

POLITECNICO DI TORINO

Collegio di Ingegneria Elettronica, delle Telecomunicazioni e Fisica (ETF)

**Corso di Laurea Magistrale in
COMMUNICATIONS AND COMPUTER NETWORKS
ENGINEERING**

Master Thesis

New Radio Interfaces Beyond 4G



Academic Advisor

Prof. Roberto Garelo

Company Advisor

Ing. Bruno Melis

Candidate

Juan Sebastián Molano Ramos

April 2018

Table of Contents

1. Introduction to 5G communications systems	4
1.1 Requirements	4
1.2 Use Cases	5
1.2.1 Enhanced Mobile Broadband (eMBB)	5
1.2.2 Ultra-Reliable Low Latency Communications (URLLC)	5
1.2.3 Massive Machine Type Communications (mMTC).....	5
2. Basic Structure of 5G NR Radio Interface in 3GPP	6
2.1 Numerology	6
2.2 Waveform.....	6
2.3 Frame Structure.....	8
2.3.1 Mini-Slots	10
2.3.2 Self-Contained Subframe	10
2.4 Resource Block	11
2.5 Resource Grid.....	11
2.7 Channel Coding.....	12
2.7.1 Low Density Parity Check (LDPC)	12
2.7.2 Quasi Cyclic LDPC	13
2.7.1 Rate Matching	16
2.8 Channel Decoding	16
2.8.1 Belief Propagation Algorithm (BPA).....	16
2.8.2 Min-sum algorithm	21
2.8.3 Offset min-sum algorithm	22
2.9 Reference Signal Structure	25
2.9.1 Demodulation Reference Signal (DMRS).....	25
3. Simulink simulation platform	30
3.1 System Functions in Simulink.....	30
3.1.2 Level 2 Matlab S-functions	31
3.2 Development of an LTE simulation chain.....	32
3.3 Development of a New Radio simulation chain.....	38
3.3.1 Outer Modem Comparison (NR vs LTE)	38
3.3.2 Increased Bandwidth and DMRS pattern	42
4. Summary and Conclusions	44
5. Appendix	45
Table A	45
Table B	45
Table C	46
Table D	46

Table E	46
Table F.....	47
Table G.....	48
Table H.....	49
Table I.....	50
6. References	52

1. Introduction to 5G communications systems

1.1 Requirements

Cellular mobile networks have been developed through the years. In the past, these networks were developed just having in mind to optimize a particular service (e.g. voice/ video streams), while other services were supported additionally (e.g. Internet browsing and Internet of Things deployment) [1]. However, nowadays new multiple applications and services are desired by industries and users.

First four mobile generations networks interconnected people by delivering better voice quality and faster data services, it is intended that 5G will do and connect much more. 5G will achieve new levels of efficiency that will benefit the entire society. Some examples applications and services that 5G will bring to life are:

- Immersive entertainment and experiences
- Safe and more autonomous transportation
- Reliable access to remote healthcare
- Improved public safety and security
- Smarter agriculture
- More efficient use of energy
- More autonomous manufacturing
- Sustainable cities and infrastructure
- Digitalized logistics and retail

In order to achieve this services and applications, the International Telecommunication Union Radiocommunications Standardization Sector (ITU-R) has announced multiple 5G mobile networks design goals known as International Mobile Telecommunications 2020 (IMT-2020). According to this, main requirements for 5G NR radio technology have been defined in 3GPP as listed in *Table 1*.

Metric	Goals
Peak Data Rate	20 Gbit/s DL - 10 Gbit/s UL
Peak Spectral Efficiency	30 bps/Hz DL - 15 bps/Hz UL
Bandwidth	400 MHz
Control Plane Latency	10 ms
User Plane Latency	0.5 ms URLLC - 4ms eMBB
Reliability	$1 \cdot 10^{-5}$ for 32 bytes with 1ms UPL
Coverage	164 dB
Battery Life	>10 years (mMTC)
Connection Density	10^6 devices/Km ²
Mobility	500 Km/h

Table 1. 5G Performance Requirements [3].

Not all the requirements need to be satisfied simultaneously. It will depend on the type of application. For example, driverless cars require low latency and high reliability compared with a high definition video streaming.

To meet these goals, the Third Generation Partnership Project (3GPP) has launched the standardization activity for the first phase 5G system in Release 15 named New Radio (NR). A second standardization phase will be needed to fulfill all IMT-2020 requirements.

1.2 Use Cases

5G wireless access is being developed with three broad use case families in mind.

1.2.1 Enhanced Mobile Broadband (eMBB)

This use case, focuses on enhancements of the data rate, capacity, user density, latency, and coverage of mobile broadband access. Enhanced mobile broadband supports high capacity and high mobility (up to 500 km/h) radio access (with 4 ms user plane latency) [5].

1.2.2 Ultra-Reliable Low Latency Communications (URLLC)

This use case will allow to machines and devices, to communicate with ultra-reliability very low latency and high availability. URLLC provides urgent and reliable data exchange. URLLC is ideal for remote surgery, industrial control, vehicular communication, smart grids and public safety applications [5].

1.2.3 Massive Machine Type Communications (mMTC)

This use case, is designed to enable communication between massive low-cost devices. Its aim is to support applications such as smart metering, logistics, and field and body sensors. This will be achieved because NR supports massive small packet transmissions [5].

To integrate these use cases, radio resource management is needed to achieve an optimized network performance. But before it's necessary to have ready deployment scenarios, numerologies, frame structure, new waveform, enhanced carrier aggregation and multiple access.

2. Basic Structure of 5G NR Radio Interface in 3GPP

2.1 Numerology

5G NR will operate in the frequency range from below 1GHz to 100 GHz with different deployments. There will be at higher carrier frequencies a limited coverage area per base station (micro and pico sites) instead at lower carrier frequencies, there will be more coverage per base station (macro site). The licensed spectrum will continue to be the backbone of the wireless network in 5G to provide high service quality and optimal reliability [5].

To enable diverse services on a wide range of frequencies and deployments, a scalable OFDM numerology is required. LTE supports carrier bandwidths up to 20MHz with mostly a fixed OFDM numerology (15kHz subcarrier spacing). 5G instead, will introduce scalable OFDM numerology to support diverse spectrums bands/types and deployments models. For example, 5G NR must be able to operate in *mmWave* bands that have wider channel widths (e.g. up to 400 MHz in a first phase) [2] [5].

The subcarrier spacing is scalable according to 15×2^n kHz, where n is an integer and 15 kHz is the subcarrier spacing used in LTE. The NR OFDM numerologies are shown below in *Table 2*.

Subcarrier Spacing	15 kHz	30 kHz (2 x 15 kHz)	60 kHz (4 x 15 kHz)	15×2^n kHz (n=3,4, ...)
OFDM Symbol Duration	66,67 us	33,33 us	16,67 us	$66,67/2^n$ us
Cyclic Prefix Duration	4,69 us	2,34 us	1,17 us	$4,69/2^n$ us
OFDM Symbol Including CP	71,35 us	35,68 us	17,84 us	$71,35/2^n$ us
Number of OFDM Symbols per Slot	14	14	14	14
Slot Duration	1000 us	500 us	250 us	$1000/2^n$ us

Table 2. Scalable OFDM numerology for NR [5].

The scaling factor 2^n ensures that symbols and slots of different numerologies are aligned in the time domain, which is important to efficiently enable TDD networks. On the other hand, the choice value of n depends on service requirements, carrier frequency, type of deployment, mobility and implementation complexity [5].

2.2 Waveform

OFDM is a mature technology broadly adopted in multiple products due to its properties as low complexity, easy integration with MIMO (Multiple Input Multiple Output), plain channel estimation and so on. It thus strongly motivates 5G NR still choosing OFDM as the basis of new waveform design. This new waveform will have to deal with two issues, the spectral containment and the Peak to Average Power Ratio (PAPR) [1].

Fundamental waveform design enables a unified air interface that supports flexibility and spectral efficiency. OFDM modulation converts high speed data into orthogonal subcarriers and uses Cyclic

Prefix (CP) to deal with Inter Symbol Interference (ISI). In 5G, there will be different applications, each one having very different requirements on air interface technology. As it's known, OFDM used in LTE cannot meet these flexibility requirements since its time-frequency resource allocation is a fixed 15 kHz of frequency division on subcarrier bandwidth. Besides, in order to better exploit the scalable numerology of NR, *Filtered OFDM (F-OFDM)* has been proposed since it can provide coexistence in the same radio channel of different subcarrier spacing and numerology for different services [6].

3GPP has agreed to adopt CP-OFDM with scalable numerology (subcarrier spacing, cyclic prefix) in both UL and DL up to at least 52.6 GHz. Therefore, the overall design is simplified by having the same waveform in both directions, especially with respect to wireless backhauling and device to device communications (D2D). Additionally, there is support for DFT-Spread OFDM in UL for coverage limited scenarios, with single stream transmissions (i.e. without spatial multiplexing) [5].

The spectrum of an OFDM signals decreases rather slowly outside the transmission bandwidth. In LTE, to limit out of band emission, the spectrum efficiency is 90%. That is to say, 100 physical resource blocks (PRB) from 111 PRB are used in a 20 MHz bandwidth. Alternatively, for NR, it has been agreed that the spectrum efficiency will be greater than 90%. To achieve this, windowing and filtering are suitable operations to confine the OFDM signal spectrum in the frequency domain. These operations, that are transparent to the receiver, are applied on top of CP-OFDM at the transmitter side to improve spectrum confinement (e.g. Filtered OFDM) [5].

It can be defined the spectrum utilization Y of a radio interface as:

$$Y = 100 * \frac{\text{Transmission Bandwidth}}{\text{Channel Bandwidth}} [\%] \quad (1)$$

As it was said before in this document, the spectrum utilization in the LTE system is 90%. This came from considering for example the case of 10MHz channel bandwidth, the transmission bandwidth that is spectrally occupied by the LTE signal is equal to 9MHz (i.e. 50 PRBs with size 180 kHz each), so that Y equals to 90%.

Some evaluations in RAN1 show that Y for a NR carrier using *f-OFDM* can be up to 98% of the evaluated channel bandwidths for both DL and UP without complexity and latency constraints.

Filtered OFDM can also facilitate the co-existence of different waveforms with different OFDM parameters. Different sub-band filters are used to create OFDM subcarrier groupings with different inter-subcarrier spacing, OFDM symbol durations and guard times. By enabling multiple parameter configurations, *f-OFDM* is able to provide a more optimum parameter choice for each service group and hence better overall system efficiency [6].

A further advantage of *f-OFDM* is the possibility to multiplex different numerologies in the frequency domain thus realizing the slicing of the radio interface also at PHY level.

2.3 Frame Structure

NR frame structure supports Frequency Division Duplex (FDD) and Time Division Duplex (TDD) transmissions and operation in both licensed and unlicensed spectrum. It allows coexistence with LTE and transmission of variable length (for example, long duration for eMBB and short duration for URLLC), low latency, fast Hybrid Automatic Repeat Request (HARQ) acknowledgment, dynamic TDD. The frame structure design follows three key design principles to reduce interactions between different features and enhance forward compatibility [5].

The first principle establish that transmissions are self-contained. This means that, data in a slot and in a beam, is decodable on its own without dependency on other slots and beams. Therefore, reference signals required for demodulation of data (see e.g. Reference Signal description in chapter 2.9) are included in a given slot and a given beam [5].

The second principle is that transmissions are well confined in frequency and time. By keeping transmissions together it's easier to introduce new types of transmission in parallel with inherited transmission in the future. In NR, frame structure avoids the mapping of control channels across full system bandwidth as occurred in LTE [5].

The third principle is to avoid static and/or strict timing relations across slots and different transmission directions. For example, asynchronous HARQ is used instead of predefined retransmission time [5].

The frame structure is presented below:

- In the time domain, both DL and UL transmissions are organized into *frames* with duration $T_f = 10ms$, consisting of ten *subframes* of duration $T_{sf} = 1ms$ each.

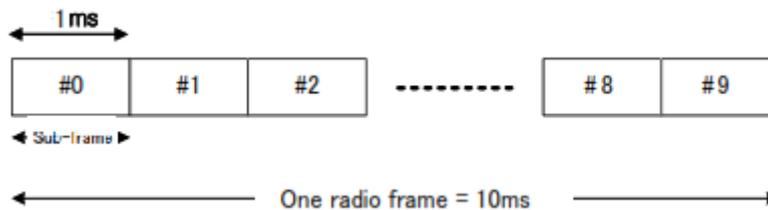


Figure 1. 5G NR Frame duration.

- Each frame is divided into two equally sized half frames of five subframes each.
- Each subframe is divided into $N_{slot}^{subframe,n}$ slots. (n represents the subcarrier spacing configuration)
- Each *slot* is composed of N_{symb}^{slot} consecutive OFDM symbols. (Normally 14 OFDM symbols).
- The number of consecutive OFDM symbols per subframe is:

$$N_{symb}^{subframe,n} = N_{symb}^{slot} \times N_{slot}^{subframe,\mu} \quad (2)$$

W here the possible numerologies are:

The same frame structure can be used for FDD, by enabling simultaneous reception and transmission (DL and UL can overlap in time).

NR frame structure allows a fast HARQ acknowledgment, in which decoding is performed during the reception of DL data and the HARQ acknowledgment is prepared by the UE (User Equipment) during the guard period, when switching from DL reception to UL transmission [5].

To obtain low latency, a slot or a set of slots in case of slot aggregation, is front-loaded with control and reference signals (see e.g. Reference Signal description in chapter 2.9) at the beginning of the slot or slots [5].

2.3.1 Mini-Slots

The new radio slot, can be complemented by mini-slots to support transmissions with a flexible start position and a duration shorter than a regular slot duration. A mini slot can start at any time and can be as short as one OFDM symbol. Indeed, mini-slots can be useful in low latency scenarios, transmissions in the millimeter wave spectrum (mmWave band) and transmissions in unlicensed spectrum as well [5].

In low latency scenarios, transmission needs to begin immediately without waiting for the start of a slot boundary. On the other hand, when transmitting in unlicensed spectrum, it is beneficial to start transmission immediately after LBT. Finally, when transmitting in mmWave band, the large available bandwidth implies that the payload supported by a few OFDM symbols is large enough for many of the user packets [9].

Each mini-slot is also able to carry control signal/channels at the beginning and/or ending of the OFDM symbols [1].

2.3.2 Self-Contained Subframe

In order to get low latency, forward compatibility and many new features in 5G NR, the self-contained integrated subframe is introduced.

The lower latency is achieved by having the data transmission and its acknowledgment all contained in the same subframe. A subframe can contain downlink control burst at the beginning, followed by downlink data burst. This arrangement would give more time budget for the receiver to decode the data and be able to provide feedback within this subframe. (As is explained in chapter 2.9 DMRS can be placed at the beginning of this subframe to facilitate the channel estimation). After a guard period for Tx/Rx switching, the last symbol can be used for uplink control channel [10].

This design reduces DL and UL control channel interferences by requiring all control bursts to be transmitted in the same direction across neighboring cells, thereby allowing more robust link direction switching. Moreover, the dynamic configuration, increases overall network efficiency and capacity by allowing faster switching based on network traffic conditions [2].

2.4 Resource Block

It has been agreed that the concept of physical resource block (PRB) of NR is same as in LTE (i.e. $N_{SC}^{RB} = 12$ consecutive subcarriers in the frequency domain).

The PRB are numbered from 0 to $N_{RB}^{max,\mu} - 1$ in the frequency domain. The relation between the physical resource block number n_{PRB} in the frequency domain and resource elements (k,l) is given by:

$$n_{PRB} = \left\lfloor \frac{k}{N_{SC}^{RB}} \right\rfloor \tag{3}$$

Where one resource element corresponds to one OFDM subcarrier.

NR face the issue of how to multiplex different numerologies. To reduce the complexity and resource fragmentation, a nested structure is adopted which can be applied to the cases where PRBs of different numerology are multiplexed in TDM or FDM [10].

