Construction and Performance of Network Codes.

Thesis:
5-year Degree in
Engineering of Telecommunication.

Author: J. de Curtò.
Advisor: M. A. Vázquez.

Cerdanyola del Vallès (Barcelona), 2013.

UAB

Universitat Autònoma de Barcelona

UAB

Projecte Fi de Carrera.

Enginyeria de Telecomunicació.

Construction and Performance of Network Codes.

J. de Curtò i DíAz.

Directora: María Ángeles Vázquez i Castro.

Departament de Telecomunicació i Enginyeria de Sistemes.

Escola Tècnica Superior d'Enginyeria (ETSE).
Universitat Autònoma de Barcelona.

Cerdanyola del Vallès (Barcelona), 2013.

UAB

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

Certifica:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumne DE CURTÒ I DÍAZ Joaquim.

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

Cerdanyola del Vallès (Barcelona), 2013.

Signatura: Vázquez.

Construction and Performance of Network Codes.

J. de Curtò i DíAz

Cerdanyola del Vallès (Barcelona), 2013.

Dedicated to De Zarzà i Cubero.

Contents

List of Figures

1 Rationale and Objectives

The aim of this project is to implement and provide a theoretical description of different schemes of Physical-layer Network Coding. We will first introduce a basic scheme and extend the given system with increasing complexity. Lattice-based network codes will be used. The theoretical tools needed to construct the system will be provided as well as the performance analysis and comparison.

MATLAB language programming has been used throughout the project. The plotted output results have been enclosed and analyzed. Further, flow chart diagrams of each one of the system codes have been attached.

2   Introduction

In the last years, the number of wireless devices has skyrocketed and, to handle the demands of ever richer multimedia applications, these devices have required higher and higher data rates. These trends, coupled with the scarcity of spectrum, mean that interference between devices will be one of the dominant bottlenecks in wireless networking for the years to come. In many cases, this interference is purely an obstacle to communication. However, in many scenarios, it is actually possible to harness interference to enable more efficient communication over a network. In this project, we are going to focus on a set of novel strategies geared at exploiting wireless interference.

Nodes in a network can have different roles, sources transmit information packets into the network, destinations recover a set of packets, and relays help to move the information between sources and destinations. In a classical wired network, relays have the only functioning of forwarding a set of packets towards the destinations. For a wired network, multiple relays and a destination, this routing strategy is optimal. However, more generally, routing cannot attain maximum throughput and relays need to combine packets using functions, rather than just forwarding. This strategy is known as Network Coding, and was first proposed by Ahlswede et al. [2000]. An overview of Secure Network Coding can be found in de Curtò et al. [2012].

In a wireless setting, transmitting a packet from one node to another causes interference to all nearby nodes. If multiple nodes transmit concurrently, the electromagnetic waves are linearly superimposed, which makes it harder for a receiver to recover the desired packets. Yet, for network coding, relays do not need to recover the contents of individual packets, only an appropriate function of them. This strategy of using the operation of network coding that comes naturally in wireless communications is known as Physical-layer Network Coding, and would be the common framework of this work.

3 Physical-layer Network Coding

The concept of Physical-layer Network Coding (PNC) was originally proposed in Zhang et al. [2006] as a way to exploit the operation of network coding that occurs naturally in superimposed electromagnetic (EM) waves. It is a simple fact in physics that when multiple EM waves come together within the same physical space, they add. This additive mixing of EM waves is a form of network coding, performed by nature. Alternatively, the additive operation of network coding can be transformed and mapped to other forms of network coding after reception. Exploiting these facts turns out to have profound and fundamental ramifications.

In many wireless communication networks today, interference is treated as a destructive phenomenon. When multiple transmitters send radio waves to their respective receivers, each one receives signals from its transmitter as well as from the others. The radio waves from the other transmitters are often treated as interference that corrupts the intended signal. In Wi-Fi networks, for example, when multiple nodes transmit together, packet collisions occur and none of the packets can be received correctly.

As originally proposed in Zhang et al. [2006], Physical-layer Network Coding was an attempt to turn the situation around. By exploiting the operation of network coding performed by nature, the interference could be embraced rather than rejected. For instance, by allowing two end nodes to transmit simultaneously to the relay and not treating this as collision, Physical-layer Network Coding can boost the system throughput.

4 PNC Preliminaries

The key insight is that the modulation and coding strategies should share a common algebraic structure across transmitters. More precisely, if the transmitter waveforms are points of a lattice, then every integer combinations of these waveforms is itself a point of the same lattice. So, receivers can decode these linear combinations with the same framework used to decode individual packets. How efficiently depends on how closely the coefficients of the desired linear combination match the observed channel strengths and phases.

To make the ideas behind Physical-layer Network Coding apparent, we need to develop network coding slightly more formally.

We will consider operations on a finite field $\mathbb{F}_q$, that is to say, a field with q elements that will be denoted by $\{0, 1, 2, \ldots, q-1\}$. We will assume that q is a prime number so that addition and multiplication over the finite field can be written as modulo addition and multiplication over the reals. For any two integers a and b in this set, we will denote addition and multiplication modulo q as

$$a \oplus b = [a+b] \bmod q$$
$$a \otimes b = [ab] \bmod q.$$

The transmitting terminal has a message that can be represented as a string of bits. This message can be broken into several packets each of which can be written as a length-k vector of elements from the finite field that we will denote $\mathbf{w}_l \in \mathbb{F}_q^k$. Say a relay in a network has received some of these packets $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L$. The relay in network coding sends a linear combination $\mathbf{v}$ of these packets towards the destination

$$\mathbf{v} = a_1 \mathbf{w}_1 \oplus a_2 \mathbf{w}_2 \oplus \ldots \oplus a_L \mathbf{w}_L$$

where $a_1, a_2, \ldots, a_L$ are coefficients over the finite field.

The goal is for each destination to collect enough linear combinations to infer the original packets. Assume a destination has successfully received linear combinations $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M$ where

$$\mathbf{v}_M = a_{m1} \mathbf{w}_1 \oplus a_{m2} \mathbf{w}_2 \oplus \ldots \oplus a_{mL} \mathbf{w}_L.$$

Then, it can solve for the original packets if the matrix of coefficients

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1L} \\ a_{21} & a_{22} & \ldots & a_{2L} \\ \vdots & \vdots & & \vdots \\ a_{M1} & a_{M2} & \ldots & a_{ML} \end{bmatrix} \tag{1}$$

has rank L. There are different strategies to find these a coefficients depending on the particular PNC scheme used. We will study and implement different approaches to generate this matrix in the last sections of this project.

We are going to see a PNC example
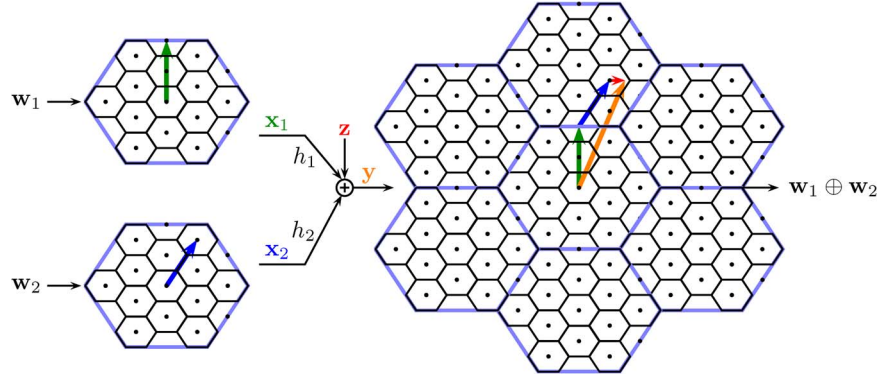
Figure 1: PNC Example.

Example 4.1. *Each source maps its finite-field message into an element of a lattice codebook and sends this vector over the channel. In this example, channel coefficients are taken $h_1 = h_2 = 1$. The receiver observes a noisy sum of the transmitted vectors and determines the closest lattice point. After taking a modulo operation the receiver can invert the mapping and determine the modulo sum of the original messages. After adequate collecting of $L = 2$ linear independent combinations, the original messages can be obtained.*

5 MAC Relay System Mode

The simplest cooperative relaying network consists of three nodes, namely source, destination, and a third node supporting the direct communication between source and destination denoted as relay. If the direct transmission of a message from source to destination is not (fully) successful, the overheard information from the source is forwarded by the relay to reach the destination via a different path. Since the two communications took a different path and take place one after another, this example implements the concept of space diversity and time diversity.

The relaying strategies can be further distinguished by the Amplify and Forward, Decode and Forward, Compress and Forward and Compute and Forward:

- The strategy Amplify and Forward allows the relay station to amplify the received signal from the source node and to forward it to the destination station.

- Relays following the strategy Decode and Forward overhear transmissions from the source, decode them and in case of correct decoding, forward them to the destination. Whenever unrecoverable errors reside in the overheard transmission, the relay can not contribute to the cooperative transmission.

- The strategy Compress and Forward allows the relay station to compress the received signal from the source node and forward it to the destination without decoding the signal where WYNER-ZIV coding can be used for optimal compression.

- The strategy Compute and Forward consists on employing a lattice codebook so that integer combinations of codewords are themselves codewords. Relays are then free to select integer coefficients that match the channel coefficients as closely as possible, thus reducing the effective noise and increasing the achievable rates. A relay can employ successive interference cancellation to remove decoded codewords from the observations of the channel. This decreases the effective noise encountered in the next step of decoding.

In this project, we are going to focus on Compute and Forward (C&F), first proposed in Nazer and Gastpar [2011a]. This novel scheme uses structured nested lattice codes. The transmitter signals are lattice points in a multidimensional lattice over integers. Based on transmitted signals, the relay decodes and forwards an integer valued linear combination of transmitter signals to maximize computation rate. For the Compute and Forward of Nazer and Gastpar, algorithms are designed in Wei and Chen [2012a] and Wei and Chen [2012b] to find optimal vectors of coefficients in terms of maximizing the transmission rate.

In this project, we will first focus on the study of the C&F uncoded scalar system, next we will proceed to implement a vectorial version of the system. The next step will be to use a HAMMING q-ary (6,4) to implement a C&F coded system. Further, we will try to improve the coefficient matrix $\mathbf{A}$ with a step by step approach: first we will do a first approach using an easy idea to improve the coefficients, then we will implement the optimal algorithm proposed in the literature and finally we will extend this optimum algorithm with an easy yet powerful idea.

6 Mathematical Tools

First, we are going to do an introduction to lattices and lattice network coding. Next, a survey of GAUSSIAN Integers (Conrad [2013]) will be presented. The theory will be interspaced with examples to help understanding. Further, two key concepts to understand the system model under study will be explained: Euclid's Algorithm and BÉZOUT Theorem. Following, the question towards what are the primes in the lattice $\mathbb{Z}[i]$ will be answered. Finally, a brief introduction to rings, fields and ideals will be done.

6.1 Introduction to Lattices

The concept of lattice comes from the geometry of numbers from the work of Minkowski [1896, 1907]. As its name suggests, the geometry of number relates to both geometry and arithmetic numbers. It is concerned with the relationship between CONVEX sets and integer points in a n-dimensional space.
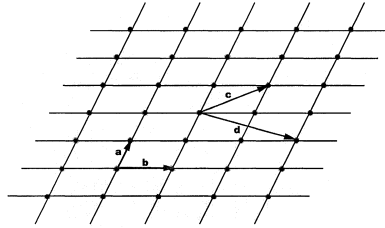
Figure 2: 2 Dimensional Lattice.

Geometrically, a lattice can be viewed as the set of intersection points of an infinite grid. One can shift any point onto any other by some shifting of the arrangement. The lines of the grid do not need to be orthogonal to each other. Lattices are powerful tools to solve many complex problems in mathematics and computer science.

A lattice is usually specified by a basis, that is to say, a set of linearly independent vectors such that any lattice point can be obtained as an integer linear combination of the basis vectors. It is obvious that the same lattice may have many different basis.

Particularly, suppose that a given matrix $B = [\mathbf{b}_1 \quad \ldots \quad \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ has full column rank, then the set

$$\mathcal{L}(B) = B\mathbf{x} : \mathbf{x} \in \mathbb{Z}^n$$

is referred to as the lattice generated by B, the set $S = \mathbf{b}_1 \quad \ldots \quad \mathbf{b}_n$ is referred to as the lattice basis, and B is referred to as the lattice basis matrix. The dimension of the lattice is said to be n. Suppose $\overline{B} = [\overline{\mathbf{b}}_1 \quad \ldots \quad \overline{\mathbf{b}}_n] \in \mathbb{R}^{m \times n}$ has full column rank. If $\mathcal{L}(B) = \mathcal{L}(\overline{B})$ are equivalent. Two basis matrices $B, \overline{B} \in \mathbb{R}^{m \times n}$ are equivalent if and only if there exists a unimodular matrix $Z \in \mathbb{Z}^{n \times n}$ (an integer matrix with determinant $det(Z) = \pm 1$) such that $\overline{B} = BZ$.

Complex R-lattices are natural generalizations of real lattices. Let R be a discrete subring of $\mathbb{C}$ forming a principle ideal domain (PID). Typical examples include the GAUSSIAN Integers $\mathbb{Z}[i]$ and EISENSTEIN Integers $\mathbb{Z}[w]$. A R-lattice $\Lambda \in \mathbb{C}^n$ is a discrete R-submodule of $\mathbb{C}^n$, consisting of all R-linear combinations of a set of basis vectors.

6.2 Lattice Network Coding

Most of the material in this subsection can be found in Sun et al. [2013].

The C&F scheme based on nested lattice codes was first proposed in Nazer and Gastpar [2011a]. Later a more general algebraic model, called lattice network coding, was developed in Feng et al. [2011]. In the following, we give a brief review of basic concepts of lattice network codes.

Definition 1. *Let R be a Principal Ideal Domain (PID), which is a commutative ring such that:*

(1) for all $a, b \in R$, $ab = 0$ if, and only if, either $a = 0$ or $b = 0$;

(2) every ideal [1] in R can be written as $aR = \{ar : r \in R\}$ for some $a \in R$.

Well known PIDs in $\mathbb{C}$ include the ring of integers $\mathbb{Z}$ and the ring of GAUSSIAN Integers $\mathbb{Z}[i] = \{a+bi : a, b \in \mathbb{Z}\}$.

Definition 2. *Let $N \leq n$. A subset Λ of $\mathbb{C}^n$ is called a N-dimensional R-lattice if it forms a R-module of rank N, that is, Λ is closed under addition and under multiplication by scalars in the ring R, and there are N linearly independent vector $\mathbf{b}_1, \ldots, \mathbf{b}_N \in \Lambda$ such that $\Lambda = \{\sum_{1 \leq c \leq N} r_c \mathbf{b}_c : r_c \in R \quad \forall c\}$. A subset Λ' of Λ is called a sublattice of Λ if it is a R-module.*

Given a R-lattice Λ and a sublattice Λ' of Λ, the quotient group $\Lambda/\Lambda' = \{\lambda + \Lambda' : \lambda \in \Lambda\}$ naturally forms a partition of Λ. For a Lattice Network Code, the message space is $W = \Lambda/\Lambda'$, which can also be regarded as a R-module. As an example, consider the PID of $\mathbb{Z}$, which itself can be regarded as a 1-dimensional $\mathbb{Z}$-lattice. Every integer corresponds to a lattice point. The set $2\mathbb{Z}$ of even integers forms a sublattice of $\mathbb{Z}$, but the set of odd integers is not a sublattice of $\mathbb{Z}$ since it is not closed under multiplication by an even integer. The quotient group $\mathbb{Z}/2\mathbb{Z}$ forms a partition of $\mathbb{Z}$ into two sets of lattice points, the set of even integers and the set of odd integers.

6.3 The Lattice of GAUSSIAN Integers $\mathbb{Z}[i]$

GAUSSIAN Integers 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 $\mathbb{Z}$ size is measured using the absolute value. In $\mathbb{Z}[i]$, we use the norm.

Definition 3. *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 on $\mathbb{Z}[i]$ instead of absolute values on $\mathbb{Z}[i]$ is that norms are integers (rather than square roots) and the divisibility properties of norms in $\mathbb{Z}$ will provide important information about divisibility properties in $\mathbb{Z}[i]$.

---

[1] An ideal in a commutative ring R means a set of elements in R that is closed under addition and under multiplication by an arbitrary element in R

The only GAUSSIAN Integers which are invertible in $\mathbb{Z}[i]$ are ± 1 and $\pm i$, this is a corollary of the theorem: the norm is multiplicative[2]. Invertible elements are called units.

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

Theorem 1. *(**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 $N(\rho) < N(\beta)$. In fact, we can choose ρ so $N(\rho) \leq (1/2)N(\beta)$.*

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

---

[2]The norm is multiplicative: for α and β in $\mathbb{Z}[i]$, $N(\alpha\beta) = N(\alpha)N(\beta)$.

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 6.2. *We now 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).

6.4   Euclid's Algorithm

We begin by defining greatest common divisors in $\mathbb{Z}[i]$.

Definition 5. *For non-zero α and β in $\mathbb{Z}[i]$, 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 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 $-r$, ir and $-ir$. Therefore, we can speak about a greatest common divisor, but not the greatest common divisor.

Definition 6. *When α and β only have unit factors in common, we call them relatively prime.*

Theorem 2. *(Euclid's Algorithm). Let $\alpha, \beta \in \mathbb{Z}[i]$ be non-zero. Recursively apply 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, provided the remainder is not zero:*

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

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 1. *For non-zero α and β in $\mathbb{Z}[i]$, 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.*

Example 6.3. *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 6.4. *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 get 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: $-1 + 2i = (-1)(1 - 2i)$.

6.5   BÉZOUT Theorem

In $\mathbb{Z}$, BÉZOUT Theorem says that for any non-zero a and b in $\mathbb{Z}$ the greatest common divisor can be expressed as $gcd(a, b) = ax + by$ 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 gives us BÉZOUT Theorem there.

Theorem 3. *(BÉZOUT Theorem). Let δ be any greatest common divisor of two non-zero GAUSSIAN Integers α and β. Then $\delta = \alpha x + \beta y$ for some $x, y \in \mathbb{Z}[i]$.*

Corollary 2. *The non-zero GAUSSIAN Integers α and β are relatively prime if and only if we can write*

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

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

Example 6.5. *We saw in Example (6.3) 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:*

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

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

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

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

We will define composite and prime GAUSSIAN Integers.

