# INVESTIGATE REDUNDANCY IN SOUNDING REFERENCE SIGNAL BASED CHANNEL ESTIMATES


*Submitted by*

Akshay Thakaramkunnath Ajayakumar

(ak7087th-s@student.lu.se)

Nishanth Bhavani Sankar

(ni7072bh-s@student.lu.se)

Department of Electrical and Information Technology
LUND UNIVERSITY

***Supervisors:*** *Jesús Rodríguez Sánchez (EIT)*
*Liang Liu (EIT)*
*Oleksiy Kuzmenok (Ericsson)*
***Examiner****: Fredrik Rusek (EIT)*


*August 2019*
*LUND-SWEDEN*

## *ABSTRACT*

5G supports enormous increase in data rate. Massive antenna beamforming is expected to play a key role in increasing capacity in case of multi-user MIMO and coverage in case of single-user MIMO. The large number of antennas in massive MIMO system will lead to enormous amount of channel state information being stored in the memory and this necessitates the use of compression techniques for efficient utilization of memory, which is limited.

Sounding Reference Signals (SRS) are transmitted in the uplink to obtain channel estimate. In TDD based systems, by exploiting channel reciprocity channel estimates received in the uplink can be used in downlink as well. The product, we work on at Ericsson, is a TDD based system and uses SRS based channel estimates to compute beamforming weights to facilitate massive antenna beamforming.

SRS based channel state information is represented by 32-bit complex number in this system, which is received per Evolved Node B (eNodeB) antenna, per User Equipment (UE) transmission antenna, and per Physical Resource Block Group (PRBG). This results in a significant amount of data that needs to be stored in the eNodeB. However, memory in the Digital Unit of eNodeB is limited. SRS based estimates occupy a major portion of this memory and therefore limit the capacity of the eNodeB for beamforming.

This thesis focuses on the evaluation and implementation of lossless and lossy compression of SRS based channel estimates to attain space savings in the shared memory of eNodeB. This will help in achieving higher capacity for reciprocity-based beamforming and prolong the lifetime of existing hardware.

Performance of various lossless data compression algorithms was analyzed based on compression ratio, speed and complexity and the optimal one was selected.

Lossy compression of SRS based channel estimates was also implemented for LOS UEs using linear regression by least squares estimate. Impact on performance due to application of lossy compression algorithm was studied.

## *Acknowledgements*

We would like to thank our supervisor Mr. Oleksiy Kuzmenok from Ericsson AB in Lund, Sweden for giving us the opportunity to do our Thesis at Ericsson and for his invaluable supervision, support and feedback throughout the duration of the Thesis.

Furthermore, we would like to express our gratitude to our supervisors at Lund University, Mr. Liang Liu and Mr. Jesus Rodriguez Sanchez, for their continual support and guidance in developing ideas for our Thesis approach and to our examiner Mr. Fredrik Rusek for his feedback.

We will forever be appreciative of the assistance and motivation our families and friends have given us during our Master's degree at Lund University. We could not have done it without them, and it was an exciting journey all through.

## *Popular science summary*

In order to reliably communicate over the air, the receiver needs to estimate the quality of the wireless link. This is done by the transmission of certain signals named ‘pilots’.

In cellular communications, such as 5G, pilots are sent in both directions, which is from base station to the user and vice versa. To support 5G systems, numerous antennas will be used at transmitter, receiver or both. Such systems are called as massive multiple input multiple output (Massive MIMO) systems. Pilots need to be transmitted for each of these antennas and this will lead to significant amount of data. For efficient and reliable systems, it is important to ensure that the least amount of such information is stored in the base station, which necessitates the use of data compression techniques.

The link quality values obtained can be compressed by lossy methods which involve loss of some information but higher compression and by lossless methods that have no loss of information but lower compression.

Aim of this thesis is to study about a certain type of pilot, namely Sounding Reference Signal, and compressing the obtained link quality values. Hence, this thesis focuses on analyzing the performance of various compression techniques based on their ability to compress this data, speed of the program and complexity of implementation and selecting the optimal technique based on the analysis.

Implementation of compression of these link quality values will help to prolong the lifetime of the equipment used now and can help in saving costs for Telecom companies.

## *Index*

## *List of Acronyms*

| | |
|---|---|
| 3GPP | Third-generation partnership project |
| ACM | Adaptive coding and modulation |
| ANSI | American National Standards Institute |
| BER | Bit error rate |
| BS | Base station |
| CP | Cyclic prefix |
| CPU | Central Processing Unit |
| CRC | Cyclic redundancy check |
| CRS | Cell-specific reference signal |
| CSI | Channel-state information |
| DFT | Discrete Fourier transform |
| DL | Downlink |
| DSP | Digital Signal Processor |
| eNodeB/eNB | Evolved UTRAN NodeB |
| EPDCCH | Enhanced physical downlink control channel |
| FDD | Frequency Division Duplex |
| FFT | Fast Fourier Transform |
| I/O | Input-Output |
| IFFT | Inverse fast Fourier transform |
| IMT | International Mobile Telecommunications |
| ITU | International Telecommunications Union |
| ITU-R | International Telecommunications Union-Radio Communications sector |
| LOS | Line-of-sight |
| LTE | Long-term evolution |
| LZO | Lempel–Ziv–Oberhumer |
| MBMS | Multimedia broadcast-multicast service |
| Mbps | Megabits per second |
| MBSFN | Multicastebroadcast single-frequency network |
| MCH | Multicast channel |
| MIMO | Multiple input-multiple output |
| mmWaves | millimeter waves |
| ms | millisecond |
| Mu-MIMO | Multi-user MIMO |
| NLOS | Non-line-of-sight |

| | |
|---|---|
| OFDM | Orthogonal frequency-division multiplexing |
| PDSCH | Physical downlink shared channel |
| PMCH | Physical multicast channel |
| PRB | Physical resource block |
| PRBG | Physical resource block group |
| PUCCH | Physical uplink control channel |
| PUSCH | Physical uplink shared channel |
| QAM | Quadrature amplitude modulation |
| QPSK | Quadrature phase-shift keying |
| RB | Resource block |
| RE | Resource element |
| RF | Radio frequency |
| RRC | Radio-resource control |
| RX | Receiver |
| SC-FDMA | Single carrier-frequency division multiple access |
| SISO | Single input-single output |
| SNR | Signal-to-noise ratio |
| SRS | Sounding reference signal |
| Su-MIMO | Single-user MIMO |
| TDD | Time division duplexing |
| TX | Transmitter |
| UE | User equipment |
| UL | Uplink |
| UpPTS | Uplink part of the special subframe, for TDD operation |
| VLC | Variable length coding |
| ZF | Zero forcing |

# CHAPTER 1

## Introduction

Mobile communication has evolved from being an expensive technology with its availability limited to a few individuals, to an everyday commodity used by majority of the world's population in the last decades. There have been four generations of mobile-communication systems and the fifth generation is on its way to a full-fledged launch [2]. Each generation was developed using a specific set of technologies and served a specific set of use cases.

