Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in
Ingegneria Informatica

Source coding with mixed side information

Relatore:
Prof. Stefano Tomasin

Laureando:
Simone Merlo

Anno Accademico 2021/2022

Contents

Abstract

0.1 English abstract

This thesis pertains the feedback of downlink channel state information from the user equipment to the base station in a cellular system. Taking into account that the base station has already estimated the uplink channel, and that the downlink and uplink channels are correlated, the base station first sends to the user equipment an encoded version of this estimate. The user equipment will use this information to encode the downlink channel state information. The user equipment will then send this encoded message to the base station, which will obtain an estimate of the downlink channel, using also its knowledge of the uplink channel. This protocol is framed as a lossy source coding problem with mixed type of side information and various coding/decoding strategies are explored.

0.2 Italian abstract

Questa tesi tratta il feedback delle informazioni riguardanti lo stato del canale di downlink, inviato dalla base station al dispositivo dell'utente in un sistema cellulare. Considerando che la base station ha precedentemente stimato il canale di uplink e che i due canali (uplink e downlink) sono correlati, la base station inoltra, come prima cosa, una versione codificata del canale di uplink al terminale dell'utente. Il dispositivo userà queste informazioni per codificare quelle riguardanti lo stato del canale di downlink e successivamente invierà quest'ultime alla base station. La base station, quindi, otterrà una stima del canale di downlink utilizzando le informazioni ricevute e la conoscenza pregressa del canale di uplink. Questo protocollo rappresenta un problema di codifica di sorgente con perdita con informazioni laterali miste. Sono state considerate diverse strategie di codifica/decodifica.

Introduction

In a cellular communication two different entities are involved: a base station and a user equipment. They use two different channels to communicate: an uplink and a downlink channel. By using some basic techniques (such as pilot messages) the user equipment is able to gain some knowledge about the downlink channel and the base station about the uplink channel. In order to adapt the messages at the transmitter, before sending them on the channels, to avoid the need of corrections at the receiver, the base station requires to know the behaviour of the downlink channel and the user equipment requires to know the behaviour of the uplink channel. Because of that, the user equipment must send the information about the downlink channel to the base station. The two channels, however, are related, so the base station sends an encoded version of the uplink channel to the user equipment and the user equipment encodes the information about the downlink channel by exploiting the information received by the base station. In particular, because of the correlation between these two channels, the base station focuses on telling something useful to the user equipment (to reduce the amount of data that the user equipment needs to forward) and the user equipment tries to understand the similarities between the two channels in order to send to the base station some new information and not something that is already known. A completely symmetrical process could be followed to send the information about the uplink channel to the user equipment.

In Chapter 1 we describe the overall system and some theoretical models that represent an abstraction of the system itself.

In Chapter 2 we propose some encoding and a decoding procedures based on the theoretical model, we recount some new methods of encoding the uplink channel at the base station and we describe some algorithms used by the procedures.

In Chapter 3 we describe how we implemented the proposed procedures in many different ways and how we evaluated the performances of these procedures and we compare the performances results.

Chapter 1

System Model

In this chapter we describe the basic behaviour of a cellular network communication system and the possible changes that can be made in order to improve this system.

1.1 FDD systems and communication channels

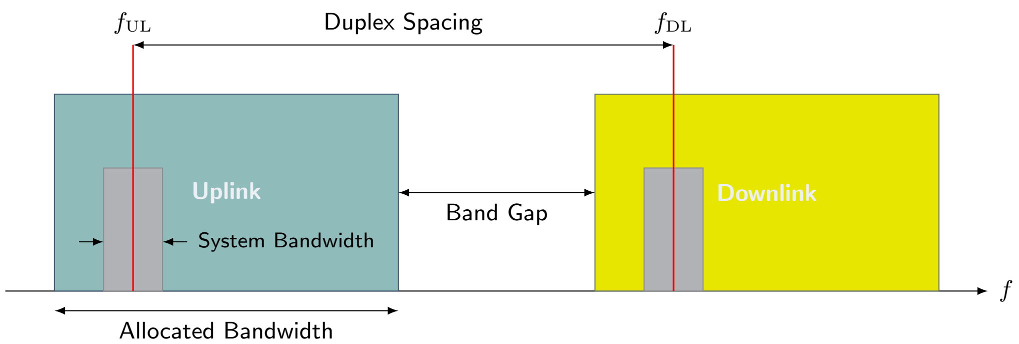


Figure 1.1: FDD system frequencies scheme.

A cellular communication system is composed by several cells, each covering a specific territory. In each cell a base station (BS), connected to a fixed network, communicates with several user equipements (UEs).
A Frequency Division Duplexing [1] system uses two different frequency bands for uplink (UL, from the UEs to the BS) and downlink (DL, from the BS to the UEs). The central frequency of the uplink band is denoted by f_{UL} and the one of the downlink band by f_{DL}, as shown in Fig. 1.1.

1.2 System model

We consider a two-way communication system where a UE and a BS represent the end-points of the communication. This system uses two separate channels (as described in Section 1.1): the UL and the DL channels (as shown in Fig. 1.2).

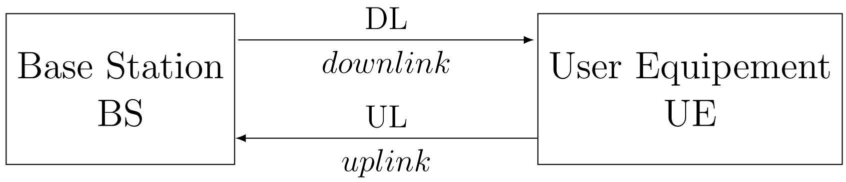


Figure 1.2: Basic system representation

The information sent on these channels is affected by the distortion introduced because of physical factors, interference and other phenomena.
If the distortion suffered by the message is known in advance, it is possible to adapt the transmitted signal to receive, at the destination endpoint, almost the original message (it is impossible to obtain the exact same message as the original because the channel changes almost constantly).

To understand the channel behaviour the BS and the UE send to each other a pre-determined signal (a pilot message) so that they can understand the changes applied to it during the communication.
The major issue, though, is the fact that the UE sends this pilot message to the BS by using the UL channel and the BS does the same by using the DL channel so, at the end of this procedure, the BS knows the UL channel and the UE knows the DL channel. However it would be useful for the BS to know the DL channel because is the DL channel to be used for the communication sent from the BS to the UE, so the BS has to adapt the signal by knowing the modification applied by the DL channel.
Thus, after this first procedure, the UE sends the knowledge about the DL channel to the BS; this exchange of information can be done in a better way by taking advantage of the knowledge already acquired by the two endpoints.
Indeed, the UL and the DL channels are usually related because they make use of the same physical channel so the physical factors (such as absorption and reflection) are nearly the same. We use this similarity to reduce the amount of information that we send from/to the BS and the UE to describe these channels.

Let $\mathbf{X}$ indicate the DL channel estimate available at the UE and $\mathbf{Z}$ indicate the UL channel estimate available at the BS. We propose a three-stage approach where:

1. The BS encodes $\mathbf{Z}$ into $\mathbf{Y}$, using the Encoder 2, and sends it to the UE.

2. The UE encodes **X** into **W** (using also **Y**), using the Encoder 1, and sends it to the BS.

3. Lastly, the BS decodes **W**, using also **Z** (and **Y**) to obtain the estimate $\hat{\mathbf{X}}$ of **X**.

The resulting schemas for the three stages are shown in Fig. 1.3.