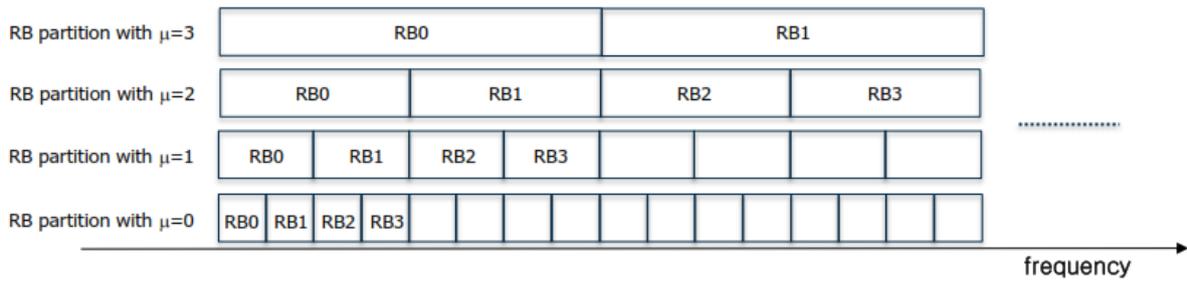


Figure 3. Nested structure of PRB.

The frame structure of NR should give more scheduling flexibility to the networks, regardless of frequency division duplex (FDD) and time division duplex (TDD) spectrum bands. In fact, the design principle of NR frame structure would be general for FDD and TDD [10].

2.5 Resource Grid

The resource grid concept of NR is the same as in LTE. There is one resource grid per antenna port p , per subcarrier spacing configuration μ , and per transmission direction (DL or UL).

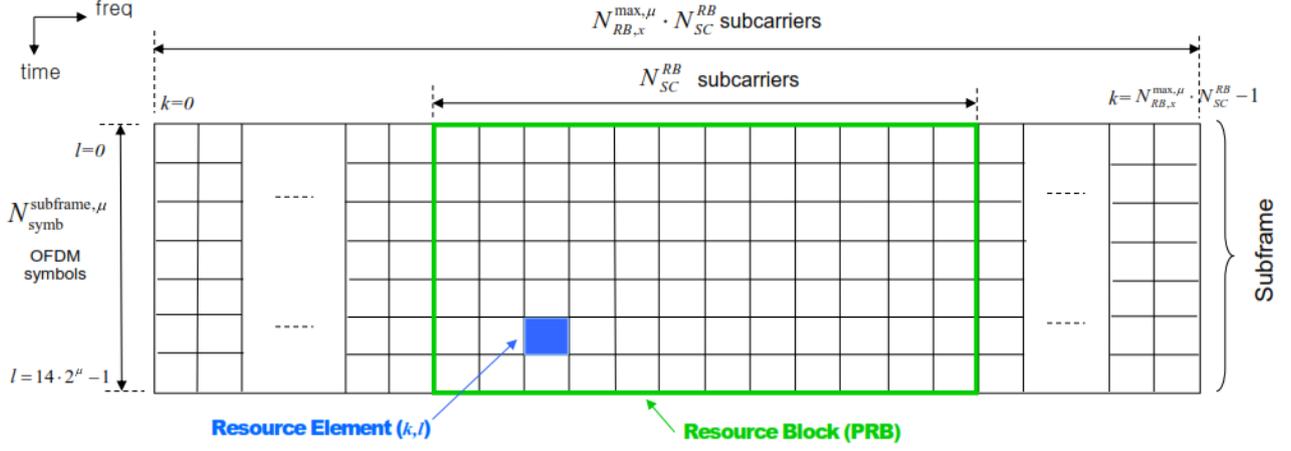


Figure 4. 5G NR resource grid.

For each carrier and numerology, a resource grid of $N_{RB,x}^{max,\mu} \times N_{SC}^{RB}$ subcarriers and $N_{subframe,\mu}^{subframe,\mu}$ OFDM symbols is defined where $N_{RB,x}^{max,\mu}$ ($N_{RB,x}^{min,\mu}$) is the maximum (minimum) number of resource blocks for the radio link x ($x = DL, UL$) and the subcarrier spacing configuration μ . ($N_{SC}^{RB} = 12$ is the number of subcarriers per RB).

The max/min occupied bandwidth (in kHz) can be obtained with the following formula:

$$OBW^{min/max} = \Delta f * N_{SC}^{RB} * N_{RB,x}^{min/max,\mu} \quad (4)$$

Where $\Delta f = 2^\mu * 15$ and $N_{SC}^{RB} = 12$ and $N_{RB,x}^{min/max,\mu}$ is the minimum/maximum number of PRB available for a given numerology.

2.7 Channel Coding

Channel codes are used in all communications systems and their aim is to protect the message sent on the channel from noise, that introduces errors on the original message. In order to preserve the information, some redundancy is introduced. So, the same word in the channel has more bits than the one output by the source. It can be defined a code rate R_c as:

$$R_c = \frac{k}{n}, \quad 0 \leq R_c \leq 1 \quad [5]$$

where k are the information bits generated by the source, whereas n represents the number of bits transmitted on the channel. So, we can obtain the amount of introduced redundancy bits as $r = n - k$.

It has been agreed by RAN1 that NR employs **low-density parity-check (LDPC)** codes for the data channel and polar codes for the control channel. The choice of LDPC was mainly dictated by the easily parallelizable decoder structure that should allow to reach the very high peak throughputs and low latencies for NR.

2.7.1 Low Density Parity Check (LDPC)

LDPC codes were originally proposed by Gallager in early 1960s [10]. They are defined by their parity check matrices \mathbf{H} , with each column representing a coded bit, and each row representing a

parity check equation. They are simply block codes in which the parity matrix \mathbf{H} is not very dense, or in which the presence of non-zero elements is limited. In other words, the parity check matrix is sparse. Because of this peculiarity, these codes are suitable for an iterative decoding approach (Explain later in this chapter) [11].

The parity check matrix \mathbf{H} can be directly defined starting from the Generating Matrix \mathbf{G} through $\mathbf{H}\mathbf{G}^T = 0$. If the generating matrix \mathbf{G} has $k \times n$ dimensions, where k is the information vector dimension whereas n is the code vector dimension, the \mathbf{H} matrix will have $(n - k) \times n$ dimension. From the parity matrix \mathbf{H} , it can be defined a Tanner graph, and this representation is useful to observe the \mathbf{H} composition. Tanner graph is a bipartite graph, this means that there are two groups of set of vertices, one for *variable nodes* and the other one for *check nodes*. The variable nodes represent the columns of the \mathbf{H} matrix while each check node denote rows of the parity check matrix \mathbf{H} . Since the number of columns in \mathbf{H} is equal to a codeword length, variable nodes have a dimension equal to the codeword [11].

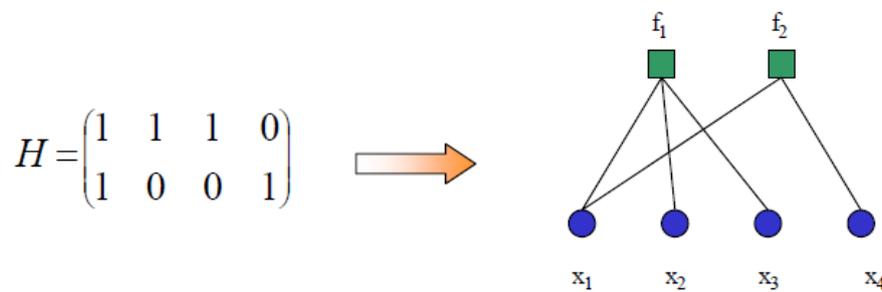


Figure 5. Tanner graph of a parity check matrix [12].

There are two types of LDPC codes: regular and non-regular. Let be w_c the number of ones in one column of the \mathbf{H} matrix and w_r be the number of ones in one row of the \mathbf{H} matrix. A LDPC code is regular if w_c and w_r are constant for all rows and columns of the \mathbf{H} matrix, otherwise is a non-regular [11].

2.7.2 Quasi Cyclic LDPC

The LDPC codes proposed for NR use a quasi-cyclic structure (QC-LDPC), where its parity check matrix is defined by a matrix \mathbf{H} of size $(mb * Z) * (nb * Z)$, which consists of sub-block matrices of size $Z \times Z$, where each sub-block matrix is composed by circularly shifted identity matrices or zero matrices. Wherein, mb, nb and Z are integers larger than one [12]. This type of LDPC codes will allow to cover the full range of information block lengths and code rates needed for NR.

The basic structure of the parity check matrix \mathbf{H} is illustrated below:

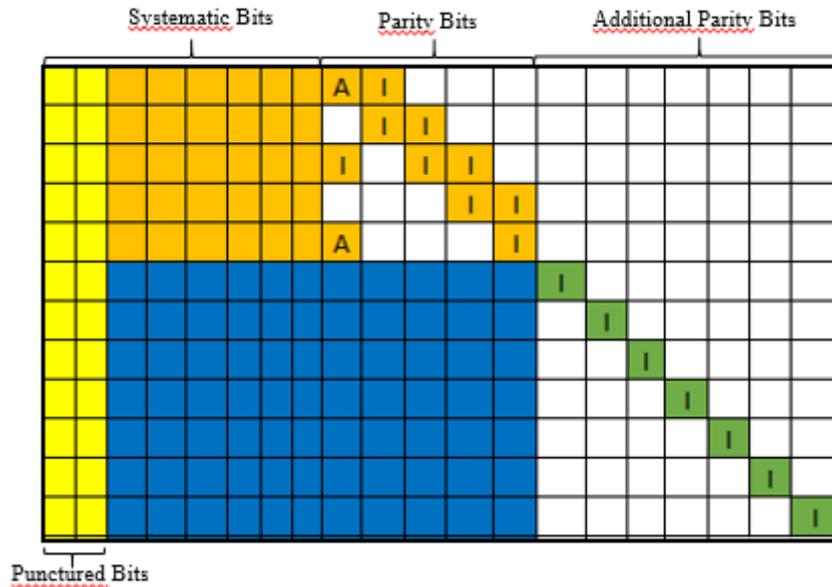


Figure 6. LDPC Code Structure [13].

LDPC codes considered in NR, use a rate compatible structure, as is shown in the figure above. The top left (orange) part of the matrix, defines a high rate code, at a rate $2/3$ or $8/9$. Then, parity bits can be generated by extending the base matrix and including rows and columns on the bottom left part (blue). This allows transmissions at lower code rate, or for generation of additional parity bits such as those used for H-ARQ operation using incremental redundancy (similar to LTE). Since parity check matrix for higher code rates is smaller, decoding latency and complexity decreases for high code rates. High peak rates and low latencies can be achieved through the high degree of parallelism (achievable through the quasi cyclic structure). This allows the LDPC codes to achieve higher coding gains also at low coding rates, making them suitable for use cases requiring high reliability [5].

The *Figure 9* also shows the first $2 \times Z$ systematic bits that are always punctured in the coding scheme adopted by NR (the set of bits corresponding to the yellow columns of the parity check matrix), a structure that has been shown to reduce the threshold of the code. The remaining systematic bits are always transmitted, while some of them may be shortened if an information block length K_{tx} , with $K_{tx} < K$ is desired, where K is the native information block length of the parity check matrix (PCM). The rate may be reduced by transmitting additional parity bits from the incremental redundancy part, as described by the rightmost part of the matrix. In case of shortening, or a desired code rate higher than the highest design rate, some of the parity bits (orange) can be punctured. Check-nodes connected to the variable-nodes of the incremental redundancy part that are not transmitted can be deactivated when decoding to reduce complexity [12].

Quasi-cyclic parity-check matrices are partitioned into square sub-blocks (sub-matrices) of size $Z \times Z$. These submatrices are either cyclic-permutations of the identity matrix or null submatrices. The cyclic-permutation matrix P_i is obtained from the $Z \times Z$ identity matrix by cyclically shifting the columns to the right by i elements. The matrix P_0 is the $Z \times Z$ identity matrix. Quasi-cyclic LDPC codes are conveniently described through a base matrix, which is a matrix where each integer i denotes the cyclic-permutation matrix P_i . Entries with $i = -1$ in the matrix denote null (zero) submatrices [12].

An example of proposed LDPC base graphs with the related parameters is described in the following table [12]:

	$K_{b,max}$	$K_{b,min}$	K_{max}	K_{min}	N_{punc}	N_{deg1}	R_{max}	R_{min}	Z
Base graph 1	32	22	16384	176	2xZ bits	2xZ bits	8/9	1/3	Lift 1: 8, 12, 16, 24 Lift 2: 32, 48, 64, 96 Lift 3: 128, 192, 256, 384, 512
Base graph 2	24	12	12288	96	3xZ bits	3xZ bits	8/11	8/31	Lift 1: 8, 12, 16, 24 Lift 2: 32, 48, 64, 96 Lift 3: 128, 192, 256, 384, 512
Base graph 3	10	6	5120	48	2xZ bits	Z bits	2/3	1/4	Lift 1: 8, 12, 16, 24 Lift 2: 32, 48, 64, 96 Lift 3: 128, 192, 256, 384, 512
Base graph 4	6	4	3072	32	2xZ bits	Z bits	1/2	1/5	Lift 1: 8, 10, 12, 14 Lift 2: 16, 20, 24, 28 Lift 3: 32, 40, 48, 56 Lift 4: 64, 80, 96, 112 Lift 5: 128, 160, 192, 224 Lift 6: 256, 320, 384, 448, 512

Table 5. LDPC base graphs [13].

The parameters shown in Table 6 are:

- $K_{b,max}$ is the maximum number of information nodes in the base graph.
- $K_{b,min}$ is the minimum number of information nodes in the base graph, after shortening is applied. The maximum number of bits that can be shortened is therefore $(K_{b,max} - K_{b,min}) * Z$.
- K_{max} is the maximum number of information bits supported.
- K_{min} is the minimum number of information bits supported after shortening is applied.
- N_{punc} is the number of punctured systematic bits after lifting.
- N_{deg1} is the number of degree-1 variable nodes included in the parity-check matrix with the highest rate.
- R_{max} is the maximum code rate supported.
- R_{min} is the minimum code rate supported.
- The Z values shown in the table are the Z values for which the base graph can be lifted.

For each base graph; a specific parity check matrix is obtained by selecting a lifting size Z with a corresponding base matrix and replacing each entry with the corresponding $Z \times Z$ matrix. The parity check matrix with exactly the desired code rate and information block length may then be constructed from the parity check matrices described in the above table through the rate matching [12]. The final base graphs that have been standardized by 3GPP for NR are provided in the 3GPP specifications [TS 38.214].

2.7.1 Rate Matching

The rate matching for LDPC code is circular buffer based (same concept as LTE). The circular buffer is filled with an ordered sequence of coded bits. The coded bits are read out sequentially from the circular buffer until the number of read out bits is equal to the number of bits to be transmitted, starting with the assigned bit location. If more bits are needed to be transmitted after the last bit in the circular buffer is selected, the procedure continues by circularly reading the first bit in the circular buffer. After the circular buffer, a bit interleaver is adopted to improve performance for higher order modulation in a fading channel [14]. An example is presented in *Figure 10*.

The NR LDPC code has been designed with $2Z$ built-in puncturing of systematic bits where Z is lifting size for LDPC code. That is first $2Z$ systematic bits are not transmitted because they are not inserted in the circular buffer.

The performance would be optimized if the start bit location in the circular buffer is selected in a proper way. It can be considered the lifting value Z to compute the starting bit location. If a starting bit location is multiple of Z , it would be beneficial to decoder complexity and latency aspect. Therefore, it would be preferable that a starting bit location are defined by uniformly distributed as below [14]:

$$S_i = \left\lfloor \frac{a \cdot N_{cb}}{4b \cdot Z} \right\rfloor \times Z \quad [8]$$

where a and b are integer numbers that depend on the redundancy version $RV_i = \{0,1,2,3\}$ and the N_{cb} is circular buffer size. For non-limited buffer, N_{cb} is 66 and 50 for base graph 1 and base graph 2 selected in 3GPP, respectively.

2.8 Channel Decoding

The technique for decoding LDPC codes is based on iterative decoding. In this section it will be presented three techniques for channel decoding in NR as: Belief Propagation algorithm (BPA), Min-Sum algorithm and Offset Min-Sum algorithm.

2.8.1 Belief Propagation Algorithm (BPA)

This algorithm is also known as Message Passing Algorithm (MPA) and the Tanner graph is essential to understand its behavior.

