Low-Density Parity-Check Code Design for the AWGN Channel with Additive Radar Interference

BY

FEDERICO BRUNERO
B.S. in Telecommunications Engineering, Politecnico di Torino, Torino, Italy, 2017

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:

Natasha Devroye, Chair and Advisor
Daniela Tuninetti
Roberto Garello, Politecnico di Torino

ACKNOWLEDGMENTS

First of all I would like to thank my entire committee, Professor Natasha Devroye, Professor Daniela Tuninetti and Professor Roberto Garello. The quality of this thesis is a direct consequence of their careful supervision on my research study and I am proud of their trust and faith in my talent. I was very lucky to meet them and to work with them, since it is thanks to them and to the work of these intense months that I decided to go for a PhD. Hoping to meet them again in the future and to work together again, I thank them for their support and their willingness during this last year.

I thank all my friends. I wish the best to all those that shared this extraordinary experience in Chicago with me: most of them will stay in the US and I hope that they will achieve their goals in this land of opportunities. I thank all those in Italy that supported me since a long time and that grew up with me during these years, making me a better person. I hope to keep in touch with all of them wherever I will be in the next years.

I want to give special thanks to my parents, Paola and Roberto, to my sister Chiara, to my grandmothers, Didi and Lina, to my whole family. I thank them for loving me and supporting me not only financially, but also and especially in the soul for all those decisions I had to take during this last year. This wonderful experience has been possible thanks to them only and I hope to pay them back with all the results that I will achieve in the next future.

Finally, I thank Ludovica, my love. If I managed to face every difficulty while I was far away from home for the first time, it is thanks to her. Her comforting words and her love, which

ACKNOWLEDGMENTS (continued)

I never felt weakening during these months, helped me overcome moments of loneliness and difficulty. She always pushed me to do better and better, therefore I am sure I will continue to do so in the future, wherever we will be together again.

FB

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARI	Additive Radar Interference
BER	Bit Error Rate
IC	Interference Cancellation
INR	Interference-to-Noise Ratio
LLR	Log-Likelihood Ratio
MAP	Maximum A Posteriori
ML	Maximum Likelihood
SNR	Signal-to-Noise Ratio
SPA	Sum-Product Algorithm
TIN	Treat Interference as Noise

SUMMARY

This thesis considers the problem of code design for a channel where communications and radar systems coexist, modeled as having both Additive White Gaussian Noise (AWGN) and Additive Radar Interference (ARI). The question of how to adapt or re-design convolutional codes (decoded by the Viterbi algorithm) and LDPC codes (decoded by the sum-product algorithm and optimized by using the EXIT chart method) to effectively handle the overall non-Gaussian ARI noise is investigated. A decoding metric is derived from the non-Gaussian ARI channel transition probability as a function of the Signal-to-Noise Ratio (SNR) and Interference-to-Noise Ratio (INR).

Two design methodologies are benchmarked against a baseline "unaltered legacy system", where a generic code (i.e, a code not specifically designed for the non-Gaussian ARI channel) is used on the radar interfered channel and is decoded by using the AWGN-only metric (i.e., as if INR is zero). The methodologies are: M1) generic codes decoded with the new metric that accounts for both SNR and INR; M2) codes optimized for the overall non-Gaussian ARI channel. Both methodologies give better average Bit Error Rate (BER) in the high INR regime compared to the baseline, while in the low INR regime they perform as the baseline, since in this case the radar interference is weak. Interestingly, the performance improvement of M2 over M1 is minimal. In practice, this implies that specifications in terms of channel error correcting codes for commercially available wireless systems need not to be changed, and that it suffices to use an appropriate INR-based decoding metric in order to effectively cope with the ARI.

CHAPTER 1

INTRODUCTION

Consider a physical channel over which we want to send several streams of information. Different multiplexing techniques have been developed in the past in order to allow the simultaneous transmission over the same physical channel:

- Frequency Division Multiplexing (FDM): this technique consists on allocating disjoint spectral bands for each stream of information at the transmitter side on the same physical link;
- Time Division Multiplexing (TDM): this technique consists on allocating disjoint time slots for each stream of information;
- Space Division Multiplexing (SDM): this technique, often coupled with FDM, consists on using the same frequency on two different spatial areas far enough such that transmissions over the same frequency do not interfere.

However, the demand for wireless services has increased in the last years, thus new solutions to open up the spectrum are required. One new approach is to let communications and radar systems share frequency bands and in this context it has been proposed a new channel model surveyed in [1; 2] and references therein.

1.1 Past Work

From the perspective of a communications system only (where the communications system seeks to adapt to the unalterable and uncooperative radar system), the authors in [1] investigated the *Shannon capacity* of the AWGN channel suffering from an additive constant-modulo interference caused by a co-existing radar transmission. Interesting results about the capacity achieving channel input distribution were obtained and it was shown that, when the radar interference is larger than the signal of interest, the (complex-valued) channel "loses" one of the two real-valued dimensions.

In [2] the Symbol Error Rate (SER) for the channel model in [1] was investigated for commercially employed *uncoded* modulation systems. The authors derived decoding regions for optimal and suboptimal detection schemes, and designed optimal signal constellations such that either the transmission rate was maximized or the SER was minimized, including extensions to OFDM systems. In particular, they verified what had already been observed in [1]: the radar interference can be treated as Gaussian noise when it is weak, while it must be dealt properly when its power is very strong. In the latter case one of the two real-valued dimensions in the complex-valued received signal is lost.

To the best of our knowledge powerful *error correcting codes* have not been investigated for the channel model in [1], not even LDPC codes, a widely studied class of powerful error correcting codes originally introduced by Gallager in [3]. The authors in [4] examined the design of capacity-approaching irregular LDPC codes; they focused mainly on channels with binary inputs (e.g., BEC, BSC and binary-input AWGN channel) and they exhibited important

analytic properties of the so-called *density evolution*, a fundamental tracking procedure for the evolution of message distribution in the Belief Propagation (BP) decoding. Considering the binary-input AWGN channel, they designed an LDPC code only 0.13 dB away from capacity and with threshold mere 0.06 dB away from capacity; similarly, in [5] a discretized version of the density evolution was introduced for the design of LDPC codes with threshold a few thousandths away from the Shannon limit. In [6] the so-called "Gaussian approximation" was proposed to simplify the density evolution analysis of LDPC codes and it has become the method of choice also for the so-called "EXIT chart method" [7, Chapter 4, p. 238], technique originally introduced by ten Brink in [8] in the context of turbo codes, but used for the design of LDPC codes as well.

In spite of the briefly presented wide literature on LDPC codes and of excellent results obtained in term of codes designed for the binary-input AWGN channel, we are not aware of any LDPC code designed for the communications and radar channel model in [1; 2]. This thesis takes the first step towards examining the performance of practical *coded* systems for wireless channels that suffer from additive radar interference, with the final goal of understanding how current wireless system coding schemes should be modified to work properly in the radar interfered channel.

1.2 Contributions

In this thesis we study the problem of code design for the channel model from [1] that is characterized by two parameters, both known at the communications receiver: the Signal-to-Noise Ratio (SNR) for the signal of interest, and the Interference-to-Noise Ratio (INR) for the

nuisance radar signal. Based on the channel transition probability, we derive a decoding metric that is a function of both SNR and INR, and that reduces to the AWGN-only one when the INR is zero.

Our baseline is an "unaltered legacy system", where a generic code, namely a code that is not specifically designed for the non-Gaussian ARI channel, is used on the radar interfered non-AWGN channel and is decoded using the AWGN-only metric, i.e., we use the decoding metric derived for the AWGN channel; we shall refer to this baseline as M0. We explore the following two design methodologies: M1) generic codes used on the radar interfered non-AWGN channel and decoded with the new metric that accounts for both SNR and INR; M2) codes optimized for the overall non-Gaussian channel. Intuitively, methodology M2 outperforms methodology M1 for "reasonably" optimized codes, while methodology M1 outperforms baseline M0. Moreover, it is expected that M0, M1 and M2 give essentially the same performance when INR is negligible with respect to the SNR.

For methodology M1, we first analyze convolutional codes decoded by the Viterbi algorithm [9], and then LDPC codes decoded by the sum-product algorithm [7]; for the latter, we take into account the modified version presented in [10] that offers a better performance than the standard tanh-based algorithm, which suffers of numerical overflows in the high SNR region. For methodology M2, we design LDPC codes for the non-Gaussian ARI channel by means of the EXIT chart method with Gaussian approximations [7, Chapter 4, p. 238] and some optimization tricks presented in [11].

Both methodologies M1 and M2 give better average Bit Error Rate (BER) compared to the baseline M0 in the high INR regime. Instead, M1 and M2 perform as M0 in the low INR regime since in this case the radar interference is weak and the non-Gaussian ARI channel reduces to the AWGN-only channel. Interestingly, we observe that M2 only offers a small improvement over M1 in general. This somewhat surprising result may be interpreted as follows, from a capacity standpoint. A Gaussian input is known to be optimal for AWGN-only channels, thus "good codes" for AWGN-only channels should "look like an iid Gaussian input". In [1] it was shown that a Gaussian input is asymptotically optimal except when INR $\approx$ SNR, therefore a "good code" for AWGN-only (which looks like a Gaussian input) should be "good" for the non-Gaussian ARI too, when it is decoded using the non-Gaussian ARI-derived decoding metric (except possibly when INR $\approx$ SNR). In practice, this may imply that current wireless system specifications in terms of forward error correcting channel codes need not to be changed, and in order to effectively cope with the additional radar interference it suffices to use the appropriate INR-based decoding metric with the already existing codes.

1.3 Thesis Organization

In Chapter 2 we introduce the novel channel model and the code schemes studied. In particular, in Section 2.1 we introduce the channel model from [1] and its channel transition probability, which is a fundamental parameter for our work; we briefly present results from [2] regarding the detectors for the uncoded system, which is used in computer simulations as reference. In Section 2.3 and Section 2.4 we derive the new decoding metric for the Viterbi algorithm and the new initialization procedure for the sum-product algorithm, using the AWGN-only case

as reference. In Section 2.5 we describe the EXIT chart method for the design of LDPC codes and we explain how the method can be modified according to the new channel model.

In Chapter 3 we present the results from our computer simulations. In Section 3.1 we report results both for the modified Viterbi algorithm and for the modified sum-product algorithm, testing generic codes. In Section 3.2 we present results when LDPC codes are optimized for the channel model with radar interference and decoded with the proper INR-based initialization of the sum-product algorithm. In all cases we evaluate the performance in terms of BER and compare with both the baseline M0 and the detectors for the uncoded system.

In Chapter 4 we conclude the thesis summarizing relevant results and briefly proposing future research.

CHAPTER 2

CHANNEL MODELS AND CODES USED

In the following plain capital letters represent random variables and plain lower case letters represent their realizations; bold capital letters represent matrices and bold lower case letters represent vectors. The codeword $\boldsymbol{x} = (x_0, x_1, \cdots, x_{N-1})$ and the complex-valued received sequence $\boldsymbol{y} = (y_0, y_1, \cdots, y_{N-1})$ are sequences of length N; we use ℓ as index to indicate the ℓ-th bit in a vector sequence. In this thesis we only report results for the case where the coded bits are mapped to the BPSK constellation, thus the coded sequence/codeword and the channel output sequence have the same length.

We consider two channel models: a) the classical AWGN channel; b) the AWGN channel with ARI from [1].

As already introduced in Chapter 1, we want to study the behavior of the Viterbi algorithm and the sum-product algorithm, when convolutional and LDPC codes are used over the non-Gaussian ARI channel. First, we recall each algorithm for the standard AWGN channel; secondly, we modify the decoding metric according to the AWGN channel with ARI, developing in such a way the metric needed in methodology M1. Finally, we develop methodology M2 introducing the EXIT chart method for the optimization of LDPC codes in the non-Gaussian ARI channel.

