Universidad de Málaga

Escuela Técnica Superior de Ingeniería de Telecomunicación

Tesis Doctoral

ENHANCED LINK ADAPTATION TECHNIQUES FOR CELLULAR NETWORKS

Autor

Francisco Blánquez Casado
Ingeniero en Telecomunicación
Máster Universitario en Tecnologías de Telecomunicación

Directores

Mª Carmen Aguayo Torres y **Gerardo Gómez Paredes**
Doctores Ingenieros de Telecomunicación

Año 2017

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente: Dr. D. ____

Secretario: Dr. D. ____

Vocal: Dr. D. ____

para juzgar la Tesis Doctoral titulada **"Enhanced link adaptation techniques for cellular networks"**, presentada por D. Francisco Blánquez Casado y dirigida por Dra. Mª Carmen Aguayo Torres and Dr. Gerardo Gómez Paredes,

acordó por ____ otorgar la calificación de

____

y, para que conste, se extiende firmada por los componentes del tribunal la presente diligencia.

Málaga, a ____ de ___________ de _____

El presidente: El secretario: El vocal:

Fdo: ________________ Fdo: ________________ Fdo: ________________

Agradecimientos

A todos mis compañeros y amigos del 1.3.5, con los que he pasado, y espero seguir pasando, tan buenos momentos. Todo es más fácil trabajando con gente así.

A Tomás, por darme la oportunidad de trabajar, aprender y hacer esta tesis rodeado de grandes personas.

A Mari Carmen y Gerardo, por toda su ayuda en estos años, y por supuesto por su ayuda en este trabajo.

A mis padres, por todo el ánimo y apoyo que me han dado siempre.

A Gádor, por estar siempre ahí.

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad (Proyecto TEC2013-44442-P).

Contents

List of Figures

List of Tables

Table of acronyms

Acronym	Meaning
3GPP	3rd Generation Partnership Project
aBLER	average BLER
ABS	Almost Blank Subframe
adSOVA	adaptive Soft Output Viterbi Algorithm
AMC	Adaptive Modulation and Coding
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLER	BLock Error Rate
BS	Base Station
CLLA	Closed Loop Link Adaptation
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Code
CRE	Cell Range Expansion
eICIC	enhanced Inter-Cell Interference Coordination
eOLLA	enhanced Outer Loop Link Adaptation
FEC	Forward Error Correction
GSM	Global Special Mobile
HARQ	Hybrid-Automatic Repeat Request
HetNets	Heterogeneous Networks
iBLER	instantaneous BLER
ICI	Inter-Cell Interference
ISI	Inter-Symbol Interference
LA	Link Adaptation
LLR	Log Likelihood Ratio
LQM	Link Quality Metric
LTE	Long Term Evolution

Acronym	Meaning
LTE-A	Long Term Evolution Advanced
M2M	Machine to Machine
MAP	Maximum a Posteriori
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MLDA	Maximum Likelihood Decoding Algorithm
OFDM	Orthogonal Frequency Domain Multiplexing
OLLA	Outer Loop Link Adaptation
PRB	Physical Resource Block
QAM	Quadrature Amplitude Modulation
SINR	Signal to Noise plus Interference Ratio
SISO	Soft Input Soft Ouput
SNR	Signal to Noise Ratio
SOVA	Soft Output Viterbi Algorithm
TB	Transport Block
TTI	Transmission Time Interval
UE	User Equipment
UMTS	Universal Mobile Telecommunication System

Table 1: Table of acronyms.

Abstract

In this thesis, a study of the link adaptation process in LTE is carried out, proposing new techniques to improve its performance. This work is focused on two key parts: 1) the turbo decoding process and 2) the adaptive modulation and coding.

Turbo decoders based on Soft Output Viterbi Algorithm (SOVA) truncate the number of operations of the decoding process to a fix value to reduce complexity. In this work we show how this truncation might degrade the decoding performance if the fixed number of operation is too low, or might carry out unnecessary operations, and so increase power consumption, if it is too high. Furthermore, the appropriate number of operations depends on the transmission conditions, which can be time variant. An adaptive SOVA (adSOVA) is proposed in this thesis to dynamically adapt this number of operations. Then, the adSOVA is able whether to increase the number of operations to avoid degradation in error correction performance or to reduce it to save power without degrading performance, depending on the transmission conditions.

Adaptive modulation and coding process is usually supported by the well-known Outer Loop Link Adaptation (OLLA) technique, which is widely used to ensure a target BLER. However, there is a lack of analysis of the OLLA behavior in the literature. In this thesis a deep analysis of the OLLA is carried out. For this purpose, binary logistic regression has been used to model BLER when turbo coding is used. Results of this analysis are the convergence conditions of the OLLA, as well as the effect of the step sizes in its performance. From the previous analysis of the traditional OLLA, an enhanced OLLA (eOLLA) is proposed, which is able to adapt its step size according to the deviance of the target BLER. In addition to that, the eOLLA is able to track temporal variations of the channel when no transmissions are done, while the traditional OLLA needs transmissions to update its offset. Thus, the proposed eOLLA outperforms the traditional OLLA, especially in low load traffic scenarios. Additionally, binary logistic regression is used to obtain a full analytic model of the adaptive modulation and coding process with channel coding based on turbo coding for constant power in LTE downlink. This model shows that an optimal solution would require of an OLLA with multiple offsets, which would lead to improve the spectral efficiency as much as 20%. Finally, the influence of the variance of the transmitted data block size in adaptive modulation and coding is studied, showing how a high variance might degrade its performance. Then, a Closed Loop Link Adaptation (CLLA) technique is proposed, which is able to modify the modulation and coding scheme suggested by the UE depending on the size of the data block to be transmitted, in order to ensure the target BLER. The CLLA outperforms OLLA in scenarios with high variance of data block sizes.

Resumen

En esta tesis se ha realizado un estudio del proceso de adaptación del enlace en LTE, a partir del cual se proponen nuevas técnicas para mejorar su rendimiento. Este trabajo se centra en dos aspectos fundamentales: 1) el proceso de turbo decodificación, y 2) la modulación y codificación de canal adaptativa.

Los turbo decodificadores SOVA típicamente truncan el número de operaciones a realizar en el proceso de turbo decodificación a un valor fijo, con el objetivo de reducir complejidad. En este trabajo se muestra cómo este truncamiento puede degradar el rendimiento de la turbo decodificación si este valor prefijado de número de operaciones es demasiado bajo, o puede suponer realizar operaciones innecesarias, y por tanto aumentar el consumo de potencia, si es demasiado alto. Este número máximo de operaciones depende de las condiciones de transmisión, que pueden ser variantes en el tiempo. En esta tesis se propone un SOVA adaptativo (adSOVA), capaz bien de aumentar el número de operaciones para evitar una degradación en la tasa de corrección de errores, bien de disminuirlo para reducir el consumo de potencia sin afectar a la capacidad de corrección de errores, en función de las condiciones de transmisión.

La modulación y codificación de canal adaptativa suele combinarse con la conocida técnica OLLA, que ayuda a asegurar el cumplimiento de una BLER objetivo. Sin embargo, hay una falta de análisis del comportamiento de la técnica OLLA en la literatura. En esta tesis se ha realizado un profundo estudio de la técnica OLLA. Para ello, la regresión logística binaria ha sido empleada para modelar la BLER cuando se emplea turbo codificación. De este estudio se han obtenido las condiciones de convergencia, así como el efecto del tamaño del paso en el rendimiento. A partir del análisis anterior, se ha propuesto un OLLA mejorado (eOLLA), capaz de adaptar el tamaño de su paso según la desviación de la BLER respecto a un valor objetivo. Además, el eOLLA es capaz de seguir las variaciones temporales del canal incluso cuando no se están realizando transmisiones, al contrario que el OLLA tradicional. Por tanto, el eOLLA propuesto mejora al OLLA tradicional, especialmente en escenarios de baja tasa de tráfico.

Adicionalmente, la regresión logística binaria se ha usado para obtener un modelo analítico completo del proceso de modulación y codificación de canal adaptativa en el enlace descendente de LTE, cuando se emplea codificación de canal basada en turbo codificación, para potencia constante. Este modelo muestra que una solución óptima requeriría de un OLLA con múltiples offsets, lo que llevaría a una mejora de la eficiencia espectral de hasta un 20%. Finalmente, la influencia en la modulación y codificación de canal adaptativa de la varianza del tamaño de los paquetes de datos transmitidos es estudiada, mostrando cómo una alta varianza puede degradar su rendimiento. En esta tesis se propone la técnica de Adaptación del Enlace de Bucle Cerrado (CLLA), que permite modificar el esquema de modulación y codificación propuesto por el UE en función del tamaño del bloque a transmitir, con el objetivo de satisfacer la BLER objetivo. La técnica CLLA mejora a la técnica OLLA en escenarios con gran varianza en el tamaño de los datos a transmitir.

Chapter 1

Introduction

1.1 Antecedents and motivation

The growth in the demand for wideband wireless communication systems together with the improvement in the transmission rates indicate a promising future for this sector in the following years. At the present time Long Term Evolution (LTE) [1] networks have been deployed in many countries, which adds wideband features to the Universal Mobile Telecommunications System (UMTS). At the same time, the designing of the next LTE version, named LTE-Advanced (LTE-A) [2], has been finished, and is being deployed too. This new version is able to achieve a peak data rate of 1 Gbps in the downlink. Finally, the evolution of LTE-A, named 5G, is being studied at the present, and the first deployment will be carried out during the Olympic Games of Tokyo, in 2020. The standardization process of these mobile systems has been carried out by agreement between companies of this sector, which meet in forums. In the case of LTE and LTE-A, the forum which is in charge of their standardization is the 3rd Generation Partnership Project (3GPP).

The new mobile communication systems such as LTE are oriented to data transmission, unlike older systems such as the Group Special Mobile (GSM). Therefore, these systems should be able to accommodate a high variety of traffic sources with different characteristics such as streaming (typically used for video) or Machine to Machine (M2M) (which refers to any kind of communications without direct human intervention), without any degradation in the system performance.

One major difficulties that wideband mobile systems presents to achieve linkś capacity, in addition to the frequency selectivity, is that link conditions are time variant. Mobile radio channel suffers from multipath propagation [3] which causes instantaneous variations in the channel state. To solve this problem, a variety of adaptive transmission techniques have been proposed. These techniques tackle the variations of the channel quality by modifying the transmission rate (through changing the modulation scheme or the coding scheme) and/or the power transmission at each frequency band. In addition to that, in recent years it has been extended the use of Orthogonal Frequency Division Multiplexing (OFDM) to counteract the frequency selectivity. The use of Multiple Input Multiple Output (MIMO) techniques and more efficient channel coding techniques, such as turbo codes [4], have also allowed to improve the link adaptation and to achieve higher transmission rates.

Typically, mobile cellular networks have been made up of a number of cells which reuses frequency bands with the purpose of taking the most of the radio electrical spectrum. This frequency reuse may cause a problem of Inter-Cell Interference (ICI) when the distance between cells that are using the same frequency is small. As all cells within an LTE network can use the same frequency band to transmit, the interference problem aggravates and limits the coverage. To alleviate this problem, mechanisms of cooperation which exchange information between cells are introduced in LTE in order to minimize the effect of inter-cell interference. In LTE-A and further standards the cooperation between cells will be used to improve spectral efficiency and transmission rates, specially for those mobile terminals that are located at the edge of the coverage area of a cell. Thus, this eases that all users have access to the system resources independently of their cell position, which allows a fair treatment for all of them.

1.1.1 Main physical layers techniques

Next some techniques used to improve the transmission rate and the link adaptation in mobile systems such as LTE and LTE-A are briefly described. Firstly, static techniques are described.

OFDM The technique that is used in LTE to implement frequency multiplexing for the different data streams is the multi-carrier modulation, also known as OFDM [3][5]. This technique is based on dividing the available transmission bandwidth into frequency bands narrow enough to be considered flat. Then, the transmission can be carried out simultaneously through each of these bands relieving the effect of the frequency selectivity of the mobile channels since the transmission can be adapted individually for each band. Furthermore, no spectral efficiency is lost as transmission pulses are designed so that even being orthogonal, they are frequency overlapped, so there is no need of guard bands.

OFDM also provides a solution for the Inter-Symbol Interference (ISI) [5]. The mobile channel response can cause an important temporal spreading when it is a wideband channel. This spreading can be mitigated through the use of a cyclic extension. The duration of this cyclic extension has to be longer than the duration of the channel response, but shorter than the duration of the OFDM symbols. Thus, ISI can be avoided without a significant loosing of transmission rate. In addition to that, the cyclic extension also avoids losing orthogonality.

Channel coding Channel coding techniques add redundancy to the transmitted symbols in order to avoid errors caused by the channel. Two kinds of strategies can be differentiated at the receiver: 1) error detection to ask data retransmission or Automatic Repeat Request (ARQ), and 2) error correction through the redundant information or Forward Error Correction (FEC). In current mobile communication systems predominate the hybrid systems or Hybrid-Automatic Repeat Request (HARQ) which combines both techniques [1]. In LTE and beyond, channel coding is performed through turbo codes, which are one of the most effective techniques to generate codes with a high error correction capability. This allows to achieve a data transmission rate close to the capacity of the channel [4]. At the receiver, the detection is based on an iterative process which estimates the more likely transmitted sequence based on the calculation of Log Likelihood Ratios (LLR) [3]. Turbo decoders are mainly based on two well-known algorithms: the Maximum A Posteriori (MAP) algorithm [6] and the Soft Output Viterbi Algorithm (SOVA) [7][4]. On the other hand, the main drawback of turbo codes is their high complexity.

MIMO New standards developed for cellular networks have included the mandatory use of multiple antennas for transmission and reception (MIMO) in their specifications. The use of multiple antennas in cellular networks can be used either to reduce the error rate or to increase the transmission rate [5]. For instance, if the link quality is good enough,MIMO schemes can be used with spatial multiplexing, which allow to increase the spectral efficiency of the system through the transmission of several data streams simultaneously. On the other hand, MIMO schemes can be used to improve the transmission quality when experiencing low Signal to Noise Ratio (SNR) due to the diversity gain which achieves techniques such as beamforming, space-time coding or space-frequency coding [8].

Link Adaptation This technique consists in adapting the parameters of the techniques previously described to the temporal variations of the channel. On the one hand, there exists techniques that adapt the Modulation and Coding Scheme (MCS) to the channel state [1, 9–11]. As the mobile channel suffers from continuous changes in its response, an adjustment can be carried out in the constellation, the coding scheme and the transmitted power[1] [13, 14] so that the throughput is maximized while keeping the error probability below a previously established limit. On the other hand, when a multiple antennas (MIMO) scheme is used, it is possible to adapt the transmission parameters, and even the specific MIMO technique to use (for instance, changing the MIMO scheme between spatial multiplexing and beamforming), according to the channel state [15–18].

To perform the Link Adaptation (LA), it is necessary to carry out an estimation of the channel state. From this estimation, the transmitter and the receiver have to negotiate through signaling the parameters to be used during the transmission.

One of the main drawbacks of the LA is the delay that the signaling of the channel state suffers from the receiver to the transmitter through the return channel. The impact of this delay is more pronounced for high speed of mobile terminals, since sent information will be earlier obsolete [19]. Another drawback is the limitation that the return channel introduces in the amount of signaling information and its accuracy due to quantification. This means that to exactly adjust the transmission parameters is not possible, causing a performance degradation.

Multi-Cell techniques With the purpose of reducing ICI, new techniques involving more than one access point has been proposed, which is known as cooperative or coordinated communications. For instance, it is possible to carry out coordinated transmissions in a way such that one access point avoids transmitting in a frequency band in order to minimize the interference caused in a terminal attached to other access point, previous information exchanging between both access points. These techniques are especially useful for terminals located at the edge of the cells, which suffers from poor coverage [20].

Multi-Cell techniques are one of the main features defined by the 3GPP for the LTE-A cellular networks with the aim of increasing coverage, spectral efficiency and transmission rate (throughput) at the cell edge [2]. For that purpose, the access points should be coordinated and synchronized through the exchange of

[1] In the downlink of LTE and LTE-A there is a little margin for operation to dynamically modify the transmission power [12].

specific information.

This kind of coordinated transmission techniques are specially useful in Heterogeneous Networks (HetNets), typically made up of Base Stations (BSs) with different power settings and/or different access point technologies [21]. HetNets are a very interesting way to increase the capacity of the mobile network through the combination of low power consumption nodes which serve small cells (femto or pico cells) with other nodes serving big cells (macro cells). Therefore, reusing frequency in different layers of the HetNets needs for coordination between the different nodes in order to achieve a good interference management.

To carry out link adaptation in those coordinated networks, techniques described in previous subsection should be configured taking into account theirs particularities in order to get the best possible performance.

1.2 Main contributions of this work

This work is focused on two main parts of the Link Adaptation (LA) process in cellular networks, whose principles were previously described: channel coding and adaptive modulation and coding. Next, targets of this works for each of these parts are described:

1. **Channel coding:**

   (a) To propose an adaptive SOVA (adSOVA) based turbo decoder [22], able to adapt the number of operations of the decoding process to transmission conditions. Then, when the data block to be decoded is received with a low coding rate, which eases the decoding process, the proposed adSOVA reduces the number of operations of the decoding process, which leads to a power consumption saving. On the other hand, when the data block to be decoded is received with a high coding rate, the adSOVA is able to increase the number of operations, if needed, to improve the error correction performance. Finally, the implementation of the proposed adSOVA does not imply a significant complexity increase.

   (b) To propose a model for the BLER of the SOVA based turbo decoder, based on binary logistic regression functions, which is not available in the literature. This model provides the insight to propose an improved LA process.

2. **Adaptive modulation and coding:** To carry out a deep study of the AMC process supported by the Outer Loop Link Adaptation (OLLA) technique [23–25], which is widely used to ensure that the adaptive modulation and coding process meets the target BLER for its specific working conditions. Furthermore, modifications to the OLLA technique have been proposed, which improve the performance of the AMC process. Specifically, the following goals have been addressed in this work:

   (a) To carry out a complete analysis of the OLLA technique, studying its convergence conditions an its performance under different scenarios. This study fills a gap existing in the literature about the OLLA technique.

   (b) To perform a constant power optimization of the AMC process of the LTE downlink, which is used to study whether the OLLA technique leads to the optimum performance. After that optimization, a multi-offset implementation of the OLLA technique is studied, which is proved to potentially achieve an optimum performance of the AMC process, according to previous constant power optimization results.

   (c) To propose a modification of the OLLA technique, named enhanced OLLA (eOLLA) [26], which is able to track the temporal channel variations more accurately than the OLLA technique, and so to improve the AMC performance, especially under certain conditions.

   (d) To propose a modification of the OLLA technique, the Closed Loop Link Adaptation (CLLA), which provides a better adjustment of the AMC process when variance in the size of the transmitted data blocks (named Transport Blocks in LTE) is high.

1.3 Outline of this work

This work is organized in 7 chapters and an appendix. A brief description of each chapter is described next:

- **Chapter 1:** An introduction where the main goals of this work are discussed.

- **Chapter 2:** This chapter focuses on turbo decoding. Turbo decoding algorithms are discussed, paying special attention to SOVA based turbo decoders. Then, an Adaptive SOVA (adSOVA) algorithm is proposed. Finally, a performance comparison between the proposed algorithm and traditional solutions are provided.

- **Chapter 3:** In this chapter a complete analysis of the OLLA technique is carried out, which is intended to fill a gap on this technique. A convergence study and a performance analysis are the main topics of this chapter.
- **Chapter 4:** In this chapter, a model of turbo decoder BLER metrics based on binary logistic regression functions is found. With these results, and results from Chapter 3, an enhanced OLLA (eOLLA) is proposed, and its performance is compared to that of the traditional OLLA.
- **Chapter 5:** In this chapter, constant power optimization of the AMC process is carried out. Then, optimum results are compared with multi-offset OLLA implementation performance.
- **Chapter 6:** In this chapter the effects of the Transport Blocks size in the performance of the turbo decoder is studied. After that, it is proposed a Closed Loop Link Adaptation (CLLA), which adapts its behavior according to the size of transmitted Transport Blocks.
- **Chapter 7:** In this chapter conclusions and future work are presented.
- **Appendix A:** Describes the LTE-A downlink simulator which is used in this work to obtain performance results of proposed techniques.

1.4 Related works

This thesis is supported by two articles:

1. F. Blanquez-Casado, F. J. Martin-Vega, D. Morales-Jimenez, G. Gomez, and J. T. Entrambasaguas, ”Adaptive SOVA for 3GPP-LTE Receivers”, *IEEE Communications Letters*, vol. 18, no. 6, pp. 991-994, June 2014.
2. F. Blanquez-Casado, G. Gomez, M. d. C. Aguayo-Torres, and J. T. Entrambasaguas, ”eOLLA: an enhanced Outer Loop Link Adaptation for Cellular Networks”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1-16, 2016. [Online].
   Available: http://dx.doi.org/10.1186/s13638-016-0518-3

In addition to these articles, the LTE-A simulator used to obtain the results that support this work has been published online:

- *http://hdl.handle.net/10630/11057*

Chapter 2

Adaptive SOVA for 3GPP-LTE receivers

Channel coding is a crucial part of current wireless and mobile communication systems. Through a good error correction method, transmission rates close to the channel capacity can be achieved [4]. Due to their high error correction capabilities, turbo codes have been chosen as the physical layer coding scheme in new cellular standards like Long Term Evolution (LTE) [1].

Classical turbo decoding is an iterative process in which two Soft Input Soft Output (SISO) decoders exchange the so called extrinsic information in form of Log Likelihood Ratio (LLR). Two well-known SISO algorithms are currently available in the literature: the Maximum A Posteriori (MAP) algorithm [6] and the Soft Output Viterbi Algorithm (SOVA) [7][4]. The MAP algorithm provides the best performance of the decoding process, but the highest complexity. In order to reduce complexity, sub-optimal forms of the MAP algorithm have been proposed, which still have better error correction performance than the SOVA. These sub-optimal forms are the Log-MAP and the Max-Log-MAP algorithms [27]. However, the SOVA decoder still is the one with the lowest complexity. For that reason, it is frequently used in practical implementations of mobile communications systems, when high processing rate is required [28].

Turbo decoding is the process with the highest complexity at an LTE receiver. Then, even when using a SOVA based decoder, decoding complexity might be too high for a real-time system.

In order to reduce SOVA complexity, a truncation of this algorithm [4] is usually performed. This truncation consists in an early stop of the SOVA process, since it is assumed that the error correction capability will not improve significantly from this point. This truncation can achieve a high complexity reduction, although it can be proved that it might cause a degradation in the error correction capability when high coding rates are used [29]. In standards like 3GPP-LTE, different Modulation and Coding Schemes (MCSs) are used due to the Link Adaptation (LA) process [12]. As it will be shown, truncating the SOVA algorithm can bring important BLock Error Rate (BLER) degradation, especially when using less robust MCSs.

In this chapter it is proposed an adaptive SOVA (adSOVA) turbo decoder for 3GPP-LTE receivers [22] which adaptively selects the minimum truncation length that reaches the error correction performance of a SOVA with no truncation. Thus, this proposed adSOVA can reduce complexity, which is defined in this work as CPU cycle consumption, while achieving the same BLER than the conventional SOVA.

2.1 Channel coding fundamentals

The aim of channel coding techniques is the detection and correction of errors caused by communications channel during a transmission. A well-known outcome from Information Theory is that theoretically there exists a coding scheme good enough to reach the channel capacity [30]. Channel coding techniques are performed by intelligently adding redundancy to the message to be transmitted, and using this redundancy to adequately recover the message during the reception process. Depending on the way this redundancy is added to the message to be transmitted, two basic coding schemes can be differentiated:

- Block coding: The coder transforms the k bits of the message into a code word of n bits. This transformation only depends on which the k bits of the original message are. A generating matrix is commonly used, thus no extra memory is necessary in this process.

- Convolutional coding: The transformation of the k bits of the original message into the n bits of the coded message does not only depend on these k bits but also on the m previous bits. Thus, it is necessary the use of an m bit memory. This coding scheme is based on generator polynomials.

Convolutional coding achieves better performance as the memory size increases. However, this growth can not be carried out indefinitely as it leads to an exponential increase of the decoding process complexity. For that reason, it has been searched for coding systems which allow to achieve the performance of large memory convolutional scheme, without effectively using this amount of memory, that is, to emulate increasing the used memory. One of these coding systems are turbo codes [4]. These coding schemes are based on concatenating more simple coding schemes so that they achieve the performance of a more complex equivalent code. In particular, a turbo coder scheme consists in the concatenation of convolutional coders separated by pseudo-random interleavers, so each coder works with a different interleaved version of the original message. Then, each convolutional coder carries out a parallel coding process, and all the resulting codes are combined to conform the coded message.

2.2 The turbo coder

The generic structure of a turbo decoder is shown in Fig. 2.1. The PADDING block appends $n - k$ tail bits to the k bits to be coded. This n bit sequence is entered into the different branches of the turbo coder, which consists in an interleaver α_k and a convolutional coder. These convolutional coders are defined by a generator polynomial, which also indicates the constraint length (or memory depth) of the turbo code, that is, the number of memory registers plus one. It is also possible the existence of a branch which has neither an interleaver nor a convolutional coder, so the output of this branch is the n input bits. Bits obtained at the output of this kind of branches are named systematic bits, while bits obtained from the rest of branches are named parity bits. Regarding the tail bits, they are used to reset the turbo coder register to zero once all the n bits are transmitted.

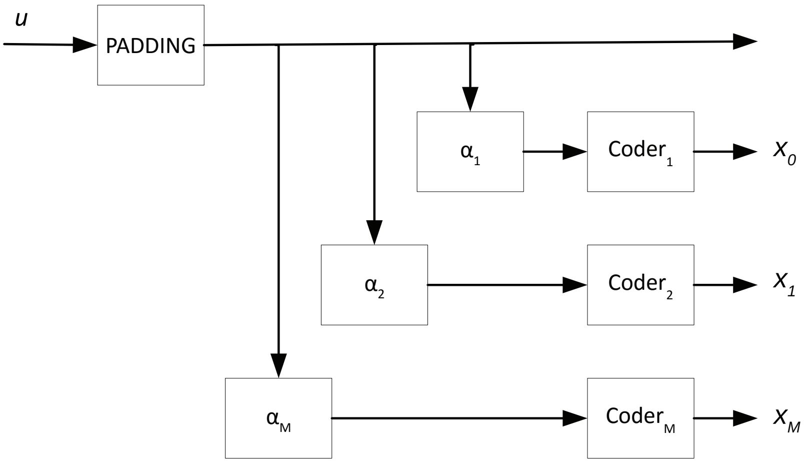

Figure 2.1: Generic turbo coder structure.

Turbo coders can be described as finite state machines whose states are defined by the content of its physical registers. Then, an input bit to be coded will produce different outputs depending of the content of these registers. Therefore, turbo coder behavior can be modeled by means of state diagrams.

The structure of the 3GPP-LTE turbo coder [31] is shown in Fig. 2.2. As it was mentioned before, a turbo coder is made up of the concatenation of convolutional coders. Specifically, the 3GPP-LTE turbo coder is based on the concatenation of two convolutional coders. Each convolutional coder is a recursively systematic coder of length 3, that is, it entirely transmits the parity bits (x_k and x'_k) as well as the input bits (z_k and z'_k). Therefore, as each coder consists of 3 registers, the total number of states for this coder is $2^3 = 8$.

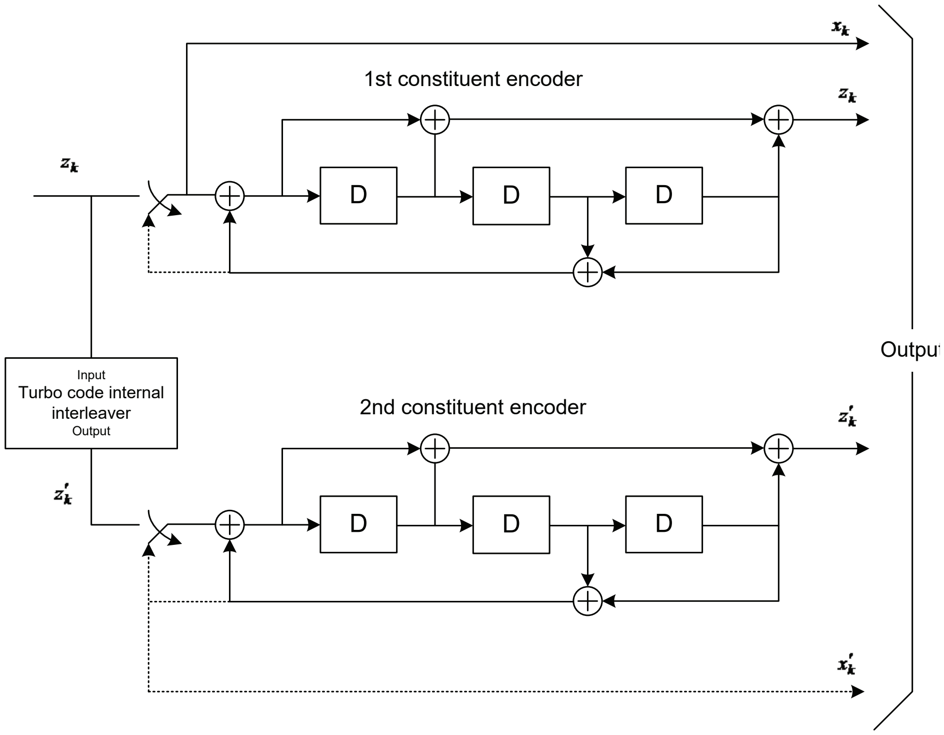

Figure 2.2: 3GPP-LTE turbo coder structure.

The concatenation of these two convolutional coders is carried out in a way that one systematic stream (z_k) and two parity streams are transmitted, being the coding rate of 1/3. The first parity stream comes from applying the convolutional coding to the original input bit stream, whereas the second parity stream is got from applying the same convolutional coding to the input bit stream after a pseudo-random interleaving. This interleaver, also known as internal interleaver, is only implemented for certain sizes of data blocks, being the maximum size 6144 bits. The turbo coder appends 12 tail bits to the coded sequence. These tail bits are obtained from switching the data input to the first recursive input, and getting four bits at the output of each branch.

Turbo coders can be characterized by their transference function, a mathematical model that describes the input and the output of a black box model. For a turbo coder, its transference model is the quotient between the input (the original data sequence to be transmitted) and the output (the coded sequence). Next, it is shown the transference function $G(D)$ of the 3GPP-LTE turbo coder [31]:

$$G(D) = \left[1, \frac{g_1(D)}{g_0(D)}\right], \tag{2.1}$$

where:

$$g_0(D) = 1 + D^2 + D^3 \tag{2.2}$$

and

$$g_1(D) = 1 + D + D^3 \tag{2.3}$$

being D the content of the shift registers (see Fig. 2.2) of the turbo coder, whose initial value is set to 0. It should be noticed that the transference function has two terms. The first one is 1, which corresponds to the systematic output, since the output is directly the input. The second one is the quotient between the generator polynomial that represents the feedback of the convolutional coder, $g_1(D)$, and the generator polynomial that represents its ouput, $g_0(D)$.

In general, the incoming data should be pre-processed in order to adequate its structure to the necessities of the turbo coder, since only certain predetermined data block sizes are allowed by the interleaver. In case the size of the data block is lower than the maximum allowable size, but is not among those predefined sizes, it is necessary to add random bits (also named filler bits [31]) at the beginning of the data block until an allowable size is achieved. Thus, these filler bits are added to the systematic as well as the parity streams. To minimize the redundancy transmission, these filler bits will be discarded from the systematic and first parity stream after the coding process, since the position in which they were appended is known. However, they can not be discarded for the second parity stream due to the interleaving process. These filler bits have to be added again in the receiver before the decoding process. Finally, in case the size of the data block to be coded is higher than the maximum allowed size of the interleaver, a segmentation process [31] is carried out, which divides the data block into smaller segments [1].

2.3 The turbo decoder

Turbo decoding strategy is based on the combination of simple codes so that each of them can be independently decoded by a low complexity SISO decoder. A SISO decoder is the one able to manage soft values, that is, to accept at its input coded bits weighted by the probability of being correctly received; and to provide at its output decoded bits weighted by the probability that the decoding process has been carried out correctly. Thus, by using SISO decoders, the information can be iteratively exchanged between decoders. In this case, the performance can be close to a Maximum Likelihood Decoding Algorithm (MLDA), that is, to a convolutional scheme with infinity memory, but with a reasonable complexity.

In Fig. 2.3 it is shown the generic scheme of a turbo decoder for a coding rate of 1/3. This turbo decoder is made up of two SISO decoders that iteratively exchange information. The exchanged information passes through an interleaver and a de-interleaver in order to compensate the effect of the turbo coder interleaver. As it can be seen, each turbo decoder has three inputs, one for the flux of systematic bits, another for one of the two fluxes of parity bits, and the last one for the information about the possible decoded bits from the other SISO decoder. This information coming from the other decoder is called the *a priori* information.

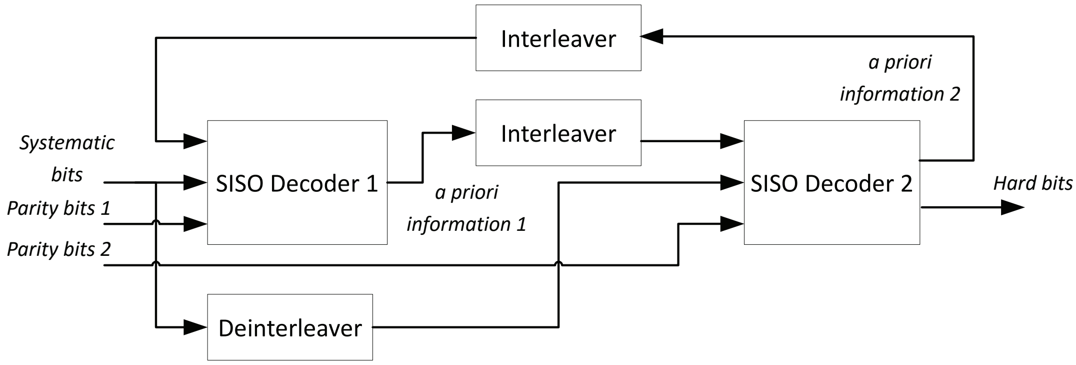

Figure 2.3: Generic turbo decoder structure

Both SISO decoders have to use the *a priori* information as well as the information coming from the received bits (both parity and systematic streams).

[1] Filler bits insertion could be previously required in the first segment to ensure that all segment sizes are valid values for the interleaver.

These decoders have also to be able to provide soft outputs for the decoded bits, which means that they must provide not only the decoded sequence but also the probability that this sequence has been correctly decoded. These soft outputs are usually represented as Log-Likelihood Ratio (LLR), so magnitudes and amplitudes of these soft outputs provide information about the sign of each bit and the probability of correct decoding, respectively. The LLR for a decoded bit u_k is given by:

$$L\left(u_k\right) = \ln\left(\frac{P\left(u_k = +1\right)}{P\left(u_k = -1\right)}\right) \tag{2.4}$$

where $P\left(u_k = +1\right)$ is the probability of the decoded bit is +1 (binary 1), and $P\left(u_k = -1\right)$ is the probability that the decoded bit is -1 (binary 0). Thus, a high positive LLR means that the decoded bit has a high probability of being +1, while a high negative LLR means that the decoded bit has a high probability of being -1.

Based on the foregoing, it follows that turbo decoder inputs coming from the channel should also be soft. That is, the receiver should be able to take soft decisions about each received bit, so the turbo decoder can estimate the LLR of each bit at its input. This soft decisions can be implemented, for instance, by estimating the SNR of each received bit.

There are two main categories of SISO decoding algorithms: algorithms that minimize the error probability of each transmitted symbol, and algorithms that minimize the error probability of the whole transmitted sequence. The first category comprises the Maximum a Posteriory (MAP) algorithms, while the second category comprises the Soft Output Viterbi Algorithms (SOVA). Next, they are described the main characteristics of each category of SISO algorithms:

- **MAP:** Also known as BCJR [6] because of the initials of its inventors (Bahl, Cocke, Jelinek and Raviv). This algorithm can be used to decode block codes as well as convolutional codes, being optimal in terms of minimizing Bit Error Rate (BER) for convolutional codes. This algorithm is different to the Viterbi algorithm [32], which minimizes the probability of the decoder selecting the incorrect sequence. The MAP algorithm is optimal in error correction. However, its main drawback is that it has to evaluate all possible sequences, which implies a high complexity. Modifications of the MAP algorithm has been carried out in order to reduce its complexity. The two most important modifications are the Log-MAP and Max-Log-MAP [27]. Both modifications are based on the fact that the MAP algorithm has to perform a very high number of multiplications and additions to get the soft output for each decoded bit. Thus, these modifications works in the logarithmic domain, transforming multiplications in adds, which reduces complexity dramatically.

- **SOVA:** The Viterbi algorithm [7][4] is able to accept soft inputs but it can not generate soft output, making it not suitable for turbo decoding. The SOVA modifies the Viterbi algorithm to take into account the information coming from the other SISO decoder (the *a priori* information) as well as to generate the correct decoding probability for each output bit. The complexity of the SOVA is lower than the complexity of the MAP algorithm whatever the modification previously described. As it was mentioned before, the SOVA algorithm belongs to the SISO decoders algorithms that minimizes the error probability of the whole sequence to be decoded.

To sum up, the MAP algorithm and its variations provides better performance than the SOVA (performance of Max-Log-MAP is 0.6dB higher than the SOVA performance [33]), but also it provides the maximum complexity (complexity of the Max-Log-MAP algorithm doubles the SOVA complexity).

2.3.1 SOVA based turbo decoder

A SOVA based turbo decoder is made up of two SOVA decoders which exchanged information through an interleaver and a de-interleaver. The SOVA [7][4] builds a Trellis tree [32] of the received sequence $\mathbf{y}$. A Trellis tree is a diagram made up of a matrix of nodes. Each node within a column represents each possible state of the turbo coder, and each column represents all the possible turbo coder states at a certain time. As it was previously explained, the state of the turbo decoder is associated to the content of its physical registers. In this case the number of columns corresponds to the number of systematic bits received plus the number of tail bits. At each instant t a new systematic bit is processed at the decoder, a new column is appended to the Trellis tree. Transmitting a systematic bit $x_t = 0$ or $x_t = 1$ causes different transitions from an encoder state s'_0 or s'_1, respectively, to a state s. Each transition is associated with a parity bit z_t and an accumulated metric $M_t(s', s)$, which is obtained as

$$M_t(s', s) = M_{t-1}(s', s) + x_t \cdot L_c \cdot y_{t,1} + z_t \cdot L_c \cdot y_{t,2} + x_t \cdot L_t \qquad (2.5)$$

being:

- L_c : the reliability of the received bit at the input of the turbo decoder.

- $y_{t,1}$: the received systematic bit.

- $y_{t,2}$: the received parity bit.

- L_t : the *a priori* reliability obtained through other SOVA decoder that integrates the turbo decoder.

During the decoding process, for each coded bit received, the transition with the highest accumulated metric $M_t(s', s)$ is selected in the Trellis tree. The systematic bit associated to this transition is selected as decoded bit u_t, and the branch that connects the current state s to the previous state s' in the Trellis tree becomes part of the *survivor* path, while the branch associated to the opposite of the decoded bit becomes part of the *competing* path. The rest of branches of these paths are obtained by linking earlier branches with the highest accumulated metrics. Then, the *survivor* path represents the most likely received sequence, while the *competing* represents the second most likely received sequence. For the branch with the highest accumulated metric, a *reliability* of the last decoded bit of the *survivor path* is defined as

$$\Delta_t^0 = \frac{1}{2} \left| M_t\left(s'_0, s\right) - M_t\left(s'_1, s\right) \right| \tag{2.6}$$

being the sub-index the instant of time the bit has been received, and the super-index the memory level, which is updated every time a new bit is received. The highest values of this memory level are associated with the oldest stored information. This reliability and its associated bit of the *survivor* path are stored. The reliability is in fact an estimation of the LLR of the decoded bit. In case that in a certain node the accumulated metric values for its two branches (one corresponding to a transmitted bit 0 and the other a transmitted bit 1) are similar, the reliability value will be low, since the probability of having transmitted one bit or another is similar. On the other hand, a high reliability will mean a high probability that the bit with the highest accumulated metric is correct.

Then, an *updating* process is done as follows. Branches of the *survivor* and *competing* paths are traced back and, when the binary decision is different in both paths at the memory level MEM, the reliability is updated as

$$\Delta_t^{MEM} = \min_{k=0,\ldots,MEM} \left\{ \Delta_t^k \right\}. \tag{2.7}$$

Finally, when the updating process is done for all entry bits, soft output values are obtained as

$$L\left(\mathbf{u_t}\right) = \boldsymbol{\Delta}_\mathbf{t} \cdot \mathbf{u_t} \tag{2.8}$$

However, this information is not what it is exchanged between the two SOVA decoders that make up the turbo decoder. Instead, the *extrinsic information* $L_e\left(u_t\right)$ is used. Since the reliability values are calculated by using the channel reliability as well as the *a priori* reliability from the other SOVA decoder, this information would be sent back again to it if the values from (2.8) are directly used, so it would be redundant. Thus, *extrinsic information* is extracted from the soft decoded values as:

$$L_e\left(u_t\right) = L\left(u_t\right) - L_t - L_c \cdot y_{t,1} \tag{2.9}$$

and sent to the other SOVA decoder that conforms the turbo decoder after a compression. Typically a logarithmic compression is used. The aim of this compression is to avoid that, in case that high reliability values are assigned to a bit that has been erroneously decoded, this error propagates to the other SOVA decoder. Please, note that this compression implies an increase of complexity due to it would be necessary to perform as many logarithmic operations as bits to be decoded at each SOVA iteration. Furthermore, it may cause a slower convergence when the bits are correctly decoded.

Updating process

In this subsection it is explained the importance of the updating process previously described. For this purpose, an example of a Trellis tree is shown in Fig. 2.4. In this example, at the instant t there is a node whose accumulated metric for the case of receiving bit 0 is the same than for the case of receiving bit 1, and in addition to that, this accumulated metric value is the maximum of the nodes corresponding to this instant t. In that case, one arbitrary bit value is selected to be part of the *survivor* path, and the reliability of this bit is 0. Next, the trace back process is carried out to get the *survivor* and the *competing* path. For the memory level 1 the bits that make up both paths have the same values, so the reliability value previously calculated for this level can be applied, since there is no other possible value for this bit. However, when the trace back process arrives to the memory level 2, there is a path divergence, and a reliability value of 30. In this case, it can not be claimed that the bit that is part of the *survivor* path is reliable, since at the memory level 0 the accumulated metric for both paths is the same, which means that both bit sequences that made up each path have the same probability of having been transmitted. Therefore, it is used the reliability of the memory level 0 to the decoded bit of the memory level 2, which is 0 in this case.

