UAB

Projecte Fi de Carrera

Enginyeria de Telecomunicació

Construction and performance of Network Codes

Joaquim Curto Díaz

Director: María Ángeles Vázquez Castro

Departament de Telecomunicació i Enginyeria de Sistemes

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

Juliol 2013

UAB

El sotasignant, *María Ángeles Vázquez Castro*, Professor de l'Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumne *Joaquim Curto Díaz*.

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

Bellaterra, *16 de Juliol de 2013*.

Signatura: *María Ángeles Vázquez Castro*

Performance and Construction of Network Codes

End of Degree Project

Author: Joaquim Curto Díaz
Advisor: M. Ángeles Vázquez Castro
July 2013, Bellaterra

Autonomous University of Barcelona

This work is dedicated to my beloved grandmother and my dear grandfather,
to my inspiring parents,
to my second family Carmen, Fermin and Filo
and to Irene, for her unconditional love, endless support and encouragement.

Acknowledgements

One of the joys of completion is to look over the journey past and remember those who have helped and supported me along this long but fulfilling road.

First, I would like to express my sincere gratitude to my advisor Dra. M. Ángeles Vázquez Castro, whose expertise, understanding, and patience, helped me during all this thesis. Her knowledge and support were invaluable to overcome many crisis situations and finish this dissertation.

I would like to honor the memory of my grandmother, Pilar Bel, who passed away three years ago. She was a woman with just a basic education, who had bigger dreams for her grandchild's. Without her financial assistant to both my brother and me, our college education could not have been realized. I will always remember her strong purpose in life and determination. I wish that she could be present for my graduation, but I know that she will be there in spirit.

I also would like to honor my grandfather, Manuel Díaz, who passed away one month prior to this project defense. He too had little formal education, but he knew much about life. I will always remember him as a tenacious hard worker, who always had in mind a new project to do.

A very special thanks goes out to all those people who have supported me during all this journey, in particular to Carmen, Fermin and Filo for your kindness and for taking care of me.

I am deeply and forever indebted to my parents for the support and encouragement throughout my entire life. You have sacrificed your lives for my brother and me and provided unconditional love and care. I would never have been able to finish this journey without your inspirational example and help. This project is yours.

Last but not least, I would like to thank Irene. There are no words to say how much I love you. You have been a true supporter and have unconditionally loved me during good and bad times. We know that these past several years have not been an easy ride, both academically and personally. We learned a lot about life and overcome difficult moments. *Thank you for all, I love you!*

Contents

List of Figures

1 Rationale and objectives

The aim of this project is to implement and provide a theoretical description of different Physical Layer Network Coding schemes. 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 analysed. 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 in [4].

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 Network Coding operation 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 [7] as a way to exploit the network coding operation 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 network coding operation 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 [7], Physical Layer Network Coding was an attempt to turn the situation around. By exploiting the network coding operation 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 strategy sends a linear combination $\mathbf{v}$ of this 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 succesfully 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

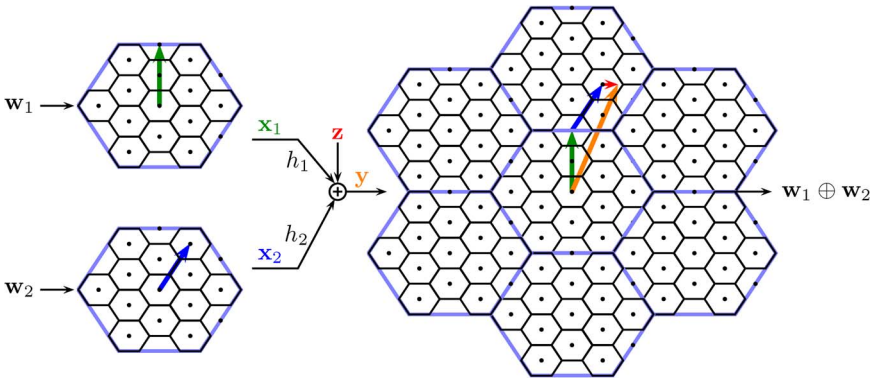


Figure 1: PNC Example

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 The relay MAC 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 strategies:

- The amplify-and-forward strategy allows the relay station to amplify the received signal from the source node and to forward it to the destination station.
- Relays following the decode-and-forward strategy 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 compress-and-forward strategy 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 compute-and-forward strategy 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 its channel observation. This decreases the effective noise encountered in the next decoding step.

In this project, we are going to focus on the Compute and Forward strategy CF, first proposed by Nazer and Gastpar in [5]. 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 Nazer-Gastpar compute-and-forward, algorithms are designed in [11] and [12] to find optimal coefficient vectors in terms of maximizing the transmission rate.

In this project, we will first focus on the study of the uncoded scalar CF system, next we will proceed to implement a vectorial version of the system. The next step will be to use a q-ary Hamming (6,4) to implement a coded CF 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 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 Bethout's 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 ([13]and [14]). 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 an n-dimensional space.

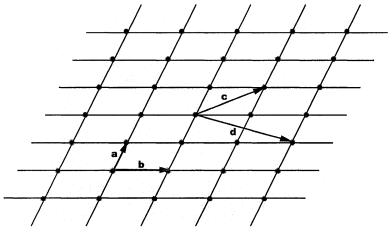


Figure 2: 2 dimensional lattice

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 $B = [\mathbf{b}_1 \quad \dots \quad \mathbf{b}_n] \in \mathbb{R}^{m\times n}$ 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 $S = \mathbf{b}_1 \quad \dots \quad \mathbf{b}_n$ 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 $\overline{B} = [\overline{\mathbf{b}}_1 \quad \dots \quad \overline{\mathbf{b}}_n] \in \mathbb{R}^{m\times n}$ 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 $\mathbb{Z}[w]$. An 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 reference [3].

The compute-and-forward scheme based on nested lattice codes was first proposed in [5]. Later a more general algebraic model, called lattice network coding, was developed in [8]. 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 an N-dimensional R-lattice if it forms an 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 $\Lambda = \{\sum_{1 \leq j \leq N} r_j \mathbf{b}_j : r_j \in R \quad \forall j\}$. A subset Λ' of Λ is called a sublattice of Λ if it is an R-module.*

Given an 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 an R-module. As an example, consider the PID of $\mathbb{Z}$ of integers, 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 norm multiplicative Theorem[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. ***(Division Theorem)****. 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 division theorem 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 division theorem 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 division theorem in $\mathbb{Z}$) rather than the closest integer from the left: 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 division theorem in $\mathbb{Z}[i]$ and the usual division theorem 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 division algorithm 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 division theorem, 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:*

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

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 division theorem), 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 Bezout's Theorem

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

Theorem 3. ***(Bezout's 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 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(α) 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:

$$2 = 1^2 + 1^2$$
$$5 = 1^2 + 2^2$$
$$13 = 2^2 + 3^2$$

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 just a commutative ring in which every nonzero element has an inverse.*

Definition 10. *An ideal in a ring* R *is a nonempty subset* J *of* R *satisfying:*

- $a - b \in J$ *for all* $a, b \in J$ *(closed under substraction)*
- ra *and* ar *are all in* J*, for all* $a \in J$*,* $r \in R$ *(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 Uncoded CF system model: scalar case

In this section, we are going to study the CF system model, where the messages w_j 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 encoding and decoding functions involved in the lattice network code. Next, some brief concepts about Maximum Likelihood detection will be introduced. Further, the CF system model will be presented and studied in detail. Finally, a derivation of the estimate probability of error of the system will be done.

7.1.1 Introduction

Most of the material in this section is based in the CF scheme studied by S. Gupta and M.A. Vázquez Castro in [1]. This information is complemented by the encoding and decoding mappings first explained by Huber in [2]. 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 $\mathbb{Z}/p\mathbb{Z} \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$, 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 modulo function $\psi : \mathbb{Z}[i] \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ is defined according to

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

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

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

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

note that in this equation we multiply up and down for π^* in order to get the $N(\pi)$ in the denominator, and $[\cdot]$ is the gaussian integer rounding defined earlier.

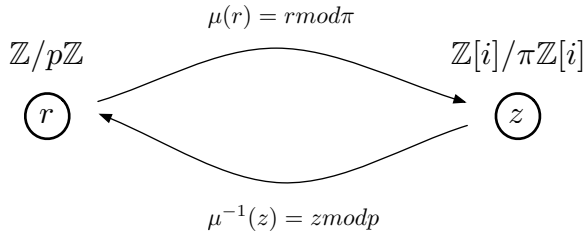
And if we solve for gamma (the residue) we get

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

Therefore,

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

Based on the previous modulo function, in order to build the CF 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 modulo function $\mu : \mathbb{Z}/p\mathbb{Z} \to \mathbb{Z}[i]/\pi\mathbb{Z}[i]$ 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 Bezout's 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 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 with z. We can write it as

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

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 Bezout's Theorem stated in equation (2)

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

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

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

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

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

Therefore, we can define the inverse function as the p-modulo function 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 $\bmod p$ function defined above:

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

Thus, we have defined the inverse function.

Once we have all mappings defined we are prepared to study the CF system model. However, first we are going to introduce some basic concepts about ML detection.

7.1.2 Maximum Likelihood Detection

If we consider the linear MIMO 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 communication channel, which is normally modeled as a linear system followed by an additive noise vector.

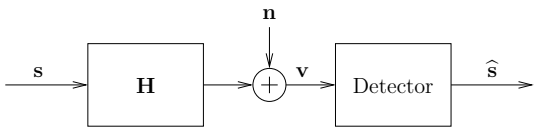


Figure 3: Simplified linear MIMO communication system diagram

We can observe in figure 3, a simplified MIMO 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:

$$s_* = arg \max_{\mathbf{s} \in \mathcal{X}^M} P(\mathbf{s} \textit{ was sent } | \mathbf{v} \textit{ is observed}) \tag{5}$$

$$= arg \max_{\mathbf{s} \in \mathcal{X}^M} \frac{P(\mathbf{v} \textit{ is observed } | \mathbf{s} \textit{ was sent}) P(\mathbf{s} \textit{ was sent})}{P(\mathbf{v} \textit{ is observed})} \tag{6}$$

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

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

A detector that always returns an optimal solution satisfying equation 7 is called Maximum Likelihood (ML) detector. If we assume that the additive noise $\mathbf{n}$ is white and Gaussian, when we can express the ML detection problem as the minimization of the squared Euclidean distance metric to a target vector $\mathbf{v}$ over and 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 sphere decoder methods to obtain ML or near-ML estimations.

7.1.3 System model

We consider the Compute and Forward system model with L sources, a relay and a destination, proposed by S. Gupta and M.A. Vázquez Castro in [1].