2.1 AWGN Channel and AWGN Channel with ARI

Consider the AWGN channel model

$$Y = \sqrt{S}X + Z, \tag{2.1}$$

where the transmitted symbol X is from the BPSK constellation $\mathcal{X} \in \{-1, +1\}$, S is the average Signal-to-Noise Ratio (SNR), $Z \sim \mathcal{N}(0, 1)$ is the complex-valued normalized Gaussian noise and Y is the output of the channel. Since the density of Z is given by

$$f_Z(z) = \frac{1}{\pi\sigma^2}\,\mathrm{e}^{-\frac{|z|^2}{\sigma^2}} = \frac{1}{\pi}\,\mathrm{e}^{-|z|^2}, \tag{2.2}$$

the channel transition probability is simply

$$f_{Y|X}(y \mid x) = f_Z\big(y - \sqrt{S}x\big). \tag{2.3}$$

Consider now the AWGN channel model with Additive Radar Interference (ARI)

$$Y = \sqrt{S}X + \sqrt{I}\,\mathrm{e}^{\mathrm{j}\Theta} + Z, \tag{2.4}$$

where the terms X, S and Z are defined as before, Θ is the radar phase uniformly distributed in the interval $[0, 2\pi)$ and I is the INR. We have mutually independence for the random variables

(X, Θ, Z) and the pair $(S, I) \in \mathbb{R}_+^2$ is fixed and known at the communications receiver. Let $W \coloneqq \sqrt{I}\,\mathrm{e}^{\mathrm{j}\Theta} + Z$; its density is given by

$$f_W(w) = \frac{\mathrm{e}^{-|w|^2 - I}}{\pi} I_0\left(2\sqrt{I}|w|\right), \tag{2.5}$$

where I_0 is the modified Bessel function of the first kind and order zero. The channel transition probability is simply

$$f_{Y|X}(y \mid x) = f_W\big(y - \sqrt{S}x\big). \tag{2.6}$$

For our purposes the channel transition probability is the main information we need from each channel model; moreover, note that when $I = 0$, the density in Equation 2.5 reduces to the density of the AWGN channel in Equation 2.2, since we have $I_0(z) = 1$ when $z = 0$.

2.2 Detectors in the Uncoded System

Considering the channel model in Equation 2.4, three detectors have been developed in [2] for the uncoded system scenario.

The *Maximum Likelihood* (ML) detector is the optimal symbol-by-symbol maximum a posteriori detector and it estimates the transmitted symbol x such that

$$\hat{x}^{(\mathrm{OPT})}(y) = \underset{x \in \{-1,+1\}}{\arg\max}\ f_{Y|X}(y \mid x) \tag{2.7}$$

$$= \underset{x \in \{-1,+1\}}{\arg\min}\ \left(|y - \sqrt{S}x| - \ln I_0\left(2\sqrt{I}|y - \sqrt{S}x|\right)\right). \tag{2.8}$$

The *Treat Interference as Noise* (TIN) detector is approximately optimal in the low INR regime, where the estimated transmitted symbol $\hat{x}$ is given by

$$\hat{x}^{(\mathrm{OPT})}(y) \approx \underset{x\in\{-1,+1\}}{\arg\min} |y - \sqrt{S}x|^2 =: \hat{x}^{(\mathrm{TIN})}(y), \tag{2.9}$$

if we consider in Equation 2.8 the low INR condition $I \ll S$ and consequently $I_0(z) \approx 1$ for $|z| \ll 1$.

The *Interference Cancellation* (IC) detector is approximately optimal in the high INR regime, where in this case the estimated transmitted symbol $\hat{x}$ is given by

$$\hat{x}^{(\mathrm{OPT})}(y) \approx \underset{x\in\{-1,+1\}}{\arg\min} \left(|y - \sqrt{S}x| - \sqrt{I}\right)^2 =: \hat{x}^{(\mathrm{IC})}(y), \tag{2.10}$$

if we consider in Equation 2.8 the high INR condition $I \gg S$ and consequently $I_0(z) \approx \mathrm{e}^{|z|}$ for $|z| \gg 1$.

2.3 The Viterbi Algorithm

The Viterbi algorithm is the optimal Maximum A Posteriori (MAP) sequence detector for the decoding of convolutional codes, i.e., it always finds the maximum likelihood path through the trellis. In practice, what has to be maximized is the log-likelihood function $\ln f_{Y|X}(\boldsymbol{y} \mid \boldsymbol{x})$, where $\boldsymbol{y}$ is the received sequence and $\boldsymbol{x}$ is the transmitted codeword; since in the following we consider only Discrete Memoryless Channels (DMC), we have

$$f_{Y|X}(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{\ell=0}^{N-1} f_{Y|X}(y_\ell \mid x_\ell), \tag{2.11}$$

where $f_{Y|X}(y_\ell \mid x_\ell)$ is the channel transition probability. Considering then a BPSK modulation with equally likely codewords, this is a minimum error probability decoding rule.

2.3.1 The Decoding Metric for the AWGN Channel

When designing coded schemes for the memoryless channel in Equation 2.1, one needs the following log-likelihood function of the received sequence $\boldsymbol{y}$ given the transmitted codeword $\boldsymbol{x}$

$$M(\boldsymbol{y} \mid \boldsymbol{x}) = \ln f_{Y|X}(\boldsymbol{y} \mid \boldsymbol{x}) = \ln \prod_{\ell=0}^{N-1} f_{Y|X}(y_\ell \mid x_\ell) \tag{2.12}$$

$$= \sum_{\ell=0}^{N-1} \ln f_{Y|X}(y_\ell \mid x_\ell) \tag{2.13}$$

$$= \sum_{\ell=0}^{N-1} \ln \left(\frac{1}{\pi} \, \mathrm{e}^{-|y_\ell - \sqrt{S} x_\ell|^2} \right) \tag{2.14}$$

$$= N \ln(1/\pi) - \sum_{\ell=0}^{N-1} |y_\ell - \sqrt{S} x_\ell|^2 \tag{2.15}$$

$$= 2\sqrt{S} \sum_{\ell=0}^{N-1} \Re(y_\ell) x_\ell + N \left[\ln(1/\pi) - S \right] - |\boldsymbol{y}|^2 \tag{2.16}$$

$$= C_1 \left(\Re(\boldsymbol{y}) \cdot \boldsymbol{x} \right) + C_2, \tag{2.17}$$

with $C_1 = 2\sqrt{S}$ and $C_2 = N \left[\ln(1/\pi) - S \right] - |\boldsymbol{y}|^2$. Since constants C_1 and C_2 do not depend on the codeword $\boldsymbol{x}$ and $C_1 > 0$, the log-likelihood function is maximized when the correlation between $\boldsymbol{x}$ and $\Re(\boldsymbol{y})$ is maximum, therefore we have

$$M(\boldsymbol{y} \mid \boldsymbol{x}) = \Re(\boldsymbol{y}) \cdot \boldsymbol{x} \tag{2.18}$$

as path metric and $M(y_\ell \mid x_\ell) = \Re(y_\ell) x_\ell$ as bit metric.

2.3.2 The Decoding Metric for the AWGN Channel with ARI

When considering the non-Gaussian ARI channel, the derivation of the decoding metric is similar to the AWGN-only case. The only difference relies on the fact that now we consider the channel transition probability in Equation 2.5. The channel is again memoryless, thus the log-likelihood function of the received sequence $\boldsymbol{y}$ given the transmitted codeword $\boldsymbol{x}$ is given by

$$
\begin{aligned}
M(\boldsymbol{y} \mid \boldsymbol{x}) &= \ln f_{Y|X}(\boldsymbol{y} \mid \boldsymbol{x}) = \ln \prod_{\ell=0}^{N-1} f_{Y|X}(y_\ell \mid x_\ell) && (2.19)\\
&= \sum_{\ell=0}^{N-1} \ln f_{Y|X}(y_\ell \mid x_\ell) && (2.20)\\
&= \sum_{\ell=0}^{N-1} \ln \left[\frac{\mathrm{e}^{-|y_\ell - \sqrt{S}x_\ell|^2 - I}}{\pi} I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}x_\ell|\right) \right] && (2.21)\\
&= \sum_{\ell=0}^{N-1} \left[-|y_\ell - \sqrt{S}x_\ell|^2 - I + \ln I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}x_\ell|\right) - \ln(\pi) \right] && (2.22)\\
&= 2\sqrt{S} \sum_{\ell=0}^{N-1} \left[\frac{1}{2\sqrt{S}} \ln I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}x_\ell|\right) + \Re(y_\ell)x_\ell \right] - \alpha. && (2.23)
\end{aligned}
$$

Since the term $2\sqrt{S} > 0$ and $\alpha = N[\ln(\pi) + I + S] + |\boldsymbol{y}|^2$ is independent of the received codeword, the log-likelihood function is maximized when the first sum is maximum. Hence, we can use the following

$$
M(\boldsymbol{y} \mid \boldsymbol{x}) = \sum_{\ell=0}^{N-1} \left[\frac{1}{2\sqrt{S}} \ln I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}x_\ell|\right) + \Re(y_\ell)x_\ell \right] \tag{2.24}
$$

as path metric and the argument of the sum, i.e. $M(y_\ell \mid x_\ell) = \frac{1}{2\sqrt{S}} \ln I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}x_\ell|\right) + \Re(y_\ell)x_\ell$, as bit metric.

It is straightforward to show that, for $I = 0$, the metric in Equation 2.24 reduces to Equation 2.18 since $I_0(z) = 1$ when $z = 0$, therefore the modified metric automatically adjusts to the AWGN-only metric when the radar interference is not present. This behavior is confirmed by computer simulations presented in Chapter 3 and is consistent with the last observation in Section 2.1 about the channel transition probability of the non-Gaussian ARI channel, which reduces to the AWGN-only one when the radar interference is absent.

2.4 The Sum-Product Algorithm

The Sum-Product Algorithm (SPA) is an approximation of the optimal maximum a posteriori symbol-by-symbol detector and is one of the most efficient iterative algorithms for the decoding of LDPC codes; it is essentially based on the computation of marginal *a posteriori bit probabilities* and it decodes symbols through an iterative procedure that improves the reliability of each decoded code symbol.

We consider the sum-product algorithm in the log-domain, in particular we implement the tanh-based SPA, considering the modified hyperbolic tangent function as in [10]. We take into account this particular version of the SPA in order to reduce the error floor due to an overflow of the decoder, when the argument x of the tanh function is large enough for having $\tanh(x) \to \pm 1$ and consequently $\tanh^{-1}(x) \to \pm\infty$.

Adopting the same notation from [10], we briefly present the description of the algorithm. Assume a binary LDPC code with parity-check matrix $\boldsymbol{H}$ with size $M \times N$, where M and N are the number of check and variable nodes, respectively. Let $M(\ell)$ and $N(m)$ be the set of the check nodes connected to the variable node ℓ and the set of variable nodes connected to the

check node m, respectively. The set $M(\ell) \setminus m$ is simply the set $M(\ell)$ without the m-th check node, while similarly $N(m) \setminus \ell$ is the set $N(m)$ without the ℓ-th variable node. Recall that we use vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, both of length N, to identify the transmitted codeword and the associated received sequence. The sum-product algorithm can be summarized in three main steps.

1. Initialization. We assign to each variable node ℓ the a posteriori Log-Likelihood Ratio (LLR) defined as

$$\mathrm{LLR}(y_\ell) \coloneqq \ln \frac{f_{X|Y}(x_\ell = -1 \mid y_\ell)}{f_{X|Y}(x_\ell = +1 \mid y_\ell)}. \tag{2.25}$$

   The initialization is done in position (m, n) where $\boldsymbol{H}_{m,n} = 1$ as

$$\lambda_{\ell \to m}(y_\ell) = \mathrm{LLR}(y_\ell), \tag{2.26}$$

$$\Lambda_{m \to \ell}(y_\ell) = 0, \tag{2.27}$$

   where the term $\lambda_{\ell \to m}(y_\ell)$ is the LLR value of the message sent by the variable node ℓ to the check node m, based on all the check node messages involving ℓ except m, and $\Lambda_{m \to \ell}(y_\ell)$ is the LLR value of the message sent by the check node m to the variable node ℓ, based on all the variable node messages involving m except ℓ.