To sum up, the aim of the updating process is to avoid assigning a high value of reliability to a decoded bit when the path to which it belongs has a low probability of having been transmitted.

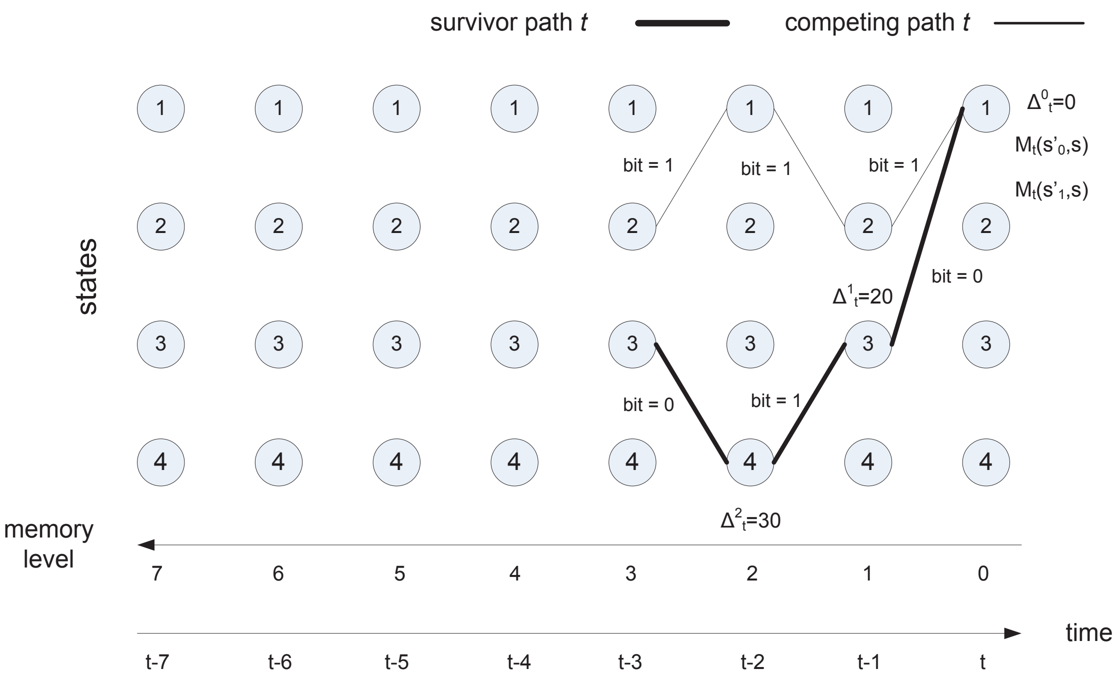

Figure 2.4: Example of the updating process

2.3.2 The Max-Log-MAP algorithm

The MAP algorithm [6] estimates the a-posteriori probability of each decoded bit u_t from the received sequence $\mathbf{y}$ in LLR form:

$$L\left(u_t\,|\mathbf{y}\right) = \log\left(\frac{P\left(u_t = +1\,|\mathbf{y}\right)}{P\left(u_t = -1\,|\mathbf{y}\right)}\right). \tag{2.10}$$

To compute this a-posteriori probability, forward $\left(A_t\left(s\right)\right)$ and backward $\left(B_{t-1}\left(s'\right)\right)$ metrics are defined for each encoder state s in a recursive manner:

$$A_t\left(s\right) = \mathop{\overset{*}{\max}}_{s'}\left[A_{t-1}\left(s'\right) + M_t\left(s', s\right)\right] \tag{2.11}$$

$$B_t\left(s\right) = \overset{*}{\underset{s}{\max}}\left[B_{t-1}\left(s\right) + M_t\left(s', s\right)\right] \tag{2.12}$$

being s' in (2.11) the states connected to s in a forward direction and s in (2.12) the states connected to s' in a backward direction. M_t is computed as in (8.1). The max* operator is the Jacobian logarithm, which is approximated as $\max(a, b)$ [27] for the Max-Log-MAP algorithm. Finally, after computing the forward and backward metric, the a-posteriori probabilities of decoded bits are calculated as

$$L\left(u_t\left|\mathbf{y}\right.\right) = \overset{*}{\underset{R_0}{\max}}\left(A_{t-1}\left(s'\right) + M_t\left(s', s\right) + B_t\left(s\right)\right) - \overset{*}{\underset{R_1}{\max}}\left(A_{t-1}\left(s'\right) + M_t\left(s', s\right) + B_t\left(s\right)\right) \tag{2.13}$$

where R_0 and R_1 are the possible transitions from states s' to states s when a bit 0 or a bit 1 is transmitted, respectively.

2.4 Proposed Adaptive SOVA

One of the main drawback of turbo codes is the high complexity of the decoding process. Regarding the SOVA based decoder, one of the most complex parts is the updating process. This process is carried out every time a new systematic bit is received. Thus, when the k^{th} systematic bit is received, $k-1$ nodes are traced back for both the *survivor* and the *competing* paths. Then, for a received block of n systematic bits, the amount of steps that comprises the updating process for each path is

$$\sum_{i=1}^{n}\left(n-1\right) = \frac{n\left(n-1\right)}{2}. \tag{2.14}$$

That is, a total of $n\left(n-1\right)$ steps are carried out at each SOVA iteration. Thus, the complexity of the updating process can be approximated as quadratic with the size of the data block to be decoded.

In order to reduce complexity, a limitation on the number of trace back steps of the updating process is carried out in the SOVA. Typically, a value of $2K$ or $3K$ (where K is the constraint length of the turbo encoder) is recommended [4]. However, this limitation may degrade the performance of the decoding process depending on the scenario. For instance, [29] evaluates the need of a deeper updating window for high coding rates to avoid BER degradation. Thus, a short updating window could lead to a degradation of the error correction performance for the highest coding rates, while a deep updating window might unnecessarily increase the complexity of the decoding process for the most robust coding rates. Moreover, the channel has also a significant impact on the SOVA performance [34]. Hence, the a priori selection of a value for the depth of the updating process could reduce the efficiency of the turbo decoder.

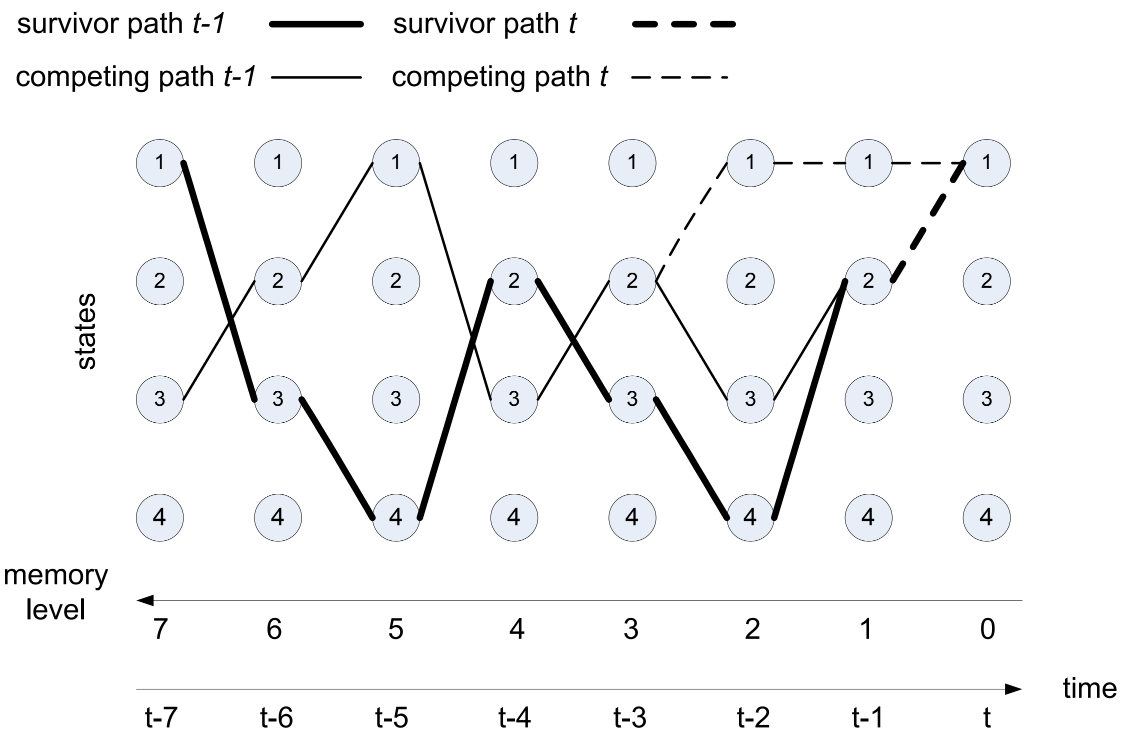

Figure 2.5: Trellis tree example

In this work an adaptive SOVA (adSOVA) is proposed, which stops the updating process as soon as no further trace back steps are needed. Hence, the proposed algorithm achieves the same error correction performance as the SOVA while reducing complexity by avoiding unnecessary processing. To illustrate this, Fig. 2.5 shows an example of the *competing* and the *survivor* paths at the instant $t-1$ and t for a four-states turbo code. At memory level 1, the *survivor* path at t matches the *survivor* path at $t-1$. The same coincidence happens with the *competing* path at memory level 3. Thus, if any of the *competing* or the *survivor* paths reaches a node at t which had been reached at $t-1$, the rest of the trace back steps coincide.

Moreover, if the reliability value obtained for the memory level 0 at the instant t (Δ_t^0) is the lowest value at this time, this value will be used during the *updating* process in the rest of memory levels when binary decisions are different. Otherwise, the reliability value used at each memory level will be the minimum value found until this level, and it will match the minimum value found until this level at $t-1$. Thus, if both the *survivor* and the *competing* paths coincide at the instant t and $t-1$, it is not necessary to complete the *updating* process from these nodes. Based on this fact, an *adaptive updating* algorithm is proposed for the adSOVA.

To define the *adaptive updating* process at time t, the proposal considers the nodes of the *survivor* and *competing* path of the instant $t-1$ as well as the minimum reliability value found during the trace back process. The proposed algorithm is given by the following steps (see Fig. 2.6):

1. Calculate the reliability value for the memory level $k=0$. This value is taken as the minimum reliability value ($\Delta_{min} = \Delta_t^0$). Then the trace back process through the survivor and competing path begins.

2. Check if the reliability value stored for each memory level is the minimum value found. In that case the minimum reliability value is updated ($\Delta_{min} = \Delta_t^k$).

3. Check if both the *survivor* and *competing* paths for each node at instant t belong to the *survivor* and the *competing* path at the instant $t-1$.

4. If step 3 is satisfied and reliability value of the memory level $k=0$ is not the minimum reliability value, the *updating* process can be finished since no new updating in the rest of the trace back process will be done. Otherwise, go back to step 2 until the trace back process is terminated, or stopping conditions are satisfied.

The adSOVA is expected to reduce on average the number of trace back steps compared to the SOVA. Regarding the number of operations, three extra comparisons are done at each trace back step of the updating process.

2.5 Performance of adSOVA in 3GPP-LTE systems

In this section the performance of the proposed adSOVA is evaluated and compared with the SOVA [4] and the Max-Log-MAP algorithm with sliding-window [35]. These algorithms have been included in a complete LTE downlink simulator [36] described in section A, which includes a 3GPP-LTE turbo encoder [31]. Main simulation parameters are listed in Table 4.3.

In 3GPP-LTE standard, BLER is a more useful statistic than BER as it is used to perform the LA [12]. In Fig. 2.7 BLER results are shown for MCSs corresponding to the Channel Quality Indicators (CQIs) 4, 9 and 14 [12], for

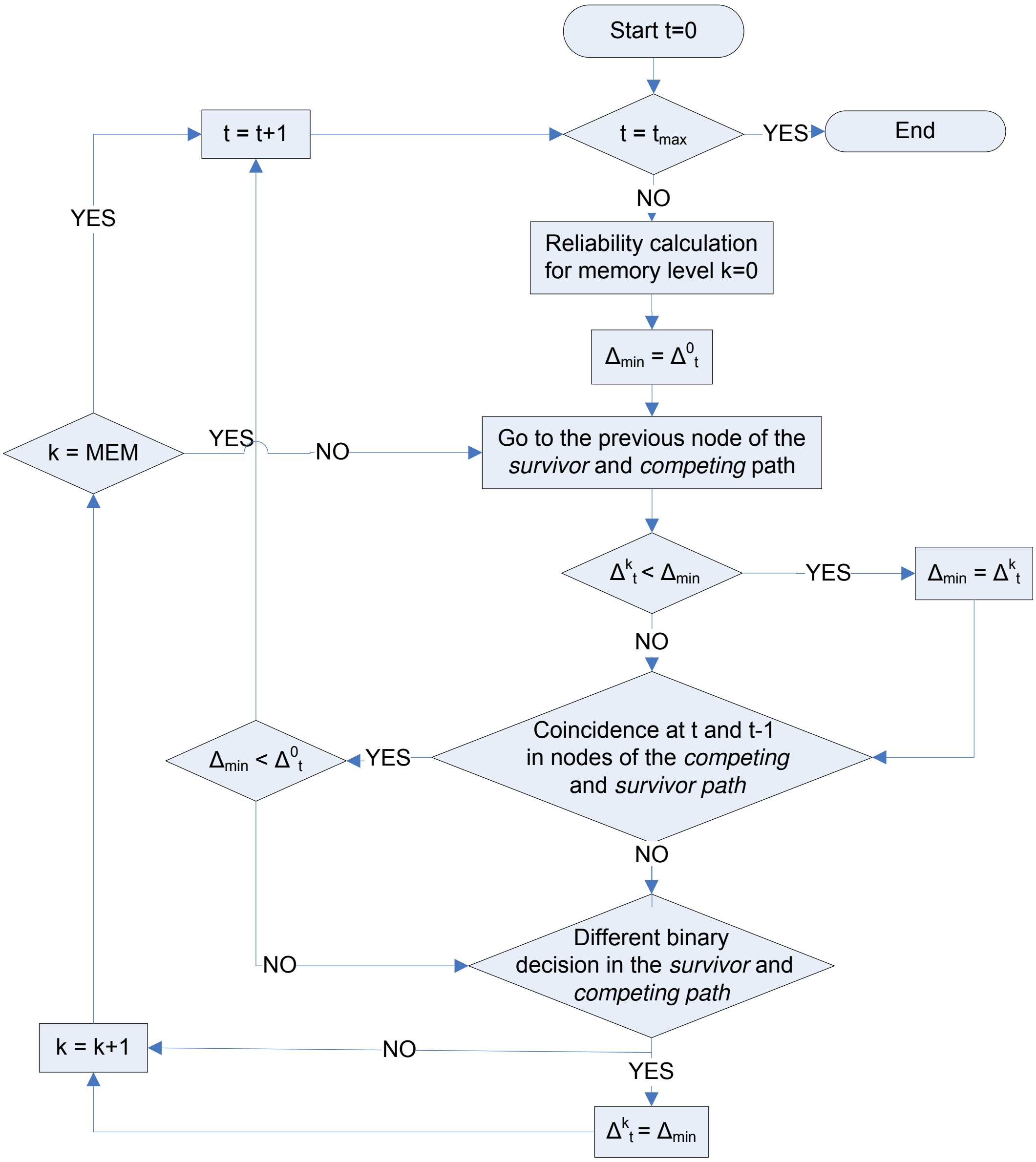
Start t=0
t = t_{max}
YES
End
NO
Reliability calculation for memory level k=0
$\Delta_{min} = \Delta^0_t$
Go to the previous node of the *survivor* and *competing* path
$\Delta^k_t < \Delta_{min}$
YES
$\Delta_{min} = \Delta^k_t$
NO
Coincidence at t and t-1 in nodes of the *competing* and *survivor path*
YES
$\Delta_{min} < \Delta^0_t$
NO
Different binary decision in the *survivor* and *competing path*
NO
YES
$\Delta^k_t = \Delta_{min}$
k = k+1
k = MEM
YES
NO
YES
t = t+1

Figure 2.6: Flow diagram of the proposed adSOVA

a SOVA decoder with different updating window depths w, and for the Max-Log-MAP decoder using a sliding-window [2] w of depth 20. Note that, as it was justified in Section 2.4, the error correction performance is identical for both the SOVA and the adSOVA for the same value of w. Regarding the SOVA, for the most robust MCS there are no major differences when the depth of the updating window increases. As the robustness of the MCS decreases, the use of a deeper updating window improves significantly the BLER performance. In particular, for a BLER of 0.1, a gain of 1.6dB is achieved when using a depth of $w = 80$ steps instead of $w = 20$ steps. However, Fig. 2.7 shows that an updating window deeper

[2] For the sake of simplicity the same letter w will be used to denote both types of windows

than 60 hardly improves the BLER performance whatever the MCS. Concerning the Max-Log-MAP algorithm, it always improves the SOVA except for the least robust MCS, where results are very similar when a deep updating window is used for the SOVA, which means that a deeper sliding-window is necessary. Hence, these results reveal the impact of the depth of the updating window in the BLER performance of a SOVA based turbo decoder.

Table 2.1: Simulation Parameters

Parameter	Value
Carrier frequency	2 GHz
Sampling frequency	30.72 MHz
System bandwidth	20 MHz
Channel model	Extended Pedestrian A [37]
Mobile terminal speed	4 km/h
Source model	Full buffer
Uncoded block length	5000 bits
CQIs	4, 9, 14
Turbo decoding iterations	5

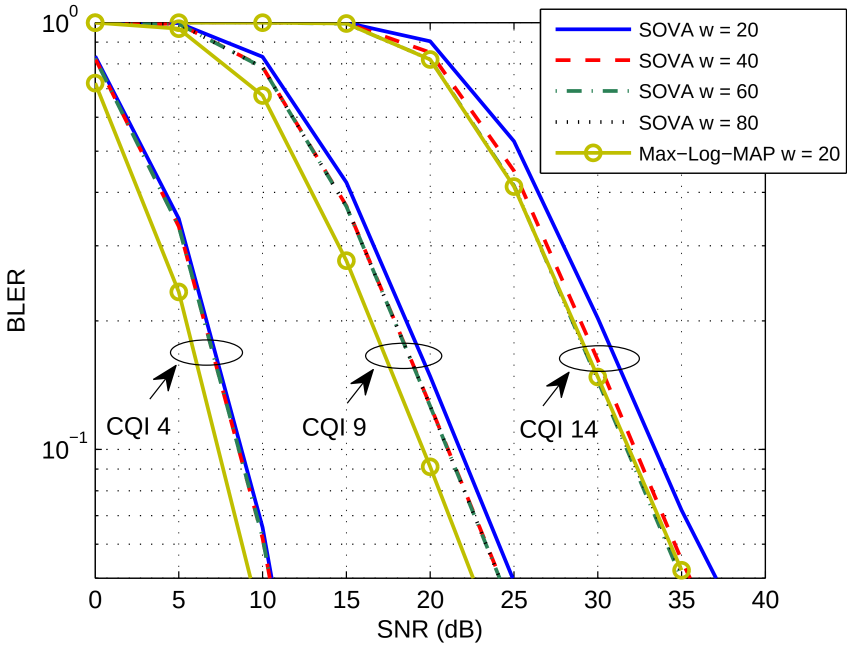

Figure 2.7: BLER results for different coding rates

Regarding complexity, Fig. 2.8 presents an average CPU cycle consumption comparison between the adSOVA and the SOVA, for different updating window

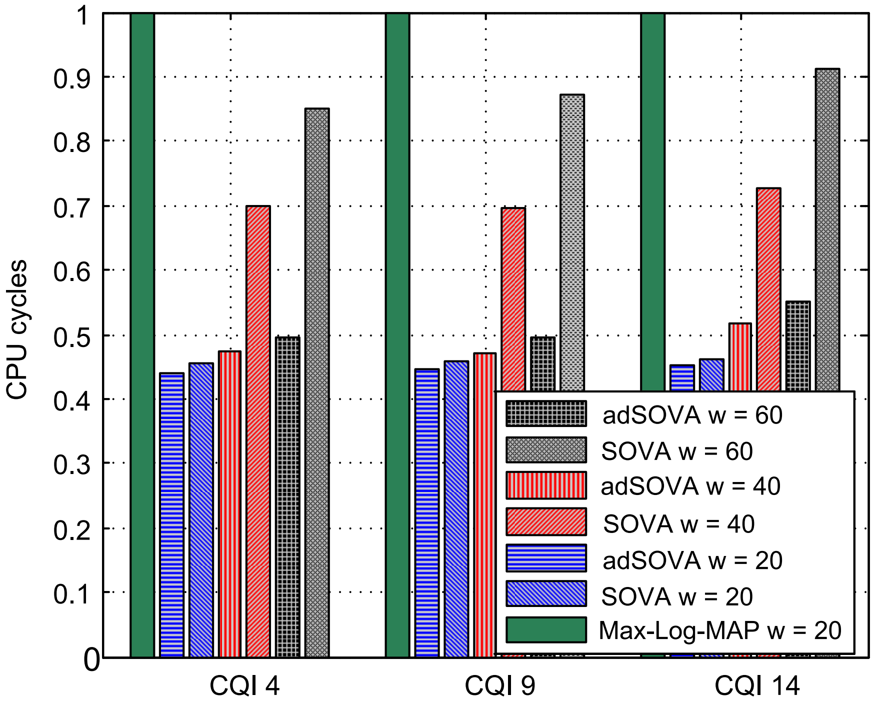

Figure 2.8: Average CPU cycle consumption for different MCSs

depths, and the Max-Log-MAP algorithm. For each MCS and each algorithm, it has been considered the average SNR for which a BLER of 0.1 is achieved (see Fig. 2.7), as this is the operating point of LTE [12]. These results have been normalized to the maximum value for each MCS, which corresponds to the Max-Log-MAP algorithm. Fig. 2.8 shows that, for the adSOVA, the number of CPU cycles consumed is always lower than those for the SOVA, for the same value of w. For instance, for the MCS corresponding to the CQI 14 and $w = 60$, a reduction of 40% is achieved. Furthermore, for the adSOVA, results when using a deep window are close to results of the SOVA when using the shortest window, i.e. $w = 20$. Thus, the proposed algorithm allows to use a higher value of w in order to improve BLER (especially for the least robust MCSs) with a complexity increase that is considerably lower than the experienced by the SOVA when the same value of w is used.

Fig. 2.9 shows the simulated Probability Mass Function[3] (PMF) of the number of trace back steps for the adSOVA with $w = 60$, for the same MCSs and mean SNR values than used in Fig. 2.8. Also, the mean values of these numbers of steps are shown. These results show that despite the variability of the number of steps, results are almost always close to the mean value, which in all cases is far

[3] Please, notice that as the number of trace back steps is an integer value, Probability Mas Function has to be used instead of Probability Density Function. Thus, only integer values has to be taken into account in Fig. 2.9.

from the maximum value of 60. Then, the complexity reduction of the adSOVA will not have high variability for each decoded block. Thus, although the turbo decoder must be worst-case designed, i.e. to assume that the number of trace back steps of the adSOVA is always the maximum value w, as this case will hardly ever occur, a complexity reduction will be obtained in almost all decoded blocks. This reduction can be used, for instance, to increase the number of iterations of the turbo decoder until achieving the worst-case complexity, thus improving the error correction performance; or to stop earlier the turbo decoding process in order to save power [38].

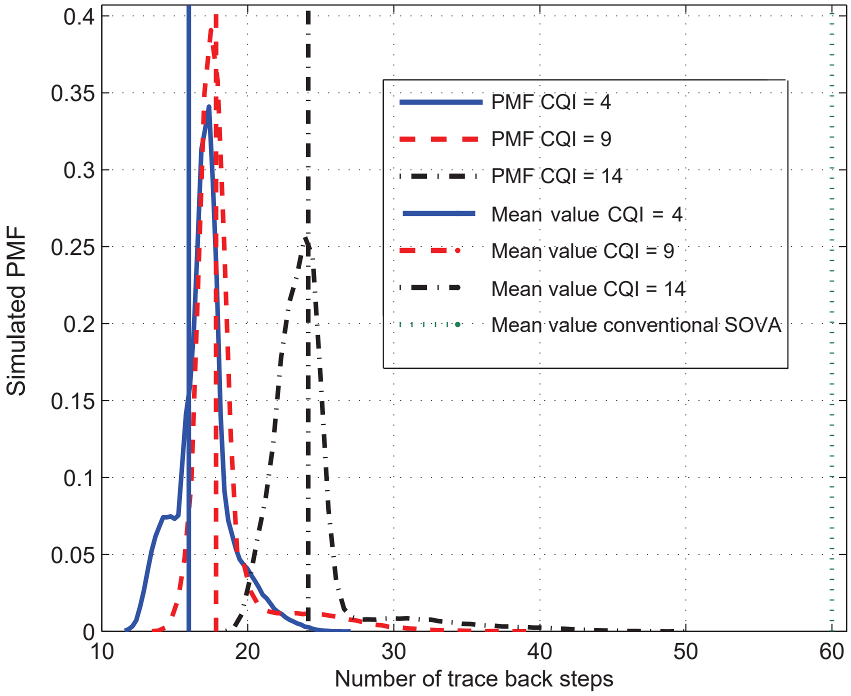

Figure 2.9: Simulated PMF of trace back steps for adSOVA with $w = 60$

Table 2.2: Number of operations

	Adds	Mults	Comparisons
SOVA	$49l$	$48l$	$15 + 2wl - w(w-1)$
adSOVA	$49l$	$48l$	$15 + 5wl - w(w-1)/2$
Max-Log-MAP	$128l$	$112l$	$40l$

In Table 2.2 a comparison of the number of operations for the three implemented algorithms for a 3GPP-LTE receiver is shown, where l is the length of the block to be decoded, and w is the depth of the updating window. The only difference between the SOVA and the adSOVA is the number of comparisons, which depends on the number of steps of the updating process. Note that while for the SOVA this number is fixed, for the proposed algorithm this number will be variable. Thus, the presented value is the one for the maximum number of steps allowed, i.e., the size of the window w. In that case, the proposed algorithm carries out $3wl - w(w-1)/2$ extra comparisons. Regarding the Max-Log-MAP algorithm, its complexity is always higher than both SOVA algorithms in terms of additions and multiplications, while the number of comparisons depends on l and w.

Chapter 3

Traditional Outer Loop Link Adaptation

The Adaptive Modulation and Coding (AMC) process carried out in the Link Adaptation (LA) is a crucial part of current wireless communication systems. This technique allows to increase the data rate that can be reliably transmitted [14] and has been adopted as a core feature in cellular standards such as Long Term Evolution (LTE)[39].

In the AMC process for the downlink of LTE [39], the User Equipment (UE) has to suggest to the Base Station (BS) an appropriate Modulation and Coding Scheme (MCS) to be used in the next transmission in order to keep the BLock Error Rate (BLER) below a target. The proposed MCS is signaled from the UE by means of a Channel Quality Indicator (CQI). Typically, each CQI is associated with a particular Signal to Noise Ratio (SNR) interval, hence MCSs are selected by mapping the estimated instantaneous SNR into its corresponding SNR interval, defined by an upper and a lower threshold.

A static selection of the values for the AMC thresholds does not perform well in practical implementations as link conditions are inherently variant. It is usual to adjust these thresholds by means of the well-known Outer Loop Link Adaptation (OLLA) technique, which was first proposed in [23]. Basically, OLLA modifies the SNR thresholds by an offset [24][25] which can be positive (making the MCS selection more robust) or negative (when the CQI selection was too strict). This offset is continuously updated based on the reliability of the received data blocks so that the average BLER is kept as close as possible to a predefined target.

Works devoted to OLLA typically address its performance from simulations and the lack of a comprehensive analysis of its behavior in the literature is noticeable. Furthermore, at the best of our knowledge, previous works do not analyze the conditions under which OLLA technique works properly. Thus, the aim of this chapter is to cover this gap by carrying out a deep study of the OLLA technique.

3.1 Modeling of Adaptive Modulation and Coding in LTE

A detailed description of the AMC system model is provided in this section (see Fig. 3.1). To perform the AMC process for the downlink of LTE, the instantaneous SNR γ is estimated at the UE for a certain number m of physical carriers[1]. This set of m estimated instantaneous SNR values can be expressed as a vector $\vec{\gamma}$. Typically, instantaneous SNR is measured over Reference Signals (RSs) [39] transmitted by the BS, and some kind of interpolation is carried out to estimate the SNR in the data carriers. Once the SNR has been estimated for the set of m physical carrier, it is calculated a Link Quality Metric (LQM) $L\left(\vec{\gamma}\right)$ value. The LQM takes into account the number of physical carriers evaluated and the SNR estimated for thess carriers to quantify the quality of the link. There are several LQMs such as raw or uncoded bit error rate mapping (RawBER) [40], exponential effective SNR mapping (EESM) [41], mutual information per coded bit mapping (MMIB) [42], packet error rate (PER) indicator method [43] or the averaged SNR. The number m of physical carriers used to calculate an LQM $L\left(\vec{\gamma}\right)$ value will typically correspond to a whole number of Physical Resource Blocks (PRBs) [39], being a PRB the minimum amount of tractable physical resources.

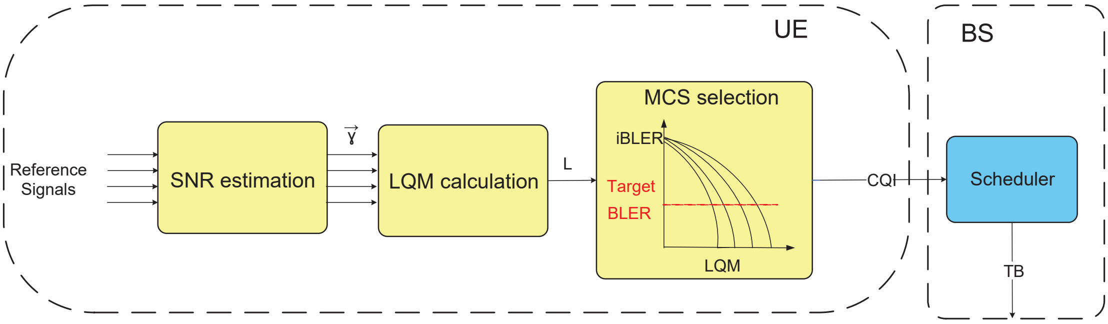

Figure 3.1: Scheme of the LA process

The calculated LQM $L\left(\vec{\gamma}\right)$ value is used to determine a Channel Quality Indicador (CQI) value that will be fed back to the BS to suggest an MCS i to be

[1] Physical carriers in LTE comprises data carriers and reference signals.

used in the next transmission in order to not exceed certain target BLER [39]. Each CQI index has associated a transmission rate R_i (bits/symbols). Thus, in order to not degrade the transmission rate, the CQI value reported has to be the highest possible that satisfies the BLER constraint.

The mapping process of an LQM value into a CQI value is carried out by calculating the instantaneous BLER for each MCS i, $iBLER_i\left(L\left(\vec{\gamma}\right)\right)$, and then choosing the value of i that provides the highest BLER value below the target. The translation from LQM values to instantaneous BLER is typically performed by lookup tables. Hence, at each request interval, the UE updates the estimated SNR and calculates the LQM to determine, for a set of PRBs [12], the highest MCS i (mapped into a CQI index) such that the transmission does not exceed a established target BLER. These CQIs are reported to the BS to schedule next transmissions.

The translation from estimated instantaneous SNR to a LQM value and its corresponding CQI has to be designed in order to accomplish certain constraint on the BLER. However, these conditions can be defined in a variety of ways such as limiting the maximum instantaneous BLER (iBLER) or defining an average BLER (aBLER) target. The latter approach (based on aBLER) is the one adopted by most of wireless technologies like LTE [14]. Therefore, our description will be focused on the aBLER scenario. Next, it is described how this translation can be implemented.

For the sake of simplicity, from this point the used LQM function will be averaging the SNR vector $\vec{\gamma}$, that is, the LQM metric will be the instantaneous averaged SNR $L\left(\vec{\gamma}\right) = \gamma$. Thus, the mapping of instantaneous averaged SNR into CQI can be performed by associating each CQI value to an SNR region comprised between two SNR thresholds. Next, it is described how to calculate these SNR thresholds.

Typically [14], a set of n consecutive and not overlapping fading regions $\{\Re_i \doteq [\Psi_i, \Psi_{i+1}]\}_{i=0,1,..,n-1}$ are considered for the instantaneous SNR averaged in the whole PRB, γ. The set of SNR thresholds $\{\Psi_i\}_{i=0,1,..,n}$ defines the intervals, with Ψ_0 representing the minimum required SNR for transmission (outage condition) and $\Psi_n = \infty$. Within the fading region $\Re_i$, a certain Quadrature and Amplitude Modulation (QAM) constellation and coding redundancy factor (R_i bits/symbol) are selected as MCS to be suggested to the BS by means of a CQI report.

For a certain average SNR Γ[2], the average BLER under AMC can be evaluated

[2] It should be noticed that Γ is obtained by averaging γ in time, while γ is the averaged instantaneous SNR for a set of physical carriers.

as

$$aBLER(\Gamma, \{\Psi_i\}) = \sum_{i=0}^{n-1} \int_{\Psi_i}^{\Psi_{i+1}} iBLER_i^{AWGN}(\gamma) p^o(\Gamma, \gamma) d\gamma, \tag{3.1}$$

being $iBLER_i^{AWGN}(\gamma)$ the instantaneous BLER for a given MCS i over an Additive White Gaussian Noise (AWGN) channel, and $p^o(\Gamma, \gamma)$ the Probability Density Function (PDF) of the instantaneous SNR conditioned to transmission.

In this work, an uncorrelated Rayleigh channel has been assumed for the analysis. Then, channel model consists of a baseband complex envelope $c(t) = \alpha(t) \exp(j\theta(t))$ plus Additive White Gaussian Noise $n(t)$. $c(t)$ exhibits Rayleigh fading with average power gain $\mathrm{E}[\alpha(t)^2] = 1$. Flat block fading channel is assumed. The average received SNR is $\Gamma = P/(N_o B_W)$, where P denotes the constant received power (including transmission power, path loss and possibly shadowing), B_W is the receiver bandwidth and N_o is the noise power spectral density. The instantaneous SNR is related to the channel through $\gamma = \alpha^2 \Gamma$. Its PDF is well known to be given for Rayleigh channels by an exponential function [14]

$$p(\Gamma, \gamma) = \frac{1}{\Gamma} e^{-\gamma/\Gamma}. \tag{3.2}$$

Then, the instantaneous SNR conditioned to transmission is given by:

$$p^o(\Gamma, \gamma) = \begin{cases} \frac{1}{A(\Gamma, \Psi_0)\Gamma} e^{-\gamma/\Gamma}, \gamma > \Psi_0 \\ \\ 0, \text{else} \end{cases} \tag{3.3}$$

where $A(\Gamma, \Psi_0)$ is the probability of not being in outage, that is,

$$A(\Psi_0) = \int_{\gamma=\Psi_0}^{\infty} \frac{1}{\Gamma} e^{-\gamma/\Gamma} d\gamma = e^{-\Psi_0/\Gamma}. \tag{3.4}$$

Modeling instantaneous BLER by binary logistic functions According to equation (3.1), in order to evaluate the aBLER it is necessary to have at our disposal an expression for $iBLER_i^{AWGN}$, but as far as we know this is not available in the literature when turbo coding is used. Moreover, the exact value of the iBLER strongly depends on the specific decoder implementation [34]. Nevertheless, since the iBLER metric represents the probability of being in one of two states $\{error, non-error\}$, we propose the use of binary logistic regression [44]. This regression is a binary classifier based on one or more input variables. Thus, it is a useful tool to model iBLER curves for each MCS i over AWGN channels, for a given instantaneous SNR γ, by means of binary logistic functions as:

$$iBLER_i^{AWGN}(\gamma) \approx f_i(\gamma) = \frac{1}{1 + e^{-\alpha_{i_0}\gamma - \alpha_{i_1}}}, \tag{3.5}$$

where α_{i_0} and α_{i_1} values (see Table 3.1) are to be found from logistic regression over results of the actual decoder implementation (see Section 4.1.2 for further details). The accuracy of logistic functions after curve fitting process is shown in Fig. 3.2, where solid lines represent the analytic BLER curves and simulation results of a Soft Output Viterbi Algorithm (SOVA) based turbo decoder [36] are marked with circles.

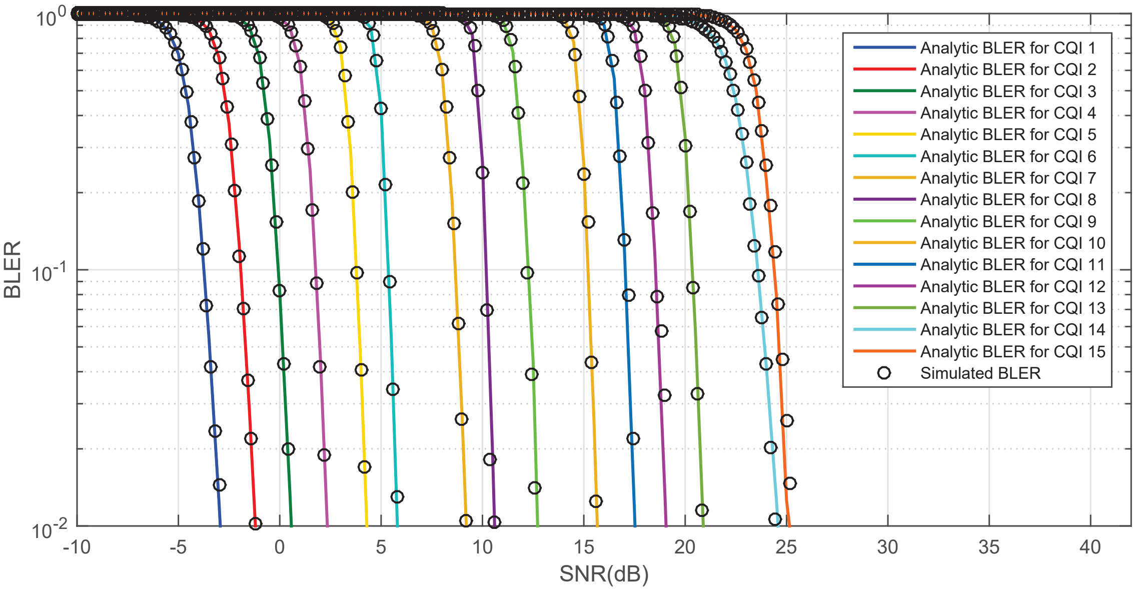

Figure 3.2: Curve fitting of logistic binary functions to model iBLER curves over AWGN channel

3.2 Outer Loop Link Adaptation (OLLA)

A static selection of the values for the AMC thresholds Ψ_i does not perfectly adjust the aBLER to a target since link conditions are inherently variant. Thus, in order to meet the $aBLER_T$, different sets of SNR thresholds $\{\Psi_i\}_{i=0,1,..,n}$ should be used for different link conditions. This can be modeled by rewriting each SNR threshold $\Psi_i = \gamma_i \cdot \theta$, being γ_i an initial value, and θ an offset which is to be design to meet the $aBLER_T$ for the specific link conditions. Then, equation (3.1) is modified to:

Table 3.1: Values of α_{i_0} and α_{i_1} of modeled iBLER curves

CQI index	α_{i_0}	α_{i_1}
1	-28.08	9.71
2	-20.59	11.05
3	-15.31	12.89
4	-11.09	14.45
5	-8.05	17.12
6	-6.56	20.56
7	-2.48	16.07
8	-2.39	22.83
9	-1.26	18.74
10	-0.67	20.02
11	-0.40	18.36
12	-0.26	16.62
13	-0.17	16.18
14	-0.04	7.02
15	-0.03	8.84

$$aBLER(\Gamma, \{\gamma_i\}, \theta) = \sum_{i=0}^{n-1} \int_{\gamma_i \theta}^{\gamma_{i+1}\theta} iBLER_i^{AWGN}(\gamma) p(\gamma) d\gamma. \tag{3.6}$$

Previous process is typically performed in practical implementations by the Outer Loop Link Adaptation (OLLA) technique [23–25]. The traditional OLLA operation consists in dynamically modifying the value of the offset, expressed in decibels, according to whether the previously transmitted data packet has been correctly received or not. In LTE, this information is extracted from the Cyclic Redundancy Code (CRC) [31]. Thus, OLLA can be seen as a discrete time system in which, each time k a CRC is received, the value of the offset $\theta[k]$ is updated according to the following equation [25]:

$$\theta_{dB}[k] = \theta_{dB}[k-1] + \Delta_{up} \cdot e[k] - \Delta_{down} \cdot (1 - e[k]), \tag{3.7}$$

being

- $\theta_{dB}[k] = 10 \cdot log_{10}(\theta[k])$.
- $e[k]$ an error indicator, whose value is 0 if the CRC is correct, or 1 if not. It corresponds to a dichotomous random variable whose average is the aBLER.
- Δ_{up} and Δ_{down} constant increment and reduction values, respectively, of the offset, in decibels. These two values are positive and should satisfy the following relation, which will be later justified, in order to meet the

$aBLER_T$ [25]:

$$aBLER_T = \frac{1}{1 + \frac{\Delta_{up}}{\Delta_{down}}}. \tag{3.8}$$

SNR threshold values are now modified at each instant k by a discrete-time offset, so they can be expressed as:

$$\Psi_i\left[k\right] = \gamma_i \cdot \theta\left[k\right]. \tag{3.9}$$

To sum up, the OLLA operation consists in increasing the offset value (and so increasing the value of the SNR thresholds) when error happens, and decreasing them when transmissions are correct. Therefore, $\theta\left[k\right]$ values higher than 1, i.e. positive values when expressed in decibels, increase transmission robustness. On the other hand, when values lower than 1, i.e. negative values when expressed in decibels, decrease it.

Note that adding an offset value (in decibels) to the SNR thresholds is equivalent to subtract the same offset value to the estimated SNR. This process of modifying the estimated SNR γ to obtain an effective SNR $\hat{\gamma}$, instead of modifying the SNR threshold, is the typical way to implement the OLLA technique. The scheme of this implementation is shown in Fig. 3.3, which is the result of adding the OLLA technique to the traditional LA scheme presented in Fig. 3.1. However, for the rest of the analysis it will be considered that the SNR thresholds are modified.