Lemma 1. *For $\alpha \neq 0$, any divisor of α whose norm is 1 or $N(\alpha)$ is a unit or is a unit multiple of α.*

This lemma is not saying the only GAUSSIAN Integers whose norm is $N(\alpha)$ are $\pm\alpha$ and $\pm i\alpha$. For instance $1 + 8i$ and $4 + 7i$ both have norm 65 and neither is a unit multiple of the other. What this lemma is really saying is that the only GAUSSIAN Integers which divide α and have norm equal to $N(\alpha)$ are $\pm\alpha$ and $\pm i\alpha$.

When $N(\alpha) > 1$, there are always eight obvious factors of α: ± 1, $\pm i$, $\pm\alpha$ and $\pm i\alpha$. We call these the trivial factors of α. (analogous to the four trivial factors ± 1 and $\pm n$ of any integer n with $|n| > 1$). Any other factor of α is called non-trivial.

Definition 7. *Let α be a GAUSSIAN Integer with $N(\alpha) > 1$. We call α composite if it has a non-trivial factor. If α only has trivial factors, we call α prime.*

For example, a trivial factorization of $7 + i$ is $i(1 - 7i)$. A non-trivial factorization of $7 + i$ is $(1 - 2i)(1 + 3i)$. A non-trivial factorization of 5 is $(1 + 2i)(1 - 2i)$, it can be observed that 5 is prime in $\mathbb{Z}$ but it is composite in $\mathbb{Z}[i]$.

Theorem 4. *If the norm of a GAUSSIAN Integer is prime in $\mathbb{Z}$, then the GAUSSIAN Integer is prime in $\mathbb{Z}[i]$.*

For example, $N(4 + 5i) = 41$, $4 + 5i$ is prime in $\mathbb{Z}[i]$. Doing the same procedure, $4 - 5i$ is also prime, as are for example $1 \pm i$ or $1 \pm 2i$.

Theorem 5. *A prime p in $\mathbb{Z}^+$ is composite in $\mathbb{Z}[i]$ if and only if it is a sum of two squares.*

Therefore, any prime p in $\mathbb{Z}^+$ which is not a sum of two squares is not composite in $\mathbb{Z}[i]$, so it stays prime in $\mathbb{Z}[i]$ (for example 3, 7, 11 and 19).

The first primes in $\mathbb{Z}^+$ which are the sum of two squares are 2, 5 and 13:

$$
\begin{aligned}
2 &= 1^2 + 1^2 \\
5 &= 1^2 + 2^2 \\
13 &= 2^2 + 3^2.
\end{aligned}
$$

Therefore each of these primes is composite in $\mathbb{Z}[i]$. The factorization of 2 is special, since its prime factors are unit multiples of each other $1 - i = i(1 + i)$:

$$
2 = -i(2 + i)^2.
$$

Corollary 3. *If a prime p in $\mathbb{Z}^+$ is composite, and $p \neq 2$, then up to unit multiple p has exactly two GAUSSIAN prime factors, which are conjugate and have norm p.*

Corollary 4. *If a prime p in $\mathbb{Z}^+$ satisfies $p \equiv 3 \bmod 4$, then it is not a sum of two squares in $\mathbb{Z}$ and it stays prime in $\mathbb{Z}[i]$.*

We can summarize the factorization of primes in $\mathbb{Z}^+$ into GAUSSIAN prime factors.

Theorem 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^* which are not unit multiples.*

- *if $p = 3 \bmod 4$ then p stays prime in $\mathbb{Z}[i]$.*

Example 6.6. *The prime 61 satisfies $61 \equiv 1 \bmod 4$, so 61 has two conjugate GAUSSIAN prime factors. Since $61 = 5^2 + 6^2$, $61 = (5 + 6i)(5 - 6i)$.*

6.7 Rings, Fields and Ideals

Definition 8. *A ring is a set R with two operations called addition and multiplication, such that the following axioms hold for every $a, b, c \in R$:*

- *Addition is associative: $a + (b + c) = (a + b) + c$.*

- *Addition is commutative: $a + b = b + a$.*

- *Zero is neutral for addition $a + 0 = a$.*

- *a has an opposite $-a$ (in R) such that $a + (-a) = 0$.*

- *Multiplication is associative: $a(bc) = (ab)c$.*

- *The element 1 is neutral for multiplication: $1a = a = a1$.*

- *Multiplication distributes across addition: $a(b+c) = ab + ac$ and $(a+b)c = ac + bc$.*

A commutative ring is a ring which also satisfies the law: $ab = ba$ for all $a, b \in R$.

Definition 9. *A field is a commutative ring in which every non-zero element has an inverse.*

Definition 10. *An ideal in a ring R is a non-empty subset C of R satisfying:*

- *$a - b \in C$ for all $a, b \in C$ (closed under subtraction).*

- *ra and ar are all in C, for all $a \in C$, $r \in R$ (closed under outside multiplication).*

Example 6.7. *In $R = \mathbb{Z}$ the subset $n\mathbb{Z}$ is an ideal, and the resulting quotient ring by that ideal is $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{0, 1, 2, \ldots, n - 1\}$.*

Theorem 7. *If $\alpha \neq 0$ in $\mathbb{Z}[i]$, then $n(\alpha) = N(\alpha)$, where $n(\alpha)$ denotes the number of GAUSSIAN Integers modulo α. That is, the size of $\mathbb{Z}[i]/\alpha\mathbb{Z}[i]$ 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 $\#(\mathbb{Z}[i]/\alpha\mathbb{Z}[i]) = N(\alpha)$ with $\alpha \neq 0$.

A fundamental example of a finite (Galois) Field is the set $\mathbb{F}_p$ of p-modulo remainders, where p is a given prime number. Here, as in $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$, the set of elements is $\{0, 1, \cdots, p - 1\}$, and the operation $\oplus$ is p-modulo addition. The multiplicative operation * is p-modulo multiplication; that is to say, multiply integers as usual and then take the remainder after division by p.

Theorem 8. *(Prime Fields). For every prime p, $\mathbb{Z}_p$ forms a field (denoted by $\mathbb{F}_p$) under p-modulo addition and multiplication.*

Theorem 9. *(Prime Field Uniqueness). Every field $\mathbb{F}$ with a prime number p of elements is isomorphic to $\mathbb{F}_p$.*

7 C&F Uncoded System Model: Scalar Case

In this section, we are going to study the C&F system model, where the messages w_c are considered scalars. We are going to do the construction and theoretical description of the system and then proceed to the implementation and analysis of its performance.

7.1 Construction

The first stage is to define the lattice codebook we are going to use and the functions of encoding and decoding involved in the lattice network code. Next, some brief concepts about ML (Maximum Likelihood) detection will be introduced. Further, the C&F system model will be presented and studied in detail. Finally, a derivation of the probability of error of the system will be done.

7.1.1 Introduction

Most of the material in this section is based in the C&F scheme studied in Gupta and Vázquez [2012] and de Zarzà et al. [2013]. This information is complemented by the mappings of encoding and decoding first explained in Huber [1994]. Thorough explanations and detailed derivations are given.

In this section we focus on GAUSSIAN Integer primes of type $p = 1 \bmod 4$, where p can be written as a product of two primes in $\mathbb{Z}[i]$, $p = \pi\pi^*$. We are interested in this kind of primes because it allows that $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ have the same number of elements and as a consequence it is possible to construct an isomorphism $\mathbb{Z}/p\mathbb{Z} \rightarrow \mathbb{Z}[i]/\pi\mathbb{Z}[i]$, i.e, the two residue class systems have the same number of elements, the same structure, and in particular, they are both fields with p elements.

Let $\mathbb{Z}[i]/\pi\mathbb{Z}[i]$ the residue class of $\mathbb{Z}[i]$ modulo π, where the function modulo $\psi : \mathbb{Z}[i] \rightarrow \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ 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 $\mathbb{Z}[i]$, and solve for the residue γ.

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

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

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 earlier.

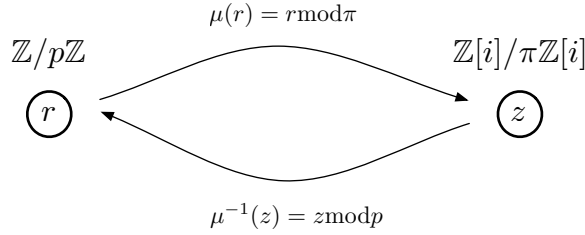And if we solve for gamma (the residue) we get

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

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 build the C&F system model, 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 p-modulo $\mathbb{Z}[i]/\pi\mathbb{Z}[i] \to \mathbb{Z}/p\mathbb{Z}$ properly, first we need to remember that π and π^* are relatively primes and in terms of BÉZOUT Theorem, it can be translated as:

$$1 = u\pi + v\pi^* \tag{2}$$

where u and v can be computed using the Euclidean algorithm.

We need to define the inverse application 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

$$r = k\pi + z \to r \bmod \pi = z \bmod \pi. \tag{3}$$

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

$$r \quad = \quad r^* = k^* \pi^* + z^* \rightarrow r \bmod \pi = (k^* \pi^* + z^*) \bmod \pi. \tag{4}$$

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 (2)

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

If 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:

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

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

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

Therefore, we can define the inverse function as the function p-modulo 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 (3) and (4). And using the function $\bmod p$ defined above:

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

Thus, we have defined the inverse function.

Once we have all mappings defined we are prepared to study the C&F system model. However, first we are going

to introduce some basic concepts about ML detection.

7.1.2 ML Detection

If we consider the MIMO linear system diagram shown in Figure 3, in order to communicate over this channel, we are faced with the task of detecting a set of M transmitted symbols from a set of N observed signals. Our observations are corrupted by a non-ideal channel of communication, which is normally modeled as a linear system followed by an additive noise vector.
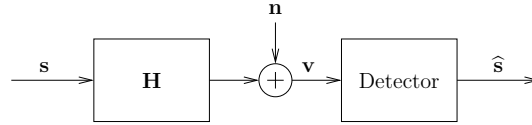

Figure 3: MIMO Simplified Linear Communication System Diagram.

We can observe in Figure 3, a MIMO simplified communication system diagram with $\mathbf{s} \in \mathcal{X}^M$, channel matrix $\mathbf{H} \in \mathbb{C}^{N \times M}$, additive noise vector $n \in \mathbb{C}^N$, received vector $\mathbf{v} \in \mathbb{C}^N$ and detected symbol vector $\hat{\mathbf{s}} \in \mathbb{C}^M$.

We draw the transmitted symbols from a finite alphabet $\mathcal{X} = x_1, x_2, x_B$ of size B. The detector's role is then to choose one of the B^M possible transmitted symbol vectors based on the available data. Our intuition correctly suggests that an optimal detector should return $\hat{\mathbf{s}} = s_*$, the symbol vector whose probability of having been sent, given the observed signal vector $\mathbf{v}$, is the largest:

$$
\begin{align}
s_* &= \arg\max_{\mathbf{s}\in\mathcal{X}^M} P(\mathbf{s} \text{ was sent} \mid \mathbf{v} \text{ is observed}), \tag{5}\\
&= \arg\max_{\mathbf{s}\in\mathcal{X}^M} \frac{P(\mathbf{v} \text{ is observed} \mid \mathbf{s} \text{ was sent})P(\mathbf{s} \text{ was sent})}{P(\mathbf{v} \text{ is observed})}. \tag{6}
\end{align}
$$

Equation (5) is known as the MAP (maximum a posteriori probability) detection rule. Making the assumption that the symbol vectors $\mathbf{s} \in \mathcal{X}^M$ are equiprobable, that is to say, that $P(\mathbf{s} \text{ was sent})$ is constant, the MAP optimal detection rule can be written as:

$$
s_* = \arg\max_{\mathbf{s}\in\mathcal{X}^M} P(\mathbf{v} \text{ is observed} \mid \mathbf{s} \text{ was sent}). \tag{7}
$$

A detector that always returns an optimal solution satisfying Equation (7) is called ML (Maximum Likelihood) detector. If we assume that the additive noise $\mathbf{n}$ is white and Gaussian, we can express the ML detection problem as the minimization of the squared Euclidean distance metric to a target vector $\mathbf{v}$ over a M-dimensional finite discrete search:

$$
\mathbf{s}_* = \arg\min_{s\in\mathcal{X}^M} |\mathbf{v} - \mathbf{H}\mathbf{s}|^2. \tag{8}
$$

In this project we are going to use two different methods of sphere decoder to obtain ML or near ML estimations.

7.1.3 System Model

We consider the system model Compute and Forward with L sources, a relay and a destination, proposed by Gupta and Vázquez [2012] and developed by de Zarzà et al. [2013].
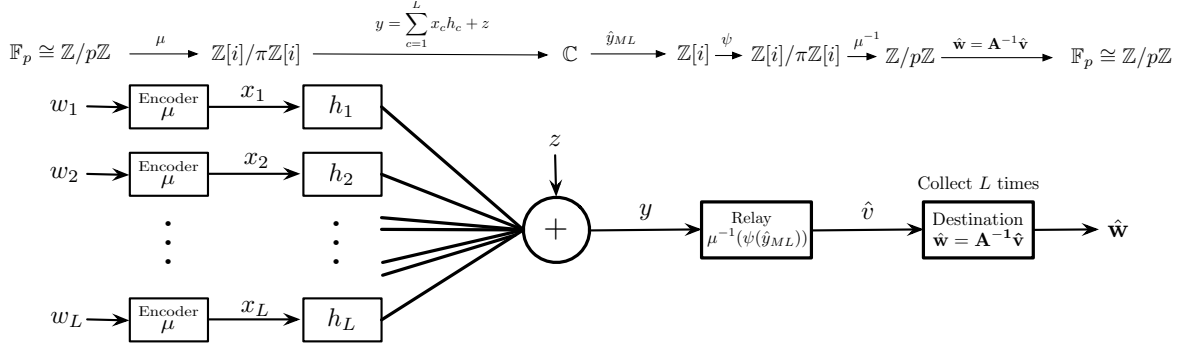

Figure 4: C&F System Model.

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 $\mathbf{w} = [w_1 \ldots w_L]$. 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:

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

The signals are transmitted across the channel to the relay. We assume that the channel undergoes slow fading and hence 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 $\psi : \mathbb{Z}[i] \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$

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

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 Euclidean Algorithm, as stated in Section 6.4.

In order to decode the linear combination v, the relay obtains a ML 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.

7.1.4 Probability of Error

In this section we propose an analytical expression for probability of error at the destination given the described system. We are going to take as a reference the Union Bound (UB) proposed in de Zarzà et al. [2013]; Gupta and Vázquez [2012]. However, the expressions found below are accurate for $L = 2$, the extension to higher dimension has to consider additional advanced lattice theory, and is far from the scope of this project (see for instance a lattice based union bound in Sun et al. [2013]).

The probability of error at the destination is defined as $Pr(\hat{w} \neq w)$. From de Zarzà et al. [2013]; Gupta and Vázquez [2012] we can see a theoretical expression for the probability of error using the UB on the given system:

Theorem 10. *The UB estimate of probability of error at the destination with L sources using a finite field of size p and GAUSSIAN Integer residue class based signal constellation is given by*

$$P_{error} \leq P_1 + (LP_R),$$

where

$$P_1 = 1 - \prod_{t=1}^{L} \left(1 - \frac{1}{pt}\right),$$

and

$$P_R = 1 - \left(erf\left(\frac{1}{2\sqrt{2}\sigma}\right)\right)$$

such that σ^2 is the variance of additive noise at the relay.

We propose a second bound where we are going to consider a complex GAUSSIAN channel (RAYLEIGH faded channel).

Proposed probability of error

We are going to derive an analytical expression for probability of error.

We can observe that the probability of error, $Pr(w \neq \hat{w})$, is given by the probability of error at the relay, $P_{errorRelayL}$, and the probability of error at the destination, $P_d = Pr(|\mathbf{A}| = 0)$. An error can occur due to rank failure and/or due to error at the relay.

Theorem 11. *The estimate of probability of error at the destination with L sources using finite field of size p and GAUSSIAN Integer residue class based signal constellation is given by*

$$P_{error} \leq 1 - P_{noerrorDestination} \cdot P_{noerrorRelayL},$$

where

$$P_{noerrorDestination} = \prod_{c=1}^{L} \left(1 - \frac{1}{p^c}\right) \qquad \text{and} \qquad P_{noerrorRelayL} = \left(1 - e^{-\frac{1}{8\sigma^2}}\right)^{L}$$

such that σ^2 is the variance of the additive noise at the relay.

Proof. First we are going to compute the probability of not having an error at the destination, $P_{noerrorDestination}$. This partial result will be based on Waterhouse [1987]. It consists on calculating the probability of being able to invert a $L \times L$ matrix.

When the entries of a matrix are independent uniformly distributed random variables then all matrices are equally likely, and we simply have to determine what portion of them are invertible, that is to say, how many matrices are

built by independent vectors.

The probability of having an invertible matrix, $P_{|\mathbf{A}|\neq 0}$, can be computed using the formula of probability:

$$P_{|\mathbf{A}|\neq 0} = \frac{\#\text{ possible outcomes}}{\#\text{ total outcomes}}.$$

We can calculate the total number of possible outcomes as follows: the first column of an invertible matrix can be any one of the $p^L - 1$ non-zero vectors: each vector has L components and each component can be one of the p elements of the field (p^L choices), finally we subtract the all zero vector in order to have all the independent choices.

Then, the second column can be any of the vectors not a multiple of the first one: $(p^L - 1) - (p - 1) = p^L - p$, where we have $p^L - 1$ possible vectors for the second column minus the first column and its multiples $p - 1$.

Proceeding inductively we see that column $k + 1$ can be chosen to be any of the $(p^L - 1) - (p^k - 1) = p^L - p^k$ independent vectors to the k columns before, where we have $p^L - 1$ possible vectors for the column $k + 1$ minus $p^k - 1$ linear combinations of the k previous vectors [3].