As first step all variable nodes send a message q_{ji} (ji means that the message is sent from variable node j to the check node i), that describes its actual state to all check nodes which are connected. At the beginning of the algorithm, these messages will be equivalent to the value of each bit received on the channel, because at this time is the only known information. The check nodes, after processing the messages sent by each variable node, send a response message r_{ij} (from check node i to variable node j) to each connected variable node. This response message is an evaluation of the actual state of the variable node and is calculated considering all variable nodes connected to the considered check node except for the one for which the response is going to be sent. The information that a check node acquire is called extrinsic information, this comes from the fact that a variable node receives an answer that is estimated using all sources except of its own f information. For a

codeword belonging to $(0,1)$, the r_{ij} message in case of hard decoding, consist in the parity check of the considered variable node. The information content of an r_{ij} check node message, would be less if the information of the variable node being analyzed is considered, because the variable node already knows its actual state. This process is called horizontal step due to the fact that rows of the parity matrix H are analyzed. In a subsequent step through the extrinsic information received from the check nodes, each variable node updates the messages sent to the check nodes. To make this, each variable node considers every check node connected to him, except the one is updating. This operation is called vertical step because the parity matrix H is analyzed by columns.

Once the variable nodes are updated, the algorithm has completed one iteration and can stop; when the received information is correct (i.e. all the parity check equations are satisfied) or re-start a new iteration because the received codeword is not correct. The parity check matrix H plays a fundamental role because is used for the check parity of the received codeword through the relation $Hx^T = 0$, where x is the decoded codeword [11].

This set of operations is one iteration of the algorithm. A summary of the BPA is shown below [11]:

- The algorithm starts at variable nodes
 1. Variable node j sends a message q_{ji} (from variable node j to check node i . Remember that i is the row index of H and j the column index) about its status to each check node connected to it.
 2. Check nodes evaluate information from each variable node connected to them and compute a response message r_{ij} to each variable node. To consider this response they consider all variable nodes connected to them except the one they are computing the response to.
 3. Variable nodes receive the message from their check nodes and use this information to evaluate their status (marginal computation). After updating their status, variable nodes can send their message to check nodes again and the algorithm can go on iterative. (Iteration: return to step 1).

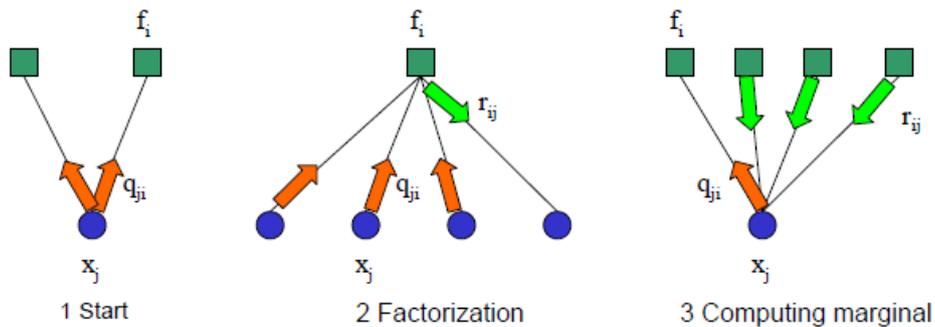


Figure 7. Belief propagation algorithm.

The BPA can have two decisions approach: hard decision and soft decision. Let analyze first the hard-decoding approach because of its simplicity as a first step towards the implementation of soft decoding, which is actually used in practice.

In BPA algorithm with soft decoding, messages sent by either variable nodes and check nodes are probabilities, which actually are real numbers, bring out more complex computations. A simplified and faster version of the iterative decoding is the hard-decoding approach. In this case, messages are not real valued probabilities but instead just binary values 0 or 1.

Remember that each row of H corresponds to a parity equation and if a received word y is a codeword the condition $Hy^T = 0$ holds.

BPA with hard-decoding is presented below [11]:

1. Variable nodes send their messages q_{ji} to check nodes. At first step these messages are the received bits on the channel. Hence, each variable node sends his bit value to his connected check nodes.

Each check node computes its corresponding parity check equation.

2. Check nodes receive the values q_{ji} from the respective variable nodes and compute a response message r_{ij} for each variable node connected to them. This operation is done according to:

$$r_{ij}^{(k)} = \sum_{j^i \in X_{i/j}} q_{j^i}^{(k-1)} \oplus \quad [9]$$

This formula can be explained as: to compute an answer, each check node exploits the information received from other variable nodes connected to it excluding the q_{ij} message received from the variable node x_j . Hence, the response message r_{ij} is the modulo 2 sum of the other messages q_{ij} .

3. Once variable nodes have received the answer messages from check nodes, they have more information to decide about the corresponding bit values. So, given the column weight w_c of a column of H and hence the degree of a variable node x_j (the degree value of a variable node is the number of check nodes connected to it), the received bit is assumed to be 0 or 1 following the majority decision. The number of votes is $w_c + 1$, because each variable node uses both the bit value received on the channel and the w_c responses r_{ij} received from the check nodes.
4. Variable nodes then update their value and send their new messages q_{ji} to check nodes. Another iteration of BPA then starts from the second step.

The hard-decoding approach has an issue when the number of 0 or 1 votes is equal, in other words there is the same number of 0s and 1s and so the decoder cannot take a decision. To solve this the soft decision approach is used.

The BPA with soft-decoding uses probabilities as messages instead of bits used with hard-decoding. In this case, it is no more taken a decision over received bits, but instead it is calculated the probability that the j -th bit of the received vector is 1 or 0 given the received value y_j [11]:

$$p(x_j = 1|y_j) = \frac{1}{1 + e^{-\frac{2y_j}{\sigma^2}}} \quad [10]$$

$$p(x_j = -1|y_j) = \frac{1}{1 + e^{\frac{2y_j}{\sigma^2}}} \quad [11]$$

The expression above is the a-posteriori probability for the received bit given and σ^2 represents the variance of an AWGN (Additive White Gaussian Noise) noise.

The decoding process can start once these quantities have been calculated for each bit. So, the variable nodes send messages to check nodes and at the first iteration these messages are equal to the probabilities given in formula [9] and [10]. It can be noticed that the information on the Tanner graph is duplicated with respect to the hard-decoding process, because it is necessary to send the probabilistic information about states 0 and 1. Then, check nodes receive the messages and compute a response for each connected variable node. For making this are taken all the variable nodes connected, except for the one which it is being calculated the reliability [11]. The following expressions represents the way the check node calculate its response, which represents the probability that a variable node is 0 or 1.

$$r_{ij}^{(k)}(0) = \frac{1}{2} + \frac{1}{2} \prod_{j' \in X_i \setminus j} (1 - 2q_{j'i}^{(k-1)}) \quad (12)$$

$$r_{ij}(1) = 1 - r_{ij}^{(k)}(0) \quad (13)$$

where the expression $X_i \setminus j$ means that all variables nodes connected to the check node are considered except the variable node j . k represents the actual iteration. Once, all r_{ij} probabilities have been calculated, these messages are sent to variable nodes; in this phase the messages q_{ji} are updated. In this case the parity matrix H is analyzed by columns (vertical step) [11].

$$q_{ji}^{(k)}(0) = \alpha_{ji} p_j(0) \prod_{i' \in F_{j \setminus i}} r_{i'j}^{(k)}(0) \quad (14)$$

$$q_{ji}^{(k)}(1) = \alpha_{ji} p_j(1) \prod_{i' \in F_{j \setminus i}} r_{i'j}^{(k)}(1) \quad (15)$$

where $F_{j \setminus i}$ are the all check nodes connected to a variable node j except the check node i . Also α_{ji} is a constant normalization term such that $q_{ji}^{(k)}(0) + q_{ji}^{(k)}(1) = 1$. Moreover, is also calculated a pseudo-posteriori probability relative to the bits [11]:

$$Q_j^{(k)}(0) = \beta_j p_j(0) \prod_{i \in F_j} r_{ij}^{(k)}(0) \quad (16)$$

$$Q_j^{(k)}(1) = \beta_j p_j(1) \prod_{i \in F_j} r_{ij}^{(k)}(1) \quad (17)$$

Where β_j is a normalization constant. Now a variable node uses information from all check nodes connected to it. This is done under the independency assumption. To compute this probability, is used the information coming from check node. Once the a-posterior pseudo-probability is calculated, this algorithm can take a decision on the value of the bit following the following relation [11]:

$$\hat{x}_j^{(k)} = \begin{cases} 1, & Q_j^{(k)}(1) > Q_j^{(k)}(0) \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

when the relation $Hx^T = 0$ is satisfied, the algorithm can be finished because the word is a codeword, instead if the equation above is not satisfied then a new iteration of decoding until up to the limit iteration is executed.

Let's see the soft decoding approach in the logarithmic domain using Log Likelihood Ratios (LLRs). With soft decoding approach the performance is better than using hard decoding at the price of the computational complexity (more product operations). A hardware implementation of this approach is quite difficult and expensive. So, to make it less computational complex, the log-likelihood version of the algorithm is used.

This approach has the same performances of the BPA in the linear domain but allows to avoid many product operations and the normalization of the probabilities because the algorithm can manage both small and large numbers.

Let us define some likelihood ratios (LLR) variables to analyze the BPA algorithm in the log-likelihood version:

$$LLR(p_j) = \ln \left(\frac{p(x_j = -1|y_j)}{p(x_j = 1|y_j)} \right) = -\frac{2y_j}{\sigma^2} \quad (19)$$

$$LLR(r_{ij}) = \ln \left(\frac{r_{ij}(0)}{r_{ij}(1)} \right) \quad (20)$$

$$LLR(q_{ji}) = \ln\left(\frac{q_{ji}(0)}{q_{ji}(1)}\right) \quad [21]$$

$$LLR(Q_j) = \ln\left(\frac{Q_j(0)}{Q_j(1)}\right) \quad [22]$$

So, variable nodes send their $LLR(q_{ji})$ messages that are the log-likelihood ratio posterior probabilities at the first step, but now a variable node sends only one message because the LLR contains both the posterior probability relative to zero and one. This is a further reduction of complexity.

At the first step, the LLR of the bit received on the channel is sent by the variable nodes to the check nodes,

$$LLR(q_{ji})^{(0)} = LLR(p_j)$$

Once all variable nodes have sent their messages, the processing of the r_{ij} messages at check nodes can start. It is known that:

$$r_{ij}^{(k)}(0) = \frac{1}{2} + \frac{1}{2} \prod_{j' \in X_{i \setminus j}} (1 - 2q_{j'i}^{(k-1)}(1)) \quad [12]$$

To obtain the LLR of $r_{ij}(x_j)$, it can be derived as follows:

$$2 r_{ij}^{(k)}(0) = 1 + \prod_{j' \in X_{i \setminus j}} (1 - 2q_{j'i}^{(k-1)}(1)) \quad [23]$$

$$1 - 2 r_{ij}^{(k)}(1) = \prod_{j' \in X_{i \setminus j}} (1 - 2q_{j'i}^{(k-1)}(1)) \quad [24]$$

It is also known that:

$$\tanh\left(\frac{1}{2} \log\left(\frac{p_j(0)}{p_j(1)}\right)\right) = 1 - 2p_j(1) \quad [25]$$

So, at the end we have:

$$\tanh\left(\frac{1}{2} \log\left(\frac{r_{ij}^{(k)}(0)}{r_{ij}^{(k)}(1)}\right)\right) = \prod_{j' \in X_{i \setminus j}} \tanh\left(\frac{1}{2} \log\left(\frac{q_{j'i}^{(k-1)}(0)}{q_{j'i}^{(k-1)}(1)}\right)\right) \quad [26]$$

$$\tanh\left(\frac{1}{2} LLR(r_{ij}^{(k)})\right) = \prod_{j' \in X_{i \setminus j}} \tanh\left(\frac{1}{2} LLR(q_{j'i}^{(k-1)})\right) \quad [27]$$

After these passages, we can write $LLR(r_{ij}^{(k)})$ as:

$$LLR(r_{ij}^{(k)}) = 2 \tanh^{-1}\left(\prod_{j' \in X_{i \setminus j}} \tanh\left(\frac{1}{2} LLR(q_{j'i}^{(k-1)})\right)\right) \quad [28]$$

From equation 28, it can be seen that there are still some products and there are also two hyperbolic tangents having high computational complexity. This is a problem because the hardware implementation cannot be done for a decoder using high complexity [11].

Once computed all $LLR(r_{ij})$, variable nodes have to update their $LLR(q_{ji})$ and compute the pseudo posterior probabilities.

$$LLR(q_{ji})^{(k)} = LLR(p_j) + \sum_{i' \in F_{j \setminus i}} LLR(r_{i'j}^{(k)}) \quad [29]$$

$$LLR(Q_j) = LLR(p_j) + \sum_{i \in F_j} LLR(r_{ij}^{(k)}) \quad [30]$$

So, we can summary the decoding in the log domain in the following steps [11]:

1. Compute the log-posterior probabilities and set all the $LLR(q_{ji})$ equals to the posterior probabilities received on the channel (the decoder only has this information).

$$LLR(p_j) = -\frac{2y_j}{\sigma^2} \quad [31]$$

$$LLR(q_{ji})^{(0)} = LLR(p_j) = -\frac{2y_j}{\sigma^2} \quad [32]$$

2. Compute all response messages (Horizontal step)

$$LLR(r_{ij}^{(k)}) = 2 \tanh^{-1} \left(\prod_{j' \in X_{i \setminus j}} \tanh \left(\frac{1}{2} LLR(q_{j'i}^{(k-1)}) \right) \right) \quad [33]$$

3. Update all $LLR(q_{ji})$ messages using both response messages and log-posterior probabilities:

$$LLR(q_{ji})^{(k)} = LLR(p_j) + \sum_{i' \in F_{j \setminus i}} LLR(r_{i'j}^{(k)}) \quad [34]$$

And also compute a log-pseudo posterior probability

$$LLR(Q_j^{(k)}) = LLR(p_j) + \sum_{i \in F_j} LLR(r_{ij}^{(k)}) \quad [35]$$

4. Finally evaluate the received codeword and take a decision. If $H\hat{x}^T = 0$ exit, otherwise go to step 2.

$$\hat{x}_j^{(k)} = \begin{cases} 1, & LLR(Q_j^{(k)}) < 0 \\ 0, & \text{otherwise} \end{cases} \quad [36]$$

2.8.2 Min-sum algorithm

As described before, the soft decoding approach in the log domain avoids the computation of many products, but the presence of hyperbolic tangents is still a problem in terms of computational complexity. For this reason, is necessary to have algorithms that approximates the same behavior. The Min-sum algorithm is an approximation algorithm with less complexity than the BPA in the log domain but presents a slight performance drop [11].

As you can see in equation 33, the most complex part of the BPA is the check nodes processing. The idea of the Min-sum algorithm is to achieve an approximate performance without using the hyperbolic tangents to get down the computational complexity and make it hardware implementable.

Without loss of precision, it can be derived from equation 33 [11]:

$$\tanh \left(\frac{1}{2} LLR(r_{ij}^{(k)}) \right) = \prod_{j' \in X_{i \setminus j}} \alpha_{j'i}^{(k-1)} \prod_{j' \in X_{i \setminus j}} \tanh \left(\frac{1}{2} \beta_{j'i}^{(k-1)} \right) \quad [37]$$

Where:

$$\alpha_{ji} = \text{sgn}(LLR(q_{ji})) \quad [38]$$

$$\beta_{ji} = |(LLR(q_{ji}))| \quad [39]$$

Equation 37 can be expressed as:

$$LLR(r_{ij}^{(k)}) = \prod_{j' \in X_{i \setminus j}} \alpha_{j'i}^{(k-1)} \phi(\sum_{j' \in X_{i \setminus j}} \phi(\beta_{j'i}^{(k-1)})) \quad [40]$$

Where $\phi^{-1}(x) = \phi(x)$ for $x > 0$.