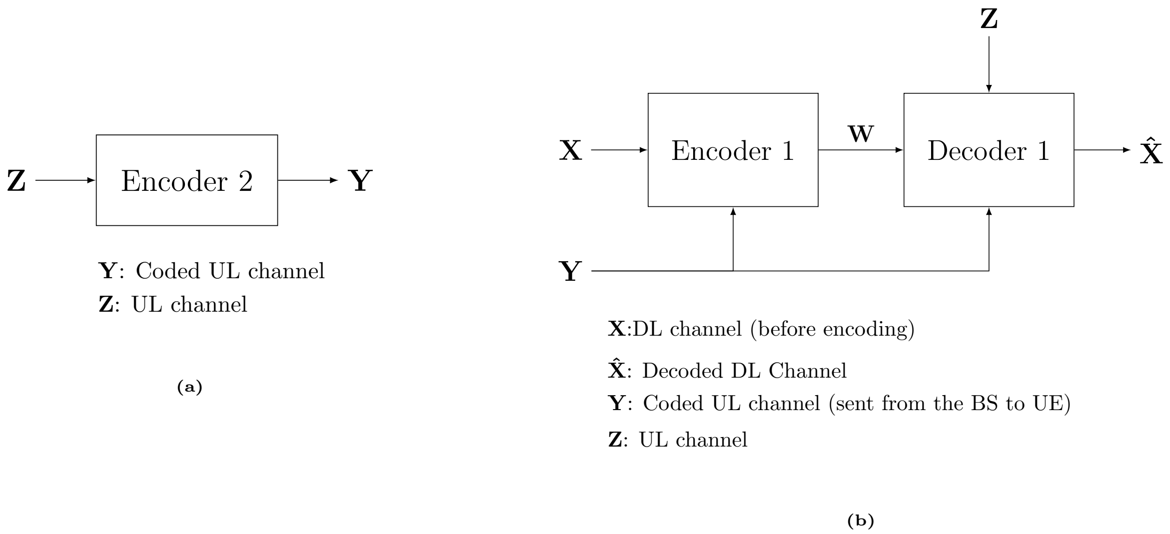


Figure 1.3: (a)Encoding of the UL channel. (b)Example of the process used to reduce the information sent.

1.3 Encoder 1

We now focus on the Encoder 1. In a Mixed Side Information (MSI) system the message (**X**) sent form the source is codified by the encoder by using the knowledge coming from a side information (**Y**); this side information is also known at the decoder which uses it in addition to an other side information (**Z**) to decode the message received (as shown in Figure 1.3(b)).

Letters **X**,**Y** and **Z** represent random variables, they could be thought as vectors of random numbers (e.g. to represent the sequence of bit transmitted on a communication channel).

We define as b_B the number of bits sent on the UL channel (used to represent **W**).

This system could be converted to the equivalent Wyner-Ziv system by defining $\mathbf{X}'$=($\mathbf{X}, \mathbf{Y}$) as the information known at the encoder and $\mathbf{Z}'$=($\mathbf{Y}, \mathbf{Z}$) as the information known at the decoder. The scheme of a Wyner-Ziv system is reported in Fig. 1.4.

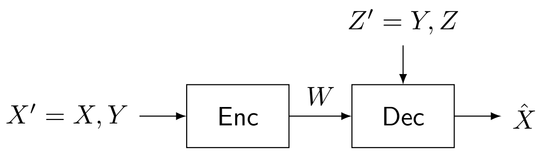


Figure 1.4: Wyner-Ziv system

In this case $\mathbf{X}'$ and $\mathbf{Z}'$ are correlated random variables.

1.4 Encoder 2

The Encoder 2 (shown in Fig. 1.3(a)) needs to encode $\mathbf{Z}$ into $\mathbf{Y}$. The encoding is done by using a lossy coding procedure, so that is possible to reduce the number of bits needed to communicate $\mathbf{Z}$ to an endpoint by sending $\mathbf{Y}$.
We define as b_F the number of bits sent on the DL channel (used to represent $\mathbf{Y}$).

Chapter 2

Encoding and Decoding

In this chapter we describe a proposed source coding procedure, we report some methods that are based on the proposed procedure and we shortly introduce some algorithms used by it.

Six different methods are described: we define as "NSI method" (No Side Information method) the first method, as "OZ method" (Only Z method) the second method, as "LZ method" (LBG on Z method) the third method, as "LTX method" (LBG on Tilde X method) the fourth method, as "LZHZ method" (LBG on Z and Hat Z method) the fifth method and as "LTXHZ method" (LBG on Tilde X and Hat Z method) the sixth method. The NSI method represents a reference so no side information is used, the OZ method focuses on the decoder and his side information $\mathbf{Z}$ as if the encoder doesn't exist, the LZ and the LTX methods represent the proposed version (Sections 2.2 and 2.3) but they differ for the way of encoding $\mathbf{Y}$ from $\mathbf{Z}$ (Section 2.5), the LZHZ and the LTXHZ methods represent also the proposed version but they differ from each other for way of encoding $\mathbf{Y}$ from $\mathbf{Z}$ (Section 2.5) and from the LZ and LTX methods for the way of creating the codebooks.

2.1 Linde, Buzo and Gray algorithm

The Linde, Buzo and Gray (LBG) algorithm [2] is a lossy source coding procedure and it uses a training sequence (TS) to design the vector quantization.
The LBG algorithm converges to a minimum but it is not guaranteed that it is a global minimum, as it depends on the choice of the TS.

Let $\mathbf{Q}_i$ be the *i-th* code vector, K be the number of vectors of the TS and $\mathbf{s}(k)$ be the *k-th* element of the TS. The code vector $\mathbf{Q}$ nearest to a given $\mathbf{s}(k)$ is defined as:

$$\mathbf{Q} = \arg\min_{\mathbf{Q}_i} \|\mathbf{s}(k) - \mathbf{Q}_i\|. \tag{2.1}$$

We use an initial codebook length $L = 1$ and the centroid of the TS as initial code vector [1].

The algorithm procedure is the following: we create two new code vectors from each centroid by slightly changing their components (for the first iteration we have only one centroid so we obtain two vectors and we can consider $L = 2$). After this we apply the following steps:

- The TS is divided into regions (one for each code vector) by associating each vector of the TS to the region corresponding to the nearest code vector.
- The centroids of the regions are computed and used as new code vectors (by replacing those used before).
- This steps are repeated until the optimum code vectors are obtained.

We now repeat the entire procedure by splitting the new centroids until we reach the desired number of code vectors (determined by the distortion bound to respect). A scheme of the procedure is shown in Fig. 2.1.

After this, the resulting codebook represents the optimal codebook for the given TS.

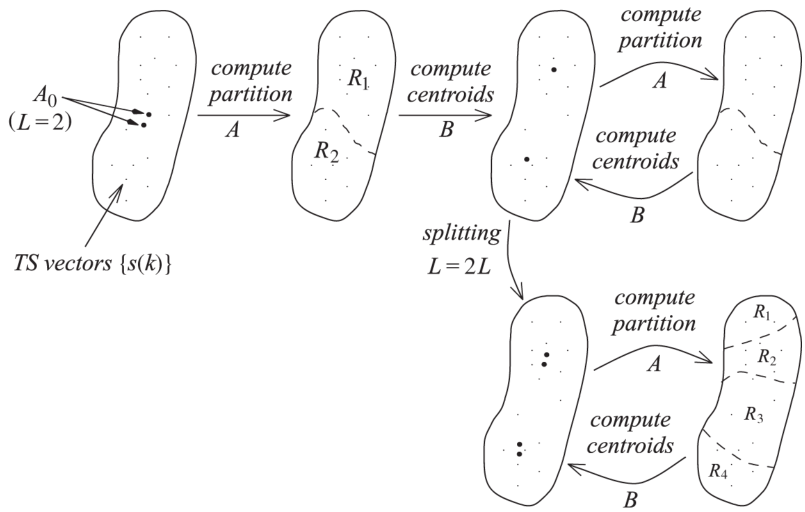


Figure 2.1: LBG algorithm scheme (credits to Algorithms for Communications Systems and their Applications [2])

[1]Note: the LBG algorithm could start by considering multiple, different vectors of the TS, $L > 1$, but they have to be sufficiently spaced in time from each other.