Therefore, we have

$$\#\text{ possible outcomes} = (p^L - 1)(p^L - p) \cdots (p^L - p^{L-1}).$$

The number of total outcomes are the total number of matrices [4]

$$\#\text{ total outcomes} = p^{L^2}.$$

Finally,

$$
\begin{aligned}
P_{|\mathbf{A}|\neq 0} &= \frac{\#\text{ possible outcomes}}{\#\text{ total outcomes}}, \\
&= \frac{(p^L - 1)(p^L - p) \cdots (p^L - p^{L-1})}{p^{L^2}}, \\
&= \frac{p^L\left(1 - \frac{1}{p^L}\right) p^L\left(1 - \frac{1}{p^{L-1}}\right) \cdots p^L\left(1 - \frac{1}{p}\right)}{p^{L^2}}, \\
&= \frac{p^{L^2}\left(1 - \frac{1}{p^L}\right)\left(1 - \frac{1}{p^{L-1}}\right) \cdots \left(1 - \frac{1}{p}\right)}{p^{L^2}}, \\
&= \left(1 - \frac{1}{p}\right) \cdots \left(1 - \frac{1}{p^L}\right)\left(1 - \frac{1}{p^{L-1}}\right),
\end{aligned}
$$

---

[3]In order to calculate how many possible linear combinations of the previous k vectors are, it is necessary to have in mind that a linear combination is a scaled sum of vectors, that is to say, we multiply each vector by a coefficient and add them together. Therefore, each coefficient has p possible values and we can choose independently one coefficient for each of the k vectors, which means that we have p^k total possibilities. Finally we have to subtract the all zero coefficients case. Thus, the total number of linear combinations is $p^k - 1$.

[4]A $L \times L$ matrix has L^2 components and each component can be one of the p elements of the field. Thus we have p^{L^2} possible matrices.

$$= \prod_{c=1}^{L}\left(1-\frac{1}{p^c}\right).$$

Thus, the probability that a $L \times L$ matrix over the field with p elements has a determinant different from zero is

$$P_{|\mathbf{A}|\neq 0} = \prod_{c=1}^{L}\left(1-\frac{1}{p^c}\right).$$

Therefore the probability of not having an error over a field with p elements and collecting L linear combinations at the destination is

$$P_{noerror\,Destination} = \prod_{c=1}^{L}\left(1-\frac{1}{p^c}\right).$$

Now, we are going to calculate the probability of no error at the relay.

In order to obtain the linear combination $\hat{v}$, the relay obtains a ML estimate of the received signal y to remove the noise and obtain the closest GAUSSIAN Integer to y. Further, the decoder at the relay is $\hat{v} = \mu^{-1}(\psi(\hat{y}_{ML}))$. Here, the only source of error is the ML estimate because the noise could have made us do an incorrect guess. Thus,

$$P_{error\,Relay} = Pr(\hat{y}_{ML} \neq h_1 x_1 + h_2 x_2 + \ldots + h_L x_L)$$

where $h_l, x_l \in \mathbb{Z}[i]$, and so the above expression is reduced to the probability that the added noise exceeds the decision threshold. In this case the decision threshold depends on the distance between two GAUSSIAN Integers. We know that between two different numbers in $\mathbb{Z}[i]$ the minimum distance is 1, so the probability that the added noise does not exceed the decision threshold is the probability that the noise norm distribution does not exceed $1/2$.

$$P_{noerror\,Relay} = Pr(||z|| \leq 1/2)$$

where $P_{noerror\,Relay}$ is the probability that there is no decoding error in one linear combination at the relay.

If we consider that we want to decode L independent linear combinations correctly at the relay, we can say

$$P_{noerror\,Relay\,L} = \prod^{L} Pr(||z|| \leq 1/2) = (Pr(||z|| \leq 1/2))^L$$

where $P_{noerror\,Relay\,L}$ is the probability that there is no decoding error in L different linear combinations at the relay.

The noise is assumed to have a circular symmetric distribution gaussian $z \sim \mathcal{CN}(0, \sigma^2)$. The norm of a circular symmetric distribution gaussian follows a RAYLEIGH distribution. This assumption may be inaccurate depending on the modeled scenario.

Let $||z||$ be a random variable with RAYLEIGH distribution its probability density function $f(x, \sigma)$ and cumulative distribution function $F_{||z||}(x) = Pr(||z|| \leq x)$ are defined by:

$$
\begin{aligned}
f(x; \sigma) &= \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \quad x \geq 0 \\
F_{||z||}(x) &= \int_{-\infty}^{x} f(t)dt = \int_{0}^{x} \frac{t}{\sigma^2} e^{-\frac{t^2}{2\sigma^2}} dt = 1 - e^{-\frac{x^2}{2\sigma^2}}.
\end{aligned}
$$

The probability of not having an error

$$
Pr(||z|| \leq 1/2) = F_{||z||}(1/2) = 1 - e^{-\frac{1}{8\sigma^2}}.
$$

Using the equation stated before

$$
\begin{aligned}
P_{noerror\,RelayL} &= (Pr(||z|| \leq 1/2))^L, \\
&= (F_{||z||}(1/2))^L, \\
&= (1 - e^{-\frac{1}{8\sigma^2}})^L.
\end{aligned}
$$

Finally, the estimate of probability of error at the destination in the system described with L sources using finite field of size p and GAUSSIAN Integer residue class based signal constellation is

$$
\begin{aligned}
P_{error} &\leq 1 - P_{noerror} = 1 - P_{noerror\,RelayL} \cdot P_{noerror\,Destination}, \\
&\leq 1 - \left(1 - e^{-\frac{1}{8\sigma^2}}\right)^L \cdot \prod_{c=1}^{L} \left(1 - \frac{1}{p^c}\right).
\end{aligned}
$$

$\square$

7.2 Performance

We are going to implement the system model with Matlab.

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, 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

$$\text{Average signal power} = \frac{1}{p}\sum_{c=1}^{p} x_c x_c^* \tag{9}$$

and therefore

$$\text{Noise power} = \frac{\frac{1}{p}\sum_{c=1}^{p} x_c x_c^*}{\text{SNR}_{Lineal}}. \tag{10}$$

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 $\mathbf{A}$ in modulo p.

First, we need to know if the output matrix $\mathbf{A}$ 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, 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 $gcd \bmod p = u \cdot det \bmod p$, 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 $\mathbf{A}$ 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 $\mathbf{A}$ modulo p, we are able to calculate $\hat{w}$.

7.2.1 $L = 2$ C&F System Implementation

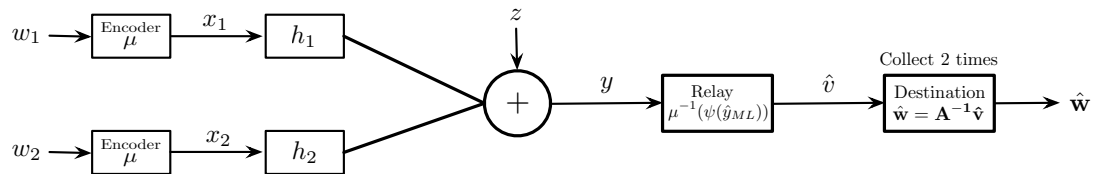

Figure 5: C&F System Model $L = 2$.

We have generated a function which gives us the probability of error and the Union Bound given p, π and L. This is a first basic approach to the implementation of the system, where the ML estimation has been simplified as an operation of rounding to the nearest point in the lattice.

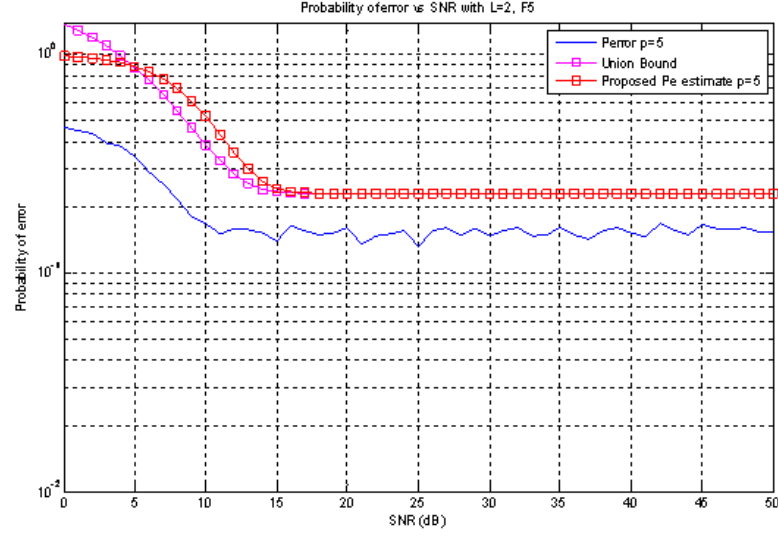We simulate the system for $p = 5$, $\pi = 2 + i$, $L = 2$, see the next figure.

Figure 6: System Simulation for $p = 5$, $\pi = 2 + i$, $L = 2$.

We can see in the plot the Union Bound (magenta line) and the proposed Pe estimate (red line). The first UB de Zarzà et al. [2013]; Gupta and Vázquez [2012] adjusts better to the simulated data for SNR low and both go towards the same value as SNR goes up. Finally, the blue line is the simulated probability of error of the system.

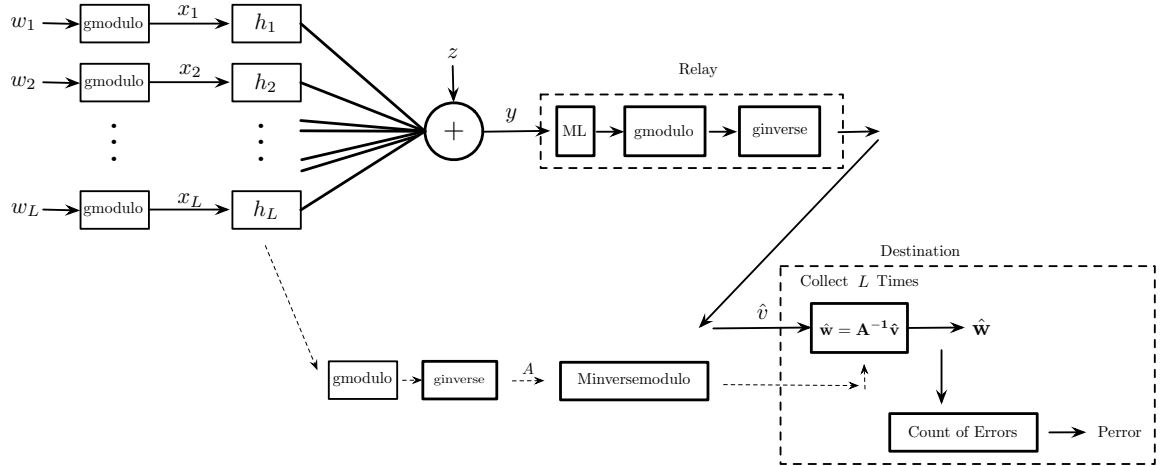### 7.2.2 C&F L-dimensional Antenna System Implementation

Figure 7: Scalar Model.

This is a general L-dimensional implementation of the C&F system. All the steps in the boxes correspond to the MATLAB functions implemented. The ML step is computed using a sphere decoder. The message w is considered scalar.

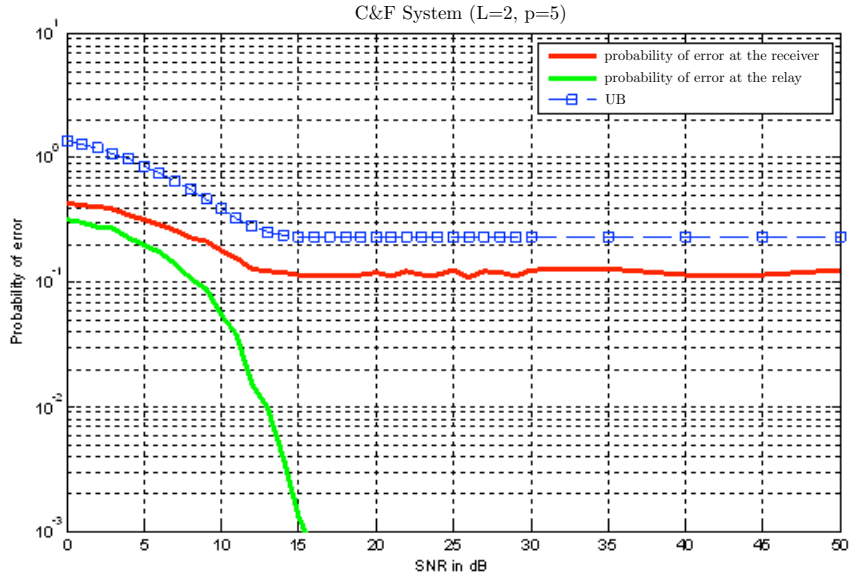We simulate the system using $L = 2$, $p = 5$ ($\pi = 2 + 1i$)

Figure 8: $p = 5$, $\pi = 2 + 1i$, $L = 2$.

The red line is the probability of error at the receiver. The green line shows the probability of error at the relay. The blue line corresponds to the UB Estimate of probability of error.

Next, we are going to simulate for $L = 4$.

Figure 9: $p = 5$, $\pi = 2 + 1i$, $L = 4$.
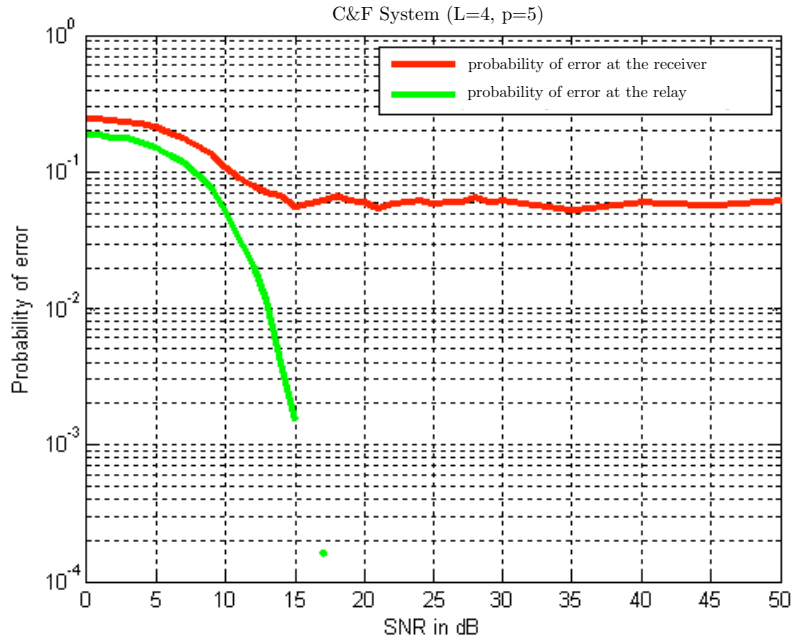
The red line is the probability of error at the receiver. The green line displays the probability of error at the relay.

In the two plots above, it can be seen that for SNR low, errors at the relay are really important, however, as SNR goes up, errors are due to rank failure.

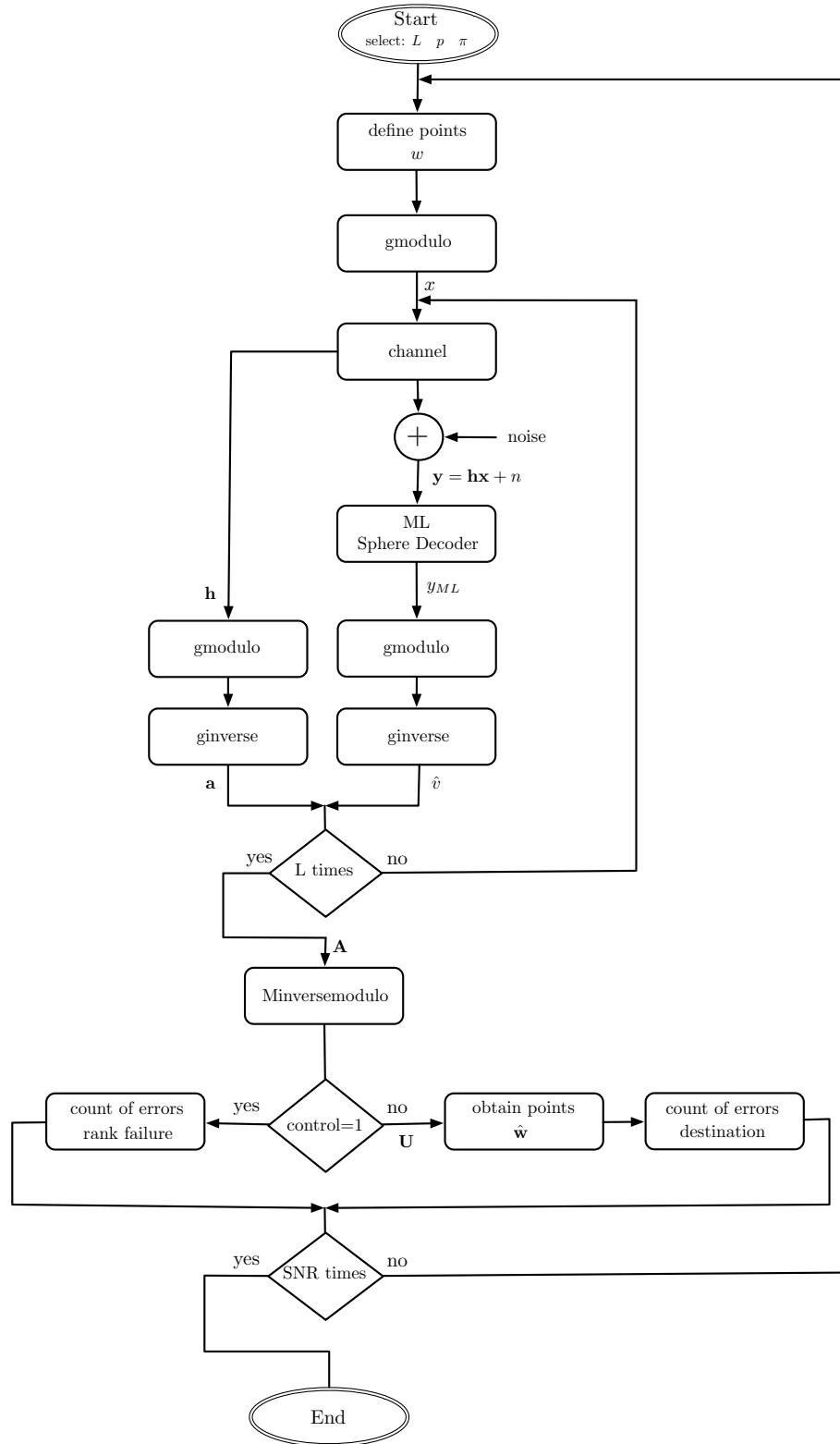We enclose the flow chart diagram of the C&F scalar system.