2. Iterative process. The check and variable node LLR values are updated as

$$\Lambda_{m \to \ell}(y_\ell) = 2 \tanh^{-1} \left[\prod_{\ell' \in N(m) \setminus \ell} \tanh \left(\frac{\lambda_{\ell' \to m}(y_{\ell'})}{2} \right) \right], \tag{2.28}$$

$$\lambda_{\ell \to m}(y_\ell) = \mathrm{LLR}(y_\ell) + \sum_{m' \in M(\ell) \setminus m} \Lambda_{m' \to n}(y_\ell), \tag{2.29}$$

$$\lambda_\ell(y_\ell) = \mathrm{LLR}(y_\ell) + \sum_{m \in M(\ell)} \Lambda_{m \to \ell}(y_\ell). \tag{2.30}$$

3. Decision. If $\lambda_\ell(y_\ell) \geq 0$, then $\hat{x}_\ell = 0$, otherwise $\hat{x}_\ell = 1$. If the syndrome $\hat{\boldsymbol{x}}\boldsymbol{H}^{\mathrm{T}} = \mathbf{0}$, the algorithm stops, otherwise all the procedure is repeated until a fixed number of decoding iterations is reached.

Authors in [10] applied a slight modification in Equation 2.28, introducing the modified tanh function

$$\tanh_{\text{modified}}(x) = \begin{cases} \tanh(x), & \text{if } |x| < \alpha \\ \operatorname{sign}(x)\tanh(\alpha), & \text{if } |x| \geq \alpha \end{cases} \tag{2.31}$$

which performs a sort of decoding normalization; this technique is called *clipping* and they found $\alpha = 7$ as the best value with trial and error approach. Similarly, they introduced

$$\tanh^{-1}_{\text{modified}}(x) = \begin{cases} \tanh^{-1}(x), & \text{if } |x| < \beta \\ \operatorname{sign}(x)\tanh^{-1}(\beta), & \text{if } |x| \geq \beta \end{cases} \tag{2.32}$$

to perform a normalization for the inverse hyperbolic tangent, with $\beta = \tanh^{-1}(\alpha) = 0.999998$. In practice, one can choose if applying the normalization to the tanh function or to its inverse; we implement the tanh-based SPA with the normalization for the tanh function.

It is evident that the dependency on the channel model relies mainly in the initialization step of the algorithm, since the LLR in Equation 2.25 is a function of the channel transition probability of the model considered.

2.4.1 The SPA Initialization for the AWGN Channel

Referring to the LLR in Equation 2.25, we have to evaluate the term $f_{X|Y}(x_\ell \mid y_\ell)$. This can be easily done recalling that from Bayes' theorem we have

$$f_{X|Y}(x_\ell \mid y_\ell) = \frac{f_{Y|X}(y_\ell \mid x_\ell) f_X(x_\ell)}{f_Y(y_\ell)}, \tag{2.33}$$

so that, considering the binary-input assumption with equally likely codeword, i.e., $f_X(x_\ell) = 1/2$ and $x_\ell \in \{+1, -1\}$, we can also evaluate

$$f_Y(y_\ell) = \int_{x_\ell} f_{Y|X}(y_\ell \mid x_\ell) f_X(x_\ell)\, \mathrm{d}x_\ell \tag{2.34}$$

$$= \frac{1}{2}\left(f_Z\big(y_\ell - \sqrt{S}\big) + f_Z\big(y_\ell + \sqrt{S}\big)\right). \tag{2.35}$$

Without too much effort, we can rewrite Equation 2.33 as

$$f_{X|Y}(x_\ell \mid y_\ell) = \frac{f_Z\big(y_\ell - \sqrt{S}x_\ell\big)}{f_Z\big(y_\ell - \sqrt{S}\big) + f_Z\big(y_\ell + \sqrt{S}\big)} \tag{2.36}$$

and consequently Equation 2.25 becomes

$$\mathrm{LLR}(y_\ell) = \ln \frac{f_Z\big(y_\ell + \sqrt{S}\big)}{f_Z\big(y_\ell - \sqrt{S}\big)} \tag{2.37}$$

$$= |y_\ell - \sqrt{S}|^2 - |y_\ell + \sqrt{S}|^2 \tag{2.38}$$

$$= -4\Re(y_\ell)\sqrt{S}. \tag{2.39}$$

2.4.2 The SPA Initialization for the AWGN Channel with ARI

If we consider the non-Gaussian ARI channel instead of the AWGN-only channel, the procedure to evaluate the LLR in Equation 2.25 is the same as before: indeed, if we use the channel transition probability in Equation 2.5, we obtain

$$f_{X|Y}(x_\ell \mid y_\ell) = \frac{f_W\left(y_\ell - \sqrt{S}x_\ell\right)}{f_W\left(y_\ell - \sqrt{S}\right) + f_W\left(y_\ell + \sqrt{S}\right)} \tag{2.40}$$

and consequently the LLR becomes

$$\begin{aligned} \mathrm{LLR}(y_\ell) &= \ln \frac{f_W\left(y_\ell + \sqrt{S}\right)}{f_W\left(y_\ell - \sqrt{S}\right)} && (2.41)\\ &= |y_\ell - \sqrt{S}|^2 - |y_\ell + \sqrt{S}|^2 + \ln \frac{I_0\left(2\sqrt{I}|y_\ell + \sqrt{S}|\right)}{I_0\left(2\sqrt{I}|y_\ell - \sqrt{S}\right)} && (2.42)\\ &= -4\Re(y_\ell)\sqrt{S} + \ln I_0\left(2\sqrt{I}|y_\ell + \sqrt{S}|\right) - \ln I_0\left(2\sqrt{I}|y - \sqrt{S}|\right). && (2.43) \end{aligned}$$

Once again it is straightforward to show that the LLR initialization in Equation 2.43 reduces to Equation 2.39 when $I = 0$, thus also in this case the new SPA initialization automatically adapts to the given INR value, coinciding to the AWGN-only case when the radar interference is really weak or not present.

2.5 The EXIT Chart Method

The EXIT chart, originally introduced by ten Brink in [8] in the context of turbo codes and based on the extrinsic information exchanged between variable and check nodes, is a helpful

visualization tool of the asymptotic performance under Belief Propagation (BP) decoding and a successful technique for the design of LDPC codes as well.

In order to use the EXIT chart method, it is convenient to introduce the so-called *degree distributions from an edge perspective*, i.e., the polynomials

$$\lambda(x) = \sum_{i=2}^{d_{\mathrm{v}}} \lambda_i x^{i-1}, \tag{2.44a}$$

$$\rho(x) = \sum_{i=2}^{d_{\mathrm{c}}} \rho_i x^{i-1}, \tag{2.44b}$$

where the coefficients λ_i and ρ_i represent the fraction of edges connected to variable and check nodes of degree i; the terms d_{v} and d_{c} are the maximum degrees of the variable and check degree distributions. The pair (λ, ρ) represents an LDPC *ensemble* with *design rate* given by

$$r(\lambda, \rho) = 1 - \frac{\int_0^1 \rho(x)\,\mathrm{d}x}{\int_0^1 \lambda(x)\,\mathrm{d}x} = 1 - \frac{\sum_i \rho_i / i}{\sum_i \lambda_i / i}. \tag{2.45}$$

Consider for the moment a generic Binary Memoryless Symmetric (BMS) channel model with BPSK modulation[1] $0 \to +1$ and $1 \to -1$, such that the LLR is now given by

$$\mathrm{LLR}(y_\ell) = \ln \frac{f_{X|Y}(x_\ell = +1 \mid y_\ell)}{f_{X|Y}(x_\ell = -1 \mid y_\ell)}. \tag{2.46}$$

---

[1]We consider this mapping to be consistent with all the derivations in [7] concerning entropy and capacity associated with a symmetric L-density; the standard BPSK mapping is used in computer simulations, but the different convention does not affect the results.

We denote the conditional density of the random variable $\mathrm{LLR}(Y)$ on $X = +1$ as $\mathsf{a}_{\mathrm{BMS}}$, and refer to it as *L-density*. Similarly, we denote the conditional density of the random variable $D = \tanh(\mathrm{LLR}(Y)/2)$ on $X = +1$ as $\mathfrak{a}_{\mathrm{BMS}}$, and refer to it as *D-density*. The *entropy* H and the *capacity* C of the channel are given by

$$\mathsf{H} = \int_{-\infty}^{+\infty} \mathsf{a}_{\mathrm{BMS}}(u) \log_2 \left(1 + \mathrm{e}^{-u}\right) \mathrm{d}u \tag{2.47}$$

$$= \int_{-1}^{+1} \mathfrak{a}_{\mathrm{BMS}}(u) \log_2 \left(\frac{2}{1+u}\right) \mathrm{d}u \tag{2.48}$$

$$= 1 - \mathsf{C}. \tag{2.49}$$

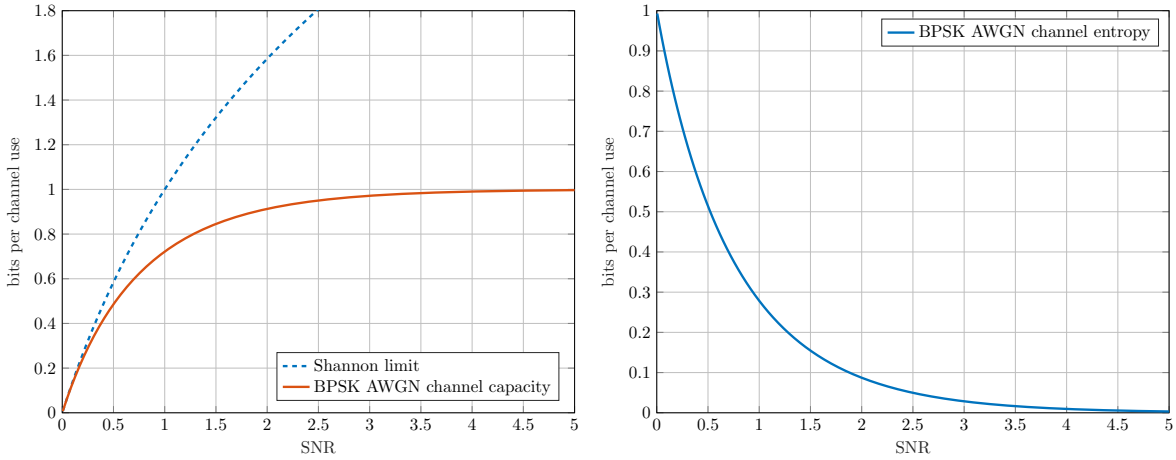
2.5.1 The EXIT Chart Method for the AWGN Channel

Let us consider the AWGN channel with BPSK modulation and the log-likelihood ratio in Equation 2.46 given by $\mathrm{LLR}(y_\ell) = 4\Re(y_\ell)\sqrt{S}$. It can be shown that the L-density and the D-density are given by

$$\mathsf{a}_{\mathrm{AWGN}} \sim \mathcal{N}(4S, 8S), \tag{2.50}$$

$$\mathfrak{a}_{\mathrm{AWGN}}(u) = \frac{1}{\sqrt{4\pi S}(1-u^2)} \, \mathrm{e}^{-\frac{(2S - \tanh^{-1}(u))^2}{4S}}. \tag{2.51}$$

The D-density is particularly convenient for the numerical evaluation of capacity and entropy shown in Figure 1.



(a) Capacity (b) Entropy

Figure 1. Capacity and entropy for the AWGN channel with BPSK modulation

Referring to the EXIT chart method stated in [7, Chapter 4, Definition 4.136], the entropy of the density at the output of check and variable nodes is given respectively by

$$\mathsf{h}_{\mathrm{c},\gamma} = \mathsf{H}(\rho(\mathsf{a}_{\mathsf{h}_{\mathrm{v},\gamma-1}})), \tag{2.52}$$

$$\mathsf{h}_{\mathrm{v},\gamma} = \mathsf{H}(\mathsf{a}_{\mathrm{AWGN}} \circledast \lambda(\mathsf{b}_{\mathsf{h}_{\mathrm{c},\gamma}})), \tag{2.53}$$

where:

- $\{\mathsf{b}_\mathsf{h}\}$ and $\{\mathsf{a}_\mathsf{h}\}$ are two families of symmetric L-densities;
- γ is the γ-th iteration of the BP decoding;
- $\lambda(\mathsf{b}_\mathsf{h}) = \sum_i \lambda_i \mathsf{b}_\mathsf{h}^{\circledast(i-1)}$ and $\rho(\mathsf{a}_\mathsf{h}) = \sum_i \rho_i \mathsf{a}_\mathsf{h}^{\boxplus(i-1)}$;

- the operators $\circledast$ and $\boxtimes$ represent the convolution in the L-domain and the convolution in the G-domain, respectively [7, p. 181].

