Physical-layer Network Coding:

Design of Constellations over Rings.

Thesis:
Degree in Mathematics.

Author: I. C. Zarza.
Advisor: M. A. Vázquez.
Tutor: J. M. Mondelo.

Bellaterra, 2013.

Universitat Autònoma de Barcelona

UAB

Projecte Fi de Carrera.

Matemàtiques.

Physical-layer Network Coding:
Design of Constellations over Rings.

Irene de Zarza i Cubero.

Directora: María Ángeles Vázquez i Castro.

Departament de Telecomunicació i Enginyeria de Sistemes.

Escola Tècnica Superior d'Enginyeria (ETSE).

Tutor: Josep Maria Mondelo i González.

Departament de Matemàtiques.

Facultat de Ciències.

Universitat Autònoma de Barcelona (UAB).

Bellaterra, 2013.

El sotasignant, VÁZQUEZ I CASTRO María Ángeles, Professora de l'Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat Autònoma de Barcelona (UAB),

Certifica:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumna DE ZARZA I CUBERO Irene.

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, 2013.

Signatura: Vázquez.

Physical-layer Network Coding: Design of Constellations over Rings.

I. C. Zarza.

Bellaterra, 2013.

Contents

List of Figures

List of Tables

1 Rationale and Objectives

The framework of the work of this project is the recently proposed scheme of communication known as Physical-layer Network Coding. A fundamental issue of the scheme is the geometrical properties of the set of transmission symbols. Such set is known in the field as signal constellation and is obtained making use of algebraic properties of commutative rings.

We have focused on designing constellations over rings of EISENSTEIN Integers and GAUSSIAN Integers in a particular system of communications of Physical-layer Network Coding.

A survey of the mathematical tools needed is presented as well as a description of the system model under study. Moreover, we will explain the performance metrics needed to do a well-round analysis and proceed to propose a methodology to design constellations.

MATLAB and PYTHON programming languages will be used throughout the project to link theory with practice.

2 Introduction

A brief explanation about the main concepts related to this project follows. We will introduce a basic system of communications and then proceed to explain what actually is Physical-layer Network Coding, as well as present what is a constellation and why we focus on its design.

First we are going to introduce a basic scheme of communications based on three parts: Transmitter, Channel and Receiver.

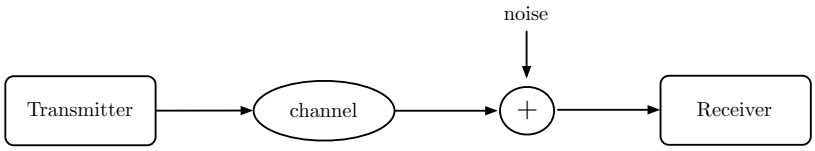


Figure 1: System of Communications.

The process of communication involves the transmission of information from one point to another through a succession of processes. At the transmitter, the message signal (voice, music, picture, or computer data) is described by a set of symbols (signal constellation), in a form suitable for transmission over a physical medium of interest. These symbols are transmitted through the channel to the desired destination. Noise due to propagation and interferences is added in the process. Finally, at the receiver, the reconstruction of the original signal is done.

For the sake of simplicity, in this scheme of communication we just consider a transmitter and a receiver. However, intermediate nodes can be added. These nodes would originally have the only function of forwarding the received packets.

2.1 Introduction to Physical-layer Network Coding

The key idea of network coding was first proposed by Ahlswede et al. [2000], who showed that, by allowing intermediate nodes to combine packets before forwarding them, a maximum information flow can be achieved in a network. An overview of Secure Network Coding can be found in Curtó et al. [2012].

Network coding was first proposed ignoring the underlying physical nature of the channels of communication. More recently, the principles of network coding have been applied at the Physical-layer, by exploiting the natural superposition of electromagnetic waves that occurs in wireless communications.

It is a simple fact in physics that, when multiple electromagnetic waves come together within the same physical space, they add. This mixing of electromagnetic waves is a form of network coding on itself, performed by nature.

This superposition is generally regarded as an obstacle to reliable communication, where the recovery of individual signals is required. For example, in Wi-Fi networks, when multiple nodes transmit together, packet collisions occur, and none of the packets can be received correctly.

Physical-layer Network Coding (PNC) (first proposed by Zhang et al. [2006]) was an attempt to turn the situation around. By exploiting the operation of network coding performed by nature, the “interference” could be put to good use and have a positive effect, enhancing efficiency in communication.

We can see a PNC scheme (Curtó and Vázquez [2013]; Gupta and Vázquez [2012]) in the next figure.

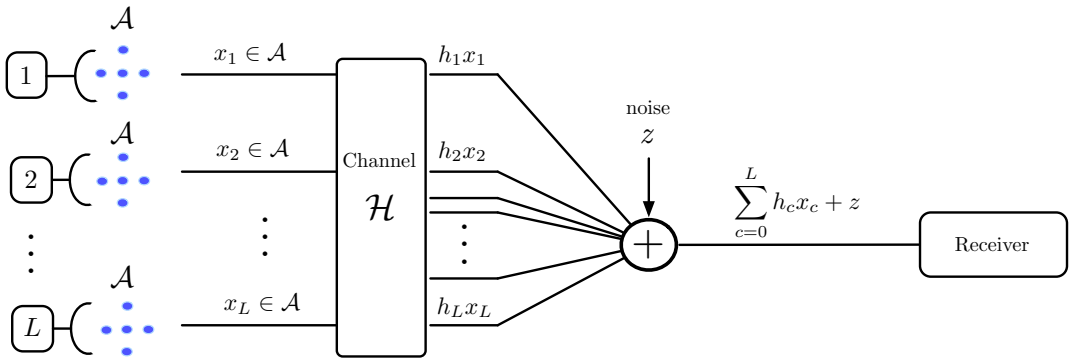


Figure 2: PNC Communication System.

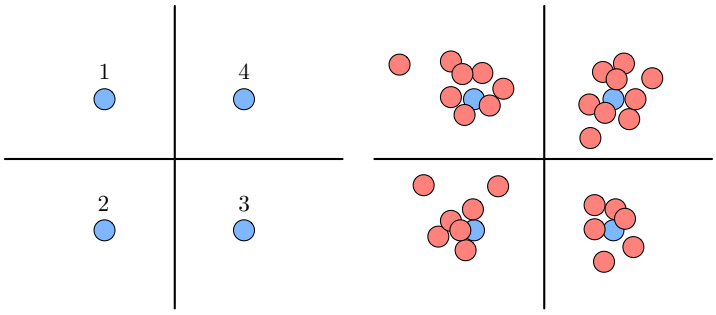
Each one of the L transmitting antennas sends a symbol from the set $\mathcal{A}$ which is the set of all possible transmitted symbols, known as signal constellation. Each one of these symbols travels over the channel of communication, which will be considered constant throughout the communication of a symbol, and can be understood as the product by a constant coefficient. We model the electromagnetic mixing property described as the sum of each component. Moreover, we add a noise z to model the random errors that appear in the process of communication. Finally, the receiver recovers the original symbols after proper processing.

2.2 The Role of a Signal Constellation in a System of Communications

A signal constellation is a set of points in the complex plane used to describe all the possible symbols used by a system to transmit data. It is an aid to design better systems of communications. They help us design a system of transmission that is less prone to errors and can possibly recover from problems of transmission.

Between the transmitter and the receiver, signals can be corrupted. A signal's original output power can be reduced or attenuated by the environment, or the signal can be shifted out of phase. The signal can become so corrupted that it becomes unrecognizable. Designing a constellation that spreads the symbols apart in such a way that they are not easily confused for one another is just part of the improvement in a system of communications.

Figure 3a is a signal constellation with four symbols. The blue dots represent the points of the constellation. Figure 3b shows what actually happens when the signal is transmitted.



(a) Signal Constellation (b) Points at the Receiver

Figure 3: Points of Transmission and Reception.

The red dots represent the value actually received. With this arrangement of the constellation, it is less likely that distortion or deterioration of the signal will result in an error during the transmission. Furthermore, it is possible for the receiver to detect errors and even correct them, if the scheme of transmission is designed well enough. The red dots around the blue dot labeled 1 are automatically corrected to be the 1 value, because that is the symbol they most closely resemble. When a constellation is designed properly, it is less possible for one signal value to be confused with another.

The aim of this project is to design constellations in such a way that the geometry between the symbols induces a good performance in the system. The performance of the constellation is measured using the shape of the decision regions and the analysis of the probability of error, which will be studied in detail throughout the dissertation.

3 Problem Statement and Mathematical Preliminaries

3.1 Problem Statement

This project aims to propose constellations for a PNC scheme (Curtó and Vázquez [2013]; Gupta and Vázquez [2012]; Nazer and Gastpar [2011]). Performance metrics will be used to do the theoretical performance analysis comparing the proposed constellation with the most common constellations used nowadays.

A constellation is a set of points $\mathcal{A} = \{x_k\}$ in the complex plane. They correspond to the symbols that will be transmitted in a system of communications. In designing constellations, we are interested in fixing a certain geometry between these points in order to improve the performance of the overall system. In order to do that, we will use modular arithmetic to define a set of mappings that will determine the points with an induced geometry of the proposed constellations.

3.2 Mathematical Preliminaries

In this work we are going to focus on the design of constellations over two different rings: GAUSSIAN Integers and EISENSTEIN Integers. Next, the theory needed will be introduced.

3.2.1 The Ring $\mathbb{Z}[i]$

Most of the material in this section is based on Huber [1994b], Stillwell [2003] and Conrad [2013].

GAUSSIAN Integers Asif et al. [2012]; Conrad [2013] are a subset of complex numbers which have integers as real and imaginary parts,

$$\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z}\}.$$

In $\mathbb{Z}$, magnitude is measured using the absolute value. In $\mathbb{Z}[i]$, we use the norm defined next.

Definition 3.1. *For $\alpha = a + bi \in \mathbb{Z}[i]$, its norm is the product*

$$N(\alpha) = \alpha\alpha^* = (a + bi)(a - bi) = a^2 + b^2.$$

The reason to deal with norms in $\mathbb{Z}[i]$ instead of absolute values is that norms are integers rather than square roots.

We are going to prove one interesting property of the norm. It is a general property for both rings: GAUSSIAN Integers and EISENSTEIN Integers.

Theorem 3.1. *The norm is multiplicative, that is to say, for α and β in $\mathbb{Z}[*]$, $N(\alpha\beta) = N(\alpha)N(\beta)$.*

Proof. Using the norm definition $N(\alpha\beta) = (\alpha\beta)(\alpha\beta)^* = \alpha\beta\alpha^*\beta^*$. Since we are in a commutative ring, $\mathbb{Z}[*]$, we can rewrite the expression as $N(\alpha\beta) = (\alpha\alpha^*)(\beta\beta^*) = N(\alpha)N(\beta)$.

□

The only GAUSSIAN Integers which are invertible in $\mathbb{Z}[i]$ are ± 1 and $\pm i$. Invertible elements are called units.

3.2.1.1 Theorem of Division

The following theorem provides the analog result in $\mathbb{Z}[i]$ of the well-known theorem of division with remainder in $\mathbb{Z}$.

Theorem 3.2. ***(Theorem of Division).*** *For $\alpha, \beta \in \mathbb{Z}[i]$ with $\beta \neq 0$, there are $\gamma, \rho \in \mathbb{Z}[i]$ such that $\alpha = \beta\gamma + \rho$ where $N(\rho) < N(\beta)$.*

Proof. Let $\alpha, \beta \in \mathbb{Z}[i]$ with $\beta \neq 0$. Then $\alpha/\beta \in \mathbb{C}$, so that $\alpha/\beta = u + iv$ for some real numbers u and v. Let a be an integer which is as close as possible to u. Then $|u - a| \leq 1/2$ so that $(u - a)^2 \leq 1/4$. Similarly, let b be an integer which is as close as possible to v, so that $(v - b)^2 \leq 1/4$. Set $\gamma = a + ib$. Then $\gamma \in \mathbb{Z}[i]$. Set $\rho = \alpha - \gamma\beta$. Then $\rho \in \mathbb{Z}[i]$ because $\alpha, \gamma, \beta \in \mathbb{Z}[i]$.

It remains to prove that $N(\rho) < N(\beta)$. Note that $\beta \neq 0$, then $N(\beta) \neq 0$. By Theorem 3.1 we can assert that $N(\rho) = N((\rho/\beta)\beta) = N(\rho/\beta)N(\beta)$. So that $N(\rho) < N(\beta)$ if and only if $N(\rho/\beta) < 1$. We know that $\rho/\beta = (\alpha - \gamma\beta)/\beta = \alpha/\beta - \gamma = (u + iv) - (a + ib) = (u - a) + i(v - b)$, so that $N(\rho/\beta) = (u - a)^2 + (v - b)^2 \leq 1/4 + 1/4 = 1/2 < 1$. Therefore $\alpha = \gamma\beta + \rho$ with $N(\rho) < N(\beta)$.

□

The numbers γ and ρ are the quotient and remainder, and the remainder is bounded in size (according to its norm) by the size of the divisor β.

We note that there is a subtlety in trying to calculate γ and ρ. This is best understood by working through an example.

Example 3.1. *Let $\alpha = 27 - 23i$ and $\beta = 8 + i$. The norm of β is 65. We want to write $\alpha = \beta\gamma + \rho$ where $N(\rho) < 65$. The idea is to consider the ratio α/β and rationalize the denominator*

$$\frac{\alpha}{\beta} = \frac{\alpha\beta^*}{\beta\beta^*} = \frac{(27 - 23i)(8 - i)}{65} = \frac{193 - 211i}{65}.$$

Since $193/65 = 2.969\ldots$ and $-211/65 = -3.246\ldots$ we replace each fraction with its closest integer from the left (as in the theorem of division in $\mathbb{Z}$) and try $\gamma = 2 - 4i$. However:

$$\alpha - \beta(2 - 4i) = 7 + 7i$$

and using $\rho = 7 + 7i$ is a bad idea: $N(7 + 7i) = 98$ is larger than $N(\beta) = 65$. The usefulness of a theorem of division is the smaller remainder. Therefore our choice of γ and ρ is not desirable. This is the subtlety referred to before we started our example.

To correct our approach, we have to think more carefully about the way we replace $193/65 = 2.969\ldots$ and $-211/65 = -3.246\ldots$ with nearby integers. Let's use the closest integer (as in the modified theorem of division in $\mathbb{Z}$) rather than the closest integer from the left and try $\gamma = 3 - 3i$. Then

$$\alpha - \beta(3 - 3i) = -2i$$

and $-2i$ has norm less than $N(\beta) = 65$. So we use $\gamma = 3 - 3i$ and $\rho = -2i$.

Formally we can note the previous rounding operation in $\mathbb{Z}[i]$ as follows:

Definition 3.2. ***(Rounding of GAUSSIAN Integers)*** *$[a + ib] = [a] + i[b]$ where $[\cdot]$ denotes rounding to the closest integer.*

There is one interesting difference between the theorem of division in $\mathbb{Z}[i]$ and the usual theorem of division in $\mathbb{Z}$ (where the rounding is done to the closest integer from the left): the quotient and remainder are not unique in $\mathbb{Z}[i]$.

Example 3.2. *Now, we give an example where the algorithm of division allows for two different outcomes. Let $\alpha = 1 + 8i$ and $\beta = 2 - 4i$. Then*

$$\frac{\alpha}{\beta} = \frac{\alpha\beta^*}{N(\beta)} = \frac{-30i + 20i}{20} = -\frac{3}{2} + i.$$

Since $-3/2$ lies right in the middle between -2 and -1, we can use $\gamma = -1 + i$ or $\gamma = -2 + i$. Using the first choice, we obtain

$$\alpha = \beta(-1 + i) - 1 + 2i.$$

Using the second choice,

$$\alpha = \beta(-2 + i) + 1 - 2i.$$

However, this lack of uniqueness in the quotient and remainder does not seriously limit the usefulness of division in $\mathbb{Z}[i]$. It is irrelevant for many important applications (such as Euclid's Algorithm).

3.2.1.2 $\mathbb{Z}[i]$ as a Principal Ideal Domain

We will introduce two definitions to make the reading of this subsection easier.

Definition 3.3. ***(Integral Domain)****. An Integral Domain is a commutative ring without zero divisors. In other words, if x and y are non-zero elements of the ring, then $xy \neq 0$.*

Definition 3.4. ***(PID)****. A Principal Ideal Domain is an integral domain in which every ideal is principal, equivalently, every ideal can be generated by a single element.*

The structure of Principal Ideal Domain (PID) in the ring of GAUSSIAN Integers provides us an interesting framework in order to design the mappings of the constellation in Sections 5.4 and 5.6.

That is because, in the design we are looking for structures of the form $R/\alpha R$ where R is a ring and αR an ideal, in such a way that $R/\alpha R$ forms a field (a commutative ring in which every non-zero element has an inverse.). When R is a PID, the theory that we will describe next allows to obtain it.

Definition 3.5. *An ideal K is a maximal ideal of a ring R if there are no other ideals contained between K and R.*

The first theorem provides us a way of obtaining this structure using maximal ideals:

Theorem 3.3. *Let R be a commutative ring with unity. Let L be an ideal of R. Then, L is a maximal ideal if and only if the quotient ring R/L is a field.*

The next proposition explains us how a maximal ideal is defined in a PID:

Proposition 3.1. *Let R be a PID and $a \in R$, $a \neq 0$. Then aR is maximal if and only if a is irreducible.*

Definition 3.6. ***(Irreducible)**. A GAUSSIAN Integer is called irreducible if its only divisors are units.*

Therefore, from the two results above we can deduce a straightforward way in order to obtain the desired structure in a PID:

Corollary 3.1. *If R is a PID and $a \in R$ is irreducible, then R/aR is a field.*

Example 3.3. *For a PID $R = \mathbb{Z}[i]$, $a = 2 + i$ is irreducible. Hence, $\mathbb{Z}[i]/(2+i)\mathbb{Z}[i]$ is a field.*

Now, we are going to do some work in the proof that $\mathbb{Z}[i]$ is a PID.

First we are going to introduce the concept of Euclidean domain defined next.

Definition 3.7. ***(Euclidean Domain)**. An integral domain R is said to be an Euclidean domain if there is a function N from the set of non-zero elements of R to the set of non-negative integers such that*

(1) (Theorem of Division) given $a, b \in R$ with $b \neq 0$ there exist $q, r \in R$ such that $a = bq + r$ where $N(r) < N(b)$, and

(2) for all non-zero elements a and b of R we have $N(a) \leq N(ab)$.

Theorem 3.4. *Euclidean domains are Principal Ideal Domains (PIDs).*

Proof. Let C be any non-zero ideal of the Euclidean domain R, and let $d \in C$ be a non-zero element of minimum norm.

We claim $(d) = C$. Certainly, $(d) \subseteq C$.

Let $a \in C$. By the Theorem of Division 3.2, $a = qd + r$, with $r = 0$ or $N(r) < N(d)$. Since $a - qd = r \in C$, by minimality of $N(d)$ we see $r = 0$ and $a = qd \in (d)$.

□

Theorem 3.5. *$\mathbb{Z}[i]$ is a Principal Ideal Domain (PID).*

Proof. We have to prove that $\mathbb{Z}[i]$ is an Euclidean domain. In order to do that first we are going to prove that it is an integral domain:

$\mathbb{Z}[i] \subseteq \mathbb{C}$ which is a field $\Rightarrow$ $\mathbb{C}$ has no zero divisors $\Rightarrow$ $\mathbb{Z}[i]$ is an integral domain.

We have proved the Theorem of Division 3.2 for the ring $\mathbb{Z}[i]$ (part (1) of the definition of Euclidean domain). Using that the norm is multiplicative (Theorem 3.1) we can prove part (2) of the definition. We have $N(a) \leq N(ab) = N(a)N(b)$ so $1 \leq N(b)$ it is true because b is a positive integer. Therefore, $\mathbb{Z}[i]$ is an Euclidean domain. Theorem 3.4 ends the proof.

□

In a PID, every prime is irreducible. Therefore for every prime p in a PID R we have that R/pR is a field. Now, the natural question is asking which are the primes in $\mathbb{Z}[i]$. We will answer it in the next subsection.

3.2.1.3 Primes in $\mathbb{Z}[i]$

The classification of the factorization of prime integers in the ring of GAUSSIAN Integers can be summarized in the next theorem:

Theorem 3.6. *Let p be a prime in $\mathbb{Z}^+$. The factorization of p in $\mathbb{Z}[i]$ is determined by $p \bmod 4$:*

- $2 = (1+i)(1-i) = -i(1+i)^2$.
- *if $p \equiv 1 \bmod 4$ then $p = \pi\pi^*$ is a product of two conjugate primes π, π^* in $\mathbb{Z}[i]$ which are not unit multiples.*
- *if $p = 3 \bmod 4$ then p stays prime in $\mathbb{Z}[i]$.*

3.2.2 The Ring $\mathbb{Z}[w]$

This section is based on Huber [1994a].