As described in [3], first generation systems were introduced in the 1980's and supported analog voice-only mobile communication. Examples of 1G technologies include Nordic Mobile Telephony (NMT), Advanced Mobile Phone System (AMPS), and Total Access Communication System (TACS). Second generation systems were introduced in the 1990's and were voice centric. However, 2G being digital, offered higher capacity than 1G systems. 2G technologies include Global System for Mobile communications (GSM), Interim Standard 95 (IS-95) / Code Division Multiple Access (CDMA), Interim Standard 136 (IS-136) / Time Division Multiple Access (TDMA) and Personal Digital Cellular (PDC). Some enhancements to 2G technologies enabled packet data services and is referred to as 2.5G [2]. An example is GSM/Enhanced Data rates for GSM Evolution (EDGE) which is still in use.

As per [14], ITU introduced IMT-2000 in the mid-1980s. IMT-2000 was developed with the aim of providing value-added services and applications based on a single standard. One key objective was to provide seamless global roaming that would enable the user to use same number and handset when moving across borders. It was also aimed at providing better data and multimedia services. Third-Generation Partnership Project (3GPP) was formed to make this possible with the development of Wideband Code Division Multiple Access (WCDMA) and Time Division-Synchronous Code Division Multiple Access (TD-SCDMA) technologies [2]. WCDMA supported circuit switched voice and video services, and data services over packet-switched domain [6]. In the mid-2000s, High Speed Packet Access (HSPA), a major enhancement to 3G technologies, enabled mobile-

broadband experience with data rates reaching several Megabits per second (Mbps) which laid the foundation for rapid growth in the use of smart phones [2]. The increase in the use of packet-data-based services such as social networking and video gaming necessitated increased capacity and improved spectral efficiency. The more the users that needed packet data services, the more there was need for higher data rates and lower latency [2]. All these contributed to the development of 4G technology.

LTE used packet switched networks to transport calls by using techniques like voice over IP (VoIP) [6]. In 2008, ITU introduced IMT-Advanced which published a set of requirements for the fourth-generation communication systems [6]. A new release of LTE (release 10), named LTE-Advanced was developed by 3GPP as a candidate technology for IMT-Advanced [2]. It had stringent requirements like high data rates, high capacity, low latency, spectrum flexibility, and commonality between Frequency-Division Duplex (FDD) and Time-Division Duplex (TDD) [2]. LTE gained worldwide acceptance as LTE-Advanced and further releases introduced improvements like multisite coordination, use of fragmented and unlicensed spectrum, densified deployments which led to increase in capacity of Evolved Node B (nomenclature for base transceiver station in LTE). LTE also offered support to additional applications such as massive machine-type communication and device-to-device (D2D) communication. Figure 1.1 illustrates the capabilities of IMT-2000 and IMT-Advanced.

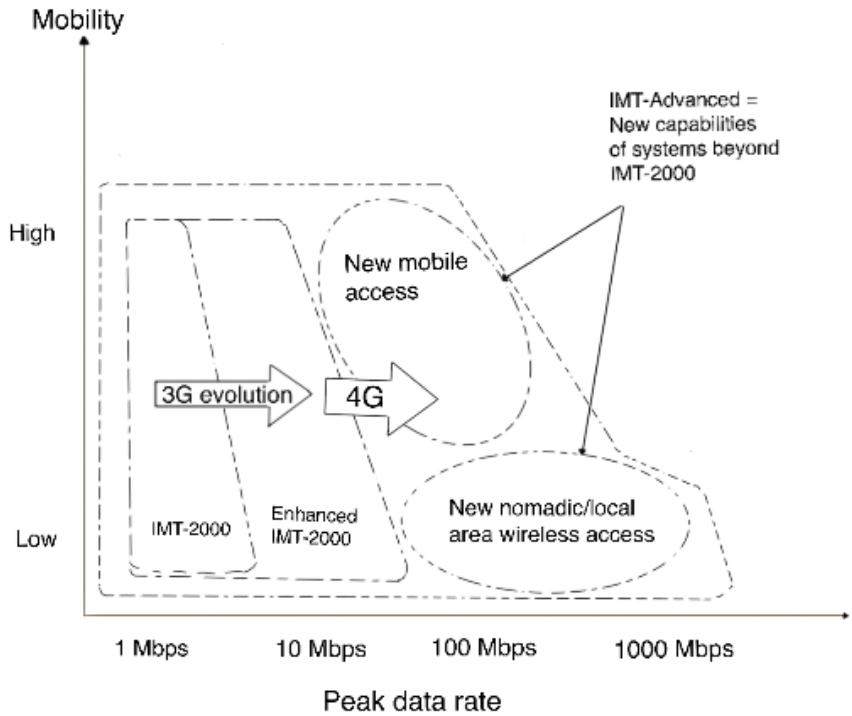


**Figure 1.1: Illustration of capabilities of IMT-2000 and IMT-Advanced**

With the introduction of internet of things (IoT), many devices are now able to send and receive data and communicate with each other through wireless networks such as the internet, leading to higher capacity requirements.

In 2012, work on the next generation of IMT systems, named IMT-2020, began and lead to the development of the fifth generation of mobile telecommunications system, namely 5G. The usage scenarios identified for 5G systems include Enhanced Mobile Broadband (eMBB), Massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low Latency Communications (URLLC) [5]. Massive Machine-Type Communication (mMTC) applications include sensor networks, traffic monitoring, and numerous other Internet of Things (IoT) applications. URLLC applications require high reliability and low latency such as vehicle-to-vehicle (V2V) communications, gaming, unmanned aerial vehicle (UAV) and robotics [2]. These use cases necessitate a massive increase in capacity.

This increase in capacity is aimed to be achieved in two ways: increase in spectrum bandwidth and increase in spectrum efficiency. Due to the lack of available spectrum in frequency bands less than 6 GHz, the increase in spectrum bandwidth involves exploiting higher frequency bands in the range of 6-30 GHz and above 30 GHz, for which LTE is not designed. Electromagnetic waves with frequency in the range of 30 GHz to 300 GHz are called as millimeter waves or mmWaves. Hence, 5G involves exploiting mmWaves. In lower frequencies, where there is a lack of available spectrum, spectrum efficiency can be improved by employing Massive MIMO systems.

LTE supports multiple antennas at both eNodeB and UE to facilitate diversity processing and spatial multiplexing [6]. Diversity processing helps in maintaining a constant received signal power by employing multiple antennas at TX, RX or both and thereby helps to mitigate fading. Spatial multiplexing involves the multiplexing of multiple receivers over the same time-frequency resources. This enhances the spectral efficiency and therefore sum rate.

Beamforming is the process of combining elements in an antenna array such that signals reach the desired UE in phase and interfere constructively, to produce high received signal power. At the other UEs, the signals reach out of phase and interfere destructively, so that received power is low [6]. When mmWaves are used in 5G, coverage will be a limiting factor, as path loss increases with frequency [2]. Beamforming can be used to increase capacity in case of multi user-MIMO (Mu-MIMO) and to enhance coverage in case of single user-MIMO (Su-MIMO) as shown in figure 1.2.