The function $\phi(x)$ represents [11]:

$$\phi(x) = -\log \left[\tanh \left(\frac{x}{2} \right) \right] = \log \left(\frac{e^x + 1}{e^x - 1} \right) \quad [41]$$

A realization of the $\phi(x)$ function is presented below:

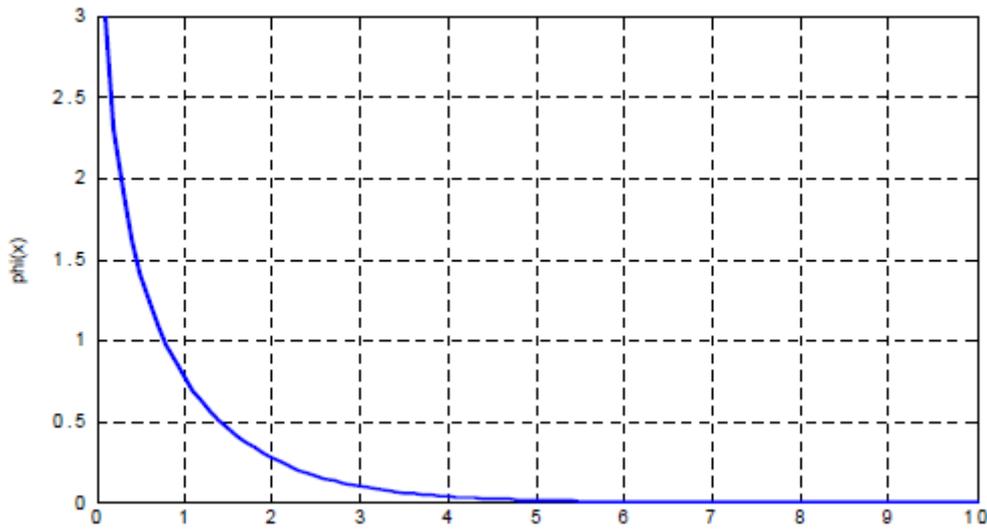


Figure 8. Function $\phi(x)$ [12].

It can be seen that the maximum of $\phi(x)$ correspond to small x values and also in equation 40 that the term prevailing in the sum is the largest one [11]:

$$\sum_{j' \in X_{i \setminus j}} \phi(\beta_{j'i}) \approx \phi(\min(\beta_{j'i})) \quad [42]$$

At the end:

$$LLR(r_{ij}^{(k)}) = \prod_{j' \in X_{i \setminus j}} \alpha_{j'i}^{(k-1)} \min_{j' \in X_{i \setminus j}} (\beta_{j'i}^{(k-1)}) \quad [43]$$

With expression 43, finally it has been removed the hyperbolic tangent and therefore the computational complexity of the decoder.

2.8.3 Offset min-sum algorithm

The idea behind this algorithm is to recover the loss of the Min-sum algorithm with respect to the BPA. In order to reduce the complexity at the check node processing, it is possible to subtract an offset factor to obtain a performance similar to the BPA (Ideal case).

$$|LLR_{offset}(r_{ij})| = \max(|LLR(r_{ij})| - offset, 0) \quad [44]$$

In this study, was used a proposed algorithm taken from [16]. In this algorithm, a uniform approximation for the check node operation of the BPA is developed through mathematical induction on Jacobian algorithm. Let us first consider a general case, as shown below [15]:

$$2 \tanh^{-1} \left(\prod_{n' \in A(m)} \tanh \left(\frac{1}{2} \beta_{mn'} \right) \right) = \beta_{m1} \otimes \beta_{m2} \otimes \beta_{m3} \dots \otimes \beta_{mn'} \quad [45]$$

Note that here the general case is targeted by considering $n' \in A(m)$, where $|A(m)| = \beta_{m1} \otimes \beta_{m2} \otimes \beta_{m3} \dots \otimes \beta_{mn'}$, and a basic computation at the check node operation of BPA as function \otimes is [15]:

$$2 \tanh^{-1} \left(\tanh \frac{\beta_1}{2} x \tanh \frac{\beta_2}{2} \right) = \beta_1 \otimes \beta_2 \quad [46]$$

The above equation can be expanded using Jacobian logarithm ($\ln(e^a + e^b) = \max(a, b) + \ln(1 + e^{-|a-b|})$) twice as follows [15]:

$$\beta_1 \otimes \beta_2 = \text{sign}(\beta_1) \text{sign}(\beta_2) (\min(|\beta_1|, |\beta_2|)) + f(|\beta_1| + |\beta_2|) - f(|\beta_1| - |\beta_2|) \quad [47]$$

Where $f(x) = \ln(1 + e^{-|x|})$. The idea is to approximate equation 47 since $f(x)$ is complex. So, from the proposed algorithm and the initial case proposed in equation 45, we can say that since function \otimes holds commutative law, it can be fairly assumed that $|\beta_{m1}| < |\beta_{m2}| < \dots < |\beta_{mn'}|$. Under this assumption, equation 45 can be expressed as [15]:

$$2 \tanh^{-1} \left(\tanh \left(\frac{\beta_{m1}}{2} \right) \dots \tanh \left(\frac{\beta_{mn'}}{2} \right) \right) \approx \text{sign}(\beta_{m1}) \dots \text{sign}(\beta_{mn'}) (\min(|\beta_{m1}|, \dots, |\beta_{mn'}|) - f(|\beta_{m2}| - |\beta_{m1}|) - f(|\beta_{m3}| - |\beta_{m1}|) - \dots - f(|\beta_{mn'}| - |\beta_{m1}|) + f(|\beta_{m2}| + |\beta_{m1}|) + f(|\beta_{m3}| + |\beta_{m1}|) + \dots + f(|\beta_{mn'}| + |\beta_{m1}|)) \quad [46]$$

Function $f(x)$ is monotonically decreasing function with $f(x) = 0$ when $x > 2.5$. Due to the relationships among $|\beta_{m1}|, \dots, |\beta_{mn'}|$, it can be derived that $|\beta_{m2}| - |\beta_{m1}|$ is the smallest one among all the arguments of $f(x)$ in the equation, therefore $-f(|\beta_{m2}| - |\beta_{m1}|)$ becomes the dominant term of all the function $f(x)$ terms. The offset term is mainly dependent on the two most unreliable inputs from the variable nodes which are denoted as β_{min1} and β_{min2} from now on. However, simply keeping the dominant term will degrade the precision computation. Therefore, approximation of all other ones is made by multiplying a normalization factor or adding an offset factor to the dominant term $-f(|\beta_{m2}| - |\beta_{m1}|)$ [16]. So, with this normalization factor γ' , equation 46 can be written as [15]:

$$2 \tanh^{-1} \left(\tanh \left(\frac{\beta_{m1}}{2} \right) \dots \tanh \left(\frac{\beta_{mn'}}{2} \right) \right) \approx \text{sign}(\beta_{m1}) \dots \text{sign}(\beta_{mn'}) (\min(|\beta_{m1}|, \dots, |\beta_{mn'}|) - \gamma' f(|\beta_{m2}| - |\beta_{m1}|)) \quad [47]$$

Also, in this project it was taken and approximation of the Jacobian logarithm ($\ln(e^a + e^b) = \max(a, b) + \ln(1 + e^{-|a-b|})$) used in equation 47, where the term Jacobian logarithm $(1 + e^{-|a-b|})$ can be approximated as [16]:

$$\delta(x) = \max \left(\frac{5}{8} - \frac{|x|}{4}, 0 \right) \quad [47]$$

Where the factors can be easily implemented in hardware.

The offset min sum algorithm was implemented for decoding in this analysis in Matlab. An optimization of the parameter gamma has been done in the reference sensitivity (REFSENS) test scenario that is based on a SISO configuration, AWGN channel and HARQ off.

The reference sensitivity is the minimum receiver input power measured at the antenna connector at which the bit error rate (BER) does not exceed a specific value.

The optimization of the parameter γ' has been done under the following parameters:

- System Bandwidth BW: 100 MHz
- Subcarrier Spacing: 60 kHz
- Slot length: 0.25 ms
- NFFT: 2048
- SISO configuration (1x1)
- AWGN channel
- Number of RB: 132
- MCS: 4
- TBS: 11528
- DMRS_CONFIG: 'TYPE_1'
- HARQ: OFF

Many values of γ' has been tested and the results obtained are shown below:

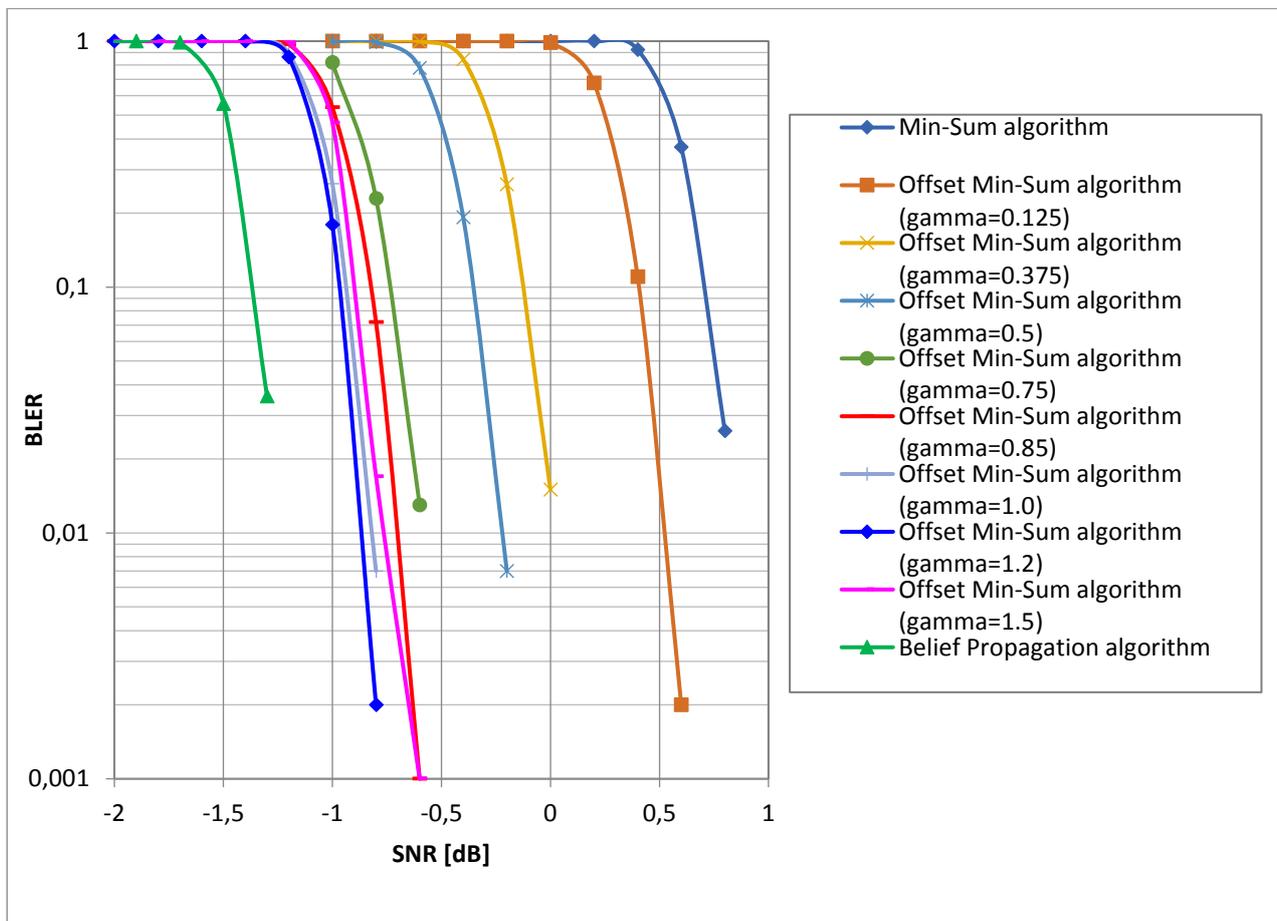


Figure 9. Optimization of parameter γ' for the offset min sum algorithm.

As is explained in this study the ideal algorithm is the Belief Propagation Algorithm (BPA) and it will take as a reference for the optimization. You can also notice that Min-sum algorithm is the one with the worst performance as it was explained in this chapter. On the other hand, it can be said that best performances in terms of BLER are the offset min-sum algorithm with gamma equal to 1 and gamma equal to 1.2 and in the following in this study we will use gamma equal 1.

2.9 Reference Signal Structure

NR has ultra-lean design that minimizes always on transmissions to enhance network energy efficiency and ensure forward compatibility. Reference signals in NR are transmitted only when necessary. The four main reference signals are: the demodulation reference signal (DMRS), phase tracking reference signal (PTRS), sounding reference signal (SRS) and channel-state information reference signal (CSI-RS).

2.9.1 Demodulation Reference Signal (DMRS)

DMRS is used to estimate the radio channel for demodulation. DMRS is UE-specific, can be beamformed, confined in a scheduled resource, and transmitted only when necessary in DL and UL.

2.9.1.1 DMRS Configurations

Variable/configurable DM-RS patterns for data demodulation are supported. At least, one configuration supports front-loaded DM-RS pattern. Front-loaded DM-RS is mapped over 1 or 2 adjacent OFDM symbols. Additional DM-RS can be configured for the later part of the slot. NR aims for performance at least comparable to DM-RS of LTE in scenarios where applicable for both LTE and NR [17].

DM-RS configuration can be up to the maximum number of DM-RS ports. At least, the 8 orthogonal DL DM-RS ports are supported for SU-MIMO (configuration type 1) and the maximal 12 orthogonal DL DM-RS ports are supported for MU-MIMO (configuration type 2). At least for CP-OFDM, NR supports a common DM-RS structure for DL and UL where the exact DM-RS location, DM-RS pattern, and scrambling sequence can be the same or different. DM-RS for same or different links can be configured to be orthogonal to each other [17].

In RAN1 NR Ad_hoc #2 was agreed that for CP-OFDM, a UE is configured by higher layers with DMRS pattern either from the front-loaded DMRS configuration type 1 or from the front-loaded DMRS configuration type 2 for DL/UL, so:

For configuration type 1:

- One symbol: Up to 4 ports through Comb-2 pattern + 2 Cyclic Shifts (CS).
- Two symbols: Up to 8 ports through Comb2 + 2CS + TD-OCC¹ ($\{1\ 1\}$ and $\{1\ -1\}$). It should be possible to schedule up to 4 ports without using both ($\{1\ 1\}$ and $\{1\ -1\}$).

For configuration type 2:

- One symbol: Up to 6 ports through 2-FD-OCC across adjacent REs in the frequency domain
- Two symbols: Up to 12 ports through 2-FD-OCC across adjacent REs in the frequency domain + TD-OCC both ($\{1\ 1\}$ and $\{1\ -1\}$).

¹ TD-OCC: Time Domain Orthogonal Cover Code

DMRS configuration type 1 is comb based pattern with cyclic shifts of length 2 and supports up to 8 orthogonal ports which consist of 2 CDM groups as shown in the following table:

2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
2/3	CDM group 1	2/3/6/7	CDM group 1
0/1	CDM group 0	0/1/4/5	CDM group 0
One Symbol		Two Symbols	

Table 6. DMRS configuration type 1.

On the other hand, DMRS configuration type 2 is FD-OCC pattern of adjacent two REs in frequency domain and supports up to 12 orthogonal DMRS ports where DMRS ports are included in three CDM groups as shown in the following table:

4/5	CDM group 2	4/5/10/11	CDM group 2
4/5		4/5/10/11	
2/3	CDM group 1	2/3/8/9	CDM group 1
2/3		2/3/8/9	
0/1	CDM group 0	0/1/6/7	CDM group 0
0/1		0/1/6/7	
4/5	CDM group 2	4/5/10/11	CDM group 2
4/5		4/5/10/11	
2/3	CDM group 1	2/3/8/9	CDM group 1
2/3		2/3/8/9	
0/1	CDM group 0	0/1/6/7	CDM group 0
0/1		0/1/6/7	
One Symbol		Two Symbols	

Table 7. DMRS configuration type 2.

Each DMRS configuration can be spanned in one or two adjacent OFDM symbols according to the gNB configuration.

In RAN1#90, the DMRS port indexing for one OFDM symbol pattern of type 1 and type 2 was decided. In this case, the DMRS port indexing goes first over the CDM/CS ports then over comb/FDM. Applying the port indexing principle used in one symbol case and applying TD-OCC together, we can extend this port indexing method for two symbol case. Therefore, resulting DMRS port indexing for two symbol case becomes CDM group 0 = {0,1,4,5} and CDM group 1 = {2,3,6,7} for DMRS configuration type1 and CDM group 0 = {0,1,6,7}, CDM group 1 = {2,3,8,9}, and CDM group 2 = {4,5,10,11} for DMRS configuration type 2.

