Design Specification

Editor: Linus Mellberg

Version 1.1

Status

Reviewed	Jonas Karhu	2013-10-16
Approved	Christopher Mollén	

PROJECT IDENTITY

2013HT, Signalspanarna
Linköping University, Department of Electrical Engineering (ISY)
Division of Communication Systems (CommSys)

Group members

Name	**Responsibility**	**Phone**	**Email**
Robin Carlsson	Project Leader (PL)	076-2709056	robca383@student.liu.se
Sebastian Faxér	LTE Expert (LE)	073-5858550	sebfa154@student.liu.se
Linus Mellberg	Chief Designer (CD)	070-3916857	linme560@student.liu.se
Henrik Rydén	Test Manager (TM)	070-8132759	henpe450@student.liu.se
Jonas Karhu	Head of Documentation (DOC)	070-2816120	jonka076@student.liu.se

Email list for the whole group: signalspanarna@googlegroups.com
Web site: To be announced

Customer: Christopher Mollén, christopher.mollen@liu.se
Course leader: Danyo Danev, danyo@isy.liu.se
Supervisor: Reza Moosavi, reza@isy.liu.se

Contents

Document history

Version	Date	Changes	Sign	Reviewed
0.1	2013-10-11	First draft	RC	JK
1.0	2013-10-13	Minor corrections	JK	RC
1.1	2013-10-16	Corrections according to customer	JK	RC

1 Introduction

This document provides a detailed description of all the blocks that together form the product MOUCE 2.0. This product will use an antenna connected to a USRP to receive LTE signals. These signals will then be sent to a computer to be processed and analyzed. Information about operator, base station and signal strength will be extracted from the signal and presented in a User Interface. This information can also be uploaded to a web page, where anyone can read more about MOUCE 2.0 and how it works.

2 Overview

The system consists of three sub systems, which are interconnected according to the figure below. The USRP and the computer communicate through a USB cable. The antenna receives radio signals, the USRP samples them, transforms them to complex baseband signals and sends them to the computer, where a program will process and analyse the signals. The signal processing is done in several steps – synchronization, channel estimation, channel equalization and decoding – and the bits are then interpreted using LTE protocols. The data that is extracted from the signals will then be presented to the user via the User Interface. Sub System II is further divided into smaller blocks, which are described in detail in Section 5.

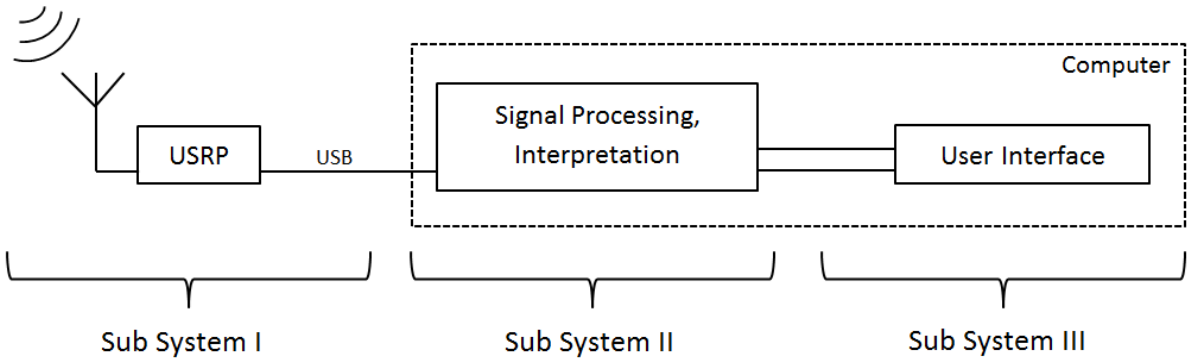


Figure 1: Overview of the system.

3 LTE

This sections contains a relevant part of the theory behind LTE that is needed to design the product.

3.1 Swedish LTE Network Operators

There are a number of Swedish LTE network operators. Table 1 displays which frequency bands they occupy. This table is taken from [1].

Frequency band (MHz)	Network Operator
2500-2520/2620-2640	Net4Mobility
2520-2530/2640-2650	Hi3G (Tre)
2530-2550/2650-2670	TeliaSonera
2550-2570/2670-2690	Net4Mobility
2570-2620 (TDD)	Hi3G (Tre)

Table 1: Frequency band of Swedish LTE network operators.

3.2 OFDM

LTE signals are modulated using Orthogonal Frequency Division Multiplexing (OFDM). In OFDM, the message to be sent is first mapped to complex-valued modulation symbols a_k. Each symbol is modulated with its own subcarrier $e^{j2\pi\Delta fkt}$ during a symbol time T_{symbol}. All subcarriers are orthogonal to each other, which is essential in OFDM. Orthogonality is accomplished by choosing a subcarrier spacing of $\Delta f = \frac{l}{T_{symbol}}$, where T_{symbol} is the symbol time and l is an integer, usually 1. The modulated symbols are multiplexed to the resulting signal $\sum_{k=0}^{N-1} a_k e^{j2\pi\Delta fkt}$ for $0 \leq t \leq T_{symbol}$.

3.3 Frames, Slots and Resource Blocks

The LTE transmissions are organized into radio frames of $T_{frame} = 307200 \cdot T_s = 10$ ms, where the sample time is defined as $T_s = \frac{1}{15\,000 \cdot 2048} \approx 3.26 \cdot 10^{-8}$ seconds. Each frame is then divided into 10 subframes of 1 ms each. Each subframe is further split up into two slots of equal length. These slots contain either six or seven OFDM symbols, depending on the use of a *normal* or *extended* cyclic prefix, this is illustrated in Figure 2. LTE uses an OFDM subcarrier spacing of $\Delta f = 15$ kHz in both down- and uplink. This corresponds to an OFDM symbol time of $T_{symbol} = \frac{1}{\Delta f} = 2048 \cdot T_s \approx 66.7$ μs, [2, Ch. 4].

Note that when using the normal cyclic prefix length, the length of the cyclic prefix for the first OFDM symbol in each slot is slightly longer than for the rest. This is simply to fill out the slot time of 0.5 ms. The reason why two cyclic prefix lengths are supported is that a longer cyclic prefix provides more robustness against a time-dispersive channel (i.e a channel subject to multi-path propagation) and may be beneficial in situations with harsh channel conditions. However, in this project we shall only consider LTE signals using the normal cyclic prefix. We motivate this by the fact that the normal cyclic prefix length is almost always employed since it increases the spectral efficiency.

The smallest physical resource defined in LTE is a resource element consisting of one subcarrier during an OFDM symbol. These are further grouped together into resource blocks of 12 consecutive subcarriers during one 0.5 ms slot. Two consecutive resource blocks within a subframe form a resource block pair and it is the minimum unit used for scheduling, [3, Ch 9.1].

3.4 Logical, Transport and Physical Channels

The LTE protocol classifies channels into the following groups.

Logical Channels: Define *what type* of information is transmitted over the air, e.g. traffic channels, control channels, broadcast channel, etc. Data and signaling messages are carried on logical channels between the RLC and MAC protocols.

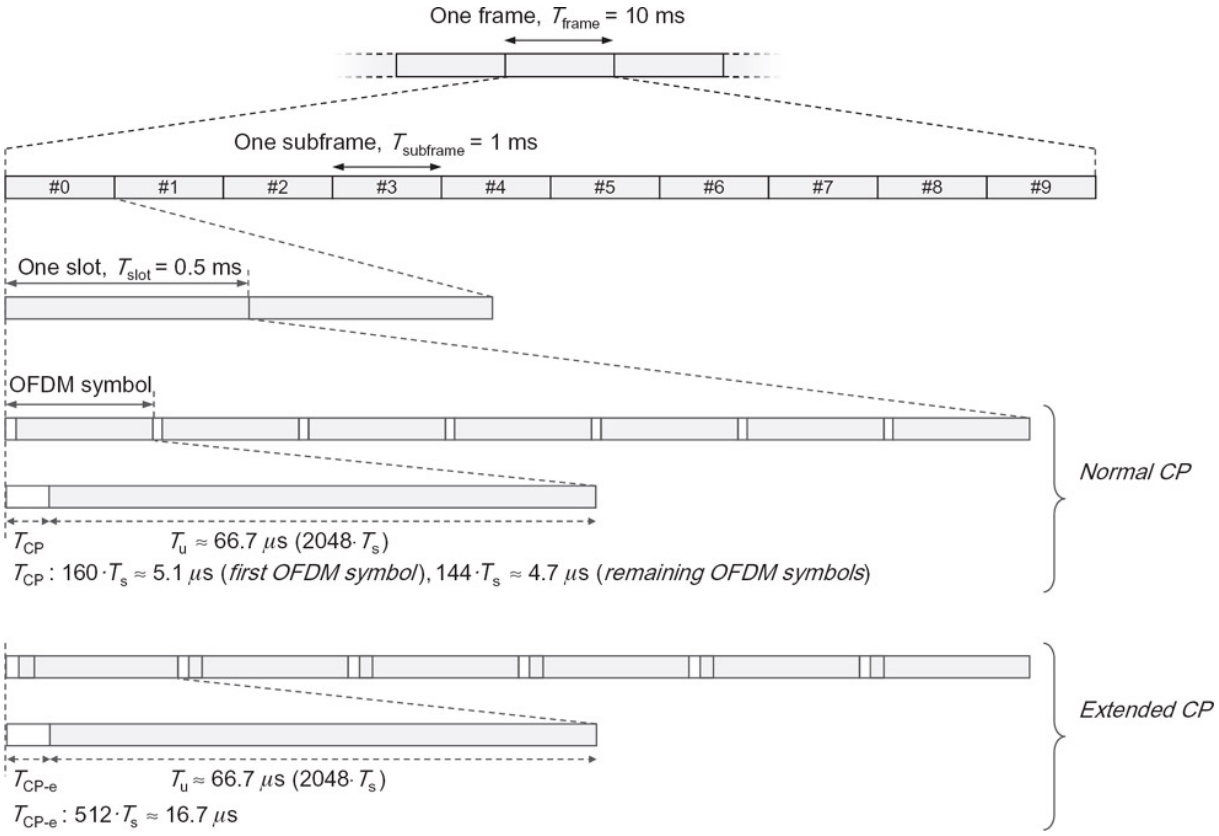


Figure 2: The structure of a radio frame, [3, Figure 9.1].

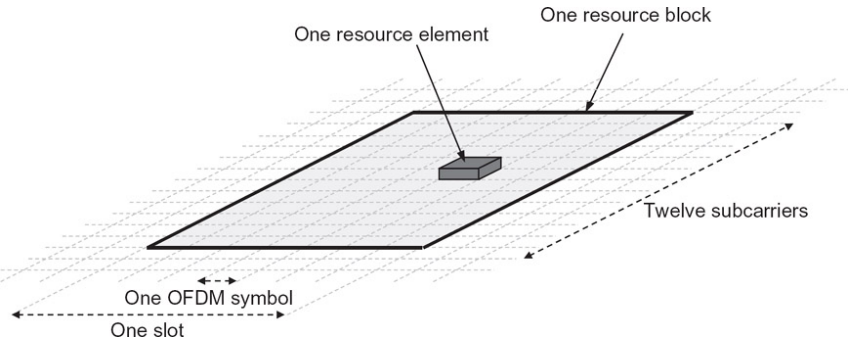


Figure 3: A resource block in the time-frequency grid, [3, Figure 9.2].

Transport Channels: Define *how* something is transmitted over the air, e.g. what encoding, interleaving options are used to transmit data. Data and signaling messages are carried on transport channels between the MAC and the physical layer.

Physical Channels: Define *where* something is transmitted over the air, e.g. first N symbols in the data link frame, what modulation format (QPSK/16-QAM/...) is used etc. Data and signalling messages are carried on physical channels between the different levels of the physical layer.

These groups define a protocol stack with the MAC layer at the top, with access to the logical channels and the RLC at the bottom, which are accessed by the physical channels. Figure 4 shows how different channels are mapped through the protocol stack. In [3], a more thorough explanation of the channels can be found, although for us the interesting part is the BCCH-BCH-PBCH relation, since the MIB is located there.

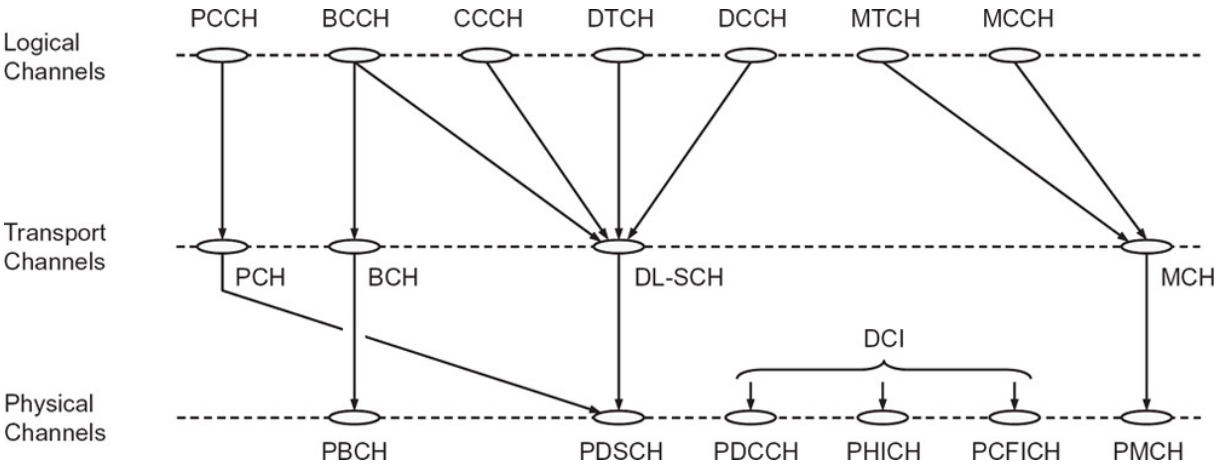


Figure 4: The protocol stack in LTE, [3, Figure 9.2].