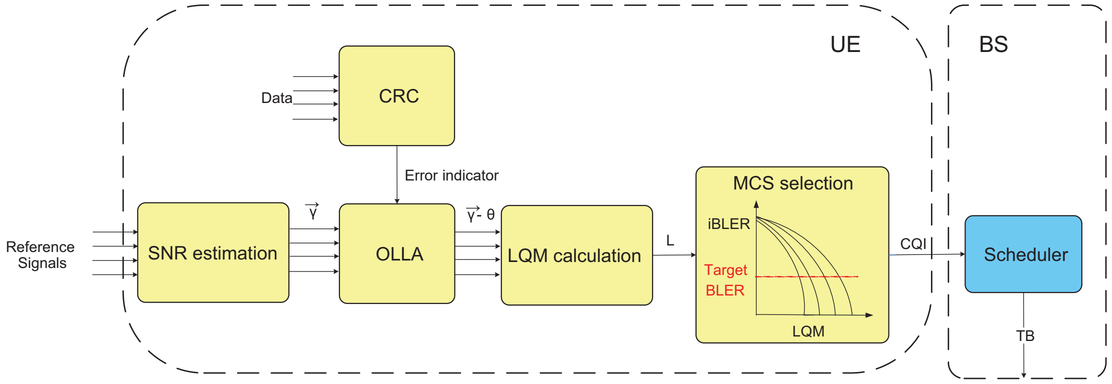

Figure 3.3: Scheme of the LA process with OLLA

3.2.1 OLLA Convergence in Average

It should be noticed that OLLA dynamics implies that the offset value $\theta_{dB}\left[k\right]$ is continuously being updated by adding Δ_{up} (if error) or subtracting Δ_{down} (if not).

Thus, it will never converge to a single value. However, the offset θ presented in (3.6) is a single value that ensures the $aBLER_T$ for stationary link conditions. The reason of this difference is that in (3.6) θ is used in an averaging process, while in case of the OLLA $\theta_{dB}[k]$ is an instantaneous value. Therefore, to study the convergence of the OLLA process, it has been used averaged values.

In this subsection it is studied the convergence in average of the OLLA algorithm for stationary link conditions, as well as conditions that should be satisfied in order to achieve this convergence. For this purpose, mathematical expectation can be applied to (3.7) to find the averaged $E[\theta_{dB}[k]] = \theta_{dB_k}$ value at time instant k, obtaining the following equation after reordering:

$$\theta_{dB_k} = \theta_{dB_{k-1}} + (\Delta_{up} + \Delta_{down}) \cdot E[e[k]] - \Delta_{down}. \tag{3.10}$$

Note that since the average value of $\theta_{dB}[k]$ depends on k, it is not an ergodic process. Only when $k \to \infty$ the process can be considered ergodic. Expression (3.10) corresponds to a difference equation, i.e. there is a recurrence relation between the terms in the form of $\theta_{dB_k} = T\left(\theta_{dB_{k-1}}\right)$, being $T(\theta_{dB})$ the recurrence function. Hence, according to the Banach fixed-point theorem [45], when $k \to \infty$ the equation will converge to the only value such that $\theta_{dB}^o = T(\theta_{dB}^o)$ if certain conditions are fulfilled. The following equation is obtained when convergence in average is reached:

$$(\Delta_{up} + \Delta_{down}) \cdot E[e[k]] - \Delta_{down} = 0. \tag{3.11}$$

Note that $E[e[k]]$ is the average error rate, i.e. the aBLER. Thus, in order to ensure that the converged offset value $\theta_{dB}^o = 10 \cdot log_{10}(\theta^o)$ meets the $aBLER_T$, the aBLER should be forced to this value, obtaining the the following relation:

$$aBLER_T = E[e[k]] = \frac{1}{1 + \frac{\Delta_{up}}{\Delta_{down}}}, \tag{3.12}$$

which was already presented in (3.8) and now it has been justified. Therefore, under certain link conditions, the OLLA algorithm converges in average to a value θ^o which is the value of θ to be introduced in (3.6) in order to meet the $aBLER_T$.

Equation (3.12) manifests that there are infinite suitable combinations of Δ_{up} and Δ_{down} that ensures the $aBLER_T$. However, the specific combination selected has to accomplish the convergence criteria required by the Banach fixed-point theorem applied to (3.10), whose two sufficient conditions are described next.

First Banach fixed-point theorem condition

The function $T\left(\theta_{dB}\right)$ should be a contraction mapping, that is, it should satisfy that

$$T\left(\theta_{dB}\right) \in \left[\theta_{min}, \theta_{max}\right], \ \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right]. \tag{3.13}$$

It can be easily shown that this condition is always fulfilled by the OLLA. Firstly, since $E\left[e\left[k\right]\right]$ is the average number of errors, i.e. the aBLER, its range of values are comprised between 0 and 1. Thus, there will be a value $\theta_{dB} = \theta_{min}$ low enough to ensure that $\mathrm{E}\left[e\left[k\right]\right] = 1$, which means that $T\left(\theta_{min}\right) = \theta_{min} + \Delta_{up}$. Then, as θ_{dB} increases, the value of $\mathrm{E}\left[e\left[k\right]\right]$ will decrease until θ_{dB} raises a value θ_{max} high enough to ensure that $\mathrm{E}\left[e\left[k\right]\right] = 0$, which means that $T\left(\theta_{max}\right) = \theta_{max} - \Delta_{down}$. As a result of that, the values of $T\left(\theta_{dB}\right)$ will be comprised between:

$$T\left(\theta_{dB}\right) \in \left[\theta_{min} + \Delta_{up}, \theta_{max} - \Delta_{down}\right] \in \left[\theta_{min}, \theta_{max}\right], \ \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right]. \tag{3.14}$$

Second Banach fixed-point theorem condition

The second condition that the function $T\left(\theta_{dB}\right)$ should fulfill is that

$$\left|T'\left(\theta_{dB}\right)\right| < 1, \quad \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right] \tag{3.15}$$

being $T'\left(\theta_{dB}\right)$ the first order derivative of $T\left(\theta_{dB}\right)$. To check this condition, it is necessary an expression for $\mathrm{E}\left[e\left[k\right]\right]$, this is, for the aBLER. This expression is provided by (3.6), and it depends on the specific parameters of the binary logistic regression carried out to model the $iBLER_i^{AWGN}$ curves. Expression in (3.6) is very complex to be included in (3.15). Then in order to find a more tractable expression, $aBLER(\Gamma, \{\gamma_i\}, \theta)$ values have been obtained numerically by simulations, and from these value, the aBLER is modelled. These simulations have been obtained for different θ_{dB} values according to (3.3), (3.5) and (3.6), and they have been carried out assuming $iBLER_i^{AWGN}$ curves presented in Fig. 3.2. Results are presented in Fig. 3.4 (solid line) for a mean SNR value of $\Gamma = 15dB$.

The shape of the aBLER curve is similar to the iBLER. Thus, it could also be modeled by means of logistic regression, which was introduced in subsection 3.1. In this case, binary logistic functions do not fit exactly with the simulated values, since the slope of each curve are different. Then, we propose to model this curve by means of a modified binary logistic function, which introduces a parameter s which controls the slope, as:

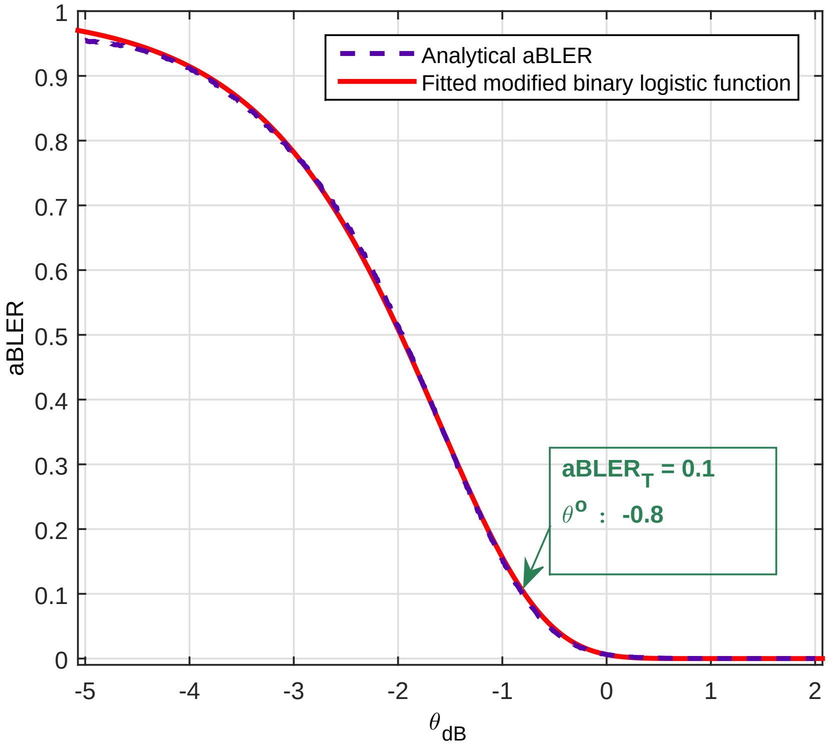

Figure 3.4: Curve fitting of modified logistic function to model aBLER for $\Gamma =$ 15dB

$$E\left[e\left[k\right]\right] = aBLER(\Gamma, \theta_{dB}) \approx f_m(\theta_{dB}) = \frac{1}{\left(1 + e^{-\alpha_0 \theta_{dB} - \alpha_1}\right)^s}. \tag{3.16}$$

After finding by logistic regression the values of α_0, α_1 and s (see Table 3.2, which shows the parameters of the aBLER modeling for $\Gamma = 15dB$) it is presented the fitted curve in Fig. 3.4 in dashed line. It can be seen how the proposed expression perfectly fits the simulated results. Note that the absolute value of the slope is higher than 1, which means that the slope of the aBLER curve is higher than slopes of iBLER curves.

Table 3.2: Parameters for aBLER modeling

Parameter	Value
Γ	15dB
α_0	1.15
α_1	-1.03
s	17
$aBLER_T$	0.1

By using the proposed modified logistic function, it can be evaluated the expression of $T\left(\theta_{dB}\right)$, according to (3.10), as:

$$T\left(\theta_{dB}\right)=\theta_{dB}+\left(\Delta_{up}+\Delta_{down}\right)\cdot f_m\left(\theta_{dB}\right)-\Delta_{down} \tag{3.17}$$

Next, it is derived the condition to ensure that $\left|T'\left(\theta_{dB}\right)\right|<1$ from (3.12), (3.16) and (3.17). First, we find the value of $T'\left(\theta_{dB}\right)$ as:

$$T'\left(\theta_{dB}\right)=1+\left(\Delta_{up}+\Delta_{down}\right)\cdot f'_m\left(\theta_{dB}\right)=1+\left(\Delta_{up}+\Delta_{down}\right)\cdot\frac{\alpha_0 e^{-\alpha_0\theta_{dB}-\alpha_1}}{\left(1+e^{-\alpha_0\theta_{dB}-\alpha_1}\right)^{s+1}} \tag{3.18}$$

being $f'_m\left(\theta_{dB}\right)$ the first order derivative of $f_m\left(\theta_{dB}\right)$. Since $f_m\left(\theta_{dB}\right)$ is a decreasing function, that is $f'_m\left(\theta_{dB}\right)\leq 0$, and both Δ_{up} and Δ_{down} are positive, the following condition has to be guaranteed in order to ensure the convergence condition:

$$\left(\Delta_{up}+\Delta_{down}\right)\cdot\left|f'_m\left(\theta_{dB}\right)\right|<2 \ \Rightarrow \left|\mathrm{f}'_{\mathrm{m}}\left(\theta_{\mathrm{dB}}\right)\right|<\frac{2}{\left(\Delta_{\mathrm{up}}+\Delta_{\mathrm{down}}\right)}. \tag{3.19}$$

In order to obtain further conditions, we find the maximum of $\left|f'_m\left(\theta_{dB}\right)\right|$, corresponding to values for which $f''_m\left(\theta_{dB}\right)=0$, to ensure that the condition imposed by (3.19) is fulfilled:

$$f''_m=-\alpha_0^2\frac{e^{-\alpha_0\theta_{dB}-\alpha_1}\left(1-se^{-\alpha_0\theta_{dB}-\alpha_1}\right)}{\left(1+e^{-\alpha_0\theta_{dB}-\alpha_1}\right)^{s+2}}. \tag{3.20}$$

Then

$$f''_m=0,\alpha_0>0\Rightarrow\left(1-se^{-\alpha_0\theta_{dB}-\alpha_1}\right)=0. \tag{3.21}$$

Therefore, we find that the offset value $\theta_{dB_{max}}$ for which the maximum of $\left|f'_m\left(\theta_{dB}\right)\right|$ is achieved is:

$$\theta_{dB_{\max}}=-\frac{1}{\alpha_0}\cdot\left(\alpha_1+\ln\left(\frac{1}{s}\right)\right) \tag{3.22}$$

Replacing in (3.19) θ_{dB} by the value $\theta_{dB_{\max}}$ found in (3.22), the maximum value of $\left|f'_m\left(\theta_{dB}\right)\right|$ is given by

$$\left|f'_m\left(\theta_{dB_{\max}}\right)\right|=\left|\alpha_0\right|\cdot\frac{1}{\left(1+{}^{1}\!/_{s}\right)^{(s+1)}} \tag{3.23}$$

As it was said before, the fact that $f_m\left(\theta_{dB}\right)$ is a decreasing function implies that $s\geq 0$. It can be easily seen that higher values of $f_m\left(\theta_{dB}\right)$ are achieved as s

is increased. Then, we get an upper bound of $f'_m(\theta_{dB})$ when $s \to \infty$ as:

$$\left| f'_{m_{\max}} \right| = \frac{|\alpha_0|}{e}. \tag{3.24}$$

Finally, from (3.12), (3.19) and (3.24) we find that the second condition to ensure the OLLA convergence is given by

$$\Delta_{down_{max}} < \frac{2e \cdot aBLER_T}{|\alpha_0|}. \tag{3.25}$$

Thus, convergence is ensured if the decrement step does not raise a maximum value, which is determined by α_0, that is, by the curve of aBLER for a certain Γ value of mean SNR. In case of parameters of Table 3.2, the value of the maximum Δ_{down} than ensures convergence is $\Delta_{down_{max}} = 0.52$.

According to previous results, there is a range of Δ_{down} values, and so of Δ_{up} values, that ensures convergence. In Fig. 3.5 it is presented a convergence process when different values of Δ_{down} are used, for given link conditions of Table 3.2.

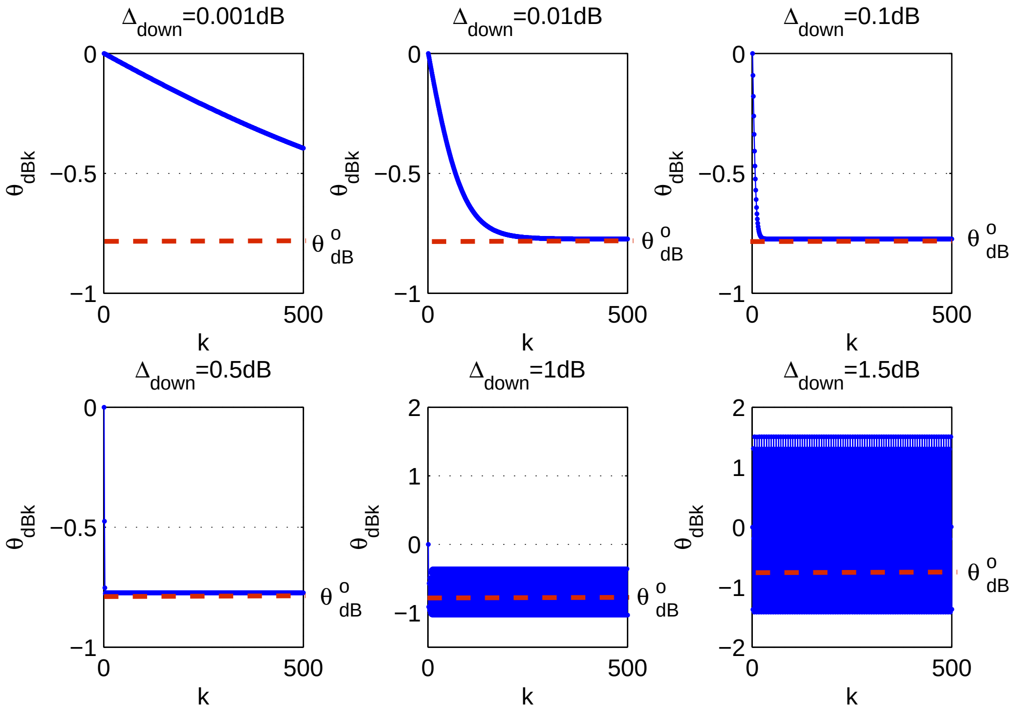

Figure 3.5: OLLA convergence process in average for different Δ_{down} values

Fig. 3.5 shows how the converged value θ_{dB_k} raises its final value $\theta^o_{dB} \approx -0.8$ (dashed line), faster as the size of Δ_{down} is higher. Note that this convergence value is the same that the obtained in Fig. 3.4 by using (3.16). Once Δ_{down} exceeds $\Delta_{down_{max}}$, whose value is 0.48 for the proposed scenario, the OLLA begins to diverge, being this divergence more remarkable as Δ_{down} increases. However, since this is a necessary but not a sufficient condition, it happens that there does not seem to be divergence for the value $\Delta_{down} = 0.5$ in this scenario. Therefore, it would be advisable to choose the highest possible value of Δ_{down} in order to achieve convergence as fast as possible.

3.2.2 OLLA performance

In this subsection the OLLA performance is analyzed, under stationary link conditions. As it will be shown, when in convergence, the instantaneous offset value $\theta_{dB}\left[k\right]$ fluctuates around the converged averaged value θ^{o}_{dB} following the instantaneous channel variations. The amplitude of these fluctuations will be related to the values of Δ_{up} and Δ_{down}. Thus, high values of these two parameters, even if they guarantee the OLLA convergence, may lead to a high variance in the instantaneous offset value that could degrade the OLLA performance. On the other hand, too small values of Δ_{up} and Δ_{down} may imply that the OLLA could not follow channel variations if they are too fast.

In Fig. 3.6 they are presented the instantaneous values of $\theta[k]$ for the same simulation conditions that in Table 6.2, for a set of values of Δ_{down} that guarantee convergence ($\Delta_{down} < \Delta_{down_{max}} = 0.48$). These results show that for the lowest Δ_{down} value, it takes several steps to the OLLA to reach an state for which the offset stabilizes around a certain value, which is the converged value $\theta^{o}_{dB} \approx -0.8$ of Fig. 3.5. As the value of Δ_{down} increases, it takes less steps to the OLLA to stabilize; however, offset variance also increases, which may cause a degradation in OLLA performance.

As it was previously stated, a high OLLA variance can degrade its performance. This fact is shown in Table 3.3, in which it is presented the achieved throughput and the aBLER for the different Δ_{down} values, under same conditions than in Fig. 3.6. Results reveal that, as the size of the step increases, there is a reduction of the spectral efficiency, although all implementations meet the $aBLER_T$. The reduction of spectral efficiency for the maximum simulated value that guarantees convergence ($\Delta_{down} = 0.48dB$) is about a 8% with respect to the step value with better throughput performance ($\Delta_{down} = 0.01dB$). The reason is that high step values may cause that when an error happens, the Δ_{up} step leads the reported CQI to a more robust MCS than that necessary to ensure an error free transmission, loosing throughput compared to that of lower step values.

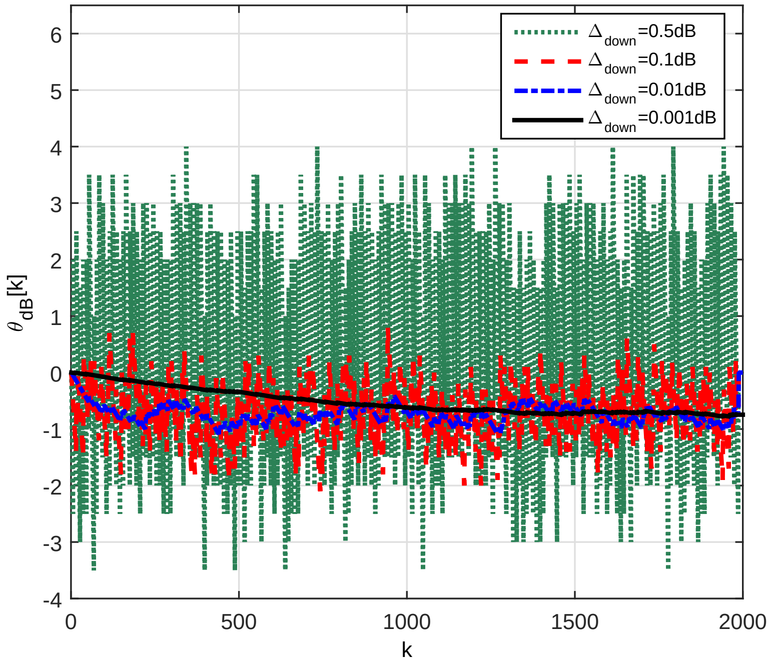

Figure 3.6: Instantaneous offset for an uncorrelated flat Rayleigh channel

Table 3.3: System performance for OLLA with different Δ_{down} sizes

Δ_{down} size (dB)	Spectral Efficiency (bps/Hz)	aBLER
0.001	2.98	0.1
0.01	3	0.1
0.1	2.96	0.1
0.5	2.76	0.1

Chapter 4

Enhanced Outer Loop Link Adaptation

In the previous chapter the traditional Outer Loop Link Adaptation (OLLA) algorithm has been deeply studied, showing that a key factor to optimize the performance is the appropriate selection of its step size. However, the way this selection should be carried out is not clear. Moreover, the performance of the OLLA will also depend on the specific implementation of features at the receiver such as the turbo decoder or the channel estimation method.

In addition to that, another problem is that the OLLA only updates its offset every time k a Cyclic Redundancy Code (CRC) is received, i.e. when a transmission is done. In real implementations, it is usual not to have full buffer traffic but discontinuous and variable traffic, such as an streaming source or a Machine to Machine (M2M) communication. Then, the User Equipment (UE) will be most of the time in idle mode [46]. In conclusion, a combination of changing channel condition together with these kinds of traffic patterns could cause a performance degradation since the OLLA may not be able to update its offset fast enough to follow the channel variations.

From the study carried out in previous chapter, improvements in the implementation of the traditional OLLA can be inferred. Thus, in this chapter a different approach to the OLLA technique is proposed, the enhanced OLLA (eOLLA) [26], which can significantly improve the performance of the traditional OLLA.

4.1 enhanced Outer Loop Link Adaptation (eOLLA)

4.1.1 Proposed eOLLA

In this section, we propose a modification of the traditional OLLA algorithm, the enhanced Outer Loop Link Adaptation (eOLLA), which is able to adapt the size of the steps according to the convergence status of the system. Furthermore, the proposed eOLLA updates its offset independently of whether a transmission is carried out or not.

The error indicator $e\left[k\right]$ used in the traditional OLLA can be seen as a one bit instantaneous BLock Error Rate (BLER) estimator (1 if error, 0 if not). Thus, this value could be replaced by a more accurate instantaneous BLER estimation. To carry out the implementation of the proposed eOLLA, it is assumed that it is available a model for the instantaneous BLER of each Modulation and Coding Scheme (MCS) i over Additive White Gaussian Noise (AWGN) channels for instantaneous Signal to Noise Ratio (SNR), this is, $iBLER_i^{AWGN}\left(\gamma\right)$. The proposed eOLLA is implemented from (3.7) as follows:

1. If a transmission is received, update the corresponding $iBLER_i^{AWGN}$ model according to the MCS i used in this transmission by using the CRC extracted from the received data (this step is optional).

2. Estimate the instantaneous SNR value at each instant t, that is, $\gamma\left[t\right]$.

3. At each Transmission Time Interval (TTI) t, calculate the estimated instantaneous BLER value $B\left[t\right] = iBLER_i^{AWGN}(\gamma\left[t\right])$.

4. Get the offset value as:

$$\theta_{dB}\left[t\right] = \theta_{dB}\left[t-1\right] + \Delta_{up} \cdot B\left[t\right] - \Delta_{down} \cdot \left(1 - B\left[t\right]\right). \tag{4.1}$$

In the eOLLA algorithm description it should be noticed that the index k of equation (3.7) of the traditional OLLA description has been replaced by the index t in equation (4.1). This means that the offset of the eOLLA will be updated every TTI t (every time a transmission can be potentially carried out) instead of every time k a CRC is received (every time a transmission is carried out). Then, in case of discontinuous transmissions, the proposed eOLLA updates its offset more frequently than traditional OLLA. This implies that the eOLLA is able to easily follow the temporal variations of the channel, thus improving the Adaptive Modulation and Coding (AMC) performance.

This continuous updating of the eOLLA offset can be easily performed in LTE since there is an estimation of the SNR available every TTI t. In the downlink of LTE and LTE-A, the Base Station (BS) transmits Reference Signals (RS) [39] at each subframe, in both connected and idle modes. Thus, every UE is able to perform SNR estimation from this RS at each TTI, and so to update its eOLLA offset.

The fitting process referred in step 1) of the proposed algorithm is used to adjust the instantaneous BLER model to the specific system implementation used. This process is optional since it can be avoided in case of a fitted model is available.

For the purpose of a better understanding of the eOLLA behavior, step 4) can be rewritten, according to (3.12), as

$$\theta_{dB}\left[t\right] = \theta_{dB}\left[t-1\right] + \Delta_{down} \cdot \left(\frac{B\left[t\right]}{aBLER_T} - 1\right). \tag{4.2}$$

From previous equation it can be deduced that the proposed eOLLA adapts the increment to be applied to the offset according to the difference between the estimated instantaneous BLER $B\left[t\right]$ and the target aBLER. Then, if $B\left[t\right]$ is equal to 0, the offset will be decreased by Δ_{down}, which will decrease the robustness of the next transmission, in the same way than in the traditional OLLA. As $B\left[t\right]$ increases, the size of the decreasing step reduces, and so increases the robustness of the next transmission until $B\left[t\right] = aBLER_T$, which is the equilibrium point. At this point, no step will be applied to the offset, since if the eOLLA remains in this state the average BLER will meet the target aBLER. In case that $B\left[t\right]$ exceeds the $aBLER_T$, the offset will be increased, and thus the robustness of the next transmission, until $B\left[t\right] = 1$. In this case, the offset will be increased by Δ_{up}, as in the traditional OLLA.

Hence, for the eOLLA, if the offset value leads to either a very high or a very low instantaneous BLER, high step values are used in order to correct this situation as fast as possible. Then, as the iBLER is closer to the average target BLER, lower step sizes are used in order to reduce the variance of the MCS to not degrade the throughput.

Since $E\left[B\left[t\right]\right]$ is the same than $E\left[e\left[t\right]\right]$, the convergence analysis of the traditional OLLA carried out in subsection 3.2.1 can be applied to the eOLLA. Thus, to ensure the convergence of the eOLLA, the condition from (3.25) has to be satisfied. Note that, under stationary conditions, the offset value for which

$E\left[B\left[t\right]\right] = aBLER_T$ corresponds to the convergence value θ^o_{dB} previously described.

4.1.2 Logistic regression implementation

From the eOLLA algorithm description it follows that it is necessary to have available an expression of the instantaneous BLER $iBLER_i^{AWGN}(\gamma\left[t\right])$. However, these expressions are not available at the literature, since they are very dependent of specific system features such as the turbo decoder implementation or the channel estimation method. Thus, in this subsection we provide a method to estimate an expression for the instantaneous BLER based on measurements of errors during the reception of transmitted data. This information is available in an LTE system whatever its specific implementation through the CRC checking. Thus, this proposed method to estimate the instantaneous BLER can be used in the step 1) of the proposed eOLLA algorithm.

According to subsection 3.1, instantaneous BLER for a MCS i over AWGN channels can be modeled by means of binary logistic functions. Therefore, logistic regression can be used to find the α_{i_0} and α_{i_1} values of equation (3.5) that better fit with the $iBLER_i^{AWGN}$ curve. In practice, this logistic regression can be easily implemented by the well-known gradient descend algorithm [47]. This algorithm is typically used to minimize a cost function $J\left(\alpha\right)$ as follows:

$$\alpha := \alpha - \lambda \frac{\partial}{\partial \alpha} J\left(\alpha\right) \tag{4.3}$$

where λ is a parameter that controls the convergence speed. Thus, first of all it is necessary to provide the logistic regression cost function, given by [48]:

$$J\left(\alpha\right) = y \cdot \log\left(f_\alpha\left(x\right)\right) - \left(1-y\right) \cdot \log\left(1 - f_\alpha\left(x\right)\right) \tag{4.4}$$

being x the input to the system, y the class where the input x falls into (with only two possible values: 0 or 1), and f_α a binary logistic function to model the mapping of x into a class y. Thus, to translate the generic logistic regression function to our problem, we associate the input x to the instantaneous SNR value γ, the classification result y as an error indicator e, and f_α as the binary logistic function of (3.5). Then, applying partial derivation of the cost function, the resulting logistic regression process to be used in proposed eOLLA is given by:

$$\begin{cases} \alpha_{i_0}^{k} = \alpha_{i_0}^{k-1} - \lambda \cdot \left(\frac{1}{1+e^{-\alpha_{i0}^{k-1} - \alpha_{i1}^{k-1} \cdot \gamma[k-1]}} - e\left[k\right] \right) \\ \alpha_{i_1}^{k} = \alpha_{i_1}^{k-1} - \lambda \cdot \left(\frac{1}{1+e^{-\alpha_{i0}^{k-1} - \alpha_{i1}^{k-1} \cdot \gamma[k-1]}} - e\left[k\right] \right) \cdot \gamma\left[t-1\right], \end{cases} \tag{4.5}$$

where $e\left[k\right]$ is, as in (3.7), an error indicator of the received packet at the instant k, which is obtained from the CRC, and $\gamma\left[t-1\right]$ is the estimated instantaneous SNR during the previous TTI t. Note that the values of $\alpha_{i_0}^{k}$ and $\alpha_{i_1}^{k}$ have to be updated simultaneously. Regarding λ, low values will lead to a slower but more accurate convergence, while high values will accelerate the convergence although it may also cause oscillations. However, it should be noticed that the logistic regression cost function is convex, so convergence is guaranteed.

4.1.3 eOLLA performance

In Fig. 4.1 it is shown the instantaneous offset value $\theta\left[t\right]$ of the eOLLA for the same conditions that in Fig. 3.6. Note that in this case $t = k$ since a transmission is carried out every TTI. In this figure it can be seen that, as it was mentioned before, convergence values are the same than those achieved by the traditional OLLA. In addition to that, the instantaneous offset values has lower variance than those obtained in Fig. 3.6 for the same value of Δ_{down}. Furthermore, in Table 4.1 results of throughput and aBLER are presented. For the eOLLA, increasing Δ_{down} does not lead to a throughput degradation as remarkable as in the case of the traditional OLLA (see Table 3.3). Thus, the throughput degradation when using the highest Δ_{down} considered in these simulations is about a 4% for the eOLLA, while for the traditional OLLA this degradation is doubled. In both cases the $aBLER_T$ is met.

Table 4.1: System performance for eOLLA with different Δ_{down} sizes

Δ_{down} size (dB)	Spectral Efficiency (bps/Hz)	aBLER
0.001	2.99	0.1
0.01	3	0.1
0.1	2.99	0.1
0.5	2.89	0.1

Results presented for the traditional OLLA in previous chapter, and the eOLLA in this chapter do not correspond to a realistic scenario, since an uncorrelated channel is assumed. However, they are useful in order to understand the dynamic of both the OLLA and the proposed eOLLA. In practice, AMC cannot be used in uncorrelated channels due to CQI reports outdating. Next,

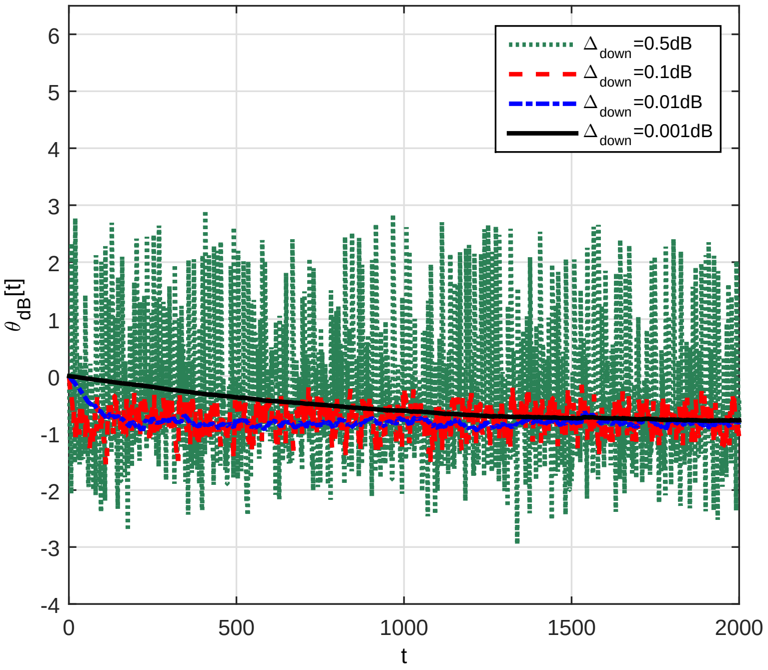

Figure 4.1: Instantaneous offset for an uncorrelated flat Rayleigh channel

both OLLA and eOLLA have been evaluated in a correlated channel scenario. Furthermore, a bursty traffic pattern has been taken into account in order to give a complete picture.

In Fig. 4.2 it is shown instantaneous offset values for both the traditional OLLA and the eOLLA, for a correlated flat Rayleigh channel with a Doppler Frequency (f_D) of 7 Hz and $aBLER_T = 0.1$, when a small Δ_{down} step is used (0.001 dB). The transmission periods are expressed in percentage of the maximum traffic load case. First of all, notice that the same results are achieved by the eOLLA whatever the traffic load, since the eOLLA only needs the estimated iBLER to update the offset, which is always available. In contrasts, different performance are achieved by the traditional OLLA depending on the traffic load, since it needs a CRC reception to update its state. As a consequence, when full buffer is assumed, i.e. $t = k$, both OLLA implementations have a similar performance; but as the traffic load decreases, it is harder for the traditional OLLA to follow the temporal channel variations since its offset is only updated when a CRC is received, thus degrading its performance compared to the eOLLA, which updates its offset every TTI.

On the other hand, if the same simulation is carried out with a high step size $\Delta_{down} = 0.5$, which is high but still ensures convergence, it can be shown that

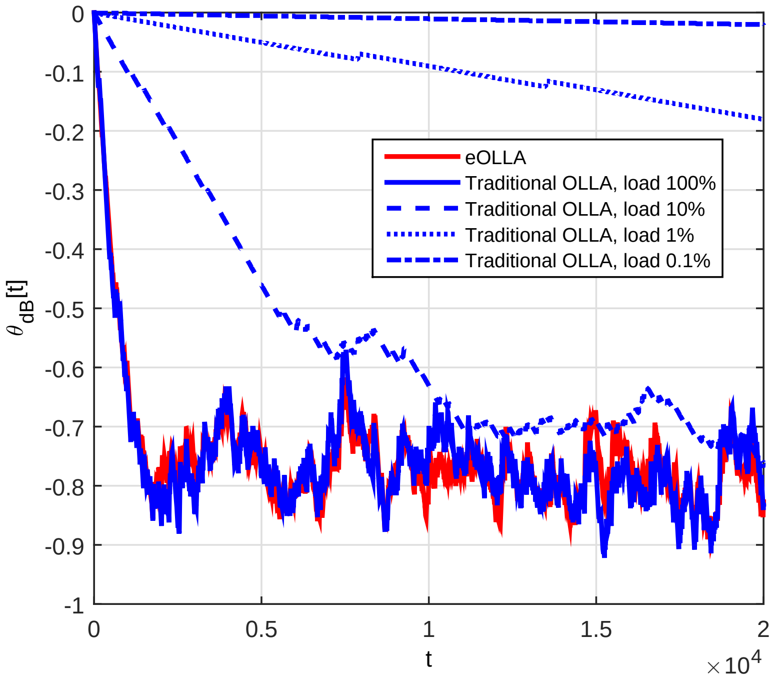

Figure 4.2: Instantaneous offset for a correlated flat Rayleigh channel ($f_D = 7Hz$) for traditional OLLA and eOLLA with $\Delta_{down} = 0.001dB$

this convergence is easily achieved whatever the traffic load. However, low load is still a problem for the traditional OLLA since it is not able to properly follow the channel, as it is shown in Fig. 4.3, where a short extract of the simulation is present in order to facilitate the observation of this fact.

Table 4.2 summarizes, for both the traditional OLLA and the eOLLA, the spectral efficiency for different traffic loads and Δ_{down} sizes, for $aBLER_T = 0.1$ and a correlated flat Rayleigh channel with $f_D = 7Hz$. First of all, it is noticeable that performance degrades for both implemented OLLA techniques for high Δ_{down} sizes, whatever the traffic load. However, while this degradation does not exceed the 4% for the eOLLA, in case of the traditional OLLA this degradation varies from the 5% to the 10%. Furthermore, given a Δ_{down} step size, the performance of the eOLLA does not vary with the traffic load. However, when using the traditional OLLA, low traffic load can degrade its performance with respect to the eOLLA up to a 17%. To sum up, the eOLLA not only improves the performance of the traditional OLLA but also reduces the influence of the Δ_{down} steps size and the traffic load in this performance.

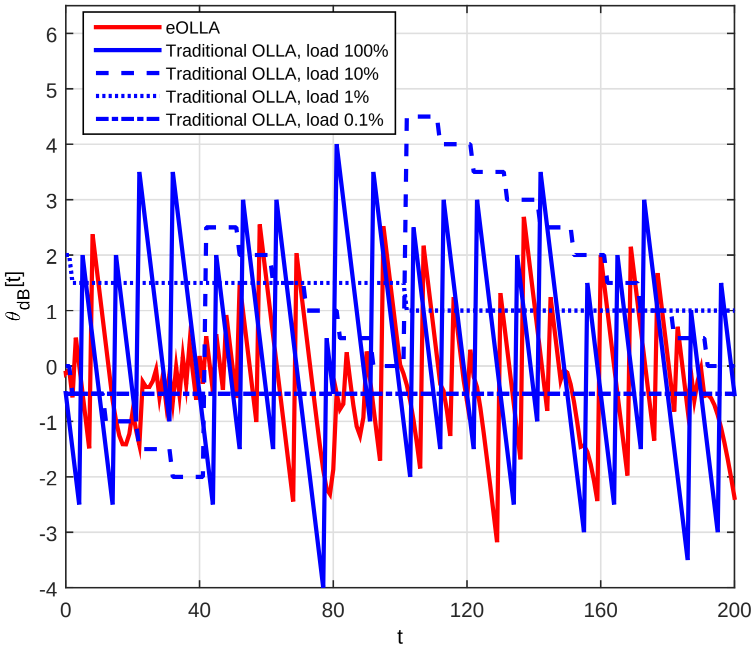

Figure 4.3: Instantaneous offset for a correlated flat Rayleigh channel ($f_D = 7Hz$) for OLLA and eOLLA with $\Delta_{down} = 0.5dB$

Table 4.2: Spectral efficiency (bps/Hz) comparison between traditional OLLA and eOLLA in correlated flat Rayleigh channel ($f_D = 7Hz$)

	load 100%		load 10%		load 1%		load 0.1%	
Δ_{Down}	OLLA	eOLLA	OLLA	eOLLA	OLLA	eOLLA	OLLA	eOLLA
0.001dB	2.43	2.45	0.24	0.245	0.0215	0.0245	0.0021	0.00245
0.01dB	2.45	2.5	0.244	0.25	0.022	0.025	0.0021	0.0025
0.1dB	2.41	2.4	0.23	0.24	0.021	0.024	0.0020	0.0024
0.5dB	2.22	2.4	0.22	0.24	0.021	0.024	0.0020	0.0024

4.2 eOLLA Application and Simulation Scenarios

In this section, the proposed eOLLA has been implemented in a complete 3GPP-LTE-A downlink simulator [36], introduced in Appendix A, in order to present realistic scenarios for which the eOLLA can significantly improve the performance of the traditional OLLA. These scenarios are based on LTE and LTE-A features. Both the traditional OLLA and the eOLLA have been evaluated for different sizes of Δ_{down}, all of them ensuring convergence according to subsection 3.2.1.

Table 4.3 summarizes the main simulation parameters used to obtain the results presented in this work.

Table 4.3: Simulation Parameters

Parameter	Value
Carrier frequency	2 GHz
Sampling frequency	7.68 MHz
System bandwidth	5 MHz
FFT size	512
Number of data subcarriers	300
OFDM symbols per subframe	14
Allocable PRBs	25
Channel model	Flat Rayleigh
Mobile terminal speed	3 km/h
Number of antennas	1x1
Channel estimation method	Low-pass filter [49]
Interference and Noise power estimation	Error based
Reference signals overhead	According to 3GPP TS 36.211 [39]
Turbo decoder	SOVA based
Number of CQI bits	4 [12]
Average SNR	15dB
Δ_{down}	0.001dB - 0.5dB

4.2.1 High traffic load with continuous transmission scenario

In this subsection it is evaluated the performance of both the traditional OLLA and the eOLLA in a scenario with high traffic load. In this scenario, both algorithms will update their offsets continuously as there will always be queued packets to be transmitted. Then, the benefit that the eOLLA provides with respect to the traditional OLLA is its ability to dynamically adapt the size of the step, so reducing the offset variance. Next, they are shown results of BLER, throughput, goodput[1] mean packet delay and jitter[2] [50] for different sizes of Δ_{down}. Except for BLER, these results are expressed in percentage of the maximum value raised (which is also indicated in the figures).

As it is shown in Fig. 4.4, there is no difference in terms of BLER since for both algorithms the $aBLER_T$ is met. However, it can be seen in Fig. 4.5 and Fig. 4.6 that a better performance in terms of throughput and goodput is achieved by the eOLLA for the majority of Δ_{down} sizes. Only for the the lowest sizes results are pretty similar since the margin of adaptation of the step is very low. As

[1] Goodput is defined as the amount of information bits that have been correctly received.

[2] Jitter is defined as the variability in the delay of received data.