Port	Frequency offset	FD-OCC	TD-OCC	
			One symbol	Two Symbol
0	0	[+1 +1]	[+1]	[+1 +1]
1	0	[+1 -1]	[+1]	[+1 +1]
2	1	[+1 +1]	[+1]	[+1 +1]
3	1	[+1 -1]	[+1]	[+1 +1]
4	0	[+1 +1]	--	[+1 -1]
5	0	[+1 -1]	--	[+1 -1]
6	1	[+1 +1]	--	[+1 -1]
7	1	[+1 -1]	--	[+1 -1]

Table 8. DMRS configuration type 1

Port	Frequency offset	FD-OCC	TD-OCC	
			One symbol	Two Symbol
0	0	[+1 +1]	[+1]	[+1 +1]
1	0	[+1 -1]	[+1]	[+1 +1]
2	2	[+1 +1]	[+1]	[+1 +1]
3	2	[+1 -1]	[+1]	[+1 +1]
4	4	[+1 +1]	[+1]	[+1 +1]
5	4	[+1 -1]	[+1]	[+1 +1]
6	0	[+1 +1]	--	[+1 -1]
7	0	[+1 -1]	--	[+1 -1]
8	2	[+1 +1]	--	[+1 -1]
9	2	[+1 -1]	--	[+1 -1]
10	4	[+1 +1]	--	[+1 -1]
11	4	[+1 -1]	--	[+1 -1]

Table 9. DMRS configuration type 2.

In 5G NR it is possible to have different configurations for DMRS. It can be used one front load symbol and add up to three additional DMRS symbols, or two front symbols and additional symbols can be added if needed (up to two more symbols). In RAN 1 #90 was agreed that for slot-based scheduling, for PDSCH, when three additional DMRS symbols are configured for the 1-symbol front load DMRS with front load DMRS on the 3rd symbol, the three additional DMRS symbol can be at least configured in the 6th, 9th and 12th symbols [18]

On the other hand, for slot based scheduling, for PDSCH, when one 2-symbol additional DMRS symbol is configured for the 2-symbol front load DMRS with the first symbol of front load DMRS on the 3rd or 4th

symbol, the one additional 2-symbol DMRS can be at least configured in the 9th-10th symbol for PDSCH spanning to the 10th or 11th or 12th symbol of the slot. The one additional 2-symbol DMRS can be also configured in the 11th -12th symbol for PDSCH spanning to the 13th or 14th symbol of the slot [19].

It must be noticed that for the 2-symbol and 4-symbol non-slot-based scheduling, no additional DMRS can be configured but instead for the 7-symbol non-slot-based scheduling, one additional DMRS can be configured [19].

2.9.1.2 DMRS mapping to physical resources

The UE shall assume the PDSCH DM-RS being mapped to physical resources according to type 1 (comb 2+2CS) or type 2 (2-FDD-OCC across adjacent REs) as given by the higher-layer parameter *DL-DMRS-config-type* [20].

The UE shall assume the sequence $r(m)$ is scaled by a factor β_{DMRS} to conform with the transmission power specified in [TS38.213] and mapped to physical resource elements according to [20]:

$$\begin{aligned}
 a_{k,l}^{(p,\mu)} &= \beta_{\text{DMRS}} w_f(k') \cdot w_t(l') \cdot r(2n+k') \\
 k &= \begin{cases} 4n+2k'+\Delta & \text{Configuration type 1} \\ 6n+k'+\Delta & \text{Configuration type 2} \end{cases} \\
 k' &= 0,1 \\
 l &= \bar{l} + l' \\
 n &= 0,1,\dots
 \end{aligned} \tag{48}$$

where $w_f(k')$ (FD-OCC), $w_t(l')$ (TD-OCC), and Δ (Frequency offset) are given by Table A and Table B (see on appendix) and the following conditions are fulfilled [20]:

- The resource elements are within the common resource blocks allocated for PDSCH transmission.

The reference point for k is:

- For PDSCH transmission carrying RMSI (Remaining minimum system information), subcarrier 0 of the lowest-numbered common resource block in the CORESET configured by the PBCH.
- otherwise, subcarrier 0 in common resource block 0.

The reference point for l and the position l_0 of the first DM-RS symbol depends on the mapping type:

For PDSCH mapping type A (Slot based mapping):

- l is defined relative to the start of the slot.
- $l_0 = 3$ if the higher-layer parameter *DL-DMRS-typeA-pos* equals 3 and $l_0 = 2$.

For PDSCH mapping type B (Non-slot-based mapping):

- l is defined relative to the start of the scheduled PDSCH resources
- $l_0 = 0$

The position of the DMRS symbols is given by \bar{l} and the last OFDM symbol used for PDSCH in the slot according to Table C and Table D (See appendix). The case DL-DMRS-add-pos equal to 3 is only supported when DL-DMRS-typeA-pos is equal to 2 [20].

If the PDSCH durations is 2, 4 or 7 OFDM symbols, and the PDSCH allocation collides with resources reserved for a CORESET, \bar{l} shall be incremented such that the first DMRS symbol occurs immediately after the CORESET [20].

If the PDSCH duration is 2 or 4 OFDM symbols, only single-symbol DMRS is supported [20].

The time-domain index l' and the supported antenna ports p are given by Table E (see appendix) where [20]:

- single-symbol DM-RS is used if the higher-layer parameter *DL-DMRS-max-len* is equal to 1.
- single-symbol or double-symbol DM-RS is given determined by the associated DCI if the higher-layer parameter *DL-DMRS-max-len* is equal to 2.

In absence of CSI-RS configuration, and unless otherwise configured, the UE may assume PDSCH DM-RS and SS/PBCH block to be quasi co-located with respect to Doppler shift, Doppler spread, average delay, delay spread, and spatial Rx. The UE may assume that the PDSCH DM-RS within the same CDM group are quasi co-located with respect to Doppler shift, Doppler spread, average delay, delay spread, and spatial Rx. The UE may assume that no DM-RS collides with the SS/PBCH block [20].

Scheduled Antenna port(s) are indicated in DCI messages.

Antenna port(s) – 4, 5, or 6 bits as defined by Tables F/G/H/I (see appendix), where the number of CDM groups without data of values 1, 2, and 3 refers to CDM groups {0}, {0,1}, and {0, 1,2} respectively.

3. Simulink simulation platform

One goal of this study was to implement a complete physical layer chain of a 4G LTE system and then extend it to a 5G NR system when the 3GPP standard will be finalized. The physical layer chain is used to evaluate the link level performance of the radio interface in terms e.g. of throughput or other physical layer performance metrics. The language in which it was developed the physical layer chain was Simulink.

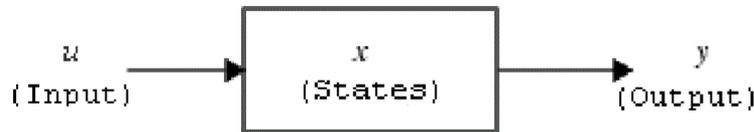
For this development it was used System Functions that are going to be briefly described.

3.1 System Functions in Simulink

S-functions (System Functions) provide a powerful mechanism for extending the capabilities of the Simulink environment. An S-function is a computer language description of a Simulink block written in Matlab, C, C++ or Fortran. S-functions are dynamically linked subroutines that the Matlab interpreter can automatically load and execute. These functions use a special calling syntax called the S-function API that enables to interact with the Simulink engine [13].

To understand how an s-function works let's first to introduce how a block in Simulink is composed and how the Simulink engine performs a simulation as well.

A Simulink block consists of a set of inputs, a set of states, and a set of outputs, where the outputs are a function of the simulation time, the inputs, and the states.



The following equations express the mathematical relationships between the inputs, outputs, states, and simulation time:

$$\begin{aligned}
 y &= f_0(t, x, u) && \text{Outputs} \\
 \dot{x} &= f_d(t, x, u) && \text{Derivatives} \\
 x_{d_{k+1}} &= f_u(t, x_c, x_{d_k}, u) && \text{Update}
 \end{aligned}$$

Where $x = [x_c, x_d]$.

On the other hand, the execution of a Simulink model proceeds in stages. First stage is the initialization phase, in which the Simulink engine incorporates library blocks into the model, propagates signal widths, data types, and sample times, evaluates block parameters, determines block execution order, and allocates memory. The engine then enters a simulation loop, where each pass through the loop is referred to as a simulation step. During each simulation step, the engine executes each block in the model in the order determined during initialization. For each block, the engine invokes functions that compute the block states, derivatives, and outputs for the current sample time. The inner integration loop takes place only if the model contains continuous states. The engine executes this loop until the solver reaches the desired accuracy for the state computations. The entire simulation loop then continues until the simulation is complete [13].

3.1.2 Level 2 Matlab S-functions

These kinds of functions provide a Matlab interface to interact with an extensive set of the S-function API and supports code generation.

Level 2 Matlab S-functions, allow to create blocks with many of the features and capabilities of Simulink built-in blocks, including [13]:

- Multiple input and output ports
- The ability to accept vector or matrix signals
- Support for various signal attributes including data type, complexity, and signal frames
- Ability to operate at multiple sample rates

A Level-2 MATLAB S-function consists of a setup routine to configure the basic properties of the S-function, and call-backs methods that the Simulink engine invokes at appropriate times during the simulation. The advantage of Level-2 MATLAB S-functions is speed of development. Developing Level-2 MATLAB S-functions avoids the time-consuming compile-link-execute cycle required when developing in a compiled language. Level-2 MATLAB S-functions also have easier access to MATLAB toolbox functions and can utilize the MATLAB Editor/Debugger [13].

3.1.2.1 Call-back methods

Every S-function must implement a set of methods, called call-back methods, that the Simulink engine invokes when simulating a model that contains the S-function. Tasks performed by S-function call-back methods include [13]:

- Initialization (e.g. initializing the SimStruct, a simulation structure that contains information about the S-function, setting the number and dimensions of input and output ports, setting the block sample times, allocating storage areas).
- Calculation of next sample hit (If you created a variable sample time block, this stage calculates the time of the next sample hit; that is, it calculates the next step size).
- Calculation of outputs in the major time step (after this call is complete, all the block output ports are valid for the current time step).
- Update of discrete states in the major time step (in this call, the block performs once-per-time-step activities such as updating discrete states).
- Integration (this applies to models with continuous states and/or non-sampled zero crossings. If your S-function has continuous states, the engine calls the output and derivative portions of your S-function at minor time steps).

Level-2 MATLAB S-functions must implement the following (mandatory) call-back methods [13]:

- Setup: specifies the sizes of various parameters in the SimStruct, such as the number of output ports for the block.
- Outputs: calculates the output of the block.
- Terminate: performs any actions required at the termination of the simulation. If no actions are required, this function can be implemented as a stub.

There are also many optional methods that can be defined to configure your block behavior.

3.1.2.2 D-Work Vectors

These kinds of vectors are useful when a block require to store data that must be persistent. D-Work vectors are blocks of memory that an S-function asks the Simulink engine to allocate to each instance of the S-function in a model. If multiple instances of an S-function can occur in a model, your S-function must use D-Work vectors instead of global or static memory to store instance-specific values of S-function variables. Otherwise, your S-function runs the risk of one instance overwriting data needed by another instance, causing a simulation to fail or produce incorrect results. The ability to keep track of multiple instances of an S-function is called re-entrancy. You can create an S-function that is re-entrant by using D-Work vectors that the engine manages for each instance of the S-function [13].

D-Work vectors have several advantages [13]:

- Provide instance-specific storage for block variables.
- Support floating-point, integer, pointer, and general data types.
- Eliminate static and global variables.
- Interact directly with the Simulink engine to perform memory allocation, initialization, and deallocation.
- Facilitate in-lining the S-function during code generation.
- Provide more control over how data appears in the generated code.

A key advantage of D-Work vectors is their connection to the Simulink engine. Over the course of the simulation, the engine relieves the S-function of all memory management tasks related to D-Work vectors. In an S-function that uses D-Work vectors, the engine, not the S-function, manages the memory for the D-Work vector. At the start of a simulation, the engine allocates the memory required for each instance of the S-function based on the size and the data type of the D-Work vector contents. At the end of the simulation, the engine automatically deallocates the memory [13].

3.2 Development of an LTE simulation chain using Simulink

To have a reference system to compare its performance with the NR radio chain, an LTE radio chain has been developed in Simulink through the Level 2 Matlab S-function (explained in the section before). Cases as MIMO 2x2, MIMO 4X4 for both cases in which one or two codewords must be transmitted have been developed.

The chain developed is composed by an Outer Modem (Turbo coding and rate matching), Inner Modem (MIMO processing and OFDM modulation), Channel, Inner Modem Receiver (OFDM demodulation, channel estimation and MIMO processing), Outer Modem Receiver (rate de-matching and Turbo decoding).

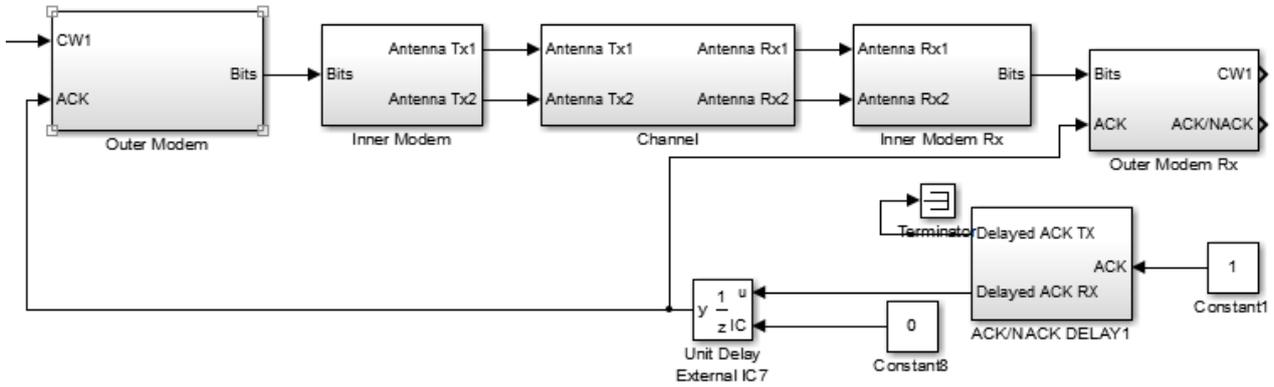


Figure 10. Simulink LTE chain for one codeword and MIMO 2x2.

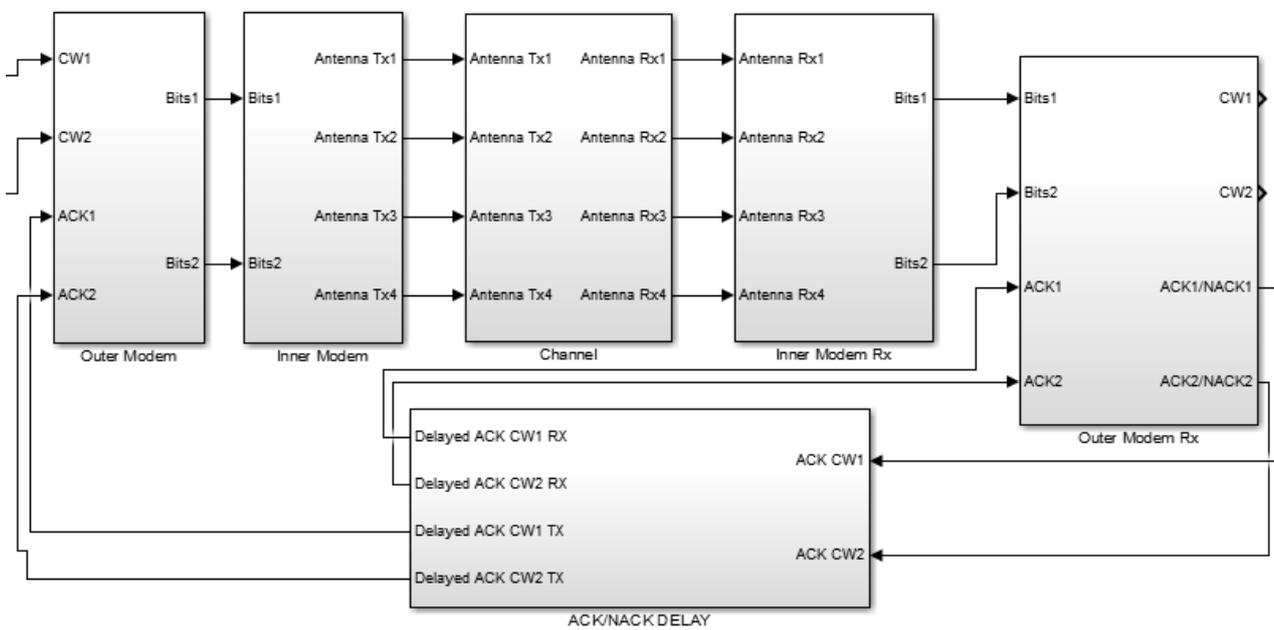


Figure 11. Simulink LTE radio chain for two codewords and MIMO 4x4.