The physical channels are mapped onto the resource grid differently according to the cell configuration. An example of how they can be mapped is depicted in Figure 5.

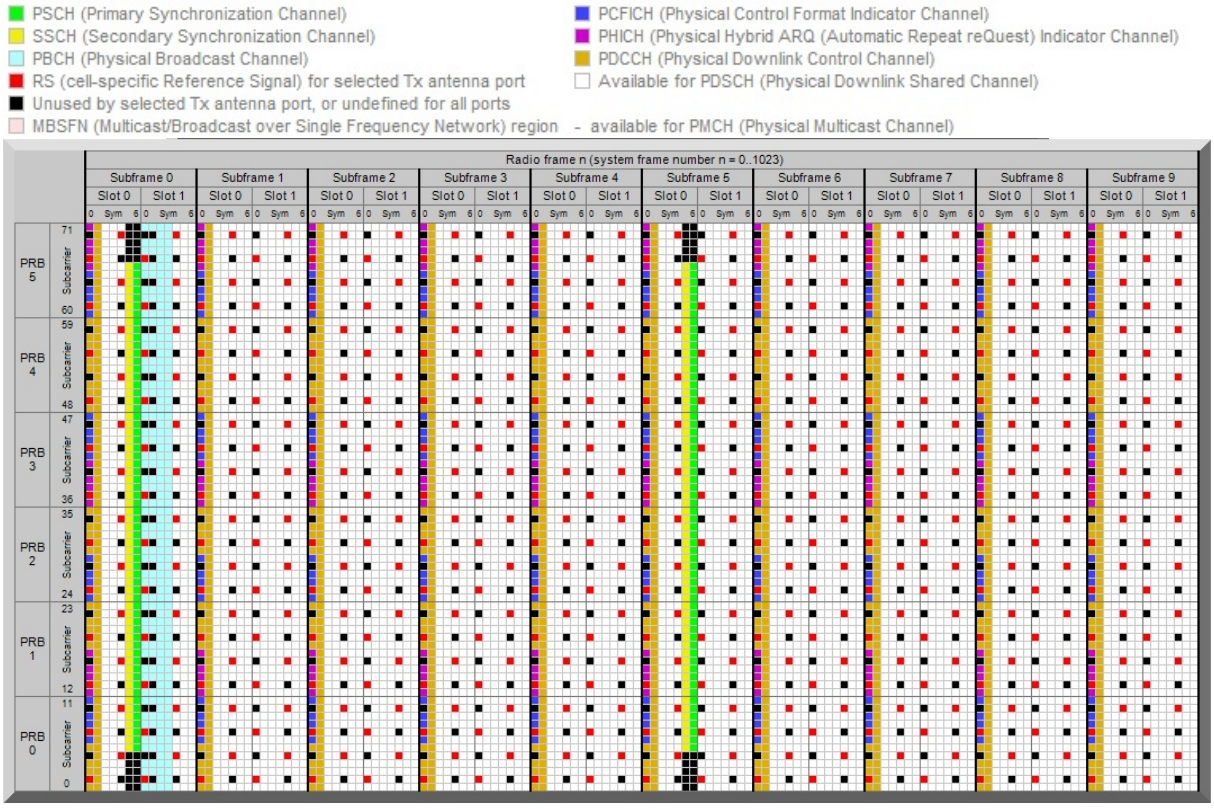


Figure 5: LTE Resource Grid. Source: [4]

The physical channels take sequences $\{d(i)\} = d(0), d(1), ...$ of coded bits (known as codewords) from the transport channels. They then scramble the codeword with a cell-specific scrambling sequence and map it onto complex valued modulation symbols. These are then mapped onto one or several layers $x(i)^{(0)}, x(i)^{(1)}, ...$ depending on the transmission mode. In case of single antenna transmission, one codeword is mapped onto one layer. In transmit diversity transmission modes, the modulation symbols in one codeword are distributed evenly on 2 or 4 layers. For spatial multiplexing, one or two codewords are distributed onto 1-4 layers.

For each sequence index i, the modulation symbols on the layers $x(i)^{(0)}, x(i)^{(1)}, ...$ are first precoded by means of multiplying by a precoding matrix, different for each transmission mode, and then mapped onto antenna ports $y(i)^{(0)}, y(i)^{(1)}, ...$ On each antenna port, the modulation symbols are mapped onto resource elements in the time-frequency grid.

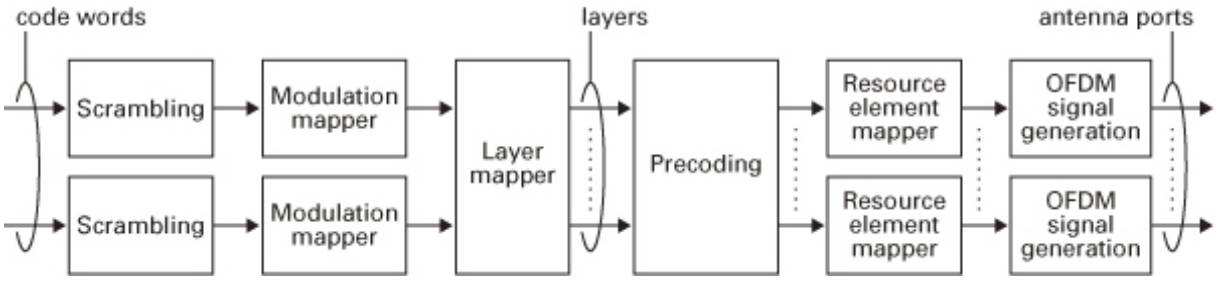


Figure 6: Physical downlink channel processing.

We need to access the Master Information Block (MIB) on the BCCH (Logical channel) → BCH (Transport channel) → PBCH (Physical channel) which contains:

- Downlink channel bandwidth in terms of resource blocks (RBs)
- PHICH configuration (PHICH duration and PHICH resource)
- Frame Number

We then need to get the System Information Block 1 (SIB1) to get the operator information. We can do this once the MIB has been successfully decoded, then we can decode the CFI (indicating PDCCH length). The PDCCH can be demodulated and the DCI messages decoded, allowing us to search for DCI messages scrambled with SI-RNTI, in order to find the System Information.

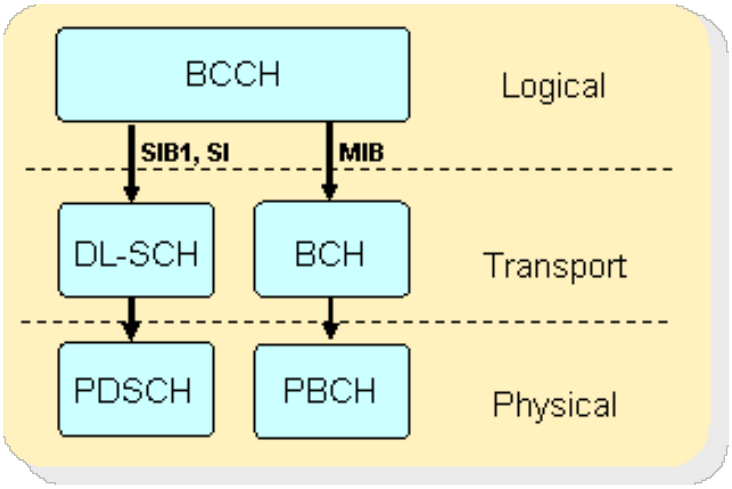


Figure 7: The broadcast channel in different layers of the protocol stack.

3.5 Synchronization Signals

This section describes the two synchronization signals that will enable adequate synchronization, namely PSS and SSS.

3.5.1 Primary Synchronization Signal

The primary synchronization signal (PSS) is located in the last OFDM symbol of the first time slot of the first subframe (subframe 0) of a radio frame as shown in Figure 5 (green squares). This enables the UE to be synchronized on subframe level. The PSS is repeated in subframe 5, which means that the UE is synchronized on a 5 ms basis, since each subframe is 1 ms. From the PSS, the UE is able to obtain sector identity ($N_s = 0$ to 2). The sector identity is obtained by correlations with known Zadoff-Chu sequences.

3.5.2 Secondary Synchronization Signal

The secondary synchronization signal (SSS) consists of a frequency domain sequence $d(n)$ with the same length as the PSS, which is an interleaved concatenation of the two length-31 binary sequences $s_0(n)$ and $s_1(n)$. In order to distinguish between different sector groups (physical cells), $s_0(n)$ and $s_1(n)$ depend on a pair of integers m_0 and m_1, which are unique for each group ID N_g (from 0 to 167). The concatenated sequences are scrambled with one of the sequences $c_0(n)$ and $c_1(n)$, which are cyclic shifted versions of the length-31 binary sequence $\tilde{c}(n)$. The shift value is depending on the sector-ID N_s, while a constant shift of 3 samples holds between $c_0(n)$ and $c_1(n)$.

Further, a pair of scrambling sequences $z_1^{m_0}(n)$ and $z_1^{m_1}(n)$ (cyclic shifted versions of the sequence $\tilde{z}(n)$), which also depend on N_g is multiplied with the odd entries of the SSS. For the definitions of these sequences, see [2, Chapter 6.11.2].

In order to enable the detection of the beginning of a radio frame, the SSS is different for each subframe index (0 or 5), thus the final SSS sequence $d(n)$ is given by

$$d(2n) = \begin{cases} s_0^{m_0}(n)c_0(n) & \text{in subframe 0} \\ s_1^{m_1}(n)c_0(n) & \text{in subframe 5} \end{cases}$$

$$d(2n+1) = \begin{cases} s_1^{m_1}(n)c_1(n)z_1^{m_0}(n) & \text{in subframe 0} \\ s_0^{m_0}(n)c_1(n)z_1^{m_1}(n) & \text{in subframe 5} \end{cases}$$

Since $d(n)$ is real valued, time domain symmetry always holds for the SSS. The overall cell-ID N_c (from 0 to 503) is then defined by the sector and group identities N_s and N_g as $N_c = 3N_g + N_s$.

3.6 Cell-Specific Reference Signals

To enable the receiver to estimate the channel, cell-specific Reference Signals (RSs) are transmitted by each antenna port at even intervals in the time-frequency grid. Cell-specific reference sequences $P(n)$ consist of complex-valued entries defined by

$$P_{l,n_s}(n) = \frac{1}{\sqrt{2}}(1 - 2c(2n)) + j\frac{1}{\sqrt{2}}(1 - 2c(2n+1)), \quad n = 0, ..., 2N_{RB}^{max} - 1$$

where N_{RB}^{max} is the maximum number of resource blocks, n_s is the slot index within the radio frame and l is the OFDM symbol index within the slot. The pseudo random sequence $c(n)$ is generated by a length-31 Gold sequence, the state of which is initialized by c_{init} at the beginning of each OFDM symbol. The initializing value is given by

$$c_{init} = 2^{10} \cdot (7 \cdot (n_s + 1) + l + 1) \cdot (2N_c + 1) + 2N_c + N_{CP}$$

where

$$N_{CP} = \begin{cases} 0, & \text{for normal cyclic prefix,} \\ 1, & \text{for extended cyclic prefix.} \end{cases}$$

This indicates that $c(n)$ and consequently the reference sequence $P(n)$ are unique for each slot, OFDM symbol and cell index, and they are also dependent on the cyclic prefix type.

The sequences $P_{l,n_s}(n)$ are mapped onto complex valued modulations symbols $a_{k,l}^{(p)}$ at resource elements (k,l) in slot n_s on antenna port p according to

$$a_{k,l}^{(p)} = P_{l,n_s}(m)$$

where

$$k = 6n + ((v + v_{shift}) \mod 6)$$

$$l = \begin{cases} 0, N_{symb} - 3 & \text{if } p = 0, 1 \\ 1 & \text{if } p = 2, 3 \end{cases}$$

$$n = 0, ..., 2N_{RB}^{max} - 1$$

$$m = n + N_{RB}^{max} - N_{RB}$$

where $N_{symb} = 7$ is the number of OFDM symbols in a slot and N_{RB} is the number of resource blocks in use by the cell. The variables v and v_{shift} define the position in the frequency domain for the different reference signals where v is given by

$$v = \begin{cases} 0 & \text{if } p = 0 \text{ and } l = 0 \\ 3 & \text{if } p = 0 \text{ and } l \neq 0 \\ 3 & \text{if } p = 1 \text{ and } l = 0 \\ 0 & \text{if } p = 1 \text{ and } l \neq 0 \\ 3(n_s \mod 2) & \text{if } p = 2 \\ 3 + 3(n_s \mod 2) & \text{if } p = 3 \end{cases}$$

and

$$v_{shift} = N_c \mod 6.$$

Resource elements (k,l), used for transmission of cell-specific reference signals on any of the antenna ports in a slot, shall not be used for transmission on any other antenna ports in the same slot and is thus set to zero.