Figure 10: Flow Chart Diagram. C&F Scalar System.

8 Extension of the C&F Uncoded System Model: Vectorial Case

In this section, we are going to use the C&F system model but considering the transmitted messages w_c as vectors. The theoretical extension will be done as well as the performance analysis.

8.1 Construction

We are going to extend the model Compute and Forward studied to higher dimension, we use the formulation described in Belfiore and Ling [2011].

We consider L sources transmitting messages $\mathbf{s}_1, \ldots, \mathbf{s}_L$ to one relay, which transmits a linear combination of these L messages. The received signal at the relay is

$$\mathbf{y} = \sum_{c=1}^{L} h_c \mathbf{x}_c + z$$

where h_c is the channel coefficient and $\mathbf{x_c}$ is the vector transmitted by source c (see that now we are transmitting a vector not a single point).

We consider $\mu(\mathbf{s}_c) = \mathbf{x_c}$, where $\mathbf{s}_c : (\mathbb{Z}/p\mathbb{Z})^n \to \mathbf{x}_c : (\mathbb{Z}[i]/\pi\mathbb{Z}[i])^n$, where the mapping is done component to component.

The relay decodes a noiseless linear combination of the transmitted messages

$$\mathbf{v} = \sum_{c=1}^{L} a_c \mathbf{s}_c$$

and retransmits it to the destination.

We consider the channel coefficients h_c complex, circular, i.i.d. Gaussian and $a_c \in \mathbb{Z}/p\mathbb{Z}$ can be found using

$$a_c = \mu^{-1}(\psi(h_c)).$$

Now, after calculating the vector $\mathbf{a} = [a_1 \quad a_2 \quad \ldots \quad a_L]^T$, we can proceed.

ML Decoder

The relay wants to decode a linear system of equations of the transmitted message and pass it to the destination. The relay obtains a linear combination of the transmitted signals, which can be written as follows:

$$\mathbf{y} = \sum_{c=1}^{L} h_c \mathbf{x}_c + z$$

where z is circular, complex, additive i.i.d. GAUSSIAN noise.

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

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

Now this signal is mapped to $(\mathbb{Z}/p\mathbb{Z})^n$. Therefore, the decoder at the relay is given by

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

The recovered linear system of equations is

$$\hat{\mathbf{v}} = \sum_{c=1}^{L} a_c \mathbf{s}_c$$

where $\mathbf{s_c} \in (\mathbb{Z}/p\mathbb{Z})^n$.

And therefore

$$\hat{\mathbf{v}} = \begin{bmatrix} a_1 & a_2 & \dots & a_L \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_L \end{bmatrix}$$

where $\mathbf{s}_c$ is a row vector.

The estimate of the linear combination $\mathbf{v}$ 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 $\mathbf{s}_c$ from $\mathbf{v}$, we need to collect k times such linear combinations. Therefore, the L linear combinations obtained at the destination can be written as

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

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

$$\hat{\mathbf{V}} = \mathbf{A} \cdot \mathbf{S} \Rightarrow \hat{\mathbf{S}} = \mathbf{A}^{-1} \hat{\mathbf{V}}.$$

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.
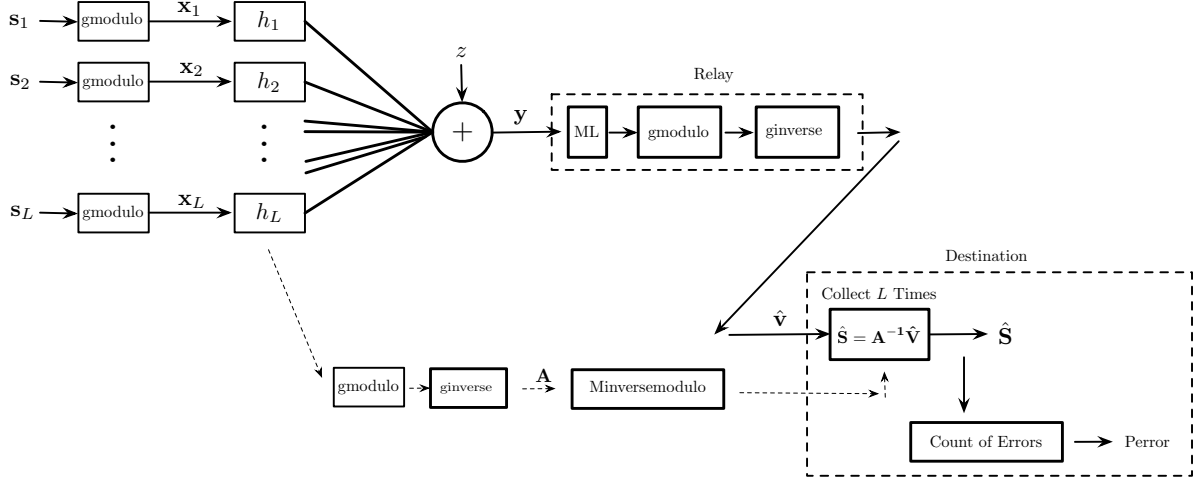
8.2 Performance

Figure 11: C&F System Model Vectorial Case.

We extend the algorithm for message w_c to be a vector $\mathbf{s}_c \in (\mathbb{Z}/p\mathbb{Z})^n$, using the theory described in the earlier section.

We simulate the system using a message vector with length $n = 4$, using $p = 5$ ($\pi = 2 + 1i$).
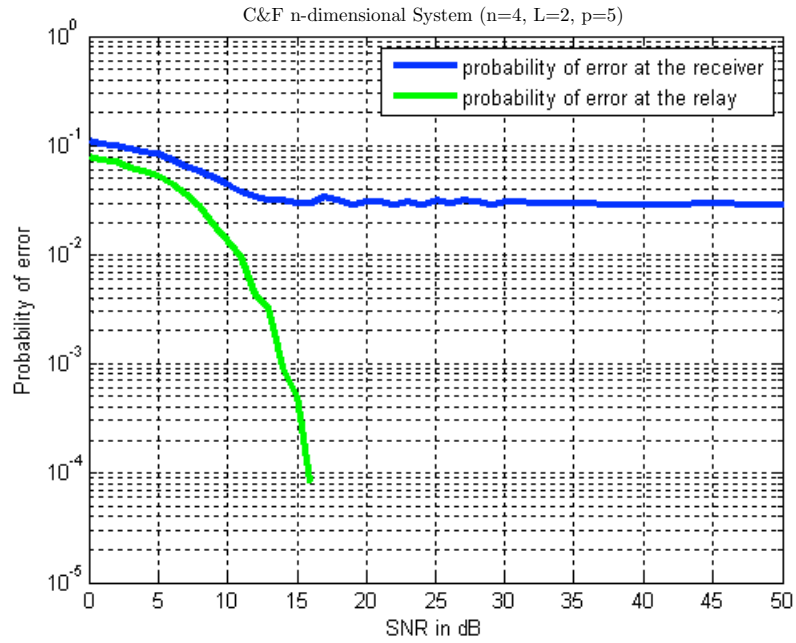
Figure 12: C&F System $p = 5$, $\pi = 2 + 1i$, $L = 2$, $n = 4$.

The blue line is the probability of error at the receiver. The green line corresponds to the probability of error at the

relay.

Here we can see that in the n-dimensional case the behavior is similar to the scalar case, where errors for SNR low are due to errors at the relay and for SNR high are caused by rank failure.

We enclose the flow chart diagram of the C&F n-dimensional system.

Figure 13: Flow Chart Diagram. C&F Vectorial System.

9 C&F HAMMING q-ary Coded System Model

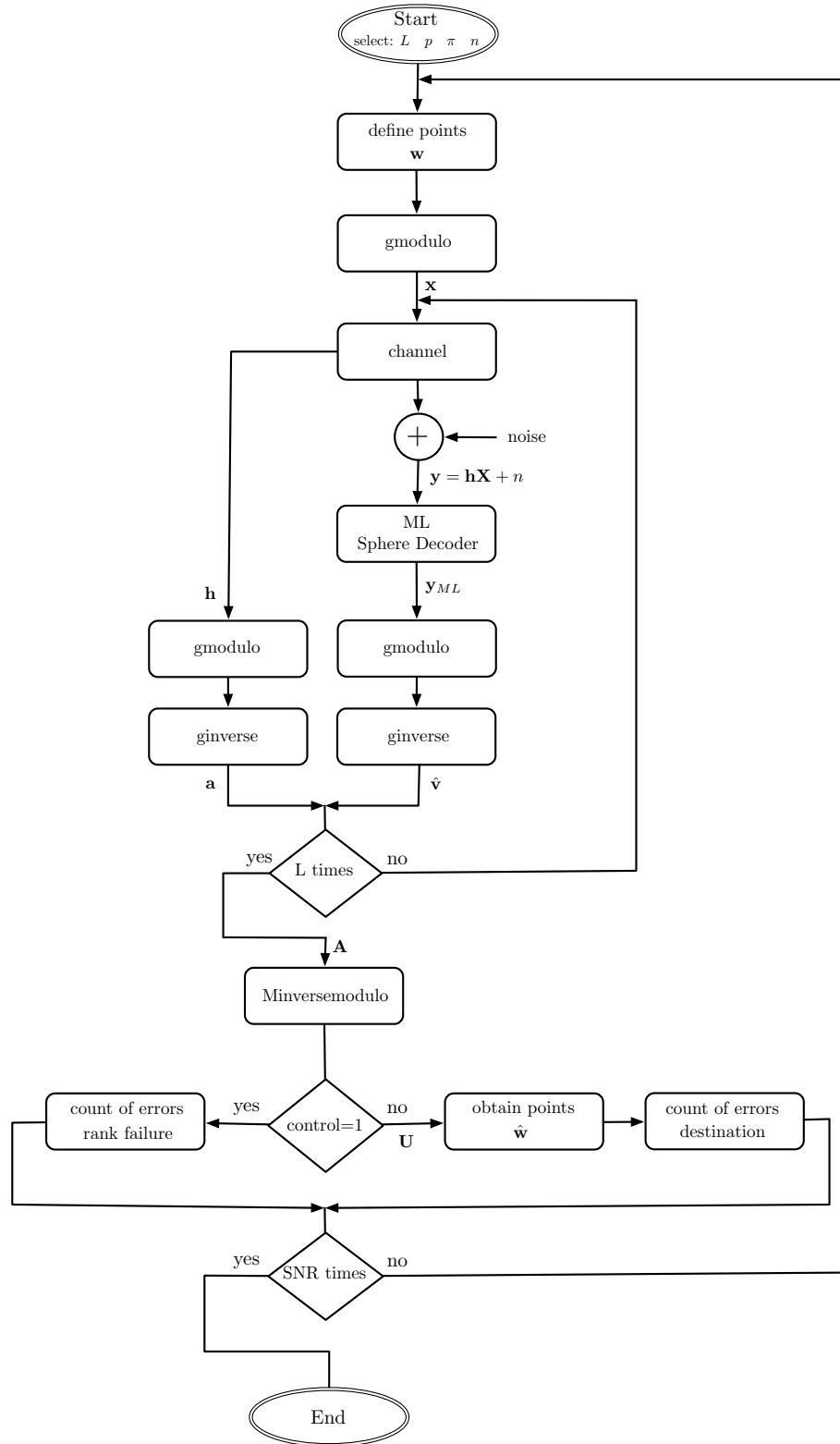In this section we are going to focus on the implementation of a C&F coded system. We are going to implement a HAMMING q-ary (6,4) into the C&F n-dimensional ($n = 4$) system. A theoretical description will be done as well as the performance analysis.

9.1 Construction

We will first start by doing a brief description of the basic theory needed about Linear Codes and HAMMING q-ary codes. Examples will be given to exemplify the process.

9.1.1 Linear Codes

Let $F = \mathbb{F}_q$ be a finite field with $q = |F|$ elements.

Definition 11. : *A linear code of dimension k and length n, that is to say, a $[n, k]$-code, over a field F is a subspace $C \subset F^n$ with $dim_F(C) = k$.*

Remark: By definition, a code $C \subset F^n$ is linear if, and only if, $v_1, v_2 \in C \quad a_1, a_2 \in F \to a_1 v_1 + a_2 v_2 \in C$.

Definition 12. *A generating matrix of a $[n, k]$-code C is a $k \times n$ matrix G such that*

$$C = uG : u \in F^k. \tag{11}$$

We say that G is systematic if $G = (I_k| - P^T)$.

Definition 13. *A parity check matrix of a $[n, k]$-code C is a $m \times n$ matrix H such that*

$$C = \{v \in F^n : Hv^t = 0\}.$$

Proposition 9.1. *If $G = (I_k| - P^T)$ is a systematic generating matrix of a $[n, k]$-code C, then a parity check matrix for C is*

$$H = (P|I_{n-k}).$$

9.1.2 HAMMING q-ary Codes

Definition 14. *Let F be a field of order q. If $m \geq 2$ is an integer, put $n = (q^r - 1)/(q - 1)$. The HAMMING q-ary code of type $[n, n - r, 3]$ is the code C_H defined by the $m \times n$ parity check matrix*

$$H = (v_1 \,|\, v_2 \,|\, \ldots \,|\, v_n)$$

where $v_1, \ldots, v_n \in F^m$ is a list of (non-zero) vectors satisfying the condition that no two vectors are scalar multiples of each other.

This can be best understood by an example:

Example 9.1. *We consider the case $\mathbb{F}_5$ and $r = 2$, $n = \frac{5^2-1}{5-1} = 6$. Therefore: $k = n - r = 4$.*

An easy way to write down a parity check matrix is to list as columns all the non-zero vectors in F_q^r whose first non-zero entry is 1.

Therefore, a straightforward way to generate a systematic HAMMING q-ary code is generating the matrix P as a $r \times k$ matrix with columns

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

And then generate H and G using $G = (I_k| - P^T)$ and $H = (P|I_{n-k})$.

If we do the computation, we get

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 4 & 4 \\ 0 & 1 & 0 & 0 & 4 & 3 \\ 0 & 0 & 1 & 0 & 4 & 2 \\ 0 & 0 & 0 & 1 & 4 & 1 \end{bmatrix}.$$

We have written a MATLAB code that generates matrices H and G given the input parameters.

- **Encoding**

  We can encode a given vector w_k using the matrix Generator as $w_k G$.

  Example 9.2.

  $$\begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 4 & 4 \\ 0 & 1 & 0 & 0 & 4 & 3 \\ 0 & 0 & 1 & 0 & 4 & 2 \\ 0 & 0 & 0 & 1 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 2 & 4 & 4 \end{bmatrix}$$

  where we can see that if $w_k = \begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix}$ the coded vector is $w_{coded} = \begin{bmatrix} 1 & 2 & 1 & 2 & 4 & 4 \end{bmatrix}$.

- **Decoding**

  Furthermore, in order to decode a given vector $w_{codederror}$ such that

  $$w_{codederror} = (c \quad + \quad \begin{bmatrix} 0 & \ldots & 0 & b & 0 & \ldots & 0 \end{bmatrix}),$$

  where b is in the k-th component and c is in the codeword space,

we proceed as

$$Hw_{coded}^T = bH^k. \tag{12}$$

That is to say, the k-th component of H multiplied by b. If the result is not zero, there is an error. The sent codeword is obtained subtracting b to the k-th component of w_{coded}.

The easiest way to understand this procedure is working through an example.

Let's first see an example where a correct codeword is received.

Example 9.3. *Suppose $w_{codednoerror} = \begin{bmatrix} 1 & 2 & 1 & 2 & 4 & 4 \end{bmatrix}$ then*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Therefore, the syndrome is 0 and it means we have no error and the received codeword is in the codeword space.

Now, let's suppose an example where we have an error in the third component by a factor $+1$.

Example 9.4. *Suppose $w_{codederror} = \begin{bmatrix} 1 & 2 & 2 & 2 & 4 & 4 \end{bmatrix}$ then*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 4 \\ 4 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

Therefore, the syndrome is in the 3rd column of H and $b = 1$ and we can find the original codeword as

$$w_{coded} = \begin{bmatrix} 1 & 2 & 2 & 2 & 4 & 4 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 2 & 4 & 4 \end{bmatrix}.$$

Finally, let's see an example where we have an error in the second component by a factor -2.

Example 9.5. *Suppose $w_{codederror} = \begin{bmatrix} 1 & 0 & 1 & 2 & 4 & 4 \end{bmatrix}$ then*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} = -1 \begin{bmatrix} 2 \\ 4 \end{bmatrix} = -2 \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Therefore, the syndrome is in the 2nd column of H and $b = -2$ and we can find the original codeword as

$$w_{coded} = \begin{bmatrix} 1 & 0 & 1 & 2 & 4 & 4 \end{bmatrix} - \begin{bmatrix} 0 & -2 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 2 & 4 & 4 \end{bmatrix}.$$

9.2 Performance

We are going to use a p-ary ($p = 5$) Hamming(6,4) as stated in the following diagram, a MATLAB function has been implemented following the next design
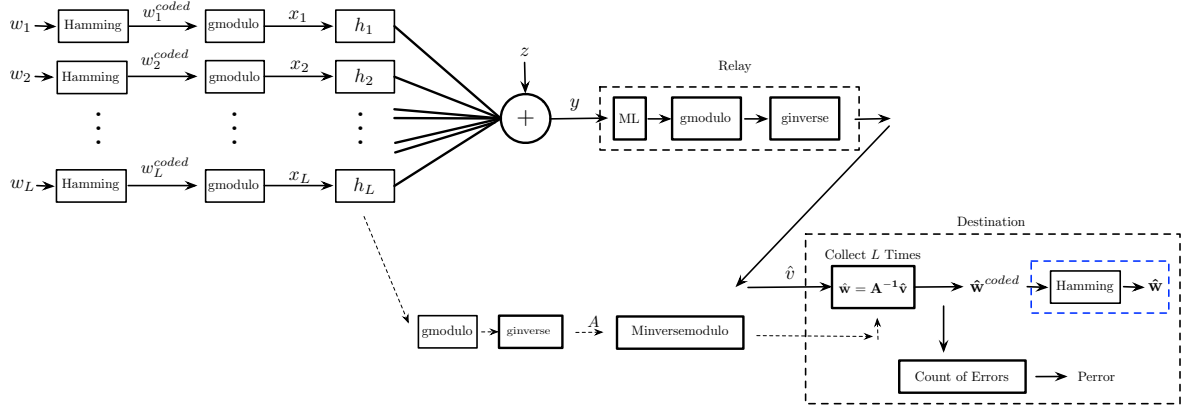

Figure 14: Coded System.