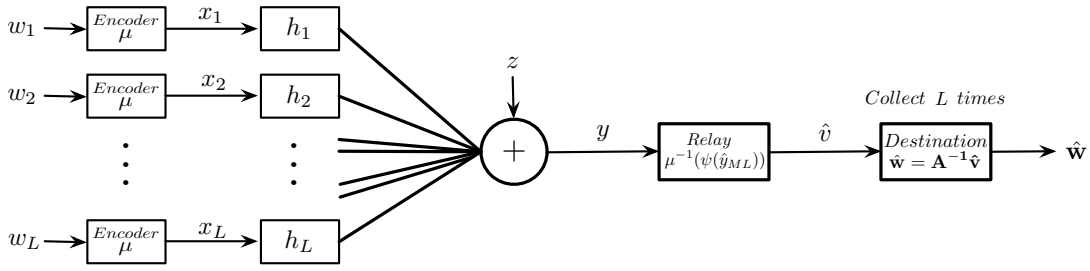


Figure 4: CF system model

Let $\omega_l \in \mathbb{F}_p$ be the message to be transmitted by the l-th source chosen from a finite field $\mathbb{F}_p$. The vector of all source messages is given by $\mathbf{w} = [w_1 \dots w_L]$. Each source encodes the message w_l into a complex signal constellation point 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:

$$\mu(w_l) = w_l \bmod \pi = w_l - \left\lfloor \frac{w_l \pi^*}{\pi\pi^*} \right\rceil \pi$$

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

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

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 maximum likelihood (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 Error probability

In this section we propose an analytical expression for probability of error estimate at the destination given the described system. We are going to take as a reference the Union Bound proposed in reference [1]. 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 reference [3]).

The probability of error at the destination is defined as $Pr(\hat{w} \neq w)$. From reference [1] we can see a theoretical expression for error probability using the union bound on the given system:

Theorem 10. *The union bound 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

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

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 estimate

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

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, $P_{dest} = Pr(|A| = 0)$. An error can occur due to rank failure and/or due to error at the relay.

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

$$P_{error} \leq 1 - P_{noerrorDest} \cdot P_{noerrorRelayL}$$

where

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

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, $P_{noerrorDest}$. This partial result will be based on reference [9]. 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, $P_{|A|\neq 0}$, can be computed using the probability formula:

$$P_{|A|\neq 0} = \frac{\#\ \textit{possible outcomes}}{\#\ \textit{total outcomes}}$$

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 substract 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

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

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

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

Finally,

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

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

[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 substract 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.

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

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

$$P_{noerrorDest} = \prod_{i=1}^{L} \left(1 - \frac{1}{p^i}\right)$$

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 maximum likelihood (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,

$$P_{errorRelay} = Pr(\hat{y}_{ML} \neq h_1x_1 + h_2x_2 + \ldots + h_Lx_L)$$

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 doesn't exceed the decision threshold is the probability that the noise norm distribution doesn't exceed $1/2$.

$$P_{noerrorRelay} = Pr(||z|| \leq 1/2)$$

where $P_{noerrorRelay}$ 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

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

where $P_{noerrorRelayL}$ 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 gaussian distribution $z \sim \mathcal{CN}(0, \sigma^2)$. The norm of a circular symmetric gaussian distribution follows a Rayleigh distribution. This supposition 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

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

Finally, the probability of error estimate 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

$$
\begin{aligned}
P_{error} &\leq 1 - P_{noerror} = 1 - P_{noerrorRelayL} \cdot P_{noerrorDest} \\
P_{error} &\leq 1 - \left(1 - e^{-\frac{1}{8\sigma^2}}\right)^L \cdot \prod_{i=1}^{L} \left(1 - \frac{1}{p^i}\right)
\end{aligned}
$$

□

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 gaussian distribution 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:

$$
SNR = \frac{\textit{Average signal power}}{\textit{noise power}} \Rightarrow \textit{noise power} = \frac{\textit{Average signal power}}{\textit{SNR}} = \sigma^2
$$

We can calculate the SNR measured in dB's as

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

The Average signal power of the constellation can be calculated as

$$\textit{Average signal power} = \frac{1}{p}\sum_{i=1}^{p} x_i x_i^* \tag{9}$$

and therefore

$$\textit{Noise power} = \frac{\frac{1}{p}\sum_{i=1}^{p} x_i x_i^*}{SNR_{Lineal}} \tag{10}$$

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 A in p-modulo.

First, we need to know if the output matrix A is invertible. An 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 following 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 greatest common divisor function. Where we use the fact that if $det \neq 0 \bmod p$ the greatest common divisor between the determinant and p is either 1 or -1. The procedure can be understood using $gcd = u \cdot det + v \cdot p$, then $gcd \mod p = u \cdot det \mod p$, where we can see that u is the multiplicative inverse we are looking for (except for a unit factor).

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

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

7.2.1 $L = 2$ CF System Implementation

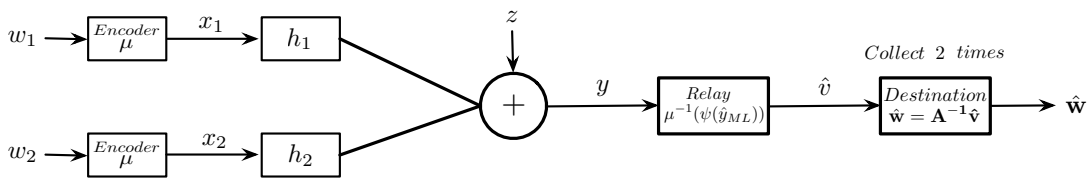


Figure 5: CF system model $L = 2$

We have generated a function system which gives us the probability of error and the Union Bound probability of error given p, π and L. This is a first basic approach to the system implementation, where the ML estimation has

been simplified as a rounding operation to the nearest point in the lattice.

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

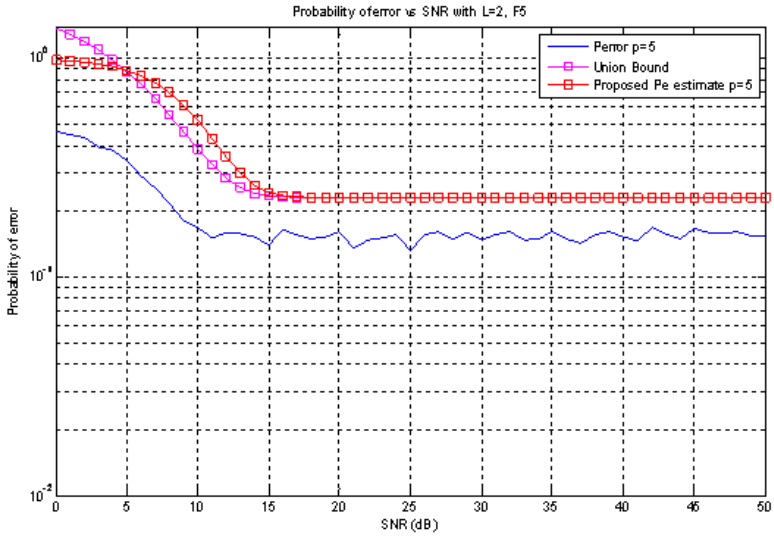


Figure 6: System Simulation for $p = 5$, $\pi = 2 + i$, $L = 2$

We can see in the plot the Union Bound (magenta line) and the proposed Pe estimate (red line). The first Union Bound (reference [1]) adjusts better to the simulated data for low SNR 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 L-dimensional antenna CF System Implementation

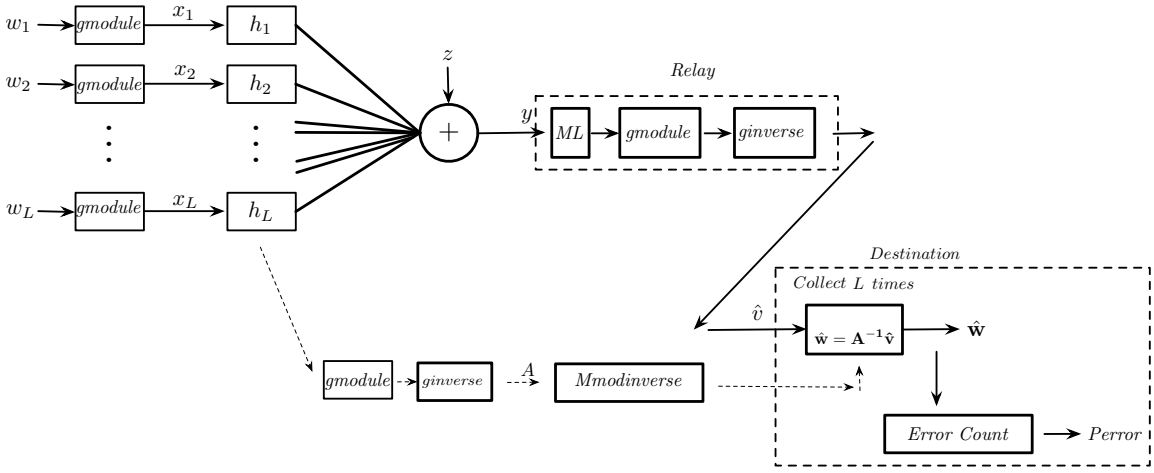


Figure 7: scalar model

This is a general L-dimensional implementation of the CF 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 $L = 2$, $p = 5$ ($\pi = 2 + 1i$)

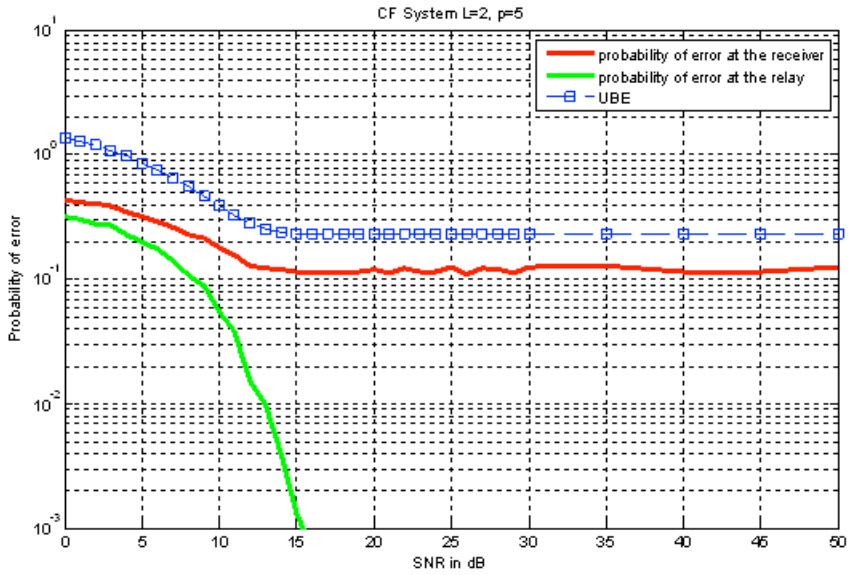


Figure 8: $p = 5$, $\pi = 2 + 1i$, $L = 2$

The red line corresponds to the probability of error at the receiver. The green line corresponds to the probability of error at the relay. The blue line corresponds to the Union Bound Estimate of probability of error.