EISENSTEIN Integers are a subset of complex numbers which have integer linear combination of unity and the primitive cube root of 1:

$$\mathbb{Z}[w] = \{a + bw : a, b \in \mathbb{Z}\}$$

with w is a primitive cube root of 1:

$$w = e^{2\pi i/3} = \cos\left(\frac{2\pi}{3}\right) + i\sin\left(\frac{2\pi}{3}\right) = \frac{1}{2}\left(-1 + i\sqrt{3}\right).$$

Magnitude is measured using the norm of $\mathbb{Z}[i]$, that has a different expression in terms of a and b:

Definition 3.8. *For $\gamma = a + bw \in \mathbb{Z}[w]$, its norm is defined as*

$$N(\gamma) = \gamma\gamma^* = a^2 - ab + b^2.$$

Note that if $\gamma = a_1 + b_1 w \in \mathbb{Z}[w]$, then $\gamma^* = a_1 + b_1 w^2$ is the conjugate of γ.

The only EISENSTEIN Integers which are invertible (unities) are the elements of $\mathbb{Z}[w]$ which have norm 1: $\pm 1, \pm w, \pm w^2$.

3.2.2.1 Theorem of Division

One reason we will be able to transfer a lot of results from $\mathbb{Z}[i]$ to $\mathbb{Z}[w]$ is the following analogue theorem of division with remainder:

Theorem 3.7. ***(Theorem of Division).*** *For $\gamma, k \in \mathbb{Z}[w]$ with $k \neq 0$, there are $\gamma_1, \gamma_2 \in \mathbb{Z}[w]$ such that $\gamma = k\gamma_1 + \gamma_2$ where $N(\gamma_2) < N(k)$.*

Proof. Let $\gamma, k \in \mathbb{Z}[w]$ with $k \neq 0$. Then $\gamma/k \in \mathbb{C}$, so that $\gamma/k = u + wv$ for some real numbers u and v. Let a be an integer which is as close as possible to u. Then $|u - a| \leq 1/2$ so that $(u-a)^2 \leq 1/4$. Similarly, let b be an integer which is as close as possible to v, so that $(v-b)^2 \leq 1/4$. Set $\gamma_1 = a + wb$. Then $\gamma_1 \in \mathbb{Z}[w]$. Set

$\gamma_2 = \gamma - \gamma_1 k$. Then $\gamma_2 \in \mathbb{Z}[w]$ because $\gamma, \gamma_1, k \in \mathbb{Z}[w]$.

It remains to prove that $N(\gamma_2) < N(k)$. Note that $k \neq 0$, then $N(k) \neq 0$. By Theorem 3.1 we can assert that $N(\gamma_2) = N((\gamma_2/k)k) = N(\gamma_2/k)N(k)$. So that $N(\gamma_2) < N(k)$ if and only if $N(\gamma_2/k) < 1$. We know that $\gamma_2/k = (\gamma - \gamma_1 k)/k = \gamma/k - \gamma_1 = (u + wv) - (a + wb) = (u - a) + w(v - b)$, so that $N(\gamma_2/k) = (u-a)^2 + (v-b)^2 - (u-a)(v-b) \leq 1/4 + 1/4 - 1/4 = 1/4 < 1$. Therefore $\gamma = \gamma_1 k + \gamma_2$ with $N(\gamma_2) < N(k)$. □

The numbers γ_1 and γ_2 are the quotient and remainder, and the remainder is bounded in size (according to its norm) by the size of the divisor k.

For EISENSTEIN Integers we can define similarly a rounding operation:

Definition 3.9. ***(Rounding of EISENSTEIN Integers)*** *Let $z = x + wy$ with x, y real numbers, then $[z]$ denotes the closest EISENSTEIN Integer to z, that is to say, $[z]$ is the EISENSTEIN Integer which gives the smallest value of $N(z - [z])$.*

3.2.2.2 $\mathbb{Z}[w]$ as a Principal Ideal Domain

Equivalently to Section 3.2.1.2 for GAUSSIAN Integers, we are interested in seeing that $\mathbb{Z}[w]$ is a PID.

Theorem 3.8. *$\mathbb{Z}[w]$ is a Principal Ideal Domain (PID).*

Proof. We have to prove that $\mathbb{Z}[w]$ is an Euclidean domain. In order to do that first we are going to prove that it is an integral domain:

$\mathbb{Z}[w] \subseteq \mathbb{C}$ which is a field $\Rightarrow$ $\mathbb{C}$ has no zero divisors $\Rightarrow$ $\mathbb{Z}[w]$ is an integral domain.

We have proved the Theorem of Division 3.7 for the ring $\mathbb{Z}[w]$ (part (1) of the definition of Euclidean domain). Using that the norm is multiplicative (Theorem 3.1) we can prove part (2) of the definition. We have $N(a) \leq N(ab) = N(a)N(b)$ so $1 \leq N(b)$ it is true because b is a positive integer. Therefore, $\mathbb{Z}[w]$ is an Euclidean domain. Theorem 3.4 ends the proof.

□

3.2.2.3 Primes in $\mathbb{Z}[w]$

There exists an analog theorem for classification of primes in the ring of EISENSTEIN Integers:

Theorem 3.9. *There are three classes of EISENSTEIN primes:*

- $1 - w$ *and its unit multiples.*
- *Numbers of the form $a + wb$ where $b = 0$ and is a prime in $\mathbb{Z}$ congruent with 2 modulo 3. That is to say, if p prime in $\mathbb{Z}$ with $p \equiv 2 \bmod 3$, p stays prime in $\mathbb{Z}[w]$.*
- *Numbers of the form $\pi = a + bw$ or its conjugate $\pi^* = a + bw^2$, where $\pi\pi^* = (a + bw)(a + bw^2) = p$ prime in $\mathbb{Z}$ congruent with 1 modulo 6. Primes of this form can always be written as $p = u^2 + 3v^2$.*

3.2.3 Euclid's Algorithm and BÉZOUT Theorem

Euclid's Algorithm and BÉZOUT Theorem will provide us with the theoretical keys to build the mappings for the designed constellations.

Euclid's Algorithm

We begin by defining greatest common divisors in $\mathbb{Z}[i]$ and $\mathbb{Z}[w]$. We will denote both rings as $\mathbb{Z}[*]$.

Definition 3.10. *For non-zero α and β in $\mathbb{Z}[*]$, a greatest common divisor of α and β is a common divisor with maximal norm.*

This is analogous to the usual definition of greatest common divisor in $\mathbb{Z}$, except that the concept does not refer to a specific number. If r is a greatest common divisor of α and β, so are its unit multiples. Therefore, we can speak of a greatest common divisor, but not the greatest common divisor.

Definition 3.11. *When α and β only have unit factors in common, we call them* relatively prime.

Theorem 3.10. ***(Euclid's Algorithm)**. Let $\alpha, \beta \in \mathbb{Z}[*]$ be non-zero. Apply recursively the theorem of division, starting with this pair, and make the divisor and remainder in one equation the new dividend and divisor in the next one, provided the remainder is not zero:*

$$\begin{aligned} \alpha &= \beta\gamma_1 + \rho_1, \quad N(\rho_1) < N(\beta) \\ \beta &= \rho_1\gamma_2 + \rho_2, \quad N(\rho_2) < N(\rho_1) \\ \rho_1 &= \rho_2\gamma_3 + \rho_3, \quad N(\rho_3) < N(\rho_2) \\ &\dots \end{aligned}$$

The last non-zero remainder is divisible by all common divisors of α and β, and is itself a common divisor, so it is a greatest common divisor of α and β.

Corollary 3.2. *For non-zero α and β in $\mathbb{Z}[*]$, let δ be a greatest common divisor produced by Euclid's Algorithm. Any greatest common divisor of α and β is a unit multiple of δ.*

Now, we are going to see some examples in the ring $\mathbb{Z}[i]$.

Example 3.4. *We compute a greatest common divisor of $\alpha = 32 + 9i$ and $\beta = 4 + 11i$.*

$$\begin{aligned} 32 + 9i &= (4 + 11i)(2 - 2i) + 2 - 5i \\ 4 + 11i &= (2 - 5i)(-2 + i) + 3 - i \\ 2 - 5i &= (3 - i)(1 - i) - i \\ 3 - i &= (-i)(1 + 3i) + 0. \end{aligned}$$

The last non-zero remainder is $-i$ a greatest common divisor, so α and β only have unit factors in common. They are relatively prime.

Example 3.5. *Here is an example where the greatest common divisor is not a unit. Let $\alpha = 11+3i$ and $\beta = 1+8i$. Then*

$$\begin{aligned} 11+3i &= (1+8i)(1-i)+2-4i \\ 1+8i &= (2-4i)(-1+i)-1+2i \\ 2-4i &= (-1+2i)(-2)+0. \end{aligned}$$

So a greatest common divisor of α and β is $-1+2i$.

We could proceed in a different way in the second equation (due to the lack of uniqueness of the theorem of division), and obtain a different non-zero remainder,

$$\begin{aligned} 11+3i &= (1+8i)(1-i)+2-4i \\ 1+8i &= (2-4i)(-2+i)+1-2i \\ 2-4i &= (1-2i)(2)+0. \end{aligned}$$

Therefore $1-2i$ is also a greatest common divisor. Our two different answers are not inconsistent: a greatest common divisor is defined at best only up to a unit multiple anyway, and $-1+2i$ and $1-2i$ are unit multiples of each other: $-1+2i = (-1)(1-2i)$.

BÉZOUT Theorem

In $\mathbb{Z}$, BÉZOUT Theorem says for any non-zero a and b in $\mathbb{Z}$ that $gcd(a,b) = ax+by$ for some x and y in $\mathbb{Z}$ found by back-substitution in Euclid's Algorithm. The same idea works in $\mathbb{Z}[i]$ and $\mathbb{Z}[w]$ and gives us BÉZOUT Theorem there.

Theorem 3.11. ***(BÉZOUT Theorem)*** *Let δ be any greatest common divisor of two non-zero elements of $\mathbb{Z}[*]$, α and β. Then $\delta = \alpha x + \beta y$ for some $x, y \in \mathbb{Z}[*]$.*

Corollary 3.3. *The non-zero elements of $\mathbb{Z}[*]$, α and β, are relatively prime if and only if we can write*

$$1 = \alpha x + \beta y$$

for some $x, y \in \mathbb{Z}[*]$.

Now, we are going to see some examples in the ring $\mathbb{Z}[i]$:

Example 3.6. *We saw in the previous example that $\alpha = 32+9i$ and $\beta = 4+11i$ are relatively prime, since the last non-zero remainder in Euclid's Algorithm is $-i$. We can reverse the calculations in this example to express $-i$ as a $\mathbb{Z}[i]$-combination of α and β:*

$$\begin{aligned} -i &= 2-5i-(3-i)(1-i), \\ &= 2-5i-(\beta-(2-5i)(-2+i))(1-i), \end{aligned}$$

$$
\begin{aligned}
&= (2-5i)(1+(-2+i)(1-i)) - \beta(1-i), \\
&= (2-5i)(3i) - \beta(1-i), \\
&= (\alpha - \beta(2-2i)(3i)) - \beta(1-i), \\
&= \alpha(3i) - \beta(7+5i).
\end{aligned}
$$

To write 1, rather than $-i$, as a combination of α and β, multiply by i:

$$1 = \alpha(-3) + \beta(5-7i).$$

3.2.4 Finite Fields

Now we are going to do an introduction to finite fields. We are interested in working with fields because it will allow us complete the process of communication. We will explain this fact in detail in the design section.

Theorem 3.12. *A finite field or Galois Field (noted by $\mathbb{F}$) is a field that contains a finite number of elements and they are classified as follows:*

- *The number of elements of a finite field is of the form p^n where p is a prime number and n is a positive integer.*
- *For every prime number p and positive integer n, there exists a finite field with p^n elements.*
- *Any two finite fields with the same number of elements are isomorphic.*

3.2.5 Homomorphism

Formally, the canonical projection presented in this subsection would be the first step in the design of the constellation.

In this work we are interested in the design of constellations with a finite number of points so we need to narrow the infinite number of points of the initial ring.

When we do the quotient of a set X we are narrowing the initial set into a subset with the representative elements of each class and that is just we are looking for.

The set of all equivalence classes in X given an equivalence relation $\sim$ is denoted as $X/\sim$ and called the quotient set of X by $\sim$.

So we are interested in building a mapping between the initial ring $\mathbb{Z}[*]$ and a quotient set of it. Note that $\mathbb{Z}[*]$ refers to $\mathbb{Z}[i]$ and $\mathbb{Z}[w]$.

A mapping that takes an element to its equivalence class under a given equivalence relation is known as the **canonical projection**.

In this work the relation will be defined as the congruence relation so we can define the canonical projection as:

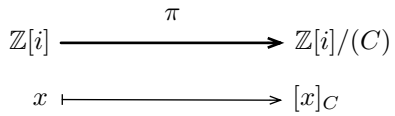


Figure 4: Canonical Projection.

where (C) is an ideal of the ring $\mathbb{Z}[*]$ and $[x]_C$ is the representative of the class x congruent C, that is to say, $[x]_C \equiv x \bmod C \Leftrightarrow [x]_C - x \in C$.

It is easy to observe that π corresponds to the function modulo and in this case it is a surjective mapping, every element in $\mathbb{Z}[*]/(C)$ has a corresponding element in $\mathbb{Z}[*]$.

The number of elements of a quotient ring using these two rings is defined as:

Theorem 3.13. *If $\alpha \neq 0$ in $\mathbb{Z}[i]$ or $\mathbb{Z}[w]$, then $n(\alpha) = N(\alpha)$, where $n(\alpha)$ denotes the number of GAUSSIAN Integers or EISENSTEIN Integers modulo α. That is, the size of $\mathbb{Z}[*]/\alpha\mathbb{Z}[*]$ is $N(\alpha)$.*

There is an analogy with the absolute value on $\mathbb{Z}$, where $\#(\mathbb{Z}/m\mathbb{Z}) = |m|$, with $m \neq 0$ and now $\#(\mathbb{Z}[*]/\alpha\mathbb{Z}[*]) = N(\alpha)$ with $\alpha \neq 0$.

3.2.6 Theorems Needed in the Design

Now we are going to introduce the theorems used in the design.

We have seen that we are interested in finding fields $R/\alpha R$ where R is a PID and αR an ideal. We have introduced these ideas briefly at the beginning of Section 3.2.1.2, however we are going to present the theorems referenced in the design section which follow from the theory already described.

Definition 3.12. *Let R be an ideal. An ideal $P \neq R$ is said to be prime if for all $a, b \in R$, $ab \in P$ implies $a \in P$ or $b \in P$.*

The first theorem tells us a way to find maximal ideals in a PID:

Theorem 3.14. *Let R be a commutative ring with unity. If R is a PID, then every non-zero prime ideal is maximal.*

We have seen in the previous theorem that prime ideals will help us to find maximal ideals in a PID. Now, we can see how to obtain prime ideals in the following theorem:

Theorem 3.15. *If R is an integral domain and $\alpha \in R$ is a non-zero, non-unit element, then (α) is a prime R-ideal if and only if α is a prime in R.*

Finally, from Theorems 3.3, 3.14 and 3.15 we can deduce the next two theorems: the first one is a version for integer ring and the second one is the version for polynomial ring:

Theorem 3.16. *For every prime p in $\mathbb{Z}$, $\mathbb{Z}/p\mathbb{Z}$ forms a field.*

In some steps in the design we will work with polynomial quotient ring in order to obtain the structure of field, in these cases we will use the next theorem:

Theorem 3.17. *For a prime p and a monic irreducible $m(x)$ in $\mathbb{F}_p[x]$ of degree n, the ring $\mathbb{F}_p[x]/(m(x))$ is a field of order p^n.*

4 Performance Metrics and Mostly Used Constellations

4.1 Performance Metrics

We are going to introduce the theoretical concepts used in the analysis of proposed constellations. The performance analysis of constellations will be based on decision regions and the analysis of the probability of error. Finally, we will do the study of two key parameters needed for the computation of the probability of error: the minimum distance and the average number of neighbors.

4.1.1 Decision Regions

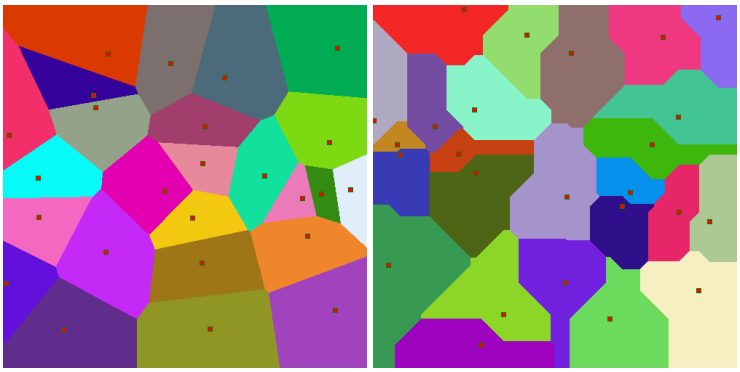
Decision regions are a fundamental concept in understanding the design of constellations. Moreover, they are of utmost importance in the computation of the probability of error. Most of the material of this section is based on Goldsmith [2005] and Madhow [2008].

Let $X = \mathbb{C}$ be the complex space endowed with a distance d and a constellation with M points defined inside it. The decision region for the point of the constellation c_z is the set of all complex numbers that are closer to c_z than to any other point of the signal constellation. Therefore X is partitioned into M decision regions $\mathcal{R}_z$, $1 \leq z \leq M$ one by each point of the constellation. More formally,

$$\mathcal{R}_z = \{x \in X : \quad d(x, P_z) \leq d(x, P_k) \quad \forall z \neq k\}$$

where $d(x, A) = \inf\{d(x, a) | a \in A\}$ denotes the distance between the point x and the subset A and P_z is the set of all points in whose distance to x is not greater than their distance to the other sites P_k, where k is any index different from z.

Decision regions defined in this way for a fixed set of points in the complex space are called VORONOI Regions. VORONOI Regions depend significantly on the metric used. We are going to show two different examples of VORONOI Regions in the complex plane choosing 25 random points with two different distances in order to see how the metric affects the shape of the region.[1]



(a) Euclidean Distance (b) MANHATTAN Distance

Figure 5: Two Examples of VORONOI Regions Using Different Measures.

[1] The code of this two examples can be found in Annex V

In Figure 5a it is shown the VORONOI Regions using the Euclidean distance defined as:

$$d(x_1, x_2) = \sqrt{(\text{Real}(x_1) - \text{Real}(x_2))^2 + (\text{Imaginary}(x_1) - \text{Imaginary}(x_2))^2} \quad \text{where} \quad x_1, x_2 \in \mathbb{C}.$$

In this case every side of a VORONOI Region is a segment of the perpendicular bisector of the line connecting two neighbors.

In Figure 5b it is used the MANHATTAN Distance defined as:

$$d(x_1, x_2) = |\text{Real}(x_1) - \text{Real}(x_2)| + |\text{Imaginary}(x_1) - \text{Imaginary}(x_2)| \quad \text{where} \quad x_1, x_2 \in \mathbb{C}.$$

In this project, we are going to focus on the Euclidean distance.

4.1.2 Probability of Error

Most of the material for this section can be found in Cioffi [2013]; Goldsmith [2005]; Waterhouse [1987]; Wozencraft and Jacobs [1965].

In order to study the probability of error of a given constellation, we are going to assume a AWGN (Additive White Gaussian Noise) channel. This channel adds to the signal $x(t)$ an uncorrelated GAUSSIAN noise $n(t)$ to the output.

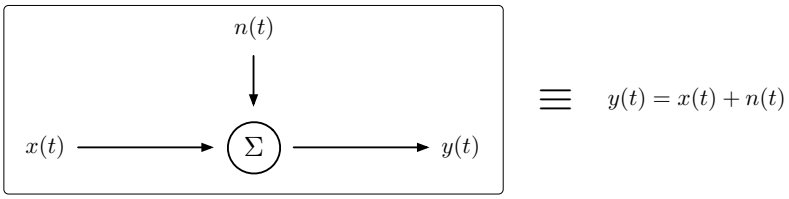


Figure 6: AWGN Channel.

The noise $n(t)$ is a 1 dimensional GAUSSIAN random signal with zero mean, variance σ^2 and probability distribution:

$$p_n(u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}u^2}. \tag{1}$$

From the hypothesis, the AWGN channel is equivalent to a vector channel with output given by

$$\mathbf{y} = \mathbf{x} + \mathbf{n}. \tag{2}$$

We can write the vector analogy of the AWGN explained above:

The noise vector $\mathbf{n}$ is a N dimensional GAUSSIAN random vector with zero mean, equal-variance and uncorrelated components in each dimension. The noise distribution is

$$p_{\mathbf{n}}(\mathbf{u}) = (\pi\mathcal{N}_0)^{-N/2} \cdot e^{\frac{1}{\mathcal{N}_0}||\mathbf{u}||^2} = (2\pi\sigma^2)^{-N/2} \cdot e^{-\frac{1}{2\sigma^2}||\mathbf{u}||^2}. \tag{3}$$

The computation of P_e assumes the inputs $\mathbf{x_c}$ equally likely, that is to say, $p_{\mathbf{x}}(c) = \frac{1}{M} \forall c$. Under this assumption and based on Wozencraft and Jacobs [1965], we can assert that the optimum detector is the ML detector, which has decision rule

$$\hat{m} \Rightarrow m_c \quad \text{if} \quad ||v - x_c||^2 \leq ||v - x_z||^2 \quad \forall z \neq c. \tag{4}$$

ML takes the the point of the constellation which the detected point is nearest to. The probability of error associated with this rule depends on the signal constellation $\mathbf{x}_c$ and the noise variance $\sigma^2 = \frac{N_0}{2}$.