The Outer Modem is composed by blocks that perform the following operations:

- Segmentation
- Channel Coding, that in this case is the Turbo Coder used in LTE.
- Rate Matching

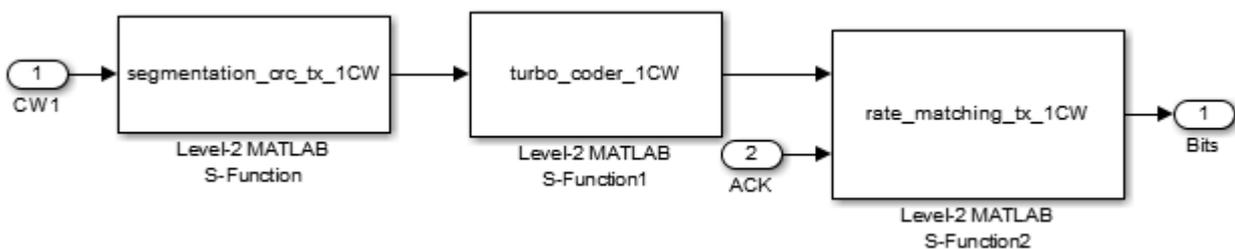


Figure 12. Outer Modem LTE for one codeword.

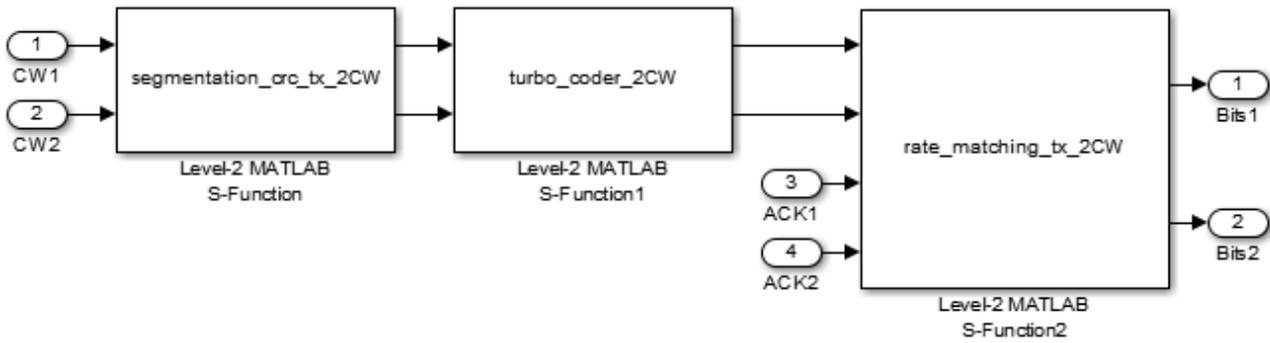


Figure 13. LTE Outer Modem for two codewords.

The Inner Modem is composed by blocks that perform the following operations:

- Bit scrambling
- Modulation
- Layer mapping
- MIMO processing
- Subcarrier mapping
- OFDM modulation
- Oversample filtering

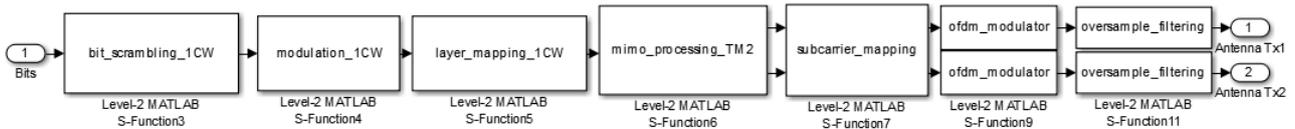


Figure 14. LTE Inner modem for one codeword MIMO 2x2.

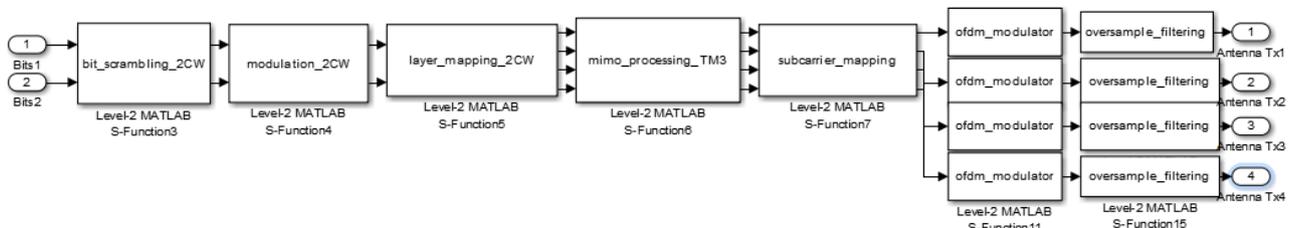


Figure 15. LTE Inner modem for two codewords MIMO 4x4.

The channel block can be one of the following options:

- Fading Channel
- Extended Pedestrian A (EPA)
- Extended Vehicular A (EVA)
- Extended Typical Urban (ETU)

The Inner Modem Receiver is composed by the following blocks:

- Downsample and filtering
- OFDM demodulator
- Pilot compensation
- Channel estimation
- Subcarrier demapping
- MIMO processing

- Layer demapping
- Symbol to bit demapping
- Bit descrambling

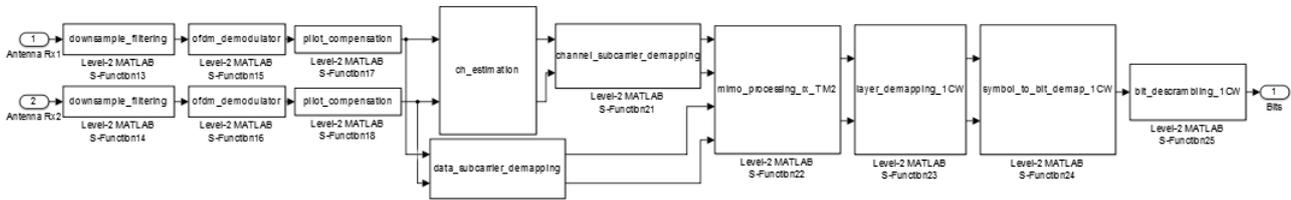


Figure 16. LTE Inner Modem receiver for one codeword MIMO 2x2.

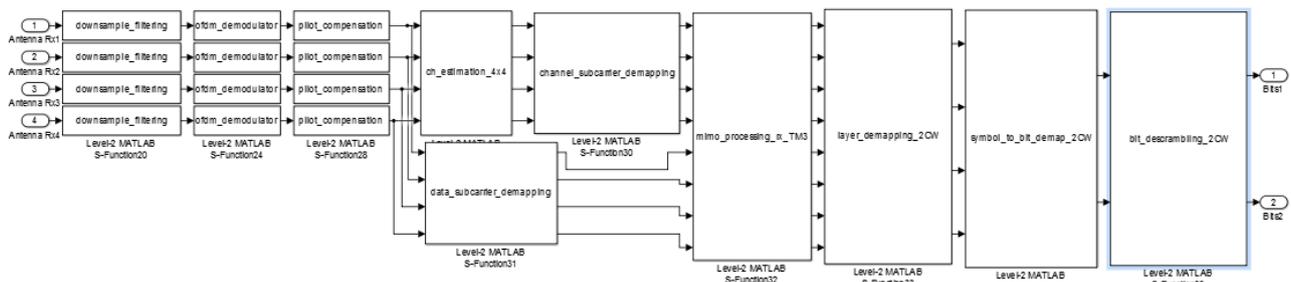


Figure 17. LTE Inner modem receiver for two codewords MIMO 4x4.

Finally, the Outer Modem Receiver is composed by the following blocks:

- Rate matching
- Channel decoding

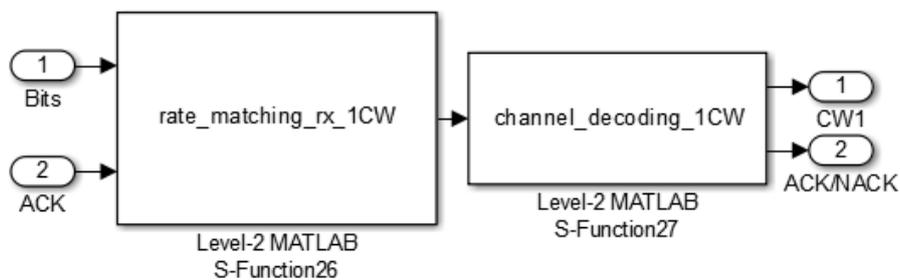


Figure 18. LTE Outer modem receiver for one codeword MIMO 2x2.

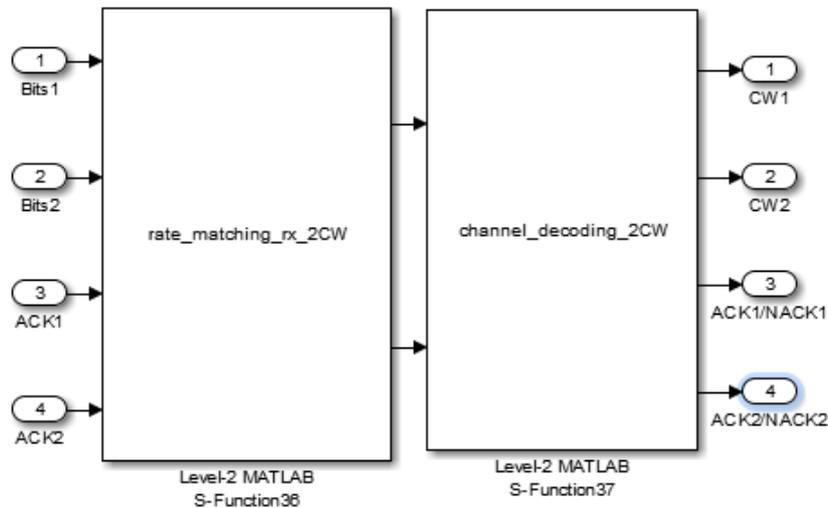


Figure 19. LTE Outer modem receiver for two codewords MIMO 4x4.

The Ack/Nack delay is a block that helps to initialize correctly the acknowledgement operation and permit to have a properly Hybrid Automatic Repeat Request (H-ARQ) procedure with feedback generated by the outer modem receiver and sent back to the outer modem at the transmitter.

Two scenarios have been tested, in order to conclude if the performances given by the Simulink developed simulator are congruent. The first scenario has been simulated using the following parameters:

- Channel: Extended Pedestrian ($v = 3$ km/h)
- MIMO 2X2 Transmit Diversity TM2
- MCS 28 (64-QAM , coding rate = 0.9309)
- PRB allocated = 50x2 per antenna
- TBS = 36696 bits
- Alamouti decoder - RV optimal = {0 2 3 1}

The results are shown in the following figure and are compared with a reference curve obtained with a link level simulator based on MATLAB:

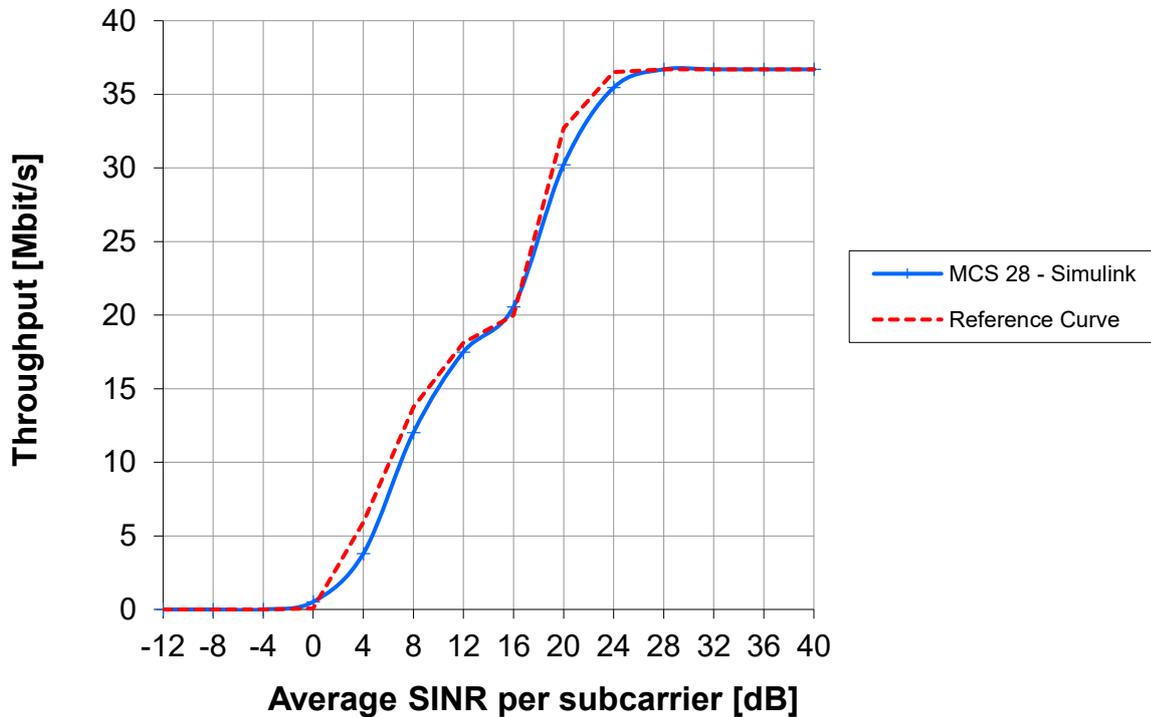


Figure 20. Comparison between Simulink and reference performance curve.

According to the figure above, the Simulink simulator developed has a similar behavior with respect to the Matlab one. For low values of SINR the Simulink simulator presents a better performance. On the other hand, for large SINR values the peak throughput is reached in this scenario. Therefore, it could be said that it has been obtained a reliable Simulink simulator.

Subsequently, both simulators have been tested in another scenario with the following parameters:

- Channel: Extended Pedestrian A ($v = 3$ km/h)
- MIMO 4X4 Open Loop Spatial Multiplexing with 4 layers (TM3)
- MCS 27 (64-QAM, coding rate = 0.8307)
- PRB allocated = 50x2 per antenna
- TBS = 63776 bits

The results obtained are presented below:

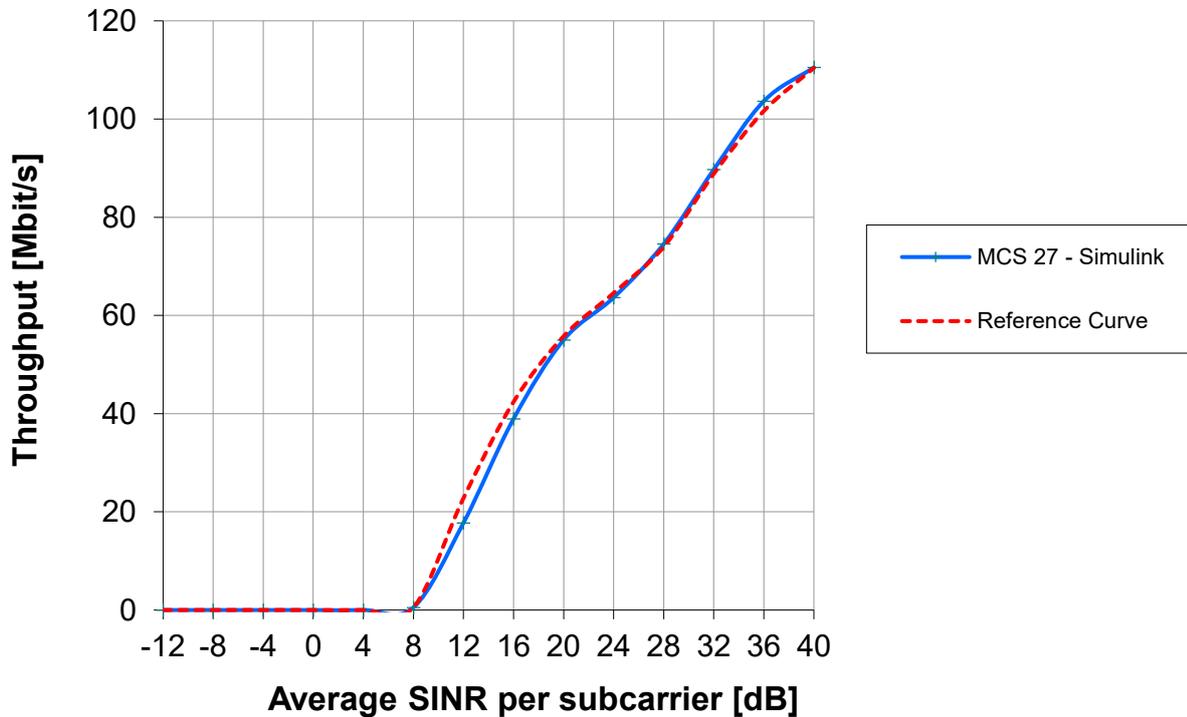


Figure 21. Comparison between Simulink and reference performance curve.