2.2 Encoding procedure

We now consider the Encoder 1 (Fig. 1.3(b)) and we suppose to have a given dataset $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_N\}$ in which each entry $\mathcal{S}_n = \{\mathbf{X}_n, \mathbf{Y}_n\}$ is composed of an instance $\mathbf{X}_n$ of the random variable $\mathbf{X}$ and the corresponding instance $\mathbf{Y}_n$ of the random variable $\mathbf{Y}$, so that an element of the dataset represents an event where $\mathbf{X} = \mathbf{X}_n$ and $\mathbf{Y} = \mathbf{Y}_n$.

The encoder starts by analyzing the dataset and by applying for each value $\mathbf{Y}_i$ present the LGB algorithm on a set $\mathcal{B}_i$ that includes only the values $\mathbf{X}_n$ found in the dataset in the entries containing the considered $\mathbf{Y}_i$; for each application of the LGB algorithm a codebook $\mathcal{C}_i$ (corresponding to each $\mathbf{Y}_i$) is obtained.

$$\mathcal{C}_i = LBG(\mathcal{B}_i) \qquad \mathcal{B}_i = \{\mathbf{X}_n | \{\mathbf{X}_n, \mathbf{Y}_i\} \in \mathcal{S}\} \quad \forall\, i. \tag{2.2}$$

After this, based on the same data set, the probability for each quantization vector (QV) in the codebook $\mathcal{C}_i$ is calculated and all the codebooks and the probabilities are stored. The Huffman coding is then applied to the QVs of each codebook by using the pre-calculated probabilities and the results of this application are also stored.

The encoder needs to encode $\mathbf{X}$ and it knows the side information $\mathbf{Y}$. The procedure used for the encoding makes use of the pre-stored data: the encoder chooses the codebook $\mathcal{C} = \mathcal{C}_\mu$, it then uses the codebook $\mathcal{C}$ to encode the given $\mathbf{X}$ (by comparing it with all the QVs of $\mathcal{C}$ and by choosing the nearest), as last step the QV obtained is codified by using the Huffman code calculated before and the coded information $\mathbf{W}$ is sent on the communication channel[2].

The nearest QV to $\mathbf{X}$ is chosen by considering the Euclidean distance as follows:

$$\mathbf{Q} = \underset{\mathbf{Q}_k \in \mathcal{C}}{\arg\min} \|\mathbf{X} - \mathbf{Q}_k\|, \tag{2.3}$$

where $\mathbf{Q}_k$ represents the *k-th* QV in codebook $\mathcal{C}$.

A diagram showing the encoding procedure is reported in Fig.2.2.

[2]Note that the Huffman coding procedure is a lossless source coding procedure used to reduce the mean number of bits sent on the communication channel. In this case in the methods described in Sections 2.6, 2.7, 2.8 and 2.9, to evaluate the performances (Chapter 3), the Huffman coding is omitted and the r.v. $\mathbf{Q}$ and $\mathbf{W}$ can be used interchangeably.

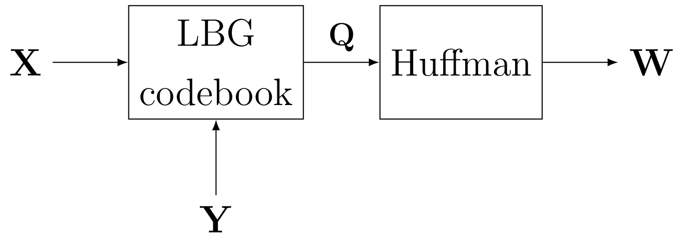


Figure 2.2: Encoding procedure

2.3 Decoding procedure

We suppose that a dataset is present also at the decoder (Decoder 1, Fig. 1.3(b)). In this dataset, represented by $\mathcal{S}' = \{\mathcal{S}'_1, \mathcal{S}'_2, ..., \mathcal{S}'_N\}$, each entry $\mathcal{S}'_n = \{\mathbf{X}_n, \mathbf{Z}_n, \mathbf{Q}_n\}$ is composed of an instance $\mathbf{X}_n$ of the random variable $\mathbf{X}$, the corresponding instance $\mathbf{Z}_n$ of the random variable $\mathbf{Z}$ and a QV $\mathbf{Q}_n$ associated to the corresponding $\mathbf{Z}_n$ and $\mathbf{X}_n$, so that an element of the dataset represents an event where $\mathbf{X} = \mathbf{X}_n$, $\mathbf{Z} = \mathbf{Z}_n$ and $\mathbf{Q} = \mathbf{Q}_n$ [3].

The decoder is given the side information $\mathbf{Z}$ and it uses the following decoding procedure: firstly it decodes the incoming, Huffman coded, information $\mathbf{W}$ so it obtains the same QV $\mathbf{Q}$ created at the encoder; after this, it uses the known $\mathbf{Z}$ and $\mathbf{Q}$ to select from the dataset all the $\mathbf{X}_n$ that appear in the entries $\mathcal{S}'_n = \{\mathbf{X}_n, \mathbf{Z}_n, \mathbf{Q}_n\}$ where $\mathbf{Z}_n = \mathbf{Z}$ and $\mathbf{Q}_n = \mathbf{Q}$.

$$\mathcal{M} = \{\mathbf{X}_n \mid \{\mathbf{X}_n, \mathbf{Z}, \mathbf{Q}\} \in \mathcal{S}'\}. \tag{2.4}$$

The decoded version $\hat{\mathbf{X}}$ of $\mathbf{X}$ is then determined as follows:

$$\hat{\mathbf{X}}(\mathbf{Z}, \mathbf{Q}) = \mathbb{E}[\mathbf{X}|\mathbf{Z}, \mathbf{Q}]. \tag{2.5}$$

The decoder, however, determines the expected value by calculating the mean between all the $\mathbf{X}_n \in \mathcal{M}$.
A diagram showing the decoding procedure is reported in Fig.2.3.

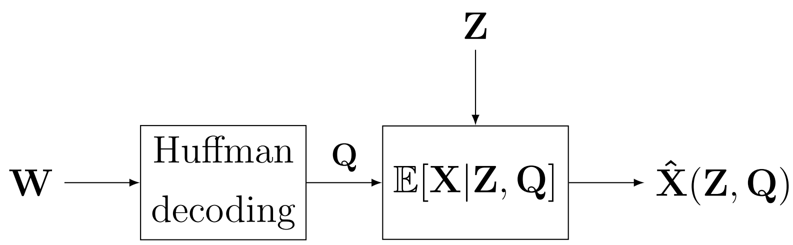


Figure 2.3: Decoding procedure

[3]Note: $\mathbf{Y}$ represents the coded version of $\mathbf{Z}$ determined by the decoder, so it is possible to calculate the same QV for the same $\mathbf{X}_n$ as the encoder.

2.4 Datasets

For all the methods we suppose that at the encoder an extended version $\mathcal{S}''$ of the dataset $\mathcal{S}$ defined in the proposed version (Section 2.2) is present.
An entry $\mathcal{S}''_n = \{\mathbf{X}_n, \mathbf{Y}_n, \mathbf{Z}_n\}$ of the dataset $\mathcal{S}''$ is composed of an instance $\mathbf{X}_n$ of the random variable $\mathbf{X}$, the corresponding instance $\mathbf{Y}_n$ of the random variable $\mathbf{Y}$ and the corresponding instance $\mathbf{Z}_n$ of the random variable $\mathbf{Z}$, so that an element of the dataset represents an event where $\mathbf{X} = \mathbf{X}_n$, $\mathbf{Y} = \mathbf{Y}_n$ and $\mathbf{Z} = \mathbf{Z}_n$ (remember that $\mathbf{Y}$ represents an encoded version of $\mathbf{Z}$ so the extended dataset $\mathcal{S}''$ is equivalent to the original dataset $\mathcal{S}$ but has an additional field).
The used datasets $\mathcal{S}''$ (encoder's dataset) and $\mathcal{S}'$ (decoder's dataset) are the same for all the methods.