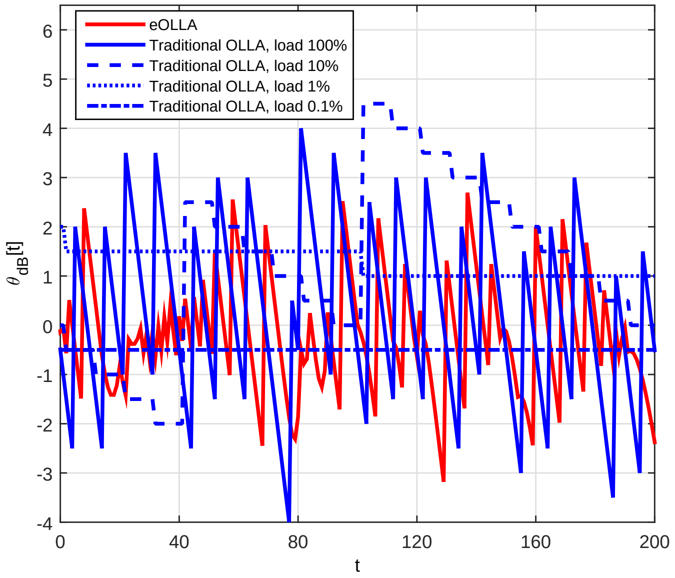

Figure 4.4: BLER comparison for high load traffic with continuous transmission scenario

the step size increases this outperforming is more remarkable, achieving a gain close to the 15%. Finally the performance in terms of delay is evaluated, which is a very useful metric [51]. Then, Fig. 4.7 and Fig. 4.8 show results for mean packet delay and jitter, respectively. In this case the eOLLA also outperforms the traditional OLLA. Finally, while for the traditional OLLA the optimum size of Δ_{down} is the lowest, the eOLLA tolerates higher sizes without a significant degradation of its performance.

4.2.2 High load traffic with bursty transmission scenario

In some scenarios, it is possible that some UEs cannot be served, even if they have queued packets and good channel conditions, because of low priority assignment or because the BS is serving a great number of UEs, remaining the UE in idle mode. In such scenarios, data packets are exchanged over long periods of time. Thus, the channel conditions may change during the time interval between packets, so the offset of traditional OLLA would be outdated since it can only be updated when an CRC is received. However, as it was shown in previous section, the eOLLA is able to follow the channel variations even if no packets are received, since it uses the instantaneous SNR estimation to update its offset.

In this section, it is evaluated an scenario for which a transmission is allowed every 100 TTIs. In Fig. 4.9-4.13 they are presented the results of performance.

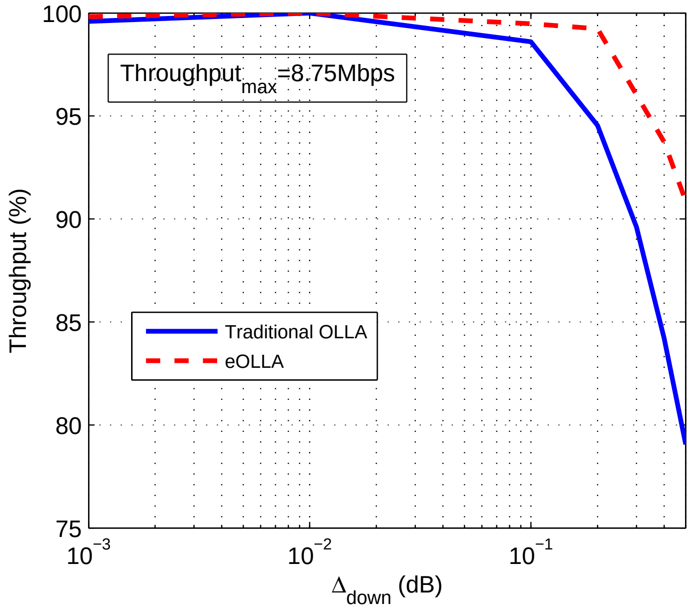

Figure 4.5: Throughput comparison for high load traffic with continuous transmission scenario

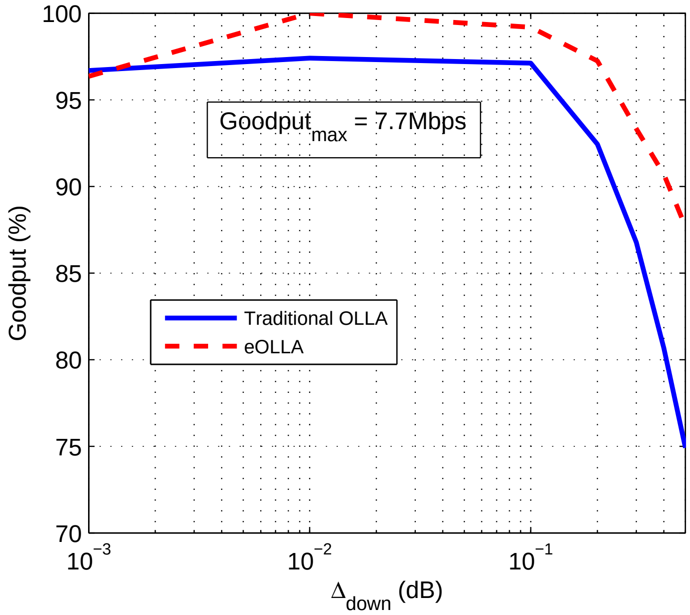

Figure 4.6: Goodput comparison for high load traffic with continuous transmission scenario

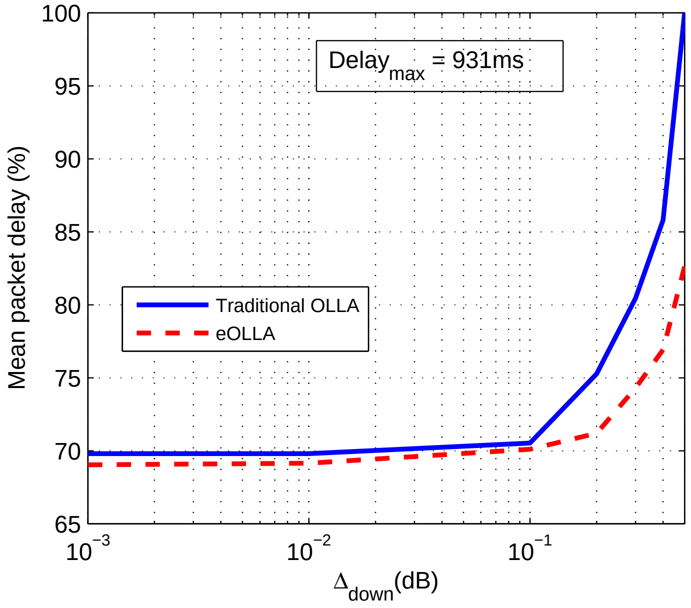

Figure 4.7: Mean packet delay comparison for high load traffic with continuous transmission scenario

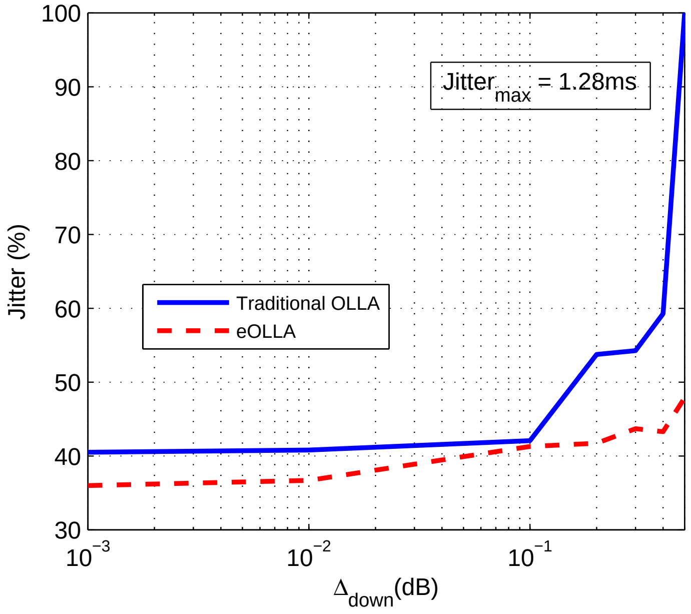

Figure 4.8: Jitter comparison for high load traffic with continuous transmission scenario

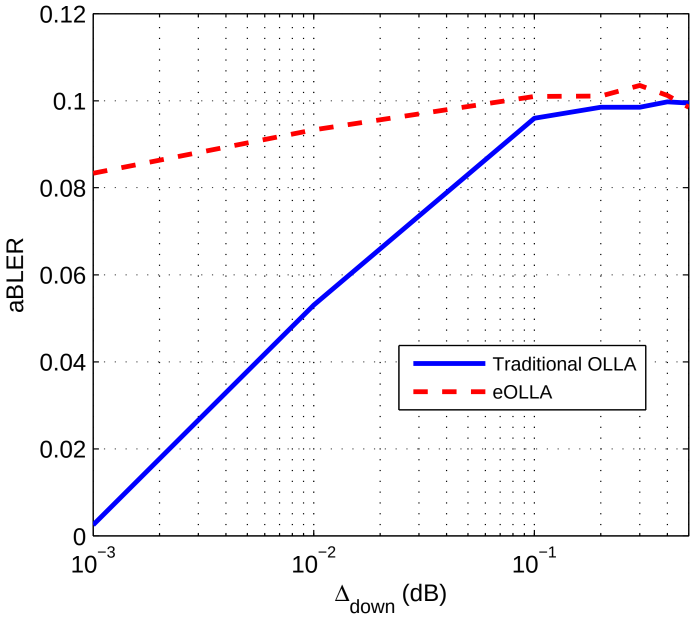

Figure 4.9: BLER comparison for high load traffic with bursty transmission scenario

Throughput, goodput, mean packet delay and jitter are expressed in percentage of the maximum value raised for both metrics, as in previous subsection.

In Fig. 4.9 it can be seen that the traditional OLLA does not raise the $aBLER_T$ for the lowest values of Δ_{Down}, due to the offset cannot be updated continuously in order to follow the channel, while the eOLLA can. As a result of that, Fig. 4.10 shows that for these lowest step sizes the traditional OLLA decreases its throughput dramatically with respect to the eOLLA, as much as a 15%. Regarding the rest of performance metrics (Fig. 4.11-4.13), the eOLLA generally outperforms the traditional OLLA while it is less influenced by the size of Δ_{down}. Only for the lowest values of Δ_{down} delay metrics are pretty similar. However, since the $aBLER_T$ is not met for the traditional OLLA, these results are not meaningful.

4.2.3 Low load traffic scenario

M2M, online gaming [51] or an Automated Teller Machine (ATM) scenario are examples of bursty traffic patterns with low data packet rates. In such scenarios, small data packets are exchanged over long periods of time, remaining the UE during these periods of time in idle mode. In LTE, there exists a minimum number of physical resources to allocate the Transport Block (TB) to be transmitted [12]. For the Downlink, this minimum depends on the system bandwidth. For

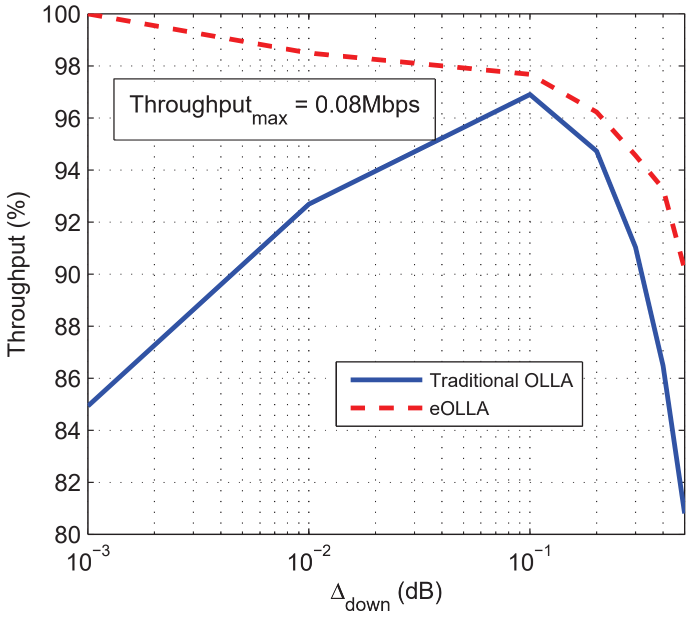

Figure 4.10: Throughput comparison for high load traffic with bursty transmission scenario

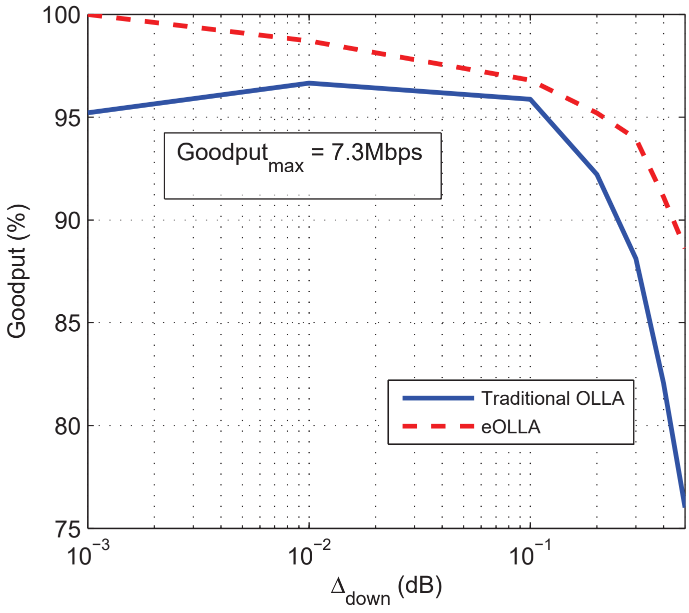

Figure 4.11: Goodput comparison for high load traffic with bursty transmission scenario

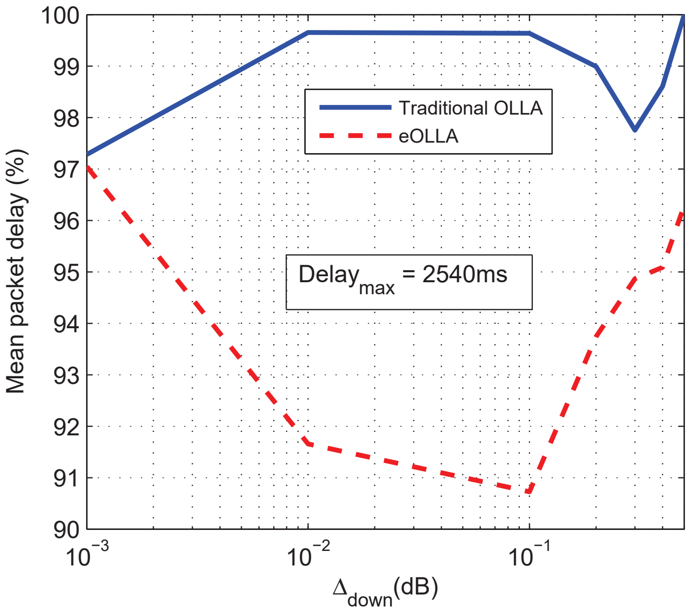

Figure 4.12: Mean packet delay comparison for high load traffic with bursty transmission scenario

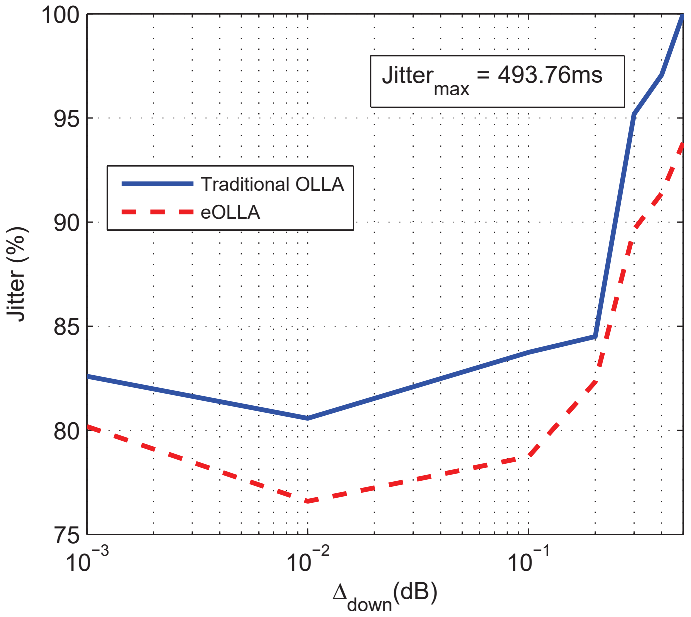

Figure 4.13: Jitter comparison for high load traffic with bursty transmission scenario

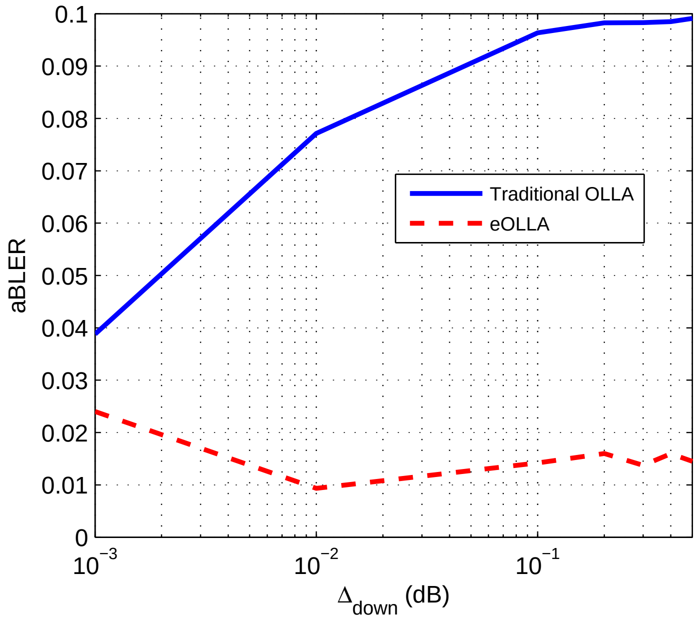

Figure 4.14: BLER comparison for low load traffic scenario

instance, for a system bandwidth of 5MHz the minimum number of physical resources corresponds to the size of 1 Physical Resource Block (PRB). Thus, in case that the size of the packets to be transmitted is smaller than the minimum number of assignable physical resources, more redundancy will be introduced in order to fulfill them, i.e. the MCS will be modified making it more robust than the proposed by the reported CQI. As a consequence, less erroneous TBs will be received and so the aBLER will be lower than the $aBLER_T$, making the traditional OLLA increase its offset in order to meet the target aBLER. Note that in this case the traditional OLLA is unnecessarily forcing the aBLER to meet the target, since decreasing the robustness of the MCS does not mean saving physical resources when the minimum number of assignable PRBs is used. Instead, it will cause erroneous received TB that could be avoided without increasing the amount of physical resource used. Therefore, if the eOLLA is used in this scenario, since its offset is not affected by the number or erroneous TBs received but by the instantaneous channel state, it will not try to force the $aBLER_T$ to be meet in a scenario in which extra redundancy is introduced.

In next figures it is presented the performance of the traditional OLLA and the eOLLA for an online gaming scenario [52], whose mean packet size is 70 bytes and its mean period between packets is 50 ms.

In Fig. 4.14 it is shown, as it was explained before, that the average BLER is not met for the eOLLA, while the traditional OLLA meets it for the majority

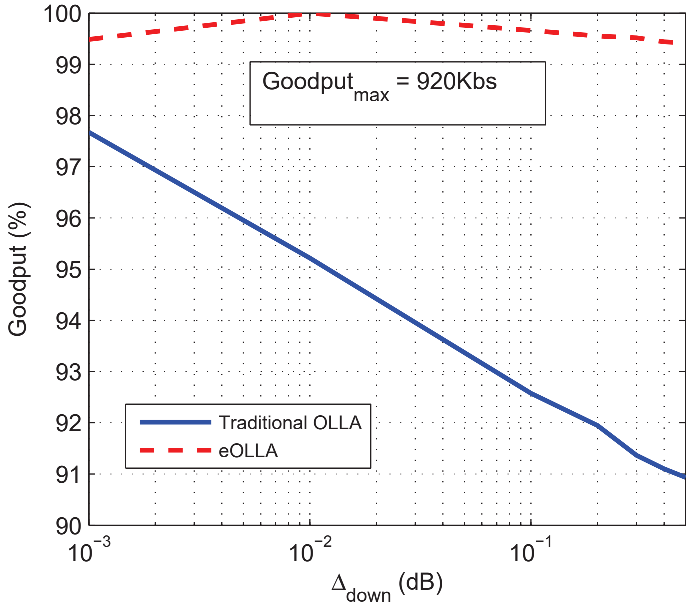

Figure 4.15: Goodput comparison for low load traffic scenario

of the Δ_{down} sizes. Regarding the throughput, it is always achieved the 100% since the capacity of the system is higher than the load of the traffic source. This fact also means that packet delay is always the minimum achievable value, that is, 1 ms. Finally, the goodput results of Fig. 4.15 show how the eOLLA significantly outperforms the traditional OLLA, and as it was explained before this improvement does not imply any extra cost in terms of physical resources usage.

4.2.4 eICIC scenario

In LTE-A it is defined the Time Domain enhanced Inter-Cell Interference Coordination (eICIC) as a technique to manage interference in Heterogeneous Networks (HetNets). An extensive description of this technique can be found in [53]. Briefly, in a scenario composed by a macro and a pico cell, a bias is applied to the coverage area of the pico cell to balance the number of users associated to both types of cells. This process is named Cell Range Expansion (CRE). Pico cell users (PUEs) that are in this CRE area suffer from very high interference from the macro cell, since its transmission power is higher than the transmission power of the pico cell. Thus, in order to improve their performance, the macro cell periodically does not schedule any transmission to their associated macro cell users (MUEs), even if they have queued packets, generating the so called Almost Blank Subframes (ABSs). During the transmission of these ABSs, the pico cell

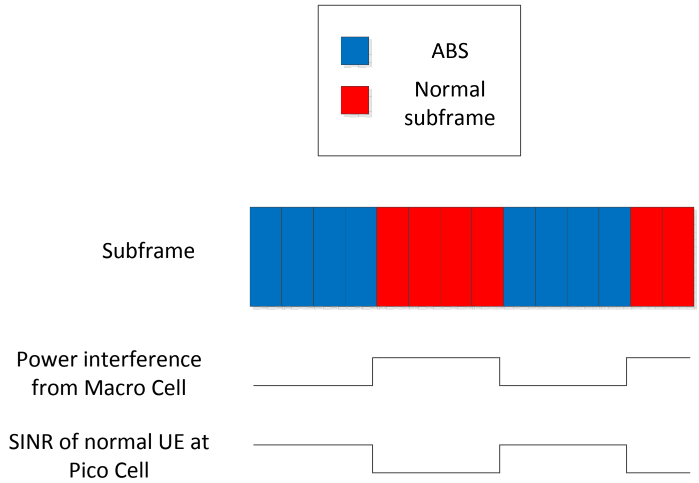
ABS
Normal subframe
Subframe
Power interference from Macro Cell
SINR of normal UE at Pico Cell

Figure 4.16: Effects of ABSs in normal PUEs

prioritizes transmission to the PUEs of the CRE area since their (Signal to Interference plus Noise Ratio) SINR conditions improve significantly. On the other hand, during the transmission of normal subframes from the macro cell, CRE PUEs are not scheduled by the pico cell.

Thus, the eICIC technique implies that MUEs and CRE PUEs are not treated over certain periods of time (see Fig. 4.16). Therefore, in a similar way as the scenario described in subsection 4.2.2, the offset of the traditional OLLA may not be able to follow the channel variations during these periods, while the eOLLA is.

Next, they are presented a performance comparison for both the traditional OLLA and the eOLLA for a MUE and 30 ABSs transmitted every 40 TTIs. Note that this scenario is a mix of the two previously evaluated high load traffic scenarios, since there is not continuous transmission, but the interval between transmission are much lower than in the case of the high load traffic with burty transmission scenario. Firstly, in Fig. 4.17 BLER for both algorithms is shown, showing that the $aBLER_T$ is met in both cases. However, for the lowest size of Δ_{down}, the eOLLA is closer to the target than the traditional OLLA (which is consistent with results of subsection 4.2.2). Furthermore, in Fig. 4.18-4.21 it is shown how for this scenario the proposed eOLLA also outperforms the traditional OLLA in terms of throughput, goodput, delay and jitter, as much as a 10%.

A summary of the maximum gain of the proposed eOLLA with respect to the traditional OLLA for the different presented scenarios is shown in Table 4.4.

To sum up, the eOLLA outperforms the traditional OLLA in the whole set of evaluated scenarios. Then, for high load traffic sources with continuous transmission, the ability of the eOLLA to adapt its step size is the cause of outperforming

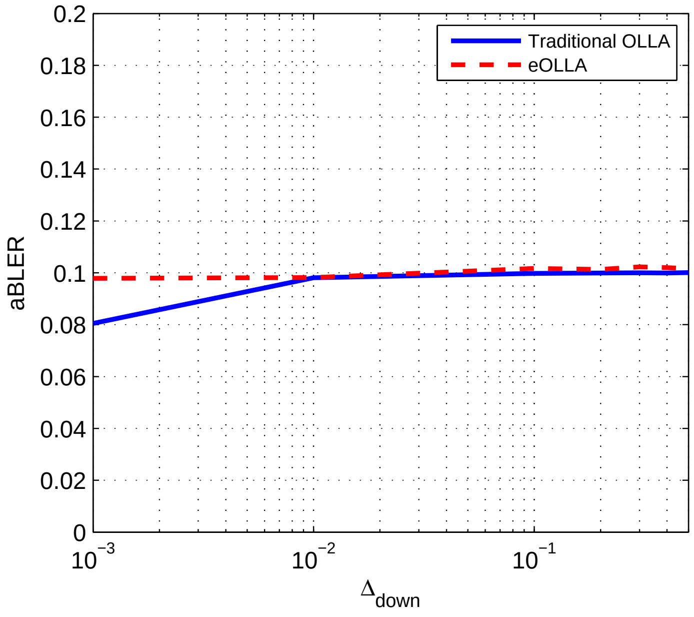

Figure 4.17: BLER comparison for eICIC scenario

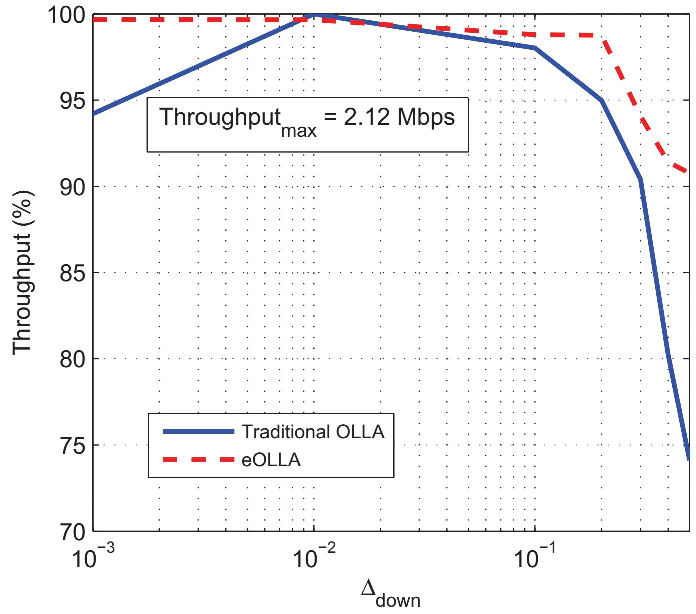

Figure 4.18: Throguhput comparison for eICIC scenario

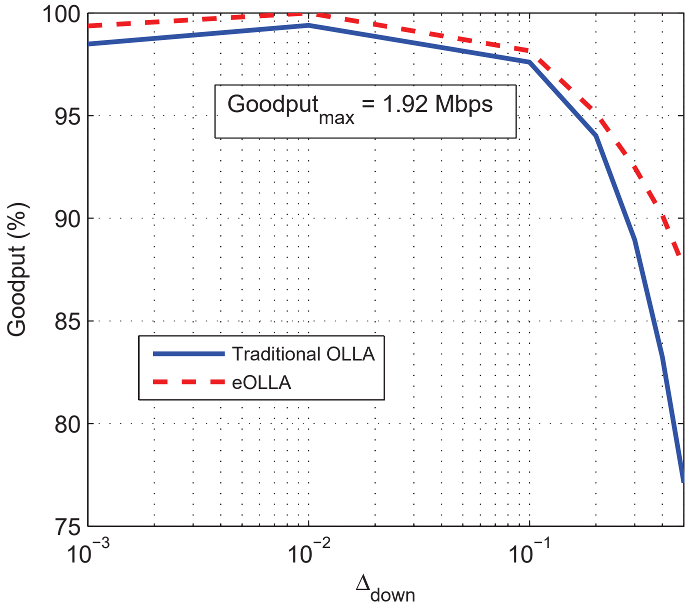

Figure 4.19: Goodput comparison for eICIC scenario

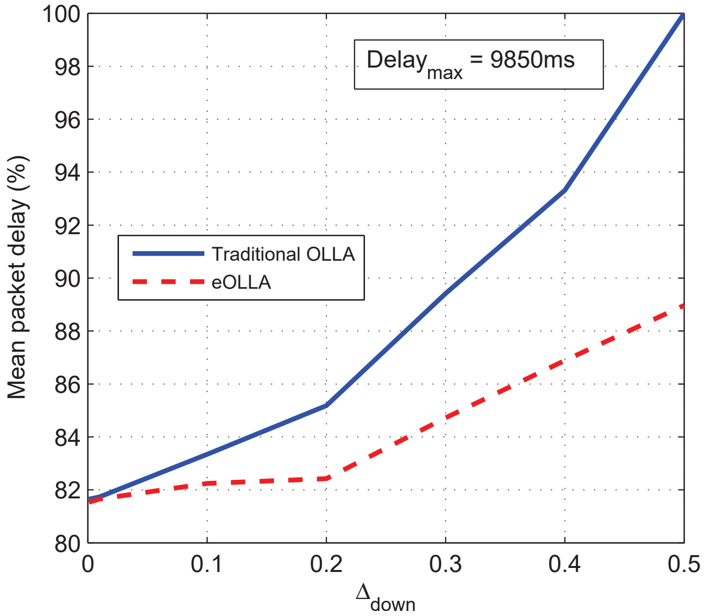

Figure 4.20: Mean packet delay comparison for eICIC scenario

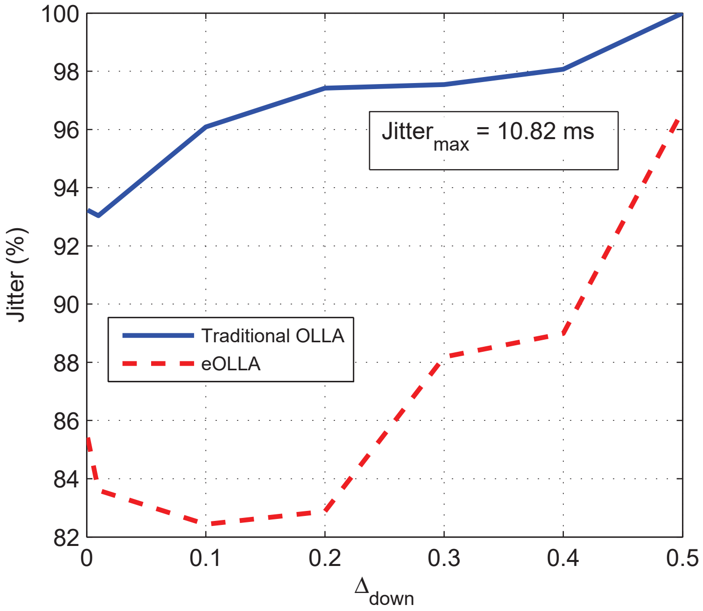

Figure 4.21: Jitter comparison for eICIC scenario

Table 4.4: Summary of eOLLA maximum gain with respect to the traditional OLLA

Subsection	Throughput gain	Goodput gain	Mean packet delay gain	Jitter gain
High traffic load with continuous transmission scenario	15%	17%	17%	52%
High load traffic with bursty transmission scenario	15%	15%	8%	7%
Low load traffic scenario	NA	8%	NA	NA
eICIC scenario	17%	10%	11%	14%

the traditional OLLA for the highest Δ_{down}. Regarding the bursty traffic sources, the ability of the eOLLA to follow the channel variations even if no packet is received is the key factor. Finally, for the eICIC scenario, a combination of both abilities are exploited. Regarding the size of Δ_{down}, previous results show that while for the traditional OLLA the selection of an appropriate value (which is different depending on the scenario) is crucial, for the eOLLA there is a wide range of values that have a similar performance.

Chapter 5

Constant Power Optimization of the AMC Process

5.1 Introduction

The Outer Loop Link Adaptation (OLLA) has been studied in chapters 3 and 4 as a technique able to modify the Signal To Noise Ratio (SNR) thresholds in order to meet a target BLock Error Rate (BLER) under the Adaptive Modulation and Coding (AMC) operation. However, it has not been studied whether OLLA can provide more than one set of SNR thresholds that meet the target BLER, and in that case, which of those sets maximizes the data rate.

The problem of selecting appropriate values for the SNR thresholds to optimize the data rate subject to certain constraints has been widely addressed in the literature from a theoretical point of view [13, 54–58]. Most referenced Goldsmith's works [14] [9] address the optimization problem as constrained by an average Bit Error Rate (BER) while allocated power can take any value and only its average is constrained; SNR thresholds are initially selected as those able to keep the instantaneous BER under a target value and later adjusted either individually [14] or keeping the relative distance among them [9]. Channel coding, when treated in the literature, is simply included as certain gain in SNR compared to the uncoded Quadrature Amplitude Modulation (QAM). None of these works addressed the previous mentioned optimization problem with channel coding and BLock Error Rate (BLER) target as standardized in LTE [12].

The aim of this chapter is to cover this gap. Thus, a model of the AMC process for the downlink of LTE is used, including the channel coding feature, to carry out an analytic optimization. This optimization consists in finding the SNR thresholds that maximize the throughput with a target BLER (as defined in LTE) constraint. Constant power has been considered as the LTE downlink has little margin for operation to dynamically modify the transmission power [12]. Finally, the results of this analytic optimization will be compared with the traditional OLLA performance.

5.2 AMC modeling extension

In section 3.1 the AMC process was modeled for an uncorrelated Rayleigh channel. However, only BLER metrics were taken into account. Then, in this section throughput metrics are also included in the model. In equation (3.1) the average BLER for an uncorrelated Rayleigh channel was expressed as

$$aBLER(\Gamma, \{\Psi_i\}) = \sum_{i=0}^{n-1} \int_{\Psi_i}^{\Psi_{i+1}} iBLER_i^{AWGN}(\gamma) p^o(\Gamma, \gamma) d\gamma,$$

where binary logistic functions were used to model $iBLER_i^{AWGN}(\gamma)$, i.e. the instantaneous BLER for Additive White Gaussian Noise (AWGN) channels when turbo codes are used (see 3.1); and the instantaneous SNR conditioned to transmission $p^o(\Gamma, \gamma)$ being defined in (3.3) as

$$p^o(\Gamma, \gamma) = \begin{cases} \frac{1}{A(\Gamma,\Psi_0)\Gamma} e^{-\gamma/\Gamma}, \gamma > \Psi_0 \\ \\ 0, \text{else} \end{cases}$$

where $A(\Gamma, \Psi_0)$ is the probability of not being in outage, that was defined in equation (3.4) as:

$$A(\Psi_0) = \int_{\gamma=\Psi_0}^{\infty} \frac{1}{\Gamma} e^{-\gamma/\Gamma} d\gamma = e^{-\Psi_0/\Gamma}.$$

The achieved throughput r for a given set of SNR thresholds $\{\Psi_i\}_{i=0,1,..,n}$ is:

$$r(\Gamma, \{\Psi_i\}) = \sum_{i=0}^{n-1} R_i \cdot p_{\Psi_i}, \tag{5.1}$$

being R_i the transmitted bits/symbol for each fadin region i, p_{Ψ_i} the probability of the instantaneous SNR falling within the interval $[\Psi_i, \Psi_{i+1})$, that is,

$$p_{\Psi_i} = e^{-\Psi_i/\Gamma} - e^{-\Psi_{i+1}/\Gamma}. \tag{5.2}$$

For a given average SNR value Γ, there are many combinations of thresholds that meet the average target BLER ($aBLER_T$). The set of possible solutions,

$S^{ave} = \{\{\Psi_i\}, aBLER(\Gamma, \{\Psi_i\}) \leq aBLER_T\}$ is in fact infinite, but only one of its elements is optimal in terms of spectral efficiency. Therefore, a careful selection of these Ψ_i values is necessary in order to maximize the data rate. As addressed in previous Goldsmith's works [14][9], where a similar optimization was carried out but without constant power constraint, this maximization can be carried out by: a) adjusting each SNR threshold value individually [14] or b) forcing the relative distance among them to keep fixed [9], which is a sub-optimal solution but less complex to obtain. Next, these two procedures are carried out for our problem.

5.2.1 Optimal Maximization

The presented data rate (or spectral efficiency) maximization problem $r(\Gamma, \{\Psi_i\})$ with average BLER constraint can be formalized as

$$\max_{\Psi_i|_{i=0,1,..,n-1}} \{r\,(\Gamma, \{\Psi_i\})\} \quad s.t.\ aBLER\,(\Gamma, \{\Psi_i\}) \leq aBLER_T \tag{5.3}$$

The optimal thresholds that maximize the spectral efficiency, subject to the aBLER constraint, can be derived by using the Lagrange method. Thus, the Lagrange equation is

$$F\,(\Gamma, \{\Psi_i\}, \lambda) = r\,(\Gamma, \{\Psi_i\}) + \lambda \cdot (aBLER\,(\Gamma, \{\Psi_i\}) - aBLER_T) \tag{5.4}$$

In order to solve this optimization problem, the following constraints have to be taken into account:

- $\Gamma \in \Re^+$. The average SNR has to be a positive real value.
- $R_i \in \Re^+,\ R_i < R_{i+1}$. Data rates have to be a positive real value. Furthermore, data rates are sorted in ascending order.
- $\Psi_i \in \Re^+,\ \Psi_n = \infty$. SNR thresholds have to be positive real values. Furthermore, the last SNR interval is infinite.
- $\Psi_i < \Psi_{i+1}$. SNR thresholds are sorted in ascending order.

Regarding the binary logistic functions used to model the instantaneous BLER (see section 3.1), the shape of this function area is shown in Fig. 5.1. Note that this function is monotonically decreasing.

Thus, the following constraints are applied to the binary logistic functions:

- $\frac{1}{1+e^{-\alpha_{i_0}\gamma-\alpha_{i_1}}} < \frac{1}{1+e^{-\alpha_{(i+1)_0}\gamma-\alpha_{(i+1)_1}}} \Rightarrow f_i\,(\gamma) < f_{i+1}\,(\gamma)$

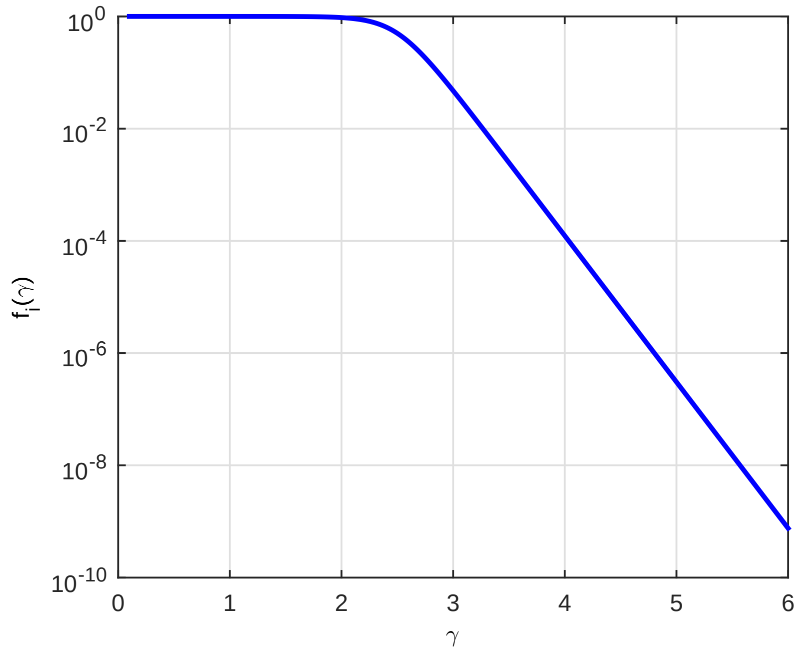

Figure 5.1: Example of the shape of a binary logistic function

- $\alpha_{i_0} \in \Re^{+}$
- $\alpha_{i_1} \in \Re^{-}$

Then, the expression within each of the integrals that compounds the average BLER function $aBLER\left(\Gamma,\{\Psi_i\}\right)=\sum\limits_{i=0}^{n-1}\int_{\Psi_i}^{\Psi_{i+1}} iBLER_i^{AWGN}\left(\gamma\right)\cdot p^{o}\left(\Gamma,\gamma\right)d\gamma$, which defines the constraint of the optimization problem, also are monotonically decreasing. This can be deduced since $iBLER_i^{AWGN}\left(\gamma\right)$ is monotonically decreasing, as it was previously said, and it is multiplied by a exponential function with a negative exponent, as shown in equation (3.3). The shape of this expression is shown in Fig. 5.2.

Therefore, the constraint consists in the sum of the areas included under curves $iBLER_i^{AWGN}\left(\gamma\right)\cdot p^{o}\left(\Gamma,\gamma\right)$ weighted by a positive constant $1/A\left(\Psi_0\right)=e^{-\Psi_0/\Gamma}$, between the SNR thresholds Ψ_i and Ψ_{i+1}. In Fig. 5.3 it is shown an example (when $n=2$) of the area to be calculated (gross line).