3.7 MIB Aquiring on the BCH and PBCH

The master information block (MIB) is transmitted on the BCH transport channel and PBCH physical channel. The MIB, consisting of 14 information bits and 10 spare bits, is first subject to BCH processing. It is appended by a 16 bit CRC-code, coded using a rate 1/3 tail biting convolutional code and finally rate matched (repetition coded) up to 1920 bits. The coded BCH transport block is mapped onto four subframes within a 40 ms interval. 40 ms timing is blindly detected, i.e. there is no explicit signalling indicating 40 ms timing. Each subframe is assumed to be self-decodable, i.e. the BCH can be decoded from a single reception, assuming sufficiently good channel conditions.

Before being mapped onto resource elements, the coded BCH transport block is subject to PBCH processing. It is first scrambled by a cell specific scrambling sequence. This scrambling sequence is reset after every 4^{th} frame (every 40 ms). When the PBCH is to be decoded by the

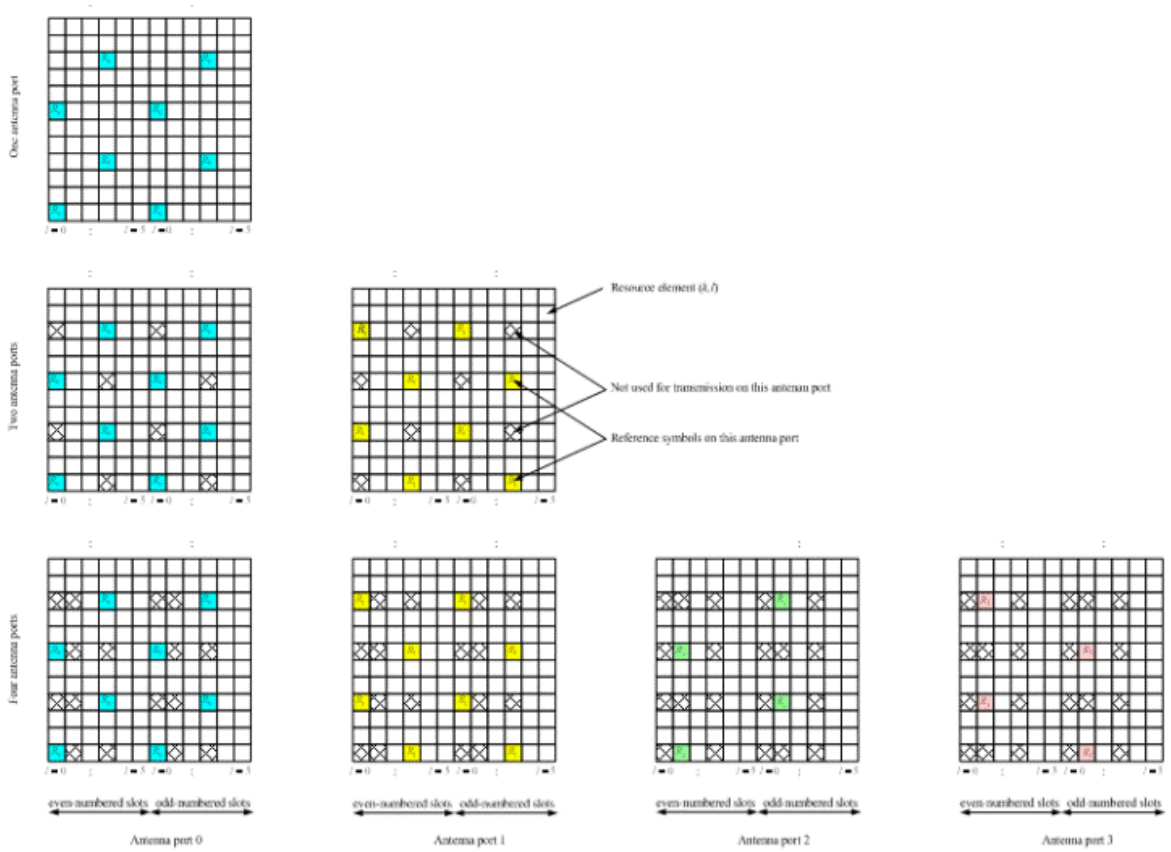


Figure 8: The reference signal structure with 1, 2 and 4 antennas used.

receiver, 40 ms timing is not yet achieved, i.e. we do not know which phase of the scrambling sequence the PBCH on the current frame is scrambled with.

The scrambled bits are further QPSK-modulated and mapped onto antenna ports according to a transmit diversity scheme. The antenna mapping, i.e layer mapping and precoding, is different depending on how many transmit antennas are used at the evolved node B (eNodeB). With two Tx antennas a space frequency block code (SFBC) is used and with 4 Tx antennas a SFBC is used in combination with frequency switched transmit diversity (FSTD). These schemes are explained in section 3.11. It is also possible for the eNodeB to use only one Tx antenna, where the QPSK symbols are mapped directly to the antenna port with no precoding.

The PBCH is mapped onto 4 OFDM symbols in the second slot of the first subframe in the time domain at 6 RBs, (72 subcarriers) excluding DC in the frequency domain. It is mapped onto resource elements, assuming reference signals from 4 antennas are used at the eNodeB, irrespective of the actual number of Tx antenna. See Figures 9 and 10.

The number of transmit antenna ports used by the eNodeB must be ascertained blindly by the receiver, by performing the decoding for each SFBC scheme corresponding to the different possible numbers of transmit antenna ports (namely 1, 2 or 4). This discovery of the number of transmit antenna ports is further facilitated by the fact that the cyclic redundancy check (CRC) on each MIB is masked with a codeword representing the number of transmit antenna ports, according to the table below.

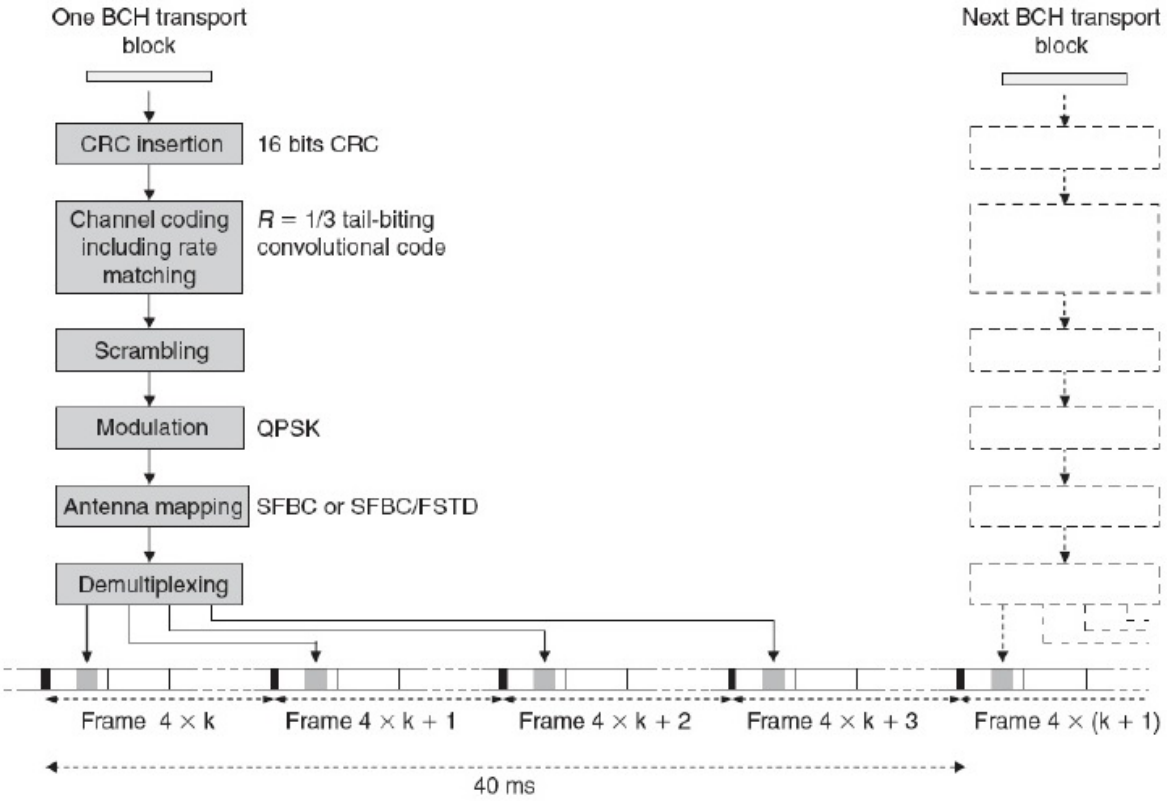


Figure 9: Process chain of the MIB, [3].

Number of transmit antenna ports at eNode-B	**PBCH CRC mask** $< x_{ant,0}, x_{ant,1}, ..., x_{ant,15} >$
1	$< 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 >$
2	$< 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 >$
4	$< 0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1 >$

The timing of the 40 ms transmission interval for each MIB on the PBCH is not indicated explicitly to the receiver, but it is ascertained implicitly from the scrambling and bit positions, which are re-initialized every 40 ms. The receiver can therefore initially determine the 40 ms timing, by performing four separate decodings of the PBCH, using each of the four possible phases of the PBCH scrambling code and then check the CRC for each decoding. Another approach is to perform the decoding using a soft combination of the PBCH over four radio frames, advancing a 40 ms sliding window one radio frame at a time until the window aligns with the 40 ms period of the PBCH and the decoding succeeds, [2]. The latter is the one that will be used.

3.8 CFI Aquiring on the PCFICH

When the MIB is read, the channel bandwidth is known and the location of the physical control format indicator channel (PCFICH) can be deduced. The next step is to decode the PCFICH and read its payload, the control format indicator (CFI) value, which can be either 1,2 or 3. For bandwidths greater than ten resource blocks, the number of OFDM symbols used to contain the downlink control information (PDCCH size) is the same as the actual CFI value. Otherwise, the span of the downlink control information is CFI+1 symbols. The PCFICH is mapped in terms of resource element groups (REGs) and is always mapped onto the first OFDM symbol. The

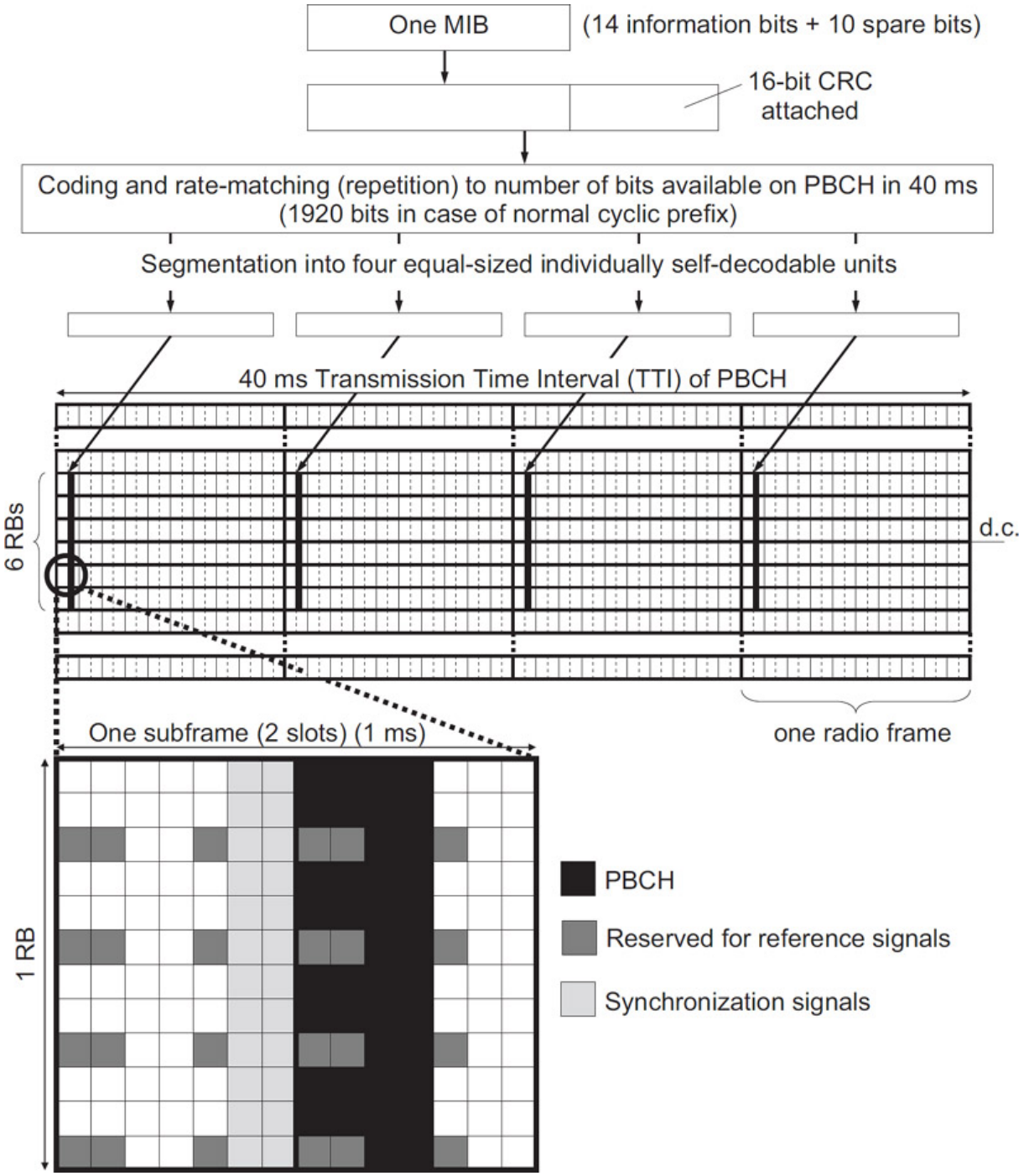


Figure 10: Placement of the MIB in the resource grid.

size of the control region can vary from subframe to subframe, so the PCFICH must be decoded every subframe.