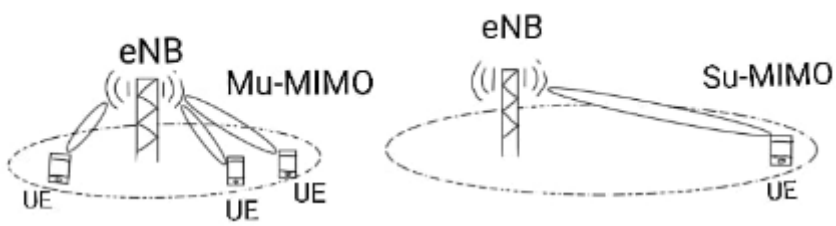


**Figure 1.2: Use of beamforming to enhance capacity in Mu-MIMO and increase coverage in Su-MIMO**

SRS based channel estimates can be used to facilitate downlink beamforming [1]. The product we worked on at Ericsson uses SRS based channel estimates to facilitate reciprocity-based beamforming (explained in Chapter 2).

## *1.1 Background and Motivation*

Each User Equipment (UE) antenna transmits SRS which is received by the eNodeB antennas. SRS is represented by a 32-bit complex number in this TDD-based system and is transmitted in the uplink to obtain channel estimate. 20 MHz bandwidth in LTE has 100 Physical Resource Blocks (PRBs). In case of channel estimates in this system, two adjacent PRBs are averaged into one Physical Resource Block Group (PRBG), resulting in 50 PRBGs. In this thesis, we examine two UEs which are in LOS conditions and one UE which is in NLOS conditions. The eNodeB in this system has 64 antennas.

$$\begin{aligned} &Channel\ State\ Information\ size\ for\ 1\ UE \\ &\quad = 64(antennas\ at\ eNB) \times 2\ (antennas\ at\ UE) \\ &\quad \times 50(Number\ of\ PRBGs) \\ &\quad \times 32\ bits\ (Size\ of\ an\ SRS\ based\ chanel\ estimate) \\ &\quad = 204800\ bits = 25600\ bytes \end{aligned}$$

The storage capacity of shared memory is 12 MB. CSI from 128 UEs is stored in the shared memory, which amounts to a size of 3.2 MB. Hence, nearly 27% of the available memory is occupied by CSI and becomes a major limiting factor for reciprocity-based beamforming. Hence, the main motivation behind this thesis is to compress SRS based channel estimates to attain space savings in the shared memory of eNodeB. The space savings achieved could be used to accommodate CSI from more UEs or for other processes at the eNodeB. This helps to prolong the lifetime of existing hardware.

Figures 1.3 illustrates the arrangement of antennas in the array at eNodeB and the transmission of SRS from UE antennas to antennas at eNodeB. Each antenna in the eNodeB receives SRS based channel estimates corresponding to 50 PRBGs and has been illustrated in Figure 1.4.

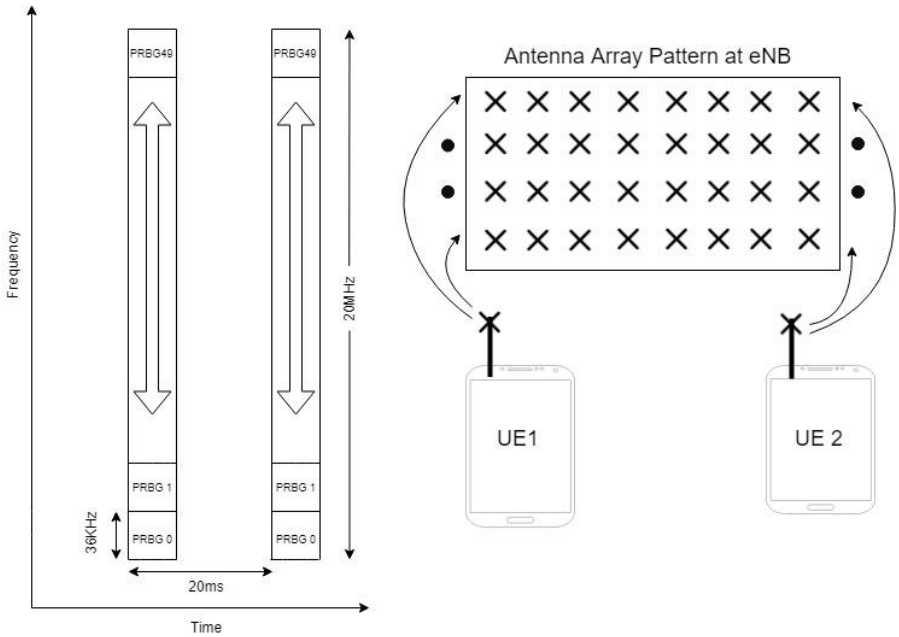


**Figure 1.3 Scenario Diagram representing relation between antennas at eNB and antennas at UE.**

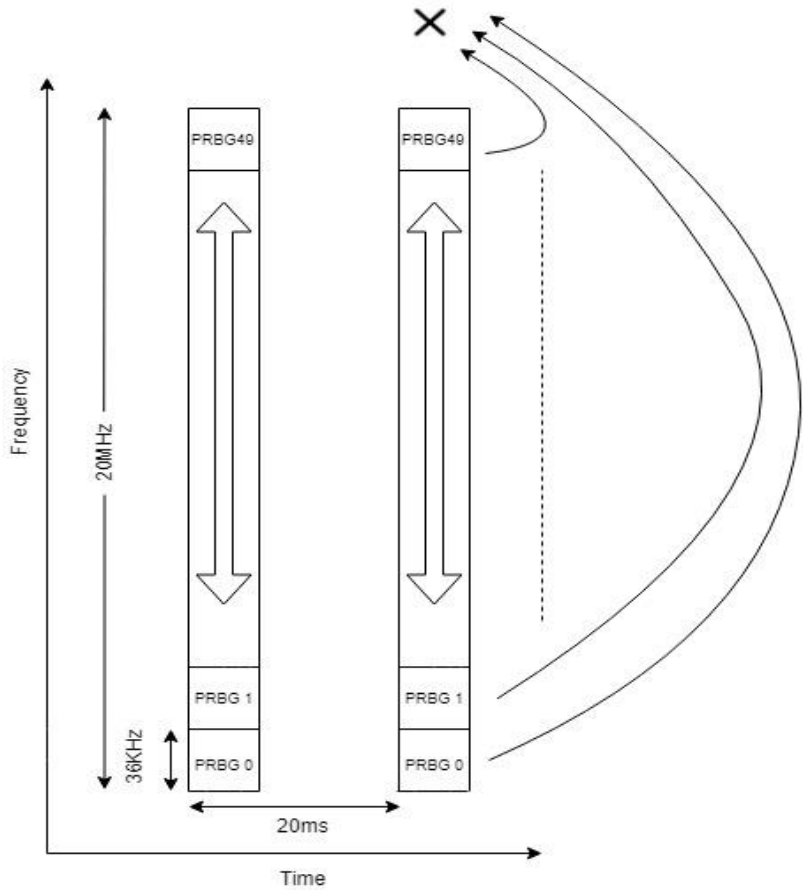


**Figure 2.4 Scenario Diagram – Each antenna receives SRS values corresponding to 50 PRBGs.**

## *1.2 Objectives of this thesis*

The objectives of this Thesis include:

1. To study SRS based channel estimates collected from eNodeB deployed in the field to analyze the following:

    a) to investigate redundancy in channel estimates values corresponding to various PRBGs (along Frequency Axis) in a given SRS occasion.

b) to investigate redundancy in channel estimates values corresponding to different elements of the antenna array.

2. To create MATLAB scripts to analyze various lossless compression algorithms based on compression ratio, speed, and complexity and select optimal compression algorithm. After selection of the algorithm, port its source code to Ericsson's test environment for further analysis. This will be performed for both line of sight (LOS) and non-line of sight (NLOS) scenarios.