In order to solve the problem, the following closed form of a integral is found:

$$\begin{gathered}\int iBLER_i^{AWGN}\left(\gamma\right)\cdot p^{o}\left(\Gamma,\gamma\right)d\gamma=\\ -e^{-\frac{\gamma}{\Gamma}}\left(1+\frac{1}{\alpha_{i_0}\Gamma-1}\cdot\left(\frac{1+e^{\alpha_{i_0}\gamma+\alpha_{i1}}}{1+e^{-\alpha_{i_0}\gamma-\alpha_{i1}}}\right)\cdot{}_2F_1\left(1,-\frac{1}{\alpha_{i_0}\Gamma};1-\frac{1}{\alpha_{i_0}\Gamma};-e^{\alpha_{i_0}\gamma+\alpha_{i1}}\right)\right)\end{gathered}\tag{5.5}$$

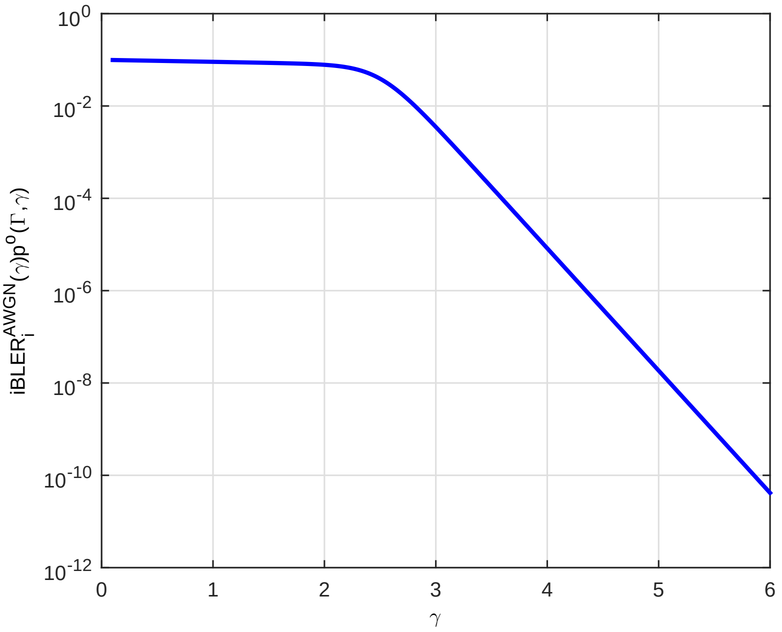

Figure 5.2: Example of the shape of a binary logistic function multiply by the instantaneous SNR conditioned to transmission

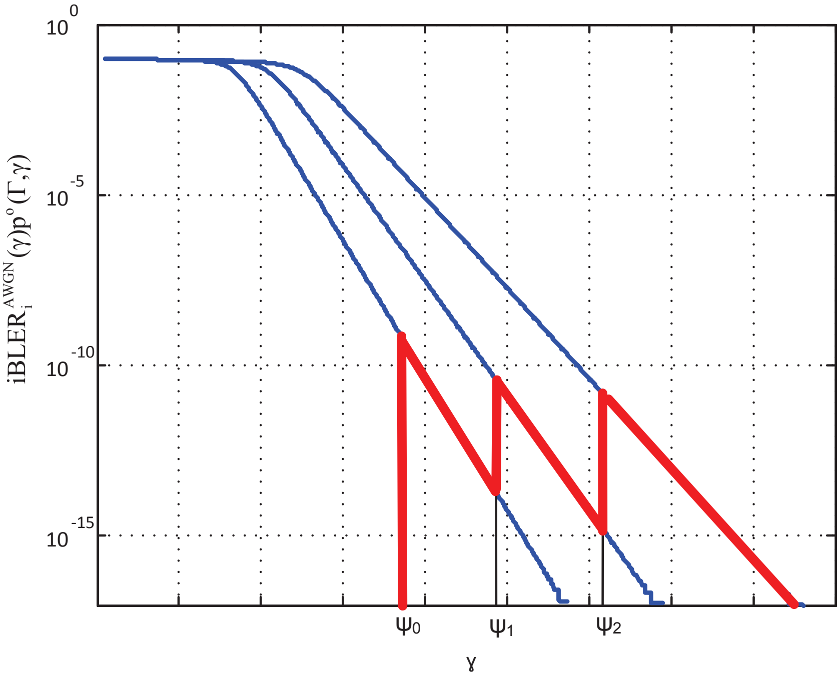

Figure 5.3: Example of area to be calculated for the optimization problem constraint

being ${}_2F_1$ the hypergeometric function:

$$ {}_pF_q(a_1, ..., a_p; b_1, ..., b_q; z) = \sum_{n=0}^{\infty} \frac{(a_1)_n (a_2)_n ... (a_p)_n}{(b_1)_n (b_2)_n ... (b_q)_n} \frac{z^n}{n!} \tag{5.6} $$

with $(a)_n = a\,(a+1)\,(a+2)\,...\,(a+n-1)$.

The optimal thresholds should satisfy that

$$ \left. \frac{\partial F\left(\Gamma, \{\Psi_i\}, \lambda\right)}{\partial \Psi_i} \right|_{i=0,1,..,n-1} = 0 \quad \text{and} \quad \frac{\partial F\left(\Gamma, \{\Psi_i\}, \lambda\right)}{\partial \lambda} = 0 \tag{5.7} $$

i.e., the values of the set S^{ave} which maximizes throughput. According to (3.1)-(3.5), (5.5) and (5.1), the resulting system of equations for a number of thresholds n is given by (5.8). Numerical methods has been used in the following to find the solution to this problem.

$$ \begin{cases} \frac{\partial F(\{\Psi_i\},\lambda)}{\partial \Psi_0} = -R_0 e^{\left(-\frac{\Psi_0}{\Gamma}\right)} - \lambda f_0\left(\Psi_0\right) + \lambda aBLER_T = 0 \\ \left.\frac{\partial F(\{\Psi_i\},\lambda)}{\partial \Psi_i}\right|_{i=1,..,n-1} = \left(R_i - R_{i-1}\right) + \lambda e^{\left(\frac{\Psi_0}{\Gamma}\right)} \left(f_{i-1}\left(\Psi_i\right) - f_i\left(\Psi_i\right)\right) = 0 \\ \frac{\partial F(\{\Psi_i\},\lambda)}{\partial \lambda} = -e^{\left(\frac{\Psi_0}{\Gamma}\right)} \sum_{i=0}^{n-1} \left[e^{\left(-\frac{\gamma}{\Gamma}\right)} {}_2F_1\left(1, -\frac{1}{a_i\Gamma}; 1 - \frac{1}{a_i\Gamma}; -e^{(a_i\gamma + b_i)}\right) \right]_{\gamma=\Psi_i}^{\gamma=\Psi_{i+1}} \\ -aBLER_T = 0 \end{cases} \tag{5.8} $$

5.2.2 Sub-Optimal Maximization

In the sub-optimal maximization, relative distances among thresholds are forced to keep fixed to those of a predesigned solution set $\{\gamma_i\}$. Therefore, the aimed thresholds can be rewritten as $\Psi_i = \gamma_i \cdot \theta$, where γ_i is the initial value assigned to the threshold i, and θ is a common offset that modifies these initial values. Note that the set $\{\gamma_i\}$ fixes the relative distance among thresholds. Hence, the resulting problem is to find the optimal value of θ that maximizes the data rate without raising the $aBLER_T$, which can be formulated as

$$ \max_{\theta} \left\{ r\left(\Gamma, \{\gamma_i\}, \theta\right) \right\} \quad s.t.\ aBLER\left(\Gamma, \{\gamma_i\}, \theta\right) \leq aBLER_T \tag{5.9} $$

with

$$ aBLER(\Gamma, \{\gamma_i\}, \theta) = \sum_{i=0}^{n-1} \int_{\gamma_i\theta}^{\gamma_{i+1}\theta} iBLER_i^{AWGN}(\gamma) p(\gamma) d\gamma. \tag{5.10} $$

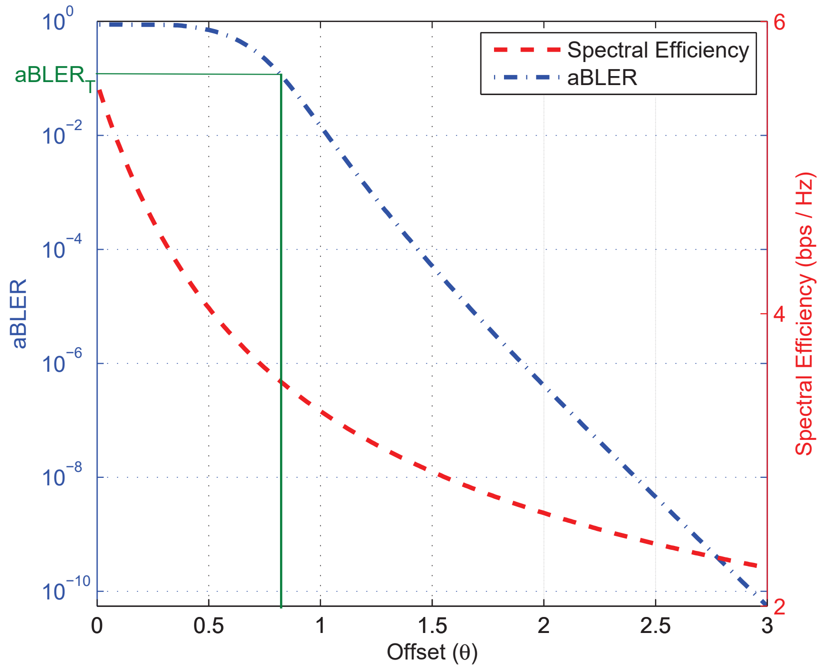

Figure 5.4: BLER vs. Spectral efficiency for different θ values

The optimum value of θ can be easily deduced. Fig. 5.4 shows the average BLER and the spectral efficiency vs. θ value for a certain average SNR Γ= 20dB and the same SOVA turbo decoder fitted in Fig. 3.2. The set of 15 initial thresholds $\gamma_i|_{i=0..n-1}$ was selected by forcing the instantaneous BLER to be below the target $iBLER \leq aBLER_T = 0.1$. As the aBLER and the spectral efficiency decrease monotonically as θ increases, there is a unique value of θ (0.78 for this specific set of values) that achieves the maximum spectral efficiency (3.4 bps/Hz): the only one that exactly meets $aBLER_T$. Thus, the optimum θ value can be simply obtained by solving

$$aBLER\,(\Gamma, \{\gamma_i\}, \theta) = aBLER_T. \tag{5.11}$$

It should be noticed that equation (5.10) was already presented in section 3.2, when traditional OLLA was described. Therefore, it can be deduced that the performance of the traditional OLLA presented int chapter 3 is equivalent in average to the sub-optimal AMC optimization previously presented, when the same initial values γ_i are used, since both procedures search for an offset value θ that meets the average target BLER. Therefore, traditional OLLA carries out a sub-optimal maximization of the AMC.

Finally, the dynamics of the traditional OLLA, which was already presented in chapter 3, is shown again in Fig. 5.5, but now introducing throughput results in addition to the aBLER results. This figure shows the evolution of the offset $\theta[k]$ obtained from simulation using the same conditions as in Fig. 5.4. Note that although its average value converges to the theoretical value previously calculated, i.e. 0.78, $\theta[k]$ fluctuates due to the continuous addition and subtraction of Δ_{up} or Δ_{down}, respectively. This fact was deeply explained in chapter 3. Furthermore, the spectral efficiency tends to the same value in both figures.

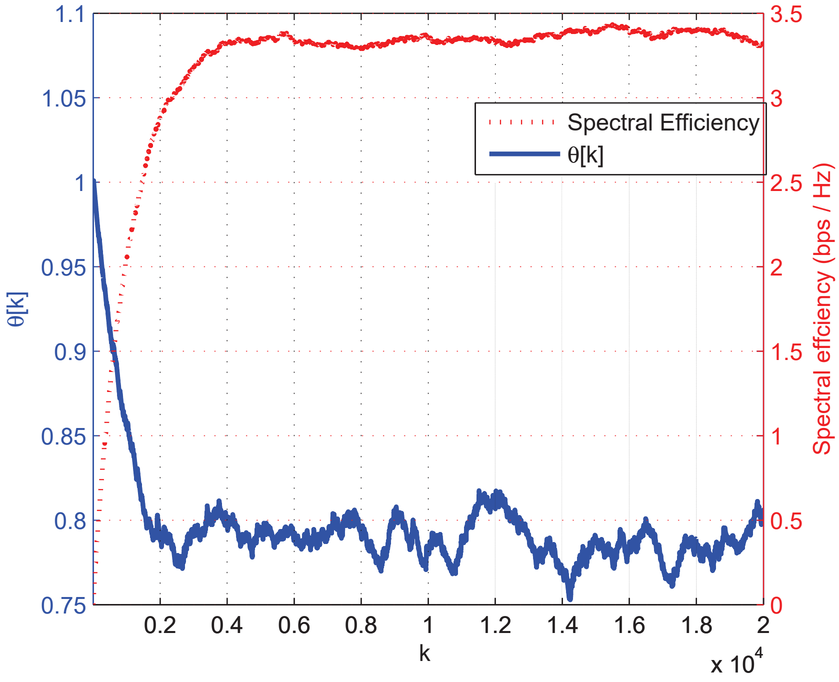

Figure 5.5: Dynamic of the single offset OLLA

5.3 Multi-Offset OLLA

So far, the OLLA technique has been studied when it is implemented by using a single offset to modify all the thresholds simultaneously [24][25]. However, it would also be possible to implement the OLLA technique by using several offsets separately [59]. Then, the multi-offset OLLA implementation is carried out by adjusting each SNR threshold Ψ_i through a different offset θ_i, that is $\Psi_i[k] = \gamma_i \cdot \theta_i[k]$, where γ_i is its initial value. In this case, only one of the n SNR thresholds is updated at each instant of time k, which is the threshold Ψ_i associated with the MCS i of the current received coded block. Hence, according to previous subsection, the multi-offset OLLA algorithm works as follows:

$$\begin{cases} \theta_{i_{dB}}[k] = \theta_{i_{dB}}[k-1] + \Delta_{up} \cdot e[k] - \Delta_{down} \cdot (1 - e[k]) \\ \theta_{j_{dB}}[k] = \theta_{j_{dB}}[k-1] \;\; \forall j = 0..n-1; \;\; j \neq i \end{cases} \tag{5.12}$$

being $\theta_{i_{dB}}[k] = 10 \cdot log_{10}(\theta_i[k])$.

Suitable values of Δ_{up} and Δ_{down} must also satisfy equation (3.8) so that each offset $\theta_i[k]$ converges to a value θ_i that ensures the $aBLER_T$.

The main difference between using one or multiple offsets is the increase of the degrees of freedom: there are multiple combinations of offset values that lead to the convergence of the aBLER to the target. However, not all of these combinations might lead to a maximal spectral efficiency. This fact is shown in Fig. 5.6 where, for the sake of simplicity, only the outage condition and MCSs corresponding to CQIs 4 and 5 [39] are considered. This figure shows the set of offset combinations (θ_0, θ_1) that leads to $aBLER_T = 0.1$, and the spectral efficiency achieved for each combination. These results are obtained from (3.1) and (5.1) by replacing $\Psi_i = \gamma_i \cdot \theta_i$, being γ_i those values that make $iBLER < aBLER_T$. Firstly, it can be seen that these offset combinations are a connected set easily shown to be closed. Then, it is also noticeable that there is one combination for which the spectral efficiency is maximum.

The offset values that maximize the spectral efficiency restricted to an average target BLER can be directly obtained from the maximization problem of section 5.2.1 by doing $\Psi_i = \gamma_i \cdot \theta_i$. Then, optimal offset values can be found through (5.8), being $\theta_0 = 0.707$ and $\theta_1 = 1.14$ in this particular case.

Finally, these offset combinations analytically calculated have been validated with the LTE downlink simulator [36] previously presented in section A with a multi-offset OLLA implementation (solid line in Fig. 5.6). As it can be seen, OLLA wanders over the majority of the analytic solutions but without converging at any point. Moreover, its throughput will be a kind of averaging of all possible values but lower than the maximum.

In conclusion, although the set of SNR thresholds that leads to maximum throughput restricted to a target aBLER is within the possible combinations achieved by the multi-offset OLLA implementation, this technique as it is currently defined cannot ensure the convergence to the optimum solution.

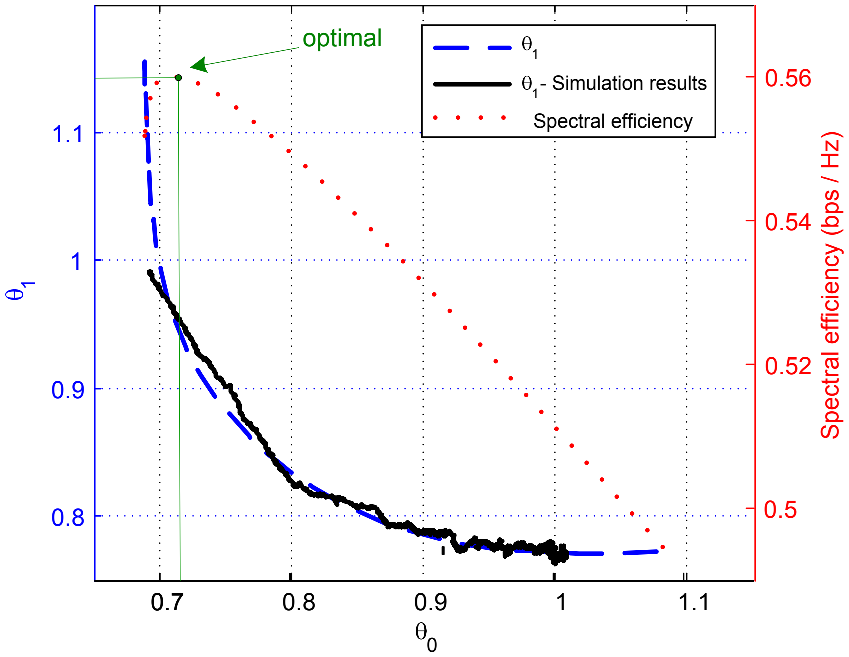

Figure 5.6: Offset combinations and their corresponding spectral efficiency

5.4 Effects of non-idealities in the AMC

As it was discussed in previous sections, the difference between the optimal and the sub-optimal AMC process is that the second one keeps the relative distance among SNR threshold constant while for the optimal solution these distances may change during the optimization process. Therefore, these relative distances among SNR thresholds are the key of the optimization process.

Thus, the sub-optimal and the optimal solution will coincide if the relative distance among SNR thresholds is chosen for the sub-optimal process so that they coincide with the relative distances that are achieved by the optimal solution. However, to carry out this selection appropriately is very dificult since in a realistic implementation of a LTE downlink system, there are several features that can have a significant impact on the AMC performance, such as imperfections in the channel estimation method, the turbo decoder performance or the channel behavior, among others. Thus, it is possible that the ideal set $\{\gamma_i\}$ (i.e. the relative distance among thresholds) does not adjust to that one to be used in a realistic implementation. Therefore, these non-ideal features may be modeled as deviations in ideal initial thresholds.

This section presents the effect of non-idealities in the optimal and sub-optimal AMC performance (see subsections 5.2.1 and 5.2.2, respectively). To facilitate the analysis, each SNR threshold Ψ_i is separated in an initial value γ_i modified by an offset, following the methodology of subsection 5.2.2. Thus, there will be a single offset θ, whose value can be obtained from (5.11), when sub-optimal AMC; and a set of n offsets $\{\theta_i\}_{i=0,1,..n}$, whose values are given by (5.8), when optimal AMC. As it was mentioned before, initial relative distances among thresholds are defined by the set $\{\gamma_i\}_{i=0,1,..n}$.

For the sake of simplicity, only outage and the MCSs corresponding to QPSK modulation with different coding schemes in LTE are considered, i.e. those corresponding from CQI 1 to 6 [39]. Then, from (3.5) we can get the values of a_i and b_i that better fit the iBLER as well as ideal initial γ_i values that make $iBLER < aBLER_T$. Parameter values used for evaluating the AMC are summarized in Table 6.2.

Table 5.1: AMC parameter values

Parameter	Value
$[a_1, a_2, a_3, a_4, a_5, a_6]$	[26.18, 18.65, 14.96, 9.61, 7.75, 6.00]
$[b_1, b_2, b_3, b_4, b_5, b_6]$	[-8.98, -10.50, -13.02, -14.20, -16.52, -15.62]
$[\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6]$	[-3.65dB, -1.6dB, 0dB, 2.25dB, 3.75dB, 4.75dB]
$aBLER_T$	0.1
Γ	0-10dB

Firstly, AMC performance for ideal initial γ_i thresholds is shown in Fig. 5.7. According to this figure, there are not significant differences between optimal and sub-optimal AMC when ideal conditions can be assumed.

Fig. 5.8 shows the performance of both optimal and sub-optimal AMC process when adding different realizations of an uniform noise $n_i \in [-0.5dB, 0.5dB]$ to each γ_i value of Table 6.2. According to these results, forcing the relative distance among thresholds to keep fixed might lead to a degradation of the spectral efficiency, which in some realizations can achieve the 20%. On the other hand, adjusting each SNR threshold value individually leads to the optimal spectral efficiency for all the realizations, i.e. becomes the AMC optimization process independent of the modification of any feature such as described before.

It should be noticed that previous optimization results when keeping fixed the relative distance among thresholds are equivalent to the results that would be obtained by the AMC supported by a single-offset OLLA; however, results for the AMC optimization when adjusting each threshold individually might not match the results of the AMC supported by multiple offsets, since this OLLA implementation does not converge to a single solution but wanders around the solutions that achieve the average target BLER.

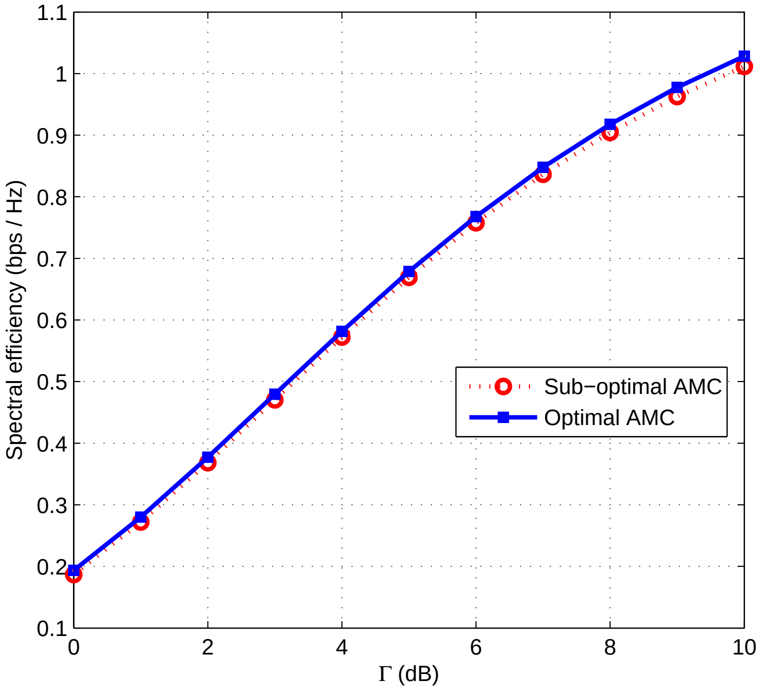

Figure 5.7: Performance comparison between optimal and sub-optimal AMC in ideal conditions

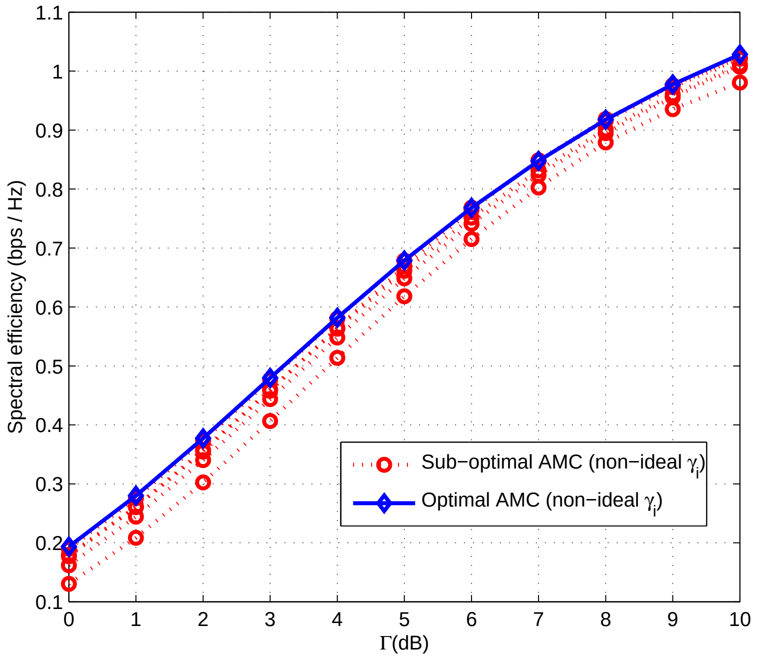

Figure 5.8: Performance comparison between optimal and sub-optimal AMC in non-ideal conditions

Chapter 6

Closed Loop Link Adaptation

In a realistic implementation of an LTE downlink system, there are many aspects that have a significant impact on the Adaptive Modulation and Coding (AMC) performance, such as imperfections in the channel estimation or in the Signal to Noise Ratio (SNR) estimation method, the channel behavior or the accuracy of the selection of the AMC SNR thresolds, among others. In fact, the implementation of some parts of the User Equipment (UE) is not specified by the standard, and as consequence each manufacturer selects each own.

One critical part of the UE that affects the AMC performance is the channel turbo decoder. In LTE, implementation of the turbo decoder is not standardized, so there are many architectures proposed in the literature. These architectures might have different performance depending on features such as the accuracy of the computation of the Log-Likelihood Ratio (LLR) of the received coded bits, the channel behavior or the mobile speed. More details about channel coding can be found in chapter 2. Thus, as each UE can have a different turbo decoder implementation, the AMC performance can vary between UEs.

It is well-known that the Bit Error Rate (BER) of a turbo decoder improves as the size of the block to be decoded increases [60]. On the other hand, for a fixed BER value, the BLock Error Rate (BLER) worsens as the size of the coded block increases [3]. In standards like 3GPP-LTE, BLER is a more useful statistic as it is used to perform the AMC. In the consulted literature the relationship between BER and BLER for a turbo decoder as the coded block sizes increases is not clear, and neither is the relationship between BLER and the coded block size. In addition to that, [61] shows that the segmentation process of Transport Blocks (TBs) carried out in LTE [31] can imply a degradation in terms of BLER when these TBs exceed a certain length. Therefore, the TB size has a significant influence in the BLER performance, and by extension, in the AMC performance.

In order to improve the performance of the AMC due to non-ideal features of the system an Outer Loop Link Adaptation (OLLA) is often used. This technique has been deeply studied in previous chapters. To sum up, the OLLA typically modifies the estimated SNR to correct deviations in the Channel Quality Indicator (CQI) selection and therefore to maintain the BLER as close as possible to the target [25].

OLLA can also correct the impact of the TB size on the AMC performance. However, in an LTE downlink there might be a wide variation in TB sizes of successive transmissions, which could result in the lack of convergence of the OLLA, and in a degradation in the AMC performance. Due to the size of the next TB to be transmitted is unknown at the UE, as it depends on features such as traffic sources or the scheduling algorithm, an adaptation of the OLLA technique with the TB size is not possible. The reason is that OLLA will adapt its offset value after the transmission of a TB with a certain size, and this offset will affect to the Modulation and Coding Scheme (MCS) to be used during the transmission of the next TB, which might have a different size than the prior one, so the offset value might not be the more suitable.

In this chapter a Closed Loop Link Adaptation (CLLA) is proposed for the downlink of LTE to improve the performance of the AMC by reducing the impact of the variability of the TB sizes. The CLLA technique has to be located at the BS, and it is independent from how the CQI selection process is implemented at the UE. Thus, the Base Station (BS) can correct the imperfect CQI selection process of a UE for a specific link scenario. Furthermore, this proposal does not imply any modification of the 3GPP-LTE standard.

6.1 Impact of the TB size on the turbo decoder performance

In Fig. 6.1 an example of the BLER performance in Additive White Gaussian Noise (AWGN) conditions for a turbo decoder based on the Soft Output Viterbi Algorithm (SOVA) architecture of [4] is shown, for a fixed MCS corresponding to the CQI index 12 and different TB sizes (see Appendix A for further details about the simulation environment). These kinds of curves are used to build the lookup tables that map the Link Quality Metric (LQM) into a CQI value. As it can be seen, different BLER results are achieved for different TB sizes. In particular, for the implemented architecture, BLER results worsen as the TB size increases. The BLER degradation for a TB size of 4490 bits is only due to the increase of the coded block size. However, for higher TB sizes a segmentation process is carried out, which will be another cause of the degradation. As the size of the next TB to be transmitted is unknown by the UE, choosing the exact BLER curve is not possible, which might provoke the inefficiency of the CQI selection process.

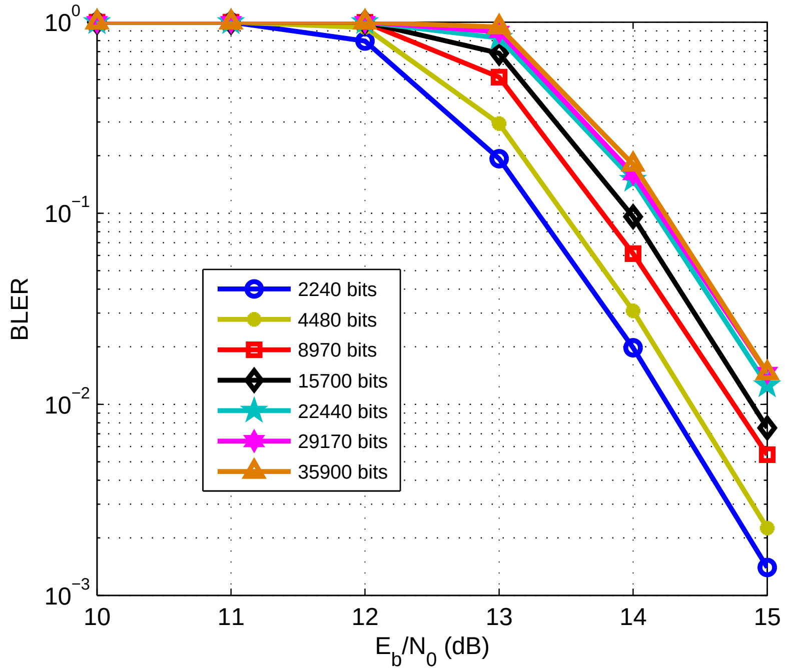

Figure 6.1: BLER results for different TB sizes and an MCS corresponding to the CQI index 12 for an AWGN channel

In Fig. 6.2 BLER results for the same fixed CQI index are presented, in this case for the Extended Pedestrian A (EPA) channel [37]. These results show the performance of the turbo decoder in the downlink of LTE for a realistic channel model. As for the AWGN channel model, BLER results worsen as the TB size increases, but for the same average SNR the results are worse than those of AWGN channel due to frequency selectivity. Thus, for a given MCS suggested by a CQI index, the target BLER might not be ensured depending on the size of the transmitted TBs.

The impact of the allocated TB size on the AMC performance can be corrected by using the OLLA technique previously described. However, in the LTE downlink, the TB size depends on several factors such as the channel state, the available number of PRBs, the traffic source pattern or the scheduling algorithm. As the UE has no information about the size of the next TB to be transmitted, an adjustment in the OLLA taking into account the transmitted TB size is not possible.

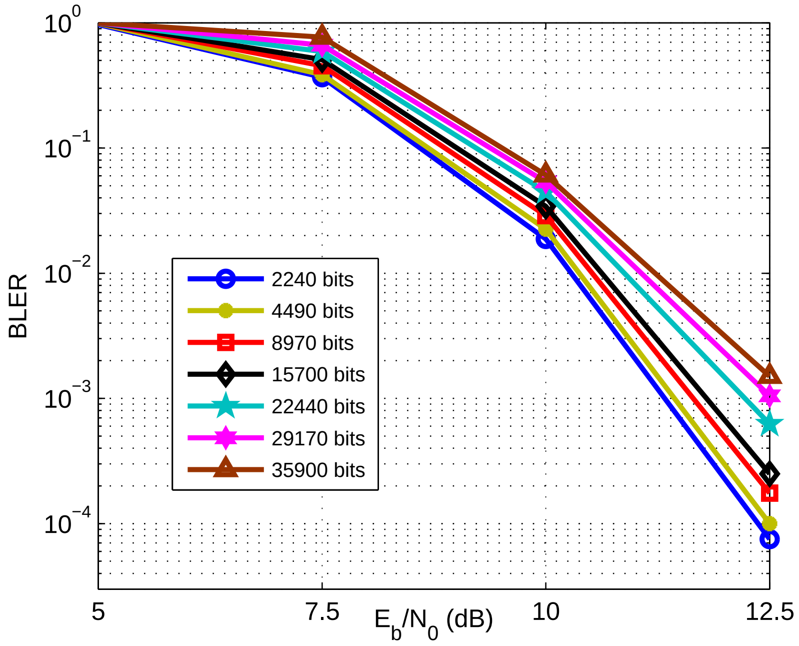

Figure 6.2: BLER results for different TB sizes and an MCS corresponding to the CQI index 12 for an EPA channel

6.2 Selection of the TB size in the LTE downlink

In section 3.1 the AMC process for the downlink of LTE was deeply described. However, it was not addressed how the TB size is selected. In this section this procedure is described for a transmission of the LTE downlink. From this description, it can be figured out which factors can influence it selection.

In [12] the full process of the TB size selection is described. Next the steps of this process are described:

1. *Selecting the set of Physical Resource Blocks to be used for transmission:* The BS has to decide for each associated UE, the number of Physical Resource Blocks (PRBs) to be used in the next subframe to be transmitted[1]. This decision is not standardized, so it can be done based on priority, traffic metrics or channel quality, among others. It also has to be taken into account that the minimum number of assignable PRBs depends on the transmission bandwidth. For instance, for a bandwidth of 20MHz, the number of assigned PRBs per slot has to be a multiple of 4.

[1] Actually, a subframe is made up of two time slots, and the same number of PRBs has to be used in both slots for a transmission. Furthermore, these PRBs have to be allocated in the same frequencies (except for certain transmission modes).

2. *Selecting a single CQI to be applied to the selected set of PRBs:* The UE reports a CQI value not for each PRB individually but for a set of PRBs, i.e. for a subband. As in the case of the minimum assignable number of PRBs, the size of the subband for which a single value of CQI is reported depends on the bandwidth transmission. It may happen that those PRBs making up the assigned bandwidth have different CQI values. Therefore, it is necessary to select a single CQI value for this set of PRBs. This selection is not standardized either, so it can be carried out, for instance, by calculating the average value, or by selecting the minimum value. In Fig. 6.3 it is shown an example of this fact.

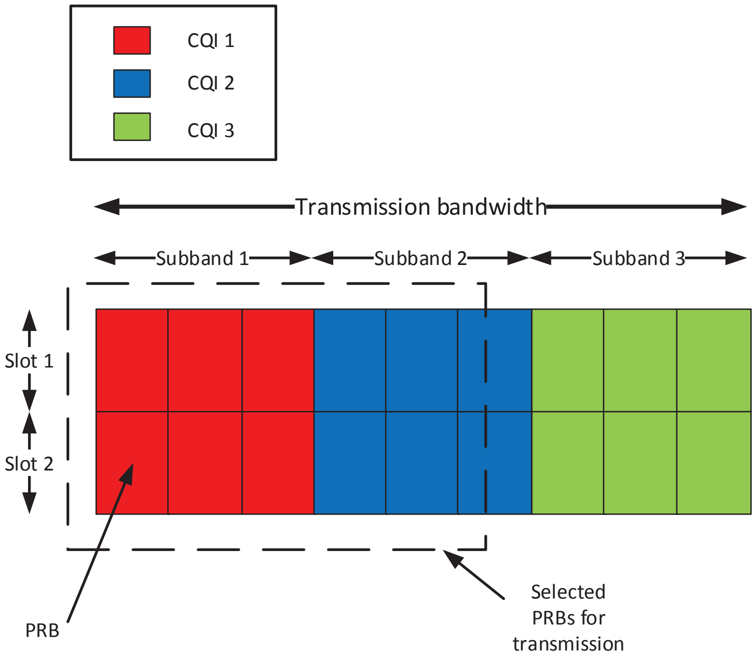

Figure 6.3: Example of the CQI selection for the transmission band

3. *Selecting the TB size:* Finally, the size of the TB to be transmitted has to be selected. First of all, it is selected the modulation scheme (QPSK, 16QAM or 64QAM). This modulation scheme will be that associated to the single CQI value of the transmission selected in previous step. Each modulation scheme has associated a standardized set of TB sizes that can be used for a certain number of assigned PRBs. From this set the value selected is that with resulting coding rate[2] closest to the coding rate associated to the

[2] The coding rate is calculated from the number of effective bits (number of physical resources multiplied by the modulation bits) and the amount of bits to be transmitted.

CQI value. Table 6.1 shows the modulation scheme and the coding rate associated to each CQI index.

Table 6.1: LTE CQI table

CQI Index	Modulation	Coding Rate
0	Out of range	Out of range
1	QPSK	0.076
2	QPSK	0.117
3	QPSK	0.188
4	QPSK	0.3
5	QPSK	0.44
6	QPSK	0.59
7	16QAM	0.36
8	16QAM	0.48
9	16QAM	0.6
10	64QAM	0.45
11	64QAM	0.55
12	64QAM	0.65
13	64QAM	0.75
14	64QAM	0.85
15	64QAM	0.93

From the previous process it can be deduced that the TB size depends on the CQI and the amount of PRBs to be used at each transmission of a subframe. Since both the selection processes the CQI and the amount of PRBs are not standardized, and these process depends on several factors, it cannot be known a priory the TB size that will be used in the next transmission.

6.3 Proposed Closed Loop Link Adaptation (CLLA)

In this section, a modification of the OLLA architecture, named Closed Loop Link Adaptation (CLLA), is presented. This proposal uses different offset values to adapt the MCS of the next TB to be transmitted, according to the number of assigned PRBs and the MCS proposed by the UE through the reported CQIs, instead of using only one offset value as in the typical OLLA. The number of PRBs and the MCS determine the TB size.

In Fig. 6.4 it is shown the LA scheme when the proposed CLLA is included. To perform the CLLA technique, the number of PRBs assigned to each UE has to be available before carrying out the offset update procedure. In the LTE downlink, the BS manages all the information about previous transmissions and schedules the new ones. Thus, unlike the OLLA technique, the CLLA technique has to be performed at the BS instead of at the UE. Due to this fact, the CLLA works with quantified information (CQIs), in contrast to the OLLA which works with exact information from the SNR estimator. The proposed CLLA is carried out as follows:

1. At transmission k, set a matrix of offsets $\Omega\left[k\right] = \left\{\omega\left[k\right]_{nl}\right\}_{N\times I}$, with $n = 1..N$ and $l = 1..L$, where N is the maximum number of PRBs assignable to a transmission ($N = 100$ in LTE for a 20MHz bandwidth), and I is the highest MCS available for the transmission ($I = 15$ in LTE).

2. The Hybrid Automatic Repeat Request (HARQ) manager reports to the CLLA block the Ack/Nack associated to a transmission of a number of PRBs n and an MCS i

   - If an Ack is received, the offset value $\omega\left[k+1\right]_{ni}$ of $\Omega\left[k+1\right]$ is increased by Δ_{up} steps.
   - If a Nack is received, the offset value $\omega\left[k+1\right]_{ni}$ of $\Omega\left[k+1\right]$ is decreased by Δ_{down} steps.

3. The scheduler selects the number of PRBs n and the tentative MCS i (based on the reported CQIs from the UE) to be used in the next transmission $k+1$, and reports these values to the CLLA block.

4. The CLLA block adjusts the MCS to a value $\hat{\imath} = round\left(i + \theta_{ni}\right)$ and returns this value to the scheduler. Note that although the MCS is an integer value, the value of the CLLA offset might not be integer, thus a rounding operation is performed.

5. The scheduler determines the TB size according to the number n of PRBs selected to the next transmission and the MCS $\hat{\imath}$.

As for the OLLA, the values of Δ_{up} and Δ_{down} has to satisfy the next equation:

$$aBLER_T = \frac{1}{1 + \frac{\Delta_{up}}{\Delta_{down}}} \tag{6.1}$$

Thus, the impact of the variability of the TB sizes on the LA performance is reduced by adjusting the CLLA offsets independently for different sets of PRBs and MCSs. This is possible due to the CLLA is located at the BS, so the exact number of PRBs to be used in the next transmission is available. Note that there must be as many CLLA blocks as UEs, to dispose of independent offset matrices for each UE.

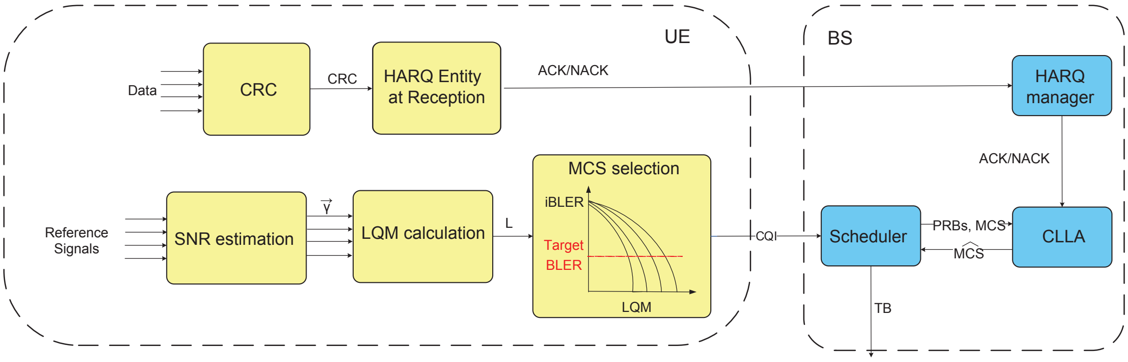

Figure 6.4: LA scheme including CLLA

A significant advantage of the CLLA is that its performance does not depend on how the UE carries out the CQI selection, a process which is not standardized for the LTE downlink. In addition to that, this proposal does not add signalling overhead and it is according to the 3GPP-LTE standard.

6.4 CLLA performance

6.4.1 Simulations environment

To evaluate the proposed CLLA, the 3GPP-LTE downlink simulator described in Appendix A is used. Table 6.2 summarizes the main simulation parameters used to obtain the results presented in this chapter. With the goal of achieving realistic performance results, several non-ideal features have been included such as channel and SINR estimation methods based on reference signals, Multiple Input Multiple Output (MIMO) precoding by codebook selection or non continuous reporting of CQI and Precoding Matrix Index (PMI) values.

For simulations, a video streaming traffic source [62] has been used. Basically this traffic source consists in a variable packet size following a truncated Pareto distribution and constant interarrival time. The simulated source had mean packet size of 3000 bytes and a mean throughput of 17.5 Mbps. Due to one single UE is simulated, all the PRBs of a subframe are available for transmission, except those in outage. The scheduler will adjust the number of PRBs according to the amount of queued data. As the size of data packets are distributed as a truncated Pareto, there will be a wide variation in the number of assigned PRBs at each transmission, and so in the size of the transmitted TBs.