A REG is made up of four resource elements (REs) on consecutive subcarriers in an OFDM symbol that is not occupied by a RS. The REGs are indentified by the pair (k, l), where k is the subcarrier index of the RE within the REG with the lowest subcarrier index and l the OFDM symbol index whitin the slot. The number of REGs allocated to the PCFICH transmission is fixed to 4, i.e. 16 REs. A PCFICH is only transmitted when the number of OFDM symbols for PDCCH is greater than zero.

The 2-bit CFI payload is first mapped onto a 32 bit codeword according to this table.

CFI	**CFI codeword** $< b_0, b_1, ..., b_{31} >$
1	$< 0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1 >$
2	$< 1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0 >$
3	$< 1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1 >$

The 32-bit codeword is then scrambled with a pseudo-random sequence created using a length-31 Gold sequence generator. At the start of each subframe, it is initialised using the slot number n_s within the radio frame and the cell ID N_c as

$$c_{init} = (\lfloor n_s/2 \rfloor) \cdot (2N_c + 1) \cdot 2 + N_c.$$

The scrambled sequence is then QPSK modulated. The resulting symbols are layer mapped and precoded according to the correct transmit diversity scheme, depending on the known number of Tx antennas, as described in section 3.11.

The complex valued symbols for each antenna are divided into quadruplets for mapping onto resource elements. Each quadruplet is mapped onto a resource element group (REG) within the first OFDM symbol. There are sixteen complex symbols to be mapped, therefore four quadruplets are created.

The first quadruplet is mapped onto a REG with subcarrier index $k = (N_{sc}^{RB}/2) \cdot (N_c \mod 2N_{RB})$, where $N_{sc}^{RB} = 12$ is the number of subcarriers per resource block and N_{RB} is the cell bandwidth expressed in multiples of N_{sc}^{RB}.

The subsequent three quadruplets are mapped onto REGs spaced at intervals of $\lfloor N_{RB}/2 \rfloor \cdot (N_{sc}^{RB}/2)$ from the first quadruplet and equally spaced between each other. This spreads the quadruplets and therefore the PCFICH over the entire subframe, this is illustrated in Figure 11.

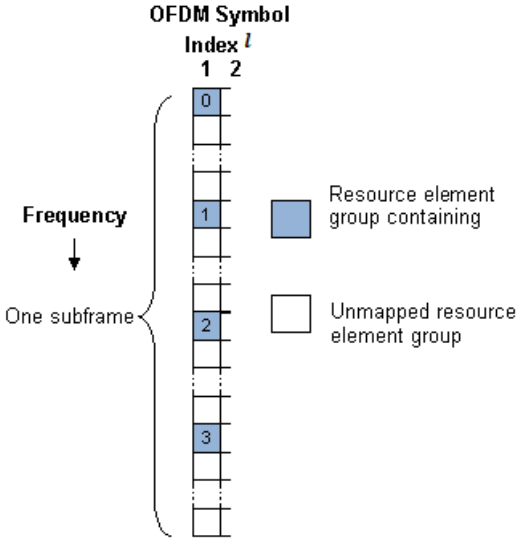


Figure 11: Spread of the PCFICH's REGs.

3.9 Scanning the PDCCH for the SIB1 DCI

On the physical downlink control channel (PDCCH), downlink scheduling control information (DCI) messages are transmitted. The DCI messages contain information about scheduling assignments for each UE (as well as the scheduling of system wide control information). It also contains information such as the modulation and coding scheme used, that is necessary to decode the transmissions. To distinguish which DCI messages are addressed to which UE, the DCI messages are CRC appended and the parity bits scrambled with a radio network temporary identifier (RNTI) corresponding to the UE. The UE tries to decode the DCI messages transmitted on the PDCCH by descrambling the parity bits with its RNTI. If the CRC calculation succeeds, the DCI message was directed to the UE.

The location of the SIB1 is transmitted on a DCI scrambled with a system information identifier (SI-RNTI). The DCIs can be in different formats, the DCI containing the SIB1 location can be either 'Format1A' or 'Format1C'.

A PDCCH is transmitted on one or an aggregation of several consecutive control channel elements (CCEs). A CCE is a group of nine consecutive REGs. The number of CCEs used to carry a PDCCH is controlled by the PDCCH format, depending on the CFI. A PDCCH format of 0, 1, 2, or 3 corresponds to 1, 2, 3 or 4 consecutive CCEs being allocated to one PDCCH.

The PDCCH region consists of CCEs, which could be allocated to a PDCCH. The configuration of how PDCCHs are mapped to CCEs is flexible. Common and UE-specific PDCCHs are mapped to CCEs differently; each type has a specific set of search spaces associated with it. Each search space consists of a group of consecutive CCEs, which could be allocated to a PDCCH called a PDCCH candidate. The CCE aggregation level is given by the PDCCH format and determines the number of PDCCH candidates in a search space.

The common search space consists of 16 CCEs and the CCEs can be aggregated to either 4 or 8 CCEs. The PDCCH decoding block will have to search each possible place where a PDCCH could be transmitted and try to decode it to try to find the SIB1-DCI. It will know if it is found by checking the CRC checksum.

After the DCI has been CRC appended, it is coded by the same tail-biting convolutional encoder and rate-matching scheme as the MIB. It is then scrambled by a length-31 Gold sequence initialized with

$$c_{init} = \lfloor n_s/2 \rfloor 2^9 + N_c$$

at the start of each subframe.

The scrambled sequence is then QPSK modulated, layer mapped and precoded according to the Tx diversity scheme used. The complex valued symbols for each antenna are then divided into quadruplets for mapping to resource elements. The sets of quadruplets then undergo interleaving and cyclic shifting before being mapped to resource elements (REs) within resource-element groups (REGs). See [2, 6.8] for details.

3.10 SIB1 Aquiring on the PDSCH

The SIB1 DCI will tell us where on the PDSCH to look for the SIB1. On the PDSCH each subframe of 1 ms corresponds to a *Transmission Time Interval (TTI)*. The data to be transmitted are organized into *transport blocks*. In each TTI either one or two transport blocks are transmitted, depending on if spatial multiplexing is used. The signal processing chain is illustrated in Figure 12 [3, Ch 10.1].

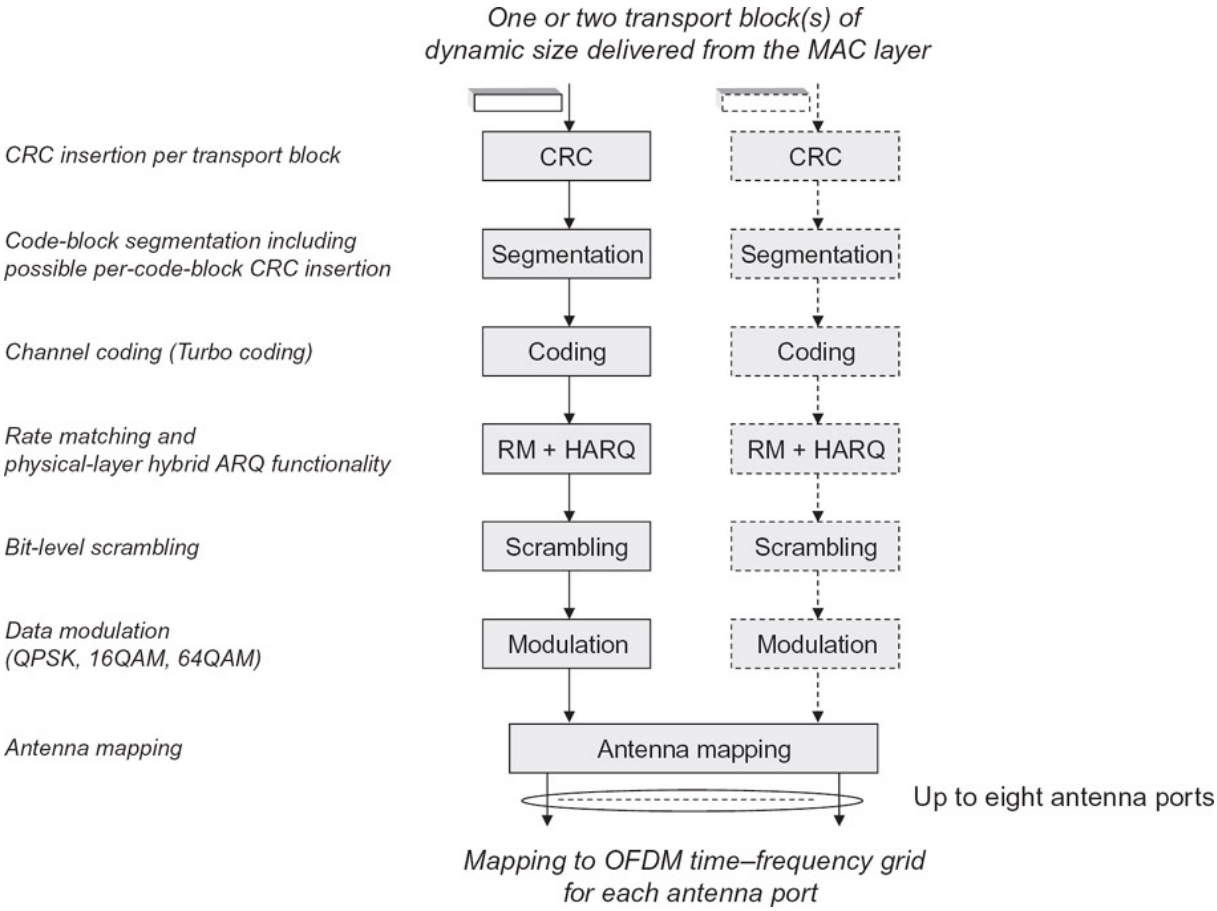


Figure 12: Signal processing chain for the DL-SCH. Source: [3, Figure 10.1]

- **CRC Insertion**
  In the first step, a 24-bit *Cyclic Redundancy Check (CRC)*-code is appended to each transport block, which forms a basic error-detection capability.

- **Code-Block Segmentation**
  The internal interleaver of the Turbo-coder used in the following step only allows for a limited number of code-block sizes, where the maximum is 6144 bits. If the CRC-appended transport block is longer than this maximum value, it is segmented into smaller *code blocks*. For each of these code blocks, a 24-bit CRC-code is calculated as well and the resulting CRC-appended code blocks are sent to the next step in the chain.

- **Turbo Coding**
  For channel coding, a Turbo encoder as illustrated in Figure 14 is used. The coder consists of two rate $\frac{1}{2}$ convolutional encoders with three memory elements. As seen, this corresponds to a total rate of $\frac{1}{3}$. The input to the second encoder is interleaved by a *Quadrature Permutation Polynomial (QPP)*-interleaver, changing the order of the bits according to a pre-defined formula.