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

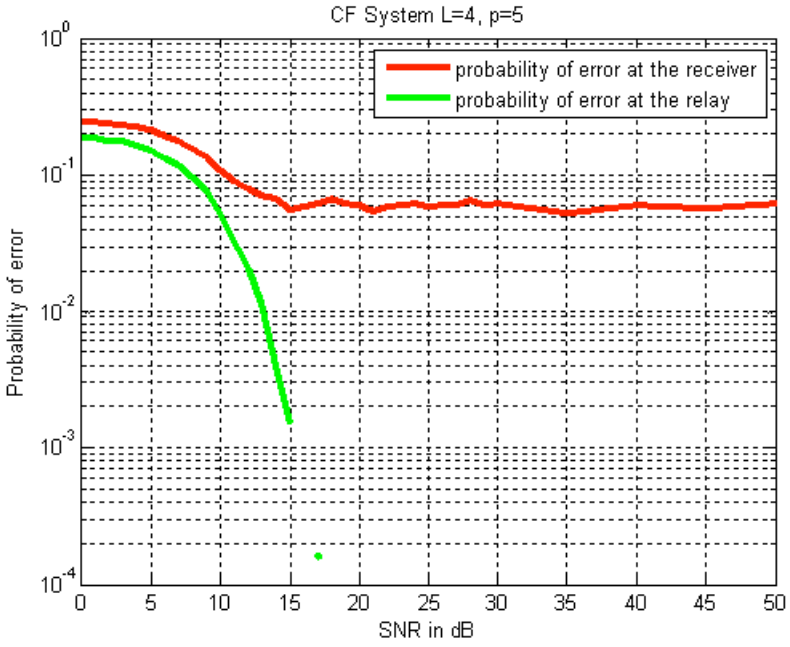


Figure 9: $p = 5$, $\pi = 2 + 1i$, $L = 4$

The red line corresponds to the probability of error at the receiver. The green line corresponds to the probability of error at the relay.

In the two plots above, it can be seen that for low SNR, 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 scalar CF system.

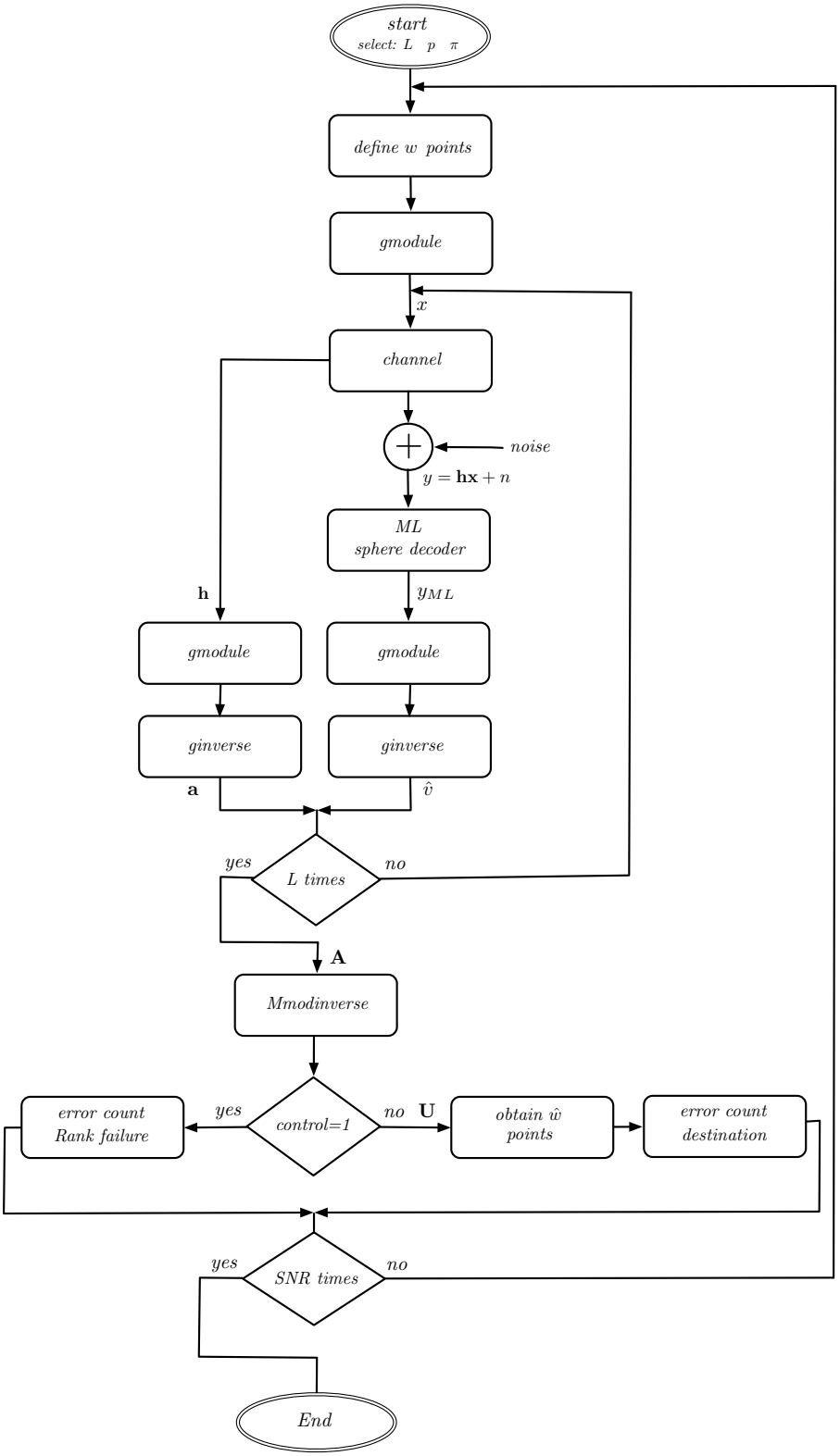


Figure 10: Flow chart diagram scalar CF system.

8 Extension of the Uncoded CF system model: vectorial case

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

8.1 Construction

We are going to extend the Compute and Forward Model studied to higher dimension, we use the model described in [6].

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

$$\mathbf{y} = \sum_{j=1}^{L} h_j \mathbf{x}_j + z$$

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

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

The relay decodes a noiseless linear combination of the transmitted messages,

$$\mathbf{v} = \sum_{j=1}^{L} a_j \mathbf{s}_j$$

and retransmits it to the destination.

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

$$a_j = \mu^{-1}(\psi(h_j))$$

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:

$$\mathbf{y} = \sum_{j=1}^{L} h_j \mathbf{x}_j + z$$

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

In order to decode the linear combination $\mathbf{v}$, the relay obtains a maximum likelihood (ML) estimate, $\phi : \mathbb{C}^n \rightarrow \mathbb{Z}[i]^n$, 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

$$\hat{\mathbf{v}} = \sum_{j=1}^{L} a_j \mathbf{s}_j$$

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

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 $\mathbf{s}_j$ 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 $\mathbf{s}_j$ 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

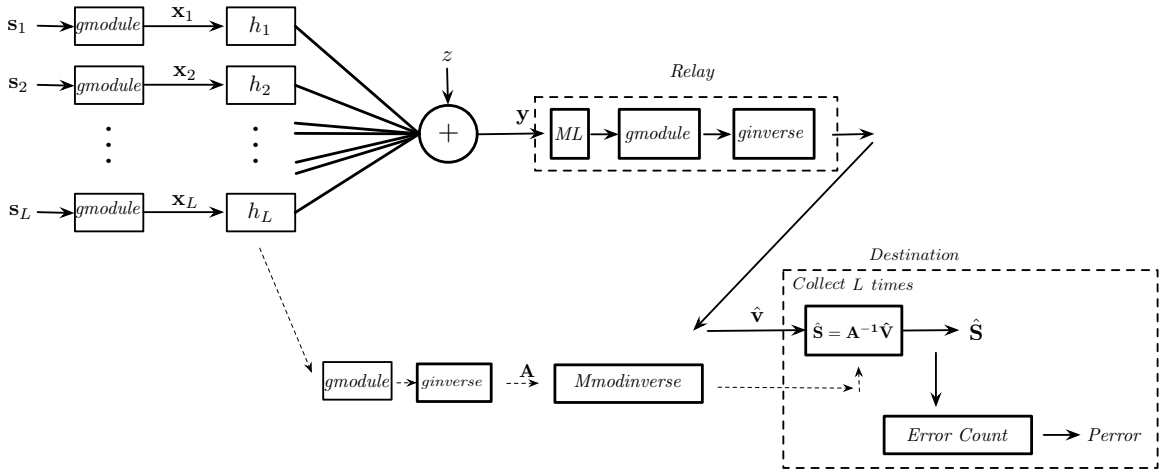


Figure 11: CF system model vectorial case

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

We simulate the system using a message vector with length $n = 4$, using $p = 5$ ($\pi = 2 + 1i$).

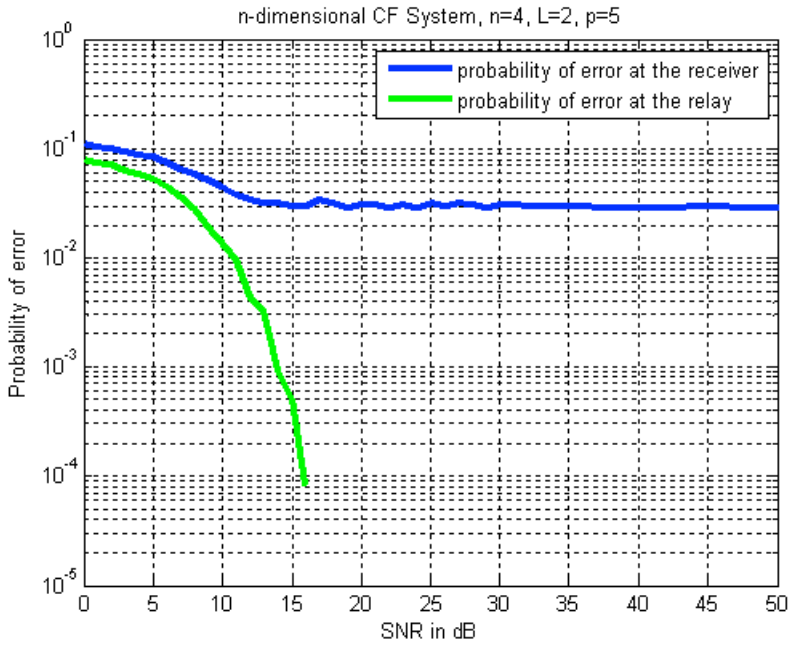


Figure 12: CF System $p = 5$, $\pi = 2 + 1i$, $L = 2$, $n = 4$

The blue line corresponds to 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 low SNR are due to errors at the relay and for high SNR are caused by rank failure.

We enclose the flow chart diagram of the n-dimensional CF system.