Table 6.2: Simulation Parameters

Parameter	Value
Carrier frequency	2 GHz
Sampling frequency	30.72 MHz
System bandwidth	20 MHz
FFT size	2048
Number of data subcarriers	1200
OFDM symbols per subframe	14
Allocable PRBs	100
Channel model	Extended Pedestrian A [37]
Mobile terminal speed	4 km/h
Transmission mode	1 layer spatial multiplexing 2x2 MIMO
Codebook	2 bits [39]
MIMO detection	Zero Forzing
Channel estimation method	Low-pass filter [49]
Interference and Noise power estimation	Error based
Reference signals overhead	According to 3GPP TS 36.211 [39]
PMI reporting interval	2 ms
Turbo decoder	SOVA based
CQI reporting interval	2 ms
Number of CQI bits	4 [12]
Target BLER	10% [12]
SNR	0dB - 30dB
Scheduling algorithm	Round Robin
Source model	Streaming
Simulation length	100s

6.4.2 Results

In this section, a comparison between the traditional OLLA and the proposed CLLA is carried out. In Fig. 6.5, BLER results are shown for a range of mean

SNR. For both the OLLA and the CLLA, BLER results are very close to the target value of 0.1. Only for the highest SNR values BLER results decrease as the MCS saturates. Also very similar system throughput results are achieved for both techniques, as it shown in Fig. 6.6. System throughput saturates when it achieves the throughput of the traffic source (around 17.5 Mbps), which happens when the mean SNR exceeds 10dB.

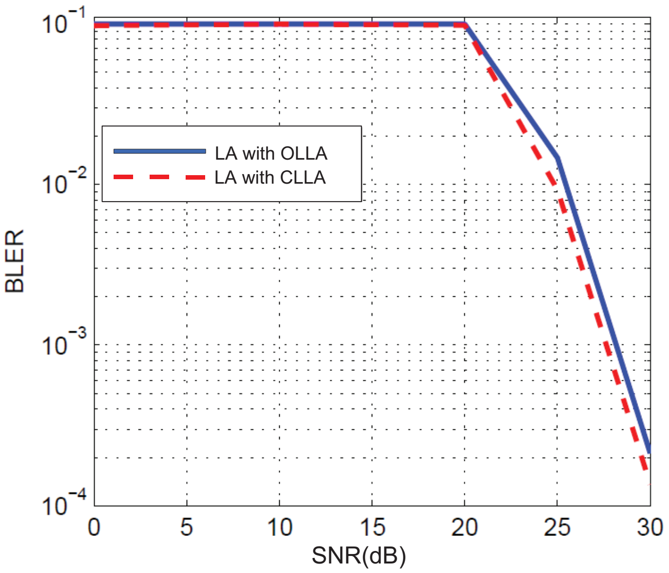

Figure 6.5: BLER results for OLLA and CLLA

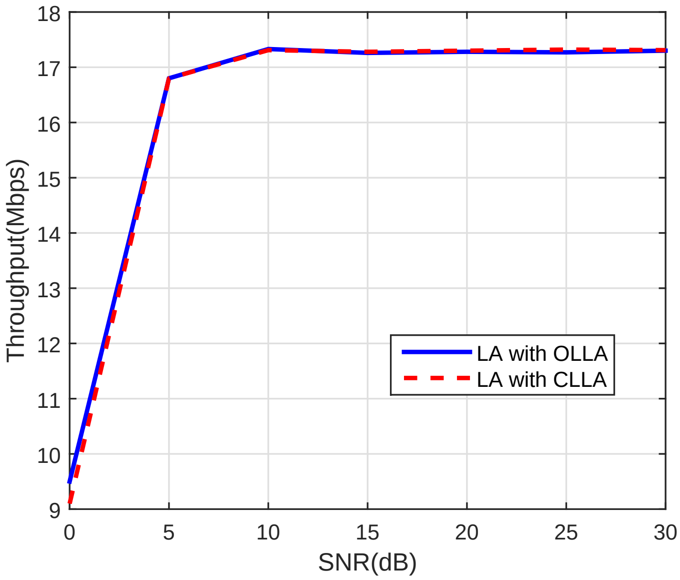

Figure 6.6: Throughput results for OLLA and CLLA

In Fig. 6.7, Fig. 6.8 and Fig. 6.9, the cumulative distribution functions (CDFs) of TB sizes are presented for a mean SNR of 5dB, 15dB and 25dB, respectively. Furthermore, BLER results for different ranges of TB sizes are also presented for the OLLA and the CLLA. Notice that since only certain values for the TB sizes are allowed [12], the CDFs might not present a smooth shape if for the evaluate mean SNR only a few TB sizes are used. In addition to that, CDFs might not always coincide for both techniques, as they modify in a different way the MCS to be used at the transmission, and so the TB size. Nevertheless, in this case CDFs are always nearly the same in all cases.

For a mean SNR of 5dB, the system throughput is lower than the throughput of the traffic source. This means that most of the time there is enough queued data to assign all the PRBs of the subframe to the UE. Thus, the variability of the TB sizes will be low (as the CDF shows), and so both techniques will work in a similar way.

A wider range of TB sizes is shown for a mean SNR of 15dB. Regarding BLER results, they are close to 0.1 for the whole range of TB sizes when the CLLA is used. However, for the OLLA, BLER results are well below 0.1 when small TBs are transmitted, while for the largest TBs the target BLER is considerably raised. Thus, although mean BLER results are close to the target for both techniques, the OLLA shows a significant variation in BLER results for different TB sizes.

Finally, for a mean SNR value of 25dB, there is also a wide range of TB sizes. However, as the MCS saturates due to the high mean SNR value, a low BLER is achieved for the whole range of TB sizes whatever the OLLA or the CLLA is used. Thus, there are not significant differences in performance between these techniques.

The effects of having different BLER results for different TB sizes are shown in Fig. 6.10, where goodput results are presented. In this figure, for a mean SNR of 15 dB, there is an increase of goodput (about a 10%) when the CLLA is used, even though mean BLER and throughput results are the same in both cases. The reason is that, as it was shown in Fig. 6.8, higher TB sizes present a greater BLER for the OLLA. Then, although the average number of erroneous TBs is similar in both cases, TBs with a larger number of bits are more likely to be erroneously received when OLLA is used. As erroneous TBs are discarded, the amount of discarded bits when the OLLA technique is used is higher than when using the CLLA technique. Also, for a mean SNR of 20dB a goodput gain of 14% is achieved. In general, better goodput results are achieved by the CLLA. Only for the lowest and the highest values of mean SNRs, results are very close

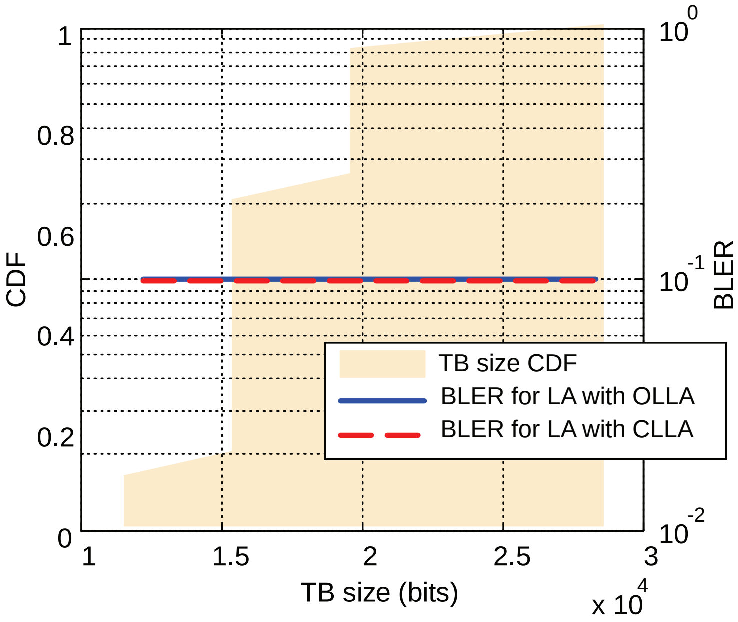

Figure 6.7: BLER results for different TB sizes and a mean SNR of 5dB

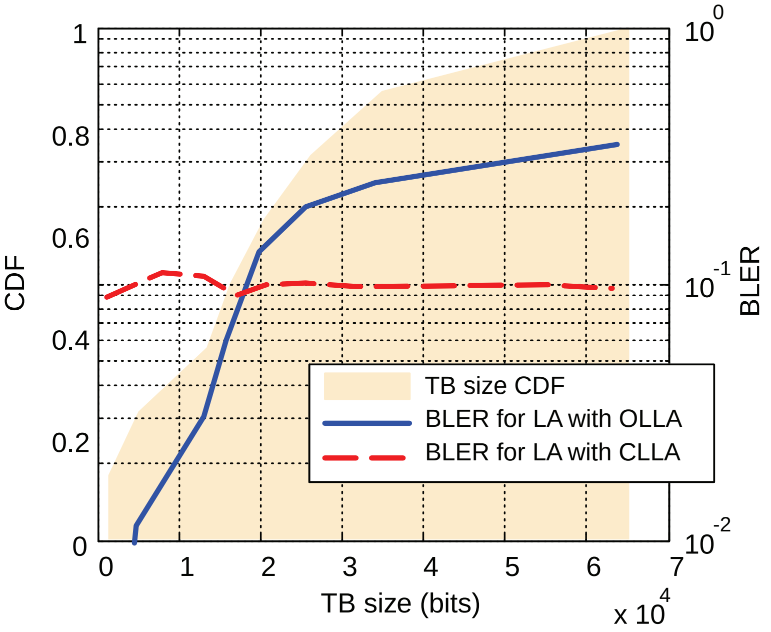

Figure 6.8: BLER results for different TB sizes and a mean SNR of 15dB

as the performance of both techniques is similar.

These results show that proposed CLLA outperforms OLLA when there is wide range of TB sizes and the MCS is not saturated. The existence of these conditions mainly depends on the traffic source and the SNR. Furthermore, these results have been obtained for one only UE. Having more than one UE would mean that the assigning process of PRBs to each UE might be different, and so the variability of the TB sizes.

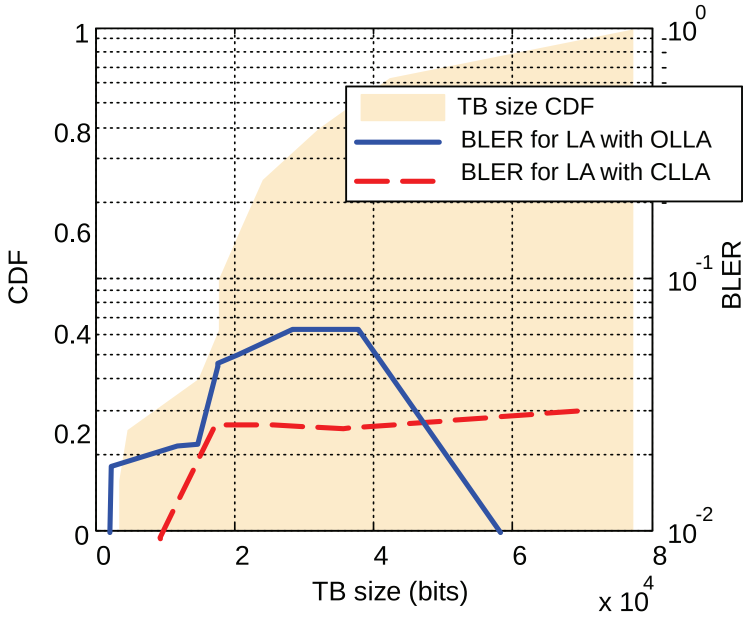

Figure 6.9: BLER results for different TB sizes and a mean SNR of 25dB

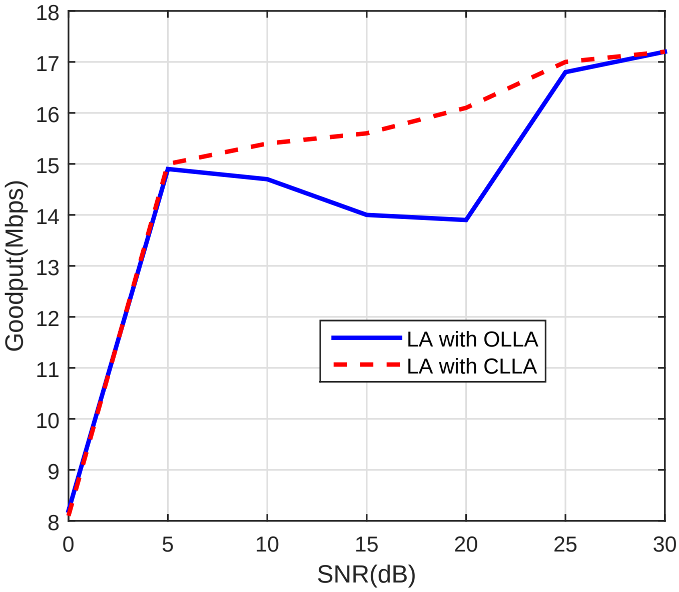

Figure 6.10: Goodput results for OLLA and CLLA

Chapter 7

Conclusions and Future Works

In this chapter conclusions of this work are presented.

- In chapter 2 principles of turbo decoding have been described. Then, SOVA based decoders have been deeply studied since they are extensively used in real implementations due to its low complexity compared with the MAP based decoders. Then, it has been shown that a truncation of the updating process of the SOVA can cause an important degradation of the BLER performance, mainly for high coding rates. Thus, we proposed a modified SOVA which adaptively adjusts the depth of the updating window to the minimum value that achieves the best error correction. Simulations results show that for typical operating points of LTE the adSOVA reduces complexity compared with the conventional SOVA, but without BLER degradation. Then, a deeper updating window can be used to improve BLER without a significant increase of complexity. Furthermore, a comparison with Max-Log-MAP algorithm has been carried out, showing the adSOVA an important complexity reduction.

- In chapter 3 the AMC process of LTE and LTE-A has been modeled in order to carry out a complete analysis of the well-known traditional OLLA algorithm. In particular, we present a model of the instantaneous BLER and the averaged BLER, when turbo codes are used, by means of binary logistic functions and modified binary logistic functions, respectively. Then, convergence of the traditional OLLA has been studied, providing an expression of the maximum step size that ensures convergence. Furthermore, the dynamics of the traditional OLLA has been studied, analyzing how different sizes of this step leads to different performance depending on the specific scenario evaluated, even if they guarantee convergence.

- In chapter 4, a new approach to the OLLA technique, the enhanced OLLA (eOLLA), has been proposed, which is able to: 1) dynamically adapt its step size according to the channel state, and 2) be able to track the channel variations when no data packet is received. This proposed technique is based on binary logistic regression functions. Furthermore, a complete comparison of the performance of both the traditional OLLA and the eOLLA algorithms has been carried out through different scenarios based on the LTE and LTE-A downlink, showing that the eOLLA outperforms the traditional OLLA.

- In chapter 5, a full analytic model of the AMC process with channel coding based on turbo coding for constant power in LTE downlink has been developed by logistic regression. Then, SNR thresholds used by AMC have been optimized to maximize the data rate (subject to an average BLER constraint and constant power) in two ways: a) adjusting each threshold individually (optimal solution), or b) forcing the relative distance among thresholds to keep fixed to those predesigned (sub-optimal). Next, OLLA technique has been analyzed when several offsets are used, comparing its performance and the performance of the traditional OLLA with a single offset with the AMC model with optimized thresholds. First, it has been shown that single-offset OLLA results match those of the sub-optimal AMC. Later, the analysis of multi-offset OLLA has revealed that, although it could potentially match the optimal AMC performance, this is not possible in practice since it does not converge to a single solution but it wanders over the set of thresholds able to keep the BLER below the target. Finally, it has been shown how adjusting each threshold individually makes the optimization procedure independent of non-ideal features in realistic LTE downlink implementations. A modification of the multi-offset OLLA in order to include rate constraints within the procedure might improve spectral efficiency as much as 20%.

- In chapter 6, it has been shown that a wide variation of TB sizes can produce a significant degradation in the AMC performance for the downlink of LTE, and that it cannot be corrected by the OLLA technique at the UE. Thus, a Closed Loop Link Adaptation (CLLA) has been proposed to avoid the impact of the variability of the TB sizes in the AMC. The proposed CLLA has been shown to outperform the OLLA when a wide range of TB sizes is used by the transmitter by maintaining the BLER close to the target for the whole range. The proposed technique is located at the BS, so it does not depend on the UE implementation, and it is according to the standard for the downlink of 3GPP-LTE.

There are several paths that can be followed in order to continue with this work. Some future works proposals are discussed next:

- Since in Max-Log-MAP algorithm implementations also make use of truncation when calculating their metrics, it could be studied if a similar modification than that could be proposed for the adSOVA, in order to improve their performance in a similar way than described in chapter 2 for the SOVA.

- Logistic regression has been proved to be able to model BLER when turbo coding is used. Then, it could be deeper studied whether logistic regression can accurately model a turbo decoder. Furthermore, average SNR has been used in this work to implement logistic regression. Thus, it also could be analyzed which other variables can help to improve this model, such as the variance of the measured SNR or the size of the data block to be decoded.

- In chapter 5 it was discussed that a multi-offset implementation of the traditional OLLA algorithm could potentially lead to a better performance than a single-offset implementation. However, this multi-offset implementation does not converge to a single point, but wanders around all these points for which the averaged BLER meets the target BLER, and which may lead to different throughput performance. Thus, a it could be studied how to add restriction to the multi-offset OLLA implementation in order to only converge to those points for which the target BLER is met at the same time that the throughput is the maximum achievable.

- In an eICIC scenario, UEs that are affected by the transmission of ABSs are allowed to transmit to sets of CQI independently, in order to differentiate link conditions when the aggressor cell transmits ABSs (low interference scenario), and when transmits normal subframe (high interference condition) [53]. Then, in an eICIC scenario we could differentiate two parallel adaptive modulation an coding process. As it was explained in chapter 5, the adaptive modulation and coding process typically meets the target BLER on average. Then, it could be studied whether these two adaptive modulation and coding processes can be jointly optimized in order to achieve the maximum throughput as possible, while meeting the target BLER. Please note that the target BLER should be met when averaging the BLER of the two adaptive and modulation process, not by each process on its own.

Chapter 8

Técnicas de adaptación del enlace para redes celulares

8.1 Contribuciones de este trabajo

Este trabajo se ha centrado en el estudio y mejora de la adaptación del enlace en redes móviles celulares, en concreto para redes Long Term Evolution (LTE) y LTE Advanced (LTE-A). Este trabajo se divide en dos partes principales:

1. Codificación de canal: Se ha estudiado el algoritmo SOVA, así como el impacto que tiene el estado del canal en su rendimiento. A partir de este análisis, se ha propuesto una modificación del algoritmo SOVA, el SOVA adaptativo (adSOVA) [22], que permite adaptar la cantidad de procesado que realiza en función de las características de los datos a decodificar. De esta forma, permite reducir procesado si la recepción se produce con buena calidad, y aumentarlo si la calidad de la recepción no es buena. Con esto, se consigue una implementación del algoritmo SOVA que permite mejorar las prestaciones del algoritmo SOVA tradicional, sin un incremento significativo de su complejidad.

2. Modulación y codificación de canal adaptativas: Se ha realizado un análisis de esta técnica para el caso de LTE y LTE-A, centrándose sobre todo en el estudio de la técnica de adaptación del enlace de bucle externo (Outer Loop Link Adaptation, OLLA), ampliamente referenciada en la literatura, pero sin la existencia de un estudio exhaustivo. Así, este trabajo se ha centrado en proporcionar un análisis completo de la técnica OLLA, incluyendo la optimización del proceso de modulación y codificación de canal adaptativas para potencia de transmisión constante junto a diversas imple-

mentaciones de la técnica OLLA. A partir de este análisis, se han propuesto modificaciones de esta técnica que permiten una mejor adaptación de la transmisisión en distintos escenariós, con el OLLA mejorado (enhanced OLLA, eOLLA) [26], y la técnica de adaptación del enlace para bucle cerrado (Closed Loop Link Adaptation, CLLA).

8.2 SOVA Adaptativo para Receptores 3GPP-LTE

8.2.1 Turbo codificación

En la Fig. 8.1 se muestra el esquema genérico de un turbo codificador.

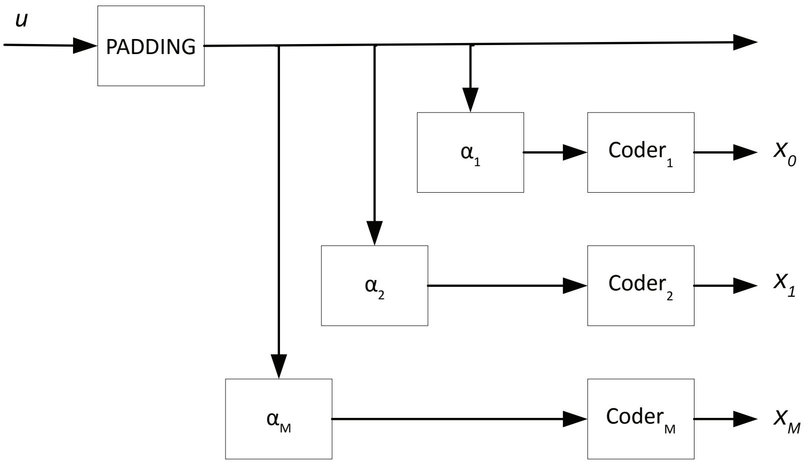

Figure 8.1: Estructura genérica turbo codificador

El turbo codificador consiste en una serie de ramas en las que se realiza una codificación convolucional, previo entrelazado del mensaje de entrada. Generalmente una de las ramas no realiza ningún procesado, poniendo a su salida el mensaje de entrada. La secuencia de bits a la salida de esta rama se conoce como bits sistemático, y las secuencias a la salida del resto de ramas se conoce como bits de paridad. Al inicio del proceso de codificación se suelen añadir unos bits de cola (bloque PADDING en la Fig. 8.1), que se encargan de resetear la memoria de los codificadores convolucionales de cada rama una vez se hayan transmitido todos los bits.

En la Fig. 8.2 se muestra el esquema del turbo codificador para LTE.

Este turbo codificador está formado por dos codificadores convolucionales, conteniendo cada uno tres registros de un bit, con lo que su profundidad de

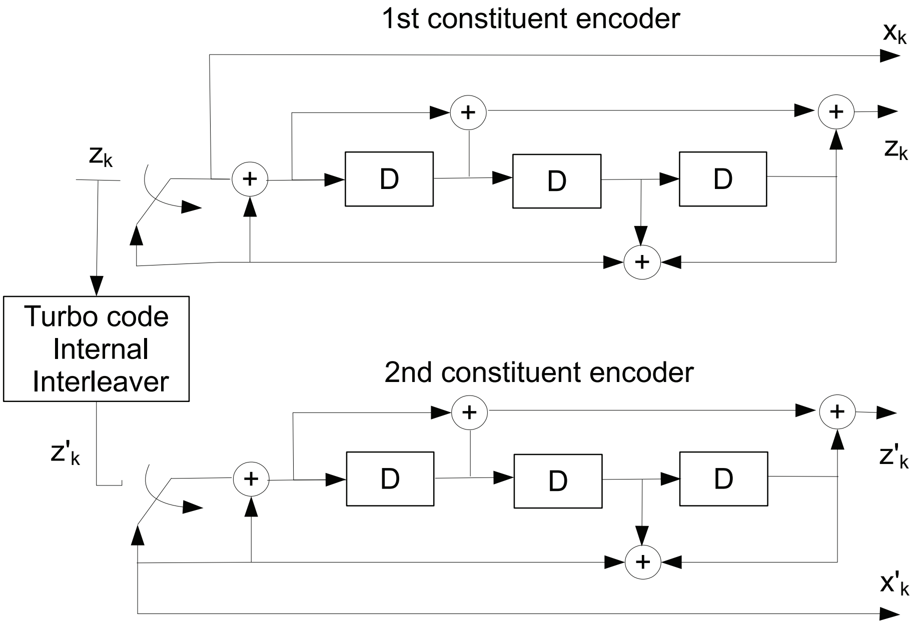

Figure 8.2: Estructura turbo codificador 3GPP-LTE

memoria es $2^3 = 8$ bits. Este turbo codificador transmite una secuencia de bits sistemáticos y dos de paridad, con lo que su tasa de codificación es de 1/3.

8.2.2 Turbo decodificación

El esquema básico de un turbo decodificador consiste en la concatenación de decodificadores basados en algoritmos de entrada y salida suave (Soft Input Soft Output, SISO), es decir, decodificadores que acepten tanto a su entrada como a su salida bits ponderados por la fiabilidad de su valor. En la Fig. 8.3 se muestra el esquema de un turbo decodificador para LTE y LTE-A.

Existen dos esquemas básicos de algoritmos SISO:

- Máximo a Posteriori (Maximum A Posteriori, MAP) [6]: Este algoritmo es óptimo en la minimización de error de bit, pero tienen una alta complejidad. Por ello se usan versiones simplificadas de este algoritmo, como el Max-Log-MAP [27].

- Algoritmo de Viterbi de Salida Suave (Soft Output Viterbi Algorithm, SOVA) [7][4]: Modificación de algoritmo de Viterbi [32] que permite generar

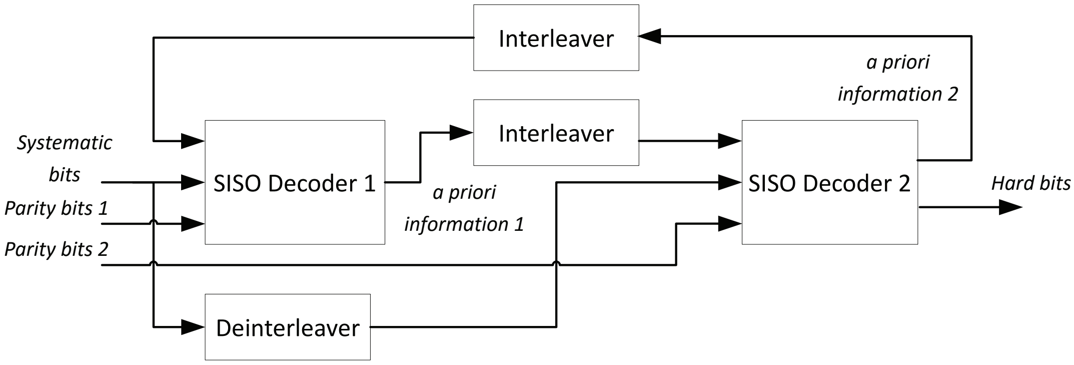

Figure 8.3: Estructura turbo decodificador para LTE y LTE-A

salidas suaves. Su capacidad de corrección de errores es menor que la del algoritmo MAP en cualquiera de sus variante, pero su complejidad es también menor.

La menor complejidad del algoritmo SOVA lo hace muy adecuado para ser empleado en la implementación de receptores móviles para LTE y LTE-A [28].

Turbo decodificador basado en SOVA

El algoritmo SOVA construye un diagrama de Trellis [32] de la secuencia $\mathbf{y}$ transmitida. Un diagrama de Trellis es una matriz de nodos en la que cada nodo representa un posible estado del turbo codificador, y cada columna todos los posibles estados en un instante de tiempo t. Recibir un bit sistemático $x_t = 0$ o $x_t = 1$ supone una transición de un estado s'_0 o s'_1, respectivamente, a un estado s. Cada transición está asociada también a un bit de paridad z_t y una métrica acumulada $M_t(s', s)$. Esta métrica acumulada se calcula:

$$M_t(s', s) = M_{t-1}(s', s) + x_t \cdot L_c \cdot y_{t,1} + z_t \cdot L_c \cdot y_{t,2} + x_t \cdot L_t \tag{8.1}$$

siendo:

- L_c : La fiabilidad del bit recibido.
- $y_{t,1}$: el bit sistemático recibido.
- $y_{t,2}$: el bit de paridad recibido.
- L_t : la fiabilidad *a priori*, es decir, la fiabilidad del bit recibido calculada por el otro decodificador SOVA que conforma el turbo decodificador.

La transición con la mayor métrica acumulada de una columna del diagrama de Trellis pasa a formar parte de la secuencia denominada camino *superviviente*, y la transición opuesta del mismo nodo pasa a formar parte del camino *competitivo*. El camino *superviviente* es la secuencia con mayor probabilidad de haber sido decodificada correctamente. La fiabilidad de cada bit del camino *superviviente* se calcula como:

$$\Delta_t^0 = \frac{1}{2} \left| M_t\left(s'_0, s\right) - M_t\left(s'_1, s\right) \right| \tag{8.2}$$

Cada vez que se decodifica un nuevo bit se realiza un proceso de actualización (updating) de todas las fiabilidades calculadas anteriormente:

$$\Delta_t^{MEM} = \min_{k=0,\ldots,MEM} \left\{ \Delta_t^k \right\} \tag{8.3}$$

siendo MEM el número de bits decodificados hasta el momento, y t el instante temporal en el que se ha recibido el último bit. Este proceso de actualización es muy costoso, por lo que en la práctica se procede a realizar un enventanado, de forma que sólo se lleve a cabo sobre los últimos w bits decodificados [4]. Sin embargo, este enventanado puede reducir la capacidad de corrección de errores [29]. Por tanto, una ventana demasiado grande puede suponer que se realicen más operaciones de las necesarias (lo que implica un aumento de tiempo de procesado y consumo de potencia) para decodificar una secuencia con una tasa de error determinada, mientras que una ventana demasiado pequeña puede conllevar una degradación de la capacidad de corrección de errores.

8.2.3 SOVA Adaptativo

Con el fin de optimizar el proceso de actualización descrito anteriormente, se propone una modificación del algoritmo SOVA, el SOVA Adaptativo (adSOVA), capaz de detener el proceso de actualización en el momento en el que no es posible conseguir un mayor rendimiento de la decodificación aunque este proceso continuara. De esta forma, el algoritmo propuesto consigue la misma capacidad de corrección de errores del algoritmo SOVA tradicional, pero disminuyendo su complejidad al reducir el procesado de la actualización.

Para ilustrar el algoritmo propuesto, en la Fig. 8.4 se muestra un ejemplo de los caminos *superviviente* y *competitivo* en el instante t y $t-1$. En esta figura puede observarse que cada camino coincide para ambos instantes temporales en el nivel de memoria 3. Por tanto, si ocurriera que la fiabilidad calculada para el instante de tiempo t no fuera menor que la calculada en el nivel de memoria $t-1$, a partir del nivel de memoria 3 no se produciría ninguna actualización nueva con respecto al proceso de actualización del instante $t-1$. De este modo, a partir de este nivel de memoria podría detenerse el proceso de actualización.

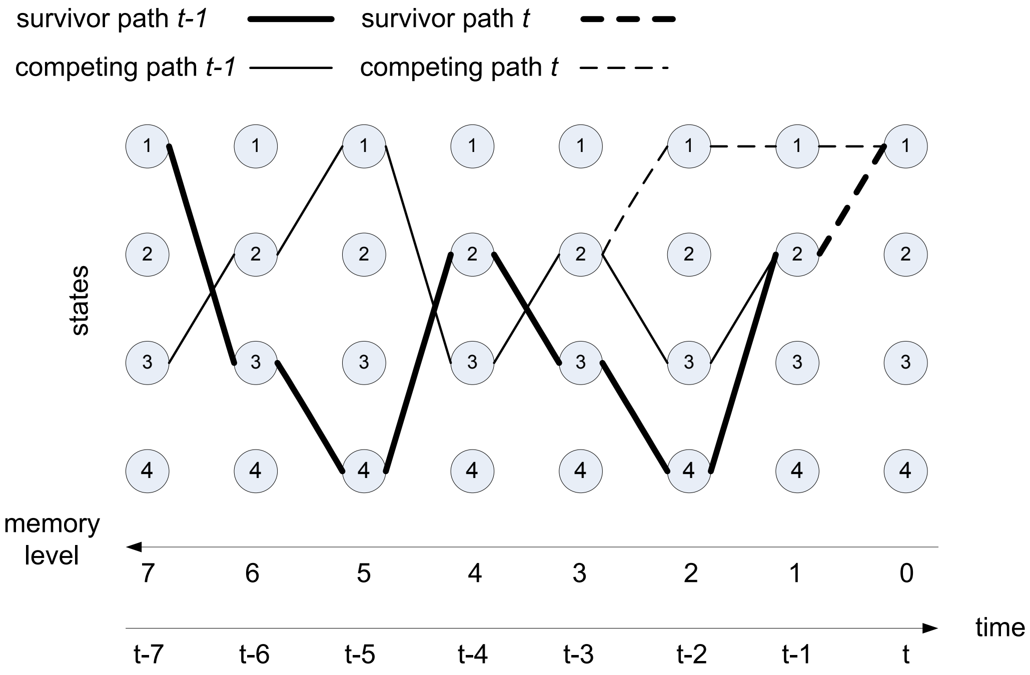

Figure 8.4: Trellis tree example

Con todo esto, se propone una modificación de este proceso de actualización, cuyo flujograma se representa en la Fig. 8.5, y cuyos pasos se describen a continuación:

1. Calcular la fiabilidad en el nivel de memoria 0, y tomarlo como mínimo valor de fiabilidad ($\Delta_{min} = \Delta_t^0$). Comenzar el proceso de actualización.

2. Comprobar en cada nodo si la fiabilidad es la menor que el valor mínimo guardado. En tal caso actualizar este valor ($\Delta_{min} = \Delta_t^k$).

3. Comprobar si los caminos *superviviente* y *competitivos* coinciden para t y $t-1$.

4. Si el paso 3 se satisface, y el valor mínimo de fiabilidad se actualizó a uno diferente al del nivel de memoria 0, el proceso de actualización puede finalizarse para el instante t. En caso contrario, volver al paso 2.

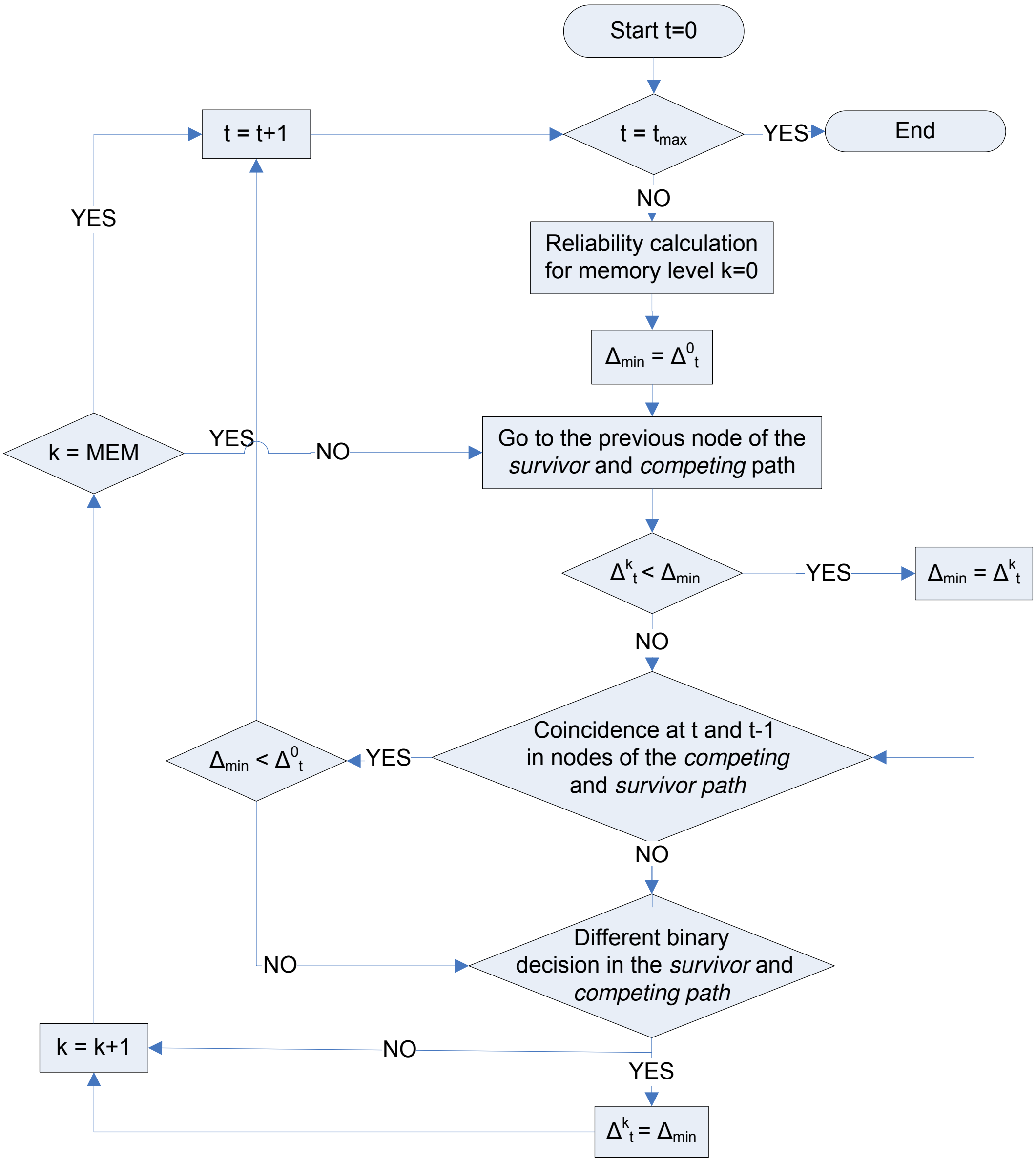
Start t=0
t = t_max
YES
End
NO
Reliability calculation for memory level k=0
Δmin = Δ0t
Go to the previous node of the survivor and competing path
Δkt < Δmin
YES
Δmin = Δkt
NO
Coincidence at t and t-1 in nodes of the competing and survivor path
YES
Δmin < Δ0t
NO
Different binary decision in the survivor and competing path
YES
Δkt = Δmin
NO
k = k+1
k = MEM
YES
NO
YES
t = t+1

Figure 8.5: Flow diagram of the proposed adSOVA

Rendimiento del algoritmo adSOVA

El rendimiento del algoritmo adSOVA se ha evaluado mediante su inclusión un simulador de LTE y LTE-A [36].

En primer lugar, se ha comparado el rendimiento en Tasa de Error de BLoque (BLock Error Rate, BLER) del algoritmo SOVA, para diferentes enventanados del proceso de actualización, con el algoritmo Max-Log-MAP, con un enventanado de $w = 20$; para distintos esquemas de modulación y codificación, indicados estos mediante un valor de Indicador de Calidad de Canal (Channel Quality Indicator, CQI) [12]. En la Fig. 8.6 se muestra cómo el algoritmo Max-Log-

MAP siempre mejora al algoritmo SOVA, así como el efecto del enventanado es más significativo para este último algoritmo conforme se usa un esquema de modulación y codificación de menor redundancia.

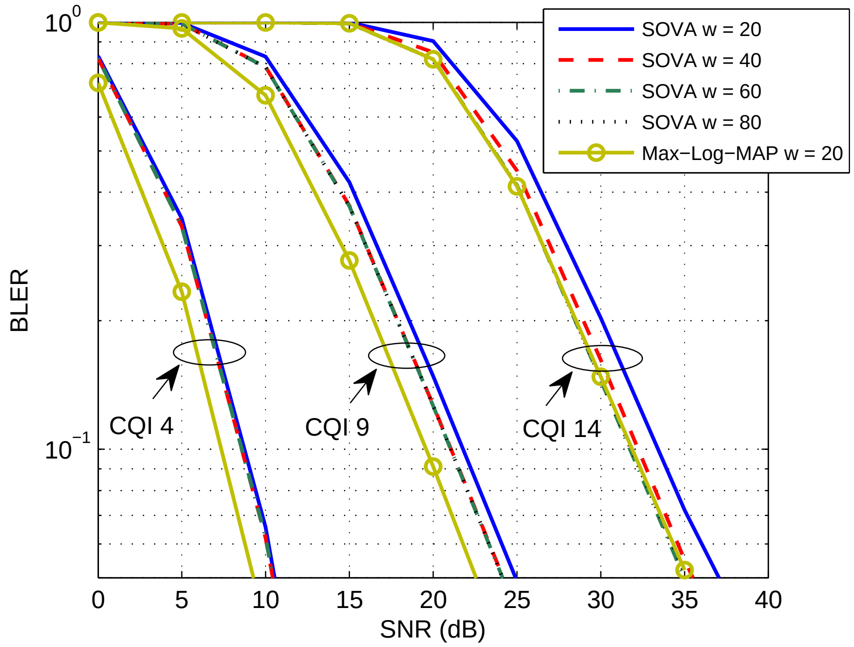

Figure 8.6: BLER para diferentes esquemas de modulación y codificación

En la Fig. 8.7 se muestra el tiempo de procesamiento empleado por cada algoritmo, en ciclos de CPU, para los resultados de BLER presentados anteriormente, en el caso de usar el algoritmo SOVA tradicional, y el algoritmo adSOVA propuesto[1].

Con los resultado anteriores, se puede deducir que el algoritmo adSOVA permite alcanzar rendimientos equivalentes al del algoritmo SOVA tradicional con una ventana $w = 60$, pero con un tiempo de procesamiento equivalente al del algoritmo SOVA para una ventana pequeña de tamanõ $w = 20$. Además, en la Fig. 8.8 se muestra la función de masa de probabilidad del número de pasos del proceso de actualización para el algoritmo adSOVA con $w = 60$. Estos resultados muestran que en general el número de pasos tiene una pequeña varianza, lo que significa que aunque se utilice un valor de ventana máxima muy grande para el caso del algoritmo adSOVA, rara vez se alcanzará dicho valor, con lo que una reducción en el tiempo de procesamiento puede alcanzarse para la mayoría de los bloques a decodificar.