Since we do not know the true "intermediate" densities a priori in the density evolution process, we pick for $\{\mathsf{b_h}\}$ and $\{\mathsf{a_h}\}$ some equivalent densities from another family of L-densities, where the equivalence rule consists in taking a new density with *equal entropy*. The preferred choice is to substitute the "intermediate" densities with $\{\mathsf{a}_{\mathrm{AWGN(h)}}\}$, i.e., with the L-densities associated to the binary-input AWGN channel with entropy h. The choice is particularly convenient because, as we already know, the L-density $\mathsf{a}_{\mathrm{AWGN(h)}}$ is a Gaussian with mean $\mu = 4S_0$ and variance $\sigma^2 = 2\mu = 8S_0$, where S_0 is the SNR such that the entropy is equal to h, thus it is possible to track each "intermediate" density with one parameter only, i.e., the mean of the Gaussian.

Given this Gaussian approximation and considering that the transmission takes place over the AWGN channel, the functions

$$v_{\hat{\mathsf{h}}}(\mathsf{h}) = \sum_i \lambda_i \psi\left((i-1)\psi^{-1}(\mathsf{h}) + \psi^{-1}(\hat{\mathsf{h}})\right), \tag{2.54}$$

$$c(\mathsf{h}) = 1 - \sum_i \rho_i \psi\left((i-1)\psi^{-1}(1-\mathsf{h})\right), \tag{2.55}$$

describe the entropy at the output of the variable and check node, respectively, as a function of the input entropy h. The function $\psi(m)$ is the entropy of a Gaussian with mean m e variance $2m$. Note that the function $c(\mathsf{h})$, beyond the Gaussian approximation itself, is actually the result of

another approximation due to the duality principle introduced by Chung in [12], which however yields a simple analytic expression of the EXIT function for the check nodes.

For the iterative algorithm to converge, the condition $v_{\hat{\mathsf{h}}}(c(\mathsf{h})) \leq \mathsf{h}$ must be satisfied by the quantities defined in Equation 2.54 and Equation 2.55. Therefore, if we fix the channel parameter $\hat{\mathsf{h}}$ and the ρ distribution, we can find the coefficients λ_i by running the following linear program [7; 11]

$$\max_{\lambda} \left\{ \sum_{i\geq 2} \frac{\lambda_i}{i} \mid \lambda_i \geq 0; \sum_{i\geq 2} \lambda_i = 1; \sum_{i\geq 2} \lambda_i \xi_{i,c} \leq \mathsf{h}; \mathsf{h} \in [0,1] \right\}, \tag{2.56a}$$

$$\xi_{i,c} \coloneqq \psi\left((i-1)\psi^{-1}(c(\mathsf{h})) + \psi^{-1}(\hat{\mathsf{h}})\right), \tag{2.56b}$$

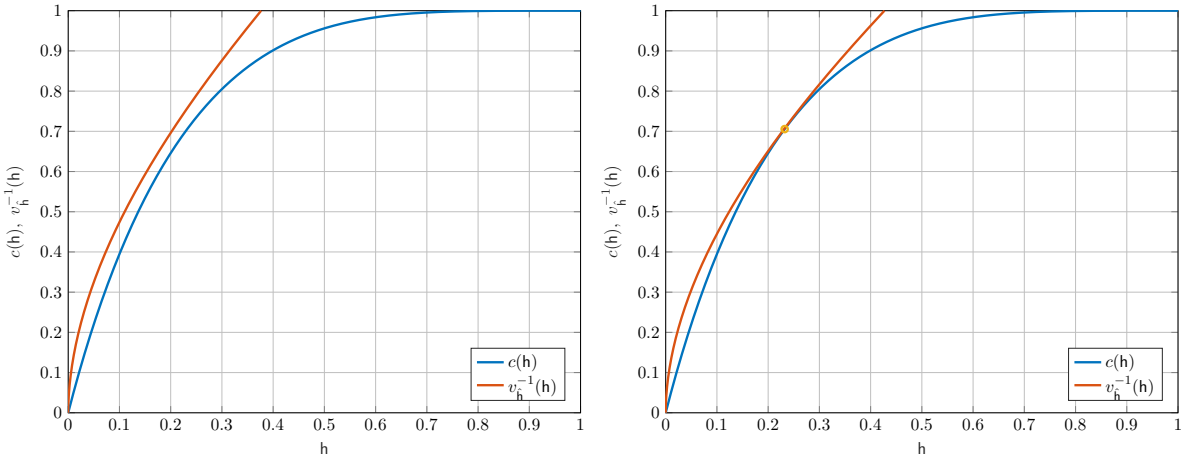
where the objective function is equivalent to the maximization of the code rate in Equation 2.45. Similarly, exchanging the roles of λ and ρ, and so keeping fixed the λ distribution, we can solve the linear program

$$\min_{\rho} \left\{ \sum_{i\geq 2} \frac{\rho_i}{i} \mid \rho_i \geq 0; \sum_{i\geq 2} \rho_i = 1; \sum_{i\geq 2} \rho_i \nu_{i,v} \geq 1 - \mathsf{h}; \mathsf{h} \in [0,1] \right\}, \tag{2.57a}$$

$$\nu_{i,v} \coloneqq \psi\left((i-1)\psi^{-1}(1 - v_{\hat{\mathsf{h}}}(\mathsf{h}))\right), \tag{2.57b}$$

to find the coefficients ρ_i, which maximize the code rate too.

In Figure 2 we can see an example of EXIT charts for the (x^3, x^6)-regular LDPC ensemble, where functions $c(\mathsf{h})$ and $v_{\hat{\mathsf{h}}}^{-1}(\mathsf{h})$ are plotted. We observe that for $\mathsf{h} \in [0,1]$ we always have $c(\mathsf{h}) \leq v_{\hat{\mathsf{h}}}^{-1}(\mathsf{h})$ when $\hat{\mathsf{h}} = 0.3765$, i.e., the convergence condition previously introduced is satisfied;



(a) $\hat{\mathsf{h}} = 0.3765$ (b) $\hat{\mathsf{h}} = 0.427$

Figure 2. EXIT chart for the (x^3, x^6)-regular LDPC ensemble on the AWGN channel and for two different channel parameters

instead, if $\hat{\mathsf{h}} = 0.427$, the curves $c(\mathsf{h})$ and $v_{\hat{\mathsf{h}}}^{-1}(\mathsf{h})$ touch in one point and when this particular condition is verified, the channel parameter $\hat{\mathsf{h}}$ is called *threshold*. In a few words, the *threshold* is the channel parameter associated to the minimum SNR such that the bit error probability goes to zero if the number of iterations in the algorithm increases to infinity. In the literature of LDPC codes, the threshold is a widely used parameter to compare different LDPC codes; in our study, we neglect the concept of threshold, since our goal is to obtain an LDPC code optimized for the non-Gaussian ARI channel with better performance than an AWGN-only optimized code, given the same rate code rate.

2.5.2 The EXIT Chart Method for the AWGN Channel with ARI

For the AWGN channel with ARI we optimize the degree distribution as before, taking advantage of a slightly modified version of the EXIT chart method.

In practice, we still consider the Gaussian approximation for the "intermediate densities" $\{\mathsf{b_h}\}$ and $\{\mathsf{a_h}\}$, but we evaluate numerically the convolution in Equation 2.53 with $\mathsf{a}_{\mathrm{AWGN+ARI}}$ as L-density[2].

Therefore, the new entropy at the output of the variable node as a function of the input entropy h is given by

$$v_{\hat{\mathsf{h}}}(\mathsf{h}) = \sum_i \lambda_i \mathsf{H}(\mathsf{a}_{\mathrm{AWGN+ARI}}(\hat{\mathsf{h}}) \circledast \mathcal{N}(m_i, 2m_i)), \tag{2.58}$$

where $m_i = (i-1)\psi^{-1}(\mathsf{h})$ is the mean of the Gaussian density. It is then easy to rewrite the linear program in Equation 2.56 as

$$\max_{\lambda} \left\{ \sum_{i\geq 2} \frac{\lambda_i}{i} \mid \lambda_i \geq 0; \sum_{i\geq 2} \lambda_i = 1; \sum_{i\geq 2} \lambda_i \xi_{i,c} \leq \mathsf{h}; \mathsf{h} \in [0,1] \right\}, \tag{2.59a}$$

$$\xi_{i,c} \coloneqq \mathsf{H}(\mathsf{a}_{\mathrm{AWGN+ARI}}(\hat{\mathsf{h}}) \circledast \mathcal{N}(m_{i,c}, 2m_{i,c})), \tag{2.59b}$$

$$m_{i,c} = (i-1)\psi^{-1}(c(\mathsf{h})). \tag{2.59c}$$

---

[2]Since the LLR expression for this channel model is complicated, the considered L-density is estimated numerically through the kernel density estimation.

Nothing changes instead for the linear program in Equation 2.57, since there is no dependency on the input L-density.

CHAPTER 3

SIMULATION RESULTS

In this chapter we present the results of our computer simulations for methodologies M1 and M2. We consider also the "unaltered legacy scheme" we refer to as M0: for this baseline scheme, we use generic codes (i.e., codes not properly designed for the AWGN+ARI channel) and we decode them with the AWGN-only metric. In particular, we look at two different cases: M0-a) an unaltered legacy system, where the decoding metric is as if the channel was $Y = \sqrt{S}X + Z$ (i.e., as if INR is zero); M0-b) the ARI is treated as a Gaussian noise, where the decoding metric is as if the channel was $Y = \sqrt{S/(1+I)}X + Z$. Note that we do not make any distinction between M0-a and M0-b when we consider the Viterbi algorithm, since the metric in Equation 2.18 is independent from $\sqrt{S}$ term. Instead, the SPA initialization depends on the $\sqrt{S}$ factor, as one can see from Equation 2.39, thus in this case we consider M0-a and M0-b cases.

We also plot the results of an uncoded system with the three *detectors* derived in [2] and briefly introduced in Section 2.2, as reference.

3.1 Methodology M1

3.1.1 Convolutional Codes

In this case we test the one-half rate convolutional code defined by octal digits $g^{(0)} = 5$ and $g^{(1)} = 7$ [9, p. 540]. In Figure 3 we plot the BER vs $(E_\mathrm{b}/N_0)_\mathrm{dB}$ for the AWGN-only case.

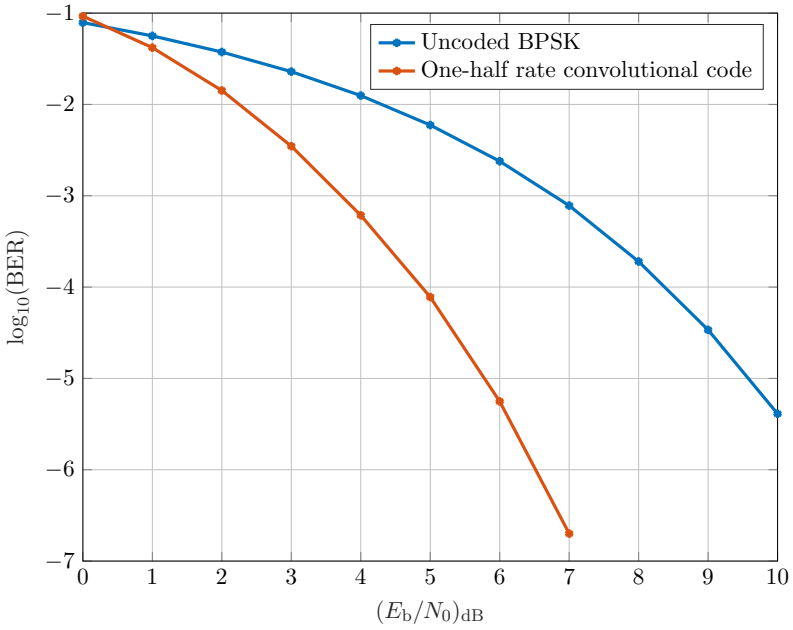


Figure 3. BER vs $(E_\mathrm{b}/N_0)_\mathrm{dB}$ for the uncoded BPSK system and the tested one-half rate convolutional code

In Figure 4 we compare the BER vs INR for the AWGN+ARI channel with fixed $S = 1\,\mathrm{dB}$ and BPSK modulation for the following receivers: baseline M0) the Viterbi algorithm with the standard AWGN-only channel metric; methodology M1) the Viterbi algorithm with the AWGN+ARI channel metric; three detectors derived in [2] for the uncoded system.