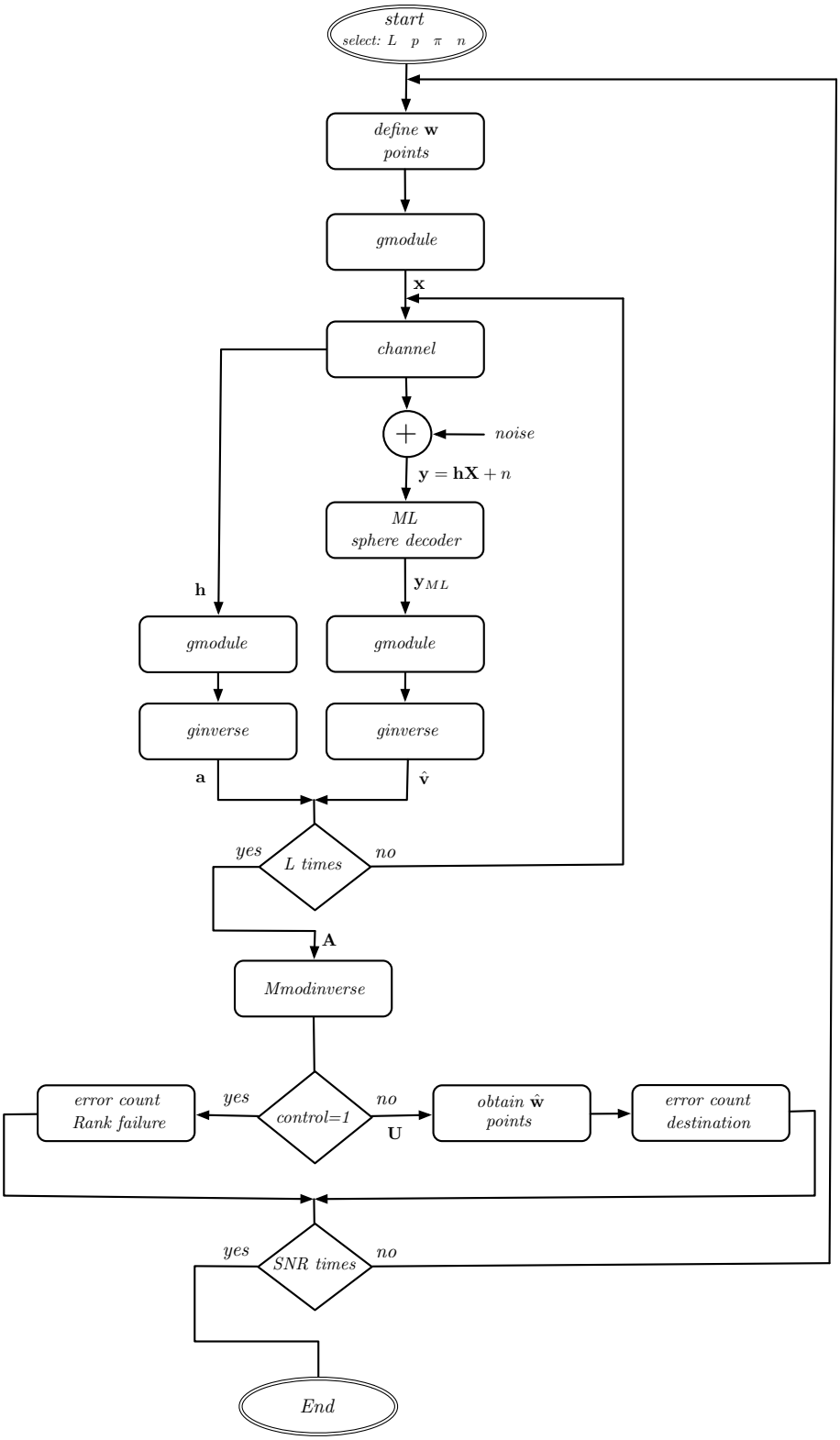


Figure 13: Flow chart diagram vectorial CF system.

9 q-ary Hamming coded CF system model

In this section we are going to focus on the implementation of a coded CF system. We are going to implement a q-ary Hamming (6,4) into de n-dimensional ($n = 4$) CF 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 q-ary Hamming 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 $v_1, v_2 \in C \quad a_1, a_2 \in F \rightarrow a_1 v_1 + a_2 v_2 \in C$

Definition 12. *A generating matrix of an $[n, k]$-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 an $[n, k]$-code C is an $m \times n$ 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 an $[n, k]$-code C, then a parity check matrix for C is*

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

9.1.2 q-ary Hamming 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 q-ary Hamming code of type $[n, n - r, 3]$ is the code C_H defined by the $m \times n$ parity check matrix*

$$H = (v_1 \mid v_2 \mid \ldots \mid v_n)$$

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 q*-ary Hamming code is generating the matrix* P *as an* $r \times k$ *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 generated a MATLAB code that generates matrix H and G given the input parameters.

- **Encoding**

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

  Example 9.2.

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

  where we can see that if $w_i = [1 \quad 2 \quad 1 \quad 2]$ *the coded vector is* $w_{coded} = [1 \quad 2 \quad 1 \quad 2 \quad 4 \quad 4]$*.*

- **Decoding**

  Furthermore, in order to decode a given vector $w_{codederror}$, where $w_{codederror} = (c \quad + \quad [0 \quad \dots \quad 0 \quad b \quad 0 \quad \dots \quad 0])$ where b is in the i-th component and c is in the codeword space.

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

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

The most 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. *Supose* $w_{codednoerror} = [1 \quad 2 \quad 1 \quad 2 \quad 4 \quad 4]$

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

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. *Supose* $w_{codederror} = [1 \quad 2 \quad 2 \quad 2 \quad 4 \quad 4]$

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

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

$$w_{coded} = [1 \quad 2 \quad 2 \quad 2 \quad 4 \quad 4] - [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0] = [1 \quad 2 \quad 1 \quad 2 \quad 4 \quad 4]$$

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

Example 9.5. *Suppose* $w_{codederror} = [1 \quad 0 \quad 1 \quad 2 \quad 4 \quad 4]$

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

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

$$w_{coded} = [1 \quad 0 \quad 1 \quad 2 \quad 4 \quad 4] - [0 \quad -2 \quad 0 \quad 0 \quad 0 \quad 0] = [1 \quad 2 \quad 1 \quad 2 \quad 4 \quad 4]$$

9.2 Performance

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

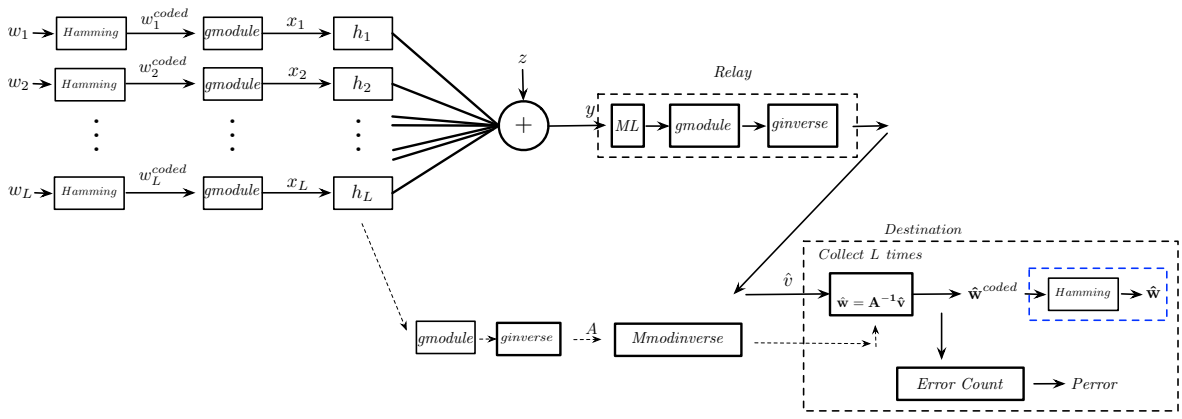


Figure 14: Coded system

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

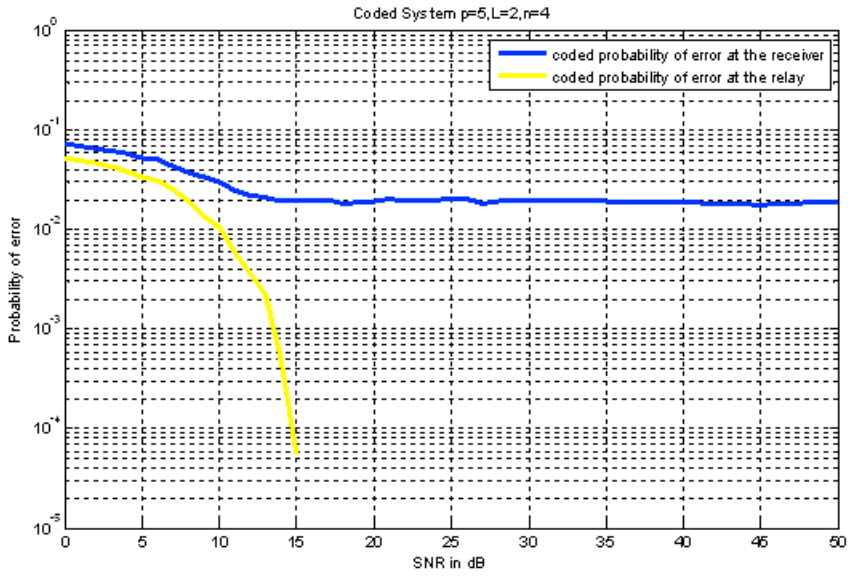


Figure 15: Coded p=5, $\pi = 2 + 1i$, $L = 2$, $n = 4$

The blue line corresponds to the probability of error at the receiver. The yellow line corresponds to the probability of error at the relay.

For low SNR relay errors continue to be really important whereas for high SNR 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.

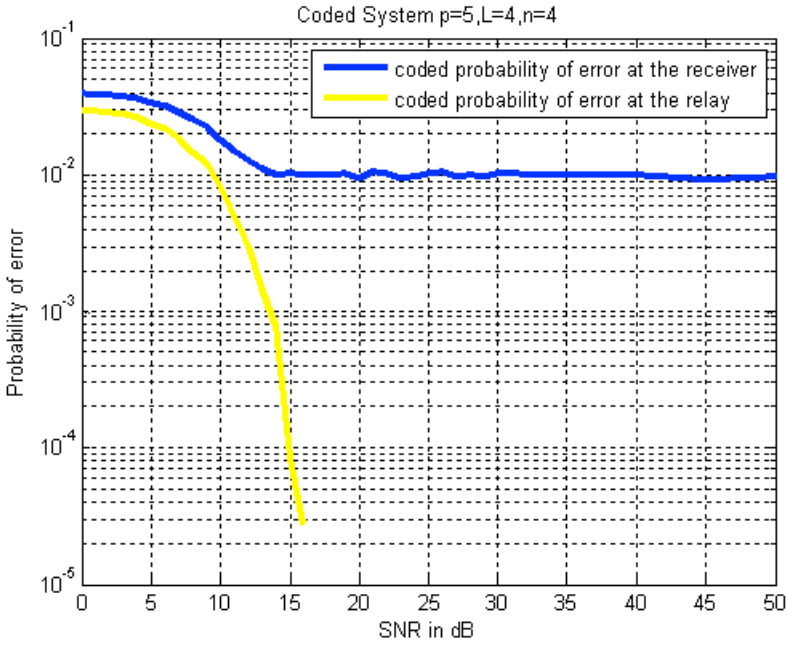


Figure 16: Coded p=5, $\pi = 2 + 1i$, L=4, n=4

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 $n = 4$.