2.5 Encoding of Y from Z

Let $\mathcal{Z}$ be the set containing all the the instances $\mathbf{Z}_n$ of $\mathbf{Z}$ present in the dataset $\mathcal{S}''$ and $\mathcal{Y}$ be the set containing all the the instances $\mathbf{Y}_i$ of $\mathbf{Y}$.
We define 2 cases YFZ ($\mathbf{Y}$ From $\mathbf{Z}$) and YFTX ($\mathbf{Y}$ From Tilde $\mathbf{X}$) that differs for the way of encoding $\mathbf{Y}$ from $\mathbf{Z}$.

YFZ: the instances $\mathbf{Y}_i$ of $\mathbf{Y}$ are obtained by applying the LBG algorithm on the instances $\mathbf{Z}_n$ of $\mathbf{Z}$ present in the dataset $\mathcal{S}''$ (see Section 2.4 for the definition of $\mathcal{S}''$), so that:

$$\mathcal{Y} = LGB(\mathcal{Z}). \tag{2.6}$$

YFTX: the instances $\mathbf{Y}_i$ of $\mathbf{Y}$ are obtained by applying the LBG algorithm on a set $\mathcal{G}$. This set is obtained by selecting for each instance $\mathbf{Z}_n$ all the instances $\mathbf{X}_n$ appearing in the dataset $\mathcal{S}''$ in the same entries as $\mathbf{Z}_n$ and by calculating $\tilde{\mathbf{X}}_n(\mathbf{Z}_n)$ as the mean between the selected $\mathbf{X}_n$, so:

$$\mathcal{G} = \{\tilde{\mathbf{X}}(\mathbf{Z}) | \tilde{\mathbf{X}}(\mathbf{Z}) = \mathbb{E}[\mathbf{X}|\mathbf{Z}]\} \quad \text{and} \quad \mathcal{Y} = LGB(\mathcal{G}). \tag{2.7}$$

By using YFTX we exploit the knowledge about $\mathbf{Z}$ to condense in $\mathbf{Y}$ what is already known, at an endpoint of communication, about $\mathbf{X}$ instead of trying to directly encode $\mathbf{Z}$ as it is done in the YFZ case. This allows to improve the performance because $\mathbf{Y}$ is sent to the endpoint that needs to encode $\mathbf{X}$ and by using YFTX instead of YFZ this endpoint is able to better understand what the other endpoint already knows.

2.6 NSI method

In the NSI method the encoder (Encoder 1, Fig. 2.4) creates a codebook $\mathcal{C}$ by applying the LBG algorithm to the instances of $\mathbf{X}$ present in the encoder's dataset $\mathcal{S}''$ (see Section 2.4 for the definition of $\mathcal{S}''$) and stores it. To encode $\mathbf{X}$ the encoder calculates the corresponding QV $\mathbf{Q}$ based on the codebook $\mathcal{C}$ and sends it to the decoder (Decoder 1, Fig. 2.4) that uses $\mathbf{Q}$ as the decoded version of $\mathbf{X}$, so in this case:

$$\mathbf{Q} = LBG(\mathbf{X}), \quad \text{and} \quad \hat{\mathbf{X}} = \mathbf{Q}. \tag{2.8}$$

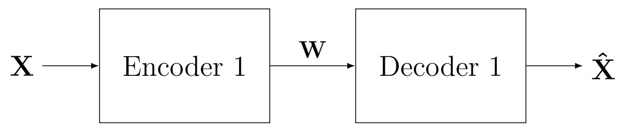


Figure 2.4: NSI scheme.

2.7 OZ method

In the OZ method the Decoder 1 calculates for each instance of $\mathbf{Z}$ present in the dataset a corresponding instance of $\hat{\mathbf{X}}(\mathbf{Z})$ as the mean of all the $\mathbf{X}_n$ appearing in the dataset in the entries when $\mathbf{Z} = \mathbf{Z}_n$, i.e.,

$$\hat{\mathbf{X}}(\mathbf{Z}) = \mathbb{E}[\mathbf{X}|\mathbf{Z}]. \tag{2.9}$$

For each instance $\mathbf{X}_m$ of $\mathbf{X}$ to encode the decoder obtains the corresponding $\hat{\mathbf{X}}_m(\mathbf{Z}_m)$ by considering only the $\mathbf{Z}_m$ related to the $\mathbf{X}_m$ (both given).

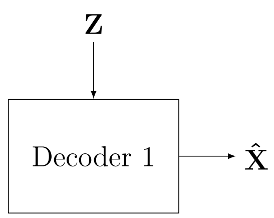


Figure 2.5: OZ scheme.

2.8 LZ and LTX methods

The LZ and the LTX methods implement the proposed procedure (explained in Sections 2.2 and 2.3) but they differ for the way of encoding $\mathbf{Y}$ from $\mathbf{Z}$. In the LZ method, $\mathbf{Y}$ is coded using the YFZ method of Section 2.5 and in the LTX method $\mathbf{Y}$ is coded using the YFTX method of Section 2.5.

The LZ method, according to the proposed procedure and YFZ, uses (2.6) and determines:

$$\mathcal{C}_i = LBG(\mathcal{B}_i) \qquad \mathcal{B}_i = \{\mathbf{X}_n | \{\mathbf{X}_n, \mathbf{Y}_i\} \in \mathcal{S}''\} \quad \forall\, i. \tag{2.10}$$

For each couple of instances $\{\mathbf{X}_m, \mathbf{Z}_m\}$ of $\mathbf{X}$ and $\mathbf{Z}$ to encode, the method obtains $\mathbf{Y}_m$ from $\mathbf{Z}_m$ ($\mathbf{Y}_m$ is the nearest $\mathbf{Y}_i$ to $\mathbf{Z}_m$) and encodes and decodes $\mathbf{X}_m$ as follows:

$$\mathbf{Y}_m = \arg\min_{\mathbf{Y}_i} \|\mathbf{Z}_m - \mathbf{Y}_i\|, \tag{2.11}$$

$$\mathcal{C} = \mathcal{C}_i \quad \text{such that} \quad \mathbf{Y}_i = \mathbf{Y}_m, \tag{2.12}$$

$$\mathbf{Q}_m = \arg\min_{\mathbf{Q}_k \in \mathcal{C}} \|\mathbf{X}_m - \mathbf{Q}_k\|, \tag{2.13}$$

$$\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m) = \mathbb{E}[\mathbf{X} | \mathbf{Z}_m, \mathbf{Q}_m]. \tag{2.14}$$

The LTX method, according to the proposed procedure and YFTX, uses (2.7), creates the codebook by using (2.10) and for each given couple $\{\mathbf{X}_m, \mathbf{Z}_m\}$ the method obtains $\mathbf{Y}_m$ from $\mathbf{Z}_m$ ($\mathbf{Y}_m$ is the nearest $\mathbf{Y}_i$ to $\tilde{\mathbf{X}}(\mathbf{Z}_m) \in \mathcal{G}$) and encodes and decodes $\mathbf{X}_m$ in the same way as the LZ method (see (2.12), (2.13), and (2.14)).

The LZ and LTX methods, compared to the NSI and OZ method, offer the same advantages as the proposed procedure explained in Chapter 1 Section 1.2.
The LTX method in addition to the LZ method improves the way of encoding $\mathbf{Y}$ as explained in Section 2.5.