The exact P_e corresponds to the sum of probabilities of having an error when transmitting a given symbol. These probabilities are disjoint and can be expressed as: $P(e \cap c) = P(e|c) \cdot P(c)$, that is to say, having an error from the symbol c. Therefore,

$$\begin{aligned} P_e &= \sum_{c=0}^{M-1} P_{e|c} \cdot P(c), & (5) \\ &= \frac{1}{M} \sum_{c=0}^{M-1} P_{e|c}, & (6) \\ &= 1 - \frac{1}{M} \sum_{c=0}^{M-1} P_{r|c}. & (7) \end{aligned}$$

This may be difficult to compute: be aware that $P_{e|c}$ depends on the geometry of the decision region and therefore it consists on the computation of an integral of the distribution Gaussian on the complementary decision region. So convenient and accurate bounding procedures can approximate P_e.

4.1.2.1 The Union Bound

We are going to propose a P_e approximation that will be useful in our analysis. In order to do that we are going to start studying a first approximation: the Union Bound. Then, we are going to go further and use this first approximation to derive a tighter bound: The Nearest Neighbor Union Bound.

P_e approximations:

1 Union Bound.

2 The Nearest Neighbor Union Bound.

First, we introduce an important metric that is of the utmost importance in the design and performance of the constellation: the minimum distance.

Definition 4.1. *The minimum distance , $d_{\min}$ is defined as the minimum distance between any two points in a*

constellation $\{x_c\}_{c=0,\cdots,M-1}$*:*

$$d_{\min} = \min_{c \neq z} ||x_c - x_z|| \quad \forall c, z. \tag{8}$$

We are going to study a simplified case that will help us in the derivation:

Suppose a system that has two possible points of the constellation in N dimensions with a AWGN channel, as illustrated for $N = 1$ dimension in Figure 7. Then the probability of error for the ML detector is the probability that the component of the noise vector $\mathbf{n}$ along the line connecting the two data symbols is greater than half the distance along this line. In this case, the noisy received vector $\mathbf{y}$ lies in the incorrect decision region, resulting in an error.

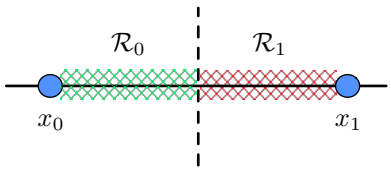


Figure 7: One Dimension Constellation with Two Symbols.

Since the noise is white Gaussian, its projection in any dimension, in particular, the segment of the line connecting the two data symbols, is of variance $\sigma^2 = \frac{N_0}{2}$. This is because AWGN statistics are rotation invariant and since a projection is a particular case of rotation, the statistics remain unchanged.

$$P_e = P\left(\langle n, \phi \rangle \geq \frac{d}{2}\right)$$

where ϕ is a unit norm vector along the line between x_0 and x_1 and $d = ||x_0 - x_1||$. This probability of error is

$$P_e = \int_{\frac{d}{2}}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}u^2} du = \int_{\frac{d}{2\sigma}}^{\infty} e^{-\frac{u^2}{2}} du = Q\left[\frac{d}{2\sigma}\right], \tag{9}$$

if we consider $\sigma^2 = \frac{N_0}{2}$

$$P_e = Q\left[\frac{d}{\sqrt{2N_0}}\right]. \tag{10}$$

This result is useful in the proof of the following theorem.

Theorem 4.1 (Union Bound)**.** *The probability of error for the ML detector on the AWGN channel, with a M-point signal constellation with minimum distance* $d_{\min}$ *is bounded by*

$$P_e \leq (M-1)Q\left[\frac{d_{\min}}{2\sigma}\right]. \tag{11}$$

Proof. We suppose an error ϵ_{cz} occurs when x_c is transmitted and the ML detector chooses $\hat{x} = x_z$. We can express it as:

$$P_{e|c} = P\left(\bigcup_{z=0(z\neq c)}^{M-1} \epsilon_{cz}\right). \tag{12}$$

Since errors are mutually exclusive, the probability of the union is the sum of the probabilities

$$P_{e|c} = \sum_{z=0(z\neq c)}^{M-1} P(\epsilon_{cz}). \tag{13}$$

We bound $P(\epsilon_{cz})$ by the probability of a given point, y, to be closer to x_z than to x_c, that is to say, $P(||y - x_z|| \leq ||y - x_c||) = P'(x_c, x_z)$:

$$P(\epsilon_{cz}) \leq P'(x_c, x_z) \tag{14}$$

and therefore

$$P_{e|c} \leq \sum_{z=0(z\neq c)}^{M-1} P'(x_c, x_z). \tag{15}$$

It is important to note that $P(\epsilon_{cz})$ is the probability that the received vector y lies inside the decision region of the point z. However, $P'(x_c, x_z)$ is the probability that the received vector y lies closer to z than c, and that can be translated as it lies in the half plane closer to z. Therefore, $P'(x_c, x_z)$ includes the region for $P(\epsilon_{cz})$, as it can be seen on the next picture.

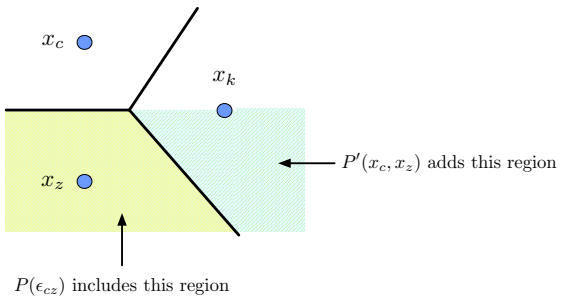


Figure 8: Probability of Error Regions.

Using the result derived earlier in Equation (10)

$$P'(x_c, x_z) = P(||y - x_z|| \leq ||y - x_c||) = Q\left[\frac{||x_c - x_z||}{2\sigma}\right], \tag{16}$$

and if we substitute

$$P_{e|c} \leq \sum_{z=0(z\neq c)}^{M-1} P'(x_c, x_z) = \sum_{z=0(z\neq c)}^{M-1} Q\left[\frac{||x_c - x_z||}{2\sigma}\right]. \tag{17}$$

If we introduce that $Q(x)$ is monotonically decreasing with x, and since $d_{\min} \leq ||x_c - x_z||$

$$Q\left[\frac{||x_c - x_z||}{2\sigma}\right] \leq Q\left[\frac{d_{\min}}{2\sigma}\right]. \tag{18}$$

If we average over all the points of the constellation we get P_e

$$\begin{aligned} P_e &\leq \sum_{c=0}^{M-1} \sum_{z=0(z\neq c)}^{M-1} Q\left[\frac{d_{\min}}{2\sigma}\right] p_x(c), & (19)\\ &= \sum_{c=0}^{M-1} (M-1) Q\left[\frac{d_{\min}}{2\sigma}\right] p_x(c), & (20)\\ &= (M-1) Q\left[\frac{d_{\min}}{2\sigma}\right]. & (21) \end{aligned}$$

□

4.1.2.2 The Nearest Neighbor Union Bound

The factor $(M-1)$ in the original Union Bound is often too large for accurate performance prediction. The Nearest Neighbor Union Bound gives a tighter bound on the probability of error for a signal constellation by lowering this factor of the Q-function.

We introduce a new performance metric:

Definition 4.2 (Average Number of Nearest Neighbors)**.** *The average number of neighbors, N_e, for a signal constellation can be defined as*

$$N_e = \sum_{c=0}^{M-1} N_c p_x(c) \tag{22}$$

where N_c is the number of neighboring points of the constellation of the point $\mathbf{x}_c$, in other words the number of other points of the constellation sharing a common decision region boundary with x_c.

Moreover, N_e is often approximated by

$$N_e \approx \sum_{c=0}^{M-1} \tilde{N}_c p_x(c) \tag{23}$$

where $\tilde{N}_c$ is the set of points at minimum distance from $\mathbf{x}_c$. This approximation is often very tight and facilitates

computation of N_e when signal constellations are complicated. N_e measures the average number of sides of the decision regions surrounding any point in the constellation.

Theorem 4.2 (Nearest Neighbor Union Bound)**.** *The probability of error for the ML detector on the AWGN channel, with a M-point constellation with minimum distance $d_{\min}$ is bounded by*

$$P_e \leq N_e Q\left[\frac{d_{\min}}{2\sigma}\right]. \tag{24}$$

Proof. For each point of the constellation, the distance to each decision region boundary must be at least $\frac{d_{\min}}{2}$. The probability of error for point $\mathbf{x}_c$, $P_{e|c}$ is upper bounded by the union bound as

$$P_{e|c} \leq N_c Q\left[\frac{d_{\min}}{2\sigma}\right].$$

Thus,

$$P_e = \sum_{c=0}^{M-1} P_{e|c} p_{\mathbf{x}(c)} \leq Q\left[\frac{d_{\min}}{2\sigma}\right] \sum_{c=0}^{M-1} N_c p_{\mathbf{x}}(c) = N_e Q\left[\frac{d_{\min}}{2\sigma}\right].$$

□

4.1.3 Computation of the Parameters

We are going to explain how the parameters involved in the formula of the probability of error have been computed.

4.1.3.1 Minimum Distance, $d_{\min}$

A fundamental parameter in order to do a well-round analysis of the probability of error is the minimum distance. Now we are going to study the implemented algorithm in order to compute it.

Compute the minimum distance in a constellation is the same that finding, in a set of points, the closest pair. So we can reformulate the problem as follows including the fact that all the constellations used are defined using Euclidean distance:

Problem: Given a set of points $\mathcal{A} = \{p_1, \cdots, p_n\}$ find the pair of points $\{p_c, p_z\}$ that are closest together, that is to say, which minimize the Euclidean distance.

Brute-force search algorithm

The most natural and straightforward way is using brute-force search that just checks each pair of points and takes the pair with minimum distance. In this case, let n be the number of points in the set we need to compare

$$\binom{n}{2} = \frac{n!}{(n-2)!2!} = \frac{n(n-1)}{2} \quad \text{pair of points.} \tag{25}$$

Here we have the pseudocode of the brute-force search taken and adapted from Cormen et al. [2001]:

Algorithm 1 Brute-Force Closest Pair.

```
1:  procedure BRUTEFORCECLOSESTPAIR(p_1, p_2, ..., p_n)
2:      if n < 2 then
3:          return ∞
4:      else
5:          minimum_distance ← |p_1 − p_2|
6:          closest_pair ← {p_1, p_2}
7:          for each c ∈ [1, n − 1] do
8:              for each z ∈ [c + 1, n] do
9:                  if |p_c − p_z| < minimum_distance then
10:                     minimum_distance ← |p_c − p_z|
11:                     closest_pair ← {p_c, p_z}
12:                 end if
13:             end for
14:         end for
15:         return minimum_distance, closest_pair
16:     end if
17: end procedure
```

It is easy to see, not only analytically by Equation (25) but also from the pseudocode: see the two for loops (Lines 7 and 8 of the Algorithm 1), that the order of the algorithm is $O(n^2)$, where n is the length of the set of points.

The MATLAB code implementation can be found in Annex II.

In order to improve the order of the brute-force search algorithm we could propose an alternative method: the recursive method divide and conquer applied to the problem of the closest pair using Euclidean distance. However, we are using constellations with a relative small number of points and therefore there is not a significant difference in the use of the brute force search algorithm or the recursive method divide and conquer.

4.1.3.2 Average Number of Neighbors, N_e

The implementation of this section, which can be found in Annex II, is an answer to the problem proposed by Burkardt [2013].

In order to compute the Average Number of Neighbors, we need to compute the neighbors of each point given a constellation and its decision regions. There are different ways to compute the decision regions of a set of points. Within Matlab, there are two commands, voronoi and voronoin. It turns out that the command voronoin returns enough information to determine the number of neighbors of each point using Matlab and we can use this information to compute N_e.

The VORONOI diagram of a set of points in the plane divides the plane into polygons and a few infinite regions bounded by straight lines. Matlab adds a point at infinity, and pretends these infinite regions all include that point as a vertex, so from now on, we can pretend that every point is contained in a closed polygon defined by the VORONOI diagram. Two points are neighbors if their polygons share an edge. So our question becomes, how

do we take the information that Matlab returns to describe a VORONOI diagram, and analyze it so that we can determine which points are neighbors?

Let's use the following array of 5 points (p=5) as an example:

```
x5 = [0,1,0,0,-1];
y5 = [0,0,-1,1,0];
```

For this small problem, we could answer our question by computing the diagram with MATLAB command:

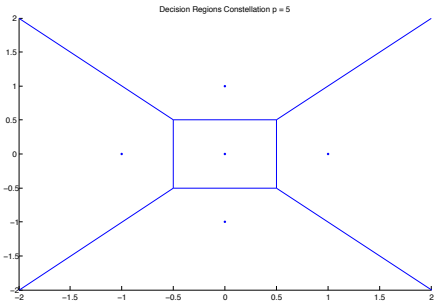


Figure 9: Decision Regions for $p = 5$.

so that, for example, we see that point (0,0) has neighbors (1,0), (0,1), (0,-1) and (-1,0).

However, we wish to be able to extract this information computationally for larger problems, and using Matlab. Let us now consider how to proceed.

MATLAB command voronoin is called using these parameters

```
[ v, g ] = voronoin ( [x(:) y(:)])
```

The vertices v are returned as

```
      Inf       Inf
  -0.5000    0.5000
   0.5000    0.5000
  -0.5000   -0.5000
   0.5000   -0.5000
```

But what is more interesting is the entries of the cell array g, which contain, for each node, the sequence of vertices that form its boundary.

For our data, the g information is:

```
5     3     2     4

      5     1     3

      5     1     4
```

```
3       1       2
```

Now two nodes are neighbors if they share an edge. The edges for node 1 are (5,3), (3,2), (2,4) and (4,5). If I rewrite these edges so the pairs of nodes are sorted, and then sort these edges by first element, I have the edges (3,5), (2,3), (2,4) and (4,5). Node 1 is a neighbor of node I if and only if node I also uses one of these edges to bound its polygonal region. So we can calculate a matrix with the points that share edges and therefore that they are neighbors.

We can see that the nodal neighbor array is:

```
    1       2       3       4       5
  +-----------------------------------
1 | 0       1       1       1       1
2 | 1       0       1       1       0
3 | 1       1       0       0       1
4 | 1       1       0       0       1
5 | 1       0       1       1       0
```

Finally, we compute N_e, the average number of neighbors, by summing columns an averaging.

4.2 Description of Mostly Used Constellations

We are going to introduce the mostly used constellations in order to compare them with the proposed constellations in subsequent sections. This section is based and adapted from Prandoni and Vetterli [2008].

4.2.1 M-PAM

We will start by looking at the simplest form of constellation: PAM (Pulse Amplitude Modulation).

We are going to define a PAM constellation with M symbols, in order to do it we first define a sequence of integers $k[n]$:

$$k[n] \in \{0, 1, \cdots, M-1\}.$$

Now, PAM can be described as a sequence of symbols $a[n]$ defined as:

$$a[n] = A((-M+1) + 2k[n])$$

where we use $M-1$ odd integers around 0. So for instance, if $M = 4$ we have $a[n] \in \{-3A, -A, A, 3A\}$. Finally the PAM constellation with $M = 4$ is defined as the set of points:

$$\mathcal{A} = \{-3A, -A, A, 3A\}.$$

Here we have in Figure 10, the example of a PAM constellation with $M = 4$:

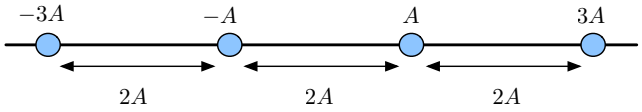


Figure 10: PAM Constellation with $M = 4$.

We can observe that distance between two transmitted symbols, is $2A$. Furthermore, the reason why we use the odd integers is because it creates a zero-mean sequence. If we assume that each symbol is equiprobable, the resulting mean is zero.

4.2.2 M-QAM

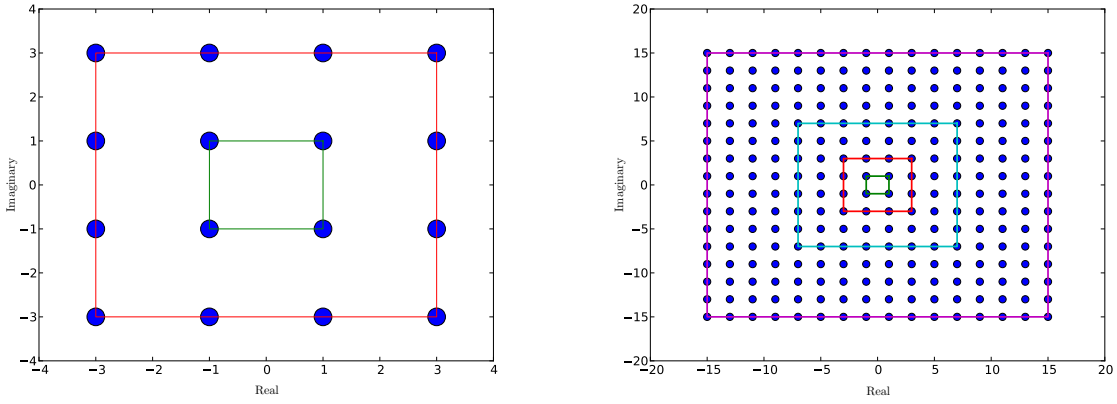
QAM (Quadrature Amplitude Modulation) is an improvement of the PAM constellation, where the main goal is to increase the throughput. In this case we use complex numbers and build a complex valued system of transmission.

The QAM symbol sequence is a sequence of complex numbers with M symbols, where the real part is a $\sqrt{M}$-PAM sequence and the imaginary part is also a $\sqrt{M}$-PAM sequence

$$a[n] = A(a_r[n] + ia_c[n]).$$

So the signal constellation $\mathcal{A}$ is given by points in the complex plane, with odd-valued coordinates around the origin.

Here in Figure 11 we have an example of QAM for different values of M:

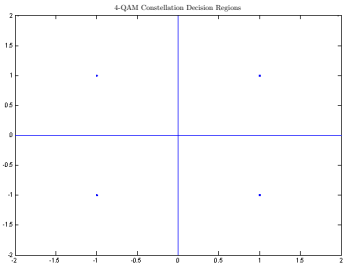


(a) QAM Constellation with $M = 16$

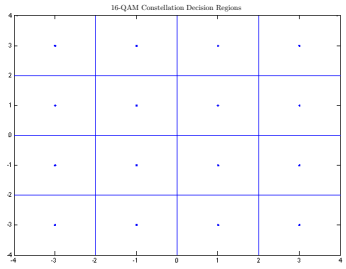
(b) QAM Constellation with $N = 256$

Figure 11: Two Examples of QAM Constellations Using Different Values of M and $A = 1$.

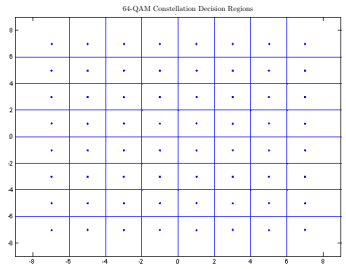
We enclose the decision regions corresponding to 4, 16, 64 and 256 QAM. In order to compute them we have used the function voronoi() implemented in Matlab.



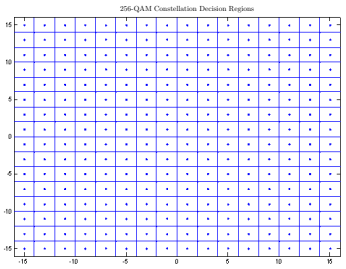
(a) 4-QAM Decision Regions Constellation



(b) 16-QAM Decision Regions Constellation



(c) 64-QAM Decision Regions Constellation



(d) 256-QAM Decision Regions Constellation

Figure 12: M-QAM Decision Regions Constellations with $M = 4, 16, 64, 256$.

Decision regions will be used both as a visual aid to understand the geometrical structure induced in the constellation and as a tool in the computation of the average number of neighbors.

We compute the minimum distance $d_{\min}$ and the average number of neighbors N_e using the algorithms described in earlier sections.

4-QAM Constellation	$d_{\min} = 2$
16-QAM Constellation	$d_{\min} = 2$
64-QAM Constellation	$d_{\min} = 2$
256-QAM Constellation	$d_{\min} = 2$

Table 1: $d_{\min}$ Numerical Results for M-QAM Constellations.