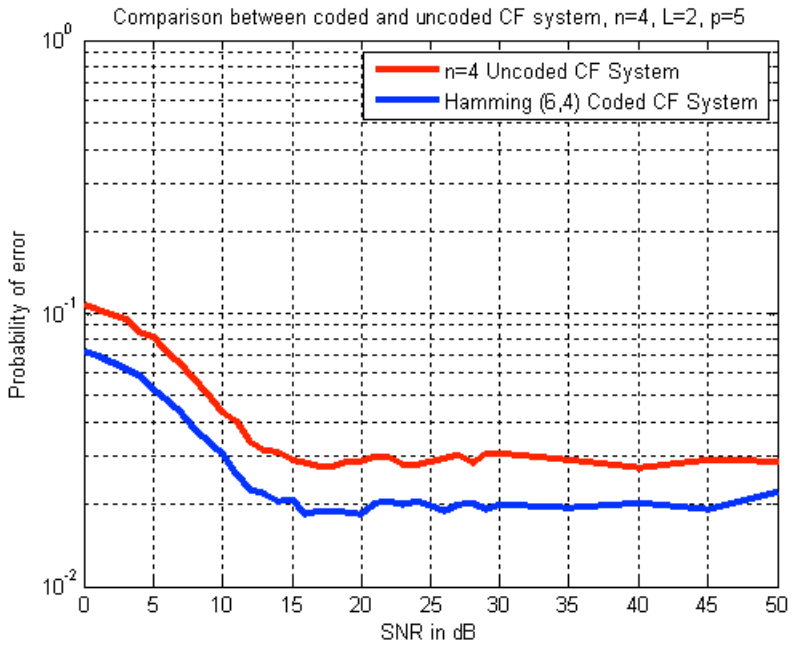


Figure 17: Uncoded vs Coded $p = 5$, $\pi = 2 + 1i$, $L = 2$, $n = 4$

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

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 $L = 2$**,** $L = 4$**,** $p = 5$

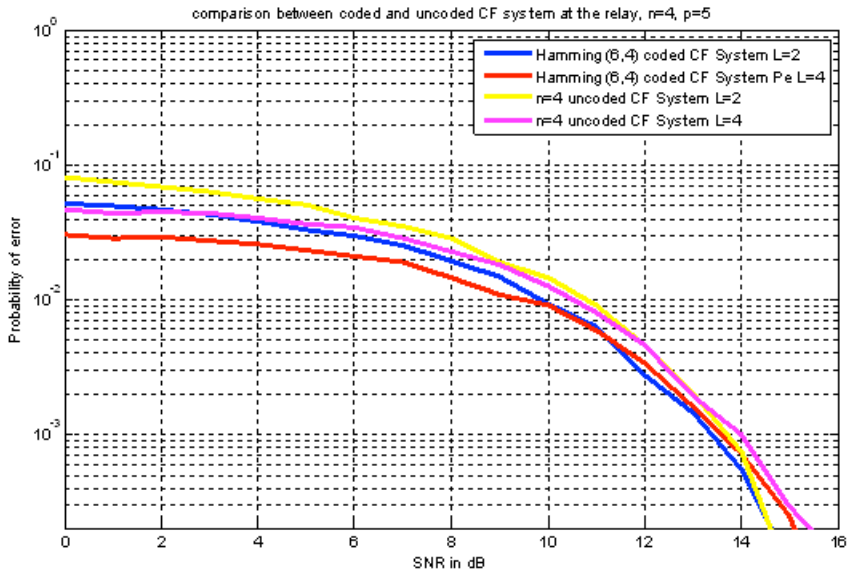


Figure 18: Uncoded vs Coded at the receiver $p = 5$, $\pi = 2 + 1i$, $L = 2$ and $L = 4$, $n = 4$

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 $L = 2$**,** $L = 4$**,** $p = 5$

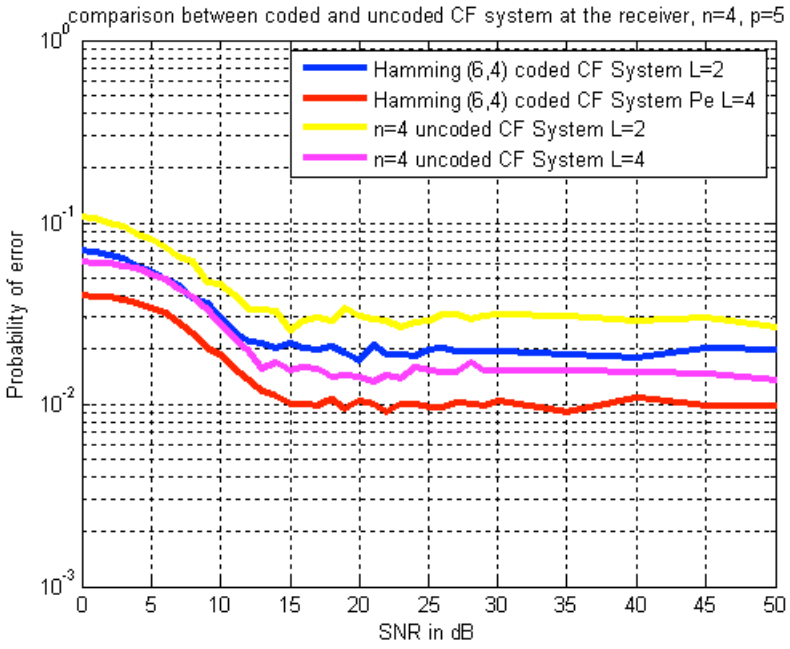


Figure 19: Uncoded vs Coded at the relay $p = 5$, $\pi = 2 + 1i$, $L = 2$ and $L = 4$, $n = 4$

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 Hamming Coded CF System.

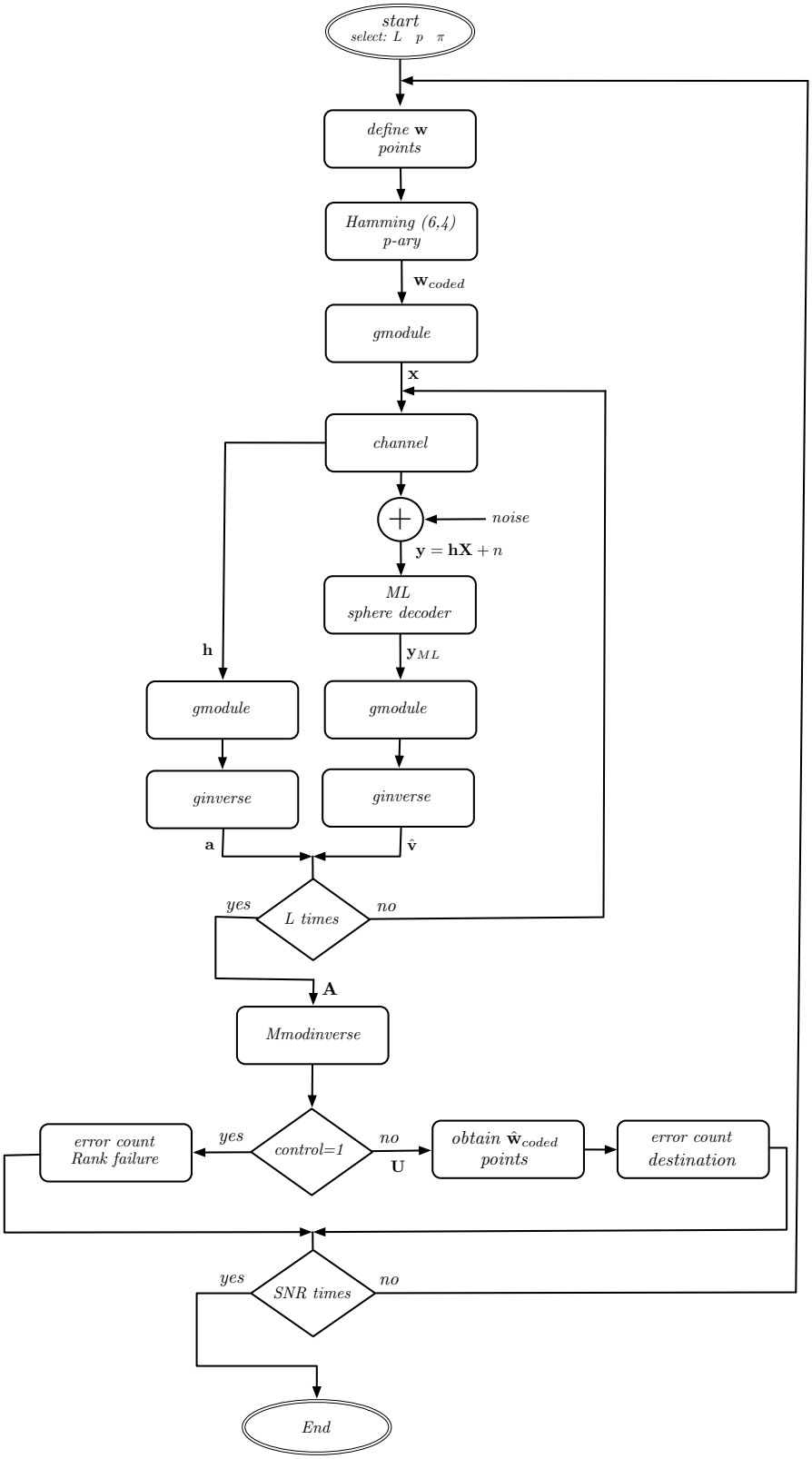


Figure 20: Flow chart diagram coded CF system.

10 Coefficient improvement: 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 CF system, one can see that there is a big number of errors due to rank failure. This is because $\det(A) = 0 \bmod p$. The first idea that comes to mind is trying to avoid this 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 CF system implemented.

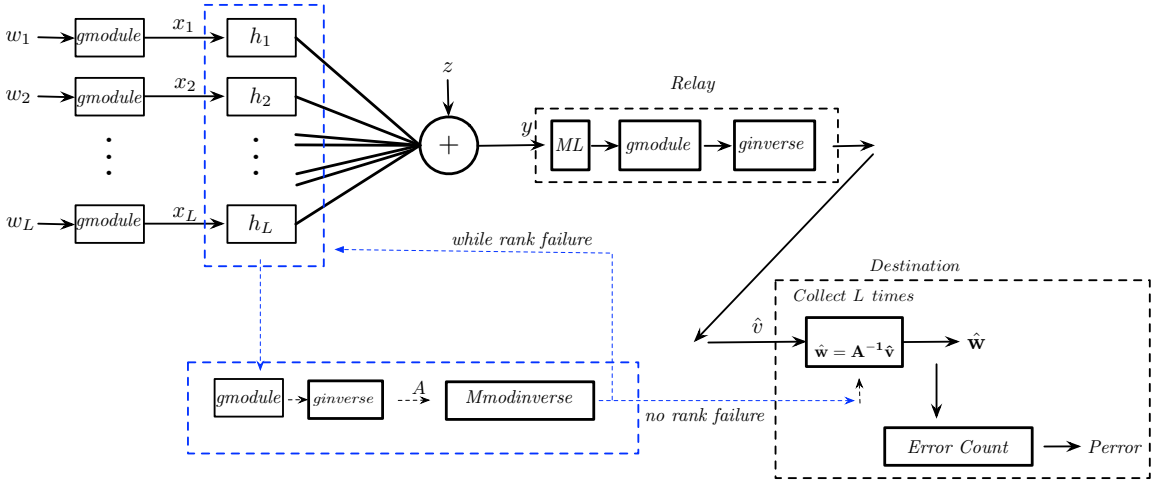


Figure 21: Improved Matrix A CF System

We can see the obtained results for the scalar case $L = 2$ and $p = 5$.