2.9 LZHZ and LTXHZ methods

The LZHZ and the LTXHZ methods correspond respectively to the LZ and to the LTX methods but they differ from them for the way of calculating the codebooks $\mathcal{C}_i$.
In this case, for each unique couple $\{\mathbf{X}, \mathbf{Y}\}$ present in the dataset $\mathcal{S}''$ (see Section 2.4 for the definition of $\mathcal{S}''$) $\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y})$ is determined from all the $\mathbf{Z}$ possible by using the MAP (maximum a posteriori) criterion. After that, the dataset $\mathcal{S}''$ is extended with a fourth random variable $\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y})$ that indicates for each couple $\{\mathbf{X}_n, \mathbf{Y}_n\}$ the instance $\hat{\mathbf{Z}}_n$ of $\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y})$ chosen, so now an entry of the dataset is composed like this: $\mathcal{S}''_n = \{\mathbf{X}_n, \mathbf{Y}_n, \mathbf{Z}_n, \hat{\mathbf{Z}}_n\}$.
A codebook $\mathcal{C}_j$ is then calculated for each instance $\hat{\mathbf{Z}}_j$ of $\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y})$ in the same way that a codebook $\mathcal{C}_i$ was calculated for each $\mathbf{Y}_i$, so, in this case:

$$\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y}) = \arg\max_{\mathbf{Z}} P(\mathbf{Z}, \mathbf{X}, \mathbf{Y}) \quad \text{for each couple} \quad \{\mathbf{X}, \mathbf{Y}\} \in \mathcal{S}'', \tag{2.15}$$

$$\mathcal{C}_j = LBG(\mathcal{B}_j) \qquad \mathcal{B}_j = \{\mathbf{X}_n | \hat{\mathbf{Z}}_n = \hat{\mathbf{Z}}_j\} \quad \forall\, j. \tag{2.16}$$

After this setup the encoding and the decoding procedure are the same followed by the LZ and the LTX methods (see Section 2.8) but instead of using $\mathbf{Y}$, $\hat{\mathbf{Z}}(\mathbf{X}, \mathbf{Y})$ is used to choose the correct codebook.

The LZHZ method determines $\mathbf{Y}$ from (2.6) and for each couple of instances $\{\mathbf{X}_m, \mathbf{Z}_m\}$ of $\mathbf{X}$ and $\mathbf{Z}$ to encode, the method obtains $\mathbf{Y}_m$ from $\mathbf{Z}_m$ ($\mathbf{Y}_m$ is the nearest $\mathbf{Y}_i$ to $\mathbf{Z}_m$), $\hat{\mathbf{Z}}_m$ from the couple $\{\mathbf{X}_m, \mathbf{Y}_m\}$ (by comparing the couple $\{\mathbf{X}_m, \mathbf{Y}_m\}$ with the couples $\{\mathbf{X}_n, \mathbf{Y}_n\}$ present in the encoder's dataset $\mathcal{S}''$) and encodes and decodes $\mathbf{X}_m$ as follows:

$$\mathbf{Y}_m = \arg\min_{\mathbf{Y}_i} \|\mathbf{Z}_m - \mathbf{Y}_i\|, \tag{2.17}$$

$$\hat{\mathbf{Z}}_m = \hat{\mathbf{Z}}(\mathbf{X}_m, \mathbf{Y}_m), \tag{2.18}$$

$$\mathcal{C} = \mathcal{C}_j \quad \text{such that} \quad \hat{\mathbf{Z}}_j = \hat{\mathbf{Z}}_m, \tag{2.19}$$

$$\mathbf{Q}_m = \arg\min_{\mathbf{Q}_k \in \mathcal{C}} \|\mathbf{X}_m - \mathbf{Q}_k\|, \tag{2.20}$$

$$\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m) = \mathbb{E}[\mathbf{X}|\mathbf{Z}_m, \mathbf{Q}_m]. \tag{2.21}$$

The LTXHZ method determines $\mathbf{Y}$ from 2.7 and for each given couple $\{\mathbf{X}_m, \mathbf{Z}_m\}$ the method obtains $\mathbf{Y}_m$ from $\mathbf{Z}_m$ ($\mathbf{Y}_m$ is the nearest $\mathbf{Y}_i$ to $\tilde{\mathbf{X}}(\mathbf{Z}_m)$), $\hat{\mathbf{Z}}_m$ from the couple $\{\mathbf{X}_m, \mathbf{Y}_m\}$ (by comparing the couple $\{\mathbf{X}_m, \mathbf{Y}_m\}$ with the couples $\{\mathbf{X}_n, \mathbf{Y}_n\}$ present in the encoder's dataset $\mathcal{S}''$) and encodes and decodes $\mathbf{X}_m$ in the same way as the LZHZ method (see (2.18), (2.19), (2.20), and (2.21)).

The LZHZ method, compared to the LTXHZ method, improves the way of encoding $\mathbf{Y}$ as explained is Section 2.5.

The LZHZ and the LTXHZ methods, compared to the LZ and the LTX methods, refine the way of creating and selecting the correct codebook by choosing most probable $\mathbf{Z}$ for the given couple $\{\mathbf{X}, \mathbf{Y}\}$ by using the MAP criterion, instead of using only $\mathbf{Y}$, in order to reduce the error probability.

Chapter 3

Performance Results

In this chapter we describe how we implemented the proposed procedures of Chapter 2 to test the performance improvement. Six different tests were made, one for each proposed method (NSI, OZ, LZ, LTX, LZHZ and LTXHZ).

All these tests have been executed multiple times by increasing the number of bits b_B (see Chapter 1 Section 1.3 for the definition of b_B) used to represent the QVs to be sent on the communication channel, starting from 0 to maximum possible bit number and the performance have been evaluated each time.
Note that in all these tests the Huffman coding is omitted because it is a lossless source coding used to reduce the average number of bits sent on the communication channel (by knowing the PDF of the data) and, in this case, it doesn't affect the performances.
The number of bits b_F (see Chapter 1 Section 1.4 for the definition of b_F) has been set to 2, so for each test only four different instances of $\mathbf{Y}$ were used.

The performance were determined by applying the tests on a test dataset $\mathcal{T}$ where each entry is composed as follows: $\mathcal{T}_m = \{\mathbf{X}_m, \mathbf{Z}_m\}$ where $\mathbf{X}_m$ represents an instance of $\mathbf{X}$ and $\mathbf{Z}_m$ represents an instance of $\mathbf{Z}$.
The used datasets $\mathcal{S}''$ (encoder's dataset), $\mathcal{S}'$ (decoder's dataset) and $\mathcal{T}$ (test dataset) were the same for all the tests.

The dataset $\mathcal{S}''$ that were used were composed of 10000 entries, the dataset $\mathcal{T}$ of 1000 entries.
The instances of $\mathbf{X}$ are four dimensional vectors obtained by the product of a four-by-four, fixed, correlation matrix $\mathbf{C1}$ of real numbers and a four dimensional vector $\mathbf{S1}$, whose elements are instances of a normally distributed random variable. Each instance of $\mathbf{Z}$ is related to the corresponding instance of $\mathbf{X}$ because these two random variables represent an uplink and a downlink channel that are related (see Chapter 1), so it is obtained by creating a four dimensional vector $\mathbf{S3}$ calculated as the product of a four-by-four, fixed, correlation matrix $\mathbf{C2}$ of real numbers and a four dimensional vector $\mathbf{S2}$, whose elements are instances of a normally distributed random variable, and by summing $\frac{1}{3} \cdot \mathbf{S3}$ and $\frac{2}{3} \cdot \mathbf{X}$.

We summarize the procedure used to obtain the instances of $\mathbf{X}$ and $\mathbf{Z}$:

$$\mathbf{X} = \mathbf{C1} \cdot \mathbf{S1}, \tag{3.1}$$

$$\mathbf{Z} = \frac{1}{3} \cdot \mathbf{X} + \frac{1}{3} \cdot \mathbf{C2} \cdot \mathbf{S2}. \tag{3.2}$$

Each element of all the instances of $\mathbf{X}$ and $\mathbf{Z}$ is then quantized to reduce the amount of possible vectors by using $M = 8$ quantization levels. The quantization is applied as follows:

$$q = \max(\min(\lfloor \frac{x}{\sigma} \cdot M \rceil, \frac{M}{2}), \frac{-M}{2} + 1), \tag{3.3}$$

where q represents the quantized element, x the original element of the vector and σ the standard deviation of the random variable.