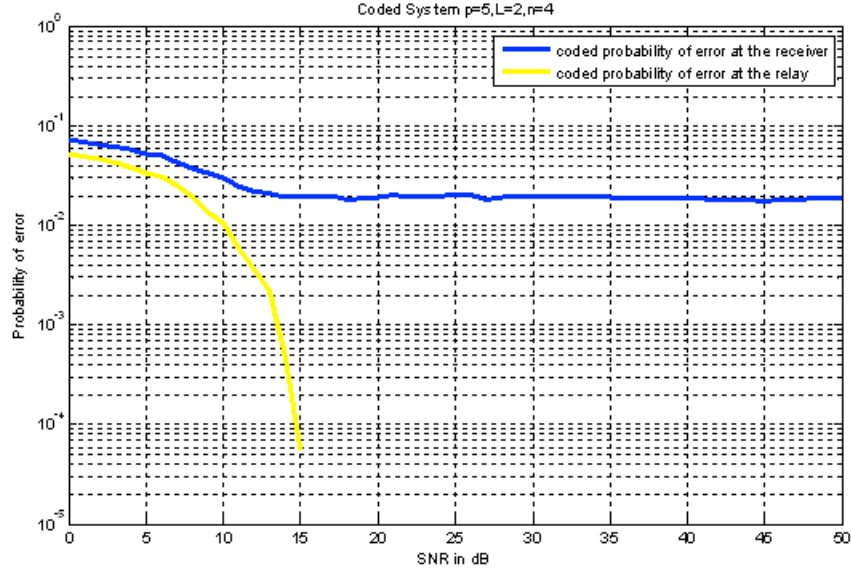We simulate the system using $n = 4$, $p = 5$ ($\pi = 2 + 1i$) and $L = 2$.

Figure 15: Coded $p = 5$, $\pi = 2 + 1i$, $L = 2$, $n = 4$.

The blue line is the probability of error at the receiver. The yellow line shows the probability of error at the relay.

For SNR low relay errors continue to be really important whereas for SNR high the most important cause of error is rank failure.

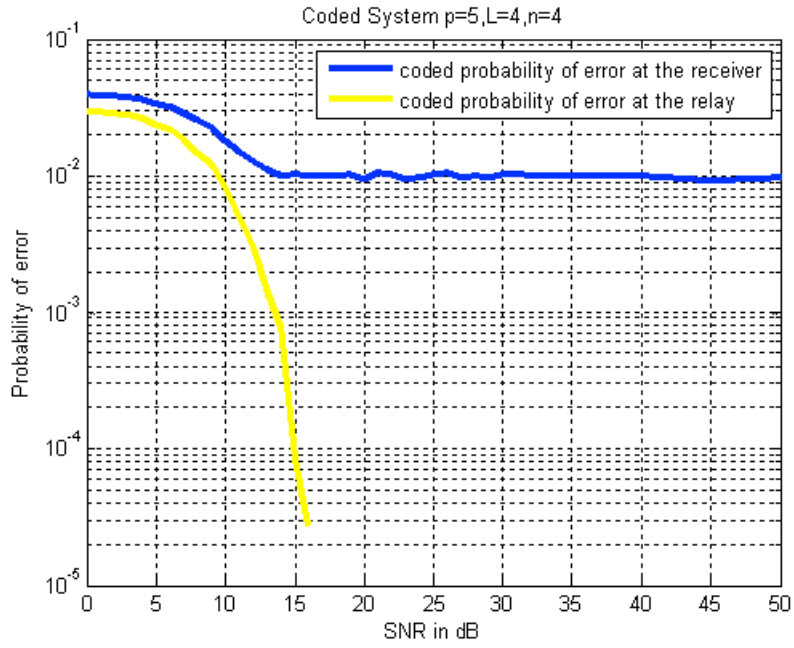If we simulate the given system using more antennas, $L = 4$, we can see the following result.

Figure 16: Coded $p = 5$, $\pi = 2 + 1i$, $L = 4$, $n = 4$.

Where we can see the same qualitative behavior than in the latter case. The $L = 4$ case seems to achieve a slightly better performance in terms of probability of error.

9.2.1 Uncoded vs Coded

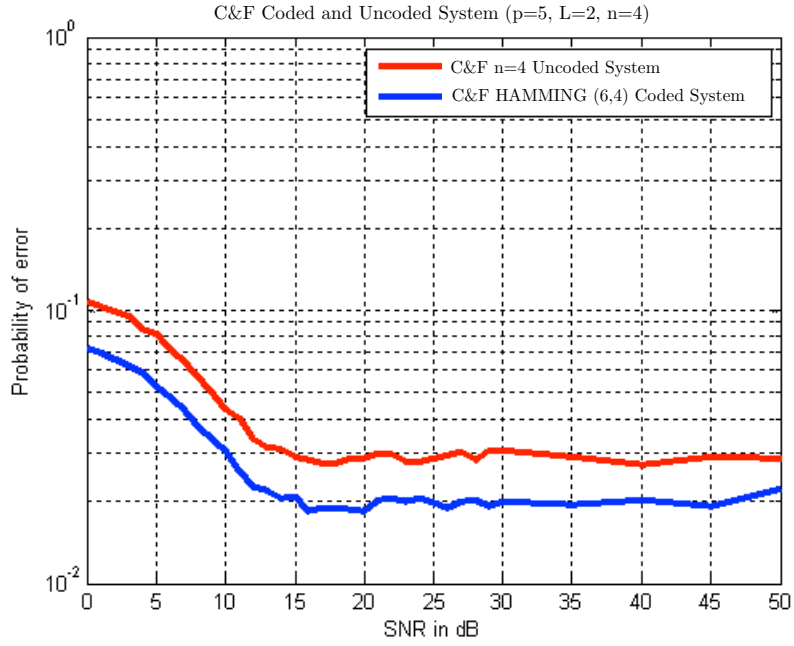We are going to compare the two systems at the same time for a given $p = 5$ ($\pi = 2 + i$), $L = 2$ and $n = 4$.

Figure 17: Uncoded vs Coded $p = 5$, $\pi = 2 + 1i$, $L = 2$, $n = 4$.

The red line is the uncoded system probability of error with $p = 5$, $L = 2$, $n = 4$ whereas the blue line displays to the coded system with $p = 5$, $L = 2$, $n = 4$ and a p-ary ($p = 5$) Hamming(6,4).

We can see how the coded system has a lower probability of error than the uncoded system, this is due to the fact that we are using a HAMMING code (Hamming(6,4)) which is able to correct one error using two redundant components.

Multiple comparison

Uncoded vs coded receiver $L = 2$**,** $L = 4$**,** $p = 5$

Figure 18: Uncoded vs Coded at the Receiver $p = 5$, $\pi = 2 + 1i$, $L = 2$ and $L = 4$, $n = 4$.

We can see how at the relay, the HAMMING coded version is better than the uncoded, however, as SNR goes up, the two behaviors are more and more similar. For instance, at 14dB the blue and yellow line (HAMMING coded $L = 2$ and uncoded $L = 2$, respectively) have almost the same behavior. The same happens for the red and magenta lines.
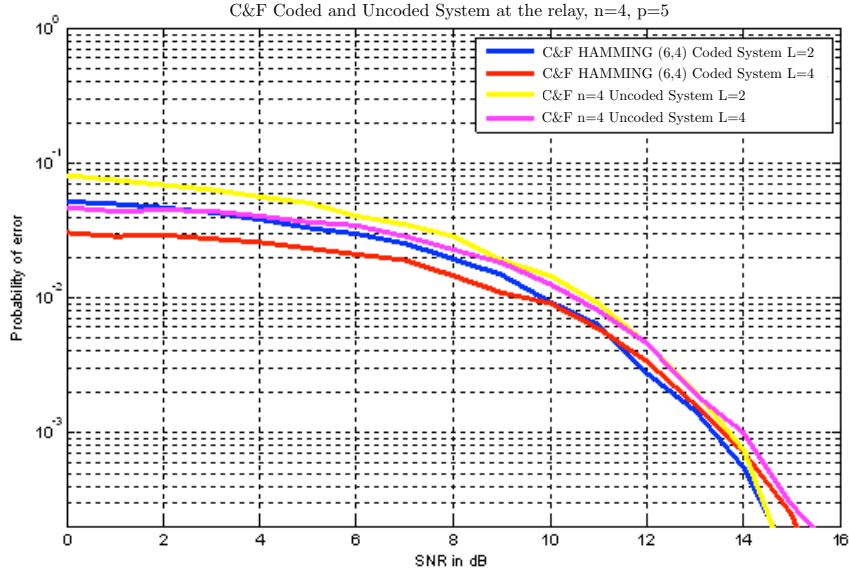
Uncoded vs coded relay $L = 2$**,** $L = 4$**,** $p = 5$
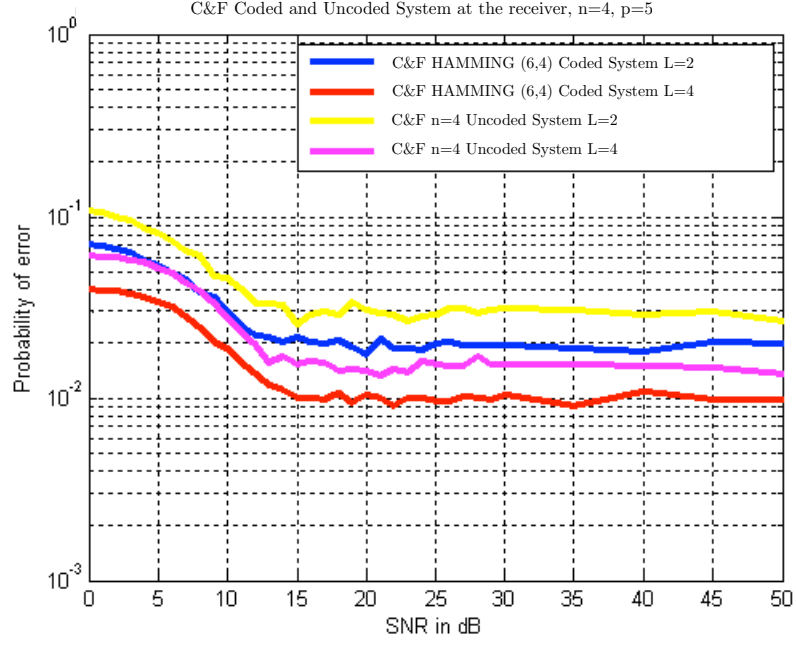
Figure 19: Uncoded vs Coded at the Relay $p = 5$, $\pi = 2 + 1i$, $L = 2$ and $L = 4$, $n = 4$.

We can observe that the HAMMING coded version of the system attains better performance than the uncoded, as expected. For example, the blue line (HAMMING coded $L = 2$) has a better behavior than the yellow line (uncoded $L = 2$) and the red line (HAMMING coded $L = 4$) is also better in terms of probability of error than the magenta line (uncoded $L = 4$).

We enclose the flow chart diagram of the C&F HAMMING Coded System.

Figure 20: Flow Chart Diagram. C&F Coded System.

10 Improvement of the Coefficients: Improved Matrix A

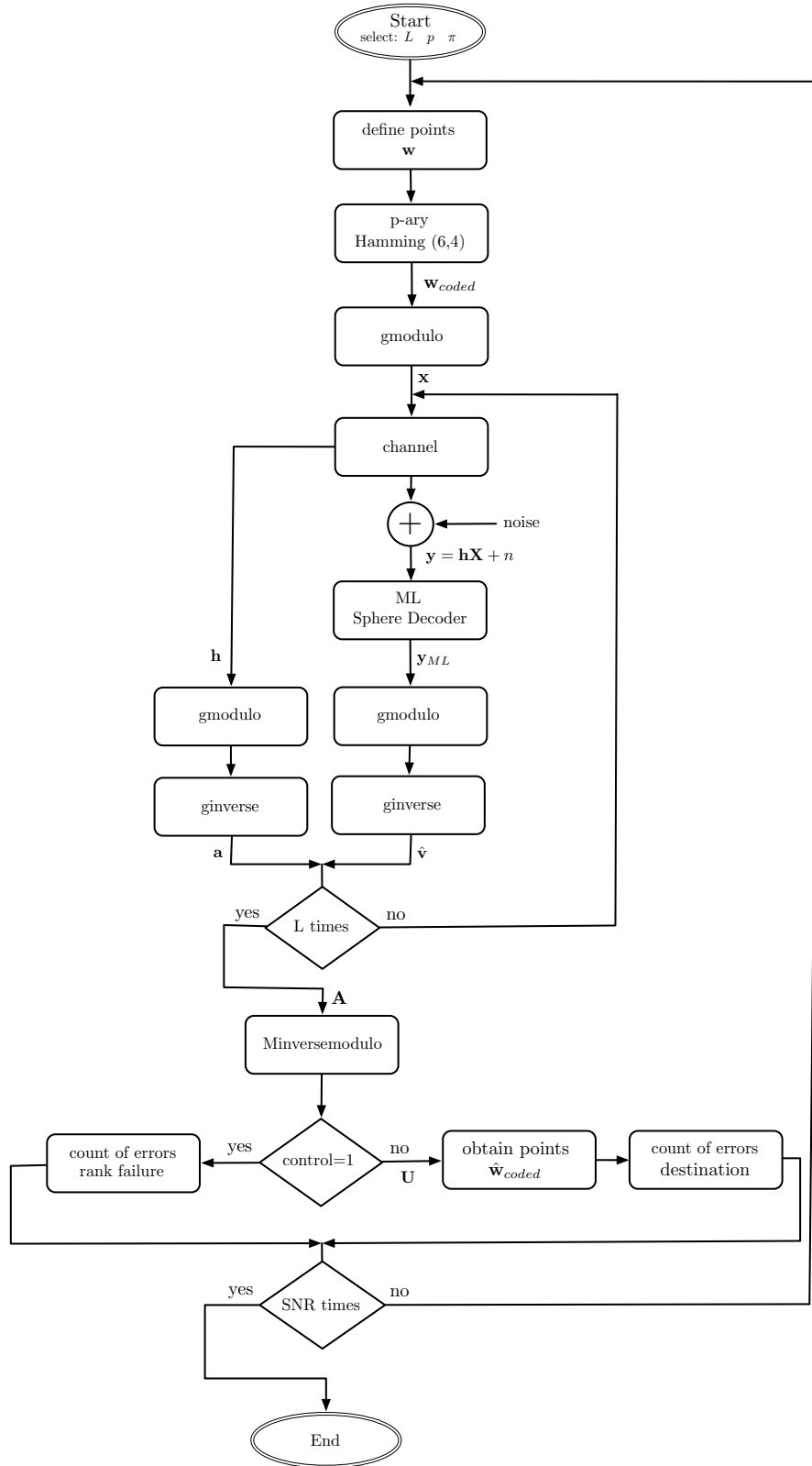The following sections will be aimed at improving the coefficient Matrix $\mathbf{A}$. The first approach described in this section is a simple yet intelligent idea to improve the overall performance of the system.

10.1 Construction

Given the original C&F system, one can see that there is a big number of errors due to rank failure. This is because $\det(\mathbf{A}) = 0 \bmod p$. The first idea that comes to mind is trying to avoid these rank failures by making the relay to wait till it has linearly independent equations and then find the original codewords. This is exactly the first approach followed in this project.

10.2 Performance
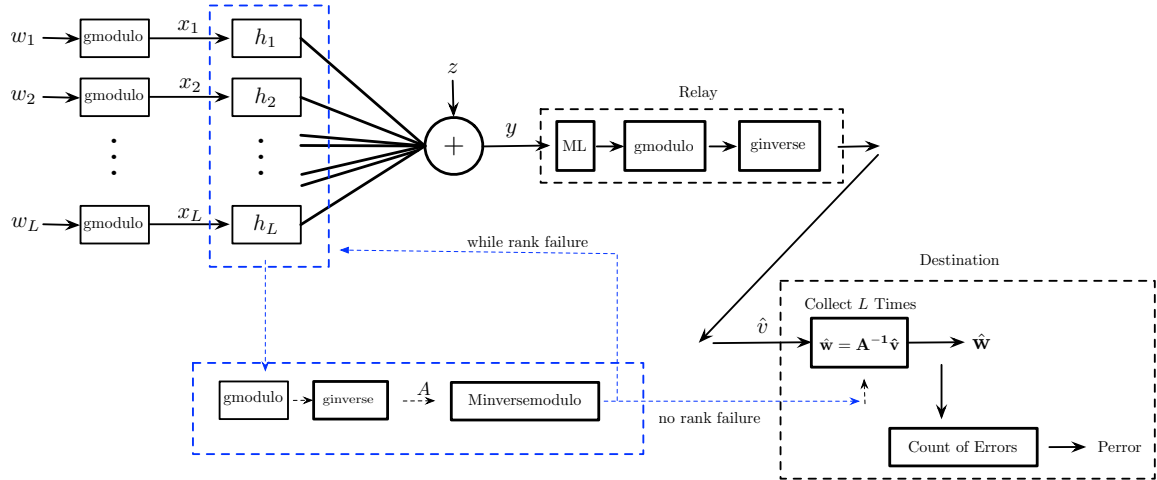
We can see the diagram of the C&F system implemented.

Figure 21: C&F Improved Matrix $\mathbf{A}$ System.

We can see the obtained results for the scalar case $L = 2$ and $p = 5$.
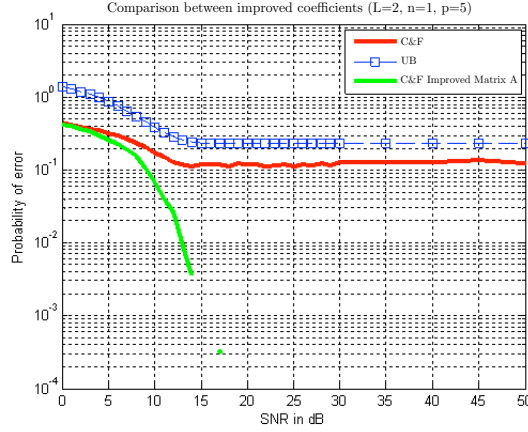
Figure 22: $L = 2$, $n = 1$, $p = 5$, Comparison between Improved Coefficients.