4-QAM Constellation	$N_e = 2$
16-QAM Constellation	$N_e = 3$
64-QAM Constellation	$N_e = 3.5$
256-QAM Constellation	$N_e = 3.75$

Table 2: N_e Numerical Results for M-QAM Constellations.

Finally, based on Proakis [2001] and Madhow [2008], we are going to compare the exact probability of error of M-QAM with the results obtained using the Union Bound and the Nearest Neighbor Union Bound.

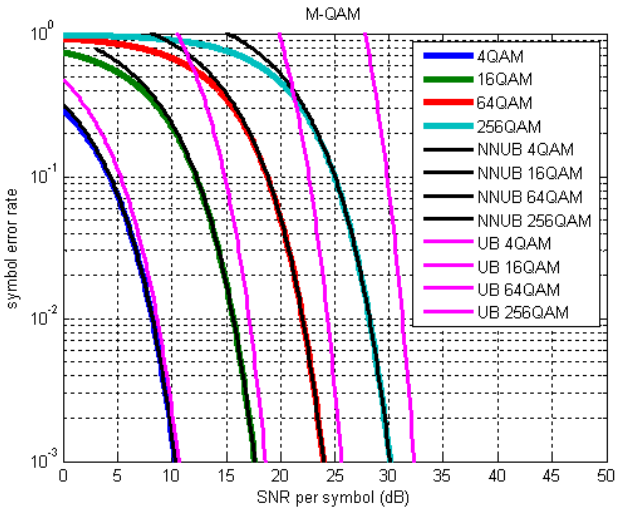


Figure 13: Comparison between NNUB, UB and the Exact Value of the Probability of Error for M-QAM Constellations.

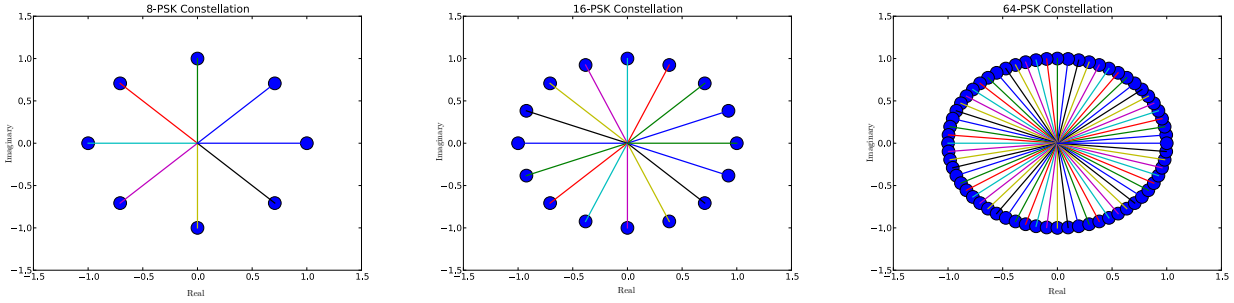
We can see how the Nearest Neighbor Union resembles almost perfectly the exact probability of error whereas the Union Bound is loose when M is large. Therefore, from now on we are going to use the Nearest Neighbor Union bound in order to do our analysis.

4.2.3 M-PSK

The M-PSK constellation is the set:

$$\mathcal{A} = \{Ae^{j2k\pi/M}\}, \quad k = 1, 2, \cdots, M.$$

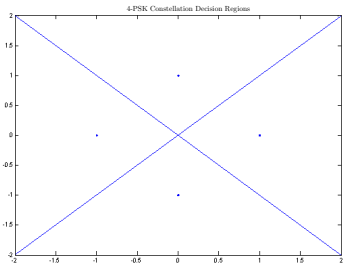
Here there are three examples for different values of M:



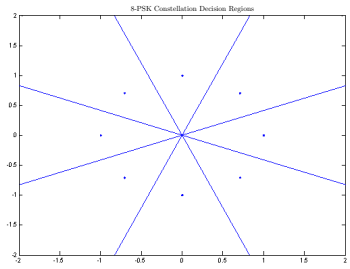
(a) PSK Constellation with $M = 8$ (b) PSK Constellation with $M = 16$ (c) PSK Constellation with $M = 64$

Figure 14: Three Examples of PSK Constellations Using Different Values of M and $A = 1$.

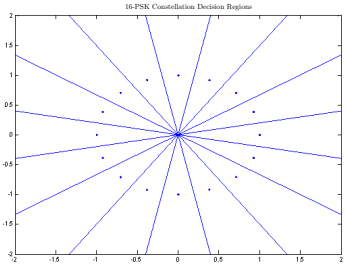
We enclose the decision regions corresponding to 4, 8, 16 and 64 PSK.



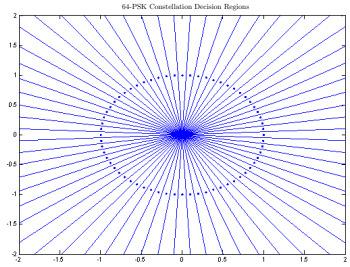
(a) 4-PSK Decision Regions Constellation



(b) 8-PSK Decision Regions Constellation



(c) 16-PSK Decision Regions Constellation



(d) 64-PSK Decision Regions Constellation

Figure 15: M-PSK Decision Regions Constellations with $M = 4, 8, 16, 64$.

We compute the minimum distance $d_{\min}$ and the average number of neighbors N_e using the algorithms described in earlier sections.

4-PSK Constellation	$d_{\min} = 1.4142$
8-PSK Constellation	$d_{\min} = 0.7654$
16-PSK Constellation	$d_{\min} = 0.3902$
64-PSK Constellation	$d_{\min} = 0.0981$

Table 3: $d_{\min}$ Numerical Results for M-PSK Constellations.

4-PSK Constellation	$N_e = 2$
8-PSK Constellation	$N_e = 2$
16-PSK Constellation	$N_e = 2$
64-PSK Constellation	$N_e = 2$

Table 4: N_e Numerical Results for M-PSK Constellations.

Finally, we plot the comparison between Nearest Neighbor Union Bound and Union Bound using these values.

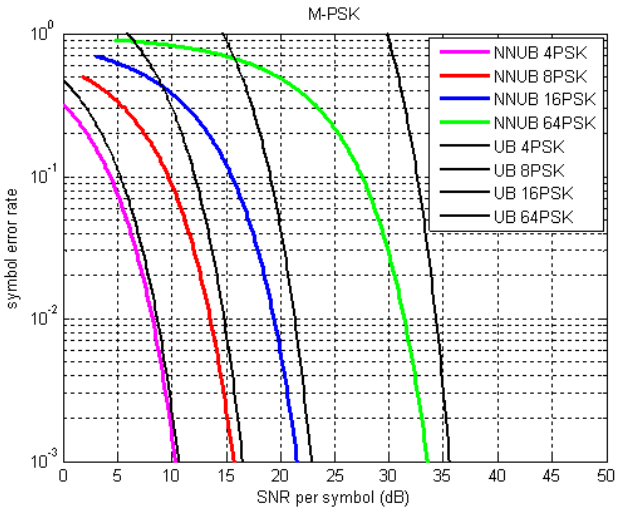


Figure 16: Comparison between NNUB and UB for M-PSK Constellations.

Where we can see that for low M the Union Bound resembles the Nearest Neighbor Union Bound whereas the Union Bound is loose when M is large. In PSK, it is important to note that $M > 16$ PSK constellations are not of practical utility because of its bad performance.

5 Design of Proposed Constellations

In this section we are going to focus on designing constellations over a PNC communication system. First, we will study a case example of design in order to introduce the particular system model under study and understand how it works. Then, we have proposed a general design methodology which will be the basis of the following proposed constellations. Finally, an analysis of each proposed constellation will be done, using the performance metrics described in the previous section.

5.1 Introduction to Design of Constellations

In this section we do a first approach to design constellations. This section builds up on Curtó and Vázquez [2013] and Gupta and Vázquez [2012]; Huber [1994b], and provides a theoretical description.

The first step is to study in detail the mappings that allow us design this first proposed constellation.

In this section we are going to propose a design of the constellation using primes p in $\mathbb{Z}^+$ congruent with 1 modulo 4. By Theorem 3.6 we know that this type of primes in $\mathbb{Z}^+$ can be written as a product of two relatively primes in $\mathbb{Z}[i]$, that is to say, $p = \pi\pi^*$.

Now, by Theorems 3.16, 3.3, 3.14 and 3.15 we can assert that $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ are fields with the same number of elements (Theorem 3.13) and therefore exists an isomorphism between them.

As a first step, we are going to define a homomorphism, explained in Section 3.2.5. Let $\mathbb{Z}[i]$ be the GAUSSIAN Integers and $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ the residue class of $\mathbb{Z}[i]$ modulo π, where the function modulo $\psi : \mathbb{Z}[i] \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ is defined according to

$$\psi(g) = g \bmod \pi.$$

We know that if g is an element of $\mathbb{Z}[i]$, in order to find the corresponding element in $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ we only need to find the remainder of g/π. Therefore, the natural idea to implement this function is to use the theorem of division in $\mathbb{Z}[i]$, explained in Section 3.2.1.1, and solve for the residue γ.

We first state the theorem of division in $\mathbb{Z}[i]$

$$\begin{aligned} &g = \lambda \cdot \pi + \gamma, \\ &\text{with } N(\gamma) < N(\pi), \\ &\text{where } \lambda = \left[\frac{g}{\pi}\right] = \left[\frac{g\pi^*}{\pi\pi^*}\right]. \end{aligned}$$

Note that in this equation we multiply up and down by π^* in order to get the $N(\pi)$ in the denominator, and $[\cdot]$ is the rounding of GAUSSIAN Integers defined in Section 3.2.1.1.

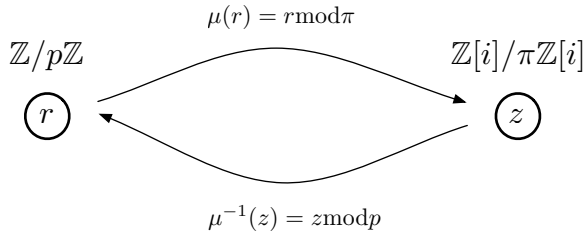
And if we solve for gamma (the residue) we get

$$\begin{aligned}\gamma &= g - \lambda\pi, \\ \gamma &= g - \left[\frac{g\pi^*}{\pi\pi^*}\right]\pi.\end{aligned}$$

Therefore,

$$\psi(g) = g \bmod \pi = \gamma = g - \left[\frac{g\pi^*}{\pi\pi^*}\right]\pi.$$

Based on the previous function modulo, in order to design the constellation, we are going to define an isomorphism between the fields $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$.



Our candidate is the function modulo $\mu : \mathbb{Z}/p\mathbb{Z} \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ defined before as follows:

$$\mu(g) = g \bmod \pi = \gamma = g - \left[\frac{g\pi^*}{\pi\pi^*}\right]\pi.$$

If we want to find the inverse function μ^{-1}, that is to say, the mapping modulo p $\mathbb{Z}[i]/\pi\mathbb{Z}[i] \to \mathbb{Z}/p\mathbb{Z}$ properly, first we need to remember that π and π^* are relatively primes and in terms of BÉZOUT Theorem, it can be translated as:

$$1 = u\pi + v\pi^* \tag{26}$$

where u and v can be computed using the Euclid's Algorithm.

We need to define the inverse mapping in a such way that two conjugated elements in $\mathbb{Z}/p\mathbb{Z}$ will have the same image in $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$.

We need to bind one-to-one element. In order to do that let's think about the element r of the field $\mathbb{Z}/p\mathbb{Z}$ related to z. We can write it as

$$r = k\pi + z \to r \bmod \pi = z \bmod \pi. \tag{27}$$

At the same time, we know that in $\mathbb{Z}$ an integer and its conjugate are the same number $r = r^*$

$$r \quad = \quad r^* = k^*\pi^* + z^* \rightarrow r \bmod \pi = (k^*\pi^* + z^*) \bmod \pi. \tag{28}$$

From the two equations above we know that $z = k^*\pi^* + z^*$ modulo π because when we apply a function to the same element ($r = r^*$) the result must be the same.

Let's now take an element z of the field $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ and multiply it by 1, and use the BÉZOUT Theorem stated in Equation (26)

$$\begin{aligned} z &= z \cdot 1, \\ &= z \cdot (u\pi + v\pi^*), \\ &= zu\pi + zv\pi^*. \end{aligned}$$

We now impose that two conjugated elements in $\mathbb{Z}/p\mathbb{Z}$ have the same image in $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$, which results in the condition $z = k^*\pi^* + z^*$ modulo π

$$\begin{aligned} z &= zu\pi + zv\pi^*, \\ &= (k^*\pi^* + z^*)u\pi + zv\pi, \\ &= k^*u\pi\pi^* + z^*u\pi + zv\pi^*, \end{aligned}$$

and apply $\bmod p$ to the equation above we get

$$\begin{aligned} z \bmod p &= (k^*up + z^*u\pi + zv\pi^*) \bmod p, \\ z \bmod p &= (z^*u\pi + zv\pi^*) \bmod p. \end{aligned}$$

Therefore, we can define the inverse function as the function modulo p as follows:

$$\mu^{-1}(z) = z \bmod p = (z^*u\pi + zv\pi^*) \bmod p.$$

Finally, let's see that effectively this gives $z \bmod p = r \bmod p$.

If r is an integer of $\mathbb{Z}/p\mathbb{Z}$ then r and r^* can be expressed as in Equations (27) and (28). And using the function modulo p defined above:

$$\begin{aligned} z \bmod p &= (z(v\pi^*) + z^*(u\pi)) \bmod p = ((r - k\pi)(v\pi^*) + (r - k^*\pi^*)(u\pi)) \bmod p, \\ &= (rv\pi^* - kv\pi\pi^* + ru\pi - k^*u\pi\pi^*) \bmod p = r(v\pi^* + u\pi) \bmod p, \\ &= r \bmod p. \end{aligned}$$

Thus, we have defined the inverse function.

Now, we will see an example in order to show exactly how the defined mappings are designing a constellation. We are going to use $p = 5$ and $\pi = 2 + i$.

Example 5.1.

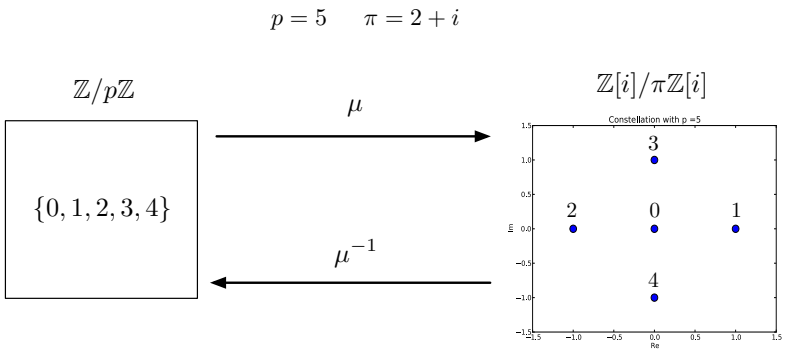


Figure 17: Design of a Constellation with $p = 5$ and $\pi = 2 + i$.

Once we have designed the constellation we are prepared to study the system model.

5.2 Description of the System Model

We consider the following system model based on Gupta and Vázquez [2012] and Nazer and Gastpar [2011] and developed by Curtó and Vázquez [2013], which is a PNC scheme with L sources, a relay and a destination, based on the same principles as described in Section 2.1.

For the sake of understanding, we are going to explain how the system works for the proposed constellation in the earlier subsection. The constellations proposed in Sections 5.4.1 and 5.6.1 can be used in the system straightforward, note that it is only necessary to change the isomorphism used. The other two proposed constellations are designed in order to check the proposed design methodology.

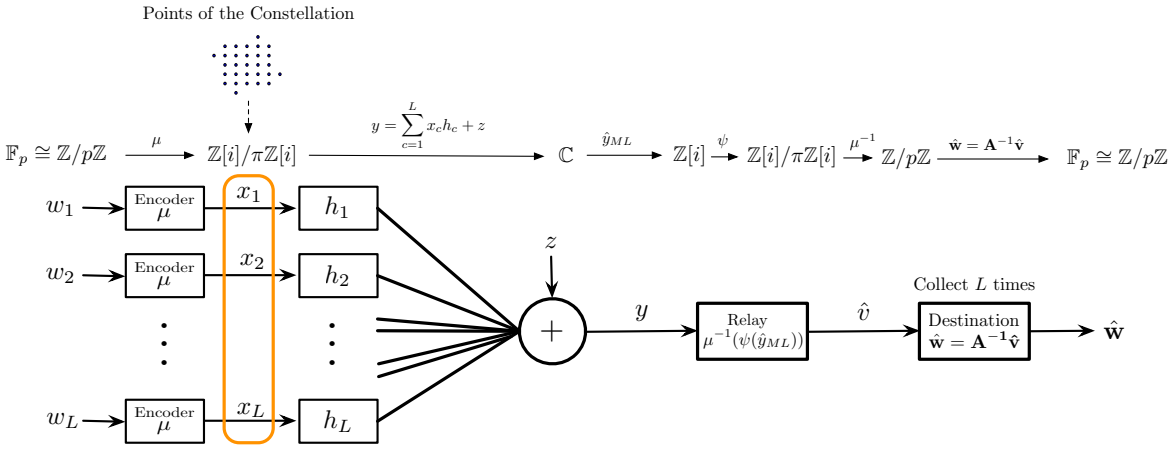


Figure 18: System Model.

Let $\omega_l \in \mathbb{F}_p$ be the message to be transmitted by the l-th source chosen from a finite field $\mathbb{F}_p$. The vector of all source messages is given by $\mathbf{w} = [w_1 \dots w_L]$. Each source encodes the message w_l into a complex point of the signal constellation using the encoder $\mu : \mathbb{F}_p \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ to obtain $x_l = \mu(w_l)$, where μ is the function defined earlier as:

$$\mu(w_l) = w_l \bmod \pi = w_l - \left\lceil \frac{w_l \pi^*}{\pi\pi^*} \right\rfloor \pi.$$

The signals are transmitted across the channel to the relay. We assume that the channel undergoes slow fading, that is to say, remains constant throughout the transmission of each signal.

The signal obtained at the relay is given by

$$y = h_1 x_1 + h_2 x_2 + \ldots + h_L x_L + z \in \mathbb{C}$$

where $h_l \in \mathbb{Z}[i]$ is the channel coefficient between transmitter l and the relay node and $z \in \mathbb{C}$ is i.i.d. GAUSSIAN Noise given by $z \sim \mathcal{CN}(0, \sigma^2)$.

The aim of the relay is to compute a linear combination of source messages in the original message space $v \in \mathbb{Z}/p\mathbb{Z}$ given by

$$v = a_1\omega_1 \oplus a_2\omega_2 \oplus \ldots \oplus a_L\omega_L$$

where $\oplus$ denotes summation over finite field and $a_l \in \mathbb{Z}/p\mathbb{Z}$ can be computed as follows:

$$a_l = \mu^{-1}(\psi(h_l))$$

where $\psi : \mathbb{Z}[i] \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$

$$\psi(h_l) = h_l \bmod \pi = h_l - \left\lceil \frac{h_l \pi^*}{\pi\pi^*} \right\rfloor \pi$$

and $\mu^{-1} : \mathbb{Z}[i]/\pi\mathbb{Z}[i] \to \mathbb{Z}/p\mathbb{Z}$

$$\mu^{-1}(\psi(h_l)) = \psi(h_l) \bmod p = (\psi(h_l)^* u\pi + \psi(h_l) v\pi^*) \bmod p$$

where u and v can be computed using the Euclid's Algorithm, as stated in the previous section.

In order to decode the linear combination v, the relay obtains a ML (Maximum Likelihood) estimate, $\phi : \mathbb{C} \to \mathbb{Z}[i]$, of the received signal y to remove the noise and obtain the closest GAUSSIAN Integer to y

$$\phi(y) = \hat{y}_{ML} = \arg \min_{t \in \mathbb{Z}[i]} ||y - t||^2 \in \mathbb{Z}[i].$$

Further, this signal is mapped to $\mathbb{Z}/p\mathbb{Z}$. Therefore, the decoder at the relay is given by

$$\hat{v} = \mu^{-1}(\psi(\hat{y}_{ML})).$$

The estimate of the linear combination $\hat{v}$ is transmitted to the destination. We assume this transmission between relay and destination to be error free, that is to say, the linear combination is obtained at the destination exactly as estimated at the relay. This procedure gives us a linear combination. However, in order to decode the L transmitted messages w_l from $\hat{v}$, we need to collect L times such linear combinations. Therefore, the L linear combinations obtained at the destination can be written as

$$\begin{bmatrix} \hat{v}^1 \\ \vdots \\ \hat{v}^L \end{bmatrix} = \begin{bmatrix} a_1^1 & \cdots & a_L^1 \\ \vdots & \ddots & \vdots \\ a_1^L & \cdots & a_L^L \end{bmatrix} \begin{bmatrix} \hat{w}_1 \\ \vdots \\ \hat{w}_L \end{bmatrix}.$$

The decoder at the destination inverts the matrix $\mathbf{A}$ and obtains an estimate of $\mathbf{w}$. Therefore,

$$\hat{\mathbf{v}} = \mathbf{A}\hat{\mathbf{w}} \Rightarrow \hat{\mathbf{w}} = \mathbf{A}^{-1}\hat{\mathbf{v}}.$$

Here the inverse of $\mathbf{A}$ is done in $\mathbb{Z}/p\mathbb{Z}$ and so $\mathbf{A}$ is required to be full rank in $\mathbb{Z}/p\mathbb{Z}$ for successful decoding. This operation of decoding is the main reason we are interested in the structure of field.