As already pointed out in [2] for uncoded systems, for coded systems we can also clearly identify two distinct regimes depending on whether we are in the low INR regime or in the high INR regime.

- When $I \ll S$, the performance of the Viterbi algorithm does not depend on the metric used; this behavior is expected as in this regime the radar interference has negligible

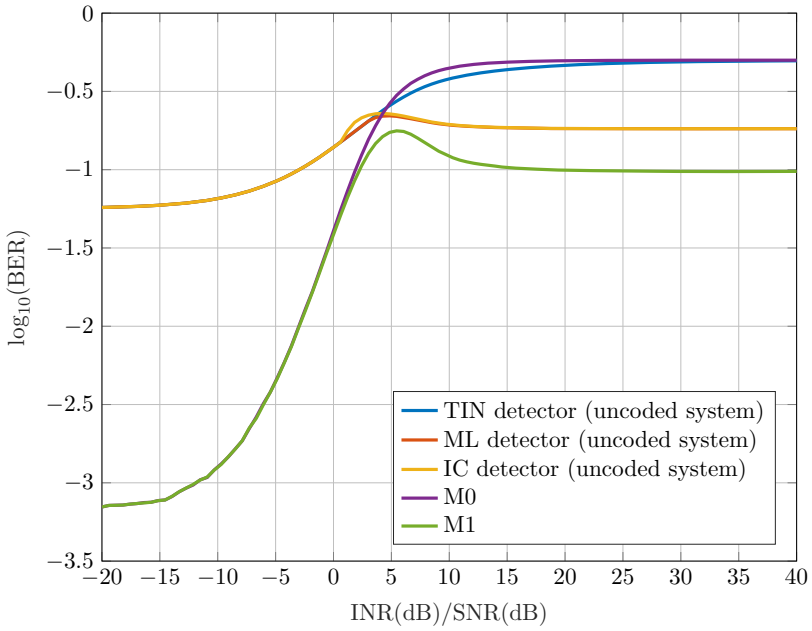


Figure 4. BER vs INR for BPSK modulation in the AWGN+ARI channel with $S_{\mathrm{dB}} = 1\,\mathrm{dB}$ (convolutional code)

power. As a sanity check, we can notice that the bottom asymptote in Figure 4 is, without surprise, the BER in Figure 3 for $(E_{\mathrm{b}}/N_0)_{\mathrm{dB}} = S_{\mathrm{dB}} - 10\log_{10}(R) \approx 4\,\mathrm{dB}$.

- When $I \gg S$, methodology M1 outperforms both baseline M0 and the uncoded schemes; this is expected as well, since in this regime the radar interference is very powerful and thus must be appropriately dealt with. Note that in this last case the baseline scheme gives BER $\approx 1/2$, which is what we would obtain if a "coin-flip" decision device was adopted; in other words, we would not even need to use a code.

We notice that, when using the new decoding metric, we need not distinguish between AWGN-only channel and AWGN+ARI channel, as the decoding metric for the AWGN+ARI channel with $I = 0$ gives the decoding metric for the AWGN-only channel, as previously specified. Therefore, in the AWGN+ARI channel one would need to measure both the SNR and the INR in order to appropriately tune the decoding metric, and not trivially "collapse" them into a single number SINR $= S/(1 + I)$.

Similar conclusions can be drawn if we keep the INR fixed and let the SNR vary, as one can see from Figure 5, where we plot the BER vs SNR for the AWGN+ARI channel with fixed values of I. On the one hand, it is evident from Figure 5c that BER curves coincide for both metrics when $I \ll S$; on the other hand, the modified metric leads to a better performance when $I \gg S$, i.e., when we are in the high INR regime.

3.1.2 LDPC Codes

In Figure 6 we fix $S = 1\,\text{dB}$ and plot the BER curves vs INR for: baseline M0-a) the SPA for the AWGN-only channel as if INR is zero; baseline M0-b) the SPA for the AWGN-only channel with SNR substituted with SINR; methodology M1) the SPA for the AWGN+ARI channel; ML detector for an uncoded system. We use the $(63, 37)$ EG-LDPC code, generated from the 2-D Euclidean geometry[1] EG$(2, 2^3)$, and set the maximum number of decoding iterations to 5, a good trade-off between the decoding speed and the performance [9, p. 884]. We see that for LDPC codes we can draw the same conclusions as before for the convolutional codes. In

[1]See Appendix A for more details about LDPC codes based on Euclidean Geometry.

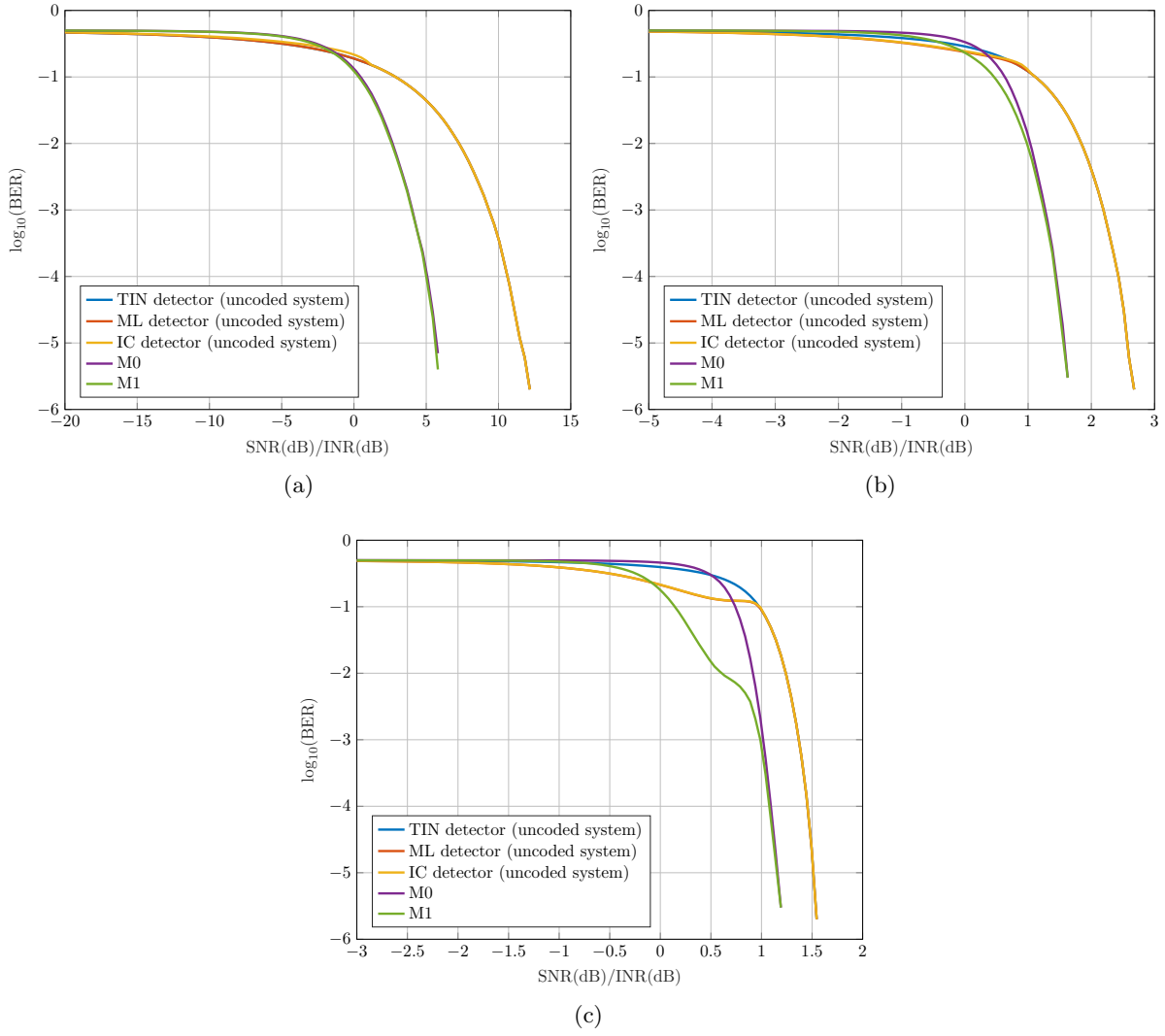


Figure 5. BER vs SNR for BPSK modulation in the AWGN+ARI channel with (a) $I = 1\,\mathrm{dB}$, (b) $I = 5\,\mathrm{dB}$ and (c) $I = 10\,\mathrm{dB}$ (convolutional code)

particular, we observe a slight improvement in the low INR region for baseline M0-b when the SNR parameter in baseline M0-a is replaced with the SINR term; this behavior is reasonable, since this methodology does not neglect completely the radar interference.

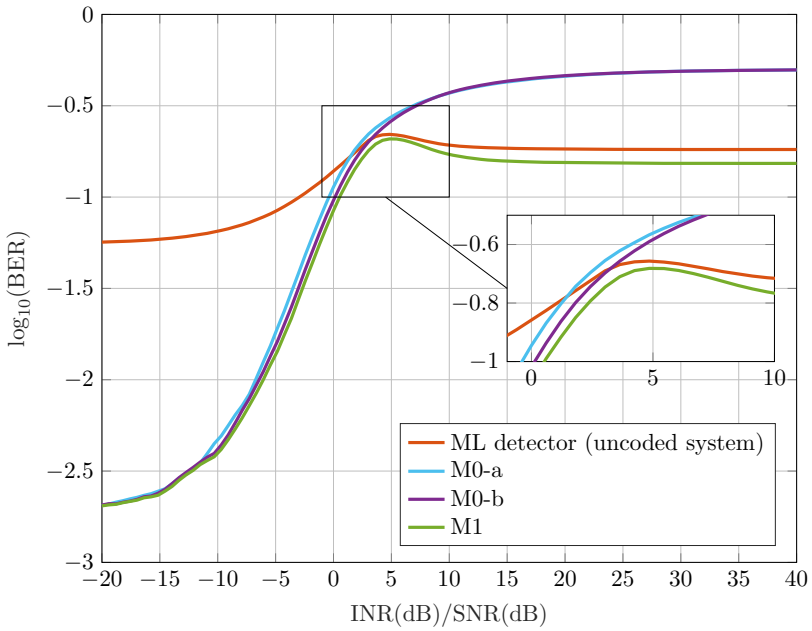


Figure 6. BER vs INR for BPSK modulation in the AWGN+ARI channel with $S_{\mathrm{dB}} = 1\,\mathrm{dB}$ (LDPC code)

3.2 Methodology M2

We run the optimization algorithm for methodology M2 as described in Section 2.5. The optimized degree distributions are reported in Table I, for which we set $d_{\mathrm{v}} = 30$ and $d_{\mathrm{c}} = 11$.