Here we can observe that there is significant improvement in terms of the probability of error. In fact, we force that there is no rank failure and so all the errors that remain are the errors in the relay, what means that we actually get the same curve of probability of error that is observed in the relay.

If we do the same for the vectorial case $L = 2$, $n = 2$, $p = 5$
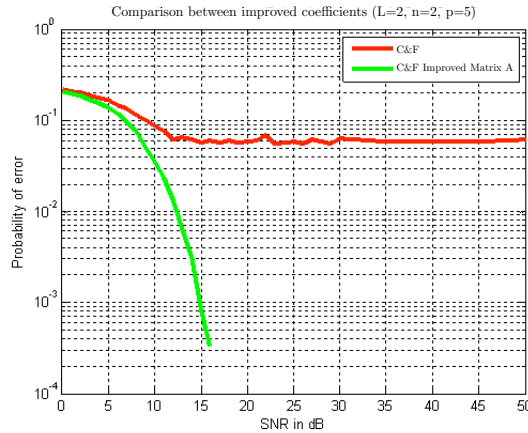
Figure 23: $L = 2$, $n = 2$, $p = 5$, Comparison between Improved Coefficients.

We can see almost the same behavior as the scalar case, here also the improvement is significant.

The question that arises next is to think if there is something better to do, and this is what will be considered in the next subsection.

We enclose the flow chart diagram for the C&F Improved Matrix **A** System.

Figure 24: Flow Chart Diagram. C&F Improved Matrix **A** System.

11 Improvement of the Coefficients: Optimum Matrix A

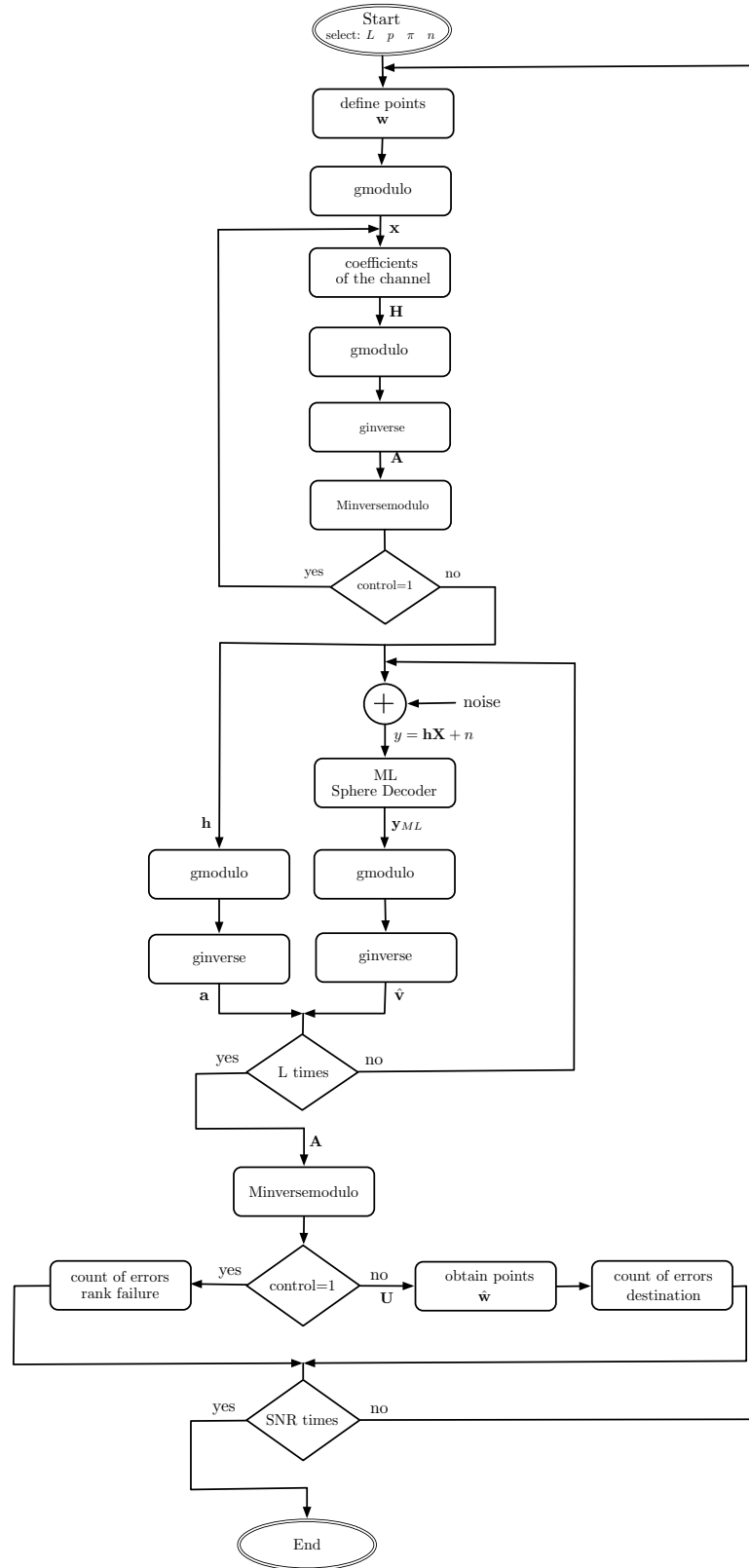The idea behind this section is based on the optimum coefficient algorithms proposed in Nazer and Gastpar [2011a] for the C&F System. Where the idea is to choose the optimal scaling factor β_m and optimal vector of coefficients a_m prescribed by the minimum criterion variance of effective noise.

11.1 Construction

We are going to propose an optimum strategy to find the matrix coefficients. However, it is proved in Wei [2012] that this strategy is optimum just when only one independent transmission is done (and one relay), and as we are using L independent transmissions (which is in fact the same as considering L relays), this method will give us optimum coefficients independently, but will not guarantee no rank failure.

The approach Compute and Forward described earlier (proposed in Nazer and Gastpar [2011a]) exploits the property that any integer combination of lattice points is again a lattice point. After receiving the noisy vector $\mathbf{y}_m$, the m-th relay will first select a scalar $\beta_m \in \mathbb{R}$ and a vector of coefficients in network coding

$$\mathbf{a}_m = [a_{m1}, a_{m2}, \ldots, a_{mL}]^T \in \mathbb{Z}^L \tag{13}$$

and then attempt to decode the lattice point $\sum_{l=1}^{L} a_{ml}\mathbf{x}_l$ from

$$\begin{aligned}
\beta_m \mathbf{y}_m &= \sum_{l=1}^{L} \beta_m h_{ml}\mathbf{x}_l + \beta_m \mathbf{z}_m, \tag{14} \\
&= \sum_{l=1}^{L} a_{ml}\mathbf{x}_l + \sum_{l=1}^{L} (\beta_m h_{ml} - a_{ml})\mathbf{x}_l + \beta_m \mathbf{z_m}. \tag{15}
\end{aligned}$$

Note that we do not need to conduct joint ML decoding to get $(\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_L)$ for network coding. Instead we decode $\sum_{l=1}^{L} a_{ml}\mathbf{x}_l$ as one regular codeword due to the lattice algebraic structure. In other words, the network coded codeword is still in the same field as original source codeword.

Our goal is to obtain the optimum matrix $\mathbf{A} = \begin{bmatrix} a_1^1 & a_2^1 & \ldots & a_L^L \\ \vdots & & & \vdots \\ a_1^L & a_2^L & \ldots & a_L^L \end{bmatrix}$ in order to recover the original codewords.

We are interested in the rate of $\sum_{l=1}^{L} a_{ml}\mathbf{x}_l$ as a whole and will capture the performance of the computation scheme by what we refer to as the computation rate, that is to say, the number of bits of the linear function successfully recovered per channel use. In Nazer and Gastpar [2011a] it is shown that a relay can often recover an equation of messages at a higher rate than any individual message. The rate is highest when the equation coefficients closely approximate the effective channel coefficients. The formal theorems can be found in Nazer and Gastpar [2011b] and Osmane and Belfiore [2011].

Theorem 12. *For AWGN real valued networks with vector of coefficients of the channel $\mathbf{h}_m \in \mathbb{R}^L$ and desired vector of coefficients $\mathbf{a}_m \in \mathbb{Z}^L$ in network coding, the following computation rate is achievable*

$$\mathscr{R}_m(\mathbf{a}_m) = \max_{\beta_m \in \mathbb{R}} \frac{1}{2} \log^+ \left(\frac{\text{SNR}}{\beta_m^2 + \text{SNR}\, ||\beta_m \mathbf{h}_m - \mathbf{a}_m||^2} \right). \tag{16}$$

Theorem 13. *The computation rate given in Equation ([16](#)) is uniquely maximized by choosing β_m to be the MMSE coefficient*

$$\beta_{MMSE} = \frac{\text{SNR} \, \mathbf{h}^T \mathbf{a}_m}{\text{SNR} \, ||h_m||^2 + 1},$$ (17)

which results in a computation rate of

$$\mathscr{R}_m(\mathbf{a}_m) = \frac{1}{2} \log^+ \left(||a||^2 - \frac{\text{SNR}(\mathbf{h}_m^T a_m)^2}{1 + \text{SNR} \, ||\mathbf{h}_m||^2} \right)^{-1}.$$ (18)

Theorem 14. *For a given vector of coefficients of the channel $\mathbf{h}_m = [h_{m1}, h_{m2}, \ldots, h_{mL}]^T \in \mathbb{R}^L$, $\mathscr{R}_m(\mathbf{a}_m)$ is maximized by choosing in network coding the vector of coefficients $\mathbf{a}_m \in \mathbb{Z}^L$ as*

$$\mathbf{a}_m = \arg \min_{\mathbf{a}_m \in \mathbb{Z}^L, \mathbf{a}_m \neq 0} (a_m^T G_m a_m),$$ (19)

where

$$\mathbf{G}_m = \mathbf{I} - \frac{\text{SNR}}{1 + \text{SNR} \, ||\mathbf{h}_m||^2 \mathbf{H}_m}$$ (20)

and $\mathbf{H}_m = [H_{cz}^{(m)}], H_{cz}^{(m)} = h_{mc} h_{mz}, 1 \leq c, z \leq L.$

Example 11.1. *Suppose $L = 2$, SNR=10dB, and $h_1 = [-4 \quad 0]$ and $h_2 = [1 \quad -4]$.*

Then,

$$G_{m1} = \begin{bmatrix} 0.062 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad G_{m2} = \begin{bmatrix} 0.9415 & 0.2339 \\ 0.2339 & 0.0643 \end{bmatrix}.$$

We want to solve the next SVP problem (which is a ILS problem)

$$\begin{aligned} \mathbf{a}_{m1} &= \arg \min_{\mathbf{a}_{m1} \in \mathbb{Z}^2, \mathbf{a}_{m1} \neq 0} a_{m1}^T G_{m1} a_{m1} \\ \mathbf{a}_{m2} &= \arg \min_{\mathbf{a}_{m2} \in \mathbb{Z}^2, \mathbf{a}_{m2} \neq 0} a_{m2}^T G_{m2} a_{m2}. \end{aligned}$$

How to solve this problem will be explained in detail in the next subsection, however, for the sake of understanding we suppose we are able to find the solution and we obtain

$$\begin{aligned} \mathbf{a}_{m1} &= \arg \min_{\mathbf{a}_{m1} \in \mathbb{Z}^2, \mathbf{a}_{m1} \neq 0} (a_{m1}^T G_{m1} a_{m1}) = [1 \quad 0] \\ \mathbf{a}_{m2} &= \arg \min_{\mathbf{a}_{m2} \in \mathbb{Z}^2, \mathbf{a}_{m2} \neq 0} (a_{m2}^T G_{m2} a_{m2}) = [0 \quad 1]. \end{aligned}$$

So, Matrix $\mathbf{A}_{optimum}$ is

$$\mathbf{A}_{optimum} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We can also find the optimum coefficients $\beta_{m1} = -0.2482$ and $\beta_{m2} = -0.2339$ and do $\beta_m\mathbf{y}$. So, using $\mathbf{A}_{optimum}$ and $\beta_m\mathbf{y}$ we are able to decode the original message.

In the latter example we have supposed we were able to solve the ILS problem (SVP), now we are going to explain in detail the algorithms involved in the computation.

11.2 Solving the ILS Problem

In this section we will show how to solve the ILS problem

$$\min_{z \in \mathbb{Z}^n} ||\mathbf{y} - \mathbf{B}\mathbf{z}||^2.$$

This problem is analogous to solving

$$\min_{z \in \mathbb{Z}^n} (\mathbf{y} - \mathbf{B}\mathbf{z})^T \mathbf{V}^{-1} (\mathbf{y} - \mathbf{B}\mathbf{z})$$

where $\mathbf{V} \in \mathbf{R}^{n \times n}$ is a matrix positive definite.

One can first compute the CHOLESKY factorization $\mathbf{V} = \mathbf{R}^T\mathbf{R}$, then solve two lower triangular linear systems $R^T\overline{y} = y$ and $R^T\overline{B} = B$.

As our real aim is to solve the SVP problem

$$\min_{z \in \mathbb{Z}^n} (\mathbf{z})^T \mathbf{V}^{-1} (\mathbf{z})$$

we use $B = -I_n$ and $y = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_n$ and therefore $\overline{B} = R^T \backslash B$ and $\overline{y} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_n$.

Finally the problem becomes

$$\min_{z \in \mathbb{Z}^n} ||\overline{\mathbf{y}} - \overline{\mathbf{B}}\mathbf{z}||^2. \tag{21}$$

The algorithm to solve the ILS problem from Equation (21) consists on two processes: lattice reduction and vector search. The purpose of the reduction process is to make the search process easier and more efficient. The reduction algorithm used in the reduction process is a modified version based on Zhou [2006] of the LLL algorithm. The search algorithm is based on the SCHNORR EUCHNER enumeration strategy found in Chang and Zhou [2011].

11.2.1 CHOLESKY Factorization

The CHOLESKY factorization is only defined for HERMITIAN positive definite matrices:

Definition 15. *A matrix $A \in C^{m \times m}$ is HERMITIAN positive definite if and only if it is Hermitian ($A^H = A$) and for all non-zero vectors $x \in C^m$ it is the case that $x^H A x > 0$. If in addition $A \in \mathbb{R}^{m \times m}$ then A is said to be positive definite.*

Theorem 15. *(CHOLESKY Factorization Theorem). Given a HERMITIAN positive definite matrix A there exists a lower triangular matrix L such that $A = LL^H$.*

The lower triangular matrix L is known as the CHOLESKY factor and can be interpreted as square root of a HERMITIAN positive definite matrix, and LL^H is known as the CHOLESKY factorization of A. It is unique if the diagonal elements of L are restricted to be positive real.

Example 11.2. *Suppose a matrix positive definite*

$$A = \begin{bmatrix} 0.0421 & -0.1916 \\ -0.1916 & 0.9617 \end{bmatrix}.$$

We can do the CHOLESKY factorization using the MATLAB command chol(A,'lower')

$$\begin{bmatrix} 0.0421 & -0.1916 \\ -0.1916 & 0.9617 \end{bmatrix} = \begin{bmatrix} 0.2053 & 0 \\ -0.9332 & 0.3015 \end{bmatrix} \begin{bmatrix} 0.2053 & 0 \\ -0.9332 & 0.3015 \end{bmatrix}^H.$$

11.2.2 LLL (Lenstra-Lenstra-Lovász) Reduction Algorithm

For a full column rank matrix $B \in \mathbb{R}^{m \times n}$, lattice basis reduction is to find a basis matrix $\overline{B}$ which is equivalent to B, and the column vectors of $\overline{B}$ are shorter than those of B according to some criteria.

One of the most widely used reductions is the LLL (Lenstra-Lenstra-Lovász) reduction. It has many applications, such as solving a shortest vector problem $\min_{x \in \mathbb{Z}^n \setminus \{0\}} ||Bx||_2$ or a closest vector problem $\min_{x \in \mathbb{Z}^n \setminus \{0\}} ||\mathbf{y} - B\mathbf{x}||_2$. We are interested in solving a shortest vector problem, which can be also referred as the ILS (Integer Least Squares) problem. When solving a SVP with a search process, the LLL reduction can be used as a preprocessing stage to make the search process more efficient.

Lenstra et al. [1982] suggested the criteria for the LLL reduction and also gave an algorithm to compute the reduction. Their motivation was to factor integer polynomials, so the algorithm assumes that the lattice is an integer lattice, that is every vector in the basis is an integer vector. For the applications such as communications or GPS, the given basis is not integer, consequently the LLL reduced matrix is not an integer. In Zhou [2006] was proposed a variant LLL algorithm which is specifically designed for real basis matrices, which is the reduction algorithm we are going to use to solve the SVP problem.

The original LLL algorithm was based on the GRAM SCHMIDT orthogonalization, however when floating point operations are to be used, the GRAM SCHMIDT orthogonalization should not be used since it may have numerical stability problems. Therefore, Zhou [2006] uses a QR factorization instead, proposing an alternative LLL algorithm for the LLL QRZ.

In order to introduce the criteria of the LLL reduction, it is necessary to orthogonalize the base vectors. The GRAM SCHMIDT orthogonalization process to make any two of the given basis vectors orthogonal to each other is the

following

$$\mathbf{b}_1^* = \mathbf{b}_1 \tag{22}$$

$$\mathbf{b}_z^* = \mathbf{b}_z - \sum_{c=1}^{z-1} u_{cz} \mathbf{b}_c^*, \qquad 2 \le z \le n, \tag{23}$$

where $u_{c,z}$ defined by

$$u_{c,z} = \frac{\mathbf{b}_z^T \mathbf{b}_c^*}{||\mathbf{b}_c^*||_2^2} \qquad 1 \le c < z \le n, \tag{24}$$

are called the GRAM SCHMIDT coefficients and $(\mathbf{b}_c^*)^T(\mathbf{b}_z^*) = 0 (c \ne z)$. It can be seen that $\mathbf{b}_z$ can be represented by a linear combination of $\mathbf{b}^*$, so this process gives an orthogonal basis $\{b_1^*, \ldots, b_n^*\}$.

Definition 16. *A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ for a lattice $\mathcal{L}$ is called LLL reduced if*

$$|u_{cz}| \le 1/2, \qquad 1 \le c < z \le n,$$
$$\delta ||\mathbf{b}_{c-1}^*||_2^2 \le ||\mathbf{b}_c^* + u_{c-1,c}\mathbf{b}_{c-1}^*||_2^2, \qquad 1 < c \le n. \tag{25}$$

The constant δ could be any real constant in $\left(\frac{1}{4}, 1\right)$.