In order to understand the inner workings of this system model a basic implementation has been done. The MATLAB code is enclosed in Annex (I).

5.3 Proposed Design Methodology

At this point we know that a constellation is a set of points $\mathcal{A}$ and we have done a first approach to design. However, we have not seen a methodology in order to build them. The reason is because there are no hard-and-fast rules for designing constellations.

In this work it is proposed a methodology to design constellations based on the following steps:

Step 1: Select a ring where we are going to design the constellation.

First we need to know which kind of points will form the constellation. In this work we will focus on the design of constellations in $\mathbb{Z}[i]$ and $\mathbb{Z}[w]$.

Step 2: Select a kind of prime.

This choice is necessary in order to define the size of the constellation in the Step 3. We have seen in Theorems 3.6 and 3.9 that each prime in $\mathbb{Z}^+$ can have one of the different types of factorization in both rings $\mathbb{Z}[i]$ and $\mathbb{Z}[w]$.

We are interested only in primes because the number of elements in a field is determined by a power of a prime and the structure of field allows us recover the message at the receiver.

Step 3: Choose the size M of the constellation.

It will be a power of the chosen prime in Step 2. This step can be summarized as selecting a power for the prime in the previous step.

Step 4: Determine a field with M elements in $\mathbb{Z}$ and a field with M elements in the selected ring.

In this step we are interested in finding fields with a finite number of elements using modular arithmetic.

Step 5: Define the mapping of the constellation and its inverse mapping.

5.1 We have to propose a mapping of the constellation using the defined fields in Step 4.

Theoretically always exists an isomorphism between two fields with the same number of elements however in practice we need to check that the proposed mapping is a bijection. If it is not, we will propose another mapping until a positive answer is obtained.

It is important to have in mind that if the proposed mapping of the constellation is not a bijection it will not have practical interest. We need to be aware that without this condition we cannot recover the message at the receiver and so we cannot complete the communication.

5.2 Finally, we need to define the inverse mapping, which existence is given by the fact that the mapping of the constellation is a bijection. This will allow us recover the message at the receiver and finish the communication.

Once we have completed this set of steps, we can build the constellation, find $\mathcal{A}$ using the mapping of the constellation and finally test it, doing the analysis and comparing the results.

In Figure 19 it is shown the flow diagram of the proposed methodology to design constellations.

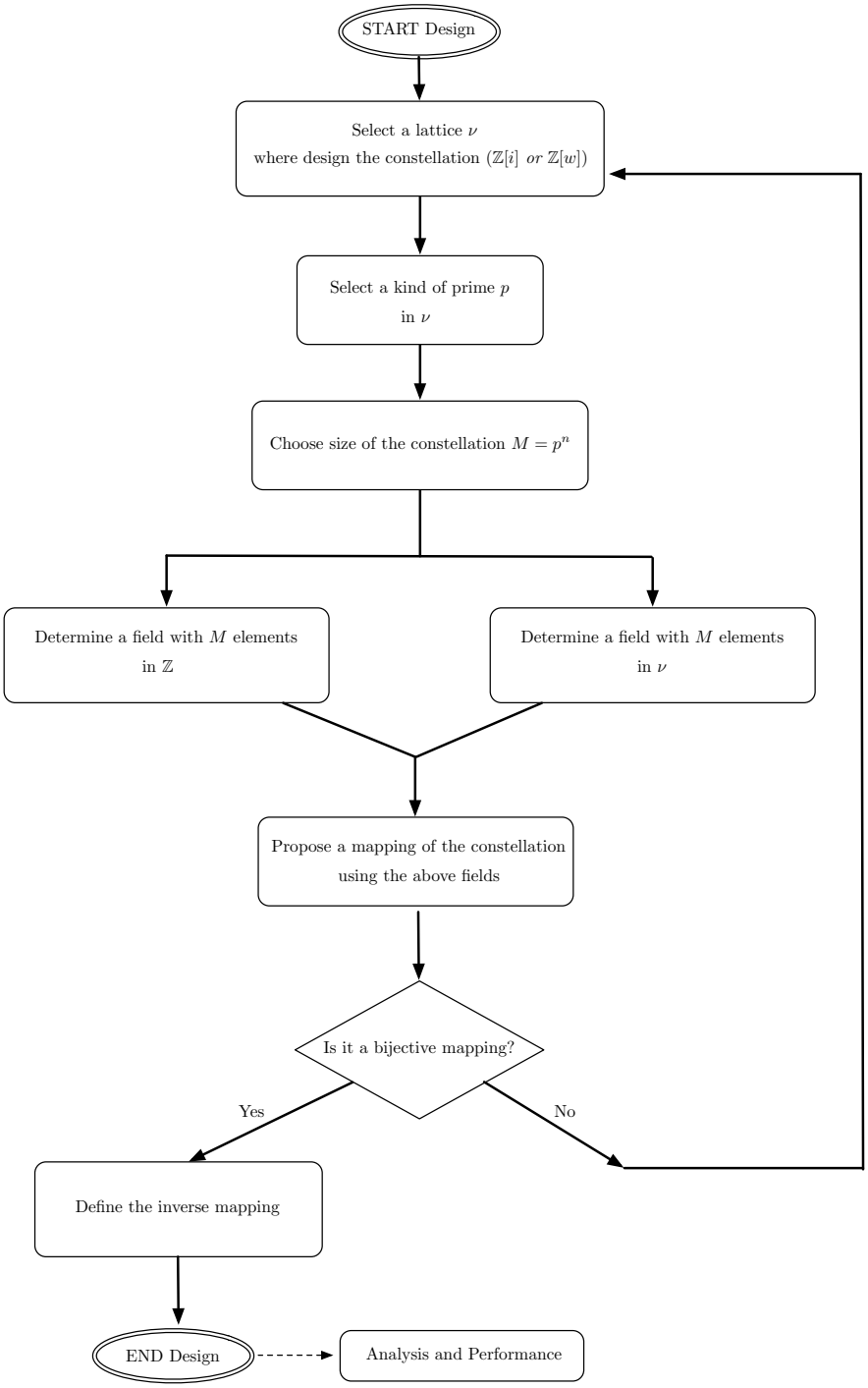


Figure 19: Flow Diagram Design Methodology.

5.4 Design in $\mathbb{Z}[i]$

5.4.1 Design of the Constellation for Primes $p \equiv 1 \bmod 4$ in $\mathbb{Z}[i]$

Step 1: Following the steps of the proposed methodology we first choose the ring $\mathbb{Z}[i]$ where we are going to design the constellation. $\mathcal{A}$ will be a set of GAUSSIAN Integers points.

Step 2: We select primes p in $\mathbb{Z}^+$ with type of factorization $p \equiv 1 \bmod 4$ in $\mathbb{Z}[i]$. We have seen in Theorem 3.6 that this type of primes in $\mathbb{Z}^+$ can be written as a product of two relative primes in $\mathbb{Z}[i]$, that is to say, $p = \pi\pi^*$.

Step 3: We choose the size of the constellation as $M = p$, or equivalently $n = 1$ using the same notation in Figure 19.

Step 4: In order to determine a field with $M = p$ elements in each ring $\mathbb{Z}$ and $\mathbb{Z}[i]$ we have used modular arithmetic:

- We know by Theorem 3.16 that if p is prime in $\mathbb{Z}$, then $\mathbb{Z}/p\mathbb{Z}$ is a field and the number of elements of this field is determined by the absolute value of p. So we propose $\mathbb{Z}/p\mathbb{Z}$ as a field in $\mathbb{Z}$ with $M = p$ elements.
- On the other hand, π is a prime in $\mathbb{Z}[i]$ and we know by Theorem 3.5 that $\mathbb{Z}[i]$ is a Principal Ideal Domain (PID) so we are in the hypothesis of Theorem 3.15 which allows us conclude that (π) is a prime ideal. Moreover, in a PID a prime ideal is a maximal ideal by Theorem 3.14 hence (π) is a maximal ideal in the ring $\mathbb{Z}[i]$. Finally by Theorem 3.3 $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ is a field.

  In this case the number of elements is determined by Theorem 3.13 using the norm of $\pi = a + bi$. It is defined as $N(\pi) = a^2 + b^2 = (a + bi)(a - bi) = \pi\pi^* = p$, so $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ is a field with p elements.

Therefore the proposed fields in this step are $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$.

Step 5: In this case, p decomposes in $\mathbb{Z}[i]$ as $p = \pi\pi^*$, with π a prime number in $\mathbb{Z}[i]$.

Since $\#\left(\mathbb{Z}[i]/\pi\mathbb{Z}[i]\right) = N(\pi) = p$, the isomorphism we are looking for is $\mathbb{F}_p \cong \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ and is obtained as follows:

5.1 The mapping of the constellation between the fields defined above is defined using the function modulo (see Huber [1994b]).

$$\begin{aligned} \mu: \ \mathbb{F}_p &\longrightarrow \mathbb{Z}[i]/\pi\mathbb{Z}[i] \\ x &\longmapsto \mu(x) = x - \left[\frac{x\pi^*}{\pi\pi^*}\right]\pi \end{aligned}$$

Figure 20: Mapping of the Constellation for Primes $p \equiv 1 \bmod 4$.

We have studied it in detail in Section 5.1.

5.2 In Section 5.1 it is proved that the function modulo defined as above is a bijective mapping which inverse is defined as:

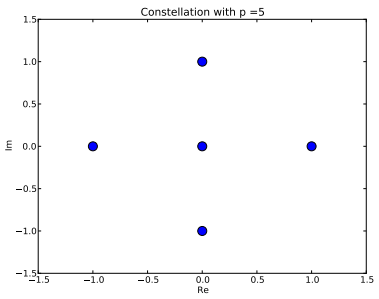
$$\mu^{-1}: \mathbb{Z}[i]/\pi\mathbb{Z}[i] \longrightarrow \mathbb{F}_p$$
$$a \longmapsto \mu^{-1}(a) = (a(v\pi^*) + a^*(u\pi^*)) \mathrm{mod} p$$

Figure 21: Inverse Mapping of the Constellation for Primes $p \equiv 1 \bmod 4$.

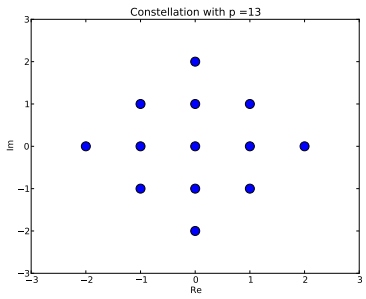
with $u\pi + v\pi^* = 1$.

This last step completes the design of $p \equiv 1 \bmod 4$ constellations in $\mathbb{Z}[i]$.

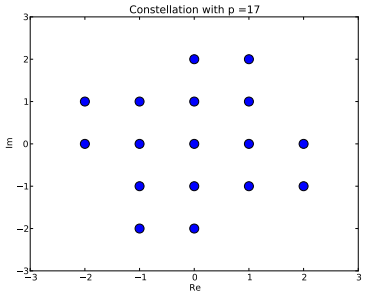
Now, we can see in Figure 22 the obtained constellations for different values of p:



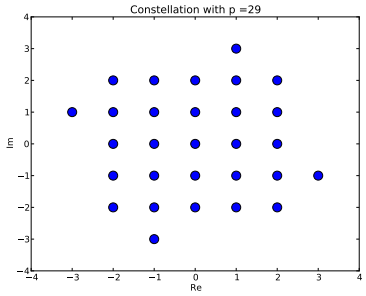
(a) $p \equiv 1 \bmod 4$ Constellation with $p = 5$



(b) $p \equiv 1 \bmod 4$ Constellation with $p = 13$



(c) $p \equiv 1 \bmod 4$ Constellation with $p = 17$



(d) $p \equiv 1 \bmod 4$ Constellation with $p = 29$

Figure 22: Four Examples of $p \equiv 1 \bmod 4$ Constellation Using Different Values of p.

5.4.2 Design of the Constellation for Primes $p \equiv 3 \bmod 4$ in $\mathbb{Z}[i]$

Similarly to the previous case and following the steps of the proposed methodology we have proposed the next design. It is based on the extension for primes $p \equiv 3 \bmod 4$ proposed in Huber [1994b] and it is extended along this section.

Step 1: We choose the ring $\mathbb{Z}[i]$. $\mathcal{A}$ will be a set of GAUSSIAN Integers points.

Step 2: We are going to work with primes p in $\mathbb{Z}^+$ with type of factorization $p \equiv 3 \bmod 4$ in $\mathbb{Z}[i]$. We have seen in Theorem 3.6 that this type of primes are primes in $\mathbb{Z}^+$ which stay primes in $\mathbb{Z}[i]$.

Step 3: We choose the size of the constellation as $M = p^2$, or equivalently $n = 2$ using the same notation in Figure 19.

Step 4: In order to determine a field with $M = p^2$ elements in each ring $\mathbb{Z}$ and $\mathbb{Z}[i]$ we have done the following:

- Since $p \equiv 3 \bmod 4$ is prime in $\mathbb{Z}[i]$. By Theorem 3.15, (p) is a prime ideal in $\mathbb{Z}[i]$, and so a maximal ideal because $\mathbb{Z}[i]$ is a PID (see Theorem 3.5). So, using Theorem 3.3 we obtain that $\mathbb{Z}[i]/p\mathbb{Z}[i]$ is a field.

  The number of elements is determined by Theorem 3.13 using the norm of p . It is defined as $N(p) = pp^* = p^2$, so we propose $\mathbb{Z}[i]/p\mathbb{Z}[i]$ as a field in $\mathbb{Z}[i]$ with $M = p^2$ elements.

- In order to build a field with $M = p^2$ elements in $\mathbb{Z}$ we need to be aware that simply doing the quotient ring $\mathbb{Z}/p^2\mathbb{Z}$ does not guarantee the structure of field because p^2 is not a prime. However, we know that when $n > 1$ $\mathbb{F}_{p^n}$ can be represented as the field of equivalence classes of polynomials whose coefficients belong to $\mathbb{F}_p$ (any irreducible polynomial of degree n yields the same field up to an isomorphism).

  We are interested in applying Theorem 3.17. We have the hypothesis that p is prime, now our goal is to determine a monic irreducible $m(x)$ in $\mathbb{F}_p[X]$ of degree $n = 2$.

  We propose $x^2 + 1$ as a monic irreducible in $\mathbb{F}_p[X]$.

  It is easy to prove its irreducibility $x^2 + 1$; it has two roots i and $-i$ but none in $\mathbb{F}_p$ so we can conclude that $x^2 + 1$ is a monic irreducible in $\mathbb{F}_p[X]$

  Finally using Theorem 3.17 we propose $\mathbb{F}_p[X]/(x^2 + 1)$ as a field with $M = p^2$ elements.

Therefore the proposed fields in this step are $\mathbb{Z}[i]/p\mathbb{Z}[i]$ and $\mathbb{F}_p[X]/(x^2 + 1)$.

Step 5: The isomorphism we are looking for is $\mathbb{Z}[i]/p\mathbb{Z}[i] \cong \mathbb{F}_p[X]/(x^2 + 1)$ with X corresponding to i and is obtained as follows:

We are going to show that $\mathbb{Z}[i]/p\mathbb{Z}[i] \cong \mathbb{F}_p[X]/(x^2 + 1)$ with X corresponding to i proving that each one is isomorphic to $\mathbb{Z}[X]/(p, x^2 + 1)$.

Proof. First we need to prove that $\mathbb{Z}[X]/(x^2 + 1) \cong \mathbb{Z}[i]$ with $X \mapsto i$.

Consider the ring morphism

$$\begin{aligned} \psi : \mathbb{Z}[X] &\longrightarrow \mathbb{Z}[i] \\ P(X) &\longmapsto P(i) \end{aligned}$$

Figure 23: Ring Morphism between $\mathbb{Z}[X]$ and $\mathbb{Z}[i]$.

This is clearly surjective: every element $a + bi$ in $\mathbb{Z}[i]$ has a corresponding element $a + bX$ in $\mathbb{Z}[X]$ given by $\psi(a + bX) = a + bi$.

The kernel contains $(x^2 + 1)$. Moreover, if one writes the Euclidean division of $P(X)$ by $x^2 + 1$, one obtains a remainder of degree 1, $a + bX$, which is zero if and only if $a + bi = \psi(P(X))$ is zero, so the kernel is the ideal $(x^2 + 1)$.

By the NOETHER First Isomorphism Theorem we know that the image of ψ is isomorphic to the quotient ring $\mathbb{Z}[X]/\text{Kernel}(\psi)$. Finally, using that ψ is surjective, $\text{Image}(\psi) = \mathbb{Z}[i]$, we obtain $\mathbb{Z}[X]/(x^2 + 1) \cong \mathbb{Z}[i]$ with $X \mapsto i$.

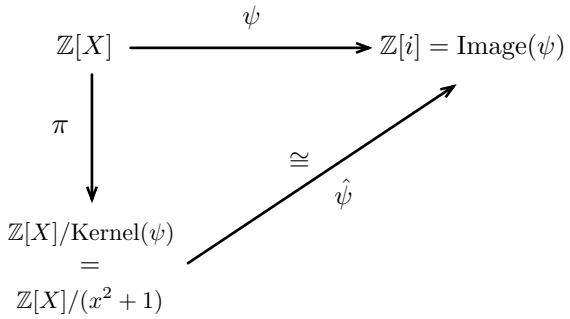


Figure 24: Diagram of the NOETHER First Isomorphism Theorem.

where π: $\mathbb{Z}[X] \to \mathbb{Z}[X]/(x^2 + 1)$ is the canonical projection studied in detail in Section 3.2.5.

Finally, let p be a prime and $\mathbb{Z}[i]$ a PID (hence an integral domain) by Theorem 3.15 we know that $p\mathbb{Z}[i]$ is a prime ideal and the inverse of a prime ideal is a prime ideal too. Therefore we can assert that $\psi^{-1}(p\mathbb{Z}[i]) = (p, x^2 + 1)$.

Hence, $\mathbb{Z}[i]/p\mathbb{Z}[i] \cong \mathbb{Z}[X]/(p, x^2 + 1)$.

On the other hand, since $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$ we have $\mathbb{F}_p[X]/(x^2 + 1) \cong \mathbb{Z}[X]/(p, x^2 + 1)$:

$$\begin{aligned} \mathbb{F}_p[X]/(x^2 + 1) &\cong (\mathbb{Z}/p\mathbb{Z})[X]/(x^2 + 1), \\ &\cong (\mathbb{Z}[X]/(p))/(x^2 + 1), \\ &\cong \mathbb{Z}[X]/(p, x^2 + 1). \end{aligned}$$

And that ends the proof.

□

This proof allows us define the mappings of the constellation as follows:

5.1 The mapping of the constellation is defined as:

$$\gamma : \mathbb{F}_p[X]/(x^2+1) \longrightarrow \mathbb{Z}[i]/p\mathbb{Z}[i]$$
$$x \longmapsto i$$

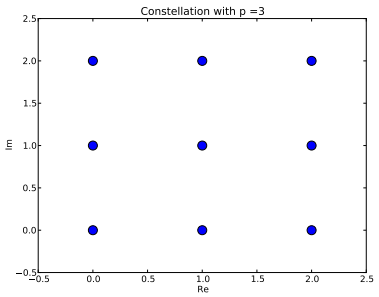
Figure 25: Mapping of the Constellation for Primes $p \equiv 3 \bmod 4$.

5.2 The inverse mapping of the constellation is defined as:

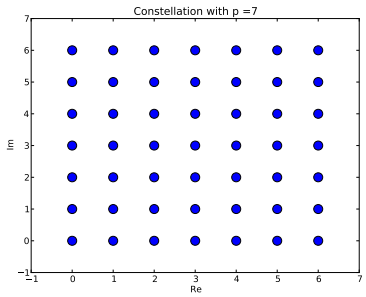
$$\gamma^{-1} : \mathbb{Z}[i]/p\mathbb{Z}[i] \longrightarrow \mathbb{F}_p[X]/(x^2+1)$$
$$i \longmapsto x$$

Figure 26: Inverse Mapping of the Constellation for Primes $p \equiv 3 \bmod 4$.