3. Implement lossy compression of SRS based estimates and evaluate the impact on performance.

## *1.3 Approach and Methodology*

This thesis was performed in four phases:

**Phase 1:**

Initial phase involved analysis of redundancy in SRS based channel estimates, that were collected from eNodeB deployed in the field, using MATLAB scripts. This phase also encompasses literature review to study various compression algorithms that can be used and then selecting the suitable ones.

**Phase 2:**

Second phase included creation of MATLAB and shell scripts to analyze the performance of selected lossless compression algorithms on the collected SRS based channel estimates and selection of optimal algorithm.

**Phase 3:**

Third phase involved the porting of optimal algorithm's source code to Ericsson's test environment and evaluate the performance of the algorithm in the test environment.

**Phase 4:**

Implemented lossy compression of SRS based channel estimates of UEs in LOS using linear regression by least squares estimation and evaluated the performance of the algorithm.

## *1.4 Previous Work*

1. One of the previous findings that can be used in this thesis is the information on number of adjacent PRBs that can be averaged for beamforming.

2. Previous studies also show that the SRS based channel estimate values corresponding to a PRBG are similar to that of another group within the same 20 MHz band in a given SRS occasion for a UE in LOS.

3. IEEE paper that proposed compression method, based on Principal Component Analysis (PCA), for CSI feedback overhead in large scale MIMO systems with negligible performance degradation.

"Channel Correlation Modeling and its Application to Massive MIMO Channel Feedback Reduction" Jingon Joung, Ernest Kurniawan and Sumei Sun.

4. An IEEE paper was published by the Department of Electrical and Information Technology, Lund University, which implemented an on-chip memory system equipped with CSI, which provides an area-efficient memory system for massive MIMO baseband processing with lossy CSI compression.

The citation for the paper is as follows:
Yangxurui Liu, Liang Liu, Ove Edfors, and Viktor Öwall, "An Area-Efficient On-Chip Memory System for Massive MIMO Using Channel Data Compression," IEEE Transactions on Circuits and Systems I: Regular Papers (Volume: 66, Issue: 1, Jan. 2019).

# CHAPTER 2

## LTE, SRS and Multiple Antenna Systems

The idea of Long Term Evolution (LTE) came into existence with sole objective to provide a new radio access technology focused on packet-switched data. 3GPP works extensively on LTE standardization to decide on performance and capability for LTE, which comprises of spectral efficiency, peak data rates, latency, throughput, multi-channel bandwidths (1.25MHz-20MHz), and compatibility with previous 3GPP radio-access technologies such as Wideband Code Division Multiple Access (WCDMA) and Global System for Mobile (GSM). 3GPP also studies on the practicability of various technical solutions chosen by developing comprehensive specifications. In LTE, OFDM and SC-FDMA are the transmission schemes in downlink and uplink transmissions respectively and have been discussed in subsequent sections. This chapter also discusses about Reference signals in LTE and SRS. Multiple antenna systems, which is a key feature in LTE and NR will also discussed in this chapter.

### *2.1 Orthogonal Frequency Division Multiplexing (OFDM)*

OFDM, which uses multiple subcarriers in a channel is a form of Frequency Division Multiplexing (FDM). Serial bits of data are parallelized and mapped to complex symbols and multiplied with sinusoids that are orthogonal for transmission which is similar to Inverse Fast Fourier Transform (IFFT). At receiver, parallel bits of data received are converted back to serial stream of symbols, which are converted to bits by the symbol de-mapper for Fast Fourier Transform (FFT) operation.

OFDM signal is group of closely packed FDM subcarriers and subcarriers in general are orthogonal. Every transmit subcarrier that uses modulation schemes such as QPSK, 16QAM, 64QAM to transmit data is a sinc function spectrum in frequency domain where side lobes result in spectra overlapping between subcarriers. This in turn, leads to subcarrier interference except at orthogonally spaced frequency position. Peaks of subcarriers align together

with nulls of other subcarriers at orthogonal frequencies. Orthogonality prevents subcarrier interference in OFDM system.

OFDM's tolerance to delay spread caused due to multipath is the most vital advantage to use in LTE because in time domain, OFDM symbol duration is quite long and this can be achieved by parallel transmission of data. This leads to uncomplicated equalization at receiver. Extended symbol duration and guard interval helps mitigate Inter Symbol Interference (ISI) as well. Orthogonal subcarrier results in increased spectral efficiency as orthogonal subcarriers permit a greater number of subcarriers in a given bandwidth and dynamic usage of frequency spectrum

Orthogonal Frequency Division Multiple Access (OFDMA) signal depends on group of orthogonal subcarriers which transmits data in parallel and each subset of subcarrier is reserved for each user. Sharing of resources among a set of users is facilitated by this multiple access technique where non-faded subcarrier is chosen for transmission to each user in a cell in order to achieve optimal data rate.

Peak to Average Power Ratio (PAPR) is defined as

$$PAPR = (P_{peak})/(P_{avg}) \quad (2.1)$$

where $P_{peak}$ is the transmit symbol power at peak amplitude value, $P_{avg}$ is the transmit symbol's average power.

Subcarriers in OFDM get assigned with complex signals that are independent identically distributed (i.i.d.) and a linear Inverse Discrete Fourier Transform (IDFT) is performed on each of these subcarriers. This is the linear transform process of high number of i.i.d. QAM-modulated symbols which are complex. This makes the transmit symbol amplitude to rely on constellation point of the modulation scheme. OFDM signal, however, can be estimated to Gaussian waveform in time domain according to central limit theorem. Therefore, transmit symbols tend to have higher amplitude variation based on this property resulting in high PAPR in OFDM.

Output power amplifiers of transmitter would not operate in respective linear region for highly varying transmit symbol. Hence, emission of power to prevent distortions and to operate in linear region would be high, as noteworthy transmit signal distortion occurs due to non-linear clipping. When it comes to uplink transmission, high power emission may drain the battery of the User Equipment (UE) device quickly. Hence Single Carrier

Frequency Division Multiple Access (SC-FDMA) scheme is used for uplink transmission.

## 2.2 Single Carrier Frequency Division Multiple Access (SC-FDMA)

SC-FDMA is used by LTE in the uplink transmission in order to tackle the power issue faced by UE discussed in the previous section. Identical to ODFM, transmission bandwidth is divided into numerous parallel subcarriers in SC-FDMA. Cyclic Prefix (CP) is added as a guard interval to establish orthogonality between subcarriers. However, in contrast to OFDM, Discrete Fourier Transform (DFT) is used in SC-FDMA in which uplink transmissions are multiplexed in a particular frequency allotment within the entire bandwidth which is allocated according to data rate needed for the user.

There is no independent allocation of data symbols to each subcarrier in SC-FDMA as in OFDM, whereas, linear combination of data symbol modulated are allocated to every subcarrier. Single carrier modulation also leads to higher Inter Symbol Interference (ISI) as compared to downlink OFDM [1]. To counter the distortion in the channel, less complex equalizer is used at the receiver (BS)[1].