Also, for a MCS 27 TM3 the developed simulator behaves as expected with respect to a reference LTE simulator developed in MATLAB. The first step of this study has then been completed with the development of LTE simulation chain in Simulink. In the following it is described the prosecution of the study that consisted in the update of some simulation blocks according to the 5G NR standard defined by 3GPP. This work represents the basis for the development of a NR simulation chain in Simulink.

3.3 Development of a New Radio simulation chain

As it was said before, one goal of this study is to start the development of a link level simulation chain for the 5G NR system. Having into account the decisions taken by the 3GPP about the NR radio interface, some blocks of the 4G LTE chain presented before have been modified to make them compliant with the NR system. Considering that the 3GPP standard at the time of writing was still under development it is possible that further refinements of the developed blocks will be necessary.

3.3.1 Outer Modem Comparison (NR vs LTE)

As first step to achieve the goal, it was changed the Outer Modem of the LTE chain, so instead using a Turbo Coder it was used the LDPC Coder (see chapter 2.7) and for decoding the data, it was used the min-sum algorithm, the offset min-sum algorithm, and the Belief Propagation Algorithm (BPA) (Section 2.7 explains these algorithms).

Three test scenarios have been simulated changing the modulation type (QPSK, 16-QAM, 64-QAM), the coding rate and therefore the transport block size (TBS). It has to be noticed that in these three simulations, either the numerology and the transmitted cell specific reference signals CRS and consequently the channel estimation for coherent demodulation at the receiver, were the same as in LTE, in order to only compare the performance of the new NR outer modem based on LDPC coding with the LTE outer modem based on the Turbo coder. The numerology used was:

- System Bandwidth $B = 10\text{MHz}$
- Subcarrier spacing $\Delta f = 15\text{kHz}$
- FFT size $N_{\text{FFT}} = 1024$
- Number of allocated Physical Resource Block $N_{\text{PRB}} = 50$
- Pilot pattern: CRS pattern of LTE

The first simulation is a scenario with Modulation and Coding Scheme (MCS) 5, which means that the modulation scheme is a QPSK, with a coding rate equal to 0.33, and Transport Block Size (TBS) equal to 4392 bits. Moreover, it has been modeled a MIMO 2x2 scheme that uses Transmit Diversity based on Alamouti coding (TM2). An EPA (Extended Pedestrian A model) channel with low correlation is used.

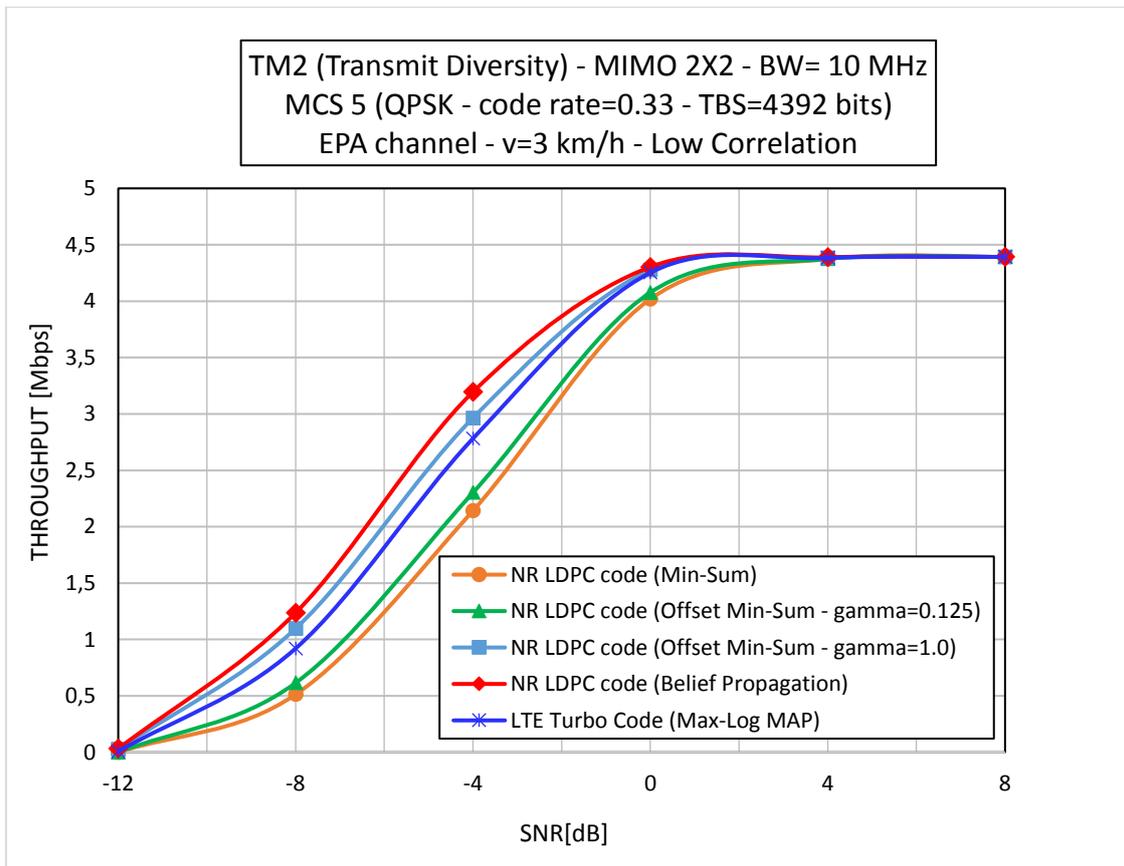


Figure 22. LDPC vs Turbo coder MCS 5.

The second simulation has the same parameters as before but it was used a MCS 14, which means that a 16-QAM modulation and a code rate of 0.5 was used. The results are shown below:

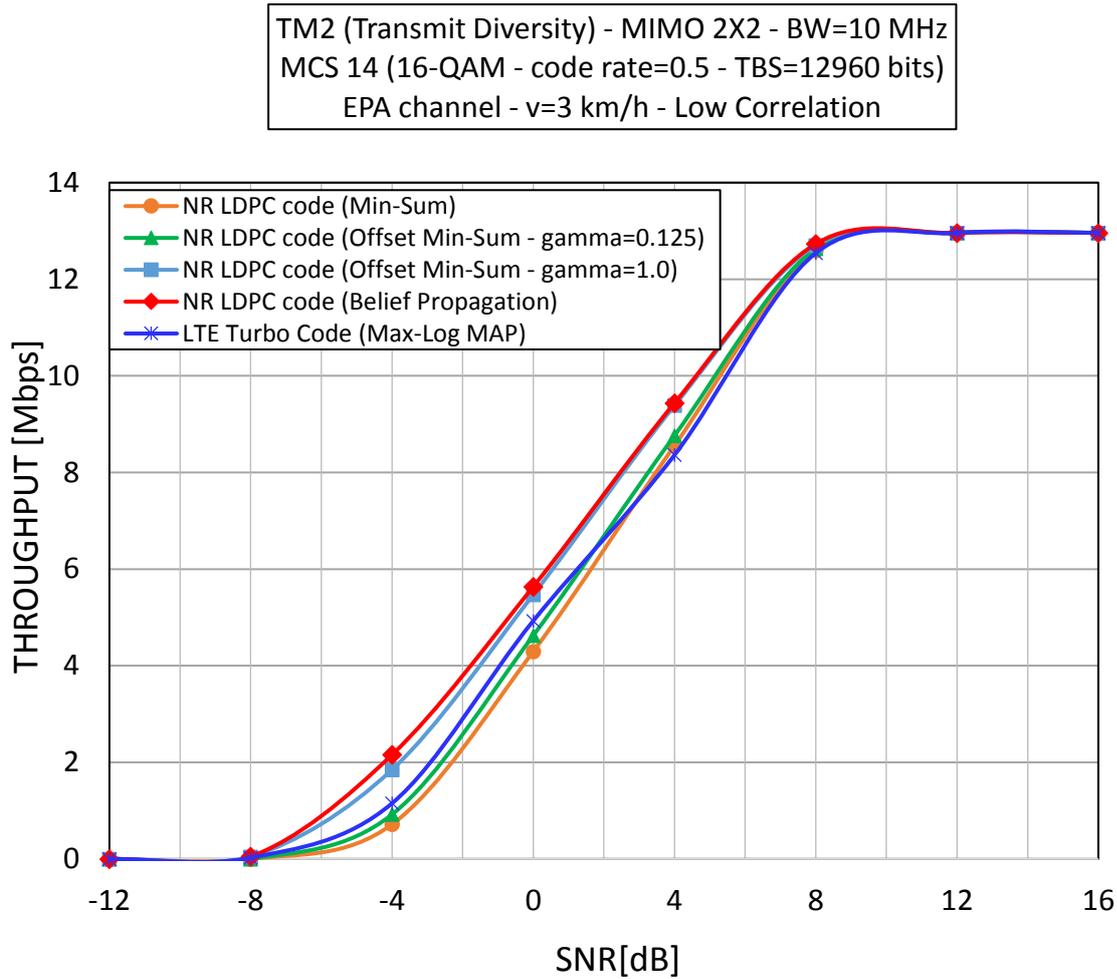


Figure 22. LDPC vs Turbo coder MCS 14.

The third simulation, also use the same parameters but only change the MCS to 27, which means that a 64-QAM and code rate of 0.8 was used as a modulation and coding scheme. The results are presented below:

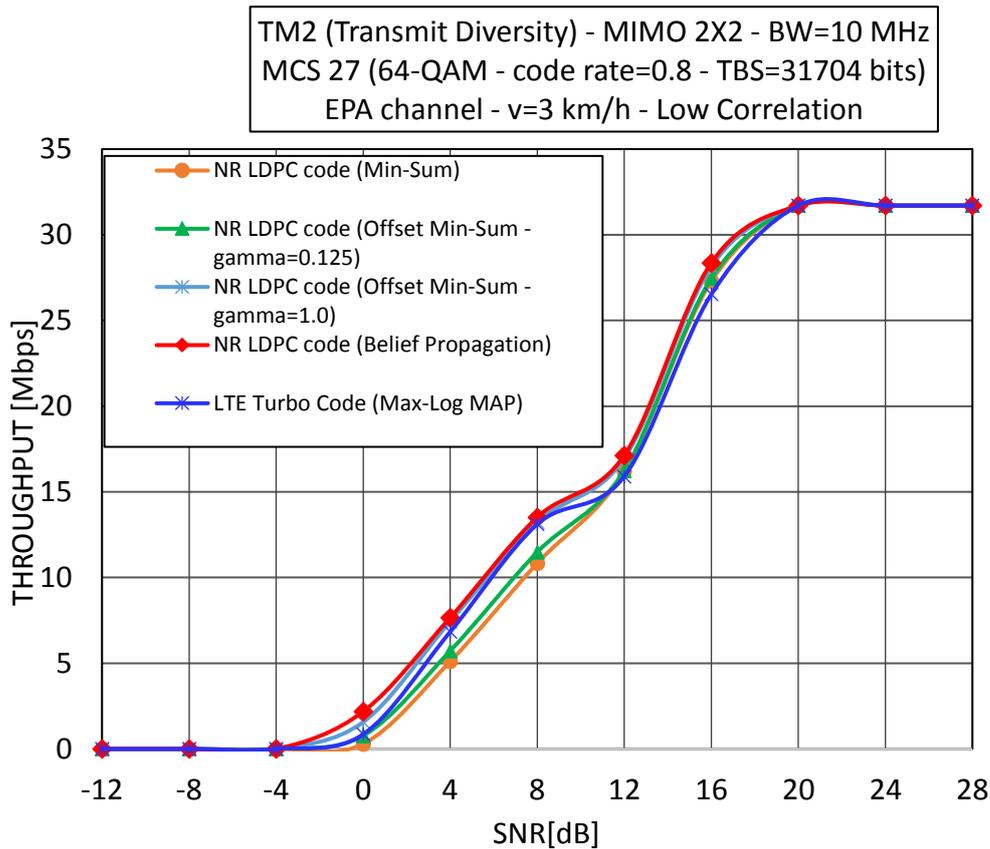


Figure 23. LDPC vs Turbo coder MCS 27.

From the 3 results before, it can be observed that using a LDPC coder and using as decoder the Belief Propagation Algorithm, the LDPC always presents slightly better performance than the LTE turbo coder as was expected. The only constraint is that the BPA is not suitable for hardware implementation due to the complexity that present for decoding, since uses hyperbolic tangents (see chapter 2.8 for explanation). For this reason, this kind of algorithm actually is not used in practice, but it can be taken as an ideal or reference system.

Analyzing performance when is used the offset min-sum; that is an approximation of the BPA, for the optimal value of the parameter $\gamma' = 1$ see chapter 2.8.3), the throughput is always slightly better than the LTE system with turbo coder independent of the modulation and coding scheme used, as is expected. On the other hand, the LDPC code with off-set min sum and $\gamma' = 1$ for the case of MCS 5 (Figure 27), approximates really good the performance of the BPA, with a difference that is less than a half decibel for cases when the SNR is between -8dB and -2 dB. Even more, for a greater MCS, for example MCS 14 and 27 where the modulation and coding rate are better, the offset min-sum with optimal gamma behaves much closer to the BPA. So, it can be achieved the optimal BPA performance through the offset min-sum with optimal gamma.

Finally, simulations made with LDPC code with min-sum algorithm show a loss compared to the LTE turbo coder performances for low SNR scenarios of about 3-4dB for MCS 5 and MCS 14 and more than 1dB for MCS 27.

3.3.2 Increased Bandwidth and DMRS pattern

As next step, to test the simulation chain in a scenario that is closer to those envisaged for the NR interface, we have considered the case of a channel bandwidth of 100 MHz. Accordingly, the OFDM parameters have been changed by using a larger subcarrier spacing of 60 kHz, compared to the 15 kHz used in the previous simulations. In this scenario the subframe duration reduces from 1 ms to 0.25 ms. In this simulation the Reference Signals of LTE have been replaced with those defined for the NR radio interface, which are denoted as DMRS (Demodulation Reference Signals) and are user specific. It must be noted that the transmission mode TM2 (Transmit Diversity) is not present in the NR standard. Therefore, the purpose of this simulation is just to verify how the physical layer throughput scales with the channel bandwidth.