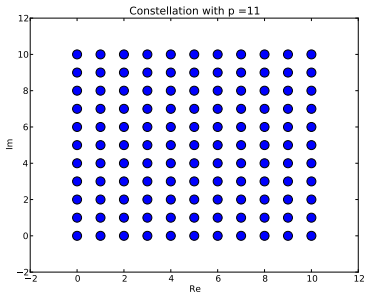
At this point we have finished the process of designing the constellation. Now in Figure 27 we can observe the resulting design of $p \equiv 3 \bmod 4$ constellations in $\mathbb{Z}[i]$ for different values of p.



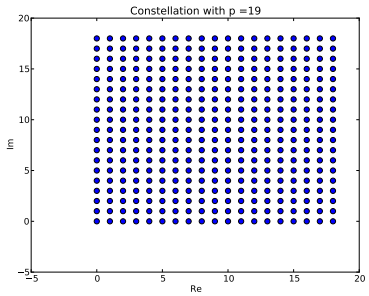
(a) $p \equiv 3$ modulo 4 Constellation with $p = 3$



(b) $p \equiv 3 \bmod 4$ Constellation with $p = 7$



(c) $p \equiv 3 \bmod 4$ Constellation with $p = 11$



(d) $p \equiv 3 \bmod 4$ Constellation with $p = 19$

Figure 27: Four Examples of $p \equiv 3 \bmod 4$ Constellation Using Different Values of p.

5.5 Best Performing Design(s) in $\mathbb{Z}[i]$

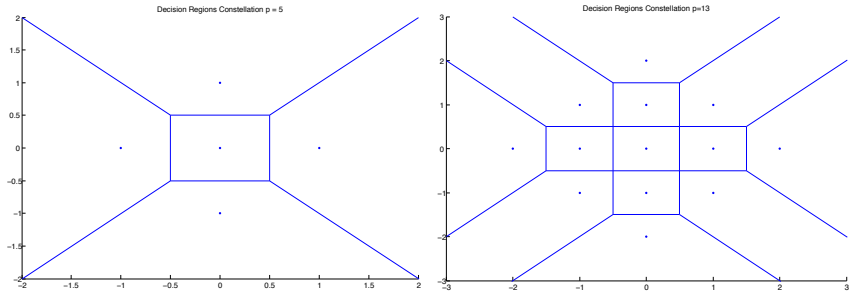
5.5.1 Decision Regions

We have seen in Section 4.1.1 that the decision region for a point x_c in the constellation $\mathcal{A} = \{x_c\}_{c=0,\cdots,M}$, denoted $\mathcal{R}_{x_c}$, is the set of points of the complex plane that are closer to x_c than to any other point of the signal

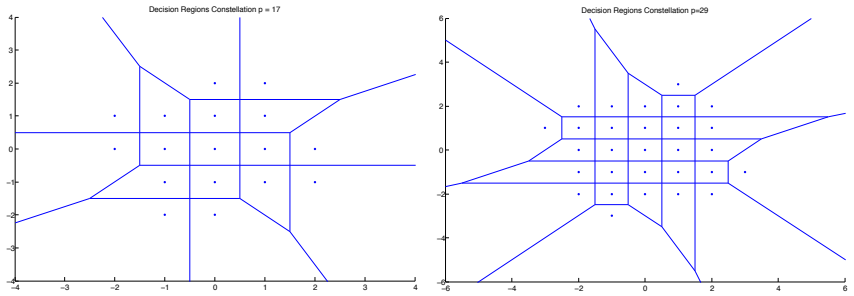
constellation. Moreover, the decision region for a point x_c is a polygon, often an irregular polygon, whose sides are the perpendicular bisectors of the lines between x_c and the neighbors of x_c. The latter is because in all the designs we are using Euclidean distance.

Now, we are interested in showing the resulting decision regions for the proposed constellations.

We have obtained the next decision regions for $1 \bmod 4$ constellations:



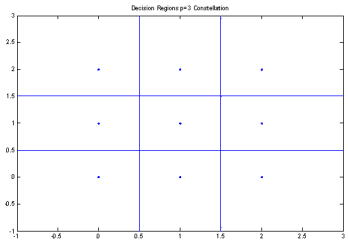
(a) $p \equiv 1 \bmod 4$ Constellation with $p = 5$ (b) $p \equiv 1 \bmod 4$ Constellation with $p = 13$



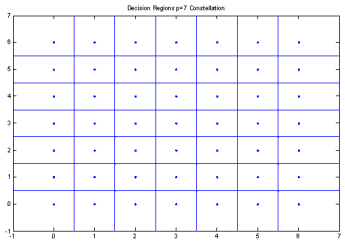
(c) $p \equiv 1 \bmod 4$ Constellation with $p = 17$ (d) $p \equiv 1 \bmod 4$ Constellation with $p = 29$

Figure 28: Decision Regions of $p \equiv 1 \bmod 4$ Constellation.

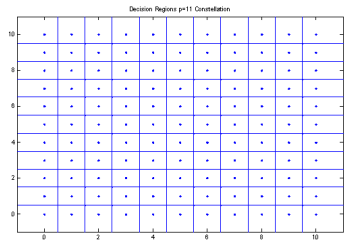
We enclose the resulting decision regions for constellations $3 \bmod 4$ in $\mathbb{Z}[i]$



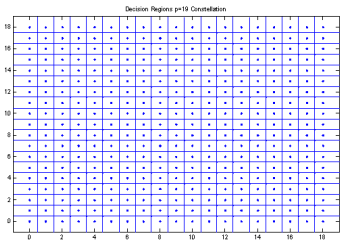
(a) $p \equiv 3 \bmod 4$ Constellation with $p = 3$



(b) $p \equiv 3 \bmod 4$ Constellation with $p = 7$



(c) $p \equiv 3 \bmod 4$ Constellation with $p = 11$



(d) $p \equiv 3 \bmod 4$ Constellation with $p = 19$

Figure 29: Decision Regions of $p \equiv 3 \bmod 4$ Constellation.

Where we can see that $3 \bmod 4$ constellations have the same decision region's shape than common QAM.

5.5.2 Analysis of the Probability of Error

We implement the Nearest Neighbor Union Bound, explained in Section 4.1.2.2

$$P_e \leq N_e \cdot Q\left[\frac{d_{\min}}{2\sigma}\right]. \tag{29}$$

We start working with $1 \bmod 4$ constellations and proceed to do the analysis with $3 \bmod 4$ constellations.

Using the algorithms described in Sections 4.1.3.1 and 4.1.3.2 we obtain $d_{\min}$ and N_e parameters for $1 \bmod 4$ constellations.

$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 5$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 13$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 17$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 29$	$d_{\min} = 1$

Table 5: $d_{\min}$ Numerical Results for $\mathbb{Z}[i]$ Constellation $1 \bmod 4$.

$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 5$	$N_e = 3.2$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 13$	$N_e = 3.6923$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 17$	$N_e = 3.7647$
$\mathbb{Z}[i]$ Constellation $1 \bmod 4$ with $p = 29$	$N_e = 4.1379$

Table 6: N_e Numerical Results for $1 \bmod 4$ Constellations in $\mathbb{Z}[i]$.

For constellations $p = 5$, $p = 13$, $p = 17$ and $p = 29$ in $\mathbb{Z}[i]$ we plot the Nearest Neighbor Union Bound in the next figure. We also plot 4-QAM and 16-QAM as a reference.

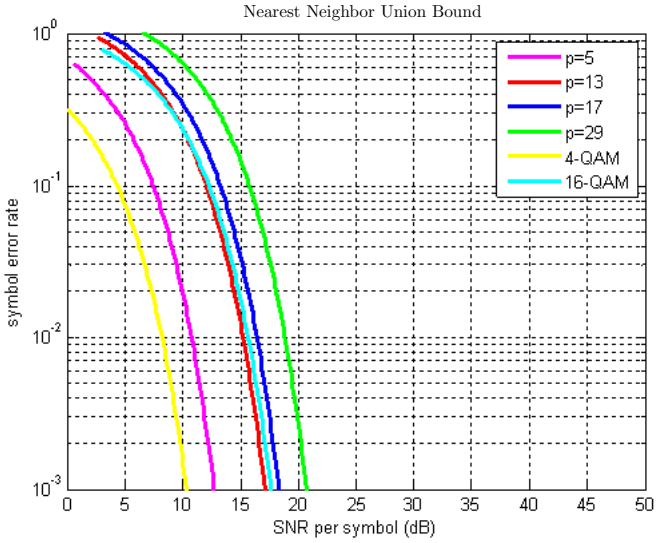


Figure 30: Nearest Neighbor Union Bound for $1 \bmod 4$ Constellations.

We can see that constellations $1 \bmod 4$ are at most as good as QAM constellations but not better. For instance, $p = 17$ and 16-QAM present almost the same behavior, we can observe a 1dB difference and just one point difference. However, for $p = 5$ and 4-QAM, common QAM outperforms constellations $1 \bmod 4$, here we can appreciate an average 3dB difference. Therefore, constellations $1 \bmod 4$ represent an alternative to common QAM but not imply improvement.

We consider now the constellations $3 \bmod 4$ in $\mathbb{Z}[i]$.

We compute $d_{\min}$

$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 3$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 7$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 11$	$d_{\min} = 1$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 19$	$d_{\min} = 1$

Table 7: $d_{\min}$ Numerical Results for $\mathbb{Z}[i]$ Constellation $3 \bmod 4$.

We can see that the $d_{\min}$ value obtained for all the constellations in $\mathbb{Z}[i]$ remains the same with a value of $d_{\min} = 1$. This can be understood as a consequence of how constellations in $\mathbb{Z}[i]$ are generated using the function modulo: the set of classes defining the points of a constellation $\mathcal{A}_1$ are included in the set of classes defining another constellation $\mathcal{A}_2$ with a larger dimension. Therefore if the first constellation achieves the minimum distance of the ring all the other constellations larger than it will have the same value of $d_{\min}$.

We also compute N_e and obtain:

$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 3$	$N_e = 2.6667$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 7$	$N_e = 3.4286$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 11$	$N_e = 3.6364$
$\mathbb{Z}[i]$ Constellation $3 \bmod 4$ with $p = 19$	$N_e = 3.7895$

Table 8: N_e Numerical Results for $3 \bmod 4$ Constellations in $\mathbb{Z}[i]$.

And we plot the Nearest Neighbor Union Bound for $p = 3$, $p = 7$, $p = 11$ and $p = 19$ for $3 \bmod 4$ constellations in $\mathbb{Z}[i]$. We plot 16-QAM, 64-QAM and 256-QAM as a reference.

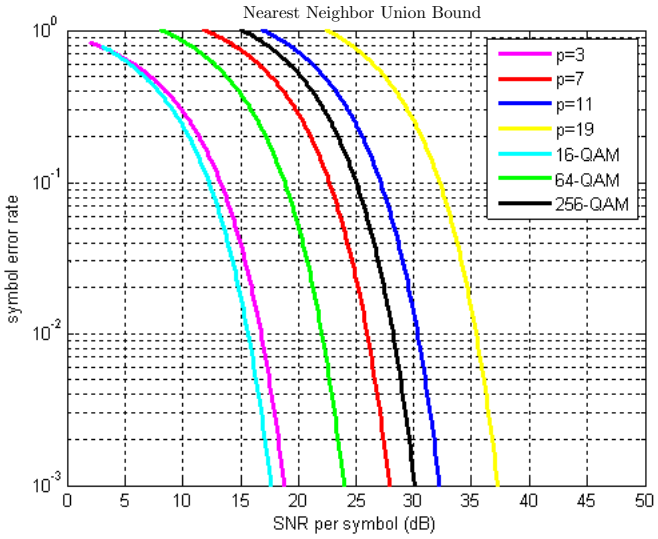


Figure 31: Nearest Neighbor Union Bound for $3 \bmod 4$ Constellations.

We can see that constellations $3 \bmod 4$ in $\mathbb{Z}[i]$ show worse results than QAM constellations. For instance, we can appreciate that common 16-QAM outperforms $p = 3$ (9 points constellation). Moreover, for $p = 7$ (49 points constellation) we can observe there is an average 4dB difference with 64-QAM, which means that $3 \bmod 4$ is worse than QAM. Therefore, $3 \bmod 4$ constellations in $\mathbb{Z}[i]$ are much worse than common QAM.

5.6 Design in $\mathbb{Z}[w]$

5.6.1 Design of the Constellation for Primes $p \equiv 1 \bmod 6$ in $\mathbb{Z}[w]$

Now, we are going to design constellations in the ring of EISENSTEIN Integers, as in the previous sections we will follow the steps of the proposed methodology.

Step 1: First, we have chosen the ring $\mathbb{Z}[w]$. The proposed constellation $\mathcal{A}$ will be a set of EISENSTEIN Integers points.

Step 2: We select primes p in $\mathbb{Z}^+$ with type of factorization $p \equiv 1 \bmod 6$ in $\mathbb{Z}[w]$. We have seen in Theorem 3.9 that this type of primes can be written as sum $p = a^2 + 3b^2$. Therefore, such primes p in $\mathbb{Z}^+$ are the product of two conjugate primes in EISENSTEIN Integers:

$$p = a^2 + 3b^2 = \pi\pi^* \tag{30}$$

where $\pi = a + b + w2b$ and the conjugate of π is $\pi^* = a + b + w^2 2b$.

Step 3: We choose the size of the constellation as $M = p$, or equivalently $n = 1$ using the same notation in Figure 19.

Step 4: In order to determine a field with $M = p$ elements in each ring $\mathbb{Z}$ and $\mathbb{Z}[i]$ we have used modular arithmetic:

- We know by Theorem 3.16 that if p is prime in $\mathbb{Z}$, then $\mathbb{Z}/p\mathbb{Z}$ is a field and the number of elements of this field is determined by the absolute value of p. So we propose $\mathbb{Z}/p\mathbb{Z}$ as a field in $\mathbb{Z}$ with $M = p$ elements.
- On the other hand, π is a prime in $\mathbb{Z}[w]$ and we know by Theorem 3.8 that $\mathbb{Z}[w]$ is a Principal Ideal Domain (PID) so we are in the hypothesis of Theorem 3.15 which allows us conclude that (π) is a prime ideal. Moreover, in a PID a prime ideal is a maximal ideal by Theorem 3.14 hence (π) is a maximal ideal in the ring $\mathbb{Z}[w]$. Finally by Theorem 3.3 $\mathbb{Z}[w]/\pi\mathbb{Z}[w]$ is a field.

  In this case the number of elements is determined by Theorem 3.13 using the norm of $\pi = a + b + w2b$. It is defined as $N(\pi) = \pi\pi^* = p$, so $\mathbb{Z}[w]/\pi\mathbb{Z}[w]$ is a field with p elements.

Therefore, the proposed fields in this step are $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}[w]/\pi\mathbb{Z}[w]$.

Step 5: In this case, p decomposes in $\mathbb{Z}[w]$ as $p = \pi\pi^*$, with π a prime number in $\mathbb{Z}[w]$.

Since $\#(\mathbb{Z}[w]/\pi\mathbb{Z}[w]) = N(\pi) = p$, the isomorphism we are looking for is $\mathbb{F}_p \cong \mathbb{Z}[w]/\pi\mathbb{Z}[w]$ and is obtained as follows:

5.1 The mapping of the constellation between the fields defined above is defined using the function modulo (see Huber [1994a, 2004]):

$$\begin{aligned} \tilde{\mu}: \ \mathbb{F}_p &\longrightarrow \mathbb{Z}[w]/\pi\mathbb{Z}[w] \\ x &\longmapsto \tilde{\mu}(x) = x - \left\lceil \frac{x\pi^*}{\pi\pi^*} \right\rfloor \pi \end{aligned}$$

Figure 32: Mapping of the Constellation for Primes $p \equiv 1 \bmod 6$.

We have studied it in detail in Section 5.4 for GAUSSIAN Integers, all the study is the same using the analog theorems for EISENSTEIN Integers.

5.2 Using the same result obtained in Section 5.4 for GAUSSIAN Integers and adapt it for EISENSTEIN Integers, we can assert that the function modulo defined above is a bijective mapping which inverse is defined as:

$$\begin{aligned} \mu^{-1} : \mathbb{Z}[w]/\pi\mathbb{Z}[w] &\longrightarrow \mathbb{F}_p \\ a &\longmapsto \mu^{-1}(a) = (a(v\pi^*) + a^*(u\pi^*)) \mathrm{mod} p \end{aligned}$$

Figure 33: Inverse Mapping of the Constellation for Primes $p \equiv 1 \bmod 6$.

with

$$u\pi + v\pi^* = 1. \tag{31}$$

Testing is straightforward, using the above equation (31) we immediately get the inverse mapping $\tilde{\mu}^{-1}$:

$$x = \tilde{\mu}^{-1}(a) \equiv (a(v\pi^*) + a^*(u\pi)) \bmod p, \tag{32}$$

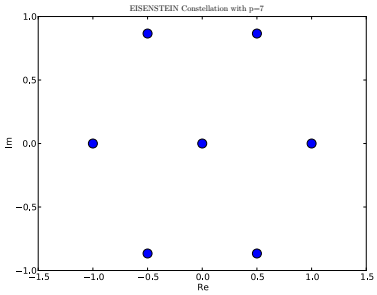
because if x is an integer of $\mathbb{F}_p$ then $x = k\pi + a$ and $x = x^* = k^*\pi^* + a^*$, hence

$$a(v\pi^*) + a^*(u\pi) = (x - k\pi)(v\pi^*) + (x - k^*\pi^*)(u\pi) \equiv (x(v\pi^* + u\pi)) \bmod p \tag{33}$$

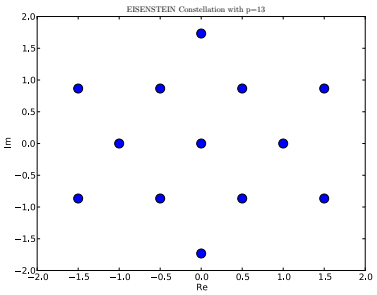
which equals x by Equation (31).

This last step completes the design of $p \equiv 1 \bmod 6$ constellations in $\mathbb{Z}[w]$.

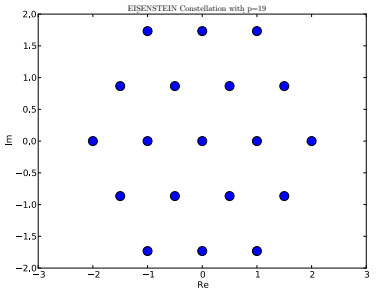
Now, we can see in Figure 34 the implemented constellations for different values of p:



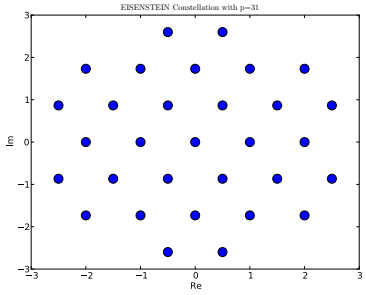
(a) $p \equiv 1 \bmod 6$ Constellation with $p = 7$



(b) $p \equiv 1 \bmod 6$ Constellation with $p = 13$



(c) $p \equiv 1 \bmod 6$ Constellation with $p = 19$



(d) $p \equiv 1 \bmod 6$ Constellation with $p = 31$

Figure 34: Four Examples of $p \equiv 1 \bmod 6$ Constellation Using Different Values of p.

5.6.2 Design of the Constellation for Primes $p \equiv 2 \bmod 3$ in $\mathbb{Z}[w]$

Similarly to the previous case and following the steps of the used methodology we have proposed the next design. It is based on the extension for primes $p \equiv 2 \bmod 3$ in Huber [1994a] and it is extended along this section.

Step 1: We choose the ring $\mathbb{Z}[w]$. The constellation, $\mathcal{A}$, will be a set of EISENSTEIN Integers points.

Step 2: We are going to work with primes p in $\mathbb{Z}^+$ with type of factorization $p \equiv 2 \bmod 3$ in $\mathbb{Z}[w]$. We have seen in Theorem 3.9 that this type of primes in $\mathbb{Z}^+$ stay primes in $\mathbb{Z}[w]$.

Step 3: We choose the size of the constellation as $M = p^2$, or equivalently $n = 2$ using the same notation in Figure 19.

Step 4: In order to determine a field with $M = p^2$ elements in $\mathbb{Z}$ and $\mathbb{Z}[w]$ we have proceeded as follows:

- Since $p \equiv 2 \bmod 3$ is prime in $\mathbb{Z}[w]$ and it is a PID (Theorem 3.8) we have by Theorem 3.15 that (p) is a prime ideal in $\mathbb{Z}[w]$. Moreover, in a PID a prime ideal is a maximal ideal by Theorem 3.14. Finally, using Theorem 3.3 we obtain that $\mathbb{Z}[w]/p\mathbb{Z}[w]$ is a field.

  The number of elements is determined by Theorem 3.13 using the norm of p. It is defined as $N(p) = pp^* = p^2$, so we propose $\mathbb{Z}[w]/p\mathbb{Z}[w]$ as a field in $\mathbb{Z}[w]$ with $M = p^2$.