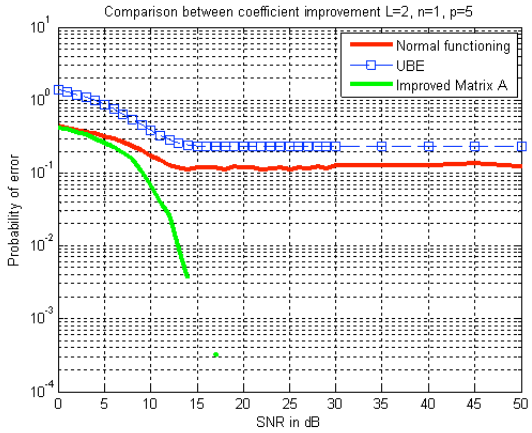


Figure 22: $L = 2$, $n = 1$, $p = 5$, Comparison between coefficient improvement

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

If we do the same for the vectorial case $L = 2$, $n = 2$, $p = 5$

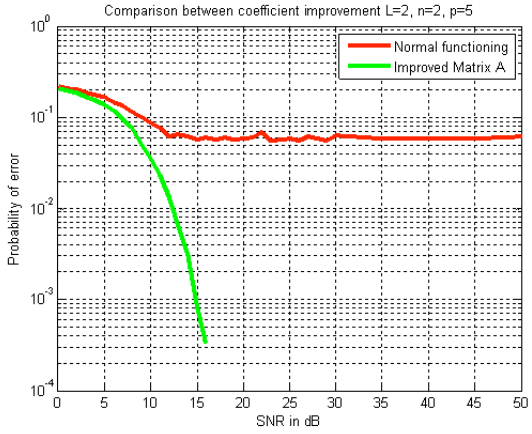


Figure 23: $L = 2$, $n = 2$, $p = 5$, Comparison between coefficient improvement

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 Improved matrix $\mathbf{A}$ CF system.

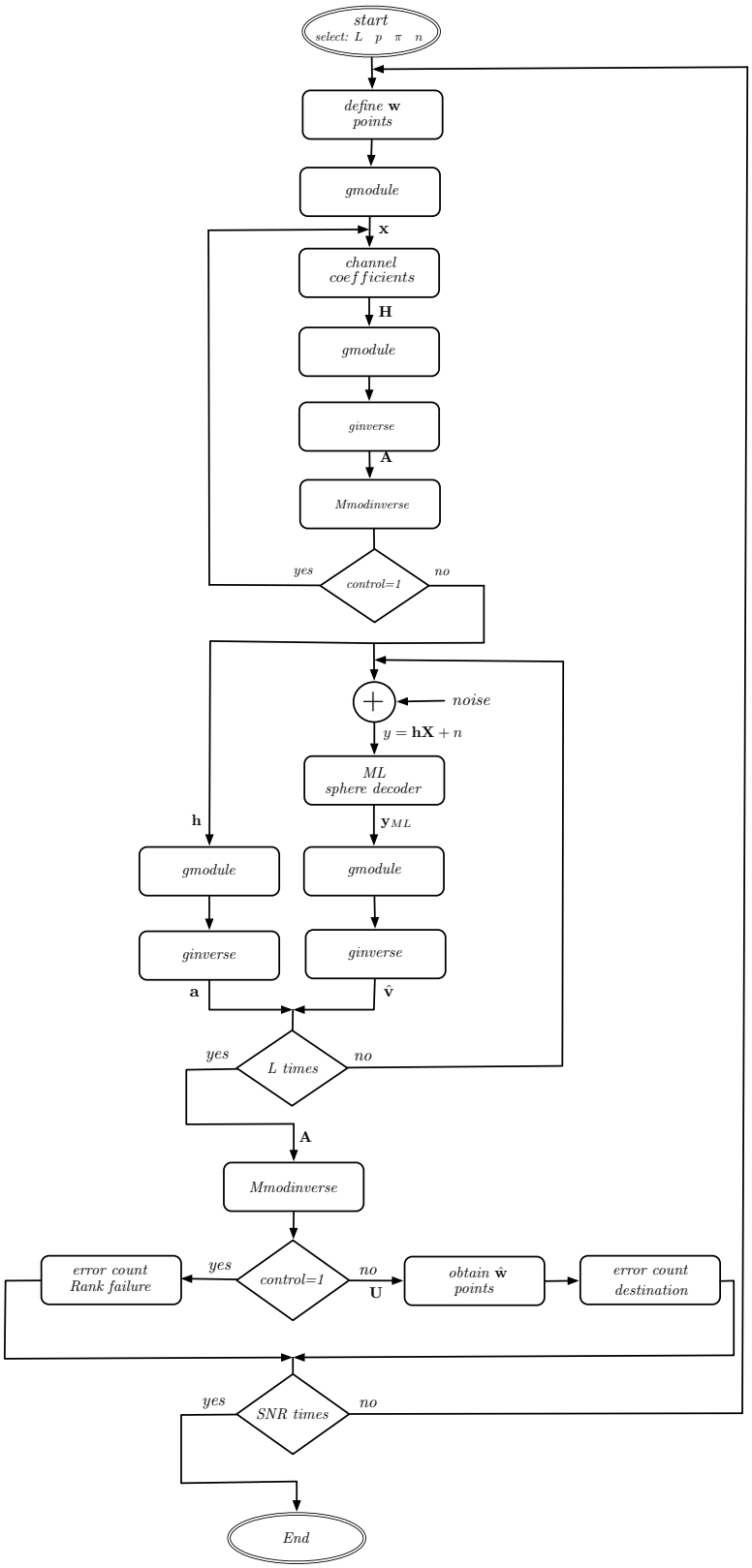


Figure 24: Flow chart diagram improved matrix $\mathbf{A}$ CF system.

11 Coefficient improvement: Optimum Matrix A

The idea behind this section is based on the optimum coefficient algorithms proposed in [5] for the CF System. Where the idea is to choose the optimal scaling factor β_m and optimal coefficient vector 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 [17] 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 Compute and Forward approach described earlier (proposed by Nazer and Gastpar in [5]) 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 an integer network coding coefficient vector

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

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

$$\beta_m \mathbf{y}_m = \sum_{l=1}^{L} \beta_m h_{ml}\mathbf{x}_l + \beta_m \mathbf{z}_m \tag{14}$$

$$= \sum_{l=1}^{L} a_{ml}\mathbf{x}_l + \sum_{l=1}^{L}(\beta_m h_{ml} - a_{ml})\mathbf{x}_l + \beta_m \mathbf{z_m} \tag{15}$$

Note that we do not need to conduct joint maximum likelihood (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 $A = \begin{bmatrix} a_1^1 & a_2^1 & \ldots & a_L^L \\ \vdots & & & \vdots \\ a_1^L & a_2^L & \ldots & a_L^L \end{bmatrix}$ 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 [5] 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 references [10] and [16].

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

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

Theorem 13. *The computation rate given in equation 16 is uniquely maximized by choosing β_m to be the MMSE coefficient*

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

which results in a computation rate of

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

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

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

where

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

and $\mathbf{H}_m = [H_{ij}^{(m)}], H_{ij}^{(m)} = h_{mi}h_{mj}, 1 \leq i, j \leq L$.

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

Then,

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

we want to solve the next SVP problem (which is an integer least squares 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 $\mathbf{Aopt}$ *is*

$$\mathbf{Aopt} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We can also find the optimum coefficients $\beta_{m1} = -0.2482$ *and* $\beta_{m2} = -0.2339$ *and do* $\beta_m \mathbf{y}$*. So, using* $\mathbf{A}_{opt}$ *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 Integer Least Squares problem.

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

this problem is analogous to solving

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

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

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 [18] of the LLL algorithm. The search algorithm is based on the Schnorr-Euchner enumeration strategy found in [19].

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 nonzero vectors $x \in C^m$ it is the case that $x^H Ax > 0$. If in addition $A \in \mathbb{R}^{m\times m}$ then A is said to be symmetric 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 symmetric positive matrix*

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

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 $\overline{B}$ is 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 $\min_{x\in\mathbb{Z}^n\setminus\{0\}} ||B\mathbf{x}||_2$ 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 integer least-squares (ILS) 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. in reference [15] 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, i.e. 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 reference [18] 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 Gramm Smidth orthogonalization, however when floating point operations are to be used, the Gram Smidth orthogonalization should not be used since it may have numerical stability problems. Therefore, the author in [18] 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.

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

$$\mathbf{b}_j^* = \mathbf{b}_j - \sum_{i=1}^{j-1} u_{ij}\mathbf{b}_i^*, \qquad 2 \leq j \leq n, \tag{23}$$

where $u_{i,j}$ defined by

$$u_{i,j} = \frac{\mathbf{b}_j^T \mathbf{b}_i^*}{||\mathbf{b}_i^*||_2^2} \qquad 1 \leq i < j \leq n, \tag{24}$$

are called the Gram-Schmidt coefficients and $(\mathbf{b}_i^*)^T(\mathbf{b}_j^*) = 0(i \neq j)$. It can be seen that $\mathbf{b}_j$ 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*

$$\begin{aligned} &|u_{ij}| \leq 1/2, \qquad 1 \leq i < j \leq n, \\ &\delta||\mathbf{b}_{i-1}^*||_2^2 \leq ||\mathbf{b}_i^* + u_{i-1,i}\mathbf{b}_{i-1}^*||_2^2, \qquad 1 < i \leq n. \end{aligned} \tag{25}$$

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

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 $\mathbf{b}_j^*$ are the orthogonal base vectors of B obtained by the Gram-Schmidt orthogonalization, and let U be a unit upper triangular matrix where its (i, j)-th element $(i < j)$ is defined by Gram-Schmidt coefficient $u_{i,j}$ 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:

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

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*

$$|r_{ij}/r_{ii}| \leq 1/2, \qquad 1 \leq i < j \leq n,$$
$$\delta r_{i-1,i-1}^2 \leq r_{ii}^2 + r_{i-1,i}^2, \qquad 1 < i \leq n, \tag{28}$$

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

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, i.e 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 $\{\mathbf{b}_i, \ldots, \mathbf{b}_n\}$ to achieve the LLL-reduced criteria, there are two types of basic operations in the algorithm of computing the LLL reduction.

- Substract one base vector times some integer from another, $\mathbf{b}_i := \mathbf{b}_i - t\mathbf{b}_j$, $t := \lfloor u_{ij} \rceil$.
- Interchange two nearby base vectors $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$.

The first operation is to ensure that new $|u_{ij}| \leq 1/2$ 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 $\mathbf{b}_i^* := \mathbf{b}_i - \sum_{j=1}^{i-1} u_{ij}\mathbf{b}_j^*$, $u_{ij} := \frac{\mathbf{b}_i^T \mathbf{b}_j^*}{||\mathbf{b}_j^*||^2}$. With these two operations, we can describe the algorithm that transforms a given basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ into an LLL reduced one.