Finalmente, se ha calculado la complejidad que implica la implementación del algoritmo adSOVA, y su comparación con el algoritmo SOVA tradicional y el algoritmo Max-Log-MAP. Estos resultados se muestran en la Tabla 8.1, siendo l

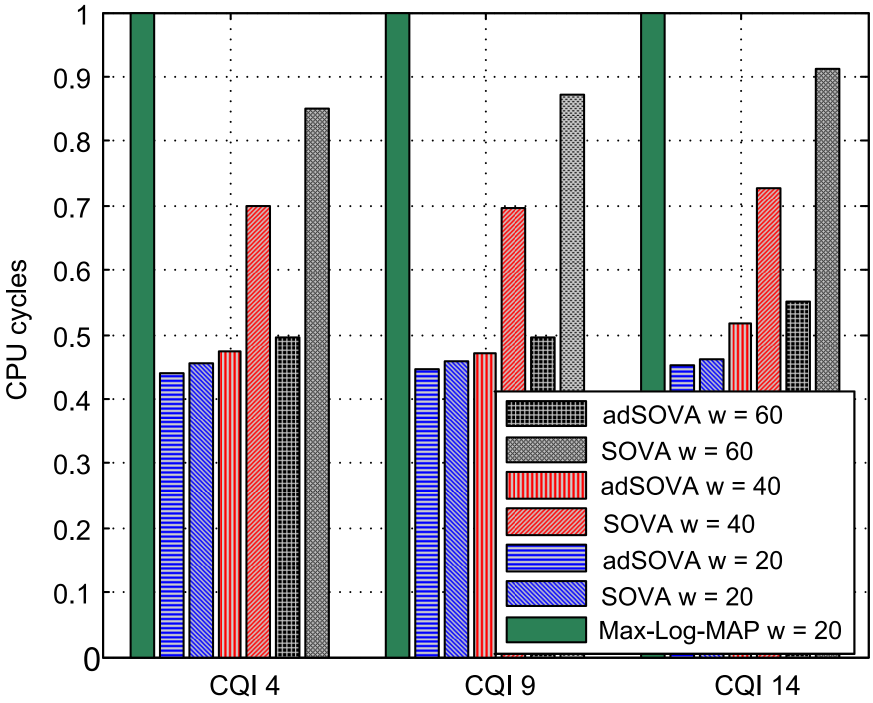

Figure 8.7: Número medio de ciclos de CPU para diferentes esquemas de modulación y codificación

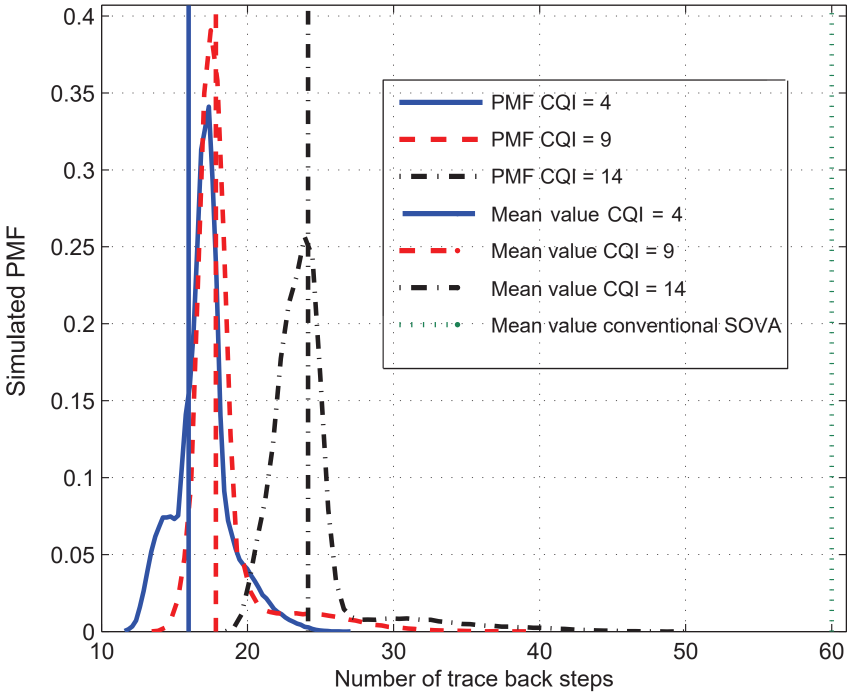

Figure 8.8: Función masa de probabilidad para el algoritmo adSOVA con $w = 60$

la longitud del bloque a decodificar, y w la longitud de la ventana del proceso de actualización updating.

Table 8.1: Número de operaciones

	Sumas	Multiplicaciones	Comparaciones
SOVA	$49l$	$48l$	$15 + 2wl - w(w-1)$
adSOVA	$49l$	$48l$	$15 + 5wl - w(w-1)/2$
Max-Log-MAP	$128l$	$112l$	$40l$

El algoritmo SOVA tradicional y el adSOVA sólo se diferencian en el número de comparaciones, que dependen del número de pasos del proceso de actualización. Como este número es variable para el caso del adSOVA, se muestra el valor para el caso de que se alcance el valor máximo w. De esta forma, como máximo, el algoritmo adSOVA realizará $3wl - w(w-1)/2$ comparaciones más que el algoritmo SOVA tradicional. En cualquier caso, el algoritmo adSOVA tiene una complejidad menor que el algoritmo Max-Log-MAP, para cualquier valor de l y w.

8.3 Modelado de la Modulación y Codificación de Canal Adaptativa en LTE

En la Fig. 8.9 se muestra un modelo del proceso de Modulación y Codificación de canal adaptativas (Adaptive Modulation and Coding, AMC) [39]. La relación señal a ruido (Signal to Noise Ratio, SNR) instantánea $\vec{\gamma}$ sobre un cierto número de subportadoras m, estimada en el receptor, se utiliza para calcular una métrica de calidad del enlace L, la cual permite determinar el CQI a utilizar para cumplir una determinada BLER media. Por simplicidad, se utilizará el valor de SNR como el valor de métrica de calidad del enlace. La traducción de SNR media a CQI se realiza mediante la definición de n intervalos de SNR instantánea, asociados cada uno a un valor de los n posibles de CQI. Los umbrales de estos intervalos $\{\Psi_i\}_{i=0,1,..,n}$ deben diseñarse correctamente para que se cumpla el valor de BLER media objetivo $BLER_T$.

El valor de BLER media (average BLER, aBLER), para un valor de SNR media temporal Γ, se obtiene a partir de la siguiente ecuación:

$$aBLER(\Gamma, \{\Psi_i\}) = \sum_{i=0}^{n-1} \int_{\Psi_i}^{\Psi_{i+1}} iBLER_i^{AWGN}(\gamma)p^o(\Gamma, \gamma)d\gamma, \tag{8.4}$$

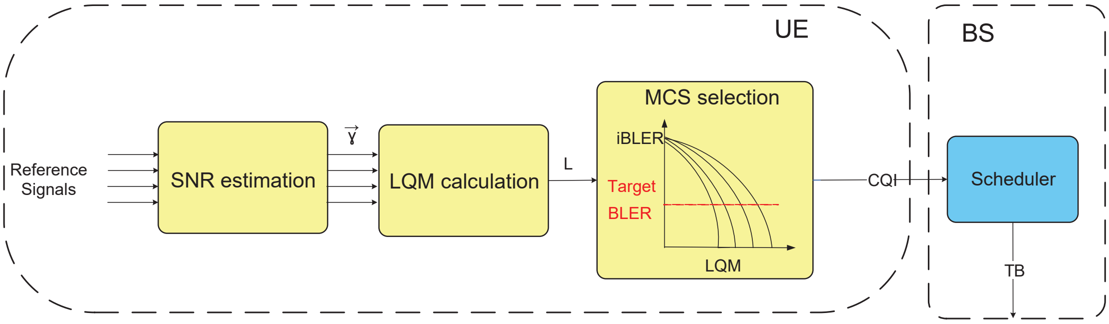

Figure 8.9: Esquema del proceso de modulación y codificación adaptativa

siendo $iBLER_i^{AWGN}(\gamma)$ la BLER instantánea (instantaneous BLER, iBLER) para un esquema de modulación y codificación i sobre un canal de Ruido Blanco Gaussiano Aditivo (Additive White Gaussian Noise, AWGN), y $p^o(\Gamma, \gamma)$ la función densidad de probabilidad de la SNR instantánea condicionada a la transmisión.

Modelado de BLER instantánea mediante regresión binaria logística

Para calcular la aBLER es necesario disponer de una expresión para $iBLER_i^{AWGN}(\gamma)$, que para el caso de uso de turbo códigos no está disponible en la literatura. Dado que la iBLER representa la probabilidad de estar en dos estados {*error*, *no error*}, se puede modelar usando la regresión logística binaria [44]. Esta regresión es un clasificador binario basado en una o más entradas. La función logística binaria propuesta es:

$$iBLER_i^{AWGN}(\gamma) \approx f_i(\gamma) = \frac{1}{1 + e^{-\alpha_{i_0}\gamma - \alpha_{i_1}}}, \qquad (8.5)$$

siendo α_{i_0} y α_{i_1} los valores a encontrar mediante la regresión logística binaria. En al Fig. 8.10 se muestra un ejemplo del ajuste alcanzado por el modelo propuesto.

8.3.1 Outer Loop Link Adaptation (OLLA)

Una selección estática de los umbrales de SNR para la implementación de la modulación y codificación de canal adaptativas no siempre se ajusta perfectamente a la BLER media, debido a que las condiciones del enlace móvil son inherentemente variantes. Por tanto, diferentes umbrales deberían usarse para diferentes condiciones de transmisión. La técnicas de adaptación del enlace de bucle externo (Outer Loop Link Adaptation, OLLA) [23–25], se encarga de modificar el valor de estos umbrales mediante un offset, en función de si los datos

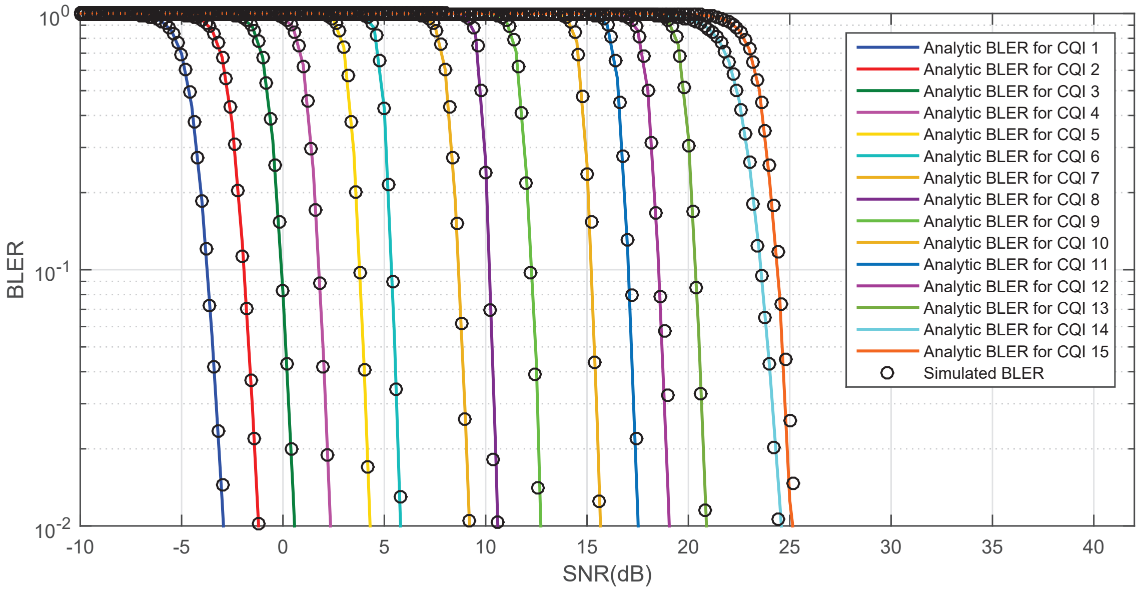

Figure 8.10: Ajuste de las funciones logísticas binarias para modelar BLER instantánea

transmitidos se están o no recibiendo correctamente[2]. Cada vez que se extrae un CRC, el valor del offset $\theta\left[k\right]$ se actualiza de la siguiente forma:

$$\theta_{dB}\left[k\right] = \theta_{dB}\left[k-1\right] + \Delta_{up}\cdot e\left[k\right] - \Delta_{down}\cdot\left(1-e\left[k\right]\right), \tag{8.6}$$

siendo

- $\theta_{dB}\left[k\right] = 10\cdot log_{10}\left(\theta\left[k\right]\right)$.

- $e\left[k\right]$ un indicador de error (0 si no error, 1 si error).

- Δ_{up} y Δ_{down} valores constantes de incremento y decremento del offset. Deben cumplir la siguiente ecuación para satisfacer la $aBLER_T$ [25]:

$$aBLER_T = \frac{1}{1+\frac{\Delta_{up}}{\Delta_{down}}}. \tag{8.7}$$

Los umbrales de SNR son por tanto modificados por un offset k discreto en el tiempo:

$$\Psi_i\left[k\right] = \gamma_i\cdot\theta\left[k\right]. \tag{8.8}$$

[2] Esta información se proporciona en LTE mediante la extracción de un código de redundancia cíclica (Cyclic Redundancy Code, CRC).

8.3.2 Convergencia en promedio del OLLA

La dinámica de la técnica OLLA implica que el offset se encontrará siendo continuamente incrementado en Δ_{up} en caso de error, o decrementado en Δ_{down} en caso contrario. Esto implica que la BLER instantánea se encontrará cambiando continuamente. Sin embargo, el objetivo es que el sistema converja a una BLER media. Por tanto, para estudiar la convergencia de la técnica OLLA se procederá a usar valores promedio. Para ello, se aplica la esperanza matemática a la ecuación (8.6) para encontrar el valor medio $E\left[\theta_{dB}\left[k\right]\right] = \theta_{dB_k}$, obteniéndose tras reordenar:

$$\theta_{dB_k} = \theta_{dB_{k-1}} + \left(\Delta_{up} + \Delta_{down}\right) \cdot E\left[e\left[k\right]\right] - \Delta_{down}. \tag{8.9}$$

Cuando $k \rightarrow \infty$ el proceso puede considerarse ergódico. La ecuación (8.9) corresponde a una ecuación en diferencias, es decir, existe una relación recurrente del tipo $\theta_{dB_k} = T\left(\theta_{dB_{k-1}}\right)$, siendo $T\left(\theta_{dB}\right)$ la función recurrente. De acuerdo la teorema del punto fijo de Banach [45], cuando $k \rightarrow \infty$ la ecuación converge a un único valor $\theta^o_{dB} = T\left(\theta^o_{dB}\right)$. La siguiente ecuación se obtiene cuando se produce esta convergencia:

$$\left(\Delta_{up} + \Delta_{down}\right) \cdot E\left[e\left[k\right]\right] - \Delta_{down} = 0 \tag{8.10}$$

siendo $E\left[e\left[k\right]\right]$ el número medio de errores, esto es, la aBLER. Por tanto, para conseguir que el valor de offset convergido asegure la $aBLER_T$ debe cumplirse:

$$aBLER_T = E\left[e\left[k\right]\right] = \frac{1}{1 + \frac{\Delta_{up}}{\Delta_{down}}}, \tag{8.11}$$

relación que fue presentada en (8.7) y ahora se ha justificado.

La ecuación (8.11) muestra que hay infinitas combinaciones de Δ_{up} y $Delta_{down})$ posibles. Sin embargo, los valores elegidos deben cumplir las dos condiciones suficientes del teorema de Banach:

1. La función $T\left(\theta_{dB}\right)$ debe ser contractiva:

$$T\left(\theta_{dB}\right) \in \left[\theta_{min}, \theta_{max}\right], \ \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right]. \tag{8.12}$$

Dado que $E\left[e\left[k\right]\right]$ está comprendido entre 0 y 1, puede demostrarse que se cumple que:

$$T\left(\theta_{dB}\right) \in \left[\theta_{min} + \Delta_{up}, \theta_{max} - \Delta_{down}\right] \in \left[\theta_{min}, \theta_{max}\right], \ \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right]. \tag{8.13}$$

2. La función debe cumplir:

$$\left|T'\left(\theta_{dB}\right)\right| < 1, \quad \forall \theta_{dB} \in \left[\theta_{min}, \theta_{max}\right] \tag{8.14}$$

Para comprobar esta condición es necesario una expresión tratable para $\mathrm{E}\left[e\left[k\right]\right]$. Para encontrarla, se han obtenido valores de $aBLER(\Gamma, \{\gamma_i\}, \theta)$ mediante simulaciones, y se ha procedido a buscar un modelo para estos valores. El modelo elegido es una función binaria logística modificada:

$$E\left[e\left[k\right]\right] = aBLER(\Gamma, \theta_{dB}) \approx f_m(\theta_{dB}) = \frac{1}{\left(1 + e^{-\alpha_0 \theta_{dB} - \alpha_1}\right)^s}. \tag{8.15}$$

Los valores de α_0 y α_1 se calculan mediante regresión logística. En la Fig. 8.11 puede verse el ajuste a los valores simulados.

Mediante este nuevo modelo se puede evaluar la expresión $T\left(\theta_{dB}\right)$, de acuerdo a (8.9), como:

$$T\left(\theta_{dB}\right) = \theta_{dB} + \left(\Delta_{up} + \Delta_{down}\right) \cdot f_m\left(\theta_{dB}\right) - \Delta_{down} \tag{8.16}$$

Operando, se obtiene que la condición que debe cumplir el sistema para que exista convergencia es:

$$\Delta_{down_{max}} < \frac{2e \cdot aBLER_T}{-\alpha_1}. \tag{8.17}$$

La ecuación (8.17) implica que hay un rango de valores para el cual existe convergencia. En la Fig.8.12 se muestra el proceso de convergencia para varios valores de Δ_{down}, mostrando cómo a partir de un valor máximo ($\Delta_{down_{max}} = 0.52$ en este ejemplo), el sistema empieza a divergir.

8.3.3 Rendimiento del OLLA

Cuando el OLLA converge, el offset instantáneo comienza a oscilar alrededor del valor de convergencia θ^o_{dB} siguiendo las variaciones instantáneas del canal. La amplitud de estas oscilaciones estará relacionada con el valor de $Delta_{down}$ y

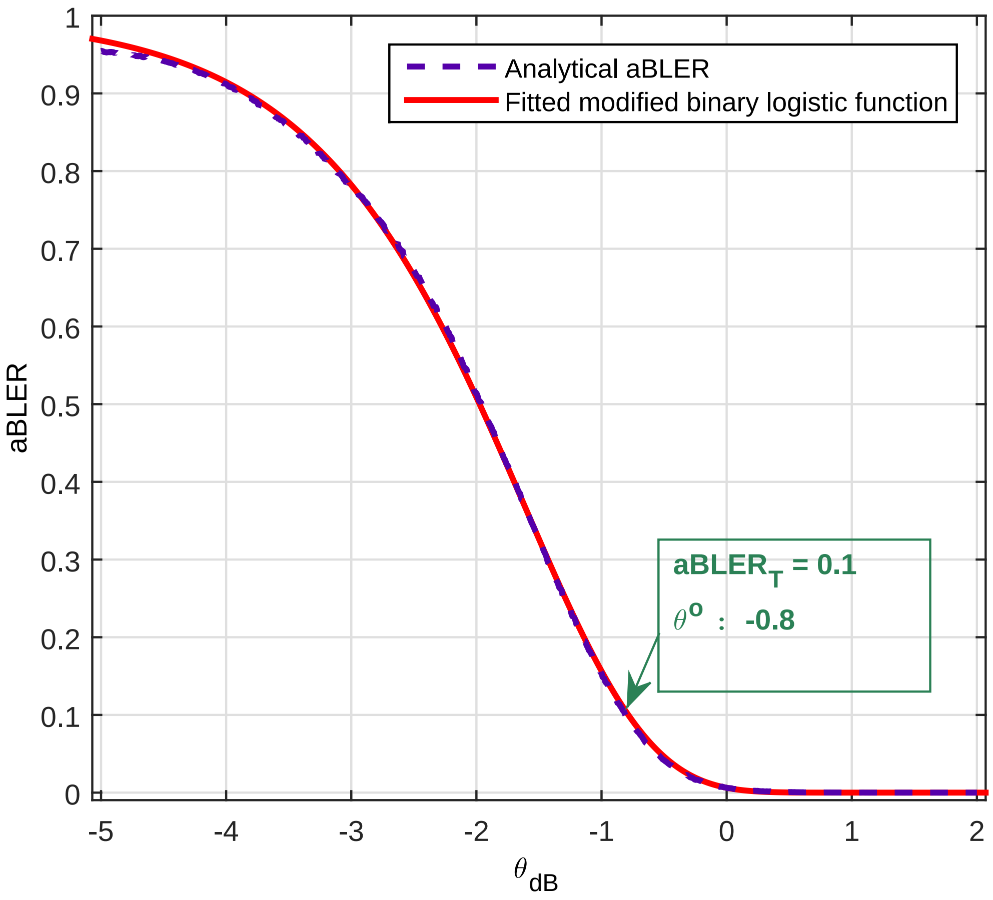

Figure 8.11: Ajuste de la función logística binaria modificada

Delta$_{up}$. Así, valores muy altos de estos parámetros, aunque garanticen convergencia, pueden llevar a grandes variaciones del offset instantáneo; mientras que valores muy pequeños pueden implicar que no se consiga seguir las variaciones instantáneas del canal si estas son muy rápidas.

En la Fig. 8.13 se muestra, para un ejemplo concreto, las variaciones del offset para distintos tamaños del paso; mientras que en la Tabla 8.1 se muestra la eficiencia espectral alcanzada para los distintos valores.

Table 8.2: Rendimiento del OLLA para distintos valores de Δ_{down}

Δ_{down} size (dB)	Spectral Efficiency (bps/Hz)	aBLER
0.001	2.98	0.1
0.01	3	0.1
0.1	2.96	0.1
0.5	2.76	0.1

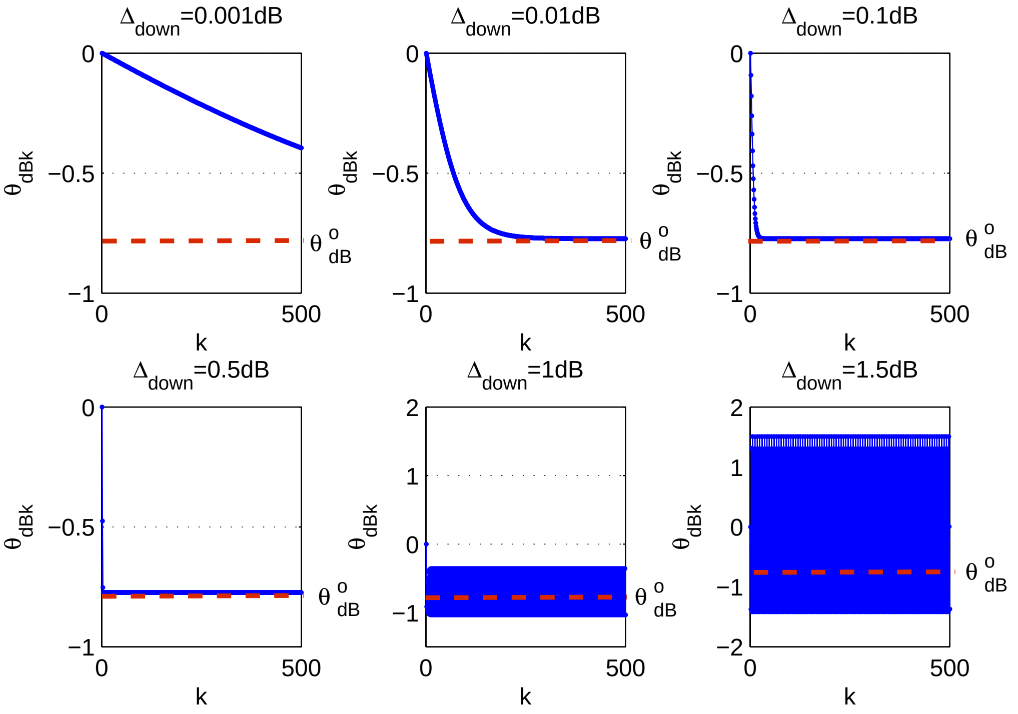

Figure 8.12: Convergencia del OLLA en promedio para diferentes valores de Δ_{down}

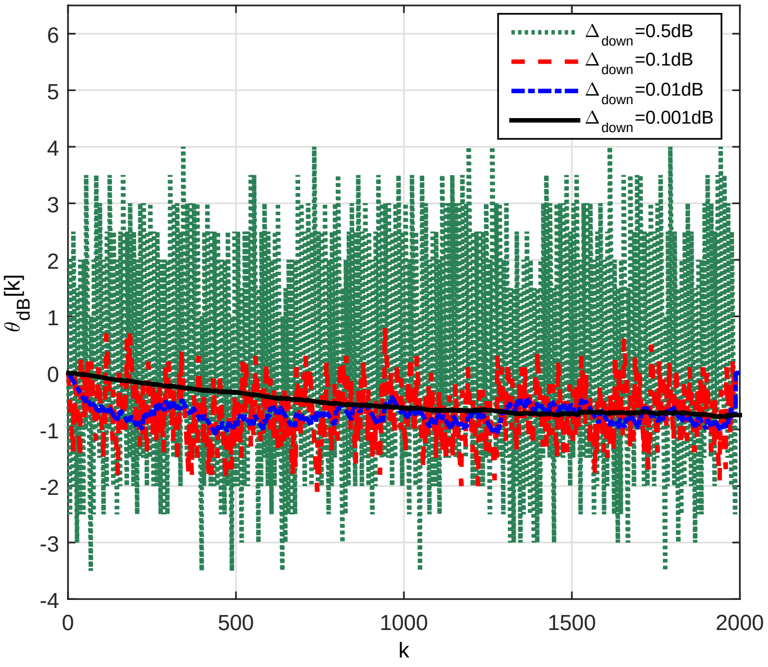

Figure 8.13: Offset instantáneo para canal de Rayleigh plano incorrelado

8.4 eOLLA: Outer Loop Link Adaptation Mejorado para redes celulares

En esta sección se propone una modificación de la técnica OLLA que permite ajustar el tamaño del paso dinámicamente, a la vez que es capaz de actualizar el valor del offset independientemente de si una transmisión ha sido o no realizada. Este algoritmo se llama OLLA mejorado (enhanced OLLA, eOLLA).

8.4.1 Algoritmo eOLLA propuesto

El indicador de error $e\,[k]$ usado en el OLLA tradicional puede ser como un indicador del error instantáneo (1 error, 0 no error). Por tanto, este indicador podría ser reemplazado por un estimador de la BLER instantánea más exacto. Así, para implementar el eOLLA, se asume que se dispone de un modelo de BLER instantánea en función de la SNR instantánea, para cada esquema de modulación y codificación de canal i sobre un canal AWGN ($iBLER_i^{AWGN}\,(\gamma)$).

La implementación del algoritmo OLLA es la siguiente:

1. Si un CRC es extraido, se actualiza el modelo de $iBLER_i^{AWGN}$ correspondiente, de acuerdo al esquema de modulación y codificación i usado en la transmisión (este paso puede ser opcional).

2. Estimar la SNR instantánea en cada instante t, esto es, $\gamma\,[t]$.

3. Para cada intervalo de transmisión t, estimar el valor de BLER instantáneo $B\,[t] = iBLER_i^{AWGN}(\gamma\,[t])$.

4. Obtener el valor del offset como:

$$\theta_{dB}\,[t] = \theta_{dB}\,[t-1] + \Delta_{up} \cdot B\,[t] - \Delta_{down} \cdot (1 - B\,[t])\,. \tag{8.18}$$

A diferencia del OLLA tradicional, donde el valor del offset se actualiza cada instante k que se recibe una transmisión de datos, en el algoritmo eOLLA la actualización se produce cada intervalo de transmisión t, se produzca transmisión o no, lo que hace que para el eOLLA sea más fácil seguir las variaciones instantáneas del canal.

Para entender el comportamiento del eOLLA, el paso 4) puede ser reescrito como:

$$\theta_{dB}\,[t] = \theta_{dB}\,[t-1] + \Delta_{down} \cdot \left(\frac{B\,[t]}{aBLER_T} - 1\right). \tag{8.19}$$

De la ecuación (8.19) se deduce que el incremento a aplicar al offset es proporcional a la diferencia entre al BLER instantánea estimada $B[t]$ y la BLER objetivo, siendo 0 si ambos valores coinciden.

8.4.2 Implementación de la regresión logística

De la descripción del algoritmo eOLLA se desprende que debe disponerse de una expresión para $iBLER_i^{AWGN}(\gamma[t])$, con el objetivo de implementar el paso 1). Como se describió previamente, la BLER instantánea para un esquema de modulación y codificación i en canales AWGN puede modelarse mediante funciones binarias logísticas. Por tanto, la regresión logística puede usarse para encontrar los valores α_{i_0} y α_{i_1} de la ecuación (8.5). En la práctica, este proceso se puede realizar mediante el algoritmo del gradiente descendiente [47]. Este algoritmo se usa para minimizar la función de coste $J(\alpha)$ de la siguiente forma:

$$\alpha := \alpha - \lambda \frac{\partial}{\partial \alpha} J(\alpha) \tag{8.20}$$

donde λ es un parámetro que controla la velocidad de convergencia. Por tanto, en primer lugar es necesario proveer de una función de costes para la regresión logística, dada por [48]:

$$J(\alpha) = y \cdot \log(f_\alpha(x)) - (1-y) \cdot \log(1 - f_\alpha(x)) \tag{8.21}$$

siendo f_α la función logística binaria que modela el mapeo de x en la clase y. Para trasladar la regresión logística a nuestro problema, asociamos la entrada x al valor de SNR instantánea γ, el resultado y de la clasificación a un indicador de error e, y f_α como la función logística binaria de (8.5). Aplicando derivadas parciales a la función de coste, el proceso de regresión binaria para el eOLLA queda:

$$\begin{cases} \alpha_{i_0}^{k} = \alpha_{i_0}^{k-1} - \lambda \cdot \left(\frac{1}{1+e^{-\alpha_{i_0}^{k-1} - \alpha_{i1}^{k-1} \cdot \gamma[k-1]}} - e[k] \right) \\ \alpha_{i_1}^{k} = \alpha_{i_1}^{k-1} - \lambda \cdot \left(\frac{1}{1+e^{-\alpha_{i_0}^{k-1} - \alpha_{i1}^{k-1} \cdot \gamma[k-1]}} - e[k] \right) \cdot \gamma[t-1] \end{cases} \tag{8.22}$$

8.4.3 Rendimiento del eOLLA

En la Fig. 8.14 se muestra, para el caso de eOLLA, el offset instantáneo $\theta[t]$ para las mismas condiciones que en el caso de la Fig. 8.13 ($t = k$). Puede observarse como para el caso de eOLLA la varianza del offset instantáneo es menor. A su vez, en la Tabla 8.3 se presentan los resultados de eficiencia espectral y aBLER, pudiéndose observar como la degradación del primero conforme aumenta el valor de Δ_{down} es menor para la técnica eOLLA.

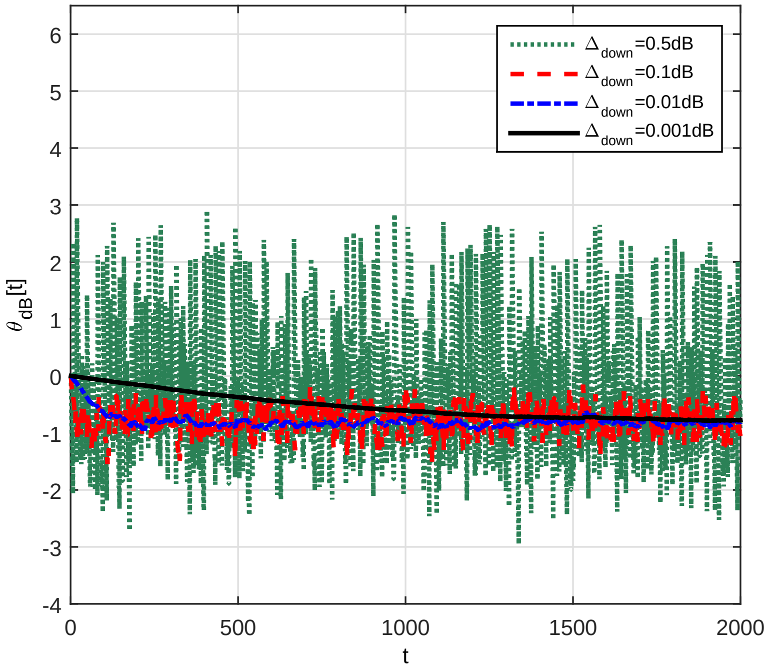

Figure 8.14: Offset instantáneo de eOLLA para canal Rayeligh plano incorrelado.

Table 8.3: Rendimiento de eOLLA para distintos tamaños de Δ_{down}

Δ_{down} (dB)	Eficiencia espectral (bps/Hz)	aBLER
0.001	2.99	0.1
0.01	3	0.1
0.1	2.99	0.1
0.5	2.89	0.1

Sin embargo, en condiciones de baja movilidad existirá correlación en el canal. Además, es habitual la presencia de tráfico a ráfagas, en lugar de transmisión continua. Para evaluar estas condiciones, en la Fig. 8.15 y 8.16 se presenta una comparativa entre el OLLA tradicional y el eOLLA, para $\Delta_{down} = 0.001dB$ y $0.5dB$ respectivamente, para canal Rayleigh plano correlado y distintas tasas de tráfico. Además, en la Tabla 8.4 se muestran los resultados de eficiencia espectral para las distintas configuraciones. Puede observarse como para el caso del eOLLA, el mismo resultado es obtenido independientemente de la tasa de tráfico, ya que su actualización no depende de esta. Por otra parte, puede observarse como el rendimiento del eOLLA es más independiente del tamaño de Δ_{down} que en el caso del OLLA tradicional.

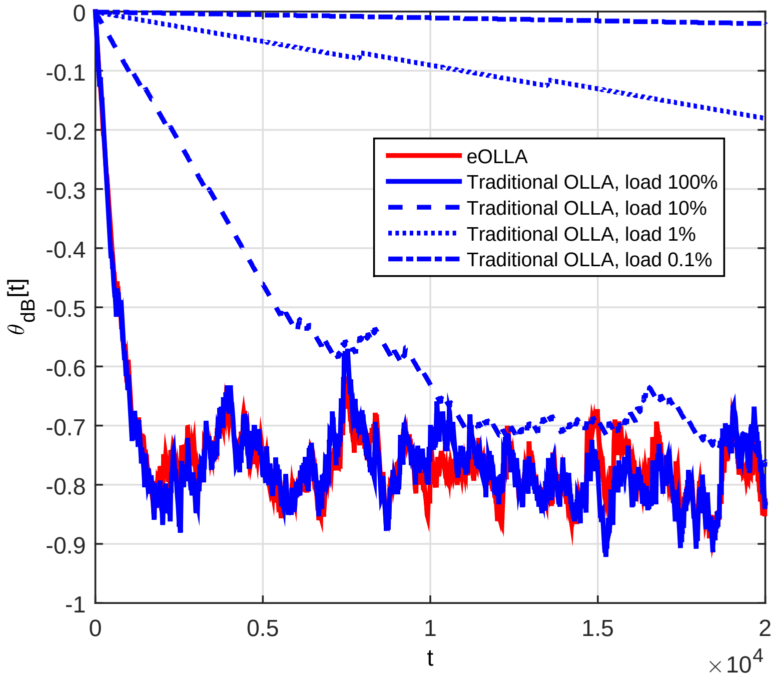

Figure 8.15: Offset instantáneo para canal Rayleigh plano correlado ($f_D = 7Hz$) para OLLA tradicional y eOLLA con $\Delta_{down} = 0.001dB$

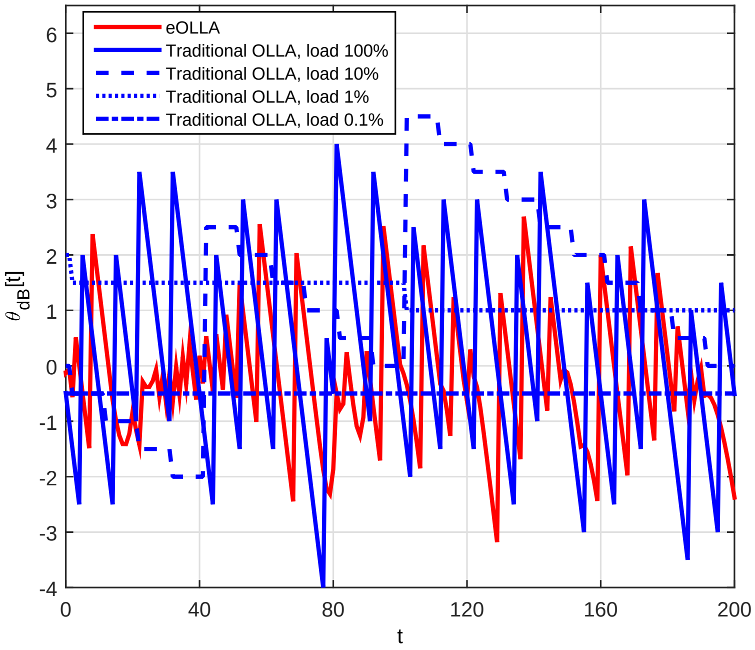

Figure 8.16: Offset instantáneo para canal Rayleigh plano correlado ($f_D = 7Hz$) para OLLA tradicional y eOLLA con $\Delta_{down} = 0.5dB$

Table 8.4: Eficiencia espectral (bps/Hz) para OLLA y eOLLA con canal Rayleigh plano correlado ($f_D = 7Hz$)

	load 100%		load 10%		load 1%		load 0.1%	
Δ_{Down}	OLLA	eOLLA	OLLA	eOLLA	OLLA	eOLLA	OLLA	eOLLA
0.001dB	2.43	2.45	0.24	0.245	0.0215	0.0245	0.0021	0.00245
0.01dB	2.45	2.5	0.244	0.25	0.022	0.025	0.0021	0.0025
0.1dB	2.41	2.4	0.23	0.24	0.021	0.024	0.0020	0.0024
0.5dB	2.22	2.4	0.22	0.24	0.021	0.024	0.0020	0.0024

8.5 Optimización del proceso de modulación y codificación adaptativas con potencia constante

El problema de optimizar el proceso de modulación y codificación de canal adaptativas está muy presente en la literatura [13, 54–58]. Sin embargo, no existen trabajos en los que la maximización de la eficiencia espectral se combine con la existencia de turbo codificación y el establecimiento de una BLER objetivo.

8.5.1 Maximización óptima

El problema de maximizar la tasa de transmisión (throughput) o eficiencia espectral $r(\Gamma, \{\Psi_i\})$ sujeto a una BLER media objetivo, puede formalizarse como:

$$\max_{\Psi_i|_{i=0,1,..,n-1}} \{r(\Gamma, \{\Psi_i\})\} \quad s.t.\ aBLER(\Gamma, \{\Psi_i\}) \leq aBLER_T \tag{8.23}$$

Los umbrales que maximizan esta expresión pueden obtenerse a partir del método de Lagrange, quedando la ecuación de Lagrange como:

$$F(\Gamma, \{\Psi_i\}, \lambda) = r(\Gamma, \{\Psi_i\}) + \lambda \cdot (aBLER(\Gamma, \{\Psi_i\}) - aBLER_T) \tag{8.24}$$

siendo

$$aBLER(\Gamma, \{\Psi_i\}) = \sum_{i=0}^{n-1} \int_{\Psi_i}^{\Psi_{i+1}} iBLER_i^{AWGN}(\gamma) p^o(\Gamma, \gamma) d\gamma$$

el valor de $iBLER_i^{AWGN}(\gamma)$ es estimado mediante funciones binarias logísticas.

Para poder resolver el problema de la optimización, debe encontrarse la forma cerrada de la siguiente expresión:

$$\int iBLER_i^{AWGN}\left(\gamma\right)\cdot p^o\left(\Gamma,\gamma\right)d\gamma =$$

$$-e^{-\frac{\gamma}{\Gamma}}\left(1+\tfrac{1}{\alpha_{i_0}\Gamma-1}\cdot\left(\tfrac{1+e^{\alpha_{i_0}\gamma+\alpha_{i1}}}{1+e^{-\alpha_{i_0}\gamma-\alpha_{i1}}}\right)\cdot_2 F_1\left(1,-\tfrac{1}{\alpha_{i_0}\Gamma};1-\tfrac{1}{\alpha_{i_0}\Gamma};-e^{\alpha_{i_0}\gamma+\alpha_{i1}}\right)\right) \tag{8.25}$$

siendo ${}_2F_1$ la función hipergeométrica:

$${}_pF_q(a_1,...,a_p;b_1,...,b_q;z)=\sum_{n=0}^{\infty}\frac{(a_1)_n(a_2)_n...(a_p)_n}{(b_1)_n(b_2)_n...(b_q)_n}\frac{z^n}{n!} \tag{8.26}$$

con $\left(a\right)_n = a\left(a+1\right)\left(a+2\right)...\left(a+n-1\right)$.

La función óptima debe satisfacer:

$$\left.\frac{\partial F\left(\Gamma,\{\Psi_i\},\lambda\right)}{\partial\Psi_i}\right|_{i=0,1,..,n-1}=0 \text{ and } \frac{\partial F\left(\Gamma,\{\Psi_i\},\lambda\right)}{\partial\lambda}=0. \tag{8.27}$$

Mediante métodos numéricos, es posible encontrar la solución al problema de maximizacíon como:

$$\begin{cases} \frac{\partial F(\{\Psi_i\},\lambda)}{\partial\Psi_0}=-R_0e^{\left(-\frac{\Psi_0}{\Gamma}\right)}-\lambda f_0\left(\Psi_0\right)+\lambda aBLER_T=0 \\ \left.\frac{\partial F(\{\Psi_i\},\lambda)}{\partial\Psi_i}\right|_{i=1,..,n-1}=\left(R_i-R_{i-1}\right)+\lambda e^{\left(\frac{\Psi_0}{\Gamma}\right)}\left(f_{i-1}\left(\Psi_i\right)-f_i\left(\Psi_i\right)\right)=0 \\ \frac{\partial F(\{\Psi_i\},\lambda)}{\partial\lambda}=-e^{\left(\frac{\Psi_0}{\Gamma}\right)}\sum\limits_{i=0}^{n-1}\begin{array}{c}\left[e_2^{\left(-\frac{\gamma}{\Gamma}\right)}F_1\left(1,-\frac{1}{a_i\Gamma};1-\frac{1}{a_i\Gamma};-e^{\left(a_i\gamma+b_i\right)}\right)\right]_{\gamma=\Psi_i}^{\gamma=\Psi_{i+1}} \\ -aBLER_T=0\end{array} \end{cases} \tag{8.28}$$

8.5.2 Maximización subóptima

Una maximización de la tasa de transmisión subóptima consistiría en mover los umbrales de SNR instantánea, manteniendo la distancia relativa entre ellos. Por tanto, los umbrales podría reescribirse como $\Psi_i = \gamma_i \cdot \theta$, donde γ_i es el valor inicial asignado al umbral i, y θ es un offset común. El problema resultante puede formularse como:

$$\max_{\theta}\left\{r\left(\Gamma,\{\gamma_i\},\theta\right)\right\} \quad s.t.\ aBLER\left(\Gamma,\{\gamma_i\},\theta\right)\leq aBLER_T \tag{8.29}$$

con

$$aBLER(\Gamma, \{\gamma_i\}, \theta) = \sum_{i=0}^{n-1} \int_{\gamma_i \theta}^{\gamma_{i+1}\theta} iBLER_i^{AWGN}(\gamma) p(\gamma) d\gamma. \tag{8.30}$$

El valor óptimo del offset puede deducirse a partir de la Fig. 8.17, en la que se muestra un ejemplo de la eficiencia espectral vs. la aBLER para un sistema con turbo decodificador SOVA, en la que se fija una BLER objetivo de 0.1. En ella puede observarse como la máxima eficiencia espectral se obtiene justamente cuando la aBLER coincide con la BLER objetivo, es decir, cuando se cumple la siguiente igualdad:

$$aBLER\,(\Gamma, \{\gamma_i\}, \theta) = aBLER_T. \tag{8.31}$$

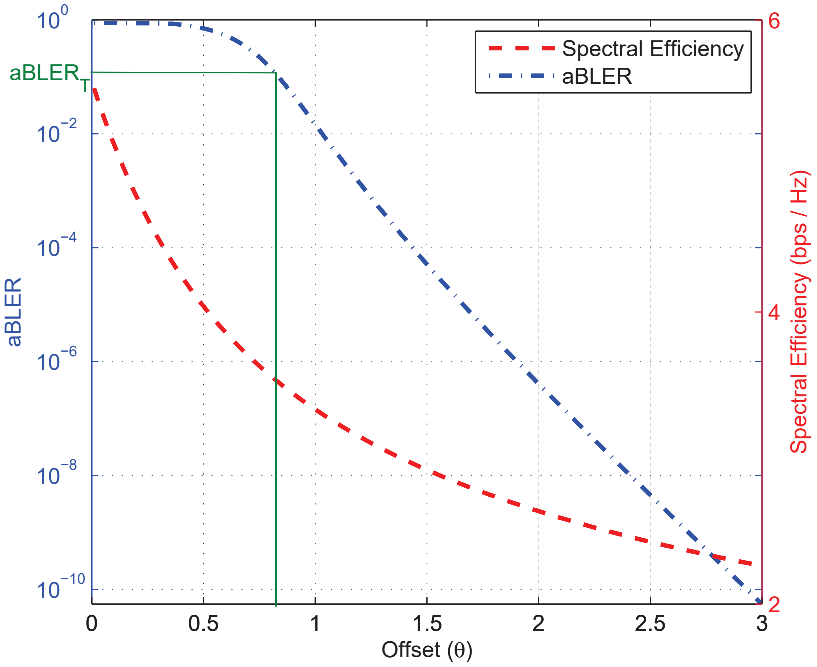

Figure 8.17: BLER vs. eficiencia espectral para diferentes offsets θ

Con estos resultados puede deducirse que la maximización subóptima coincide con el resultado que proporciona la modulación y codificación de canal adaptativas complementadas con la técnica OLLA, ya que esta lo que hace justamente es fijar el valor de la aBLER a la BLER objetivo, mediante unos umbrales de SNR fija y un sólo offset.