## 2.3 Time-Frequency Frames

Time domain in LTE is divided into frames and each radio frame is of 10 ms duration. Each of these frames are divided further into subframes of duration 1 ms. Hence there are 10 subframes in a radio frame. Every subframe is further divided into 2 slots of duration 0.5 ms each. OFDM symbols are present in these slots. There will be 7 symbols in a slot for normal Cyclic Prefix (CP) and 6 OFDM symbols for extended CP. Basic time unit for LTE

is defined as $T_s$ = (1/15000 x 2048) s = (1/301720000) s; and useful symbol time is as follows $T_u$ = (20148 x $T_s$) ~ 66.7µs [2].

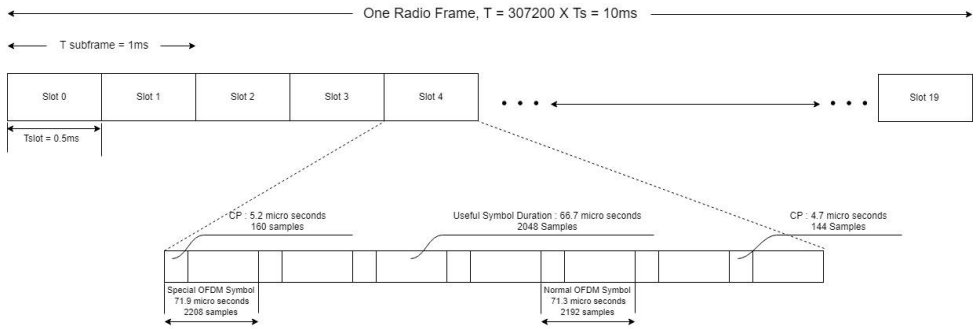


**Figure 2.1. Time frame in OFDM**

Duration of CP for extended mode is $T_{cp}$ = 512 x $T_s$ ~ 16.7µs. Duration of CP of the first symbol in normal mode is $T_{cp}$ = 160 x $T_s$ ~ 5.2µs, while duration of cyclic prefix of all other symbols in that slot is $T_{cp}$ = 144 x $T_s$ ~ 4.7µs. This difference in cyclic prefix between 1$^{st}$ symbol and remaining symbol is because the overall slot duration will be the one divisible by 15360 for easy calculations [2].

## *2.4 Resource Blocks in LTE*

Each Resource Element (RE) comprises only one subcarrier during one OFDM symbol, that is, each RE has place for one modulation symbol and is the smallest resource in LTE. These REs are grouped into Resource Block (RB) such that an RB has one 0.5 ms slot in time domain and 12 consecutive subcarriers in frequency domain. Thus, an RB that uses normal Cyclic Prefix (CP) has 84 RE and 72 RE when the RB uses extended CP [1].

Smallest allocation unit of resource in LTE is a Resource Block (RB). Each User Equipment (UE) when needed to transmit data gets RB allocated by the scheduler at eNodeB. It is represented in time and frequency domains as shown in figure 2.2.

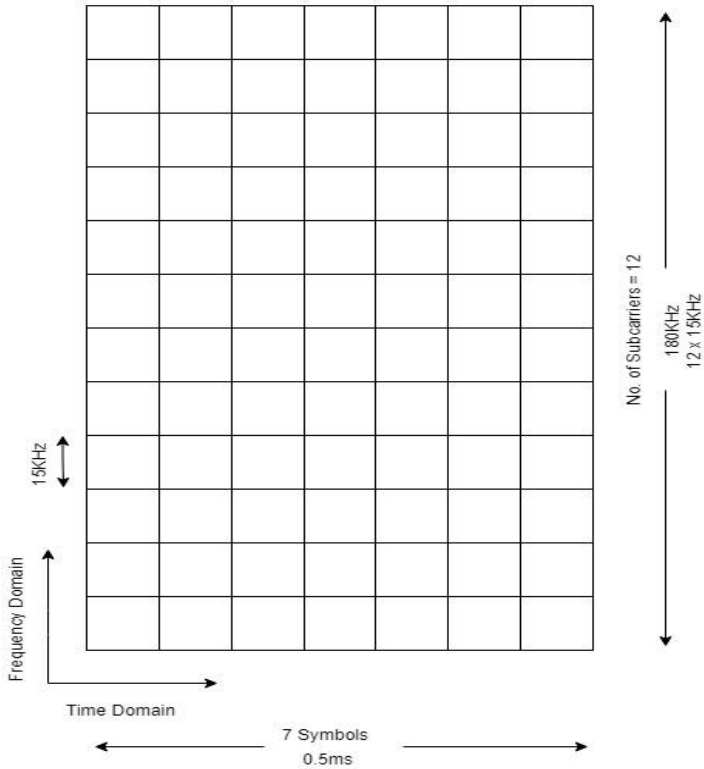


**Figure 2.2. Resource Block Diagram for Normal CP**

Duration of a single RB in time domain is 0.5 ms, where 7 OFDM symbols can fit in normal cyclic prefix conditions, whereas, 6 OFDM symbols fit in an extended cyclic prefix condition. In frequency domain, each RB has 12 subcarriers, each of which takes 15 kHz. Hence, in total, each RB takes 180 kHz (12 x15 kHz) in frequency domain [2].

## *2.5 Reference Signals*

Reference signals are predefined signals that occupy specific resource elements in the time-frequency grid. Reference signals are used to estimate the channel to facilitate coherent demodulation, channel-dependent scheduling or link adaptation [2]. Reference signals are sent in both downlink (downlink reference signals) and uplink (uplink reference signals) and the resulting overhead in LTE is 10% [6]. Reference signals can also be classified as Dedicated Reference Signal and Common Reference Signal based on the number of targets. Dedicated Reference Signal is intended for a

specific terminal whereas Common Reference Signal is shared among a group of terminals [2].

### 2.5.1 Downlink Reference Signals

As mentioned in the section 2.5, downlink reference signals are transmitted using specific resource elements in downlink time-frequency grid. Different types of downlink reference signals are transmitted and are intended for various purposes [2]:

- Cell-specific reference signals (CRS) are sent in every resource block in frequency domain and in every downlink subframe. Devices use CRS for channel estimation for coherent demodulation of all downlink physical channels except PMCH, PDSCH for devices configured in transmission modes 7-10 and EPDCCH control channel. CRS also serves as the basis for cell-selection and handover decisions.
- Demodulation Reference Signal (DM-RS) is used by devices for channel estimation for coherent demodulation of PDSCH in transmission modes 7-10 EPDCCH physical channel.
- Positioning reference signals are intended to estimate geographical position of the device.
- MBSFN reference signals are used by devices, in case of MCH transmission using MBSFN, for channel estimation for coherent demodulation.
- CSI reference signals (CSI-RS) are used specifically by devices configured in transmission modes 9 and 10 to obtain Channel State Information (CSI). CSI contributes to less overhead and offers more flexibility compared to CRS.

### 2.5.2 Uplink Reference Signals

Uplink reference signals, which are of two types, are transmitted using resource elements in uplink time-frequency grid [2].

- Uplink DM-RS are used by eNodeB for channel estimation for coherent demodulation of uplink physical channels: PUSCH which carries both user data and control signal data and PUCCH which carries different types of uplink L1/L2 control signaling. A DM-RS

is always transmitted along with a physical channel and both have the same frequency range.

