. Deep learning based communication over the air. **IEEE Journal of Selected Topics in Signal Processing**, v. 12, n. 1, p. 132–143, feb. 2018. ISSN 1941-0484.

ECKFORD, A.; KSCHISCHANG, F.; PASUPATHY, S. Analysis of low-density parity-check codes for the Gilbert-Elliott channel. **IEEE Transactions on Information Theory**, v. 51, n. 11, p. 3872–3889, nov. 2005.

ELLIOTT, E. O. Estimates of error rates for codes on burst-noise channels. **The Bell System Technical Journal**, v. 42, n. 5, p. 1977–1997, sep. 1963.

EMIR, A. et al. DeepMuD: Multi-user detection for uplink grant-free NOMA IoT networks via deep learning. **IEEE Wireless Communications Letters**, v. 10, n. 5, p. 1133–1137, may 2021.

FANG, Y. et al. Outage-limit-approaching channel coding for future wireless communications: Root-protograph low-density parity-check codes. **IEEE Vehicular Technology Magazine**, v. 14, n. 2, p. 85–93, jun. 2019.

FERTONANI, D.; COLAVOLPE, G. On reliable communications over channels impaired by bursty impulse noise. **IEEE Transactions on Communications**, v. 57, n. 7, p. 2024–2030, jul. 2009.

GARNIER, J.-R. et al. On the performance of QPSK modulation over downlink NOMA: from error probability derivation to SDR-based validation. **IEEE Access**, v. 8, p. 66495–66507, 2020.

GEORGE, D.; HUERTA, E. Deep neural networks to enable real-time multimessenger astrophysics. **Physical Review D**, v. 97, n. 4, p. 044039, feb. 2018.

GILBERT, E. N. Capacity of a burst-noise channel. **The Bell System Technical Journal**, v. 39, n. 5, p. 1253–1265, sep. 1960.

GILMORE, R.; LEFRANC, M. **The Topology of Chaos: Alice in Stretch and Squeezeland**. Edição: 2. : Wiley-VCH, 2012.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, Massachusetts: MIT Press, 2016. ISBN 978-0-262-03561-3.

GRUBER, T. et al. On deep learning-based channel decoding. In: **Annual Conference on Information Sciences and Systems (CISS)**. 2017. p. 1–6.

HAN, W. et al. Study of SER and BER in NOMA systems. **IEEE Transactions on Vehicular Technology**, v. 70, n. 4, p. 3325–3340, apr. 2021.

HE, H. et al. Deep learning-based channel estimation for beamspace mmWave massive MIMO systems. **IEEE Wireless Communications Letters**, v. 7, n. 5, p. 852–855, oct. 2018.

HE, Q.; HU, Y.; SCHMEINK, A. Closed-form symbol error rate expressions for non-orthogonal multiple access systems. **IEEE Transactions on Vehicular Technology**, v. 68, n. 7, p. 6775–6789, jul. 2019.

HOCEVAR, D. E. A reduced complexity decoder architecture via layered decoding of LDPC codes. In: **IEEE Workshop on Signal Processing Systems (SIPS)**. 2004. p. 107–112.

HUANG, H. et al. Deep learning for physical-layer 5G wireless techniques: opportunities, challenges and solutions. **IEEE Wireless Communications**, v. 27, n. 1, p. 214–222, feb. 2020.

HUANG, X.; REDDY, R.; ACERO, A. **Spoken Language Processing: A Guide to Theory, Algorithm and System Development**. Upper Saddle River, NJ: Prentice Hall, 2001. ISBN 978-0-13-022616-7.

IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: **International Conference on Machine Learning (ICML)**. Lille, France: [s.n.], 2015. v. 37, p. 448–456.

IRAQI, Y.; AL-DWEIK, A. Power allocation for reliable SIC detection of rectangular QAM-based NOMA systems. **IEEE Transactions on Vehicular Technology**, v. 70, n. 8, p. 8355–8360, aug. 2021.

JIANG, W.; SCHOTTEN, H. D. Deep learning for fading channel prediction. **IEEE Open Journal of the Communications Society**, v. 1, p. 320–332, 2020.

JIANG, W.; STRUFE, M.; SCHOTTEN, H. D. Long-range MIMO channel prediction using recurrent neural networks. In: **Annual Consumer Communications Networking Conference (CCNC)**. 2020. p. 1–6.

JIANG, Y. et al. DEEPTURBO: Deep turbo decoder. In: **International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)**. 2019. p. 1–5.

JIANG, Y. et al. LEARN codes: Inventing low-latency codes via recurrent neural networks. In: **IEEE International Conference on Communications (ICC)**. 2019. p. 1–7.