In order to use matrix language to describe the LLL reduction, we first need to show the GRAM SCHMIDT orthogonalization is equivalent to QR factorization. Let $B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ where $\mathbf{b}_z^*$ are the orthogonal base vectors of B obtained by the GRAM SCHMIDT orthogonalization, and let U be a unit upper triangular matrix where its (c, z)-th element $(c < z)$ is defined by GRAM SCHMIDT coefficient $u_{c,z}$ given in Equation (24). B^* can be factorized into an orthonormal matrix Q_1, that is to say, $Q_1^T Q_1 = I$, and a diagonal matrix D:

$$Q_1 = \left[\frac{\mathbf{b}_1}{||\mathbf{b}_1||}, \ldots, \frac{\mathbf{b}_n}{||\mathbf{b}_n||}\right] \qquad D = diagonal(||\mathbf{b}_1||, \ldots, ||\mathbf{b}_n||). \tag{26}$$

If we now define $R = DU$, notice that R is an upper triangular matrix with positive diagonal entries. Then from the GRAM SCHMIDT orthogonalization process it is easy to verify

$$B = B^*U = Q_1 DU = Q_1 R. \tag{27}$$

This gives the QR factorization of B.

The LLL reduced condition in Equation (25) can be also expressed using the QR factorization.

Definition 17. *If B has QR factorization $B = Q_1 R$ then the matrix B is LLL reduced if*

$$|r_{cz}/r_{cc}| \le 1/2, \qquad 1 \le c < z \le n,$$
$$\delta r_{c-1,c-1}^2 \le r_{cc}^2 + r_{c-1,c}^2, \qquad 1 < c \le n, \tag{28}$$

where B has full column rank. The constant δ could be any real number in $\left(\frac{1}{4}, 1\right)$.

Now we want to cast the LLL reduction as a matrix factorization. Suppose $B \in \mathbb{R}^{m \times n}$ has full column rank, we refer to the following factorization as a QRZ factorization of B:

$$B = [Q_1, Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} Z = Q_1 R Z, \tag{29}$$

where $[Q_1, Q_2] \in \mathbb{R}^{m \times m}$ is orthogonal, $R \in \mathbb{R}^{n \times n}$ is upper triangular and $Z \in \mathbb{Z}^{n \times n}$ is unimodular, that is Z is an integer matrix and $|det(Z)| = 1$. We call it a LLL QRZ factorization if R is LLL reduced. It is obvious that if R is LLL reduced, $\hat{B} = Q_1 R$ is also LLL reduced, and vice versa.

For a given basis $\{\mathbf{b}_c, \dots, \mathbf{b}_n\}$ to achieve the LLL reduced criteria, there are two types of basic operations in the algorithm of computing the LLL reduction.

- Subtract one base vector times some integer from another, $\mathbf{b}_c := \mathbf{b}_c - t\mathbf{b}_z$, $t := \lfloor u_{cz} \rceil$.

- Interchange two nearby base vectors $\mathbf{b}_{c-1}$ and $\mathbf{b}_c$.

The first operation is to ensure that new $|u_{cz}| \leq 1/2$ after updating. The second operation is to meet the second criterion of the LLL reduction. After the permutation, $\mathbf{b}^*$ and u should be updated $\mathbf{b}_c^* := \mathbf{b}_c - \sum_{z=1}^{c-1} u_{cz} \mathbf{b}_z^*$, $u_{cz} := \frac{\mathbf{b}_c^T \mathbf{b}_z^*}{||\mathbf{b}_z^*||^2}$. With these two operations, we can describe the algorithm that transforms a given basis $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ into a LLL reduced one.

LLL algorithm for the LLL QRZ factorization

The algorithm is based on Zhou [2006]. The basic idea is the following: We first factorize $B = Q_1 R$ by the QR factorization. Then we apply size reductions to $R(:, 2)$ and check whether the LLL criterion is satisfied for $R(:, 1)$ and $R(:, 2)$. If the criterion holds, we go to the next column. Otherwise, we apply a column permutation and go back to the previous column if it exists. We continue in this way until the criterion holds for the last pair $R(:, n-1)$ and $R(:, n)$.

We are going to use the LLL algorithm found on the MILES MATLAB package.

11.2.3   SCHNORR EUCHNER Enumeration

This section is based on Chang and Zhou [2011].

After the reduction, a search strategy is used to enumerate possible $z \in \mathbb{Z}^n$

$$\min_{z \in \mathbb{Z}^n} ||\mathbf{y} - \mathbf{R}\mathbf{z}||^2. \tag{30}$$

Suppose that the optimal $\mathbf{z}$ satisfies the following bound

$$f(z) = ||\mathbf{y} - \mathbf{R}\mathbf{z}||^2 < \beta \tag{31}$$

or equivalently

$$\sum_{k=1}^{n}(y_k - \sum_{c=k}^{n} r_{kc}z_c)^2 < \beta. \tag{32}$$

This is an ellipsoid and our problem is to search this ellipsoid to find the optimal solution.

If we define

$$c_n = y_n/r_{nn}, \qquad c_k = (y_k - \sum_{v=k+1}^{n} r_{kv}z_v)/r_{kk}, \qquad k = n-1, \ldots, 1. \tag{33}$$

Notice that c_k depends on $z_n, z_{n-1}, \ldots, z_{k+1}$ and it is determined when the latter are determined.

Then Equation (32) can be rewritten as

$$\sum_{k=1}^{n} r_{kk}^2(z_k - c_k)^2 < \beta. \tag{34}$$

From this, it follows that

$$\text{level } n: \quad r_{nn}^2(z_n - c_n)^2 < \beta, \tag{35}$$

$$\vdots$$

$$\text{level } k: \quad r_{k,k}^2(z_k - c_k)^2 < \beta - \sum_{v=k+1}^{n} r_{vv}^2(z_v - c_v)^2, \tag{36}$$

$$\vdots$$

$$\text{level } 1: \quad r_{1,1}^2(z_1 - c_1)^2 < \beta - \sum_{v=2}^{n} r_{vv}^2(z_v - c_v)^2. \tag{37}$$

Based on these bounds a search procedure can be developed.

First, at level n we choose $z_n = \lfloor c_n \rceil$. If it does not satisfy the bound from Equation (35), no any integer will satisfy it, thus there is no integer point within the ellipsoid. This will not happen if the initial ellipsoid bound β is large enough. If it satisfies the bound, we proceed to level $n-1$. At this level we compute c_{n-1} by Equation (33) and choose $z_n = \lfloor c_{n-1} \rceil$. If z_{n-1} does not satisfy the bound with $k = n-1$, then move back to level n and choose z_n to be the second nearest integer to c_n, and so on; otherwise, we proceed to level $n-2$. We continue this procedure until we reach level 1 and obtain an integer point $\hat{z}$. We store this point and update the bound β by taking $\beta = ||\mathbf{y} - \mathbf{Rz}||^2$. Note that the ellipsoidal region is shrunk. Then we start to try to find an integer point within the new ellipsoid. The basic idea is to update the latest found integer point $\hat{z}$. Obviously, we cannot update only its first entry z_1, since at level 1, we cannot find any integer z_1 to satisfy Equation (37), which is now an equality. Thus we move up to level 2 to update the value z_2 by choosing z_2 to be the next nearest integer to c_2. If it satisfies the bound at level 2, we move down to level 1 to update the value of z_1 and obtain a new integer point (note that z_2 has just been updated and $z_3, \ldots, z_n$ are the same as those corresponding entries of $\hat{z}$), otherwise we

move up to level 3 to update the value of z_3, and so on. Finally, when we fail to find a new value for z_n to satisfy the bound from Equation (35) at level n, the search process stops and the latest found integer point is the optimal solution we seek.

The initial bound β is set to be ∞ and we refer to the first found integer point $\hat{z}$ as the BABAI integer point. The algorithm described finds only one optimal solution. How can we modify it in order to find p optimal solutions of the ILS problem? At the beginning we set β to be infinity. Denote the first integer point obtained by the search process (BABAI point) $\mathbf{z}^{(1)}$. Then we take the second integer point $\mathbf{z}^{(2)}$ to be identical to $\mathbf{z}^{(1)}$ except that the first entry in $\mathbf{z}^{(2)}$ is taken as the second nearest integer to c_1. The third $\mathbf{z}^{(3)}$ is chosen to be the same as $\mathbf{z}^{(1)}$ except that its first entry is taken as the third nearest integer to c_1, and so on. In this way we obtain p integer points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(p)}$. Obviously we have $f(\mathbf{z}^{(1)}) \leq \ldots \leq f(\mathbf{z}^{(p)})$. Then we shrink the ellipsoidal region by setting $\beta = f(\mathbf{z}^{(p)})$ and start to seach for a new integer point within the new ellipsoid. Suppose the new integer point we have found is $\mathbf{z}^{(new)}$ and $f(\mathbf{z}^{(v-1)}) \leq f(\mathbf{z}^{(new)}) \leq f(\mathbf{z}^{(v)})$. We remove the point $\mathbf{z}^{(p)}$ and rename $\mathbf{z}^{(new)}, \ldots, \mathbf{z}^{(p-1)}$ as $\mathbf{z}^{(v)}, \ldots, \mathbf{z}^{(p)}$, respectively. Then we shrink the ellipsoidal region again by setting $\beta = f(\mathbf{z}^{(p)})$ and continue the above process until we cannot find a integer point. Finally we end up with p optimal ILS points.

We are going to use the SCHNORR EUCHNER enumeration algorithm found in the MILES MATLAB package.

11.3 Performance
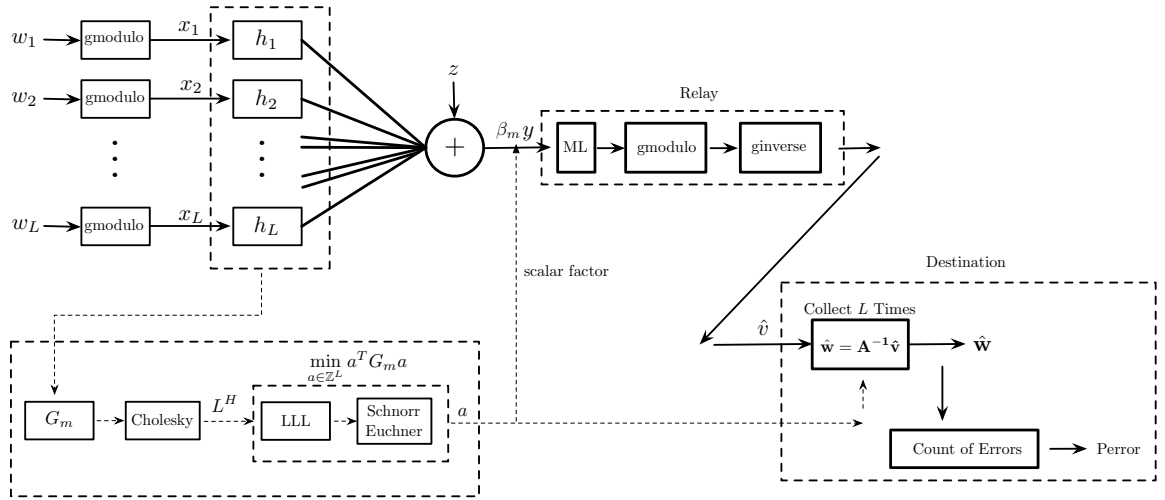
Here we can see the diagram of the C&F System implemented.

Figure 25: C&F Optimum Matrix $\mathbf{A}$ System.

If we do the simulation for the scalar case, using $L = 2$ and $p = 5$, we can see
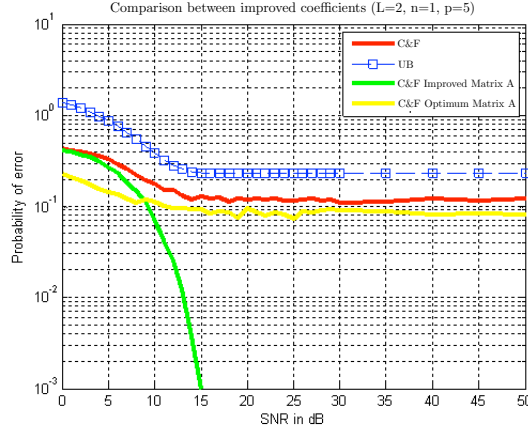
Figure 26: $L = 2$, $n = 1$, $p = 5$, Comparison between Improved Coefficients.

If we compare the Optimum Matrix **A** (yellow line) with the normal functioning (red line) of the system, we can see a slight improvement. However, as we have explained in the beginning of this section, this optimization does not guarantee full rank in independent transmissions, and therefore errors due to rank failure continue to affect the system.

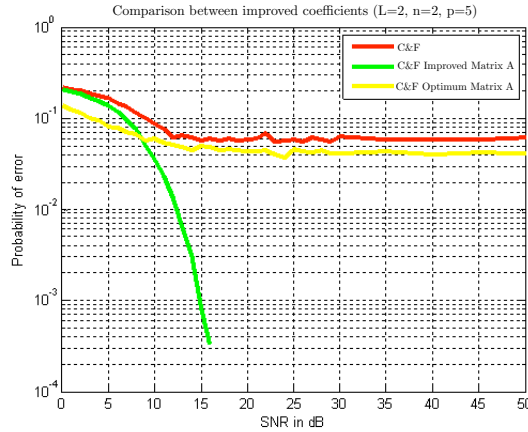If we do the simulation for the vectorial case, using $L = 2$, $n = 2$ and $p = 5$.

Figure 27: $L = 2$, $n = 2$, $p = 5$, Comparison between Improved Coefficients.

We can observe almost the same results as in the scalar case.

The next question that arises is if there is a way to solve the problems due to rank failure. The answer can be found in the next section.

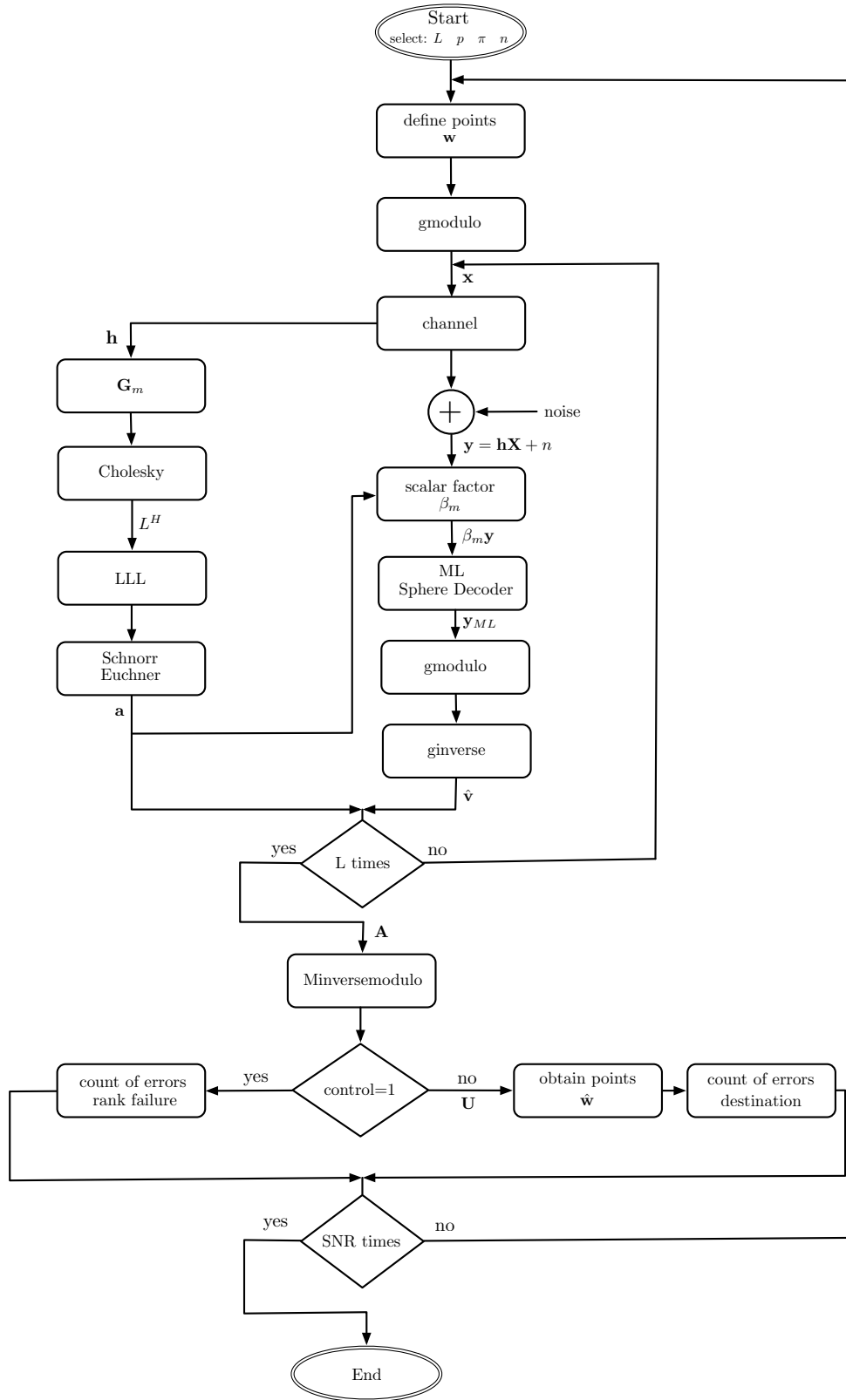We enclose the flow chart diagram of the C&F Optimum Matrix **A** System.

Figure 28: Flow Chart Diagram. C&F Optimum Matrix **A** System.

12 Improvement of the Coefficients: Improved Optimum Matrix A

The aim of this section is to improve the C&F system implemented in the latter section, improving the performance of the optimum coefficient algorithm used.

12.1 Construction

Given the method described in the earlier section, the idea is to think a strategy to obtain matrices full rank. The idea is the following: as in our system we do L transmissions, this is equivalent to have a L-relay system, which is in fact what we have. These relays will resend its information till the received coefficients are full rank. Using this idea we will obtain optimum coefficients with matrices full rank.