LLL algorithm for the LLL-QRZ factorization

The algorithm is based on reference [18]. 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 reference [19].

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

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

or equivalently

$$\sum_{k=1}^{n}(y_k - \sum_{j=k}^{n} r_{kj}z_j)^2 < \beta \tag{32}$$

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

If we define

$$c_n = y_n/r_{nn}, \qquad c_k = (y_k - \sum_{j=k+1}^{n} r_{kj}z_j)/r_{kk}, \qquad k = n-1, \ldots, 1 \tag{33}$$

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

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

$$\vdots$$

$$\textit{level } k: \quad r_{k,k}^2(z_k - c_k)^2 < \beta - \sum_{i=k+1}^{n} r_{ii}^2(z_i - c_i)^2, \tag{36}$$

$$\vdots$$

$$\textit{level } 1: \quad r_{1,1}^2(z_1 - c_1)^2 < \beta - \sum_{i=2}^{n} r_{ii}^2(z_i - c_i)^2. \tag{37}$$

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 (formula 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 $\hat{\mathbf{z}}$. 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 an 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 $f(\mathbf{z}^{(j-1)}) \leq f(\mathbf{z}^{(new)}) \leq f(\mathbf{z}^{(j)})$. We remove the point $\mathbf{z}^{(p)}$ and rename $\mathbf{z}^{(new)}, \ldots, \mathbf{z}^{(p-1)}$ as $\mathbf{z}^{(j)}, \ldots, \mathbf{z}^{(p)}$, 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 Schnor Euchner Enumeration algorithm found in the MILES Matlab package.

11.3 Performance

Here we can see the diagram of the CF System implemented.

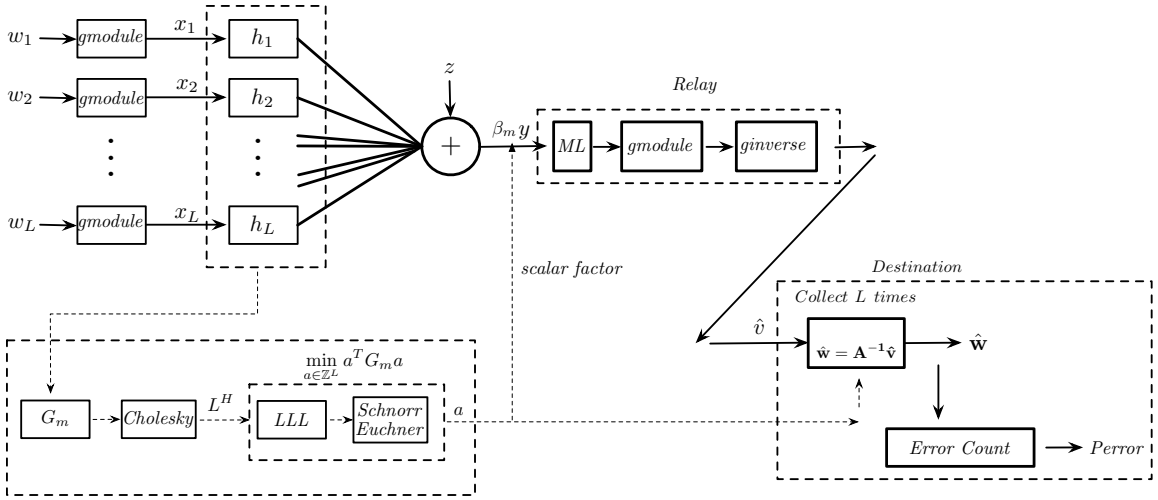


Figure 25: Optimum Matrix A CF System

If we do the simulation for the scalar case, using $L = 2$ and $p = 5$, we can see

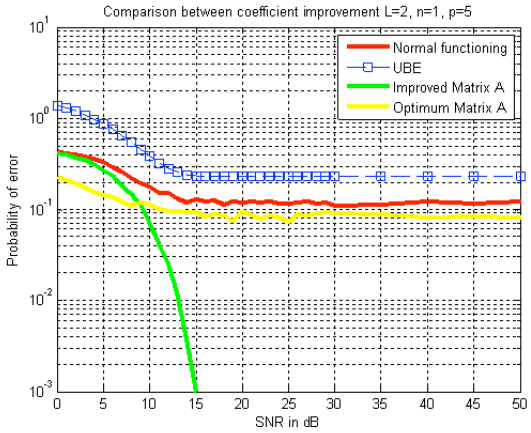


Figure 26: $L = 2$, $n = 1$, $p = 5$, Comparison between coefficient improvement

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 full rank continue to affect the system.

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

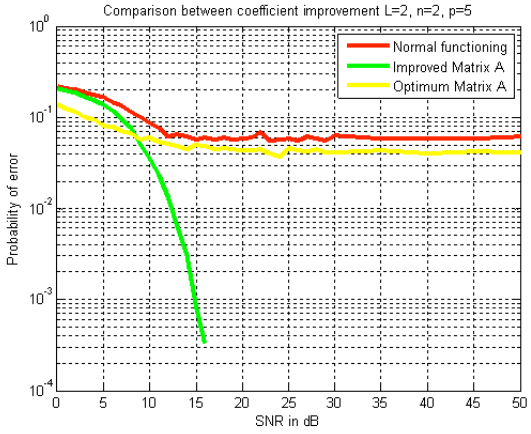


Figure 27: $L = 2$, $n = 2$, $p = 5$, Comparison between coefficient improvement

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

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

We enclose the flow chart diagram of the Optimum Matrix $\mathbf{A}$ CF System.

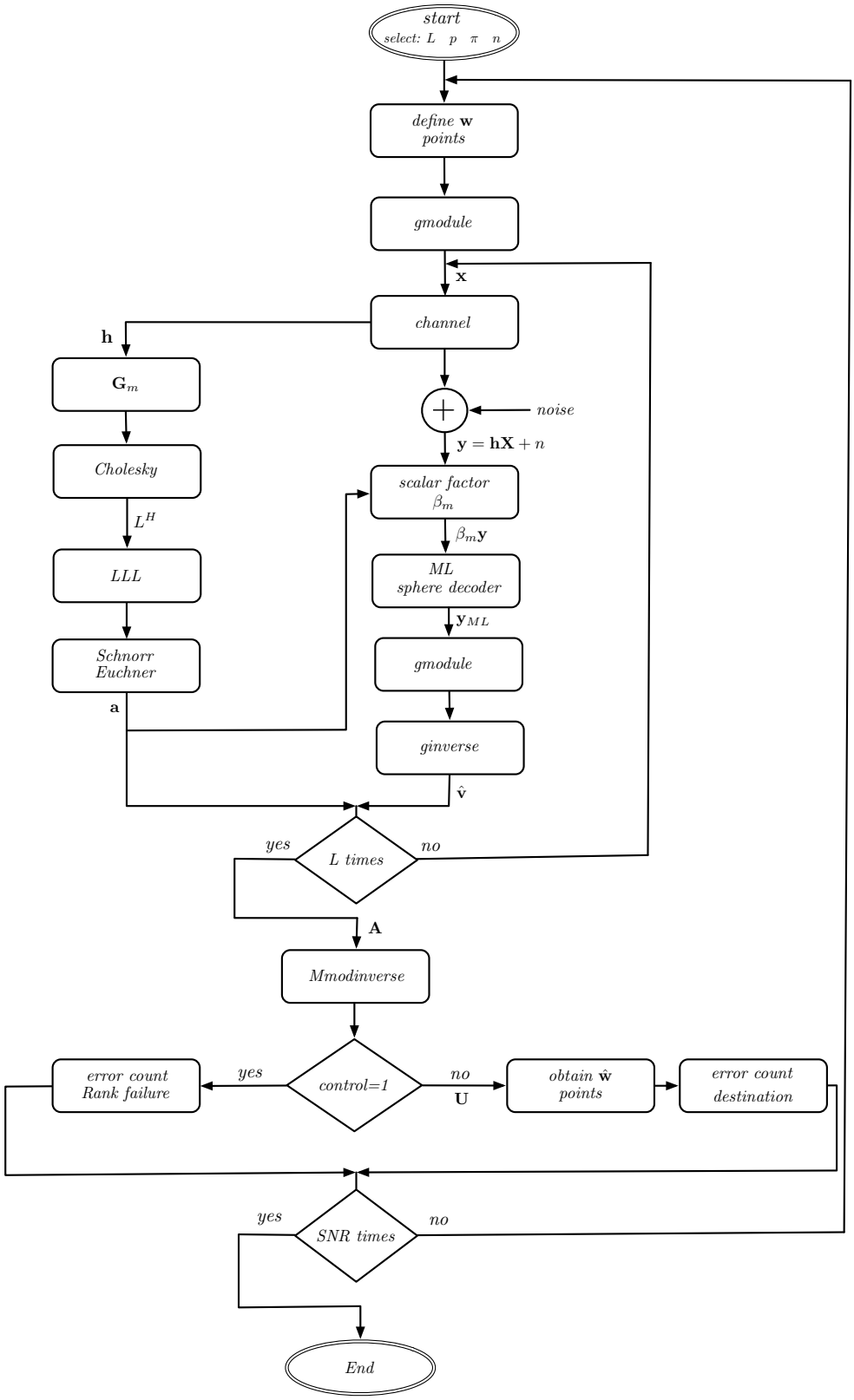


Figure 28: Flow chart diagram improved matrix **A** CF system.

12 Coefficient improvement: Improved Optimum Matrix A

The aim of this section is to improve the CF 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 an strategy to obtain full rank matrices. 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 full rank matrices.

12.2 Performance

In the next diagram, it can be seen the CF system implemented

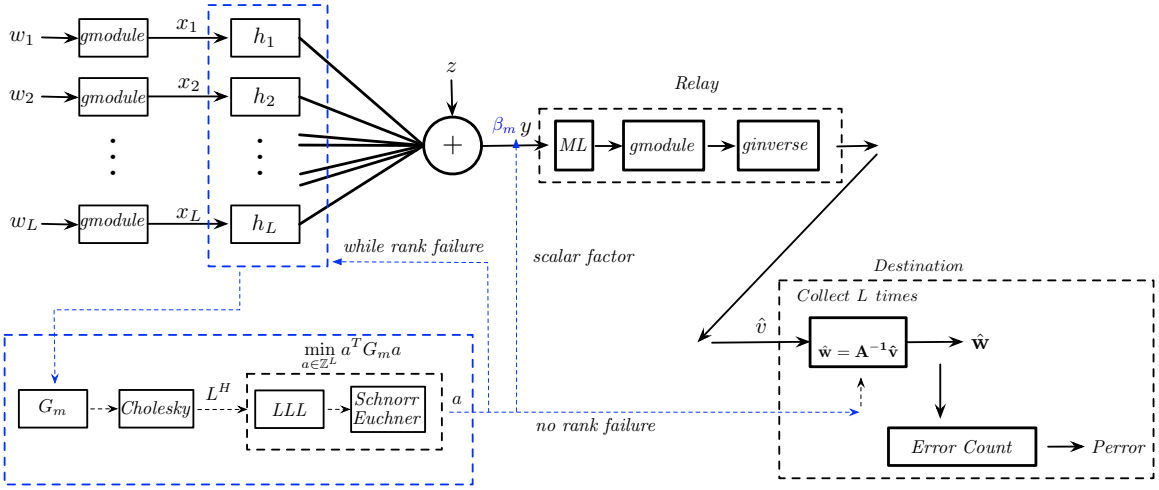


Figure 29: Improved Optimum Matrix A CF System

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

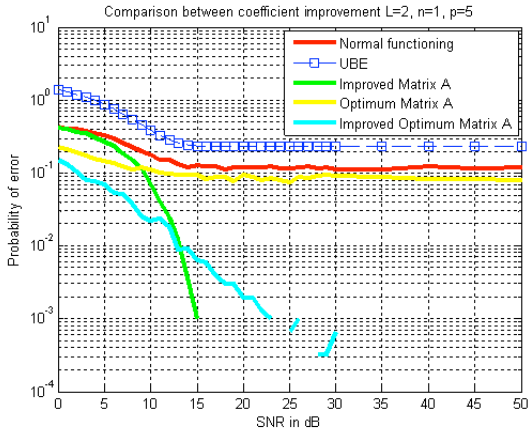


Figure 30: $L = 2$, $n = 1$, $p = 5$, Comparison between coefficient improvement

we can see that for low SNR (below 12dB) the improved optimum matrix A (cyan line) is the best method found. However, if SNR goes up, the first approach used, the Improved Matrix A (green line), attains better results. In the plot we can observe how both Improved Optimum Matrix A and Improved Matrix 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 $p = 5$.