The matrices $\mathbf{C1}$ and $\mathbf{C2}$ used to determine the instances of $\mathbf{X}$ and $\mathbf{Y}$ are the following:

$$\mathbf{C1} = \begin{bmatrix} 89 & 94 & 74 & 38 \\ 49 & 29 & 99 & 16 \\ 84 & 6 & 58 & 10 \\ 56 & 99 & 67 & 49 \end{bmatrix}, \quad \mathbf{C2} = \begin{bmatrix} 17 & 38 & 42 & 7 \\ 31 & 0 & 90 & 60 \\ 96 & 36 & 35 & 32 \\ 73 & 94 & 97 & 85 \end{bmatrix}. \tag{3.4}$$

3.1 Error calculation

To calculate the error we consider the instances $\mathbf{X}_m$ of $\mathbf{X}$ and their decoded version $\hat{\mathbf{X}}_m$. For each $\mathbf{X}_m$ we calculate the L^2-norm as $\|\mathbf{X}_m\|^2$ and the mean square error (MSE) as $\|\hat{\mathbf{X}}_m - \mathbf{X}_m\|^2$. After that, we determine the expected values of both the L^2-norm and the MSE (the expected value, in this case, is calculated as the mean between the values obtained for each $\mathbf{X}_m$) and we divide the expected value of the MSE by the expected value of the L^2-norm, i.e.,

$$\text{NMSE} = \frac{\mathbb{E}[\, \|\hat{\mathbf{X}}_m - \mathbf{X}_m\|^2 \,]}{\mathbb{E}[\, \|\mathbf{X}_m\|^2 \,]}. \tag{3.5}$$

3.2 NSI

To test the NSI method (Section 2.6) we apply it to each $\mathbf{X}_m$ present in the test dataset $\mathcal{T}$.

The error is then calculated by considering $\mathbf{X}_m$ and its decoded version $\hat{\mathbf{X}}_m$ (see Section 3.1 to understand how the error is determined) and the results are reported in Fig. 3.1.

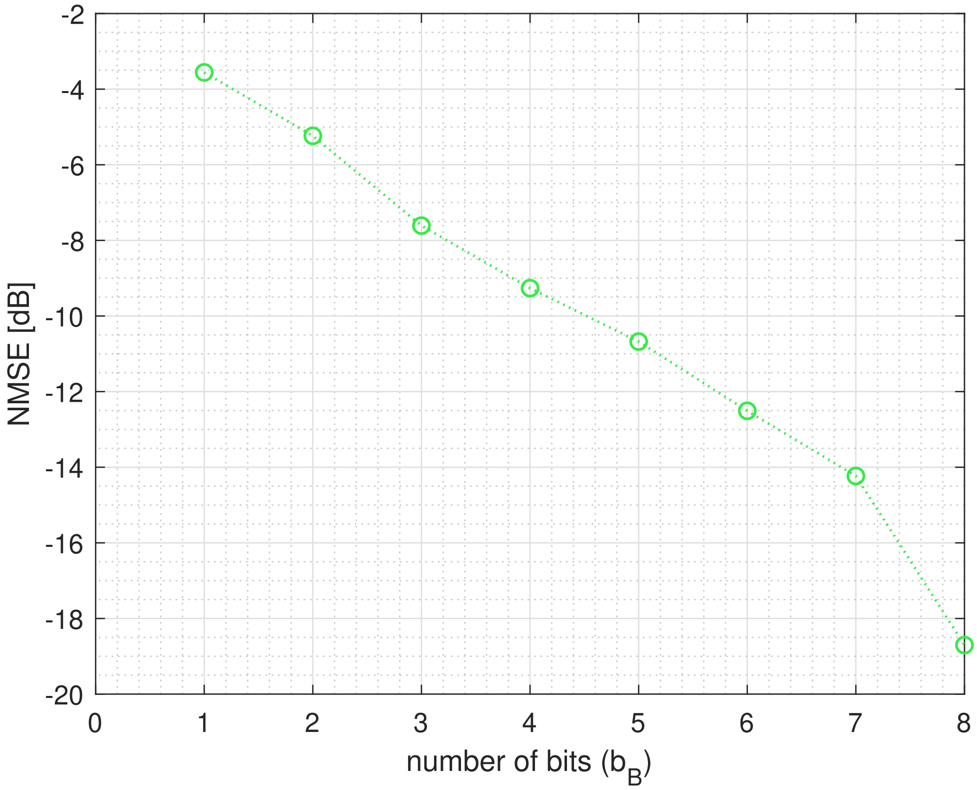


Figure 3.1: Test without side information

In Fig. 3.1 the abscissa axis contains the number of bit b_B used to represent the QV $\mathbf{Q}$ (so if bits=1 only two different instances of $\mathbf{Q}$ and, consequently, of $\mathbf{W}$ are possible, if bits=2 only four instances of $\mathbf{Q}$ and $\mathbf{W}$ etc.) and the ordinate axis indicates the NMSE in decibel.

As can be seen from Fig. 3.1 by increasing the number of bits the error decreases, as expected.

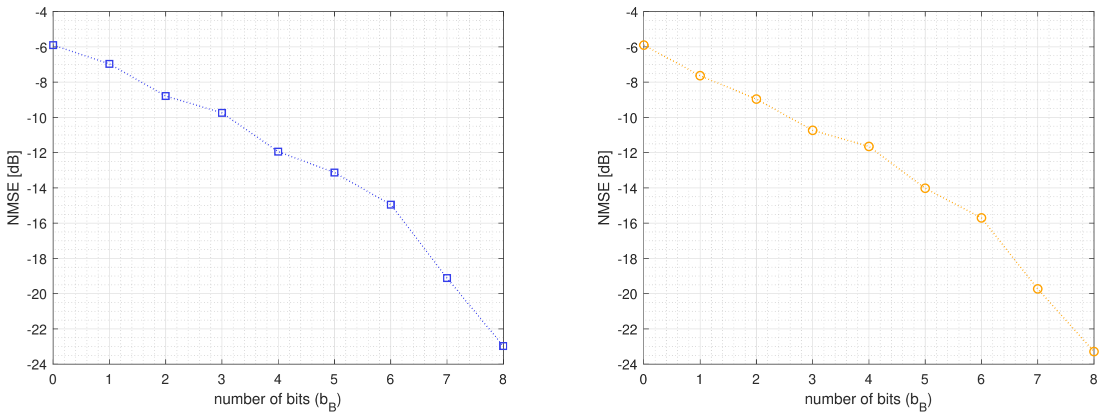
3.3 OZ

To test the OZ method (Section 2.7) we apply the procedure to each $\mathbf{X}_m$ present in the test dataset $\mathcal{T}$.
For each instance $\mathbf{X}_m$ the decoder obtains the corresponding $\hat{\mathbf{X}}_m(\mathbf{Z}_m)$. The NMSE is then calculated by considering $\mathbf{X}_m$ and $\hat{\mathbf{X}}_m(\mathbf{Z}_m)$ (see Section 3.1 to understand how the error is determined). In this case the value reported is only one because in this case no $\mathbf{W}$ is sent on the communication channel so the error is the same for each possible number of bits. The results is NMSE=-5,9 dB.

3.4 LZ and LTX

The performances of the LZ and LTX are evaluated separately. We apply the methods to each $\mathbf{X}_m$ present in the test dataset $\mathcal{T}$ to obtain the corresponding $\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m)$.

The NMSE is then calculated in the same way for both the LZ and the LTX methods, by considering $\mathbf{X}_m$ and $\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m)$ (see Section 3.1 to understand how the error is determined) and the results are reported in Fig. 3.2.