- Sounding Reference Signal (SRS) is used to obtain the uplink channel estimate at different frequencies which can then be used for uplink frequency dependent scheduling, adaptive coding and modulation (ACM) and precoder selection. SRS is also used when there is no data to send, but an uplink transmission is required. One such application is uplink-timing-alignment procedure where SRS is used to estimate uplink receive timing. In Time Division Duplex (TDD) based systems, uplink and downlink use the same frequency resource [5]. Thus, uplink channel estimates can be utilized for downlink transmission purposes and this is referred to as channel reciprocity. By exploiting channel reciprocity, SRS based channel estimates can also be used for downlink transmission purposes. Unlike DM-RS, SRS may have a different frequency range compared to the physical channel transmitted along with it.

In this TDD based system at Ericsson, eNodeB uses SRS based channel estimates for reciprocity-based beamforming. This thesis focuses on investigating the redundancy between SRS based channel estimates and implementing lossless and lossy compression schemes for the estimates to enhance the cell capacity for reciprocity-based beamforming. SRS will be discussed in detail in the subsequent section.

## *2.6 Sounding Reference Signals (SRS)*

SRS can be classified as periodic SRS or aperiodic SRS based on if they are sent at regular time intervals or are triggered when there is a requirement [2].

**Periodic SRS transmission**

Periodic SRS transmissions occur regularly in time domain as frequently as once every 2 ms or infrequently as once every 160 ms. SRS is sent in the last symbol of a subframe and in TDD, SRS can be transmitted in UpPTS as well [2]. The bandwidth of SRS is a multiple of four resource blocks. SRS transmissions need to extend over the frequency range of interest to the scheduler. This can be realized in two ways [1]:

A wideband SRS that spans over the entire frequency range of interest in a single SRS transmission may be used. This is typically used in scenarios with

good coverage. A UE at cell boundary may not have enough transmit power to send a wideband SRS.

The second approach is to use frequency hopping where narrowband SRS transmissions that hop in frequency are used to cover the frequency range of interest over multiple transmissions. This technique is preferred when channel conditions are poor.

By exploiting orthogonality, SRS can be transmitted from different antennas at the same time using the same frequency without interfering with each other. One method to achieve orthogonality is to apply different cyclic shifts to SRS. In case of SRS, there are 8 different cyclic shifts possible. This means that SRS from 8 UEs can be transmitted using the same time-frequency resources [1].

Another way of multiplexing SRS transmission is by mapping SRS to every second subcarrier creating a comb-like pattern. Release 13 and above permit the use of up to four different combs instead of two [2]. In the case of four different combs, SRS is mapped to every fourth subcarrier. This increase will help in supporting the increase in antennas with the introduction of FD-MIMO. With the help of above mentioned multiplexing strategies, 16 UEs can be multiplexed in case of two combs (8 cyclic shifts x 2 combs) and 32 UEs can be multiplexed in case of four combs (8 cyclic shifts x 4 combs) over the same time-frequency resources.

A device is configured with a set of parameters that defines the characteristic of an SRS transmission by means of higher-layer (RRC) signaling.

**Aperiodic SRS transmission**

Aperiodic SRS transmissions are one-shot transmissions that are carried out when the device is explicitly triggered to do so. When such a trigger is received, the device sends SRS in the next available aperiodic SRS instant and further SRS transmissions are made if additional triggers are received [2]. Aperiodic SRS is sent in the last symbol of a subframe and has the same frequency-domain structure as periodic SRS. Higher-layer (RRC) signaling is used to configure frequency-domain parameters of aperiodic SRS.

## 2.7 Multiantenna Systems

Multiple Input Multiple Output (MIMO) is a radio communication arrangement in which transmit and receive antennas are used for data transmission. It is widely used in many standards such as Long Term Evolution (LTE), WiFi 802.11n, Fifth Generation (5G) cellular uplink and downlink.

In traditional Single Input Single Output (SISO) system, capacity can be increased only by increasing transmitting power or bandwidth based on Shannon channel capacity theorem:

$$C = BW(log2(1 + SNR)) \text{ bits/s/Hz} \quad (2.2)$$

where BW is bandwidth and SNR is Signal to Noise Ratio.

However, increasing bandwidth is not an optimal solution to increase data rate as frequency spectrum is a limited resource. Capacity in SISO increases logarithmically with increasing power. However, increasing transmit power requires expensive radio frequency (RF) amplifier as it would interfere with neighboring cells and make battery less efficient as well [19].

Hence MIMO increases channel capacity, also known as spectral efficiency, without increasing bandwidth or power and this capacity increases linearly with increasing number of antennas. Antennas in this configuration uses the same bandwidth, hence, all the antennas on the receiver side receive data from all the transmitting antennas. A system with $N_r$ RX antennas and $N_t$ TX antennas forms a $N_r \times N_t$ MIMO system. A 2x2 MIMO system has been illustrated in figure 2.4:

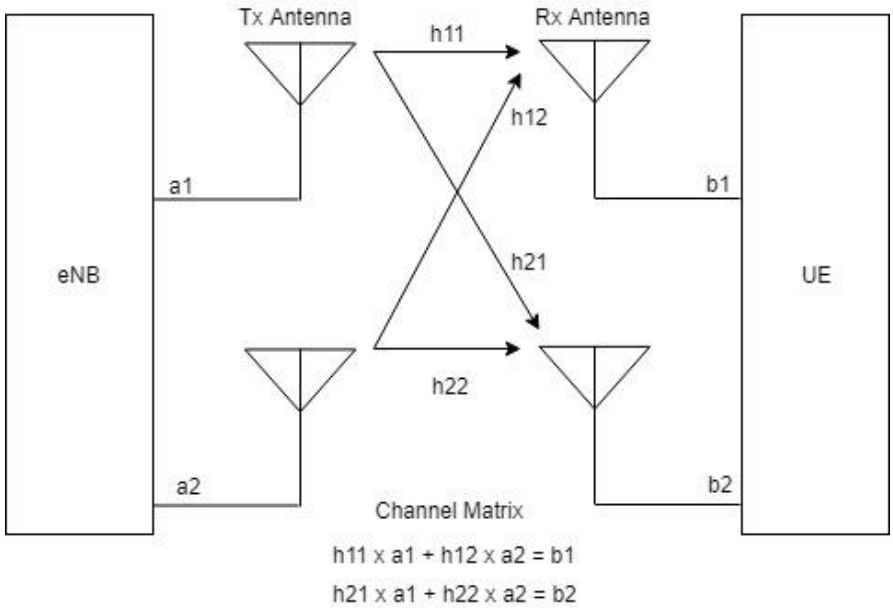


**Figure 2.3: A 2x2 MIMO System.**

By using MIMO systems, two types of gains can be obtained: spatial multiplexing and diversity gain. Spatial multiplexing involves multiple parallel transmissions on the same time-frequency resource. Sending data in parallel channels gives rise to a higher transmission rate. Secondly, diversity gain can be obtained by transmitting same information through multiple transmitting antennas which results in better link reliability. Hence, even if some of the links are down, receiver can efficiently decode the transmitted data with the help of remaining working links. However, increasing spatial multiplexing will reduce diversity gain, as they are inversely proportional to each other. Therefore, this trade-off must be considered while designing a MIMO system.