- **Rate Matching and Hybrid-ARQ**
  This block selects which of the output bits of the encoder to actually transmit. As illustrated in Figure 15, the three outputs from the turbo encoder are first interleaved separately and then put in what can be seen as a circular buffer. Then a number of consecutive bits from the buffer are chosen for transmission depending on the desired code rate. If

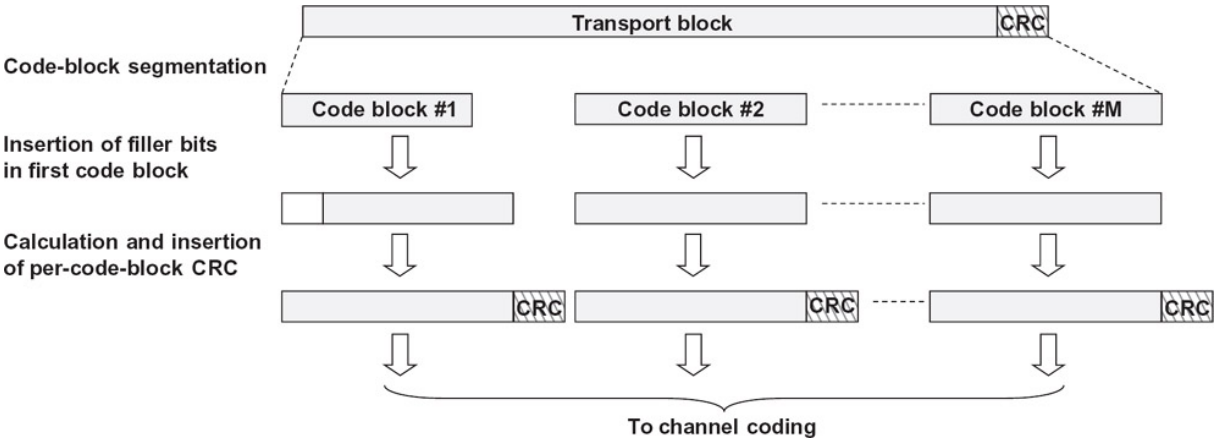


Figure 13: Segmentation of a transport block. Source: [3, Figure 10.2]

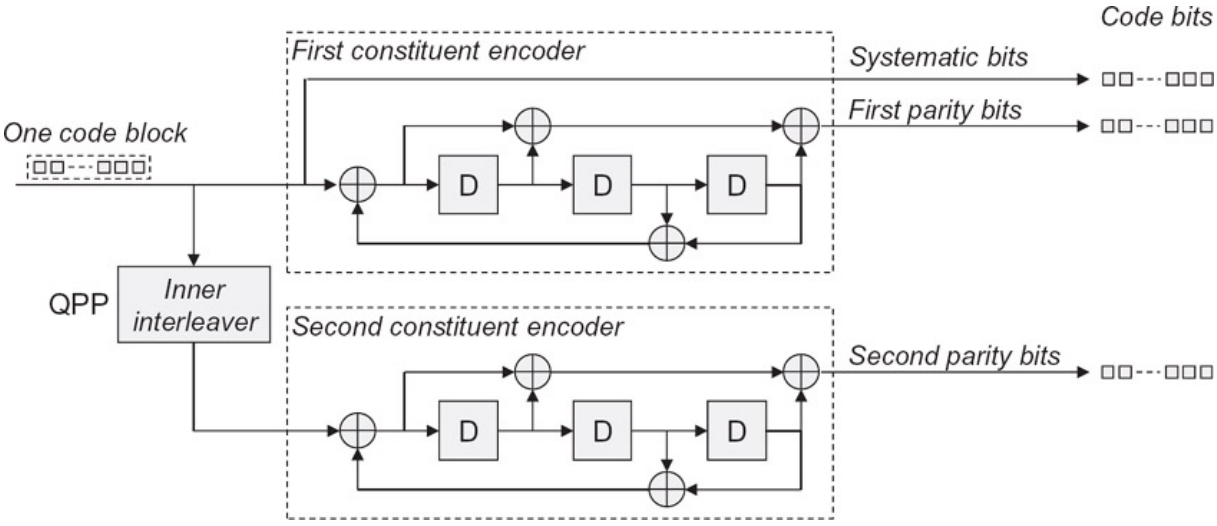


Figure 14: The Turbo encoder used in LTE. Source: [3, Figure 10.3]

a low number of bits is chosen for transmission it corresponds to a high rate code and vice versa. The starting point where to begin choosing bits in the buffer depends on the *Redundancy Version (RV)*. As illustrated in the figure there are four choices of the RV, each corresponding to a starting point in the buffer.[5, Ch. 5.1].

- **Scrambling**
  Next, the block of bits are multiplied bitwise by a *scrambling sequence*. The purpose of this is to make the resulting sequence more *random-like*, and by applying different scrambling sequences at neighbouring cells the interference between them can be minimized.

- **Modulation**
  The scrambled bits are modulated using QPSK, 16QAM or 64QAM. What modulation order to use is specified in the DCIs.

- **Antenna Mapping**
  The resulting modulation symbols are then mapped onto different antenna ports. Several antenna mappings, called *transmission modes*, are possible and depending on which one is used, transmit diversity, beamforming and/or spatial multiplexing can be achieved.

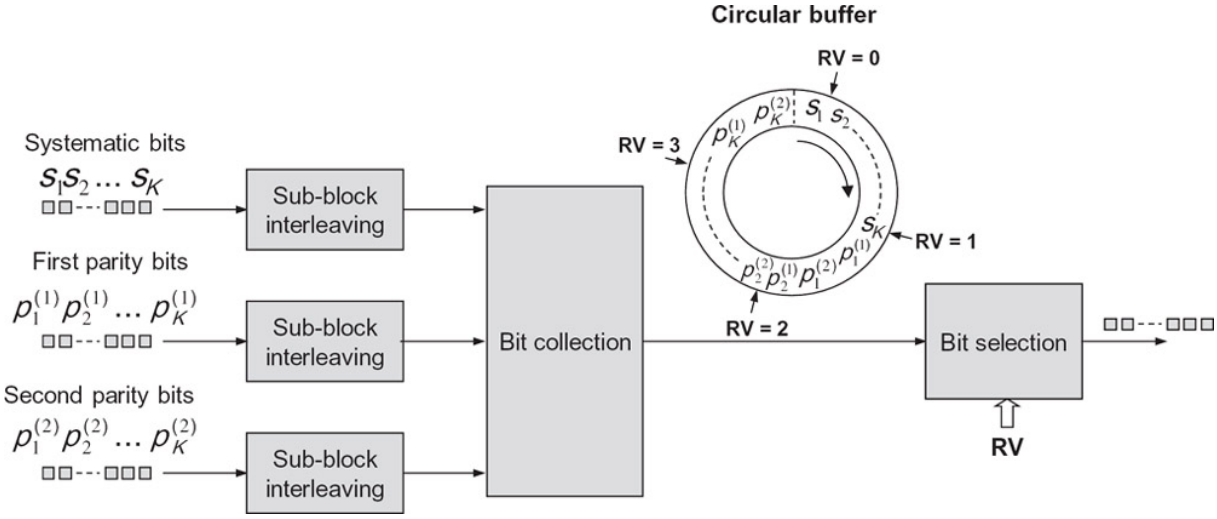


Figure 15: Rate matching. Source: [3, Figure 10.5]

- **Resource-Block Mapping**
  The symbols assigned to each antenna port is mapped onto the resource elements belonging to the set of resource blocks assigned for the transmission [2, Ch 6.3].

3.11 Transmit Diversity Schemes

If a physical channel in LTE is configured for transmit diversity operation using two eNodeB antennas, pure SFBC is used. SFBC is a frequency domain version of the well known space-time block codes (STBCs), also known as Alamouti codes. This family of codes are designed so that the transmitted diversity streams are orthogonal and achieve the optimal SNR with a linear receiver. Such orthogonal codes only exist for the case of two transmit antennas.

STBC is used in UMTS, but in LTE the number of available OFDM symbols in a subframe is often odd while STBC operates on pairs of adjacent symbols in the time domain. The application of STBC is therefore not straightforward for LTE, while the multiple subcarriers of OFDM lend themselves well to the application of SFBC.

For SFBC transmission in LTE, the symbols transmitted from the two eNodeB antenna ports on each pair of adjacent subcarriers are defined as

$$\begin{bmatrix} y^{(0)}(1) & y^{(0)}(2) \\ y^{(1)}(1) & y^{(1)}(2) \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix}$$

where $y^{(p)}(k)$ denotes the symbols transmitted from antenna port p on the k^{th} subcarrier.

Since no orthogonal codes exist for antenna configurations beyond 2 x 2, SFBC has to be modified in order to apply it to the case of 4 transmit antennas. In LTE, this is achieved by combining SFBC with frequency switched transmit diversity (FSTD).

General FSTD schemes transmit symbols from each antenna on a different set of subcarriers. In LTE, FSTD is only used in combination with SFBC for the case of 4 transmit antennas, in order to provide a suitable transmit diversity scheme, where no orthogonal rate-1 block code exists. The LTE scheme is in fact a combination of two 2 x 2 SFBC schemes mapped to independent subcarriers as

$$\begin{bmatrix} y^{(0)}(1) & y^{(0)}(2) & y^{(0)}(3) & y^{(0)}(4) \\ y^{(1)}(1) & y^{(1)}(2) & y^{(1)}(3) & y^{(1)}(4) \\ y^{(2)}(1) & y^{(2)}(2) & y^{(2)}(3) & y^{(2)}(4) \\ y^{(3)}(1) & y^{(3)}(2) & y^{(3)}(3) & y^{(3)}(4) \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & x_4 \\ -x_2^* & x_1^* & 0 & 0 \\ 0 & 0 & -x_4^* & x_3^* \end{bmatrix}$$

where, as previously, $y^{(p)}(k)$ denotes the symbols transmitted from antenna port p on the k^{th} subcarrier. Note that the mapping of symbols to antenna ports is different in the 4 transmit antenna case compared to the 2 transmit antenna SFBC scheme. This is because the density of cell-specific RSs on the third and fourth antenna ports is half that of the first and second antenna ports, hence the channel estimation accuracy may be lower on the third and fourth antenna ports. Thus, this design of the transmit diversity scheme avoids concentrating the channel estimation losses in just one of the SFBC codes, resulting in a slight coding gain.

4 Sub System I - USRP

Sub System I consists of an antenna connected to a USRP. The USRP is set up with a SBX 400-4400 MHz Rx/Tx daughterboard.

4.1 Interface

The USRP is the radio link interface of the system. It does down conversion of LTE radio signals to complex baseband samples that will be processed in Sub System II.

4.1.1 Input

To set up the USRP it needs to be supplied with

- Sample rate
- Center frequency
- Gain

These will be supplied by Sub System II. When set up, the USRP will receive radio signals from nearby base stations.

4.1.2 Output

The output of the USRP is a stream of pairs of 16 bit signed integers, which make up the I and Q part of complex baseband symbols. There is also a possibility to set the size of samples to 8 bit integers and thus double the sampling rate. This will increase the quantization error of the receiver, but might be useful to examine a greater bandwidth of the LTE signal.

5 Sub System II - Signal Processing and Interpretation

Sub System II consists of a program developed by the project group. This program shall process and interpret the data received from Sub System I and pass the result on to Sub System III.

5.1 Interface

Sub System II is connected to Sub System I via USB, it will do initialization of Sub System I and then receive data from it. Sub System II is also connected to Sub System III, this will be done via function calls.

5.1.1 Overview of the Sub System

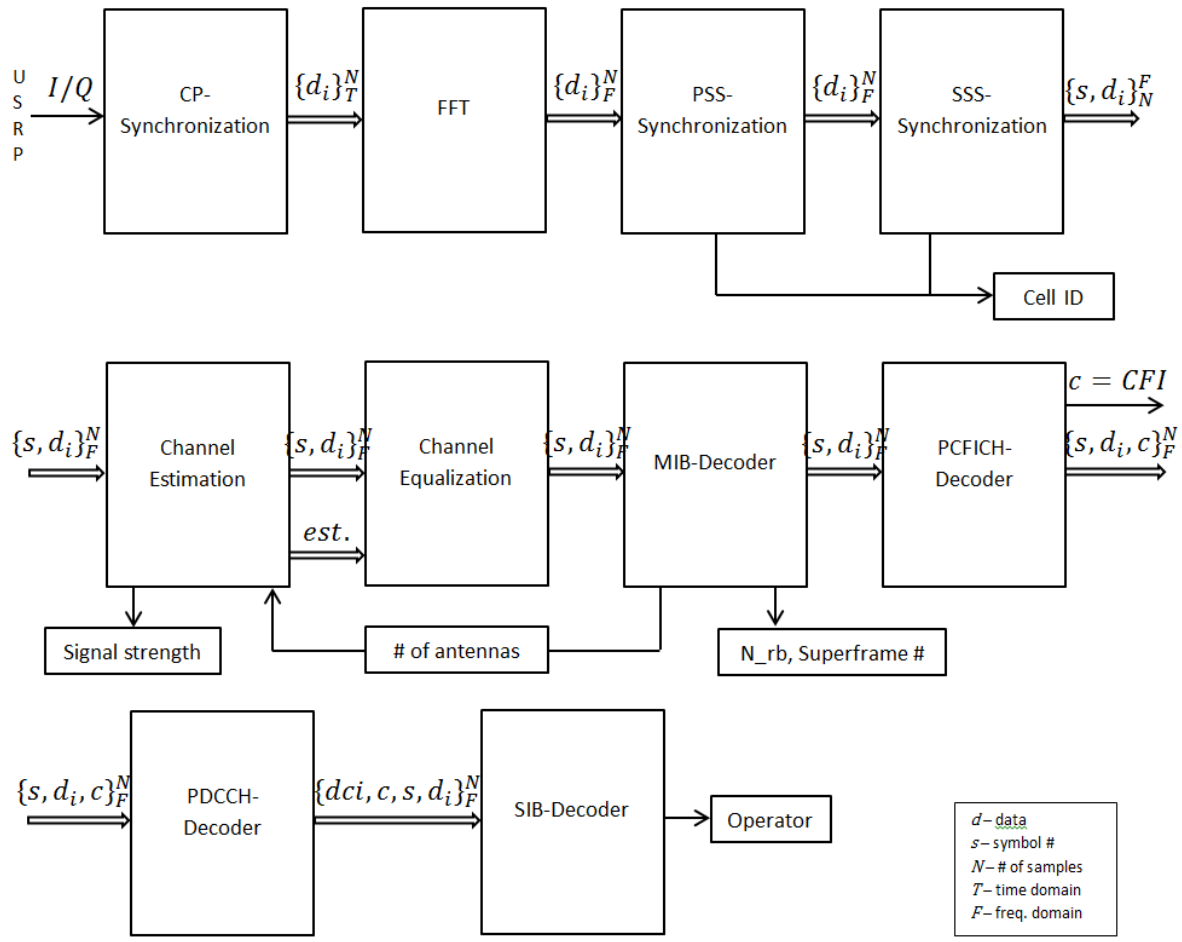


Figure 16: Block chart of Sub System II.