(a) LZ method

(b) LTX method

Figure 3.2: Tests error diagrams

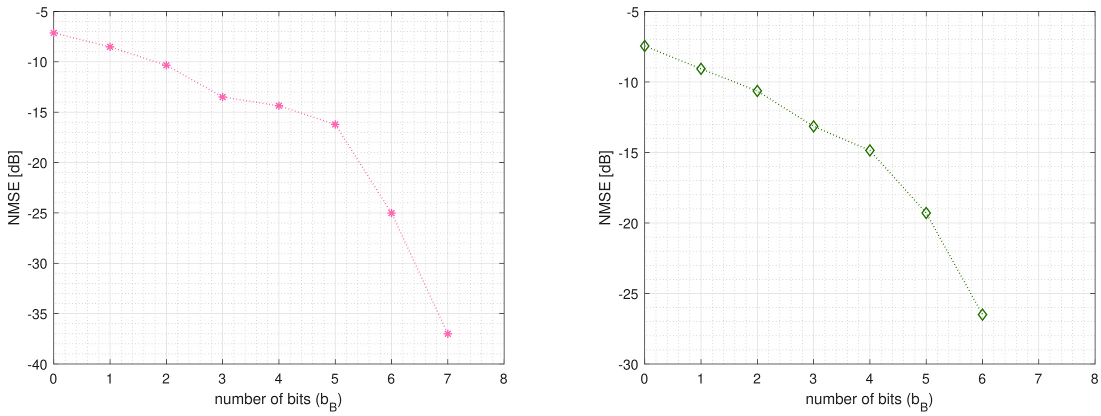
In both Fig. 3.2(a) and Fig. 3.2(b) the abscissa axis contains the number of bits b_B used to represent the QV $\mathbf{Q}$ (so if bits=1 only two different instances of $\mathbf{Q}$ and, consequently, of $\mathbf{W}$ are possible, if bits=2 only four instances of $\mathbf{Q}$ and $\mathbf{W}$ etc.) and the ordinate axis indicates the NMSE in decibel.

As can be seen in Fig. 3.2 the performances of the LTX method in most cases are better than the LZ method, that is due to the fact that the LZ method uses the YFZ procedure to encode $\mathbf{Y}$ and the LTX method uses the YFTX procedure to encode $\mathbf{Y}$ and this allows to better exploit the side information $\mathbf{Z}$ as explained in Chapter 2 Section 2.5.

3.5 LZHZ and LTXHZ

The performances of the LZHZ and LTXHZ are evaluated separately. We apply the methods to each $\mathbf{X}_m$ present in the test dataset $\mathcal{T}$ to obtain the corresponding $\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m)$.

The NMSE is then calculated in the same way for both the LZHZ and the LTXHZ methods, by considering $\mathbf{X}_m$ and $\hat{\mathbf{X}}_m(\mathbf{Z}_m, \mathbf{Q}_m)$ (see Section 3.1 to understand how the error is determined) and the results are reported in Fig. 3.3.



(a) LZHZ method

(b) LTXHZ method

Figure 3.3: Tests error diagrams

In both Fig. 3.3(a) and Fig. 3.3(b) the abscissa axis contains the number of bits b_B used to represent the QV $\mathbf{Q}$ (so if bits=1 only two different instances of $\mathbf{Q}$ and, consequently, of $\mathbf{W}$ are possible, if bits=2 only four instances of $\mathbf{Q}$ and $\mathbf{W}$ etc.) and the ordinate axis indicates the NMSE in decibel.

As can be seen in Fig. 3.3 the performances of the LTXHZ method in most cases are better than the LZHZ method, that is due to the fact that the LZHZ method uses the YFZ procedure to encode $\mathbf{Y}$ and the LTXHZ method uses the YFTX procedure to encode $\mathbf{Y}$ and this allows to better exploit the side information $\mathbf{Z}$ as explained in Chapter 2 Section 2.5.

3.6 Comparison

We report a diagram showing the performance of all the tested methods:

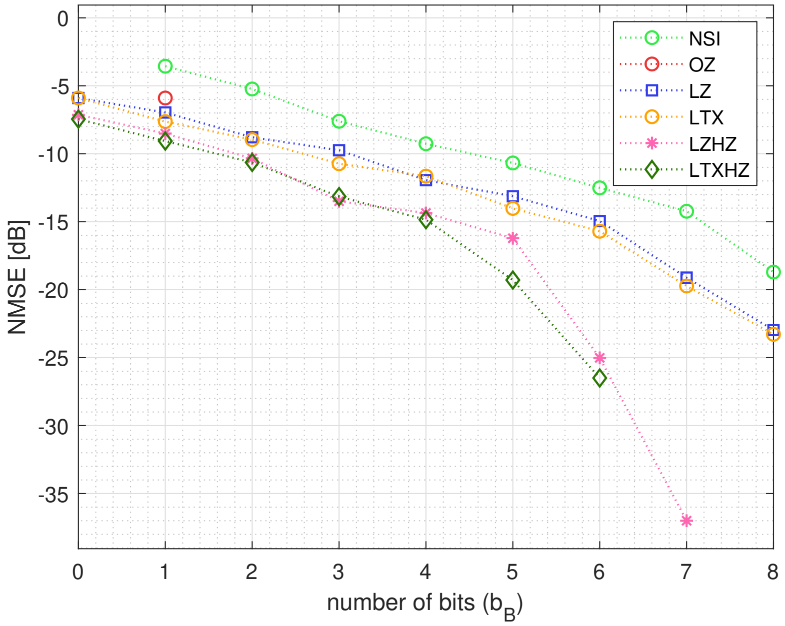


Figure 3.4: Comparison between all the methods

As can be seen from Fig. 3.4, the performance of the proposed version of the algorithm are considerably better than the NSI and the OZ methods that represent a standard procedure.

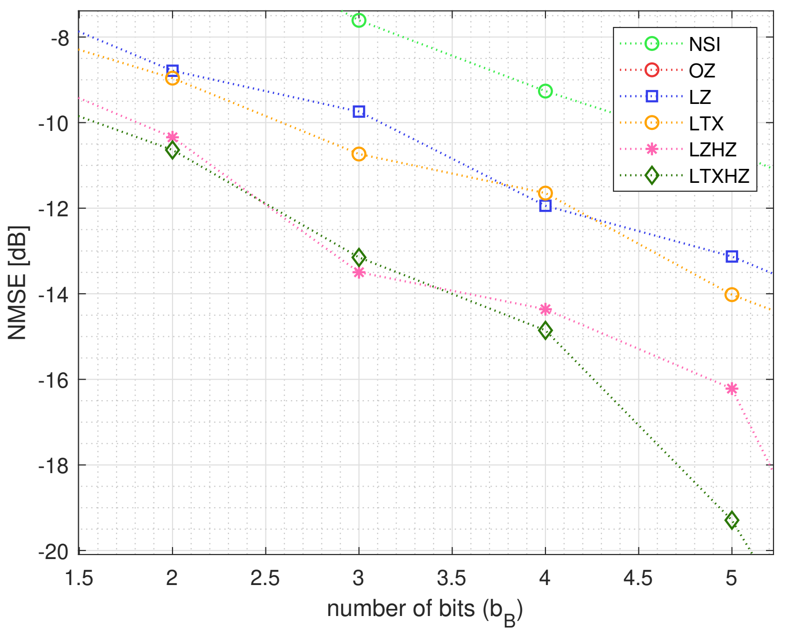
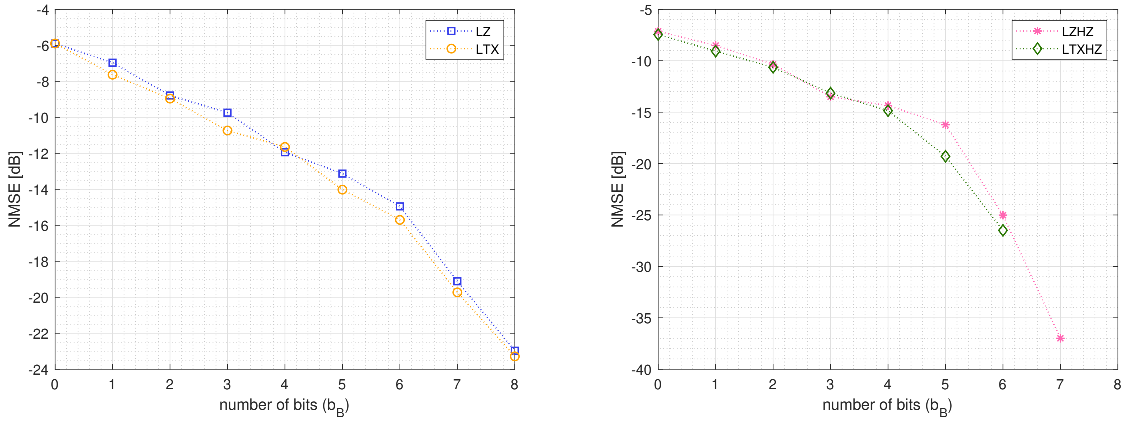