TABLE I: OPTIMIZED DEGREE DISTRIBUTIONS

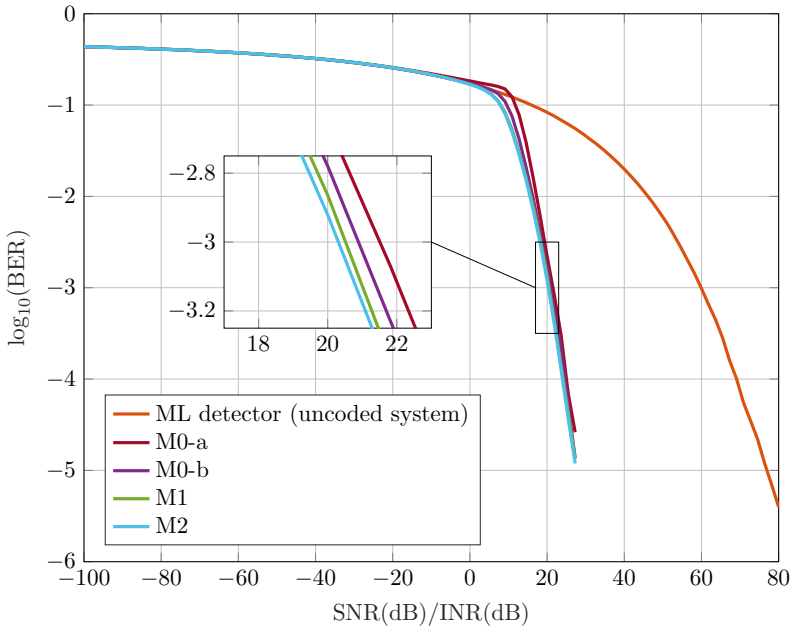
λ_i or ρ_i	AWGN-only	AWGN+ARI $S \gg I$	AWGN+ARI $S \ll I$
λ_2	0.1907	0.1937	0.1962
λ_3	0.0963	0.0596	0.0774
λ_4	0.1126	0.2325	0.1944
λ_5	0.1095		
λ_{30}	0.4909	0.5142	0.5320
ρ_{10}	0.5193	0.5266	0.4991
ρ_{11}	0.4807	0.4734	0.5009

For the optimization in the AWGN-only channel we choose $S_{\text{dB}} = -2.53\,\text{dB}$, while for the AWGN+ARI channel we consider two cases: $I \ll S$ with $S_{\text{dB}} = 0.45\,\text{dB}$, $I_{\text{dB}} = 0.15\,\text{dB}$, and $I \gg S$ with $S_{\text{dB}} = 2.75\,\text{dB}$, $I_{\text{dB}} = 8.25\,\text{dB}$.

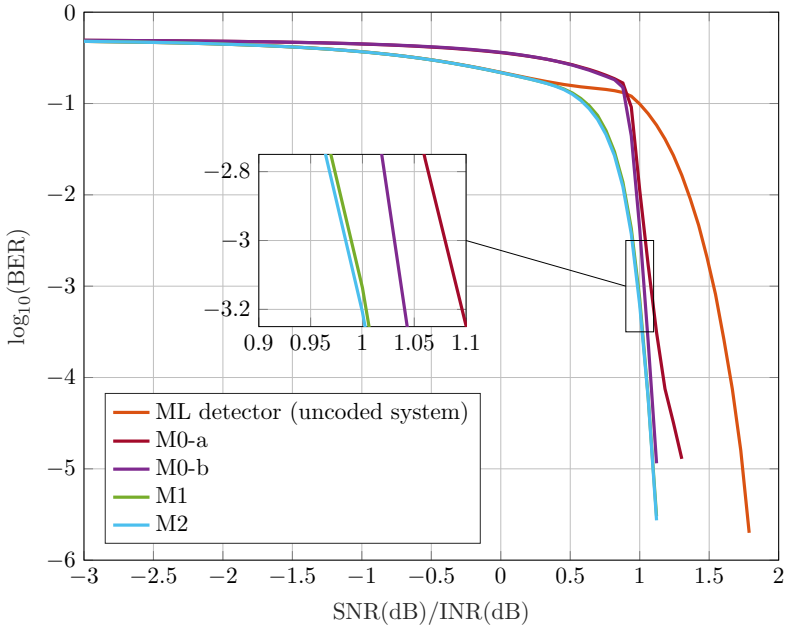
We convert our degree distributions into a parity-check matrix by means of the geometrical systematic approach[2] presented in [13; 9, Chapter 17, pp. 922-929] that avoids, by construction, the presence of cycles of length 4. We generate a $(4032, 1984)$ LDPC code with rate $R = 0.49$, so as to have a fair comparison between AWGN-only and AWGN+ARI codes.

In Figure 7 we plot the BER vs SNR curves for the following codes: M0-a) LDPC with AWGN-only initialization as if INR is zero and code optimized for the AWGN-only channel; M0-b) LDPC with AWGN-only initialization with the SINR instead of the SNR and code optimized for the AWGN-only channel; M1) LDPC with AWGN+ARI initialization and code

[2]See Appendix B for more details about the geometric approach for the construction of LDPC codes.



(a)



(b)

Figure 7. BER vs SNR for BPSK modulation in the AWGN+ARI channel with (a) $I_{\mathrm{dB}} = 0.15\,\mathrm{dB}$ and (b) $I_{\mathrm{dB}} = 8.25\,\mathrm{dB}$ (LDPC code)

optimized for the AWGN-only channel; M2) LDPC with AWGN+ARI initialization and code optimized for the AWGN+ARI channel. We also plot the BER of the ML detector for the uncoded system for comparison.

First of all, we observe that baseline M0-b outperforms again baseline M0-a, which shows in Figure 7b an error floor in spite of the modified SPA implementation from [10] that we adopted. Then, we see that the best BER is obtained, unsurprisingly, with methodology M2; however, somewhat surprisingly, the performance gain of methodology M2 over methodology M1 does not appear so significant (green and light blue lines in Figure 7 almost overlap).

This last observation can be explained by keeping in mind that codes designed for AWGN-only have been previously observed to behave well for a large class of channels [4]. We speculate that an even more accurate optimization (e.g., density evolution rather than EXIT charts) would not yield significantly better codes for the reasons outlined in Chapter 1.

We observe especially from Figure 7b that, for good BER performance, it is more important to use the correct decoding metric than to re-design the code. In practice, this is a good news: current wireless system specifications in terms of channel codes need not to be changed when the band is shared with radar systems, and in order to effectively cope with the extra interference it suffices to use the appropriate INR-based decoding metric.

CHAPTER 4

CONCLUSIONS

In this thesis we studied channel codes for wireless channels with radar interference. We derived a new decoding metric for the Viterbi algorithm and a new initialization for the sum-product algorithm whether convolutional codes or LDPC codes are used.

We compared two design methodologies: using generic codes (i.e., codes not explicitly designed for the non-Gaussian ARI channel) with the new decoding metric, and designing codes directly optimized for the radar interfered channel. We found that the BER improvement of the latter over the former is not so significant, meaning that in future wireless systems it is much more critical to use the correct decoder that to re-design codes.

This work focused on binary codes mapped onto the BPSK modulation; extensions to other modulation schemes, and possibly other classes of codes (such as Polar codes) is an interesting and unexplored research direction.

4.1 Thesis Contributions

- Derived new decoding metric for the Viterbi algorithm that accounts for both the SNR and the INR.
- Derived a new initialization of the sum-product algorithm that accounts for both the SNR and the INR.

- Optimized LDPC codes for the radar interfered channel by means of the EXIT chart method properly modified for the considered channel model.

4.2 Future Work

We propose the following future research.

- Extension of coding scheme analysis to other modulation schemes.
- Analysis of other classes of codes, e.g., Polar codes.
- Analysis of LDPC code optimization as a function of SNR and INR parameters, specifying a new concept of threshold given the pair (S, I).

APPENDICES

Appendix A

LDPC CODES BASED ON EUCLIDEAN GEOMETRY

LDPC codes can be constructed algebraically considering Euclidean and projective geometries over finite fields. In this appendix we focus on the so-called *type-I Euclidean geometry (EG)-LDPC codes*, in particular on the 2-D type-I cyclic $(0, s)$-th order EG-LDPC codes.

A.1 Euclidean Geometry over the Galois Field GF(2^s)

Consider all the possible m-tuples $(a_0, a_1, \ldots, a_{m-1})$, where each component a_i is taken from the Galois field GF(2^s). The total number of m-tuples $(2^s)^m = 2^{ms}$ over GF(2^s) forms an m-dimensional Euclidean geometry over GF(2^s), which we refer to as EG($m, 2^s$).

The Galois field GF(2^{ms}) forms an m-dimensional Euclidean geometry EG($m, 2^s$). Each element in GF(2^{ms}), by using the power representation, can be written like

$$\alpha^i = a_{i0} + a_{i1}\alpha + a_{i2}\alpha^2 + \cdots + a_{i,m-1}\alpha^{m-1}, \tag{A.1}$$

where $a_{ij} \in$ GF(2^s) $\forall i \in [0, m]$ and α is a primitive element of GF(2^{ms}); it is evident the *one-to-one correspondence* we have between each element α^i and the m-tuple $(a_{i0}, a_{i1}, \ldots, a_{i,m-1})$ over GF(2^s), therefore the Galois field GF(2^{ms}) can be considered as the Euclidean geometry EG($m, 2^s$).

For our purposes it is important to introduce the concepts of *point*, *line*, *flat*, *parallel bundle* and *incidence vector*.

Appendix A (continued)

- A *point* in EG$(m, 2^s)$ is simply an m-tuple $\boldsymbol{a} = (a_0, a_1, \ldots, a_{m-1})$. The *origin* is the point such that $a_i = 0 \; \forall i \in [0, m-1]$, i.e., $\boldsymbol{a} = \mathbf{0}$.

- Consider $\boldsymbol{a} \neq \mathbf{0}$. A *line* in EG$(m, 2^s)$ is the set of 2^s points $\{\beta\boldsymbol{a} : \beta \in \mathrm{GF}(2^s)\}$. Since for $\beta = 0$ we obtain the all-zero m-tuple, we say that this line passes through the origin; if instead we consider the set of 2^s points $\{\boldsymbol{a}_0 + \beta\boldsymbol{a} : \beta \in \mathrm{GF}(2^s),\ \boldsymbol{a}_0 \neq \mathbf{0}\}$ such that $\boldsymbol{a}_0$ and $\boldsymbol{a}$ are two linearly independent points, then we say that this line passes through the point $\boldsymbol{a}_0$. When two lines do not have any point in common, they are said to be *parallel*.

- For $1 \leq \mu \leq m$, a μ-flat is a μ-dimensional subspace of the vector space made up of all the m-tuples from the finite field GF(2^s) and consists of $2^{\mu s}$ points of the form

$$\boldsymbol{a}_0 + \beta_1\boldsymbol{a}_1 + \beta_2\boldsymbol{a}_2 + \cdots + \beta_\mu\boldsymbol{a}_\mu, \tag{A.2}$$

  with $\beta_i \in \mathrm{GF}(2^s)$ for $1 \leq i \leq \mu$. The μ-flat in Equation A.2 passes through the point $\boldsymbol{a}_0$, thus if $\boldsymbol{a}_0 = \mathbf{0}$, the μ-flat passes through the origin. If two μ-flats have no point in common, they are said to be disjoint. If $\mu = 1$, the 1-flat is a simple line.

- Given a μ-flat, there exist $2^{(m-\mu)s}$ parallel μ-flats which form a *parallel bundle*. Since each μ-flat contains $2^{\mu s}$ points, a parallel bundle contains all the 2^{ms} points in the considered Euclidean geometry.

Appendix A (continued)

- Let us consider a line $\mathcal{L}$ that does not pass through the origin (i.e., the point $\alpha^{\infty} = 0$ is not considered). The *incidence vector* for the line $\mathcal{L}$

$$\boldsymbol{v}_{\mathcal{L}} = (v_0, v_1, \ldots, v_{2^{ms}-2}) \tag{A.3}$$

  is the the vector whose i-th component v_i is 1 if the point α^i is contained in the line $\mathcal{L}$.