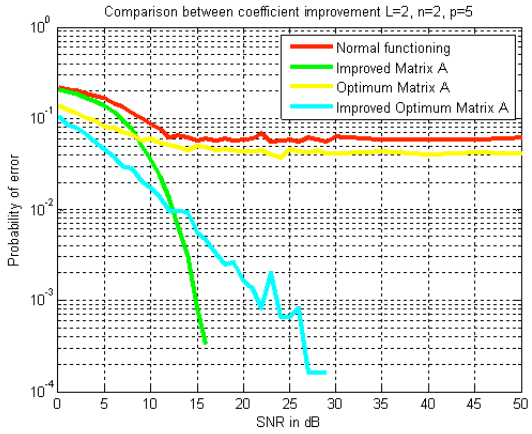


Figure 31: $L = 2$, $n = 2$, $p = 5$, Comparison between coefficient improvement

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

We enclose the flow chart diagram for the Improved Optimum Matrix $\mathbf{A}$ CF system implemented

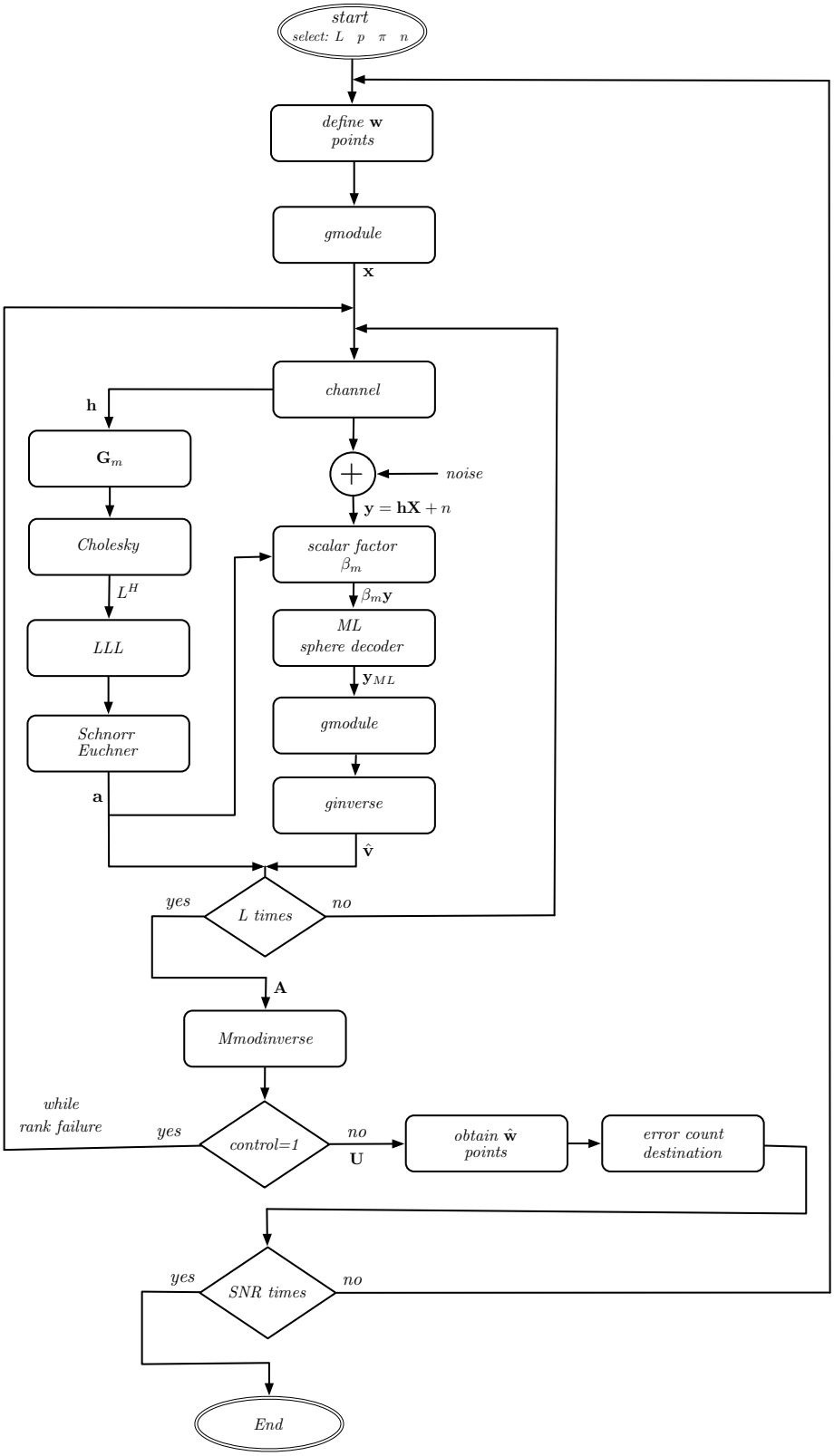


Figure 32: Flow chart diagram Improved Optimum matrix **A** CF system.

13 Conclusions and further work

In this project we have done a survey of the mathematical theory needed to understand the CF system. Both an introduction to Physical Layer Network coding and Lattice Network Codes has been presented.

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

The next step has been involved in the improvement of the coefficient matrix A. We proposed and implemented an easy yet intelligent idea to avoid full rank. Next, the optimum algorithm found in the literature has been studied and implemented. Concepts related to solving an ILS problem (Integer Least Squares) using lattice reduction and vector search has 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 improved Matrix A algorithm, attains the best performance for high SNR. However, is slow in terms of time computing resources. The Improved Optimum Matrix A algorithm, has good performance and is the best for low SNR. Moreover, it has better time computing performance. Both algorithms reach zero errors with increasing SNR, which can be easily understood because they are designed in order to avoid full rank, which is the main cause of error for high SNR.

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 $\mathbb{Z}[i]$.

Bibliography

[1] S. Gupta, M. A. Vázquez-Castro *Compute and Forward: End to End Performance over Residue Class Based Signal Constellations*, 2012

[2] K. Huber. *Codes over Gaussian integers*, IEEE Trans. on Information Theory, vol. 40, no. 1, pp 207-216, Jan. 1994.

[3] Q.T. Sun, J. Yuan, T. Huang, K.W. Shum *Lattice Network Codes Based on Eisenstein Integers*, IEEE Trans. on communications, 2013.

[4] R. Ahlswede, N.Cai, S.-Y.R.Li and R.W. Yeung. *Network Information flow*, IEEE. Trans Inf. Theory, vol.46, no.4, pp.1024-1216, Jul.2000.

[5] B. Nazer & M. Gastpar. *Compute and Forward: Harnessing Interference through structured codes*, IEEE Trans, on Information Theory, vol. 57, no.10, pp 6463-6484, Oct 2011.

[6] Jean-Claude Belfiore & Cong Ling. *The Flatness Factor in Lattice Network Coding: Design Criterion and Decoding Algorithm*, IEEE Trans, on Information Theory, vol. 57, no.10, pp 6463-6484, Oct 2011.

[7] S. Zhang, S.C. Liew, P.P. Lam, Hot topic: physical-layer network coding, ACM MobiCom '06, Sept. 2006, pp. 358–365.

[8] C. Feng, D. Silva and F.R. Kschischang. *An algebraic approach to physical-layer network coding*, 2011

[9] William. C. Waterhouse. *How often do determinants over finite fields vanish?*, Discrete Mathematics, Volume 65, Issue 1, pp 103-104, May 1987.

[10] B. Nazer, M. Gastpar. *Reliable Physical Layer Network Coding*, Proceedings of the IEEE, Vol.99, no 3, March 2011.

[11] L. Wei and W. Chen. *Compute-and-forward network coding design over multi-source multi-relay channels*, IEEE Trans. Wireless Commun., vol. 11 no.9, pp.3348-3357, September 2012

[12] L. Wei and W. Chen *Efficient compute-and-forward network codes search for two-way relay channels*, IEEE Commun. Lett., vol.16, no8, pp.1204-1207, August 2012

[13] H. Minkowski. Geometrie der zahlen. Teubner, 1896.

[14] H. Minkowski. Diophantische approximationen. Teubner, 1907.

[15] A.K. Lenstra, H.W. Lenstra, and L. Lovász. *Factoring polynomials with rational coefficients*. Mathematische Annalen, 261:515-534, 1982.

[16] A. Osmane and J.C Belfiore. *The compute-and-forward protocol: implementation and practical aspects* IEEE Communication Letters

[17] Lili Wei. *Network Coding Design in Wireless Cooperative Networks*, Chinese PostDoc Shanghai Jiao Tong University, January 21, 2012.

[18] T. Zhou *Modified LLL algorithms*, Master of Science Thesis, McGill University, 2006

[19] X. W Chang, T. Zhou. *MILES: MATLAB package for solving Integer Least Squares problems. Theory and Algorithms*, McGill University, January 2011.

[20] K. Su, *Efficient ML detection for communication over MIMO channels*, Technical Report, Feb. 2005, University of Cambridge.

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'anàlisis 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 CF System. Després, s'analitza i implementa el primer esquema bàsic. A partir del qual, implementem una versió vectorial del CF 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 CF system. Lo siguiente es analizar y implementar el primer esquema básico . A partir del cual, implementamos una versión vectorial del CF 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 Physical Layer Network Coding schemes. Using a basic scheme as starting point, the project presents the construction and performance of different communication systems 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 CF System are presented. Further, the first basic scheme is analysed and implemented. The next step consists on implementing a vectorial CF System and a coded q-ary Hamming version of the System. Finally, different approaches to improve the matrix coefficient A are studied and implemented.

Joaquim Curto Díaz
July 2013, Bellaterra