5.1.2 Input

From Sub System I, there will be a stream of complex baseband samples, that is, raw radio data that needs to be processed. From Sub System III there will be incoming control signals, mainly to choose a center frequency to start the base station search at.

During development and testing, there will be a need for adjusting a number of parameters for subsystem II and these will be added when needed.

5.1.3 Output

Sub System II will control Sub System I by setting frequency, sample rate and gain. Sub System II will also output the information it gathers from the radio interface to Sub System III. This information will include signal strength, network operator name and bandwidth utilization. There will also be some indication about the status of the system, what it is doing and how far in the base station discovery it has reached.

During development, there will be a need for a lot of other information with much higher detail. There will also be a need for testing of the performance of the system and this information needs to be output in a structured way.

5.2 GNU Radio, Build Systems and Tools

Sub System II will be implemented using the free and open-source software development toolkit GNU Radio. GNU Radio provides a large toolbox of software digital signal processing (DSP) tools, a USRP interface and a framework for putting them all together.

GNU Radio is a modular framework that provides signal processing blocks. These blocks can be put together to create chains or flowgraphs of DSP tools that process signals. The blocks themselves are written in C++ and are put together in flowgraphs using Python. Each block receives a stream of input items, which are then processed inside the block. Each block then outputs data that is passed forward to the next block in the flowgraph.

These blocks can also be sources or sinks. Sources produces data either from external sources (eg. a USRP, microphone or data file) or by creating it. Sinks are endpoints of data streams, which usually extract some final data and then use some external way of displaying, saving or sending it (e.g. GUI, another USRP, or a file).

GNU Radio also has support for creating custom DSP blocks to do signal processing that it cannot do with its native blocks. These are written in C++ and can, after creation, be used as any other block in a flowchart. This will be the primary way of implementing LTE signal processing in this project.

GNU Radio aids in the creation of Software Defined Radio (SDR) as it provides a framework for a lot of basic features such as buffering and handling data streams. Modularity of GNU Radio makes it possible to implement abstraction and to do signal processing sequential in an easy manner.

Unit tests are supported in the Python framework. This can be used for writing tests for custom blocks, which makes it possible to automatically test these. Testing will be necessary to ensure that blocks developed during the project behave as they should. Unit tests also make it easier to change code and still be sure that it works.

GNU Radio uses CMake, and a build manager of choice, as its build tool, which is used to build GNU Radio itself and the custom code that uses it. This project will use Gmake as build manager. GNU Radio also supplies gr-modtool, which aids in the creation of custom DSP blocks.

5.3 Synchronization

To receive LTE signals, the receiver needs to be synchronized with the base station. This is done at two levels. First at the radio level with timing of symbols and synchronization of frequency, then at the protocol level with synchronization of radio frame.

5.3.1 System Model

According to [6], the system is modeled as follows. The transmitted discrete baseband OFDM stream $s(n)$ is assumed to be given by

$$s(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi kn/N}, 1 \leq n \leq N,$$

where X_k is the modulated data on the k^{th} subcarrier and N is the FFT size. The channel exhibits multipath propagation and introduces additive white Gaussian noise (AWGN). Also,

the local oscillators (LO) in the sender and receiver will have a mismatch in phase, which has to be considered. This implies that the received signal will be

$$r(n) = [s(n) \otimes h(n) + w(n)]e^{j2\pi\epsilon n/N},$$

where $h(n)$ is the channel impulse response (CIR), ϵ is the frequency mismatch with respect to the subcarrier spacing, $w(n)$ is the noise and $\otimes$ denotes linear convolution. Further, under the assumption that there is perfect timing, after the N-point FFT, the OFDM symbol will be

$$Y_l = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} r(n)e^{-j2\pi ln/N} = \frac{1}{N} \sum_{k=0}^{N-1} H_k X_k \sum_{n=0}^{N-1} e^{2\pi n(k-l+\epsilon)/N} + W_k$$

where

$$H_k = \sum_{l=0}^{N_l-1} h_l e^{-j2\pi kl/N} \quad \text{and} \quad W_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} w(n)e^{-j2\pi nk/N}.$$

Due to the frequency mismatch ϵ, not only H_k and W_k will affect the received OFDM symbol. In each OFDM symbol, each subcarrier is phase shifted equally and this is known as the common phase error (CPE). This yields a loss of orthogonality between subcarriers and has a noise like impact called inter-carrier interference (ICI). The CPE of different OFDM symbols are often uncorrelated and therefore might have to be determined symbol by symbol, as mentioned in [7].

5.3.2 Symbol and Fractional Frequency Synchronization (CP-Synchronization block)

The first step in synchronization with an LTE base station is to find LTE symbol boundaries and the frequency of the base station. This is done in the CP-Synchronization block. It takes the I- and Q-samples received from the USRP as input and outputs fractional frequency corrected I- and Q-samples. When first trying to synchronize with a base station, there will be a carrier frequency offset (CFO) between the receiver and base station. The CFO ε, expresssed as multiples of the subcarrier spacing, can be split into two parts, a fractional part $0 \leq \varepsilon_F < 1$ and a integer part n_I.

$$\varepsilon = n_I + \varepsilon_F$$

The fractional CFO will introduce ICI (inter-carrier interference) and integer CFO will shift the OFDM symbols in the frequency domain.

If the $r(n)$ is the received baseband signal and $y(n)$ is the transmitted baseband signal, the CFO will create a frequency shift in $r(n)$.

$$r(n) = y(n)e^{j2\pi Tn\Delta f\varepsilon} = y(n)e^{j2\pi Tn\Delta f(\varepsilon_F+n_I)} = y(n)e^{j2\pi n(\varepsilon_F+n_I)/N}$$

since the sampling rate $\frac{1}{T} = \Delta f \cdot N$, where N is the number of samples per OFDM symbol. If there is an estimated fractional CFO $\hat{\varepsilon}_F$, then $y(n)$ can be estimated by

$$\hat{y}(n) = r(n)e^{-j2\pi n\hat{\varepsilon}_F/N} = y(n)e^{j2\pi n(n_I+\varepsilon_F-\hat{\varepsilon}_F)/N} \approx y(n)e^{j2\pi n(n_I)/N} \tag{1}$$

According to [6], the fractional CFO and the symbol timing can be estimated using a single algorithm which uses the cyclic prefix used in LTE OFDM. If the start of an OFDM symbol is denoted θ the log-likelihood function for symbols start (θ) and fraction CFO (ε_F) is

$$\Lambda(\theta, \varepsilon_F) = 2|\gamma(\theta)| \cos(2\pi\varepsilon_F + \angle\gamma(0)) - \rho\varepsilon(\theta),$$

where $\angle$ denotes the angle of a complex number,

$$\gamma(n) \equiv \sum_{k=n}^{n+L-1} r(k)r^*(k+N)$$

and

$$\varepsilon(n) \equiv \sum_{k=n}^{n+L-1} |r(k)|^2 + |r(k+N)|^2,$$

where L is the length of the cyclic prefix and N is the length of an OFDM symbol.

The maximum likelihood estimations, according to [6], is

$$\hat{\theta}_{ML} = \arg\max_{\theta}(2|\gamma(\theta)| - \rho\varepsilon(\theta))$$

$$\hat{\varepsilon}_{F,ML} = -\frac{1}{2\pi}\angle\gamma(\hat{\theta}_{ML}) \tag{2}$$

Where $\rho \equiv \frac{\sigma_s^2}{\sigma_s^2+\sigma_n^2}$. σ_s^2 and σ_n^2 is signal energy and noise energy respectively. Calculation of $\gamma(n)$ can be done by the following recursive algorithm

$$\gamma(n+1) = \gamma(n) + r(n+L)r^*(n+L+N) - r(n)r^*(n+N) \tag{3}$$

Calculation of $\varepsilon(n)$ can be done in the same way. This makes it a low complexity operation that can be done fairly quickly.

The constant ρ above is not known. It will be assumed that the signal energy σ_s^2 is much larger than the noise. This will make $\rho \approx 1$ and this will be assumed in our system.

The actual values of L and N will depend on the sampling rate that will be used. Since the USB interface between the USRP and PC will not support such high data rates that the entire cell bandwidth can be scanned, the sampling rate will effectivly decide the bandwidth of the signal that is used for synchronization (sampling theorem). Higher bandwidth will make the estimations described above better as long as its not larger than the total bandwidth of the base station. Since the bandwidth of the base station is not yet known a good choice for the sample rate will need to be found. The sample rate should be as high as possible but within the base station bandwidth, the number of samples per symbol should be a power of two, to make FFT calculations faster. The symbol length in LTE is set to $\frac{1}{\Delta f} = 66.7\mu$s, the length of the cyclic prefix is 4.69μs. If the number of samples per symbol is N, the sample time will be $T = \frac{66.7\cdot10^{-6}}{N}$. The number of samples in the cyclic prefix will be $L = \frac{4.69\cdot10^{-6}}{T}$ and the sample rate will be $\frac{1}{T}$.

The CP synchronization block will receive a stream of complex samples, sample number is n. The following algorithm can be used to extract OFDM symbols from this stream.

1. Calculate $\gamma(n)$ and $\varepsilon(n)$ with Eq 3 (using $\gamma(n) = r(n) = \varepsilon(n) = 0, n < 0$). $\gamma(n+1)$ and $\varepsilon(n)$ can be calculated when sample $n + N + L$ has been received and needs to start with $\gamma(-N - L + 1)$. The values of $\gamma(0)$ and $\varepsilon(0)$ will be the first that are correct.

2. Keep track of the position $\hat{n}$ of the highest value of $2|\gamma(n)| - \varepsilon(n)$

3. After $N + L$ samples have been received, mark the position of the best candidate for a symbol start $\hat{\theta}_{ML} = \hat{n}$.

4. Calculate $\hat{\varepsilon}_{F,ML}$ from $\hat{\theta}_{ML}$ with Eq 2.
5. Correct for fine frequency CFO (Eq 1) and extract the symbol $s = \{r(n)e^{-j2\pi Tn\hat{\varepsilon}_F} : \hat{\theta}_{ML} \leq n < (\hat{\theta}_{ML} + N + L)\}$
6. Output s to the next block.
7. Wait until $\gamma(\hat{\theta}_{ML} + \frac{N+L}{2})$ can be calculated, then reset $\hat{n}$.
8. Go to 2.

This algorithm can be improved by the fact that the difference between the start of two consecutive symbols is $N + L$. It also possible to estimate the performance of the synchronization by looking at the difference between estimated symbol starts. If it always is close to $N + L$, the algorithm is doing good.

5.3.3 FFT and CP-removal (FFT Block)

When symbol synchronization is done, the OFDM symbols can be extracted from the stream of samples. This is done in the FFT block. It takes the I- and Q-samples from the CP synchronization block as input and outputs OFDM symbols. The CP synchronization block will deliver blocks of samples that contain a symbol and its corresponding cyclic prefix, the length is $N + L$. N is the number of samples in one symbol and L is the number of samples in the cyclic prefix (see section 5.3.2). To extract the OFDM symbol an N-point FFT is made on the block of samples. This will result in N complex numbers that correspond to each subcarrier in the symbol.

Since the CP is included in the input sample block, this has to be removed. The FFT should thus start at sample L (0-indexed) in the received block.

5.3.4 PSS Detection (PSS-Synchronization Block)

After the OFDM symbols are put together, it is time to calculate the sector ID and perform from synchronization. This is done in the PSS-Synchronization block. It takes OFDM symbols from the FFT block as input and outputs frame synchronized OFDM symbols, along with the sector ID. In order to determine the sector ID, we perform a cross-correlation with three different Zadoff-Chu sequences. The sector identity is the one with the largest correlation. The known Zadoff-Chu sequences are

$$s_u^{ZC}(n) = \begin{cases} e^{-j\frac{\pi un(n+1)}{63}}, & n = 0, 1, ..., 30, \\ e^{-j\frac{\pi u(n+1)(n+2)}{63}}, & n = 31, 32, ..., 61. \end{cases}$$

for $u = 25, 29, 34$. These sequences are mapped on subcarrier symbols $d(k)$, where $k = -31, ..., -1, 1, ..31$ denotes subcarrier index with respect to DC. The correlation is given by

$$Q_i(n) = \sum_{\substack{k=-31 \\ k\neq 0}}^{31} d_i^*(k)R(n+k),$$

where $R(n)$ denotes the OFDM symbol and $d_i(k)$ the mapped Zadoff-Chu sequences s_u^{ZC} according to

$$i = 0 \leftrightarrow u = 25, i = 1 \leftrightarrow u = 29, i = 2 \leftrightarrow u = 34,$$

The i and n with the largest correlation corresponds to the sector identity N_s (0,1,2) and the integer CFO n_I. Since we do not know if the received symbol is a PSS symbol, the peak to average ratio of the correlation needs to be above a certain threshold. The threshold will be determined by trial and error.