A.2 2-D Type-I Cyclic EG-LDPC Codes

Given the description of an m-dimensional Euclidean geometry over GF(2^s), the construction of a 2-D type-I cyclic EG-LDPC code based on EG($2, 2^s$) is relatively simple. Let us consider the incidence vector $\boldsymbol{v}_{\mathcal{L}}$ of a line $\mathcal{L}$ in EG($2, 2^s$) not passing through the origin; we can build the $(2^{2s} - 1) \times (2^{2s} - 1)$ square parity-check $\boldsymbol{H}_{\mathrm{EG}}$ by cyclically shifting the vector $\boldsymbol{v}_{\mathcal{L}}$ for $2^{2s} - 2$ times. The resulting circulant matrix $\boldsymbol{H}_{\mathrm{EG}}$ has both row and column weight equal to 2^s, i.e., we have $\rho = \gamma = 2^s$; moreover, two rows or two columns cannot have more than one 1-element in common by construction. Codes of this kind have parameters described in Table II, while codes with different values for s are reported in Table III. In Figure 8 we plot the BER vs $(E_{\mathrm{b}}/N_0)_{\mathrm{dB}}$ for codes with $2 \leq s \leq 5$.

Example

Let us consider the example 17.5 from [9, Chapter 17, p. 865]. We take $m = 2$ and $s = 2$, i.e., we consider the Euclidean geometry EG($2, 2^2$) over the Galois field GF(2^2), where every line consists of four points. In order to build the parity-check matrix $\boldsymbol{H}_{\mathrm{EG}}$, we need to take an incidence vector of a line $\mathcal{L}$. We choose the set of points $\{\alpha^{14} + \beta\alpha\}$ as the line $\mathcal{L}$, i.e., we

Appendix A (continued)

TABLE II: PARAMETERS OF 2-D TYPE-I CYCLIC $(0, S)$-TH ORDER EG-LDPC CODES

Length	$n = 2^{2s} - 1$
Number of parity bits	$n - k = 3^s - 1$
Dimension	$k = 2^{2s} - 3^s$
Minimum distance	$d_{\min} = 2^s + 1$
Density	$r = \dfrac{2^s}{2^{2s} - 1}$

TABLE III: 2-D TYPE-I CYCLIC $(0, S)$-TH ORDER EG-LDPC CODES

$\boldsymbol{s}$	$\boldsymbol{n}$	$\boldsymbol{k}$	$\boldsymbol{d_{\min}}$	$\boldsymbol{\rho}$	$\boldsymbol{\gamma}$	$\boldsymbol{r}$
2	15	7	5	4	4	0.267
3	63	37	9	8	8	0.127
4	255	175	17	16	16	0.0627
5	1023	781	33	32	32	0.0313
6	4095	3367	65	64	64	0.01563
7	16383	14197	129	128	128	0.007813

take one line passing through α^{14}; moreover, we have $\beta \in \mathrm{GF}(2^2)$, $\beta = \alpha^5$ and α as a primitive element of $\mathrm{GF}(2^{2\times 2})$. We can find that the set of four points is $\{\alpha^7, \alpha^8, \alpha^{10}, \alpha^{14}\}$, therefore the incidence vector associated to it is $\boldsymbol{v}_{\mathcal{L}} = (000000011010001)$; cyclically shifting this vector, it is possible to build a 15×15 square matrix $\boldsymbol{H}_{\mathrm{EG}}$ corresponding to a 2-D type-I cyclic $(0, 2)$-th order EG-LDPC code with rate $R = 0.47$.

Appendix A (continued)

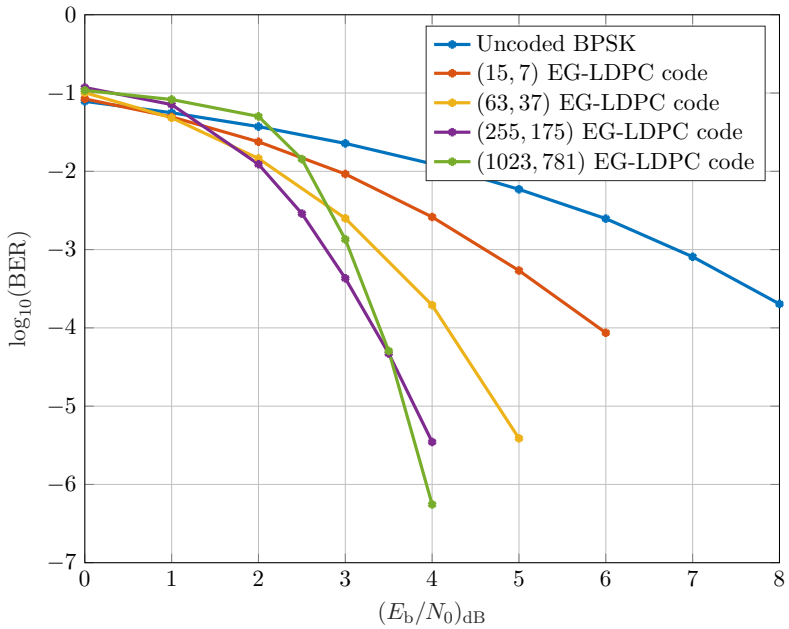


Figure 8. BER vs $(E_b/N_0)_{dB}$ for 2-D type-I cyclic EG-LDPC codes with $2 \leq s \leq 5$, decoded with tanh-based SPA modified as in [10] and with 100 maximum iterations

Appendix B

GEOMETRIC APPROCH FOR THE CONSTRUCTION OF LDPC CODES

Given the normalized variable and check node degree distributions from a node perspective, the construction of irregular LDPC codes can be done systematically by using geometry decomposition and masking techniques. This construction method is particularly convenient because problems related to the random-based construction of LDPC codes, e.g., the presence of cycles of length 4, are avoided.

The presented geometric approach relies on the Euclidean geometry introduced in Appendix A.

B.1 Geometry decomposition and masking technique

Consider the Euclidean geometry $\mathrm{EG}(m, 2^s)$ over $\mathrm{GF}(2^s)$. Let us indicate with $P(m, m-1)$ a parallel bundle of 2^s $(m-1)$-flats in $\mathrm{EG}(m, 2^s)$, where each flat consists of $2^{(m-1)s}$ points and all flats are mutually disjoint, i.e., they have no points in common. Since we have $J = 2^{(m-1)s}(2^{ms}-1)/(2^s-1)$ lines in $\mathrm{EG}(m, 2^s)$ and $J_0 = 2^{(m-2)s}(2^{(m-1)s}-1)/(2^s-1)$ lines in each $(m-1)$-flat in $P(m, m-1)$, let Q be the set of $J_2 = J - 2^s J_0 = 2^{2(m-1)s}$ lines not contained in the parallel bundle $P(m, m-1)$; the lines in Q can be partitioned into $n_0 = 2^{(m-1)s}$ groups, which we refer to as $Q^{(1)}(m,1), Q^{(2)}(m,1), \ldots, Q^{(n_0)}(m,1)$. Each of these groups contains n_0 lines and forms another parallel bundle in $\mathrm{EG}(m, 2^s)$. Every line in Q contains one and only

Appendix B (continued)

one point from each of the 2^s $(m-1)$-flats in $P(m, m-1)$, because all 2^{ms} points in $\mathrm{EG}(m, 2^s)$ are contained in the parallel bundle $P(m, m-1)$; in fact, if a line contained two points from the $(m-1)$-flat in $P(m, m-1)$, then it would be part of the flat itself.

We build the $n_0 \times n$ matrix $\boldsymbol{A}_i$ with rows equal to the incidence vectors of the lines in $Q^{(i)}(m, 1)$, where $n = 2^{ms}$ is the number of points in $\mathrm{EG}(m, 2^s)$. Considering the structure of the incidence vector, the matrix $\boldsymbol{A}_i$ can be seen also as a row of 2^s $n_0 \times n_0$ square matrices $\boldsymbol{A}_i = \begin{bmatrix} \boldsymbol{A}_{i,1} & \boldsymbol{A}_{i,2} & \cdots & \boldsymbol{A}_{i,2^s} \end{bmatrix}$, where each $\boldsymbol{A}_{i,j}$ with $1 \leq j \leq 2^s$ is a *permutation matrix*; the matrix $\boldsymbol{A}_i$ is called the *incidence matrix* of the parallel bundle $Q^{(i)}(m, 1)$. We can build the bigger matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_{n_0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \cdots & \boldsymbol{A}_{1,2^s} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \cdots & \boldsymbol{A}_{2,2^s} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n_0,1} & \boldsymbol{A}_{n_0,2} & \cdots & \boldsymbol{A}_{n_0,2^s} \end{bmatrix}, \tag{B.1}$$

which is an $n_0 \times 2^s$ array of $n_0 \times n_0$ square matrices. This matrix corresponds to the n_0 parallel bundles built with all the lines that are not included in $P(m, m-1)$, thus two rows (or two columns) cannot have more than one 1-element in common by construction, since two lines in $\mathrm{EG}(m, 2^s)$ either are disjoint or intersect in only one point.

From the matrix in Equation B.1, by choosing $1 \leq \gamma \leq n_0$ and $1 \leq \rho \leq 2^s$, we can form the matrix

$$\boldsymbol{H}(\gamma, \rho) = \begin{bmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} & \cdots & \boldsymbol{A}_{1,\rho} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} & \cdots & \boldsymbol{A}_{2,\rho} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{\gamma,1} & \boldsymbol{A}_{\gamma,2} & \cdots & \boldsymbol{A}_{\gamma,\rho} \end{bmatrix}, \tag{B.2}$$

with column and row weights equal to γ and ρ, respectively.

Appendix B (continued)

We saw how to build the matrix $\boldsymbol{H}$ through the decomposition of the *incidence matrix* of a parallel bundle of lines into a row vector of permutation matrices. Now we have to introduce the concept of *masking*. Given the *masking matrix* $\boldsymbol{Z}$ and the *base matrix* $\boldsymbol{H}(\gamma, \rho)$, the *masked matrix* $\boldsymbol{M}$ is defined as

$$\boldsymbol{M} = \boldsymbol{Z} \otimes \boldsymbol{H}(\gamma, \rho) = \begin{bmatrix} z_{1,1}\boldsymbol{A}_{1,1} & z_{1,2}\boldsymbol{A}_{1,2} & \cdots & z_{1,\rho}\boldsymbol{A}_{1,\rho} \\ z_{2,1}\boldsymbol{A}_{2,1} & z_{2,2}\boldsymbol{A}_{2,2} & \cdots & z_{2,\rho}\boldsymbol{A}_{2,\rho} \\ \vdots & \vdots & \ddots & \vdots \\ z_{\gamma,1}\boldsymbol{A}_{\gamma,1} & z_{\gamma,2}\boldsymbol{A}_{\gamma,2} & \cdots & z_{\gamma,\rho}\boldsymbol{A}_{\gamma,\rho} \end{bmatrix}, \tag{B.3}$$

where we have

$$z_{i,j}\boldsymbol{A}_{i,j} = \begin{cases} \boldsymbol{A}_{i,j}, & \text{if } z_{i,j} = 1, \\ \\ \boldsymbol{0}, & \text{if } z_{i,j} = 0, \end{cases} \tag{B.4}$$

and $\boldsymbol{0}$ is the all-zero $n_0 \times n_0$ matrix.

B.2 Irregular Masking

Let us introduce the *normalized degree distributions from a node perspective*, i.e., the polynomials

$$L(x) = \sum_{i=1}^{d_\mathrm{v}} \Lambda_i x^i, \tag{B.5}$$

$$R(x) = \sum_{i=1}^{d_\mathrm{c}} \mathrm{P}_i x^i, \tag{B.6}$$

where the coefficients Λ_i and P_i represent the fraction of variable and check nodes, respectively, with degree i; the terms d_v and d_c denote again the maximum variable and node degrees. In this case we talk about *irregular* LDPC codes since the degrees of variable and check nodes are

Appendix B (continued)

chosen according to some degree distributions. In practice, this means that the parity-check matrix associated to an irregular LDPC code has varying column and row weights. If we have the distribution pair (λ, ρ) in Equation 2.44, it is easy to get $L(x)$ and $R(x)$ as follows:

$$L(x) = \frac{\int_0^x \lambda(z)\,\mathrm{d}z}{\int_0^1 \lambda(z)\,\mathrm{d}z}, \tag{B.7}$$

$$R(x) = \frac{\int_0^x \rho(z)\,\mathrm{d}z}{\int_0^1 \rho(z)\,\mathrm{d}z}. \tag{B.8}$$

Taking advantage of geometry decomposition and masking techniques derived in Section B.1, we can convert degree distributions in a parity-check matrix with desired length N and rate R following these steps:

1. we choose an Euclidean geometry $\mathrm{EG}(m, 2^s)$ such that $2^{ms} \geq N$, $2^s \geq d_\mathrm{c}$ and $2^{(m-1)s} \geq d_\mathrm{v}$;
2. we choose γ and ρ such that $\rho \geq d_\mathrm{c}$, $\gamma \geq d_\mathrm{v}$, $\rho 2^{(m-1)s} \approx N$ and $(\rho - \gamma)/\rho \approx R$;
3. we construct the base matrix $\boldsymbol{H}(\gamma, \rho)$ and the $\gamma \times \rho$ masking matrix $\boldsymbol{Z}$ according to the column and row weight distributions $L(x)$ and $R(x)$;
4. we build the masked matrix $\boldsymbol{M} = \boldsymbol{Z} \otimes \boldsymbol{H}(\gamma, \rho)$, whose null space gives the desired irregular LDPC code.