12.2 Performance

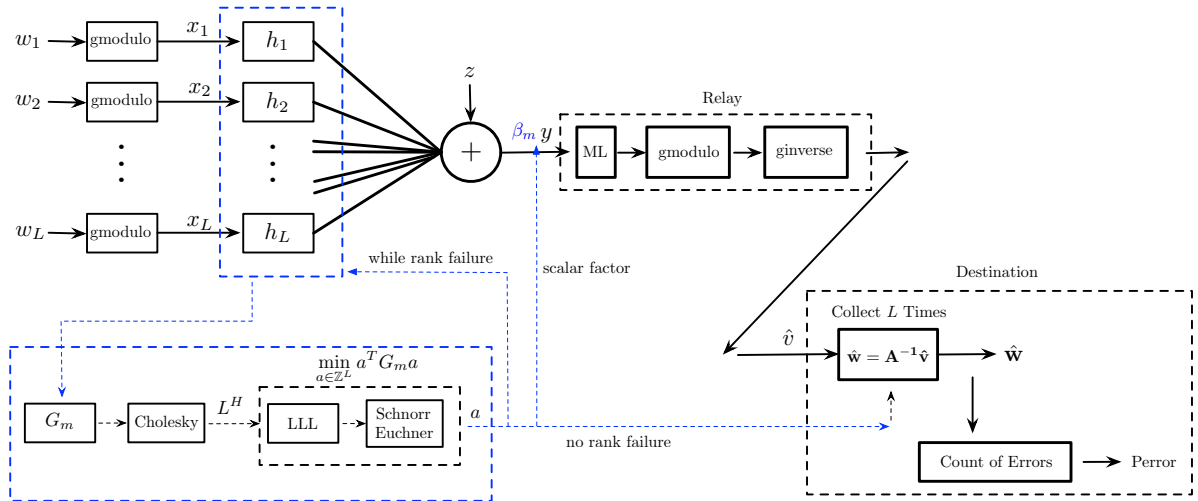In the next diagram, it can be seen the C&F system implemented

Figure 29: C&F Improved Optimum Matrix **A** System.

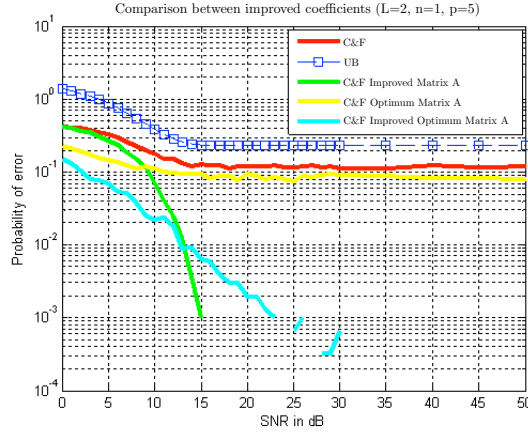If we simulate for the scalar case, using $L = 2$ and $p = 5$

Figure 30: $L = 2$, $n = 1$, $p = 5$, Comparison between Improved Coefficients.

we can see that for SNR low (below 12dB) the Improved Optimum Matrix $\mathbf{A}$ (cyan line) is the best method found. However, if SNR goes up, the first approach used, the Improved Matrix $\mathbf{A}$ (green line), attains better results. In the plot we can observe how both Improved Optimum Matrix $\mathbf{A}$ and Improved Matrix $\mathbf{A}$ go down with SNR, but the second has a steepest slope.

If we simulate for the vectorial case, using $L = 2$, $n = 2$ and $p = 5$.
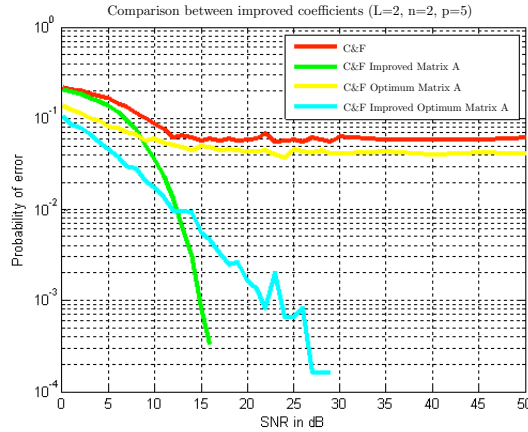
Figure 31: $L = 2$, $n = 2$, $p = 5$, Comparison between Improved Coefficients.

Here we can observe exactly the same behavior that in the scalar case. The cyan line (Improved Optimum Matrix $\mathbf{A}$) attains better performance under 12dB but the green line (Improved Matrix $\mathbf{A}$) has better results for SNR high. The two methods decrease till they reach zero errors with SNR increasing.

We enclose the flow chart diagram for the C&F Improved Optimum Matrix $\mathbf{A}$ System implemented.

Figure 32: Flow Chart Diagram. C&F Improved Optimum Matrix **A** System.
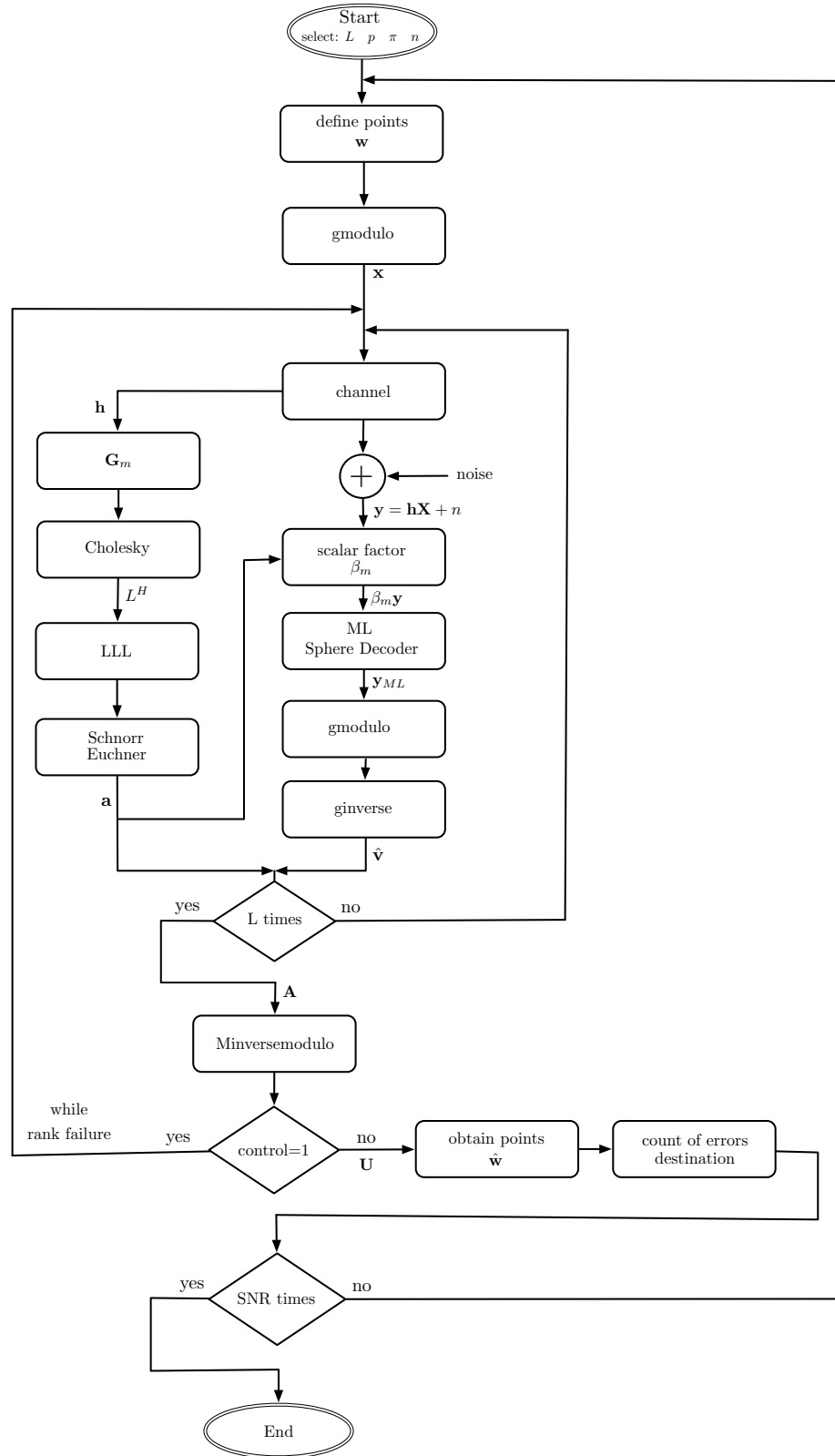
13 Conclusions and Further Work

In this project we have done a survey of the mathematical theory needed to understand the C&F system. Both an introduction to Physical-layer Network Coding and lattice network codes have been presented.

The work has involved the construction and performance of the system Compute and Forward with increasing complexity. We started implementing a scalar case version of the C&F System with L antennas. This section has been the common base framework for future designs. Next, we have implemented a C&F Vectorial System. Further, a C&F HAMMING q-ary coded system has been studied. The basic theory of linear codes and HAMMING q-ary codes have been exposed.

The next step has involved the improvement of the coefficient matrix $\mathbf{A}$. We proposed and implemented an easy yet intelligent idea to avoid rank failure. Next, the optimum algorithm found in the literature has been studied and implemented. Concepts related to solving a ILS (Integer Least Squares) problem using lattice reduction and vector search have been explained in detail. Finally, an improved version of the optimum algorithm has been developed.

The results obtained show that the first approach used, the algorithm Improved Matrix $\mathbf{A}$, attains the best performance for SNR high. However, is slow in terms of time computing resources. The algorithm Improved Optimum Matrix $\mathbf{A}$ has good performance and is the best for SNR low. Moreover, it has better time computing performance. Both algorithms reach zero errors with SNR increasing, which can be easily understood because they are designed in order to avoid rank failure, which is the main cause of error for SNR high.

Future work in the field can gear towards implementing an optimum algorithm for independent transmissions, that is to say, an optimum algorithm for a n-relay system. Another interesting path to follow would be to use a different lattice network code such as the lattice $\mathbb{Z}[w]$, which has been proved that attains better performance than the lattice $\mathbb{Z}[i]$.

References

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

J. C. Belfiore and C. Ling. 2011. The Flatness Factor in Lattice Network Coding: Design Criterion and Decoding Algorithm. IEEE Transactions on Information Theory. 48

X. W Chang and T. Zhou. 2011. MILES: MATLAB Package for Solving Integer Least Squares Problems. Theory and Algorithms. McGill. 70, 73

K. Conrad. 2013. Expository Papers. University of Connecticut. 24

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

I. de Zarzà, M. A. Vázquez, and J. M. Mondelo. 2013. Physical-layer Network Coding: Design of Constellations over Rings. Universitat Autònoma de Barcelona. 32, 36, 37, 43

C. Feng, D. Silva, and F. R. Kschischang. 2011. An Algebraic Approach to Physical-layer Network Coding. IEEE Transactions on Information Theory. 25

S. Gupta and M. A. Vázquez. 2012. Compute and Forward: End to End Performance over Residue Class Based Signal Constellation. arXiv:1212.3289. 32, 36, 37, 43

K. Huber. 1994. Codes over GAUSSIAN Integers. IEEE Transactions on Information Theory. 32

A. K. Lenstra, H. W. Lenstra, and L. Lovász. 1982. Factoring Polynomials with Rational Coefficients. Mathematische Annalen. 71

H. Minkowski. 1896. Geometrie der Zahlen. Teubner. 24

H. Minkowski. 1907. Diophantische Approximationen. Teubner. 24

B. Nazer and M. Gastpar. 2011a. Compute and Forward: Harnessing Interference through Structured Codes. IEEE Transactions on Information Theory. 22, 25, 68

B. Nazer and M. Gastpar. 2011b. Reliable Physical-layer Network Coding. Proceedings of the IEEE. 68

A. Osmane and J. C Belfiore. 2011. The Compute-and-forward Protocol: Implementation and Practical Aspects. IEEE Communications Letters. 68

Q. T. Sun, J. Yuan, T. Huang, and K. W. Shum. 2013. Lattice Network Codes Based on EISENSTEIN Integers. IEEE Transactions on Communications. 25, 37

W. C. Waterhouse. 1987. How Often Do Determinants over Finite Fields Vanish? Discrete Mathematics. 38

L. Wei. 2012. Network Coding Design in Wireless Cooperative Networks. 68

L. Wei and W. Chen. 2012a. Compute-and-forward Network Coding Design over Multi-source Multi-relay Channels. IEEE Transactions on Wireless Communications. 22

L. Wei and W. Chen. 2012b. Efficient Compute-and-forward Network Codes Search for Two-way Relay Channels. IEEE Communications Letters. 22

S. Zhang, S. C. Liew, and P. P. Lam. 2006. Hot Topic: Physical-layer Network Coding. ACM MobiCom. 18

T. Zhou. 2006. Modified LLL Algorithms. McGill. 70, 71, 73

Resum:

El principal objectiu d'aquest treball és implementar i exposar una descripció teòrica per a diferents esquemes de Physical-layer Network Coding. Utilitzant un esquema bàsic com a punt de partida, el projecte presenta la construcció i l'anàlisi de diferents esquemes de comunicació on la complexitat va augmentant a mesura que anem avançant en el projecte.

El treball està estructurat en diferents parts: primer, es presenta una introducció a Physical-layer Network Coding i a Lattice Network Codes. A continuació, s'introdueixen les eines matemàtiques necessàries per entendre el sistema de Compute and Forward (C&F). Després, s'analitza i implementa el primer esquema bàsic. A partir del qual, implementem una versió vectorial del C&F System i una versió codificada amb un HAMMING q-ari. Finalment, s'estudien i implementen diferents estratègies per millorar la matriu de coeficients A.

Resumen:

El principal objetivo de este trabajo es implementar y exponer una descripción teórica para diferentes esquemas de Physical-layer Network Coding. Utilizando un esquema básico como punto de partida, el proyecto presenta la construcción y el análisis de distintos sistemas de comunicaciones dónde la complejidad va aumentando a medida que avanzamos en el proyecto.

El proyecto está estructurado en diferentes partes: primero, se presenta una introducción a Physical-layer Network Coding y a Lattice Network Codes. A continuación, se introducen las herramientas matemáticas necesarias para entender el sistema de Compute and Forward (C&F). Lo siguiente es analizar y implementar el primer esquema básico. A partir del cual, implementamos una versión vectorial del C&F System y una versión codificada con un HAMMING q-ario. Finalmente, se estudian y implementan diferentes estrategias para mejorar la matriz de coeficientes A.

Summary:

The main goal of this work is to implement and provide a theoretical description for different schemes of Physical-layer Network Coding. Using a basic scheme as starting point, the project presents the construction and performance of different systems of communications with increasing complexity.

The project is structured in different parts: first, an introduction to Physical-layer Network Coding and Lattice Network Codes is done. Next, the mathematical tools needed to understand the system of Compute and Forward (C&F) are presented. Further, the first basic scheme is analysed and implemented. The next step consists on implementing a vectorial C&F System and a HAMMING q-ary coded version. Finally, different approaches to improve the matrix coefficient A are studied and implemented.

J. de Curtò i DíAz

GENERAL
INFORMATION

E-mail: c@decurto.be
Webpage: https://www.decurto.be

EDUCATION

Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona).

5-year Degree in Engineering of Telecommunication, Second Cycle. 2011 - 2013. Specialization in Communications, Signal Processing and Microwave Engineering. School of Engineering.

Thesis: Construction and Performance of Network Codes.
Grade: Excellent. First Class with Distinction.

Universitat Politècnica de Catalunya (UPC). Barcelona.

5-year Degree in Engineering of Telecommunication, First Cycle. 2006 - 2009.

University Entrance Examination.
Average Grade: 9.22/10. 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.

- University scholarship for an outstanding academic performance. Technological Baccalaureate. Government of Salou. This award is given to the top students in each graduation year by the local authority.

Technological Baccalaureate. 2004 - 2006.
Average Grade: 9.7/10. First Class Degree and Honorary Scholarship.

Academic Distinctions:

- Outstanding Thesis of Research. Development and Design of a Virtual Shop in Visual Basic on the .NET Framework.

- Outstanding Curriculum.

PUBLICATIONS

De Curtò i DíAz, De Zarzà i Cubero and Vázquez.
Secure Network Coding: Overview and State-of-the-art.
Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona). 2012.
https://blogs.uab.cat/curto/files/2019/05/nc_decurto12.pdf

De Curtò i DíAz, Moreno, Torrellas, Bofill and Muñoz.
Dear New Student: A Comparison between a Frontal and an Active Approach.
ALE 2007. Toulouse.

DISSERTATION

5-year Degree in Engineering of Telecommunication.
Construction and Performance of Network Codes.
Supervisor: Vázquez.
Universitat Autònoma de Barcelona. Cerdanyola del Vallès (Barcelona). 2013.
https://blogs.uab.cat/curto/files/2019/05/pfc_decurto.pdf
https://blogs.uab.cat/curto/files/2019/05/slides_pfc_decurto.pdf

LANGUAGES

English -

TOEFL Internet Based test. 26-11-2016. **Score 114/120.**

First Certificate in English. December 2005. **Grade: A.**

CAREER

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Research Scientist. April 2010 - June 2010.

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Internship. January 2010 - February 2010.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament de Teoria del Senyal i Comunicacions. September 2009 - December 2009.

CELLS ALBA Synchrotron Facility. Cerdanyola del Vallès (Barcelona).
Internship. July 2009 - September 2009.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament d'Arquitectura de Computadors. 2008 - 2009.

Universitat Politècnica de Catalunya (UPC). Barcelona.
Teaching Assistant. Departament d'Arquitectura de Computadors. 2006 - 2007.

SERVICES

First European Training School in Network Coding: Random Network Coding and Designs over $GF(q)$. IEEE Information Theory Society. Universitat Autònoma de Barcelona. Cerdanyola del Vallès (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.

. Barcelona. 18 - 22 July 2008.

Scientific Volunteer.

EXTRACURRICULAR
ACTIVITIES

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

Course in Investment and Financial Markets. Barcelona. 18 April 2011 - 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++, MATLAB, HTML, VHDL and Assembly.

SOFTWARE

LaTeX, Maple, PSpice and ADS.

DE CURTÒ I DÍAZ Joaquim.
Cerdanyola del Vallès (Barcelona), 2013.