5.3.5 SSS Detection (SSS-Synchronization Block)

With the help of the SSS (see Section 3.5.2), the group ID N_g and the subframe index within the radio frame (0 or 5) can be estimated. This is done in the SSS Synchronization block. It takes OFDM symbols from the PSS block as input, and outputs the same symbols, their subframe index, as well as the group ID. The basic concept is to exploit the cyclic shifts of the two length-31 binary sequences $s_0(n)$ and $s_1(n)$ according to the pair of integers m_0 and m_1, which identify the group-ID (see [2, Table 6.11.2.1-1]). The method used will follow these steps:

- Do a N-point DFT on the OFDM symbol containing the SSS.
- Separate the length-62 sequence $d(n)$ into sequences $d(2n)$ and $d(2n+1)$, consisting of even and odd subcarrier symbols.
- Divide $d(2n)/c_0(n)$ in order to obtain the sequence $s_0^{(m0)}(n)$. Sequence $c_0(n)$ is known since it only depends on the already estimated sector-ID N_s.
- Build a reference sequence $s_{ref}(n)$, which is a duplicated version of $s_0^{(m0=0)}(n)$ with the length of 62.
- Apply a cross-correlation between $s_0^{(m0)}(n)$ and the reference sequence $s_{ref}(n)$. The magnitude of the correlation term should show a significant maximum at $32 \leq i_{max} \leq 62$, which indicates $m_0 = 31 - i_{max}$.
- After estimating the integer m_0, we are able to compute $z_1^{(m0)}(n)$ and afterwards divide $d(2n+1)/(c_1(n)z_1^{(m0)}(n))$ in order to obtain the sequence $s_1^{(m1)}(n)$.
- Apply a cross-correlation between $s_1^{(m1)}(n)$ and $s_{ref}(n)$. The magnitude of the correlation term should again show a significant maximum at $32 \leq i_{max} \leq 62$, which indicates $m_1 = 31 - i_{max}$.
- The estimated m_0 and m_1 give the group-ID N_g according to [2, Table 6.11.2.1-1].
- Finally, the overall cell-ID $N_c = 3N_g + N_s$ is computed. This allows us the find the cell-specific reference signals.

Significant peaks will only be generated if the received sequence $d(n)$ is positioned correctly on the frequency grid. $d(n)$ should be positioned correctly, since the integer CFO is calculated and compensated for in the PSS block.

5.4 Channel Estimation (Channel Estimation Block)

Channel estimation is a necessary part of OFDM in order to retreive any information from the channel at all. The channel estimation is done in the Channel Estimation block. It takes synchronized OFDM symbols from the SSS-Synchronization block as input and outputs the same OFDM symbols, together with an estimation matrix. The size of the matrix depends on the number of Tx antennas.

In LTE, each cell uses a specific reference signal, that is, a set of pilot symbols (PS) that is placed in a predetermined way into the time-frequency resource grid, see Section 3.6. The values of the PSs are known in the sense that they are generated with a length-31 Gold Sequence based on the cell ID. How they are placed in the resource grid depends on the number of Tx antennas. Based on this, channel estimates $\hat{H}$ at the PS's places can be done in the frequency domain.

$$\hat{H}_{PS,LS} = \frac{Y_{PS}}{X_{PS}},$$

where $\hat{H}_{PS,LS}$ is the least square estimate of the channel at PS locations, Y_{PS} is the received PS and X_{PS} is the transmitted PS. In order to get a channel estimate for the entire resource grid, interpolation between the $\hat{H}_{PS}$'s has to be done. But before that, it is necessary to average these estimates to get rid of some noise. This can be done by applying an averaging window of appropriate size, depending on how much noise there is and how fast the channel is fading. Since the resource elements along the edge of the resource grid may lack the presence of a nearby PS location, interpolation cannot be done directly. To get around that problem, it is possible to introduce virtual pilot symbols to enable channel estimation for the whole resource grid. See Figure 17.

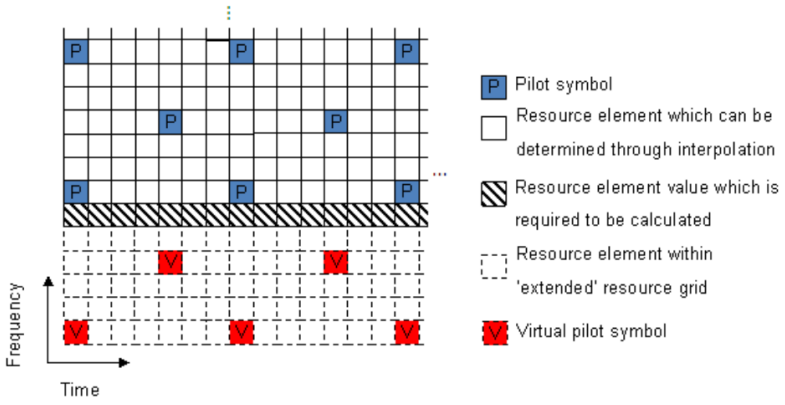


Figure 17: Placement of virtual PSs to enable interpolation for channel estimates, [8].

5.5 Channel Equalization (Channel Equalization Block)

After the channel has been estimated, the OFDM symbols should be equalized according to the estimation. This is done in the Channel Equalization block. It takes the synchronized OFDM symbols along with the estimation matrix from the Estimation block as input and outputs equalized OFDM symbols. Channel equalization has to be done on pairs or quadruplets of subcarrier symbols, depending on the number of antennas employed in the Tx diversity scheme of the eNodeB (see section 3.11). In the case of two Tx antennas, the received symbols r_k, r_{k+1} on pairs of adjacent subcarriers will be

$$\begin{aligned} r_k &= H_k^{(0)} x_k - H_k^{(1)} x_{k+1}^* + n_k \\ r_{k+1} &= H_{k+1}^{(0)} x_{k+1} + H_{k+1}^{(1)} x_k^* + n_{k+1} \end{aligned}$$

where $H_k^{(p)}$ is the estimated channel tap from antenna port p on subcarrier k, n_k is the noise and x_k is the symbol on subcarrier k before Tx diversity precoding.

A simple zero-forcing (ZF) equalizer, where the impact of the noise is ignored, will be used. The symbol estimates are

$$\hat{x}_k = \frac{1}{H_k^{(0)} H_{k+1}^{*(0)} + H_k^{(1)} H_{k+1}^{*(1)}} \left(H_{k+1}^{*(0)} r_k + H_k^{(1)} r_{k+1}^* \right)$$

$$\hat{x}_{k+1} = \left(\frac{1}{H_k^{(0)} H_{k+1}^{*(0)} + H_k^{(1)} H_{k+1}^{*(1)}} \right)^* \left(-H_{k+1}^{(1)} r_k^* + H_k^{*(0)} r_{k+1} \right)$$

where * denotes the complex conjugate as before. In the case of the 4 Tx antenna SFBC+FSTD scheme (as described in section 3.11), the extension of the equalizer is obvious. As is the case where only one Tx antenna is used, resulting in a trivial equalizer.

5.6 MIB Decoding (MIB-Decoder Block)

The MIB decoding block takes tagged OFDM symbols $\{s, d_i\}$ as input and passes them through to the output. If a PBCH is contained in the OFDM symbol, it is processed and the MIB is interpreted. The SFN is tagged on to the output while N_{RB} is updated as a global variable. An overview of the MIB Decoding block can be seen in Figure 18.

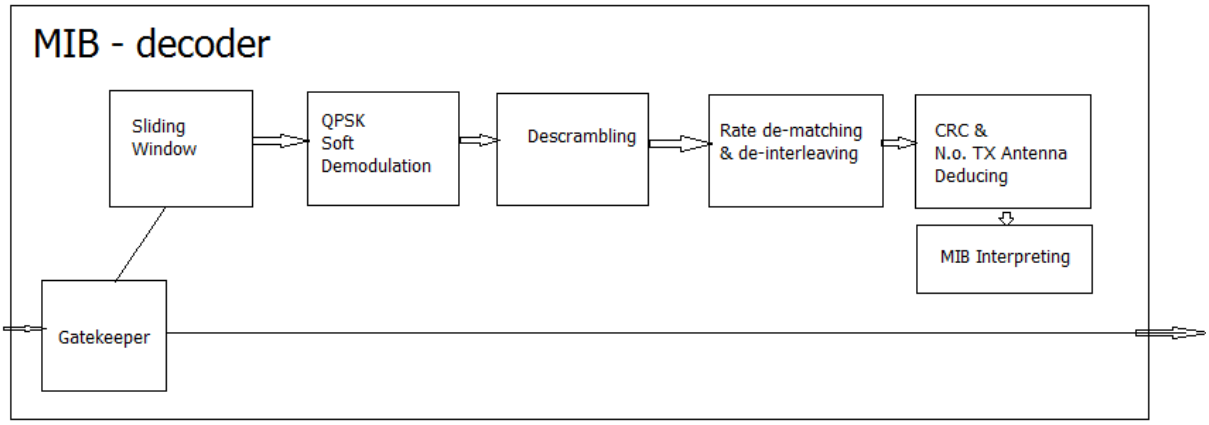


Figure 18: Processing in the MIB decoding block

The sub-blocks in the MIB decoder are not actual standalone GNU Radio blocks, rather they are implemented as C++ functions inside the MIB decoder block. This is in part to avoid performance drops due to unneccesarry buffering between GNU Radio blocks and in part to "keep it simple".

5.6.1 Gatekeeper

The Gatekeeper block takes as input a tagged OFDM symbol. If the tag indicates that part of a PBCH is contained in the OFDM symbol, it is extracted. Otherwise, the tagged OFDM symbol is just forwarded in the processing chain to the next block, skipping PBCH processing. When a complete PBCH has been intercepted (4 OFDM symbols), it is fed to the Sliding Window Block for PBCH processing.

5.6.2 Sliding Window Block

A 40ms sliding window, collecting the PBCH data from the 4 latest frames. Block input is a vector of 240 complex symbols containing the PBCH, block output is a vector of $4 \cdot 240 = 960$ symbols containg the PBCH of the last 4 frames.

5.6.3 QPSK Soft Demodulation

As mentioned earlier, the MIB is modulated using QPSK. That is, the pair of bits $b(i), b(i+1)$ are mapped to $\pm\frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}}$ where $b(i)$ is mapped to the real part ($1 \rightarrow -\frac{1}{\sqrt{2}}, 0 \rightarrow +\frac{1}{\sqrt{2}}$) and $b(i+1)$ to the imaginary part.

The input to this block is complex samples y. To enhance performance, soft demodulation shall be used where log-likelihood ratios (LLRs)

$$\hat{b}(i) = \log\frac{p(b_i = 0|y)}{p(b_i = 1|y)} = \log\frac{e^{-\frac{1}{2\sigma^2}\left(\Re\{y\}+\frac{1}{\sqrt{2}}\right)^2}}{e^{-\frac{1}{2\sigma^2}\left(\Re\{y\}-\frac{1}{\sqrt{2}}\right)^2}} = \frac{\sqrt{2}}{\sigma^2}\Re\{y\}$$

where the noise is assumed to be zero-mean Gaussian with variance σ^2 per dimension. The noise power is assumed constant during a transport block, so the soft bit outputs can be normalized as

$$\begin{aligned}\hat{b}(i) =& \Re\{y\}\\ \hat{b}(i+1) =& \Im\{y\}\end{aligned}$$

5.6.4 Descrambling

The soft bit inputs are descrambled (their sign flipped accordingly, corresponding to a bitwise XOR) by the scrambling sequence $c(i)$ [2, Section 7.2]. The sequence is initialized with $cinit = N_c$, the cell ID, every fourth radio frame.

5.6.5 Rate Dematching and De-Interleaving

The 1920 descrambled soft bits are segmented into sixteen 120-bit segments, which are added to each other to form a stronger bit estimate, inverting the repetition coding. The 120-bit sequence is split into three 40-bit segments, each corresponding to an output from the rate 1/3 convolutional encoder. On each 40-bit segment, sub-block de-interleaving (according to the pattern defined in [5, Section 5.1.4.2.1]) is performed.

5.6.6 Convolutional Decoder (Viterbi)

The output from the rate-dematching is put into a Viterbi decoder. The code used is a tailbiting convolutional code with constraint length $K = 7$. A model of the convolutional code can be seen in 19.

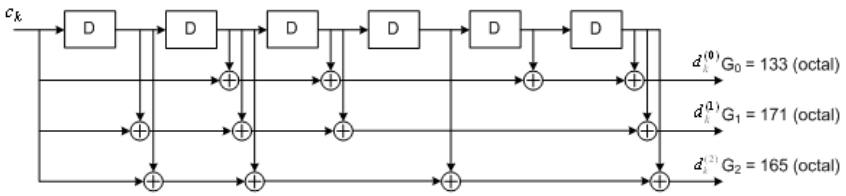


Figure 19: The convolutional code used.