The simulation parameters are:

- System Bandwidth B = 100MHz
- Subcarrier spacing $\Delta f = 60\text{KHz}$
- MCS 27 (64-QAM, code rate = 0.8)
- EPA channel (3 km/h)
- Pilot pattern: DMRS pattern of NR

The results obtained are presented in the figure below:

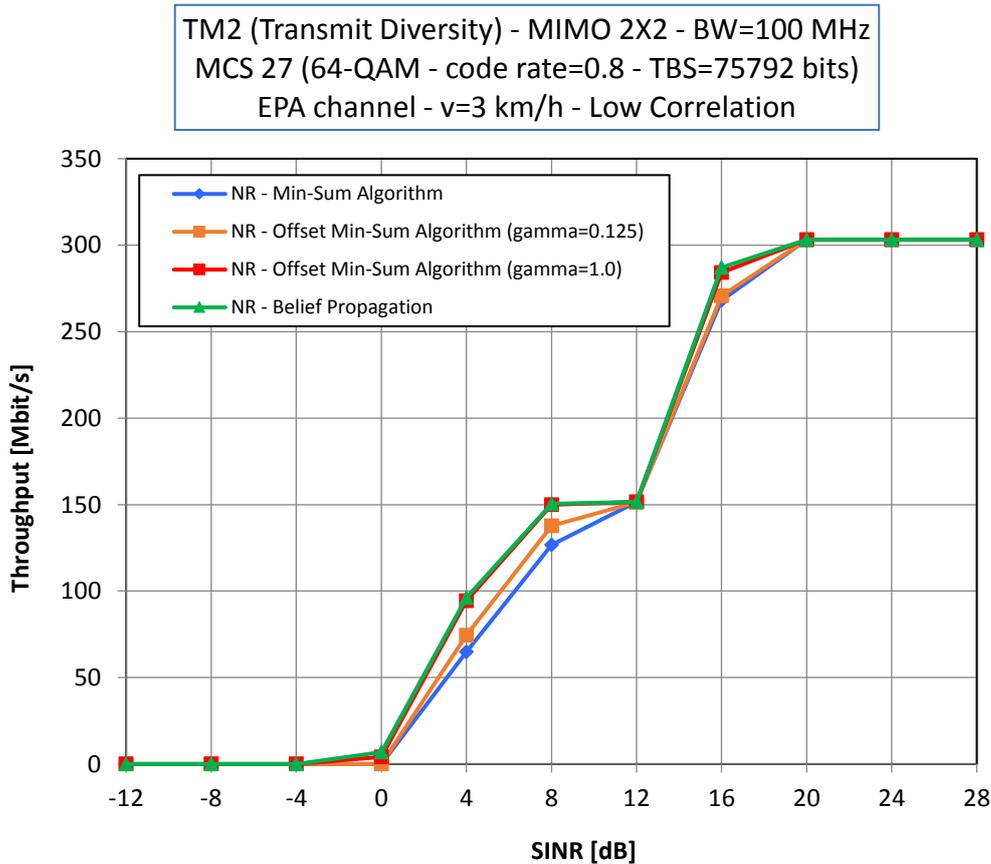


Figure 24. Test of the simulation chain in case of the channel bandwidth. BW=100MHz

As we expected the throughput has increased by a factor of about 10 in comparison with the simulation made in section 3.3.1 Figure 29. Actually the increase factor is less than the ideal 10 due to the fact that the used DMRS pilot pattern of NR introduces a slightly higher overhead compared to LTE, because 24 REs/PRB² are used for pilot compared to 12 REs/PRB of LTE, but at least 300 Mbit/s of peak throughput has been reached. It can be also said, that as in the other simulations, the offset min-sum algorithm with parameter gamma equal 1 (see section 2.8.3 for optimization) is a good approximation of the ideal BPA and the min sum algorithm provides the worst performance.

² REs/PRB: Resource Elements per Physical Resource Blocks (PRB) occupied by DMRS

4. Summary and Conclusions

A link level simulation chain for the 4G LTE system has been developed in this thesis using Simulink® as simulation platform. The main goal of this work was to implement a software model of the physical layer, useful to evaluate the performance of the 4G LTE system. The subsequent step has been the modification of some blocks, according to the 3GPP specifications for the new 5G NR system, and start the development of a simulation chain for this new advanced communication system.

The simulation chains developed in this study are based on Level 2 S-functions, which are a useful tool in order to create signal processing blocks with a desired behaviour that provide a powerful mechanism for extending the capabilities of the Simulink environment.

The channel coding schemes based either on Turbo Code (4G LTE) and LDPC codes (5G NR) have been analyzed and compared in detail. The simulations show and compare the performance of different decoding algorithms and provide in the LDPC case some result for the optimization of the decoder parameters.

Three techniques of channel decoding for LDPC have been evaluated in this study: Belief Propagation algorithm (BPA), Min-Sum algorithm and Offset Min-Sum algorithm. The BPA is the algorithm that provides the better performance at the price of a large computational complexity (i.e. computation of hyperbolic functions) in both hard and soft decoding cases. However, the hardware implementation of the BPA in practice is not feasible for complexity reasons. The Min-sum and the Offset Min-sum algorithm have been proposed in literature to approximate the BPA algorithm by reducing the computational complexity at the cost of a slightly lower performance. The Offset Min-sum algorithm in particular allows to recover a significant part of the performance gap between the BPA and the Min-sum algorithm by optimizing a specific parameter that is used in the computation of the decoding metrics.

Another aspect that has been analyzed in this study are the Reference Signal (RS) used by the 5G NR system to estimate the radio channel for demodulation. In particular a simulation block that generates the DMRS and maps them in the time-frequency grid of the OFDM modulation has been implemented and tested. These RS are denoted in the 3GPP standard as DMRS (Demodulation Reference Signals). The DMRS are user specific and transmitted only in the resources allocated to the user for data transmission. The basic DMRS pattern support low latency applications. For low speed scenarios, DMRS uses low density in the time domain. However, for high speed scenarios the time density of DMRS is increased to track fast changes in the radio channel.

A further aspect that has been analyzed is the scalability of the OFDM parameters adopted in the 5G NR standard. In particular the subcarrier spacing that is fixed in the LTE system (fixed and equal to 15 kHz) has been made scalable in the 5G NR system to support different scenarios/requirements like for example support services that require very low latency or facilitate the deployment of NR at high frequency bands in the mmWave range. A first simulation test that has been done is the usage of a large subcarrier spacing (60 kHz) that allows to extend the channel bandwidth above the 20 MHz value that represents the maximum for LTE. Using a larger subcarrier spacing it has been then tested the case of a channel bandwidth of 100 MHz for the NR system, demonstrating also that the throughput scales as expected nearly proportionally to the transmission bandwidth.

5. Appendix

In the following are listed some Tables derived from the 3GPP specification [20], related to the DMRS configuration of NR

Table A

Parameters for PDSCH DM-RS configuration type 1 [20].

p	CDM group	Δ	$w_f(k')$		$w_t(l')$	
			$k' = 0$	$k' = 1$	$l' = 0$	$l' = 1$
1000	0	0	+1	+1	+1	+1
1001	0	0	+1	-1	+1	+1
1002	1	1	+1	+1	+1	+1
1003	1	1	+1	-1	+1	+1
1004	0	0	+1	+1	+1	-1
1005	0	0	+1	-1	+1	-1
1006	1	1	+1	+1	+1	-1
1007	1	1	+1	-1	+1	-1

Table B

Parameters for PDSCH DM-RS configuration type 2 [20].

p	CDM group	Δ	$w_f(k')$		$w_t(l')$	
			$k' = 0$	$k' = 1$	$l' = 0$	$l' = 1$
1000	0	0	+1	+1	+1	+1
1001	0	0	+1	-1	+1	+1
1002	1	2	+1	+1	+1	+1
1003	1	2	+1	-1	+1	+1
1004	2	4	+1	+1	+1	+1
1005	2	4	+1	-1	+1	+1
1006	0	0	+1	+1	+1	-1
1007	0	0	+1	-1	+1	-1
1008	1	2	+1	+1	+1	-1
1009	1	2	+1	-1	+1	-1
1010	2	4	+1	+1	+1	-1
1011	2	4	+1	-1	+1	-1

Table C

PDSCH DM-RS positions \bar{l} for single-symbol DM-RS [20].

Duration of PDSCH transmission	DM-RS positions \bar{l}							
	PDSCH mapping type A				PDSCH mapping type B			
	DL-DMRS-add-pos				DL-DMRS-add-pos			
	0	1	2	3	0	1	2	3
≤ 6	l_0	-	-	-	l_0			
7	l_0	-	-	-	l_0	$l_{0,4}$		
8	l_0	-	-	-	l_0			
9	l_0	$l_{0,7}$	-	-	l_0			
10	l_0	$l_{0,9}$	$l_{0,6,9}$	-	l_0			
11	l_0	$l_{0,9}$	$l_{0,6,9}$	-	l_0			
12	l_0	$l_{0,9}$	$l_{0,6,9}$	$l_{0,5,8,11}$	l_0			
13	l_0	$l_{0,11}$	$l_{0,7,11}$	$l_{0,5,8,11}$	l_0			
14	l_0	$l_{0,11}$	$l_{0,7,11}$	$l_{0,5,8,11}$	l_0			

Table D

PDSCH DM-RS positions \bar{l} for double-symbol DM-RS [20].

Duration of PDSCH transmission	DM-RS positions \bar{l}					
	PDSCH mapping type A			PDSCH mapping type B		
	DL-DMRS-add-pos			DL-DMRS-add-pos		
	0	1	2	0	1	2
≤ 8	l_0	-		l_0		
8	l_0	-		l_0		
9	l_0	-		l_0		
10	l_0	$l_{0,8}$		l_0		
11	l_0	$l_{0,8}$		l_0		
12	l_0	$l_{0,8}$		l_0		
13	l_0	$l_{0,10}$		l_0		
14	l_0	$l_{0,10}$		l_0		

Table E

PDSCH DM-RS time index l' and antenna ports p .

Single or double symbol DM-RS	l'	Supported antenna ports p	
		Configuration type 1	Configuration type 2
single	0	1000 – 1003	1000 – 1005
double	0, 1	1000 – 1007	1000 – 1011

Table F

Antenna port(s) (1000 + DMRS port), *DL-DMRS-config-type=1*, *DL-DMRS-max-len=1* [20].

One Codeword: Codeword 0 enabled, Codeword 1 disabled		
Value	Number of DMRS CDM group(s) without data	DMRS port(s)
0	1	0
1	1	1
2	1	0,1
3	2	0
4	2	1
5	2	2
6	2	3
7	2	0,1
8	2	2,3
9	2	0-2
10	2	0-3
11	2	0,2
12-15	Reserved	Reserved

Table G

Antenna port(s) (1000 + DMRS port), *DL-DMRS-config-type=1*, *DL-DMRS-max-len=2* [20]

One Codeword: Codeword 0 enabled, Codeword 1 disabled				Two Codewords: Codeword 0 enabled, Codeword 1 enabled			
Value	Number of DMRS CDM group(s) without data	DMRS port(s)	Number of front-load symbols	Value	Number of DMRS CDM group(s) without data	DMRS port(s)	Number of front-load symbols
0	1	0	1	0	2	0-4	2
1	1	1	1	1	2	0,1,2,3,4,6	2
2	1	0,1	1	2	2	0,1,2,3,4,5,6	2
3	2	0	1	3	2	0,1,2,3,4,5,6,7	2
4	2	1	1	4-31	reserved	reserved	Reserved
5	2	2	1				
6	2	3	1				
7	2	0,1	1				
8	2	2,3	1				
9	2	0-2	1				
10	2	0-3	1				
11	2	0,2	1				
12	2	0	2				
13	2	1	2				
14	2	2	2				
15	2	3	2				
16	2	4	2				
17	2	5	2				
18	2	6	2				
19	2	7	2				
20	2	0,1	2				
21	2	2,3	2				
22	2	4,5	2				
23	2	6,7	2				
24	2	0,4	2				
25	2	2,6	2				
26	2	0,1,4	2				
27	2	2,3,6	2				
28	2	0,1,4,5	2				
29	2	2,3,6,7	2				
30	2	0,2,4,6	2				
31	Reserved	Reserved	Reserved				

Table H

Antenna port(s) (1000 + DMRS port), *DL-DMRS-config-type=2*, *DL-DMRS-max-len=1[2]*

One codeword: Codeword 0 enabled, Codeword 1 disabled			Two codewords: Codeword 0 enabled, Codeword 1 enabled		
Value	Number of DMRS CDM group(s) without data	DMRS port(s)	Value	Number of DMRS CDM group(s) without data	DMRS port(s)
0	1	0	0	3	0-4
1	1	1	1	3	0-5
2	1	0,1	2-31	reserved	reserved
3	2	0			
4	2	1			
5	2	2			
6	2	3			
7	2	0,1			
8	2	2,3			
9	2	0-2			
10	2	0-3			
11	3	0			
12	3	1			
13	3	2			
14	3	3			
15	3	4			
16	3	5			
17	3	0,1			
18	3	2,3			
19	3	4,5			
20	3	0-2			
21	3	3-5			
22	3	0-3			
23	2	0,2			
24-31	Reserved	Reserved			

Table I

Antenna port(s) (1000 + DMRS port), *DL-DMRS-config-type=2*, *DL-DMRS-max-len=2*

One codeword: Codeword 0 enabled, Codeword 1 disabled				Two Codewords: Codeword 0 enabled, Codeword 1 enabled			
Value	Number of DMRS CDM group(s) without data	DMRS port(s)	Number of front-load symbols	Value	Number of DMRS CDM group(s) without data	DMRS port(s)	Number of front-load symbols
0	1	0	1	0	3	0-4	1
1	1	1	1	1	3	0-5	1
2	1	0,1	1	2	2	0,1,2,3,6	2
3	2	0	1	3	2	0,1,2,3,6,8	2
4	2	1	1	4	2	0,1,2,3,6,7,8	2
5	2	2	1	5	2	0,1,2,3,6,7,8,9	2
6	2	3	1	6-63	Reserved	Reserved	Reserved
7	2	0,1	1				
8	2	2,3	1				
9	2	0-2	1				
10	2	0-3	1				
11	3	0	1				
12	3	1	1				
13	3	2	1				
14	3	3	1				
15	3	4	1				
16	3	5	1				
17	3	0,1	1				
18	3	2,3	1				
19	3	4,5	1				
20	3	0-2	1				
21	3	3-5	1				
22	3	0-3	1				
23	2	0,2	1				
24	3	0	2				
25	3	1	2				
26	3	2	2				
27	3	3	2				
28	3	4	2				
29	3	5	2				
30	3	6	2				
31	3	7	2				
32	3	8	2				
33	3	9	2				
34	3	10	2				
35	3	11	2				
36	3	0,1	2				
37	3	2,3	2				
38	3	4,5	2				
39	3	6,7	2				
40	3	8,9	2				
41	3	10,11	2				
42	3	0,1,6	2				
43	3	2,3,8	2				
44	3	4,5,10	2				
45	3	0,1,6,7	2				
46	3	2,3,8,9	2				
47	3	4,5,10,11	2				
48	1	0	2				
49	1	1	2				
50	1	6	2				
51	1	7	2				
52	1	0,1	2				

53	1	6,7	2				
54	2	0,1	2				
55	2	2,3	2				
56	2	6,7	2				
57	2	8,9	2				
58-63	Reserved	Reserved	Reserved				

6. References

- [1] S. Y. Lien, S. L. Shieh, Y. Huang, B. Su, Y. L. Hsu and H. Y. Wei, "5G New Radio: Waveform, Frame Structure, Multiple Access, and Initial Access," in *IEEE Communications Magazine*, vol. 55, no. 6, pp. 64-71, 2017.
- [2] QUALCOMM Technologies, Inc. "Making 5G NR a reality. Leading the technology inventions for a unified, more capable 5G air interface", whitepaper, December 2016.
- [3] 3GPP TR 38.913, "Study on Scenarios and Requirements for Next Generation Access Technologies", release 14.
- [4] NOKIA, "5G a System of Systems for a programmable multi-service architecture", 2016.
- [5] ERICSSON, "5G New Radio: Designing for the future", in Ericsson technology review, 2017.
- [6] HUAWEI, "Up in the air with 5G", October 2016.
- [7] HUAWEI, "Filtered-OFDM Enabler for Flexible Waveform in the 5th Generation Cellular Networks", Globecom, 2015.
- [8] HUAWEI, "5G: New Air Interface and Radio Access Virtualization", whitepaper, April 2015.
- [9] 3GPP R1-1612904, "On mini-slots". Ericsson, 3GPP TSG RAN WG1 #87, Reno, NV, USA, November 14-18, 2016.
- [10] Y. Yuan, X. Wang, "5G New Radio: Physical Layer Overview", published online, June 2017.
- [11] M. La Paglia, "Analysis and Simulation of LDPC Codes", Master Degree Thesis 2007.
- [12] ERICSSON, "LDPC Code Design for NR", R1-1608875, October 2016.
- [13] MATHWORKS, "Developing S-functions manual" , R2013b.
- [14] SAMSUNG, "Rate matching for data channel coding", R1-1714590, August 2017.
- [15] W. Jin, M. Hamaminato, H. Nakayama and S. Goto, "A novel hardware-friendly self-adjustable offset min-sum algorithm for ISBD-S2 LDPC decoder", EUPISCO, August 2010.
- [16] M. Mansour, R. Shanbhag, "High throughput LDPC decoders", IEEE, December 2003.
- [17] 3GPP TR 38.912, "3rd generation partnership project; Technical specification group radio access network; study on New Radio access technology", release 14, March 2017.
- [18] R1-1718516, "On remaining issues of DM-RS for NR physical data channels". Nokia, RAN1 #90bis

- [19] R1-1720808, “Remaining details on DM-RS”. NTT DoCoMo, RAN1 #91.
- [20] 3GPP TS 38.211, “3rd generation partnership project; Technical specification group radio access network; Physical channels and modulation”, release 15.