JIANG, Y. et al. Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels. In: WALLACH, H. et al. (Ed.). **Advances in Neural Information Processing Systems 32 (NeurIPS)**. : Curran Associates, Inc., 2019. p. 2758–2768.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 2. ed. Upper Saddle River, N.J: Prentice Hall, 2008. ISBN 978-0-13-187321-6.

KADDOUM, G. Wireless chaos-based communication systems: A comprehensive survey. **IEEE Access**, v. 4, p. 2621–2648, 2016.

KARA, F.; KAYA, H. BER performances of downlink and uplink NOMA in the presence of SIC errors over fading channels. **IET Communications**, v. 12, n. 15, p. 1834–1844, 2018.

KARA, F.; KAYA, H. Performance analysis of SSK-NOMA. **IEEE Transactions on Vehicular Technology**, v. 68, n. 7, p. 6231–6242, jul. 2019.

KARA, F.; KAYA, H. Error probability analysis of NOMA-based diamond relaying network. **IEEE Transactions on Vehicular Technology**, v. 69, n. 2, p. 2280–2285, feb. 2020.

KARA, F.; KAYA, H. Error probability analysis of non-orthogonal multiple access with channel estimation errors. In: **IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)**. 2020. p. 1–5.

KIM, H. et al. Communication algorithms via deep learning. In: **International Conference on Learning Representations (ICLR)**. 2018.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: **International Conference on Learning Representations (ICLR)**. 2015.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, may 2015.

LEE, I.; KIM, J. Average symbol error rate analysis for non-orthogonal multiple access with m-ary QAM signals in Rayleigh fading channels. **IEEE Communications Letters**, v. 23, n. 8, p. 1328–1331, aug. 2019.

LETIZIA, N. A.; TONELLO, A. M. Capacity-approaching autoencoders for communications. **arXiv:2009.05273**, sep. 2020.

LIANG, F.; SHEN, C.; WU, F. An iterative BP-CNN architecture for channel decoding. **IEEE Journal of Selected Topics in Signal Processing**, v. 12, n. 1, p. 144–159, feb. 2018.

LUGOSCH, L.; GROSS, W. J. Neural offset min-sum decoding. In: **IEEE International Symposium on Information Theory (ISIT)**. 2017. p. 1361–1365.

LUO, Q. et al. An error rate comparison of power domain non-orthogonal multiple access and sparse code multiple access. **IEEE Open Journal of the Communications Society**, v. 2, p. 500–511, 2021.

MITRA, J.; LAMPE, L. On joint estimation and decoding for channels with noise memory. **IEEE Communications Letters**, v. 13, n. 10, p. 730–732, oct. 2009.

MITRA, J.; LAMPE, L. Convolutionally coded transmission over Markov-Gaussian channels: Analysis and decoding metrics. **IEEE Transactions on Communications**, v. 58, n. 7, p. 1939–1949, jul. 2010.

MOON, J.-W.; WONG, T. F.; SHEA, J. M. Pilot-assisted and blind joint data detection and channel estimation in partial-time jamming. **IEEE Transactions on Communications**, v. 54, n. 11, p. 2092–2102, nov. 2006.

MOON, T. K. **Error Correction Coding: Mathematical Methods and Algorithms**. 1. ed. Hoboken, N.J: Wiley-Interscience, 2005. ISBN 978-0-471-64800-0.

MOREIRA, I.; PIMENTEL, C. Iterative decoding of finite-state Markov channels with soft-decision. In: **IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)**. 2017. p. 12–17.

MUSHKIN, M.; BAR-DAVID, I. Capacity and coding for the Gilbert-Elliott channels. **IEEE Transactions on Information Theory**, v. 35, n. 6, p. 1277–1290, nov. 1989.

NACHMANI, E. et al. RNN decoding of linear block codes. **arXiv:1702.07560**, feb. 2017.

NEUMANN, D.; WIESE, T.; UTSCHICK, W. Learning the MMSE channel estimator. **IEEE Transactions on Signal Processing**, v. 66, n. 11, p. 2905–2917, jun. 2018.

NURELLARI, E.; İNCE, E. A. Image transmission over Gilbert-Elliot and ITU fading channels using DVB-T2 channel coding and QPSK-OFDM. In: **Signal Processing and Communications Applications Conference (SIU)**. 2012. p. 1–4.

OORD, A. v. d. et al. WaveNet: A generative model for raw audio. **arXiv:1609.03499**, sep. 2016.

O'SHEA, T. J.; KARRA, K.; CLANCY, T. C. Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention. In: **IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)**. 2016. p. 223–228.