8.5.3 OLLA con multi-offset

Si bien se ha mostrado como el OLLA tradicional proporciona una maximización de la eficiencia espectral subóptima, a continuación se comprueba que ocurren en una implementación de la técnica OLLA que implique un offset para cada umbral de SNR. La actualización de cada offset se realizaría de la siguiente forma:

$$\begin{cases} \theta_{i_{dB}}\left[k\right] = \theta_{i_{dB}}\left[k-1\right] + \Delta_{up} \cdot e\left[k\right] - \Delta_{down} \cdot \left(1 - e\left[k\right]\right) \\ \theta_{j_{dB}}\left[k\right] = \theta_{j_{dB}}\left[k-1\right] \;\; \forall j = 0..n-1; \;\; j \neq i \end{cases} \tag{8.32}$$

La diferencia con el OLLA de un sólo offset es el aumento de grados de libertad. Así, existen diversas combinaciones de offsets que cumplen la BLER objetivo, pero no todas esas combinaciones maximizan la eficiencia espectral. Este hecho se muestra en el ejemplo de la Fig. 8.18, en el que por simplicidad sólo se han tenido en cuenta los esquemas de modulación y codificación correspondientes al CQI 4 y 5 (θ_0 y θ_1 respectivamente). En esta figura se muestran por un lado los resultados analíticos obtenidos mediante la ecuación (8.28), y por otro lado los resultados obtenidos mediante simulación. Los resultados analíticos muestran como hay un par de valores (θ_0, θ_1) que maximizan la eficiencia espectral; mientras que los resultados de simulación muestran como el OLLA con multi-offset se desplaza por los valores que cumplen la BLER objetivo, pero no permanece en ninguno de ellos, por lo que no asegura la maximización de la eficiencia espectral.

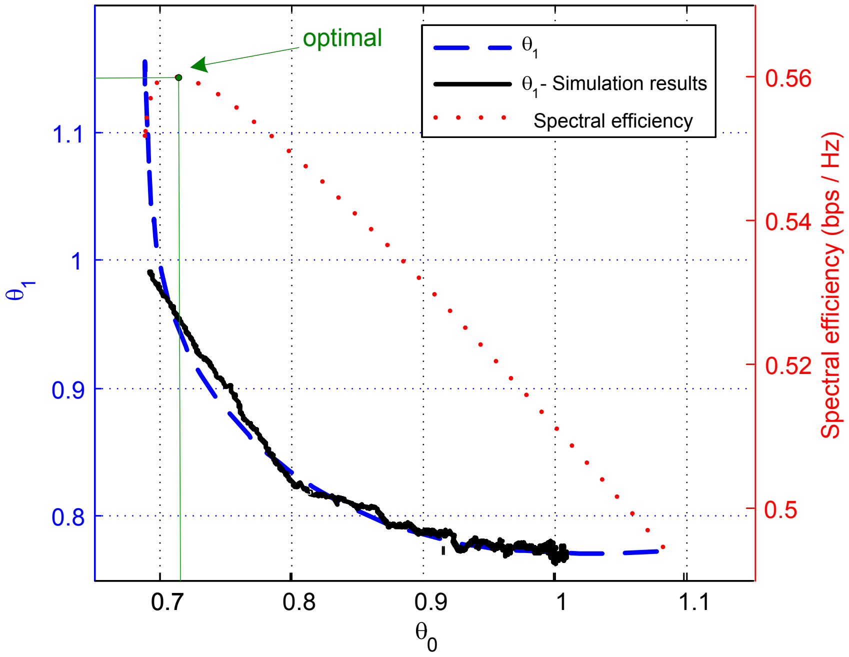

Figure 8.18: Combinción de offsets y su correspondiente eficiencia espectral

8.6 Closed Loop Link Adaptation

La turbo codificación es un elemento con mucha influencia en el rendimiento de la modulación y codificación de canal adaptativas. El comportamiento de los turbo decodificadores depende de muchos elementos, siendo uno de ellos el tamaño de los bloques a decodificar. Es bien conocido que el rendimiento en BER mejora conforme mayor es el tamaño del bloque a decodificar [60], si bien el efecto en la BLER no está tan claro. Además, en LTE debe tenerse en cuenta el efecto de la segmentación [31][61]. La técnica OLLA es capaz de corregir las desviaciones que provoque en los resultados el efecto del tamaño de los Bloques de Transporte (Transport Blocks, TBs). Sin embargo, si en sucesivas transmisiones se produce mucha varianza en estos tamaños, es posible que el OLLA no sea capaz de ajustarse correctamente.

8.6.1 Impacto del tamaño del TB en el rendimiento del turbo decodificador

En la Fig. 8.19 puede observarse el efecto del tamaño del TB para el caso de un esquema de modulación y codificación de canal correspondiente al CQI 12, para canal AWGN y decodificador SOVA. Esta figura muestra como la BLER se va degradando conforme el tamaño del TB aumenta. Nótese que hasta el tamaño de 4490 bits, esta degradación se debe exclusivamente al incremento del tamaño del bloque a decodificar, mientras que para tamaños mayores, en la degradación también influye también el proceso de segmentación [31]. Este tipo de curvas son utilizadas para obtener los umbrales de SNR instantánea de la modulación y codificación de canal adaptativas. Sin embargo, debido a que el tamaño del TB a utilizar en la siguiente transmisión no es conocido a priori, seleccionar la curva exacta a utilizar no es posible.

8.6.2 Implementación de la técnica Closed Loop Link Adaptation

La determinación del tamaño de TB a transmitir depende de muchos factores, como puede ser el estado del canal, número de recursos disponibles o el patrón del tráfico. En esta sección se presenta la técnica de Adaptación del Enlace de Bucle Cerrado (Closed Loop Link Adaptation, CLLA), que es capaz de adaptar el esquema de modulación y codificación de canal al tamaño del siguiente TB a transmitir. Esta técnica necesita del número de Bloques de Recurso Físicos

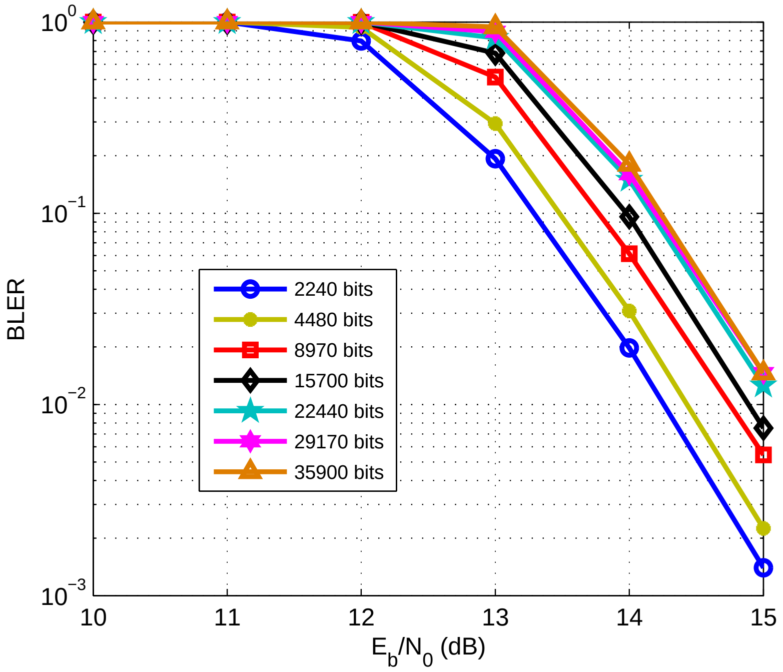

Figure 8.19: BLER para diferentes tamaños de TB correspondientes a CQI 12 y canal AWGN

(Physical Resource Blocks, PRBs) disponibles para la transmisión, por lo que su implementación se llevará a cabo en la estación base, en lugar de en el terminal, como se muestra en la Fig. 8.20.

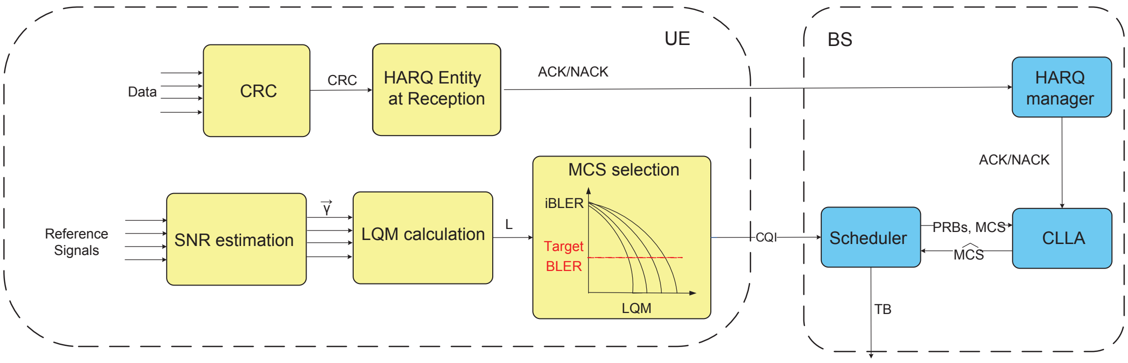

Figure 8.20: Esquema de adaptación del enlace incluyendo CLLA

La técnica CLLA se implementa de la siguiente forma:

1. En la transmisión k, establecer una matriz de offsets $\Omega[k] = \{\omega[k]_{nl}\}_{N\times I}$, con $n = 1..N$ y $l = 1..L$, donde N es el máximo número de PRBs asignables en una transmisión ($N = 100$ end LTE), e I es el mayor esquema de modulación y codificación ($I = 15$ in LTE).

2. Los procesos de retransmisión automática híbridos (Hybrid Automatic Repeat Request, HARQ) reportan al CLLA los Ack/Nack asociados a una transmisión para un número de PRBs n y un esquema de modulación y codificación i:

    - Si un Ack es recibido, el offset $\omega\left[k+1\right]_{ni}$ de $\Omega\left[k+1\right]$ se incrementa en Δ_{up}.
    - Si un Nack es recibido, el offset $\omega\left[k+1\right]_{ni}$ de $\Omega\left[k+1\right]$ se decrementa Δ_{down}.

3. El planificador (scheduler) selecciona el número de PRBs n y un esquema de modulación y codificación tentativo i (basados en los CQIs reportados por el terminal) para usar en la siguiente transmisión $k+1$, y reporta estos valores al CLLA.

4. El CLLA ajusta el esquema de modulación y codificación $\hat{i} = round\left(i+\theta_{ni}\right)$ y devuelve este valor al planificador.

5. El planificador determina el tamaño del TB de acuerdo al número de PRBs n seleccionados para la siguiente transmisión, y al esquema de modulación y codificación $\hat{i}$.

8.6.3 Rendimiento del CLLA

A continuación se muestra una comparativa del rendimiento del OLLA tradicional y el CLLA, para una fuente de tráfico de video en streaming [62], la cual implica transmisión discontinua y diferentes tamaños de los paquetes generados. En Fig. 8.21 se muestra la función de densidad acumulativa de los tamaños de TBs, y la BLER obtenida para el OLLA y el CLLA. Puede observarse como la BLER obtenida por el OLLA no coincide con la BLER objetivo de 0.1 para la mayoría de los tamaños de TB, debido a que esa varianza de tamaños hace que el sistema no converja correctamente. Sin embargo, para el caso del CLLA, al realizar un ajuste en función del tamaño de TB, se consigue que la BLER coincida con la BLER objetivo para todo el rango de tamaños de TBs.

Este desajuste en la BLER para el caso del OLLA implica una pérdida de rendimiento. Así, mientras en la Fig. 8.22 se muestra como el throughput alcanzado cuando se usa el OLLA y el CLLA coincide, en la Fig. 8.23 se muestra cómo el goodput (número de bits recibidos correctamente) se degrada para el caso del OLLA. Esto se debe a que en el caso del OLLA la BLER es mayor para el caso de

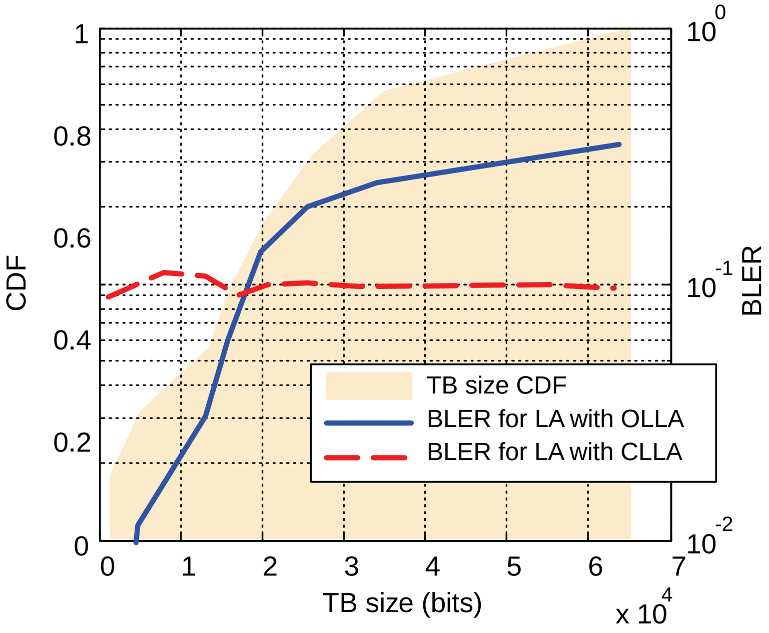

Figure 8.21: BLER para distinos tamaños de TBs con SNR media de 15dB

tamaños de TB grandes, es decir, aquellos con mayor número de bits. Por tanto, en el caso del OLLA existe una mayor probabilidad de recibir los TBs de mayor tamaño erróneamente, mientras que para el caso del CLLA todos los TBs tienen una probabilidad similar de ser recibidos erróneamente.

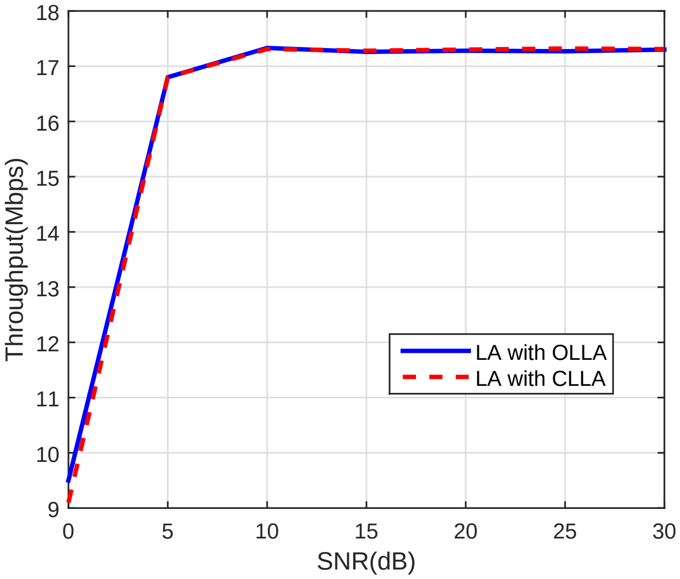

Figure 8.22: Comparativa de throughput para OLLA y CLLA

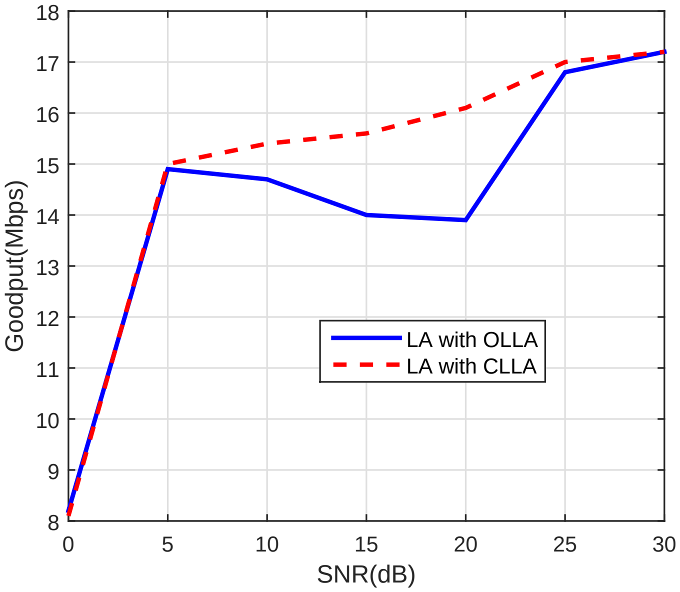

Figure 8.23: Comparativa de goodput para OLLA y CLLA

Appendix A

3GPP-LTE/LTE-A downlink simulator

Techniques studied and proposed in this work have been evaluated in a complete multi-user 3GPP-LTE/LTE-A downlink simulator [36], in order to study their performance in realistic scenarios. This simulator includes most of the features of Physical (PHY) and Medium Access Control (MAC) layers. Furthermore, it includes the LTE-A enhanced Inter Cell Interference Coordination (eICIC) feature [53]. A scheme of this simulator is shown in Fig. A.1.

The MAC layer is made up of a Hybrid Automated Repeat Request (HARQ) process and a channel aware scheduler. This scheduler allocates the data from the MAC layer given to the PHY layer, named Transport Block (TB), by assigning a set of Physical Resource Blocks (PRBs) with a certain Modulation and Coding Scheme (MCS) [39] to each User Equipment (UE) in order to meet some criteria. A Link Adaptation (LA) process is included in order to suggest to the transmitted a suitable MCS to be used.

At the Base Station (BS), the PHY layer is made up of a coder system, which includes a Cyclic Redundancy Code (CRC) and a turbo coder; a Quadrature Amplitude Modulation (QAM) mapper and an Orthogonal Frequency Domain Multiplexing (OFDM) modem. An Additive White Gaussian Noise (AWGN) channel and a multipath channel with temporal fading are included. At the UE, channel estimation and Signal to Noise Ratio (SNR) estimation methods are implemented. With those methods, the Channel State Information (CSI) is reported to the BS [39]. The CSI is composed of the Channel Quality Indicator (CQI), the Precoding Matrix Indicator (PMI) and the Rank Indicator (RI). A QAM demapping and a channel decoding process, which includes a CRC decoder and turbo decoder, is carried out to obtain the transmitted TBs. Finally, a HARQ

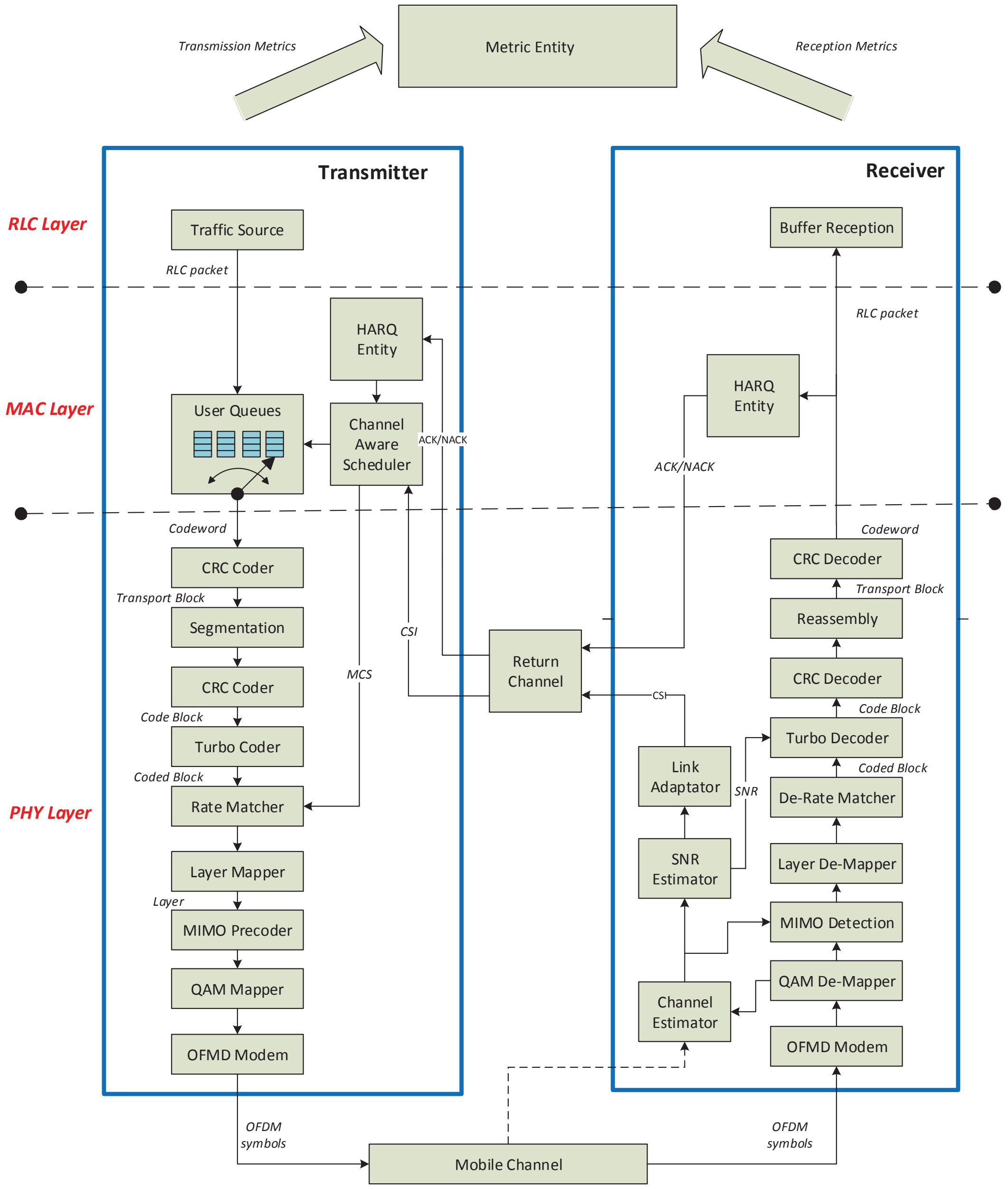

Figure A.1: Scheme of the LTE/LTE-A downlink simulator.

entity manages the report of the Ack/Nack to the BS.

The simulator allows to modify different transmission parameters such as the bandwidth, the channel model, the traffic source or the use or a fixed or adaptive Modulation and Coding Scheme (MCS), among others. This provides a great flexibility to evaluate the proposed algorithms in this work with different transmission configurations.

The different parts of the simulator are briefly described next.

- **Traffic source:** It generates the Radio Link Control (RLC)[63] packets to be transmitted. The source allows to generate different traffic models such as full buffer model or streaming.

- **Channel Aware Scheduler:** This scheduler allocates the TBs by assigning a set of PRBs with a certain MCS [39] to each UE in order to meet some criteria such as fairness or minimizing packet delay.

- **HARQ Transmission Entity:** The Hybrid Automated Repeat Request (HARQ) transmission entity receives the Acknowledgment/Not Acknowledgment (Ack/Nack) reports from the UE, processes them and reports them to the scheduler in order to generate retransmissions (if allowed) or to ask for new transmissions.

- **User Queues:** In this entity packets are generated, according to a certain traffic source, and are queued for each user. Furthermore, time stamps are given to each of these generated packets for monitoring the packet delay. The scheduler indicated to the User Queues entity the size of the data block to be dequeued. This dequeued data block are named codewords. Up to two codewords[1] can be multiplexed by a single user.

- **CRC Coder:** It appends a Cyclic Redundancy Code (CRC) to the codeword [31]. This CRC has a length of 24 bits, which are obtained from the original message. The sequence of bits made up of the bits to be transmitted extracted from the user queues plus this CRC are named Transport Block (TB).

- **Segmentation:** If the size of the TB to be transmitted is higher than the maximum size allowed by the internal interleaver of the turbo coder (6144 bits), a segmentation process is carried out. In this process, the original TB is divided in two or more segments whose sizes are accepted by the internal interleaver. A new CRC has to be appended to each of these segments. These resulting segments are known as Code Blocks (CBs).

- **Turbo Coder:** It adds redundancy to the CBs to protect them to errors in the transmission channel [4], generating the coded blocks. A coding rate of 1/3 is defined for turbo coding in LTE [31].

- **Rate Matcher:** Once the segmentation process (if applicable) and the turbo coding process is carried out, they are generated the coded blocks

[1] To transmit two codewords simultaneously, at least a MIMO 2x2 configuration shall be configured.

to be allocated in the physical resources assigned by the scheduler for the transmission. The ratio between the size of the coded blocks and the amount of physical resources assigned determines the effective coding rate. The rate matching process adapts the coded block to this effective coding rate. The Rate Matcher firstly carries out an interleaving process to the output bits of the turbo coder to add diversity. Next, the adaptation of the transmission rate to the assigned physical resources is carried out. This adaptation can be done in two different ways:

  - *Increasing the coding rate:* In this case part of redundancy bits of the coded block are discarded to achieve an effective coding rate higher than the provided by the turbo coder. This process is known as puncturing.
  - *Decreasing the coding rate:* If the assigned physical resources allows to transmit a number of bits higher than the size of the coded block, part of this coded block is repeated until all these physical resources are filled.

- **Layer Mapper:** Maps the rate matched coded block into layers. The numbers of layers depends on the Multiple Input Multiple Output (MIMO) antenna configuration and the number of transmitting and receiving antennas [12].

- **MIMO Precoder:** Carries out a MIMO precoding, according to the PMI and the RI reported by the UE. This precoding process tries to improve the quality of the reception process.

- **QAM Mapper:** This entity maps the bits of each layer into physical carriers by using QAM symbols. Specifically, LTE uses for data transmission QPSK, 16QAM and 64QAM modulations.

- **OFDM Modem in Transmission:** It generates the Orthogonal Frequency Domain Modulation symbols from the physical carriers, which will be transmitted by the antennas through the mobile channel.

- **Mobile Channel:** It transmits the OFDM symbols to the receptor. There are two possible configurations:

  - *AWGN Channel:* It generates additive white Gaussian noise, so there are no error bursts.

  - *Rayleigh Channel:* Is a channel with temporal fading. Furthermore, it can be configured to be frequency selective. Thus, it can cause error burst in the transmitted data.

- **Return Channel:** Through this channel the UE feeds back the CSI and Ack/Nack reports to the BS.

- **OFDM Modem in Reception:** It demodulates the received OFDM symbols from the mobile channel to obtain the QAM symbols.

- **QAM Demapper:** It maps the QAM symbols from the OFDM modem into bits.

- **Channel Estimator:** This entity estimates the shape of the mobile channel. This estimation can be done ideally or using the information of the Reference Signals (RSs) [39] included in the OFDM symbols.

- **SNR Estimator:** This entity estimates the SNR of each transmitted symbol. This estimation, as in case of the Channel Estimator, can be done ideally or from the information of the RSs. This estimated SNR is used to quantify the quality of the received bits at the input of the turbo decoder, among other reception functions.

- **Link Adaptator:** This entity, from the information of the Channel and the SNR Estimator, generates the Channel State Information (CSI), which is reported to the BS [39]. The CSI is composed of the CQI, the PMI and the RI for the next transmission.

- **MIMO Detection:** By using the estimated channel, detects the layers received from the mobile channel.

- **Layer De-Mapper:** Maps the received layers into coded blocks, which will be decoded later.

- **De-Rate Matcher:** This entity receives the coded blocks with rate matching, and undoes the interleaving process carried out in the transmitter. Next, it locates the bits that were removed in the rate matching process (puncturing) and assignes a random value to them, or combines the bits that where repeated, as appropriate. Then, the coded block transmitted is recovered.

- **Turbo Decoder:** This entity decodes the received bits, which could have suffered puncturing, i.e. all the information about the CB would not be available. The turbo decoder is configurable, so different algorithms can be used.

- **CRC Decoder:** This entity receives the TB, or a segment of it, recalculates the CRC and compares it with the CRC extracted from the received block. In case that both CRCs coincide, the block (TB or segment) is considered to be correct. Otherwise, the block is considered erroneous.

- **Reassembly:** In case that segmentation is carried out in the transmitter, the Reassembly block concatenates all the segments in which the original TB was segmented, previous extraction of the CRC of each segment.

- **HARQ Reception Entity:** This block generates the Ack/Nack reports, according to whether the CRC Decoder decides that the TB is correct or not.

- **Buffer Reception:** Processes the received codewords to recover the RLC packets that were generated by the Traffic Source.

- **Metric Entity:** This block recovers metrics of the transmission process such as Bit Error Rate (BER), BLock Error Rate (BLER), RLC packet delay, throughput, goodput, among others.

Bibliography

[1] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution - From Theory to Practice.* John Wiley & Sons, 2009.

[2] 3GPP TR 36.912, “Feasibility study for Further Advancements for E-UTRA (LTE Advanced),” Tech. Rep., 2011.

[3] A. Goldsmith, *Wireless Communications.* Cambridge University Press, 2005.

[4] C. Berrou, P. Adde, E. Angui, and S. Faudeil, “A Low Complexity Soft-Output Viterbi Decoder Architecture,” in *IEEE International Conference on Communications (ICC)*, vol. 2, May 1993, pp. 737 –74.

[5] G. Proakis, *Digital Communications*, M. Hill, Ed., 2001.

[6] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284 – 287, Mar 1974.

[7] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm with Soft-Decision Outputs and its Applications,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 3, Nov 1989, p. 16801686.

[8] C. Oestges and B. Clerckx, *MIMO Wireless Communications: From Real-World Propagation to Space-Time Code Design*, Elsevier, Ed., 2007.

[9] A. Goldsmith and S.-G. Chua, “Variable-Rate Variable-Power MQAM for Fading Channels,” *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1218–1230, Oct 1997.

[10] A. Goldsmith and S. Chua, “Adaptive Coded Modulation for Fading Channels,” *IEEE Trans. Comm*, pp. 595–610, May 1998.

[11] S. Falahati, A. Svensson, M. Sternad, and T. Ekman, “Adaptive Modulation Systems for Predicted WIreless Channels,” *IEEE Trans. Comm.*, pp. 307–316, Feb 2004.

[12] 3GPP TS 36.213, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10)*, Std., Mar 2011.

[13] C. Kose and D. Goeckel, “On Power Adaptation in Adaptive Signaling Systems,” *IEEE Trans. Commun*, vol. 48, no. 11, pp. 1769–1773, Nov 2000.

[14] S. Chung and A. Goldsmith, “Degrees of Freedom in Adaptive Modulation: a Unified View,” *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1561–1571, Sep 2001.

[15] S. Catreux, V. Erceg, D. Gesbert, and R. Heath, “Adaptive Modulation and MIMO Coding for Broadband Wireless Data Networks,” *IEEE Comm. Mag.*, pp. 108–115, Jun 2002.

[16] H. Bolcskei, “MIMO-OFDM Wireless Systems: Basics, Perspectives, and Challenges,” *IEEE Wir. Comm.*, vol. 13, pp. 31–37, 2006.

[17] W. Cheng and R. Murch, “Adaptive Downlink Multi-User MIMO Wireless Systems for Correlated Channels with Imperfect CSI,” *IEEE Trans. Wir. Comm.*, vol. 5, pp. 2435–2446, May 2006.

[18] C. Chae, A. Forenza, R. Heath, M. McKay, and I. Collings, “Adaptive MIMO Transmission Techniques for Broadband Wireless Communication Systems,” *IEEE Comm. Mag.*, pp. 112–118, May 2010.

[19] G. Oien, H. Holm, and K. Hole, “Impact of Channel Prediction on Adaptive Coded Modulation in Rayleigh Fading,” *IEEE Trans. Veh. Technol.*, pp. 758–769, May 2004.

[20] G. Kramer, M. Gastpar, and P. Gupta, “Cooperative Strategies and Capacity Theorems for Relay Networks,” *IEEE Trans. Inf. Theory*, pp. 3037–3063, Sep 2005.

[21] S. Landstrm, A. Furuskar, K. Johansson, L. Falconetti, and F. Kronestedt, “Heterogeneous Networks - Increasing Cellular Capacity,” Ericsson review, 2011.

[22] F. Blanquez-Casado, F. J. Martin-Vega, D. Morales-Jimenez, G. Gomez, and J. T. Entrambasaguas, “Adaptive SOVA for 3GPP-LTE Receivers,” *IEEE Communications Letters*, vol. 18, no. 6, pp. 991–994, June 2014.

[23] A. Sampath, P. Sarath Kumar, and J. Holtzman, “On Setting Reverse Link Target SIR in a CDMA System,” in *IEEE Vehicular Technology Conference (VTC Spring)*, vol. 2, May 1997, pp. 929–933.

[24] P. Song and S. Jin, “Performance Evaluation on Dynamic Dual Layer Beamforming Transmission in TDD LTE System,” in *International Conference on Communications and Information Technology (ICCIT)*, Jun 2013, pp. 269–274.

[25] K. Pedersen, G. Monghal, I. Kovacs, T. Kolding, A. Pokhariyal, F. Frederiksen, and P. Mogensen, “Frequency Domain Scheduling for OFDMA with Limited and Noisy Channel Feedback,” in *IEEE Vehicular Technology Conference (VTC Fall)*, Oct 2007, pp. 1792 –1796.

[26] F. Blanquez-Casado, G. Gomez, M. d. C. Aguayo-Torres, and J. T. Entrambasaguas, “eOLLA: an enhanced Outer Loop Link Adaptation for Cellular Networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–16, 2016. [Online]. Available: http://dx.doi.org/10.1186/s13638-016-0518-3

[27] P. Robertson, E. Villebrun, and P. Hoeher, “A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain,” in *IEEE International Conference on Communications (ICC)*, vol. 2, Jun 1995, pp. 1009–1013.

[28] L.-H. Ang, W.-G. Lim, and M. Kamuf, “Modification of SOVA-Based Algorithms for Efficient Hardware Implementation,” in *IEEE Vehicular Technology Conference (VTC Spring)*, 2010, pp. 1–5.

[29] O. J. Joeressen, M. Vaupel, and H. Meyr, “Soft-Output Viterbi Decoding: VLSI Implementation Issues,” in *IEEE Vehicular Technology Conference (VTC Fall)*, 1993, pp. 941–944.

[30] T. Cover and J. Thomas, *Elements of Information Theory.* New York: Wiley Interscience, 1991.

[31] 3GPP TS 36.212, *Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)*, Std., Mar 2011.

[32] J. Forney, G.D., “The Viterbi Algorithm,” *Proc. IEEE.*, vol. 61, no. 3, pp. 268–278, March 1973.

[33] J. Woodard and L. Hanzo, “Comparative Study of turbo Decoding techniques: an Overview,” *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2208 –2233, Nov 2000.

[34] J. Ortin, P. Garcia, F. Gutierrez, and A. Valdovinos, “Performance Analysis of Turbo Decoding Algorithms in Wireless OFDM Systems,” *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, pp. 1149–1154, Aug 2009.

[35] C.-M. Wu, M.-D. Shieh, C.-H. Wu, Y.-T. Hwang, and J.-H. Chen, “VLSI Architectural Design Tradeoffs for Sliding-Window Log-MAP Decoders,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 439–447, 2005.

[36] G. Gómez, D. Morales-Jiménez, J. J. Sánchez-Sánchez, and J. T. Entrambasaguas, “A Next Generation Wireless Simulator Based on MIMO-OFDM: LTE Case Study,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, pp. 15:1–15:12, Apr 2010.

[37] 3GPP TS 36.803, *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception; (Release 8)*, Std., Apr 2008.

[38] O. Leung, C.-Y. Tsui, and R. Cheng, “Reducing Power Consumption of Turbo-Code DecoderUsing Adaptive Iteration with Variable Supply Voltage,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 1, pp. 34–41, 2001.

[39] 3GPP TS 36.211, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 10)*, Std., Mar 2011.

[40] M. Lamarca and F. Rey, “Indicators for PER Prediction in Wireless Systems: A Comparative Study,” in *IEEE Vehicular Technology Conference (VTC Spring)*, vol. 2, May 2005, pp. 792 – 796.

[41] S. T. Ericsson and A. Soong, *Effective-SNR Mapping for Modeling Frame Error Rates in Multiple-state Channels*, in 3rd Generation Partnership Project2 3GPP2 Standardization Document, 2003.

[42] K. S. Motorola and J. Zhuang, *Link Performance Abstraction Based on Mean Mutual Information per Bit (MMIB) of the LLR Channel*, in IEEE 802.16 Broadband Wireless Access Working Group Standardization Document, 2007.

[43] M. Lampe, T. GieBel, H. Rohling, and W. Zirwas, "PER-Prediction For PHY Mode Selection in OFDM Communication Systems," in *IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 1, Dec 2003, pp. 25 – 29.

[44] A. Agresti, *Categorical Data Analysis.* New York: John Wiley and Sons, 1990.

[45] S. Banach, "Sur les Operations dans les Ensembles Abstraits et leur Application aux Equations Integrales," Ph.D. dissertation, University of Lwow, Poland (now Ukraine), 1920.

[46] 3GPP TS 36.304, *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode (Release 10)*, Std., Mar 2011.

[47] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing.* New York: John Wiley and Sons, 1993.

[48] A. El-Koka, K.-H. Cha, and D.-K. Kang, "Regularization Parameter Tuning Optimization Approach in Logistic Regression," in *International Conference on Advanced Communication Technology (ICACT)*, Jan 2013, pp. 13–18.

[49] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems," *IEEE Trans. Broadcast.*, vol. 48, no. 3, pp. 223–229, 2002.

[50] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: a transport protocol for real-time applications, RFC-3550*, Std., Ju 2003.

[51] I. Delgado-Luque, F. Blanquez-Casado, M. Fuertes, G. Gomez, M. Aguayo-Torres, J. Entrambasaguas, and J. Banos, "Evaluation of Latency-Aware Scheduling Techniques for M2M Traffic over LTE," in *European Signal Processing Conference (EUSIPCO)*, Aug 2012, pp. 989–993.

[52] D. Drajic, S. Krco, I. Tomic, P. Svoboda, M. Popovic, N. Nikaein, and N. Zeljkovic, "Traffic Generation Application for Simulating Online Games and M2M Applications Via Wireless Networks," in *Conference on Wireless On-demand Network Systems and Services (WONS)*, Jan 2012, pp. 167–174.

[53] K. Pedersen, Y. Wang, S. Strzyz, and F. Frederiksen, "Enhanced Inter-Cell Interference Coordination in Co-channel Multi-Layer LTE-advanced Networks," *Wireless Commun.*, vol. 20, no. 3, pp. 120–127, Jun 2013.

[54] H. Matsuoka, S. Sampei, N. Morinaga, and Y. Kamio, "Adaptive Modulation System with Variable Coding Rate Concatenated Code for High Quality Multi-Media Communication Systems," in *IEEE Vehicular Technology Conferenc (VTC Fall)*, vol. 1, Apr 1996, pp. 487–491.

[55] L. T. Ong, M. Shikh-Bahaei, and J. Chambers, "Variable Rate and Variable Power MQAM System Based on Bayesian Bit Error Rate and Channel Estimation Techniques," *IEEE Trans. Commun*, vol. 56, no. 2, pp. 177–182, Feb 2008.

[56] Z. Dong, P. Fan, W. Zhou, and E. Panayirci, "Power and Rate Adaptation for MQAM/OFDM Systems under Fast Fading Channels," in *IEEE Vehicular Technology Conference (VTC Spring)*, May 2012, pp. 1–5.

[57] A. Gjendemsjo, G. Oien, and P. Orten, "Optimal Discrete-Level Power Control for Adaptive Coded Modulation Schemes with Capacity-Approaching Component Codes," in *IEEE International Conference on Communications (ICC)*, vol. 11, Jun 2006, pp. 5047–5052.

[58] D. Goeckel, "Adaptive Coding for Time-Varying Channels Using Outdated Fading Estimates," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 844–855, Jun 1999.

[59] D. Paranchych and M. Yavuz, "A Method for Outer Loop Rate Control in High Data Rate Wireless Networks," in *IEEE Vehicular Technology Conference (VTC Fall)*, vol. 3, 2002, pp. 1701–1705.

[60] C. Wang, "On the Performance of Turbo Codes," in *IEEE Military Communications Conference (MILCOM)*, vol. 3, Oct 1998, pp. 987 –992.

[61] J. Ikuno, S. Schwarz, and M. Simko, "LTE Rate Matching Performance with Code Block Balancing," *European Wireless Conference*, pp. 1–3, Apr 2011.

[62] *Draft 802.20 Permanent Document, Traffic Models for IEEE 802.20 MBWA System Simulations*, Std., July 8 2003.

[63] 3GPP TS 36.322, *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification (Release 10)*, Std., March 2011.