- As in Section 5.4.2, in order to build a field with $M = p^2$ elements in $\mathbb{Z}$ we need to be aware that simply doing the quotient ring $\mathbb{Z}/p^2\mathbb{Z}$ does not guarantee the structure of field because p^2 is not a prime. However, we know that when $n > 1$ $\mathbb{F}_{p^n}$ can be represented as the field of equivalence classes of polynomials whose coefficients belong to $\mathbb{F}_p$.

  We are interested in applying Theorem 3.17. We have the hypothesis that p is prime, now our goal is to determine a monic irreducible $m(x)$ in $\mathbb{F}_p[X]$ of degree $n = 2$.

  We propose $x^2 + x + 1$ as a monic irreducible in $\mathbb{F}_p[X]$.

  It is easy to prove that $x^2 + x + 1$ is irreducible in $\mathbb{F}_p[X]$; $x^2 + x + 1$ has two roots $(-1 - \sqrt{-3}i)/2$ and $(-1 + \sqrt{-3}i)/2$ but none in $\mathbb{F}_p$ so $x^2 + x + 1$ as a monic irreducible in $\mathbb{F}_p[X]$.

  Finally using Theorem 3.17 we propose $\mathbb{F}_p[X]/(x^2 + x + 1)$ as a field with $M = p^2$ elements.

Therefore, the proposed fields in this step are $\mathbb{Z}[w]/p\mathbb{Z}[w]$ and $\mathbb{F}_p[X]/(x^2 + x + 1)$.

Step 5: The isomorphism we are looking for is $\mathbb{Z}[w]/p\mathbb{Z}[w] \cong \mathbb{F}_p[X]/(x^2 + x + 1)$ with X corresponding to w and it is obtained step by step using the same proof as in Section 5.4.2. Hence this proof allows us define the mappings of the constellation as follows:

5.1 The mapping of the constellation is defined as:

$$\begin{aligned} \tilde{\gamma} : \mathbb{F}_p[X]/(x^2 + x + 1) &\longrightarrow \mathbb{Z}[w]/p\mathbb{Z}[w] \\ x &\longmapsto w \end{aligned}$$

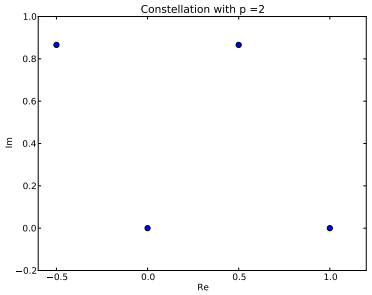
Figure 35: Mapping of the Constellation for Primes $p \equiv 2 \bmod 3$.

5.2 The inverse mapping of the constellation is defined as:

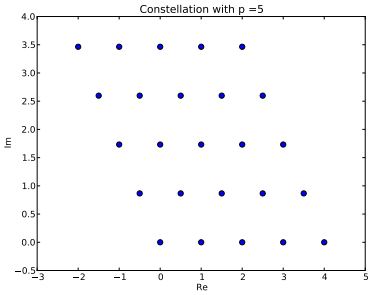
$$\begin{aligned} \tilde{\gamma}^{-1} : \mathbb{Z}[w]/p\mathbb{Z}[w] &\longrightarrow \mathbb{F}_p[X]/(x^2 + x + 1) \\ w &\longmapsto x \end{aligned}$$

Figure 36: Inverse Mapping of the Constellation for Primes $p \equiv 2 \bmod 3$.

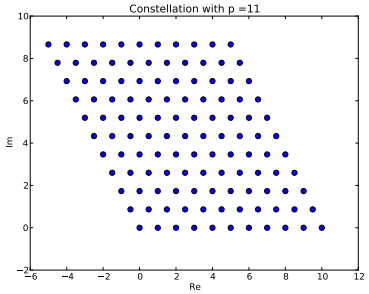
At this point we have finished the process of design. Now, in Figure 37 we can observe the resulting design of $p \equiv 2 \bmod 3$ constellations in $\mathbb{Z}[w]$ for different values of p:



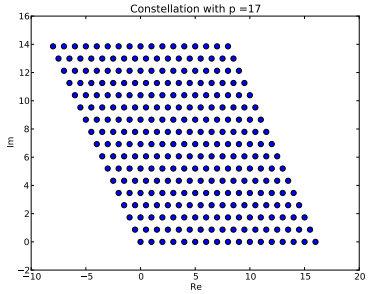
(a) $p \equiv 2 \bmod 3$ Constellation with $p = 2$



(b) $p \equiv 2 \bmod 3$ Constellation with $p = 5$



(c) $p \equiv 2 \bmod 3$ Constellation with $p = 11$



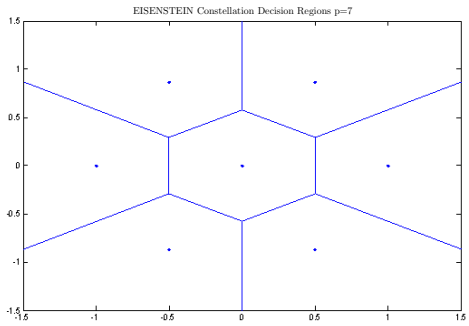
(d) $p \equiv 2 \bmod 3$ Constellation with $p = 17$

Figure 37: Four Examples of $p \equiv 2 \bmod 3$ Constellations Using Different Values of p.

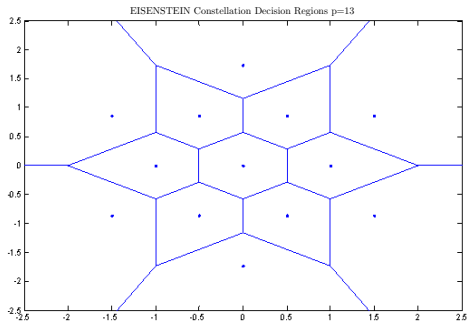
5.7 Best Performing Design(s) in $\mathbb{Z}[w]$

5.7.1 Decision Regions

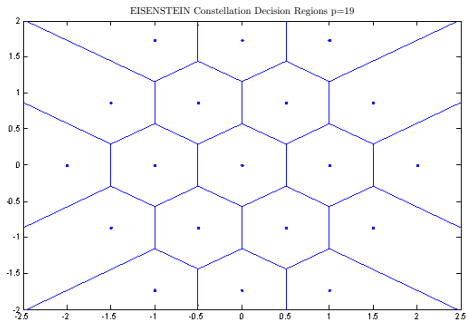
We plot the resulting decision regions for constellations $1 \bmod 6$ in $\mathbb{Z}[w]$:



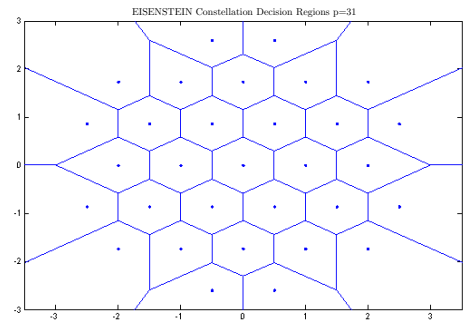
(a) EISENSTEIN Constellation $p \equiv 1 \bmod 6$ with $p = 7$



(b) EISENSTEIN Constellation $p \equiv 1 \bmod 6$ with $p = 13$



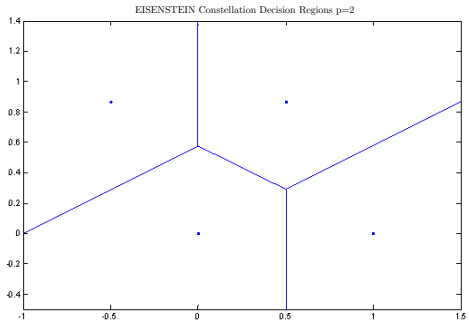
(c) EISENSTEIN Constellation $p \equiv 1 \bmod 6$ with $p = 19$



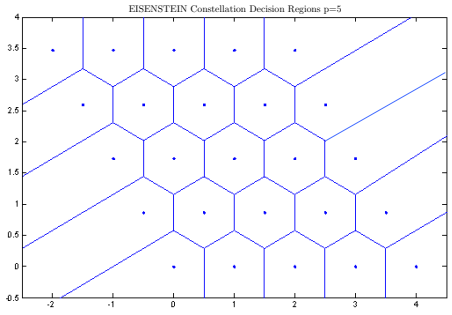
(d) EISENSTEIN Constellation $p \equiv 1 \bmod 6$ with $p = 31$

Figure 38: EISENSTEIN Constellation Decision Regions of $p \equiv 1 \bmod 6$.

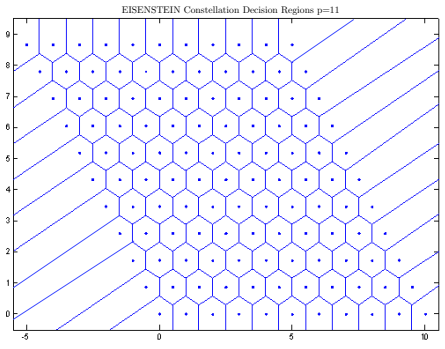
We plot the same results for constellations $2 \bmod 3$ in $\mathbb{Z}[w]$:



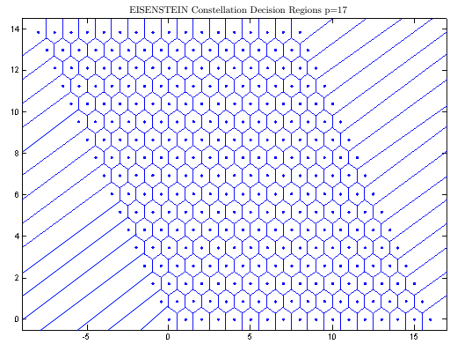
(a) EISENSTEIN Constellation $p \equiv 2 \bmod 3$ with $p = 2$



(b) EISENSTEIN Constellation $p \equiv 2 \bmod 3$ with $p = 5$



(c) EISENSTEIN Constellation $p \equiv 2 \bmod 3$ with $p = 11$



(d) EISENSTEIN Constellation $p \equiv 2 \bmod 3$ with $p = 17$

Figure 39: EISENSTEIN Constellation Decision Regions of $p \equiv 2 \bmod 3$.

5.7.2 Analysis of Probability of Error

We implement the Nearest Neighbor Union Bound explained in Section 4.1.2.2

$$P_e \leq N_e \cdot Q\left[\frac{d_{\min}}{2\sigma}\right].$$

We start working with $1 \bmod 6$ constellations and proceed to do the analysis with $2 \bmod 3$.

Using the algorithms described in Sections 4.1.3.1 and 4.1.3.2 we obtain $d_{\min}$ and N_e parameters for $1 \bmod 6$ constellations.

$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 7$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 13$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 19$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 31$	$d_{\min} = 1$

Table 9: $d_{\min}$ Numerical Results for $\mathbb{Z}[w]$ Constellation $1 \bmod 6$.

$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 7$	$N_e = 3.4286$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 13$	$N_e = 4.6154$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 19$	$N_e = 4.4211$
$\mathbb{Z}[w]$ Constellation $1 \bmod 6$ with $p = 31$	$N_e = 5.0323$

Table 10: N_e Numerical Results for $1 \bmod 3$ Constellations in $\mathbb{Z}[w]$.

The results are consistent with the obtained decision regions for each constellation; constellations $1 \bmod 6$ in the ring $\mathbb{Z}[w]$ have a higher N_e value in comparison with the constellations in $\mathbb{Z}[i]$. This is due to the hexagonal shape of their decision regions which allows them to have a bigger average number of neighbors.

We plot the Nearest Neighbor Union Bound for primes $1 \bmod 6$ with $p = 7$, $p = 13$, $p = 19$ and $p = 31$. We also plot 8-PSK and 16-QAM as a reference.

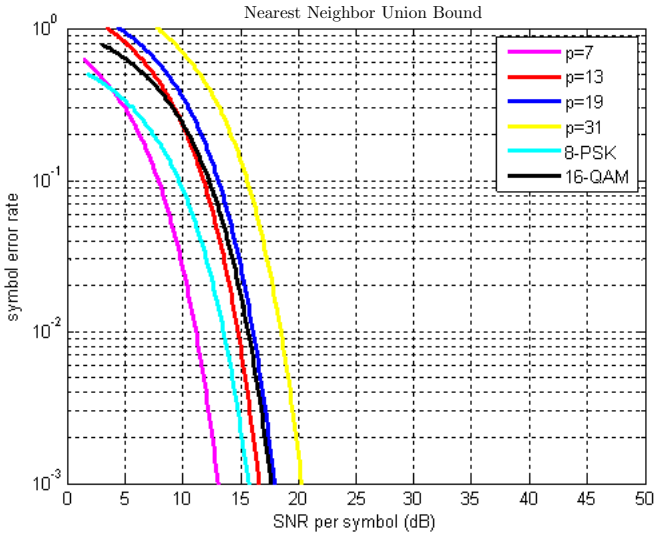


Figure 40: EISENSTEIN Constellations. Nearest Neighbor Union Bound for $1 \bmod 6$.

We can see that constellations $1 \bmod 6$ in $\mathbb{Z}[w]$ show consistently better results than common QAM. In fact, here we can appreciate something really interesting, for example, $p = 7$ constellation outperforms by far 8-PSK, we can observe an average 3dB difference, which is an important improvement considering there is just one point difference. Moreover, $p = 19$ presents almost the same behavior as 16-QAM, which means that we are able to send 3 more points using the same SNR. Therefore, constellations $1 \bmod 6$ in $\mathbb{Z}[w]$ show impressive results when compared to common QAM and will definitely improve the performance of the overall system.

We consider now the constellations $2 \bmod 3$ in $\mathbb{Z}[w]$.

First we are going to compute $d_{\min}$

$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 2$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 5$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 11$	$d_{\min} = 1$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 17$	$d_{\min} = 1$

Table 11: $d_{\min}$ Numerical Results for $\mathbb{Z}[w]$ Constellation $2 \bmod 3$.

We can see that the $d_{\min}$ value obtained for all the constellations in $\mathbb{Z}[w]$ remains the same with a value of $d_{\min} = 1$.

This can be understood as a consequence of how constellations in $\mathbb{Z}[w]$ are generated using the function modulo: the set of classes defining the points of a constellation $\mathcal{A}_1$ are included in the set of classes defining another constellation $\mathcal{A}_2$ with a larger dimension. Therefore if the first constellation achieves the minimum distance of the ring all the other constellations larger than it will have the same value of $d_{\min}$.

$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 2$	$N_e = 2.5$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 5$	$N_e = 4.48$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 11$	$N_e = 5.2893$
$\mathbb{Z}[w]$ Constellation $2 \bmod 3$ with $p = 17$	$N_e = 5.5363$

Table 12: N_e Numerical Results for $2 \bmod 3$ Constellations in $\mathbb{Z}[w]$.

Finally, we plot the Nearest Neighbor Union Bound with primes $2 \bmod 3$ in $\mathbb{Z}[w]$ with $p = 2$, $p = 5$, $p = 11$ and $p = 17$. We also plot 4-QAM, 16-QAM and 256-QAM as a reference.

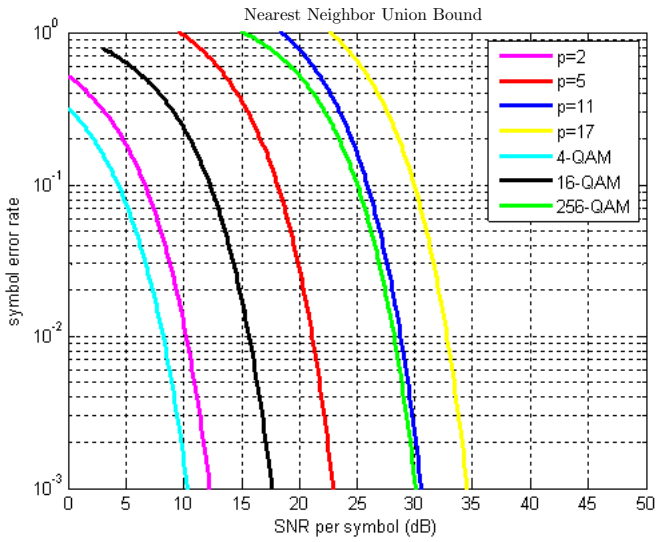


Figure 41: EISENSTEIN Constellations. Nearest Neighbor Union Bound for $2 \bmod 3$.

We can see that constellations $2 \bmod 3$ in $\mathbb{Z}[w]$ are worse than common QAM. We can appreciate for example that for $p = 2$ (4 points) the performance is worse than 4-QAM, where we can observe an average 2dB difference. For $p = 11$ (121 points) we can notice that there is just an average 0.5dB difference with 256-QAM. This implies

that QAM is able to send 256 points with the same overall performance as $p = 11$ (121 points) and therefore constellations $2 \bmod 3$ in $\mathbb{Z}[w]$ are not better than common QAM.

6 Conclusions and Further Work

We have studied in detail constellations over GAUSSIAN Integers and EISENSTEIN Integers. A theoretical introduction has been done as well as a description of the performance metrics used in the analysis. We have provided a basic implementation of the system model together with the necessary MATLAB codes to do the analysis.

As a conclusion of the proposed design, we have obtained good results using $1 \bmod 6$ constellation in the ring of $\mathbb{Z}[w]$, which is the best studied design of constellation. Moreover, $1 \bmod 4$ in $\mathbb{Z}[i]$ appears as a good alternative to QAM, with similar performance results. However, it is also important to note that common QAM constellations have better performance than $3 \bmod 4$ in $\mathbb{Z}[i]$ and $2 \bmod 3$ in $\mathbb{Z}[w]$.

As a further work, designing constellations over other types of integers could be proposed (for instance, HURWITZ Integers). Another important field of study is to consider different distances, not only the Euclidean distance, and how this would affect the probability of error. Moreover, a more refined implementation of the PNC scheme in Matlab would be of importance to do a more thorough analysis. Finally, the extension of these types of constellations to a more general system can be done.

7 Annex I

Implementation

We are going to implement the system model with Matlab (a useful primer of the language is Attaway [2012]).

We use a RAYLEIGH faded channel model with coefficients rounded to the nearest GAUSSIAN Integer, which can be generated using a distribution Gaussian both in the real and imaginary axis.

```
h=round((1/sqrt(2))*(randn(1,L)+j*randn(1,L)));
```

and circular symmetric complex GAUSSIAN noise $n \sim \mathcal{CN}(0, \sigma^2)$, where σ^2 is the noise power and can be calculated as:

$$\text{SNR} = \frac{\text{Average signal power}}{\text{Noise power}} \Rightarrow \text{Noise power} = \frac{\text{Average signal power}}{\text{SNR}} = \sigma^2.$$

We can calculate the SNR measured in dB's as

$$\begin{aligned} \text{SNR}|_{dB} &= 10 \cdot log(\text{SNR}_{Lineal}) \\ \text{SNR}_{Lineal} &= 10^{\frac{\text{SNR}|_{dB}}{10}}. \end{aligned}$$

The average signal power of the constellation can be calculated as

$$\text{Average signal power} = \frac{1}{p} \sum_{c=1}^{p} x_c x_c^* \tag{34}$$

and therefore

$$\text{Noise power} = \frac{\frac{1}{p} \sum_{c=1}^{p} x_c x_c^*}{\text{SNR}_{Lineal}}. \tag{35}$$

Next, we generate the system model studied in the previous sections and we collect L times the $\hat{v}$ values in order to estimate w.

A really important step in the implementation is computing the inverse matrix A in modulo p.

First, we need to know if the output matrix A is invertible. A straightforward way is to compute its determinant and if the determinant is 0 or has multiple factors with the modulo then the matrix is not invertible.

In the case $det \neq 0 \bmod p$ we need to follow the next steps in order to compute properly the inverse matrix.

We have to compute the inverse element modulo p of the determinant in absolute value, using the extended Euclidean algorithm Wagon [1990], which can be done using the function greatest common divisor. Where we use the fact that if $det \neq 0 \bmod p$ the greatest common divisor between the determinant and p is either 1 or -1. The procedure can be understood using $gcd = u \cdot det + v \cdot p$, then $gcd \mod p = u \cdot det \mod p$, where we can see that u is the multiplicative inverse we are looking for (except for a unit factor).

Further, we need to calculate the adjoint matrix of A and multiply it by the sign of the determinant. Finally, we multiply the inverse modulo p of the determinant with the adjoint matrix, and do the modulo p.

Once we have the inverse matrix A modulo p, we are able to calculate $\hat{w}$.