The initial value of the shift register of the encoder is set to the values corresponding to the last 6 information bits in the input stream so that the initial and final states of the shift register are the same. The branch metric of the Viterbi decoder is the Euclidean distance between triplets of soft bits. The Viterbi decoder outputs an estimate of the 40 bit payload.

5.6.7 CRC Calculation and Number of Antennas Deducing

A CRC is calculated from the 24 first bits of the output of the Viterbi decoder. The CRC is scrambled according to the number of Tx antennas believed to be in use and compared to the last 16 bits from the output of the Viterbi decoder. If the checksum is correct, the number of Tx antennas is assumed to be deduced and the 24 non-parity bits is sent to the next block for interpreting.

Initially, the number of Tx antennas is unknown. Each antenna configuration (1,2 or 4) shall be tried until either a successful decoding or 32 failed decodings have occurred. In the latter case, the next antenna configuration shall be tried and the Channel Equalization and Channel Estimation blocks shall be notified.

5.6.8 MIB Interpreting

Of the 24 non-parity bits, only 14 are actually used. These are interpreted and

1. PHICH duration, Normal or Extended
2. PHICH resource (six, half, one, two)
3. Downlink system bandwidth N_{RB}, expressed in multiples of resource blocks
4. Most significant bits of the System Frame Number (SFN) are aquired.

5.7 PCFICH Decoding (PCFICH-Decoder Block)

The PCFICH Decoding block works in a similar fashion as the MIB Decoder block. It takes a tagged OFDM symbol as input and passes it through to the output, adding a tag about the aquired CFI value of the subframe.

5.7.1 Gatekeeper Block

The Gatekeeper block extracts the 16 complex valued modulation symbols on the PCFICH, if the tag indicates that the OFDM symbol is in the first slot in a subframe.

5.7.2 QPSK Soft Demodulation

The symbols are demodulated, as described in section 5.6.3.

5.7.3 Descrambling

The soft bits are descrambled, with the sequence described in section 3.8.

5.7.4 CFI Decoding

The CFI decided to be $k \in \{0, 1, 2\}$according to

$$\arg\min_k \sum_{i=0}^{31} (b(i) - \frac{1}{\sqrt{2}}(1 - 2 \cdot d_k(i))^2$$

where d_k is the codeword corresponding to CFI$= k$ and b the descrambled bit sequence. I.e. the decision metric is the Euclidian distance.

5.8 PDCCH Decoding (PDCCH-Decoder Block)

The PDCCH Decoder block is similar in structure to the MIB and PCFICH Decoder blocks. It takes as input an OFDM symbol, tagged with time index and CFI value of the subframe. It outputs the same OFDM symbol with an additional message if the SIB1 DCI was found.

5.8.1 Gatekeeper

The Gatekeeper block takes as input a tagged OFDM symbol. If the tag indicates that part of a PDCCH is contained in the OFDM symbol, it is extracted. Elsewhise, the tagged OFDM symbol is just forwarded in the processing chain to the next block, skipping PDCCH processing. When a complete control region has been intercepted, it is fed to the De-Interleaving Block.

5.8.2 De-Interleaving

The permutations and cyclic shifts of the modulation symbols are inverted (The symbols are modulated according to [2] chapter 6.8.5).

5.8.3 QPSK Soft Demodulation

The symbols are demodulated, as described in section 5.6.3.

5.8.4 Descrambling

The soft bits are descrambled, with the sequence described in 3.9.

5.8.5 Rate De-Matching

The descrambled soft bits are rate-dematched and de-subinterleaved with the same algorithm as in the MIB Decoder block. Except the rate is different.

5.8.6 Convolutional Decoder (Viterbi)

The rate-dematched soft bits are put into the same Viterbi decoder as in the MIB Decoder block.

5.8.7 DCI Searching

The DCI Searching Block looks for DCI messages scrambled with the SI-RNTI at each place. It can be located in the common search space. If the SIB1 DCI is found, it is tagged onto the OFDM stream to the SIB Decoding block.

5.9 SIB Decoding (SIB-Decoder Block)

The SIB Decoding block works similar to the previously described blocks. It takes a tagged OFDM symbol as input. If a DCI message is tagged on the input, it parses it and prepares to decode the SIB1. The SIB decoder serves as a sink to the OFDM symbols. It only outputs the decoded SIB1 information, including operator name, if found.

5.9.1 DCI Parser and Symbol Extraction

The DCI message is parsed to know the location of the SIB1. The resource elements containing the SIB1 are extracted and fed to the next block.

5.9.2 Constellation Demodulation

The constellation type (QPSK,16-QAM,64-QAM) of the SIB1 transmission is aquired from the DCI. Corresponding soft demodulation is done.

5.9.3 Descrambling

Bit level descrambling is done.

5.9.4 Rate De-Matching

Rate de-matching according to the transport block size aquired from the DCI is performed according to [5, Section 5.3.2].

5.9.5 Turbo decoding

Turbo decoding according to [5, Section 5.3.2.3].

5.9.6 CRC check

A CRC checksum is calculated to see if the SIB1 was decoded properly.

5.9.7 SIB1 Interpreting

The SIB1 message is parsed.

6 Sub System III - User Interface

Sub System III consists of the User Interface, or UI. The UI is what will be used by the common user to control the program. It will also display data received from Sub System II.

6.1 Interface

The section describes the interface of the system.

6.1.1 Input

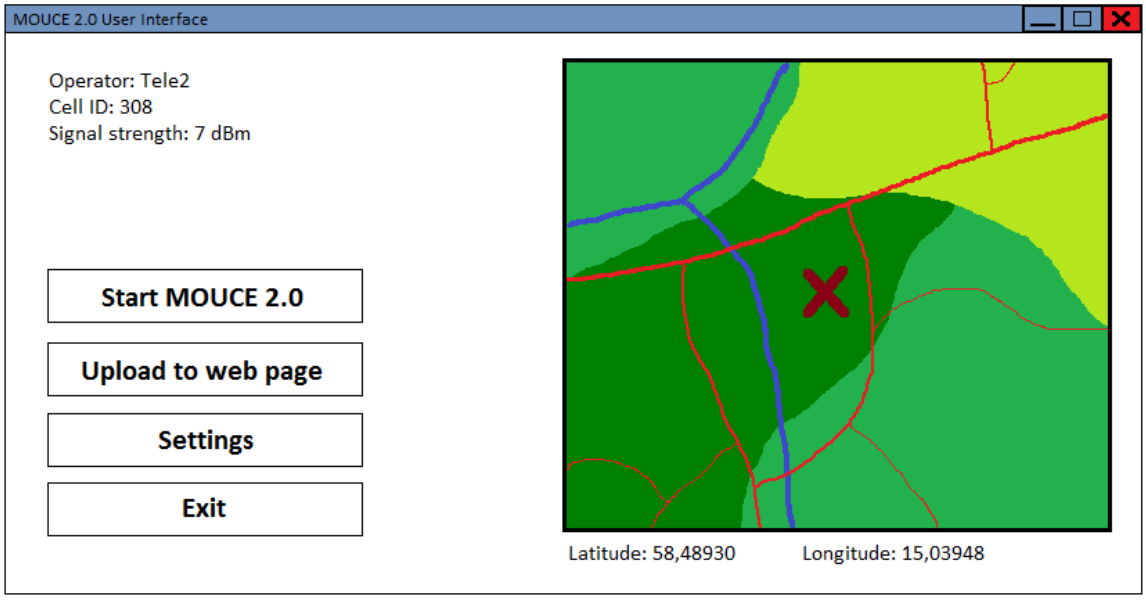
Data from Sub System II. This data consists of cell ID, operator, signal strength and bandwidth utilization.

6.1.2 Output

Data to web page. This data includes the information received from Sub System II and a time stamp.

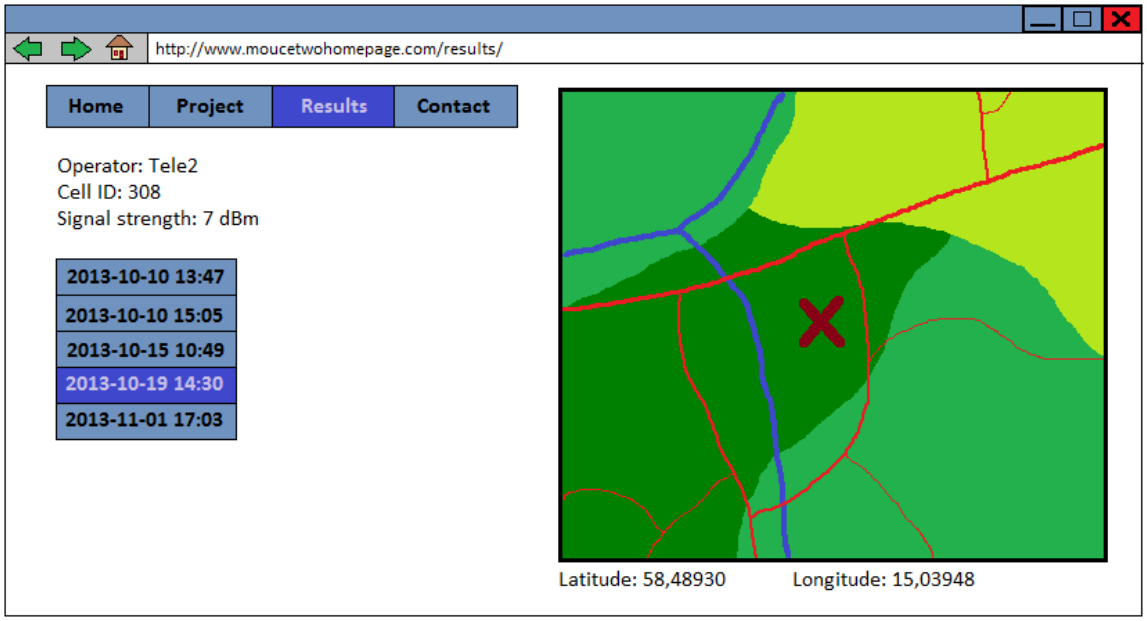
6.2 Layout

The UI will have a simple design. The user shall be able to change basic settings in the program (like what frequency to scan), start the MOUCE software with the click of a button, and see information about the received signal. There will also be an option to upload the retrieved information to a web page. A map might also be incorporated, where the used base station will be marked. Below is a sketch of the UI:



6.3 Web Page

A web page will be created, where a description of the project, a presentation of the project members and some results from the finished product will be displayed. The web page will be quite simple, and will mainly contain basic text information and some pictures. A function for uploading data from the program to the web page will be created. Below is a sketch of the web page:



7 Implementation strategy

7.1 Sub System I

Sub System I needs very little implementation, it will consist of setting up hardware.

7.2 Sub System II

Developing Sub System II will be the hardest part of the project and will require a lot of time from the group members. To make development efficient the blocks from GNU Radio will be used when possible. When custom blocks are developed they should be well commented and easily testable. It will be important to identify problems early and to be able to find them quickly.

There is a need for a good testing framework. Every block should have a unit test that is testing if the block is working correctly. These tests should be good enough to test vital functionality of the block and find blocks that are not working correctly. Preferably these test should include both simulated test data and real data from blocks earlier in the chain.

A developer UI of some kind should also be developed for this system. This UI should be able to display performance data about different blocks in the running system. This data might be information about what the block is doing and how well it is doing it. For example the CP synchronization block can measure variance in the difference of symbol starts. If this variance is low the block is probably doing well. This kind of data is useful for pinpointing where in the processing chain there are problems.

Sub System II will be implemented roughly in the order of the processing chain. The implementation will start with the CP synchronization block then the FFT block and so on. This will make it possible to test the different blocks together early on.

7.3 Sub System III

Sub System III can be developed in parallel with Sub System II. This Sub System should also have unit tests that are used to verify that it is functional.

References

[1] I. Wireless, "*Frekvensband för GSM, 3G och 4G (LTE) (Online)*." `http://www.induowireless.com/nu/gsm-3g-4g-frekvensband/`. Accessed: 2013-10-11.

[2] 3GPP, " *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 11)*," Technical Specification TS 36.211, 2012.

[3] E. Dahlman, S. Parkvall, and J. Sk´old, *4G: LTE/LTE-Advanced for Mobile Broadband: LTE/LTE-Advanced for Mobile Broadband*. Elsevier Science, 2011.

[4] P. Wad, "*LTE Resource Grid (Online)*." `http://paul.wad.homepage.dk/LTE/lte_resource_grid.html`. Accessed: 2013-10-08.

[5] 3GPP, " *Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 11)*," Technical Specification TS 36.212, 2012.

[6] K. M. et. al, "*A Closed Concept for Synchronization and Cell Search in 3GPP LTE Systems (Online)*." `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04917491`. Accessed: 2013-10-08.

[7] M. S. et. al, "*Optimal Receiver Design for Wireless Broad-Band Systems Using OFDM–Part I (Online)*." `http://gps-tsc.upc.es/comm/eco/TechnicalDocs/Papers/Synchronization_SISO/Sincro_OFDM_Speth1.pdf`. Accessed: 2013-10-08.

[8] S. Ascent, "*Channel Estimation (Online)*." `http://www.steepestascent.com/content/mediaassets/html/LTE/Help/Channel%20Estimation.html`. Accessed: 2013-10-08.