Figure 3.5: LZ, LTX, LZHZ and LTXHZ methods tests

In Fig. 3.5 can be seen as the LZHZ and the LTXHZ methods have better performances than the LTX and the LZ methods. This happens because by creating the codebooks by

considering $\hat{\mathbf{Z}}$ than $\mathbf{Y}$, at the encoder, we exploit even more the information known about $\mathbf{Z}$. That is due to the fact that by knowing $\mathbf{X}$ and $\mathbf{Y}$ we choose the most probable $\mathbf{Z}$ (see Section 2.9) so we reduce the error probability.
By increasing the number of bits the performances of the proposed version's are likely the same because the number of possible QVs becomes more similar to number of possible $\mathbf{X}$ considered in each codebook so the error tends to zero.



(a) LZ and LTX methods

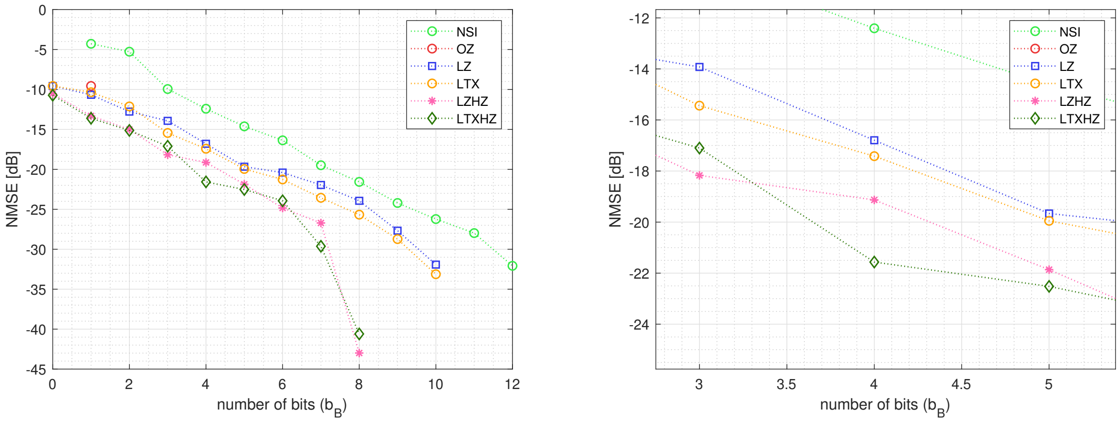
(b) LZHZ and LTXHZ methods

Figure 3.6

Fig. 3.6 shows respectively the LZ and the LTX methods performances (Fig. 3.6(a)) and the LZHZ and the LTXHZ methods performances (Fig. 3.6(b)). As can be seen using the LBG algorithm on $\tilde{\mathbf{X}}$ than directly on $\mathbf{Z}$ does slightly improve the performance.

3.7 Other tests

The same six methods treated in Sections 2.6, 2.7, 2.8 and 2.9 were repeated by changing only M, the number of quantization level.
The results obtained (reported in Fig. 3.7 and 3.8) show how the proposed procedures behave in the same way even if we change M.
By considering $M = 256$ we obtained the results of Fig. 3.7.

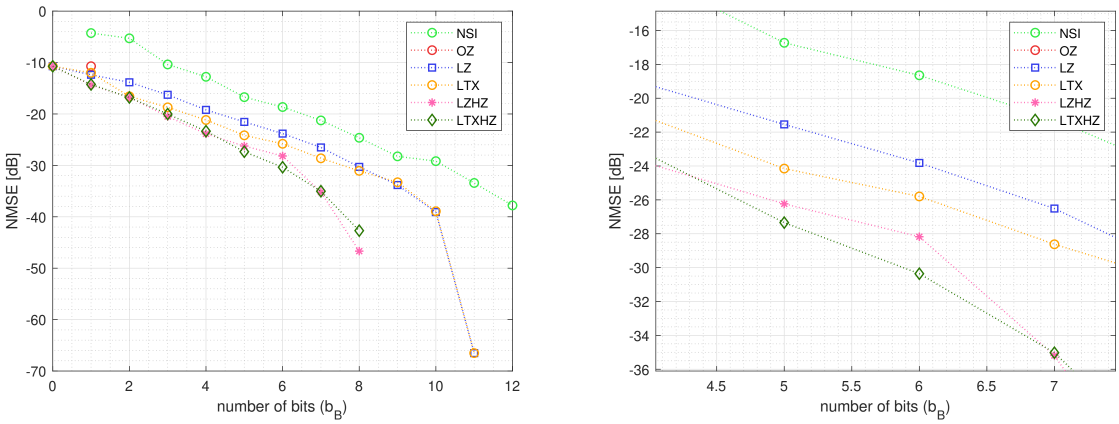


(a) Comparison between all the methods with M=256

(b) LZ, LTX, LZHZ and LTXHZ methods with M=256

Figure 3.7: Tests error diagrams with M=256

By considering $M = 5000$ we obtained the results of Fig. 3.8



(a) Comparison between all the methods with M=5000

(b) LZ, LTX, LZHZ and LTXHZ methods with M=5000

Figure 3.8: Tests error diagrams with M=5000

In both Fig. 3.7(a), 3.7(b), 3.8(a) and 3.8(b) the abscissa axis contains the number of bit used to represent the QV $\mathbf{Q}$ (so if bits=1 only two different instances of $\mathbf{Q}$ and, consequently, of $\mathbf{W}$ are possible, if bits=2 only four instances of $\mathbf{Q}$ and $\mathbf{W}$ etc.) and the ordinate axis indicates the NMSE in decibel.
The OZ method peformances are represented with one single point because this method considers only the decoder (see Section 2.7) so no bits are sent on the communication channel.

Conclusions

In this thesis we dealt with the problem of reducing the amount of data exchanged from two endpoint of communication to describe the behaviour of a communication channel. We proposed some new algorithms and methods and we tested their performances. We demonstrated that the proposed methods, compared to the ”standard” source coding procedures, allow to improve the exchange of information by decreasing the error for the same number of bits.

Bibliography

[1] X. Zhang, M. Centenaro, S. Tomasin, N. Benvenuto, X. Luo, and S. Yang, "A study on csi feedback schemes exploiting feedforward information in fdd cellular systems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e3628, 2021, e3628 ett.3628. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3628

[2] N. Benvenuto, G. Cherubini, and S. Tomasin, *Vector quantization*. John Wiley I& Sons, Ltd, 2021, ch. 5, pp. 229–248. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119567998.ch5