Code

```
function [Perror,Perror_theory1, Perror_theory2]= system(p,pz,L)

wn=0:p-1;
iterations=500;
elements=[0:1:p-1];
residue_elements=elements-round(elements*conj(pz)/(p))*pz;
SNR_db = [0:1:50];
SNR = 10.^(SNR_db./10);
avg= sum(abs(residue_elements).^2)/p;
sigma2 = avg./SNR;

for m=1:length(SNR)
    derror=0;
    error2=0;
    for s=1:iterations

        %create index permutations
        v2 = randperm(size(wn,2));

        %permute w vector
        w2 = wn(v2);

        %choose L values
        w=w2(1:L);

        for r=1:L

            x=w-round(w*conj(pz)/(p))*pz;

            %E=sum(sqrt(x.*conj(x)))/L;

            h=round((1/sqrt(2))*(randn(1,L)+j*randn(1,L)));
            z = sqrt(sigma2(m)).*(1/sqrt(2))*(randn(1,1)+j*randn(1,1));
            y = sum(x.*h) + z;
```

```
            yml=round(y);
            phi = yml-round(yml*conj(pz)/(p))*pz;

            %Computing u and v
            [g,u,v]=CMPLX_GCD (pz, conj(pz));
            u = conj(g)*u;
            v = conj(g)*v;

            %Inverse
            invs = phi*(v*conj(pz))+conj(phi)*(u*pz);

            %modulo p
            muinvs(r) = mod(invs-round(invs/p)*p,p);

            %Coefficients Matrix A
            a = h-round(h*conj(pz)/(p))*pz;
            invsa = a*(v*conj(pz))+conj(a)*(u*pz);
            muinvsa = mod(invsa-round(invsa/p)*p,p);
            A(r,:)=muinvsa;

        end

        determinant=det(A);
        vdett=abs(determinant);

        if mod(round(vdett),p) ~= 0

            [g,u,v]=CMPLX_GCD (round(vdett),p);
            u=g*u;
            dettinvs=mod(round(u),p);
            adjunct=sign(determinant)*adj(A);
            D=dettinvs*adjunct;
            Ainvs=mod(round(D),p);

            finalw = mod(Ainvs * muinvs',p);

            if (length(find(finalw'~=w))>0)
                error2=error2+1;
            end
        else
            derror=derror+1;
        end
    end
    Perror(m)=(error2+derror)/(iterations*L);
end

t=0:1:length(SNR)-1;
P1=(1-(1/p));
for s=2:L
```

```
    P1=P1*(1-(1/p^s));
end
P11=1-P1;

PR=1-erf(1./(2.*sqrt(2.*sigma2)));
Perror_theory1=P11+L.*PR;
Perror_theory2=1-P1*(1-exp(-1./(8*(sigma2)))).^L;

end
```

8 Annex II

N_e Code

```
%constellation used
 x=[0,1,0,0,-1];
 y =[0,0,-1,1,0];

clear C
clear D
clear dimn
clear F

if length(x) == 4
    N_e=2;
else
[a,b]=voronoin([x(:) y(:)]);

for cc=1 : length(b)
    disp (b{cc});
end

C=[];
D=[];
for cc=1: length(b)
    B = cell2mat(b(i));
    if length(B)==2
        dimn(cc) = 1;
    else
        dimn(cc) = length(B);
    end

    for zz=2 : length(B)
        C=[C; [min(B(zz),B(zz-1)),max(B(zz),B(zz-1))]];
        if strmatch([min(B(zz),B(zz-1)),max(B(zz),B(zz-1))],D) < 1
            D = [D;[min(B(zz),B(zz-1)),max(B(zz),B(zz-1))]];
        end
    end
    if length(B)~=2
        C = [C; [min(B(1),B(end)),max(B(1),B(end))]];
        if strmatch([min(B(1),B(end)),max(B(1),B(end))],D) < 1
           D = [D;[min(B(1),B(end)),max(B(1),B(end))]];
        end
    end
end

F = zeros(length(x));

```

```
for k=1: length(D)
    v = strmatch(D(k,:),C);
    for zz=1:length(v)
        suma = 0;
        for hh =1:length(dimn)
            suma = suma + dimn(hh);
            if v(zz) <= suma
                t(zz) = hh;
                break
            end
        end
    end

    if(length(t)>2)
        for cc=2:length(t)
            d(cc-1)=sqrt((x(t(cc))-x(t(cc-1)))^2+(y(t(cc))-y(t(cc-1)))^2 );
        end
        d(length(t)) = sqrt((x(t(1))-x(t(end)))^2+(y(t(1))-y(t(end)))^2 );

        [BX,IX] = sort(d);
        if  IX(1)==length(t)
            t2=[t(end) t(1)];
        else
            t2=[t(IX(1)) t(IX(1)+1)];
        end

        if  IX(2)==length(t)
            t3=[t(end) t(1)];
        else
            t3=[t(IX(2)) t(IX(2)+1)];
        end

        for m=1:length(t2)
            for n =1:length(t2)
                if n~=m
                    F(t2(m),t2(n)) = 1;
                end
            end
        end
        for m=1:length(t3)
            for n =1:length(t3)
                if n~=m
                    F(t3(m),t3(n)) = 1;
                end
            end
        end
        clear t2
        clear t3
        clear d
```

```
    else
        for m=1:length(t)
            for n =1:length(t)
                if n~=m
                    F(t(m),t(n)) = 1;
                end
            end
        end
    end

    clear t
    clear v
end

%Ne result:
N_e = mean(sum(F,1))
end
```

$d_{\min}$ Code

```
function [minim]=dmin(x,y)

minim = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2);

for c=2:length(x)-1
    d = sqrt((x(c)-x(c+1))^2+(y(c)-y(c+1))^2);
    if d < minim
        minim = d;
    end
end
end
```

9 Annex III

Constellations

The constellations are implemented in Python (a useful primer of the language is Guttag [2013]).

$1 \bmod 6$ **Constellation in** $\mathbb{Z}[w]$

```
def EISENSTEIN_constellation(a,b,p):

    real = []

    imaginary = []

    for n in range(0,p):

        axr1 = (float(n*a)/p)
        axr2 = (float(n*b)/p)

        a1 = math.ceil(axr1)
        a11 = math.floor(axr1)

        b1 = math.ceil(axr2)
        b11 = math.floor(axr2)

        alpha = n -(axr1 + axr2*cmath.exp(-2j*math.pi/3))*(a+b*cmath.exp(2j*math.pi
            /3))

        alpha1 = n -(a1 + b1*cmath.exp(-2j*math.pi/3))*(a+b*cmath.exp(2j*math.pi/3))

        alpha2 = n -(a11 + b1*cmath.exp(-2j*math.pi/3))*(a+b*cmath.exp(2j*math.pi/3)
            )

        alpha3 = n -(a11 + b11*cmath.exp(-2j*math.pi/3))*(a+b*cmath.exp(2j*math.pi
            /3))

        alpha4 = n -(a1 + b11*cmath.exp(-2j*math.pi/3))*(a+b*cmath.exp(2j*math.pi/3)
            )

        number2 = alpha - alpha1

        alpha_final = alpha1

        for cc in [alpha2,alpha3,alpha4]:

            number1 = alpha - cc

            c1 = number1*number1.conjugate()

```

```
            c2 =  number2*number2.conjugate()

            if c1.real < c2.real:

                number2 = number1

                alpha_final = cc

        real.append(alpha_final.real)

        imaginary.append(alpha_final.imag)

##    pylab.scatter(real, imaginary, s=150)
##
##
##    pylab.title('EISENSTEIN Constellation with p ='+str(p))
##    pylab.xlabel('Real')
##    pylab.ylabel('Imaginary')
##
##    pylab.show()

    return (real,imaginary)
```

$1 \bmod 4$ **Constellation in** $\mathbb{Z}[i]$

```
def modulo_p(n,p,pi):

        z = n*pi.conjugate()/p

        alpha = n - complex(round(z.real),round(z.imag)) *pi

        return alpha


def Z_constellation(p, pi):

    real = []

    imaginary = []

    for c in range(0,p):

        alpha = modulo_p(c,p,pi)

        real.append(alpha.real)

        imaginary.append(alpha.imag)
```

```
23
24      print real
25
26      print imaginary
27
28  ##    pylab.scatter(real, imaginary, s=150)
29  ##
30  ##    pylab.title('Constellation with p ='+str(p))
31  ##    pylab.xlabel('Real')
32  ##    pylab.ylabel('Imaginary')
33  ##
34  ##
35  ##    pylab.show()
36
37      return (real,imaginary)
```

$3 \bmod 4$ Constellation in $\mathbb{Z}[i]$

```
1   def constellation_3mod4(p):
2
3       real = []
4
5       imaginary = []
6
7       for z in range(0,p):
8
9           for k in range(0,p):
10
11              real.append(z)
12
13              imaginary.append(k)
14
15  ##    print real
16  ##
17  ##    print imaginary
18  ##
19  ##    pylab.scatter(real, imaginary, s=50)
20  ##
21  ##    pylab.title('Constellation with p ='+str(p))
22  ##    pylab.xlabel('Real')
23  ##    pylab.ylabel('Imaginary')
24  ##
25  ##
26  ##    pylab.show()
27
28      return (real,imaginary)
```

$2 \bmod 3$ Constellation in $\mathbb{Z}[w]$

```
def EISENSTEIN_2mod3(p):

    real = []

    imaginary = []

    for f in range(0,p):

        for k in range(0,p):

            n =k*cmath.exp(2j*math.pi/3)+f

            real.append(n.real)

            imaginary.append(n.imag)

##    print real
##
##    print imaginary
##
##    pylab.scatter(real, imaginary, s=50)
##
##    pylab.title('Constellation with p ='+str(p))
##    pylab.xlabel('Real')
##    pylab.ylabel('Imaginary')
##
##
##    pylab.show()

    return (real,imaginary)
```

QAM Square Constellations

```
def QAM(n):

    real = []

    imaginary = []

    k = []

    for c in range(0,2**(n/2)):

        k.append(c)

```

```
    for z in k:

        real.append(int(-2**(n/2.)+1+2*z))

        imaginary.append(int(-2**(n/2.)+1+2*z))

    r=[]
    imy=[]

    lista = [(x,y) for x in real for y in imaginary]

    for c in range(0,len(lista)):

            r.append(lista[c][0])

            imy.append(lista[c][1])

    for c in range(0,n+1,2):

            # M-QAM Square
            pylab.plot([-(-2**(c/2)+1),(-2**(c/2)+1),(-2**(c/2)+1),-(-2**(c/2)+1)
                ,-(-2**(c/2)+1)],[-(-2**(c/2)+1),-(-2**(c/2)+1),(-2**(c/2)+1),(-2**(
                c/2)+1),-(-2**(c/2)+1)], linewidth =1.7)


    pylab.scatter(r, imy, s=50)


##    pylab.title(str(n*n)+'-QAM Constellation')
    pylab.title('64-QAM Constellation')
    pylab.xlabel('Real')
    pylab.ylabel('Imaginary')

    pylab.show()

    return (r,imy)
```

PSK Constellations

```
def PSK(M):

        real =[]
        imaginary =[]

        for c in range(1,M+1):

               k = math.e**(2j*c*math.pi/M)
```

```
9
10                  real.append(k.real)
11                  imaginary.append(k.imag)
12
13
14          pylab.scatter(real, imaginary, s=300)
15
16
17          for c in range(0,M):
18
19                  pylab.plot([0,real[c]],[0,imaginary[c]], linewidth =1.7)
20
21
22          pylab.title(str(M)+'-PSK Constellation')
23          pylab.xlabel('Real')
24          pylab.ylabel('Imaginary')
25
26          pylab.show()
27
28          return (real,imaginary)
```

10 Annex IV

The code of the Figure 5 has been extracted from http://rosettacode.org/wiki/Voronoi_diagram and modified in order to show the points and different distances. The final version of the code after these modifications is the next:

```
from PIL import Image
import random
import math
import ImageDraw

def generate_diagram_voronoi(width, height, num_cells):

        image = Image.new("RGB", (width, height))
        draw = ImageDraw.ImageDraw(image)

        putpixel = image.putpixel
        imgx, imgy = image.size
        nx = []
        ny = []
        nr = []
        ng = []
        nb = []

        for c in range(num_cells):
                nx.append(random.randrange(imgx))
                ny.append(random.randrange(imgy))
                nr.append(random.randrange(256))
                ng.append(random.randrange(256))
                nb.append(random.randrange(256))

        for y in range(imgy):
                for x in range(imgx):
                        dmin = math.hypot(imgx-1, imgy-1)
                        z = -1

                        for c in range(num_cells):
                                #d = math.hypot(nx[c]-x, ny[c]-y)
                                d = abs(nx[c]-x)+abs(ny[c]-y)

                                if d < dmin:
                                        dmin = d
                                        z = c

                                draw.polygon([(nx[c]+3, ny[c]+3), (nx[c]-3,ny[c]+3),
                                     (nx[c]-3,ny[c]-3), (nx[c]+3,ny[c]-3)], fill="
                                    red", outline="green")

                        putpixel((x, y), (nr[z], ng[z], nb[z]))
```

```
42
43          #image.save("Diagram_Euclidean_Voronoi.png", "PNG")
44          image.save("Diagram_Manhattan_Voronoi.png", "PNG")
45
46  #generate_diagram_voronoi(500, 500, 25)
47  generate_diagram_voronoi(500, 500, 25)
```

References

R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung. 2000. Network Information Flow. IEEE Transactions on Information Theory. 16

H. M. Asif, E. Gabidulin, and B. Honary. 2012. Rank Codes over GAUSSIAN Integers and Space Time Block Codes. Mathematics of Distances and Applications. 20

S. Attaway. 2012. *Matlab: A Practical Introduction to Programming and Problem Solving*. Elsevier. 78

J. Burkardt. 2013. Determining VORONOI Neighbors using Matlab's voronoin Command. Florida State. 39

J. M. Cioffi. 2013. Signal Processing and Detection. Stanford. 33

K. Conrad. 2013. Expository Papers. University of Connecticut. 20

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. *Introduction to Algorithms*. MIT Press. 39

J. D. Curtó and M. A. Vázquez. 2013. Construction and Performance of Network Codes. Universitat Autònoma de Barcelona. 17, 20, 48, 51

J. D. Curtó, I. C. Zarza, and M. A. Vázquez. 2012. Secure Network Coding: Overview and State-of-the-art. Universitat Autònoma de Barcelona. 16

A. Goldsmith. 2005. *Wireless Communications*. Cambridge. 32, 33

S. Gupta and M. A. Vázquez. 2012. Compute and Forward: End to End Performance over Residue Class Based Signal Constellation. arXiv:1212.3289. 17, 20, 48, 51

J. V. Guttag. 2013. *Introduction to Computation and Programming Using Python*. MIT Press. 86

K. Huber. 1994a. Codes over EISENSTEIN-JACOBI Integers. Contemporary Mathematics. 24, 65, 67

K. Huber. 1994b. Codes over GAUSSIAN Integers. IEEE Transactions on Information Theory. 20, 48, 56, 57

K. Huber. 2004. Decoding of Icyclic Codes for the MANNHEIM Metric. ISITA. 65

U. Madhow. 2008. *Fundamentals of Digital Communication*. Cambridge. 32, 43

B. Nazer and M. Gastpar. 2011. Compute and Forward: Harnessing Interference through Structured Codes. IEEE Transactions on Information Theory. 20, 51

P. Prandoni and M. Vetterli. 2008. *Signal Processing for Communications*. EPFL Press. 41

J. Proakis. 2001. *Digital Communications*. McGraw-Hill. 43

J. Stillwell. 2003. *Elements of Number Theory*. Springer. 20

S. Wagon. 1990. The Euclidean Algorithm Strikes Again. American Mathematical Monthly. 79

W. C. Waterhouse. 1987. How Often Do Determinants over Finite Fields Vanish? Discrete Mathematics. 33

J. M. Wozencraft and I. M. Jacobs. 1965. *Principles of Communication Engineering*. Wiley. 33, 34

S. Zhang, S. C. Liew, and P. P. Lam. 2006. Hot Topic: Physical-layer Network Coding. ACM MobiCom. 16

Resum:

L'objectiu principal d'aquest treball és el disseny de constel·lacions per a un sistema de comunicacions basat en Physical-layer Network Coding. El disseny es durà a terme dins de dos anells commutatius: els GAUSSIAN Integers i els EISENSTEIN Integers.

Usant com a punt de partida aquest sistema concret, primer hem identificat la teoria necessària per al disseny, així com les 'performance metrics' d'utilitat per analitzar les constel·lacions proposades. Tot seguit hem presentat les constel·lacions més usades avui en dia per utilitzar-les com a referència i finalment hem proposat una metodologia de disseny, a partir de la qual s'han proposat quatre constel·lacions.

Per últim, s'ha dut a terme l'anàlisi dels resultats obtinguts així com la implementació del sistema estudiat de Physical-layer Network Coding.

Resumen:

El objetivo principal de este trabajo es el diseño de constelaciones para un sistema de comunicaciones basado en Physical-layer Network Coding. El diseño se llevará a cabo dentro de dos anillos commutativos: los GAUSSIAN Integers y los EISENSTEIN Integers.

Usando como punto de partida este sistema concreto, primero hemos identificado la teoría necesaria para el diseño, así como las 'performance metrics' de utilidad para analizar las constelaciones propuestas. A continuación hemos presentado las constelaciones más usadas hoy en día para utilizarlas como referencia y finalmente hemos propuesto una metodología de diseño, a partir de la cual se han propuesto cuatro constelaciones.

Por último, se ha realizado el análisis de los resultados obtenidos, así como la implementación del sistema estudiado de Physical-layer Network Coding.

Summary:

The main goal of this work is to design constellations in a system of Physical-layer Network Coding. The design is based on constellations over two commutative rings: GAUSSIAN Integers and EISENSTEIN Integers.

Using this particular system as a working base, first we have identified the needed theory in the design, as well as the needed performance metrics in order to analyse the proposed constellations. Then, we have presented the most used constellations nowadays in order to be used as a comparison. Finally, we have proposed a design methodology and four constellations.

Further, the obtained results have been analysed and the particular studied system of Physical-layer Network Coding has been implemented.

I. C. Zarza

General information

Legal Name: DE ZARZA I CUBERO Irene.
e-mail: z@dezarza.ch webpage: http://dezarza.ch

Education

Universitat Autònoma de Barcelona (UAB). Barcelona.

Degree in Mathematics. 2011 - 2013.
Minor in Pure Mathematics.
Faculty of Sciences.

Thesis: Physical-layer Network Coding: Design of Constellations over Rings.
Grade: Excellent. First Class with Distinction.

Universitat de Barcelona (UB). Barcelona.

Mathematics, Licentiate. First Cycle. 2007 - 2011.
Department of Mathematics and Computer Science.

University Entrance Examination.
Average Grade: 8.93/10. First Class with Distinction.

Academic Distinctions:

- First year scholarship for university studies. Ministry of Education.
  This award is given to the top nationwide first year university students.
- First year scholarship for university studies. Caixa Manresa.
  This award is given to the top university entrance examination average grades in the region of Catalunya.

Technological Baccalaureate. 2005 - 2007.
Average Grade: 10/10. First Class Degree and Honorary Scholarship.

Academic Distinctions:

- Outstanding Thesis of Research: Squaring the Circle. Study of the different mathematical approaches to solve the ancient problem of the quadrature of the circle.
- Outstanding Curriculum.

Publications

Curtó, **Zarza** and Vázquez.
Secure Network Coding: Overview and State-of-the-art.
Universitat Autònoma de Barcelona. Bellaterra. 2012.
http://blogs.uab.cat/zarza/files/2019/05/snc_decurto12.pdf

Dissertation

Degree in Mathematics.
Physical-layer Network Coding: Design of Constellations over Rings.
Supervisors: Vázquez and Mondelo.
Universitat Autònoma de Barcelona. Bellaterra. 2013.
http://blogs.uab.cat/zarza/files/2019/05/pfc_dezarza.pdf
http://blogs.uab.cat/zarza/files/2019/05/slides_pfc_dezarza.pdf

Languages

English -

TOEFL Internet Based test. 11-12-2016. **Score 102/120.**

Career

Teaching Assistant. Institut d'Educació Secundària Joan Coromines. Barcelona. 2005 - 2006.

Services

First European Training School in Network Coding: Random Network Coding and Designs over $GF(q)$. Universitat Autònoma de Barcelona (UAB). Barcelona. 4 - 8 February 2013.

> From designs over $GF(q)$ to applications of networking: a cross-road for mathematics, computer science and engineering.
>
> Attendee and Volunteer.

Extracurricular Activities

Symposium of the Royal Society of Mathematics, The Millennium Problems. Awarded with an assistance grant by Institut de Matemàtica de la Universitat de Barcelona (UB). Barcelona. 1 - 3 June 2011.

Course in Investment and Financial Markets. Technical Analysis and Risk Management. Barcelona. 23 - 26 May 2011.

Course in Investment and Financial Markets. Barcelona. 18 - 21 April 2011.

Competition of Entrepreneurship. EMPRÈN UPC. 1st Edition. Universitat Politècnica de Catalunya (UPC). Finalist project awarded with honorable mention and 1000 euros. Barcelona. 14 March 2011 - 14 June 2011.

Programming

C, C++, Python, MATLAB and Prolog.

Software

LaTeX, R, Maple, Mathematica and EViews.

DE ZARZA I CUBERO Irene.
Bellaterra, 2013.