As already pointed out, this algebraic construction method yields a parity-check matrix free of cycles of length 4, since by construction the base matrix $\boldsymbol{H}(\gamma, \rho)$ is free of cycles of length 4; moreover, it avoids computer search for the construction of a large random graph, while it is much easier to construct the smaller masking matrix $\boldsymbol{Z}$ with given column and row weight distributions.

Appendix B (continued)

Some precautions must be taken in order to satisfy certain conditions. As already said, the constructed parity-check matrix is free of cycles of length 4 by construction; however, we want to cope with the large fraction of degree-2 variable nodes since they induce a high error floor. We can solve this problem following these design rules:

1. make the degree-2 variable nodes cycle-free;
2. make the degree-2 variable nodes corresponding to the parity-check bits.

Rule 1 is achieved if we put the code in systematic form; rule 2 is obtained only when the number of degree-2 variable nodes does not exceed the number of check nodes. Both cases are dealt in the following example.

Example

Let us consider the degree distributions for the AWGN-only case in Table I

$$\lambda(x) = 0.1907x + 0.0963x^2 + 0.1126x^3 + 0.1095x^4 + 0.4904x^{29}, \tag{B.9}$$

$$\rho(x) = 0.5193x^9 + 0.4807x^{10}. \tag{B.10}$$

We want to design an irregular LDPC code with desired length $N_{\mathrm{d}} = 4000$ and rate close to the desired design rate $r_{\mathrm{d}}(\lambda, \rho) = 0.5067$. First of all, we convert the pair (λ, ρ) into the pair (L, R) of normalized degree distributions from a node perspective

$$L(x) = 0.4918x^2 + 0.1656x^3 + 0.1452x^4 + 0.1130x^5 + 0.0844x^{30}, \tag{B.11}$$

$$R(x) = 0.5430x^{10} + 0.4570x^{11}, \tag{B.12}$$

Appendix B (continued)

by means of Equation B.7 and Equation B.8; then, we construct the base matrix $\boldsymbol{H}_{\mathrm{EG}}(\gamma, \rho)$ by considering the Euclidean geometry $\mathrm{EG}(2, 2^6)$ over $\mathrm{GF}(2^6)$. Since we have $d_{\mathrm{v}} = 30$ and $d_{\mathrm{c}} = 11$, we choose $\rho = 63 \geq d_{\mathrm{c}}$ and $\gamma = 32 \geq d_{\mathrm{v}}$, such that the rate is $R = 0.4921$ and the length of the code is $N = \rho \times 2^6 = 4032$; both R and N are close to the desired values. Consequently, the base matrix $\boldsymbol{H}_{\mathrm{EG}}(\gamma, \rho)$ is a 32×63 array of 63×63 permutation matrices. Since the number of degree-2 variable nodes $\Lambda_2 \times 4032 = 1983$ is smaller than the number of check nodes $\gamma \times 2^6 = 2048$, we do not have to apply any modification to the degree distribution in Equation B.11 and degree-2 variable nodes can be made cycle-free, satisfying in this way design rule 1; differently, we should convert some variable nodes of degree-2 to degree-3.

Now, we have to construct the masking matrix $\boldsymbol{Z}$ with column and row weight distributions from $L(x)$ and $R(x)$ degree distributions. Note that desired distributions reported in Table IV are obtained rearranging a little bit the number of columns (rows) for a given column (row) weight such that the total number of columns (rows) is equal to ρ (γ); in particular, we consider $\gamma - 1$ as the maximum number of columns with weight 2 in order to make the degree-2 variable nodes cycle-free. We then construct matrix $\boldsymbol{Z}$ in the following way

$$\boldsymbol{Z} = \left[\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & \cdots & 0 & \\ 1 & 1 & 0 & 0 & \cdots & 0 & \\ 0 & 1 & 1 & 0 & \cdots & 0 & \\ 0 & 0 & 1 & 1 & \cdots & 0 & \text{other columns} \\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & \\ 0 & 0 & 0 & 0 & \cdots & 1 & \end{array}\right], \tag{B.13}$$

Appendix B (continued)

so that 2-weight columns are put into parity-check positions and are cycle-free thanks to the applied downward shifting.

TABLE IV: DESIRED COLUMN AND ROW WEIGHT DISTRIBUTIONS FOR THE MASKING MATRIX $\boldsymbol{Z}$

Column weight distribution		**Row weight distribution**	
Column weight	No. of columns	Row weight	No. of rows
2	31	10	17
3	10	11	15
4	9		
5	7		
30	6		

Appendix B (continued)

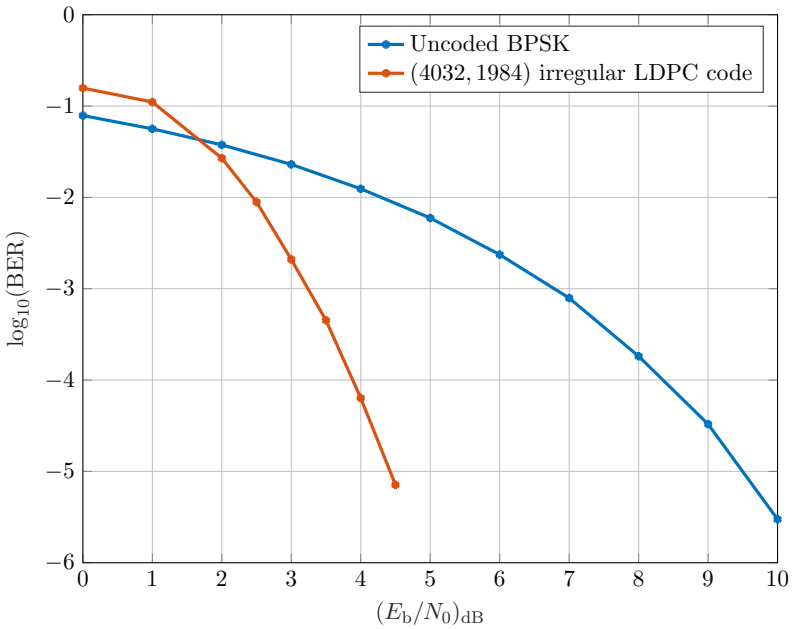


Figure 9. BER vs $(E_b/N_0)_{dB}$ for BPSK modulation in the AWGN-only channel with irregular $(4032, 1984)$ LDPC code and 5 maximum iterations

CITED LITERATURE

1. Shahi, S., Tuninetti, D., and Devroye, N.: On the capacity of the AWGN channel with additive radar interference. In 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pages 902–907, Sep. 2016.

2. Nartasilpa, N., Salim, A., Tuninetti, D., and Devroye, N.: Communications system performance and design in the presence of radar interference. IEEE Transactions on Communications, 66(9):4170–4185, Sep. 2018.

3. Gallager, R. G.: Low-density parity-check codes, 1963.

4. Richardson, T. J., Shokrollahi, M. A., and Urbanke, R. L.: Design of capacity-approaching irregular low-density parity-check codes. IEEE Transactions on Information Theory, 47(2):619–637, Feb 2001.

5. Chung, S.-Y., Forney, G. D., Richardson, T. J., and Urbanke, R.: On the design of low-density parity-check codes within 0.0045 db of the shannon limit. IEEE Communications Letters, 5(2):58–60, Feb 2001.

6. Chung, S.-Y., Richardson, T. J., and Urbanke, R. L.: Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation. IEEE Transactions on Information Theory, 47(2):657–670, Feb 2001.

7. Richardson, T. and Urbanke, R.: Modern Coding Theory. Cambridge University Press, 2008.

8. ten Brink, S.: Convergence of iterative decoding. Electronics Letters, 35(10):806–808, May 1999.

9. Lin, S. and Costello, D. J.: Error control coding: fundamentals and applications. Upper Saddle River, NJ, Pearson/Prentice Hall, 2004.

10. Papaharalabos, S., Sweeney, P., Evans, B. G., Mathiopoulos, P. T., Albertazzi, G., Vanelli-Coralli, A., and Corazza, G. E.: Modified sum-product algorithms for decoding low-density parity-check codes. IET Communications, 1(3):294–300, June 2007.

CITED LITERATURE (continued)

11. Amraoui, A.: Asymptotic and finite-length optimization of LDPC codes. Doctoral dissertation, EPFL, Lausanne, 2006.

12. Chung, S.-Y.: On the Construction of Some Capacity-approaching Coding Schemes. Doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000. AAI0802707.

13. Xu, J., Chen, L., Djurdjevic, I., Lin, S., and Abdel-Ghaffar, K.: Construction of regular and irregular ldpc codes: Geometry decomposition and masking. IEEE Transactions on Information Theory, 53(1):121–134, Jan 2007.

14. MacKay, D. J. C.: Good error-correcting codes based on very sparse matrices. IEEE Transactions on Information Theory, 45(2):399–431, March 1999.

15. Kou, Y., Lin, S., and Fossorier, M. P. C.: Low-density parity-check codes based on finite geometries: a rediscovery and new results. IEEE Transactions on Information Theory, 47(7):2711–2736, Nov 2001.

16. Kschischang, F. R., Frey, B. J., and Loeliger, H. .: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory, 47(2):498–519, Feb 2001.

17. MacKay, D. J. C.: Information Theory, Inference & Learning Algorithms. New York, NY, USA, Cambridge University Press, 2002.

18. Brunero, F., Tuninetti, D., and Devroye, N.: On code design for wireless channels with additive radar interference. In 2019 IEEE Information Theory Workshop (ITW) (IEEE ITW 2019), Visby, Sweden, August 2019.

19. Declercq, D., Fossorier, M., and Biglieri, E.: Channel Coding: Theory, Algorithms, and Applications Academic Press Library in Mobile and Wireless Communications. Orlando, FL, USA, Academic Press, Inc., 1st edition, 2014.

VITA

NAME	Federico Brunero
EDUCATION	Master's Degree in Communications and Computer Networks Engineering, Politecnico di Torino, July 2019, Italy
	Master of Science in Electrical and Computer Engineering, University of Illinois at Chicago, May 2019, USA
	Bachelor's Degree in Telecommunications Engineering, Politecnico di Torino, July 2017, Italy
LANGUAGE SKILLS	
Italian	Native speaker
English	Full working proficiency
	2017 – IELTS examination (7.0)
	A.Y. 2018/2019 One year of study abroad in Chicago, Illinois, USA
	A.Y. 2017/2018 Lessons and exams attended exclusively in English at the Politecnico di Torino, Italy
SCHOLARSHIPS	
Fall 2018	Italian scholarship for final project (thesis) at UIC
Fall 2018	Italian scholarship for students of the TOP-UIC project
WORK EXPERIENCE	
10/17 – 07/18	MATLAB Student Ambassador, The MathWorks srl, Torino, Italy
	Managed a community of students at Politecnico di Torino through a Facebook group; organized seminars/events about the MATLAB software and its toolboxes; organized special events regarding MATLAB role in image processing applications and linear algebra field.
03/17 – 06/17	Curriculum internship, VEM Solutions S.p.A., Venaria Reale, Italy
	Analyzed and interpreted data from accelerometers, with implementation of an instrument of data extraction and processing in `C#`; exploited digital filters for the statistical analysis of data extracted with the developed application.