RABINER, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. **Proceedings of the IEEE**, v. 77, n. 2, p. 257–286, feb. 1989.

RASKUTTI, G.; WAINWRIGHT, M. J.; YU, B. Early stopping for non-parametric regression: An optimal data-dependent stopping rule. In: **Annual Allerton Conference on Communication, Control, and Computing (Allerton)**. 2011. p. 1318–1325.

RAUNIYAR, A.; ENGELSTAD, P.; ØSTERBØ, O. N. An adaptive user pairing strategy for uplink non-orthogonal multiple access. In: **2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications**. 2020. p. 1–7.

REN, H.-P.; BAPTISTA, M. S.; GREBOGI, C. Wireless communication with chaos. **Physical Review Letters**, v. 110, n. 18, p. 184101, may 2013.

SAAD, W.; BENNIS, M.; CHEN, M. A vision of 6G wireless systems: applications, trends, technologies, and open research problems. **IEEE Network**, v. 34, n. 3, p. 134–142, may 2020.

SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, v. 45, n. 11, p. 2673–2681, nov. 1997.

SHLEZINGER, N. et al. ViterbiNet: A deep learning based Viterbi algorithm for symbol detection. **IEEE Transactions on Wireless Communications**, v. 19, n. 5, p. 3319–3331, may 2020.

SIDEY-GIBBONS, J. A. M.; SIDEY-GIBBONS, C. J. Machine learning in medicine: a practical introduction. **BMC Medical Research Methodology**, v. 19, n. 1, p. 64, mar. 2019.

SILVER, D. et al. Mastering the game of Go with deep neural networks and tree search. **Nature**, v. 529, n. 7587, p. 484–489, jan. 2016.

SILVER, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. **Science**, v. 362, n. 6419, p. 1140–1144, dec. 2018.

SOLTANI, M. et al. Deep learning-based channel estimation. **IEEE Communications Letters**, v. 23, n. 4, p. 652–655, apr. 2019.

SOUZA, C. E. C. et al. Chaos-based space-time trellis codes with deep learning decoding. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 68, n. 4, p. 1472–1476, apr. 2021.

SOUZA, C. E. C.; PIMENTEL, C.; CHAVES, D. P. B. A symbolic dynamics approach to trellis-coded chaotic modulation. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 67, n. 10, p. 2189–2193, oct. 2020.

SPROTT, J. C. Some simple chaotic flows. **Physical Review E**, v. 50, n. 2, p. R647–R650, aug. 1994.

TANDLER, D. et al. On recurrent neural networks for sequence-based processing in communications. In: **Asilomar Conference on Signals, Systems, and Computers**. 2019. p. 537–543.

TAROKH, V.; SESHADRI, N.; CALDERBANK, A. Space-time codes for high data rate wireless communication: performance criterion and code construction. **IEEE Transactions on Information Theory**, v. 44, n. 2, p. 744–765, mar. 1998.

WANG, X.; LABEAU, F.; MEI, L. Closed-form BER expressions of QPSK constellation for uplink non-orthogonal multiple access. **IEEE Communications Letters**, v. 21, n. 10, p. 2242–2245, oct. 2017.

WANG, Y. et al. A minimum error probability NOMA design. **IEEE Transactions on Wireless Communications**, v. 20, n. 7, p. 4221–4237, jul. 2021.

WEI, F. et al. BER analysis for uplink NOMA in asymmetric channels. **IEEE Communications Letters**, v. 24, n. 11, p. 2435–2439, nov. 2020.

WU, N. et al. A CNN-based end-to-end learning framework towards intelligent communication systems. **IEEE Access**, v. 7, p. 110197–110204, 2019.

WU, X.; JIANG, M.; ZHAO, C. Decoding optimization for 5G LDPC codes by machine learning. **IEEE Access**, v. 6, p. 50179–50186, 2018.

YAO, J.-L. et al. Chaos-based wireless communication resisting multipath effects. **Physical Review E**, v. 96, n. 3, p. 032226, sep. 2017.

YAO, J.-L. et al. Experimental wireless communication using chaotic baseband waveform. **IEEE Transactions on Vehicular Technology**, v. 68, n. 1, p. 578–591, jan. 2019.

YEOM, J. S. et al. BER performance of uplink NOMA with joint maximum-likelihood detector. **IEEE Transactions on Vehicular Technology**, v. 68, n. 10, p. 10295–10300, oct. 2019.

YUAN, J. et al. Performance and design of space-time coding in fading channels. **IEEE Transactions on Communications**, v. 51, n. 12, p. 1991–1996, dec. 2003.