# CHAPTER 3

## *Correlation, Data compression and Entropy*

In this chapter, correlation between SRS based channel estimates corresponding to different PRBGs and correlation of SRS values between columns and rows of antenna are determined and analyzed. Results obtained from the tests performed in this chapter form the basis for rest of the thesis as they help in understanding the statistical properties of the channel estimates, and therefore provide insights on compression capabilities. This chapter also discusses about lossless and lossy compression schemes. Entropy, which is an important parameter in data compression is also discussed in this chapter.

## *3.1 Correlation*

Correlation is a statistical tool for discovering and measuring the relationship between variables [20]. The measure of correlation that summarizes direction and degree of correlation in one figure is called as correlation coefficient or correlation index. A positive correlation specifies the degree to which the variables increase or decrease in parallel and a negative correlation represents the degree to which one variable increases as the other decreases. Correlation is also a measurement of similarity.

Pearson's coefficient of correlation is widely used for correlation analysis between variables and is defined as [20]:

$$r = \frac{\sum xy}{N\sigma_x\sigma_y} \tag{3.1}$$

where x = (X - $\bar{X}$) and y = (Y - $\bar{Y}$), $\sigma_x$ and $\sigma_y$ are standard deviations of series X and Y respectively, N is the number of pairs of observations and r is the correlation coefficient.

When r = +1, it means perfect positive correlation; r = -1, means perfect negative correlation and r = 0 means there is no relationship between the variables [20].

The correlation between two discrete-time complex signals h(n) and g(n) is defined in equation 3.2 [19]:

$$R(\tau) = \sum_{n=1}^{N} h(n) \times g^*(n+\tau) \quad (3.2)$$

where complex conjugation is denoted by an asterisk and N is the number of samples. Function R($\tau$) represents the cross correlation between h(n) and g(n). It is also fine to introduce time lag in h(n) instead of g(n) to investigate similarity between the two signals. The function R($\tau$) peaks around some value of $\tau$ when h(n) and g(n) are same, whereas function R($\tau$) would be low over the values of $\tau$, if the two signals are different.

## 3.2 Autocorrelation

Autocorrelation is the correlation of a signal with itself and is defined as follows [19]:

$$R(\tau) = \sum_{n=1}^{N} h(n) \times h^*(n+\tau) \quad (3.3)$$

**Cauchy-Schwarz Inequality**

Cauchy-Schwarz Inequality is used to normalize correlation values such that magnitude of correlation is not bigger than 1. Autocorrelation with Cauchy-Schwarz normalization is given by [21]:

$$R(\tau) = \frac{\sum_{n=1}^{N} h(n) \times h^*(n+\tau)}{\sqrt{\sum_{n=1}^{N} h(n) \times h^*(n)} \times \sqrt{\sum_{n=1}^{N} h(n+\tau) \times h^*(n+\tau)}} \quad (3.5)$$

Where h(n) is the sample, $\tau$ refers to the separation between samples (number of antennas or PRBGs in between the samples under consideration), N is the number of samples.

Equation 3.5 was implemented in MATLAB to study the correlation between SRS values corresponding to different PRBGs and correlation between SRS values corresponding to different columns and rows of antenna array. Correlation between SRS values will give us an idea of how much redundancy is present in them and this is important as higher the redundancy, more compressible the data is. The results of correlation analysis for LOS and NLOS scenarios are shown below:

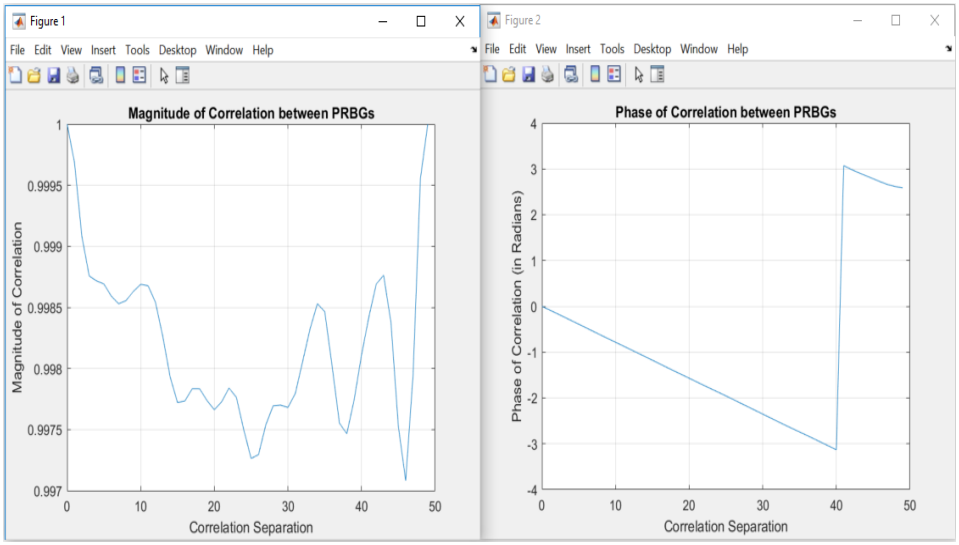


**Figure 3.1: Magnitude and phase of correlation between PRBGs for UE1 in LOS conditions**

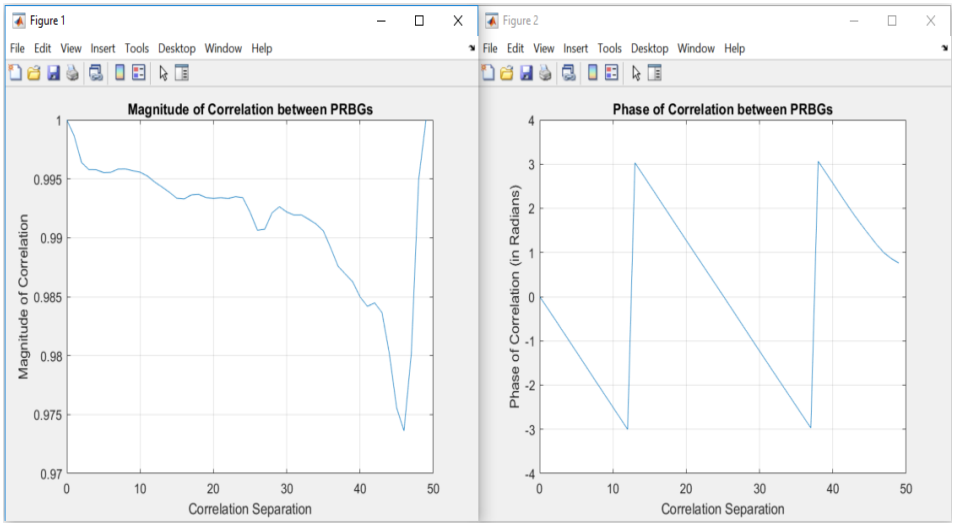


**Figure 3.2: Magnitude and phase of correlation between PRBGs for UE2 in LOS conditions**

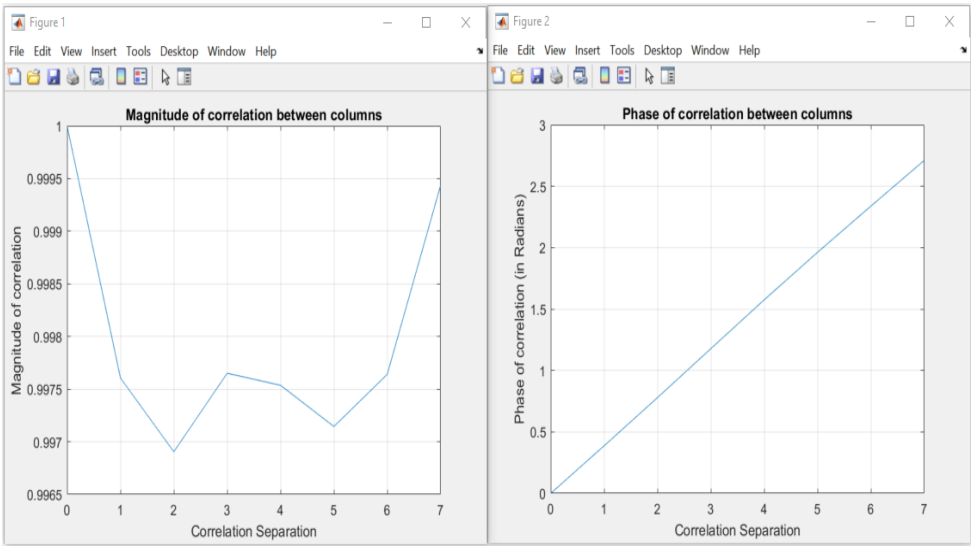


**Figure 3.3: Magnitude and phase of correlation between columns of antenna array for UE1 in LOS conditions**

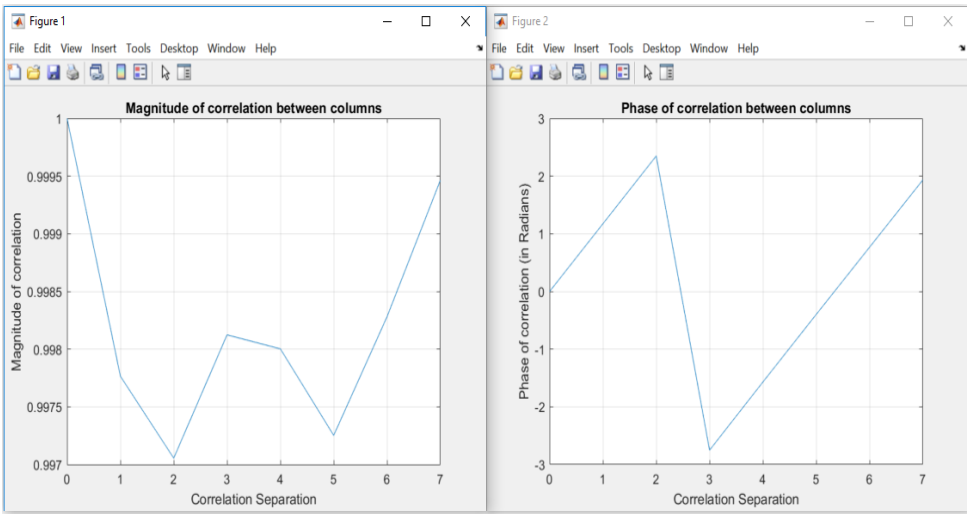


**Figure 3.4: Magnitude and phase of correlation between columns of antenna array for UE2 in LOS conditions**

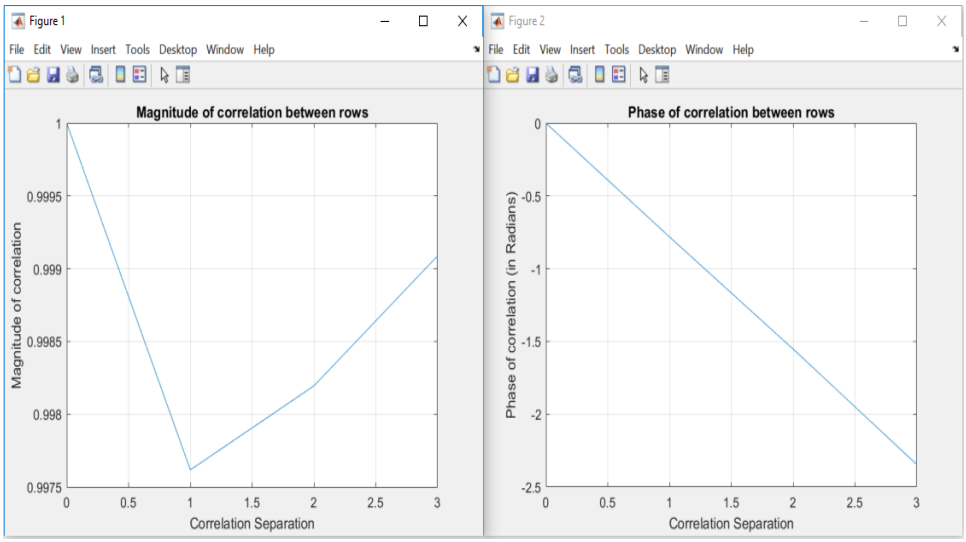


**Figure 3.5: Magnitude and phase of correlation between rows of antenna array for UE1 in LOS conditions**

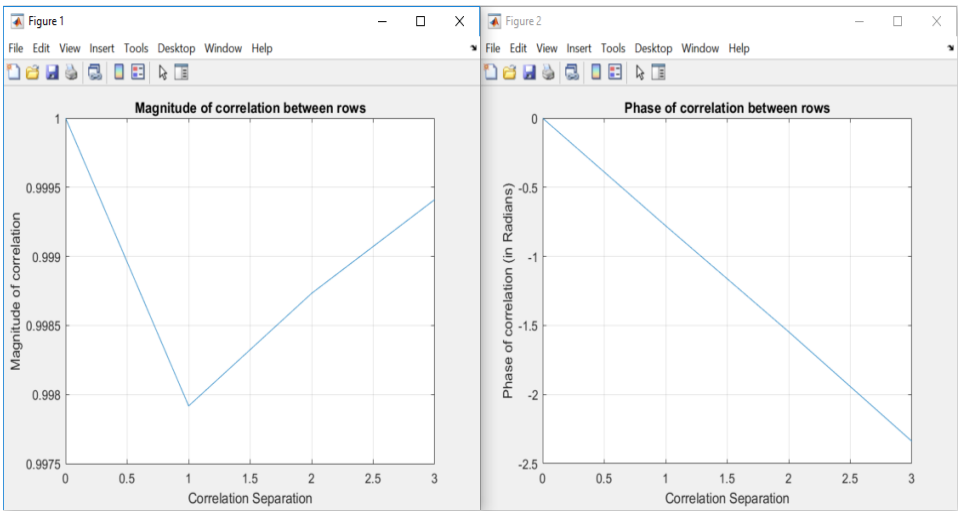


**Figure 3.6: Magnitude and phase of correlation between rows of antenna array for UE2 in LOS conditions**

It can be seen from the plots that the magnitude of correlation is high (close to 1) between PRBGs and columns and rows of antenna array; and phase of correlation is linear. The high magnitude of correlation implies that the signals received at eNodeB are of similar levels across PRBGs and across the antenna array. Linear phase of correlation in case of PRBGs is due to the fixed delay and in case of columns and rows of antenna array is due to fixed angle of arrival. High correlation between SRS values observed in case of LOS UE implies that there is high similarity between SRS values across PRBGs and antenna array, which makes the data redundant and possible to compress.

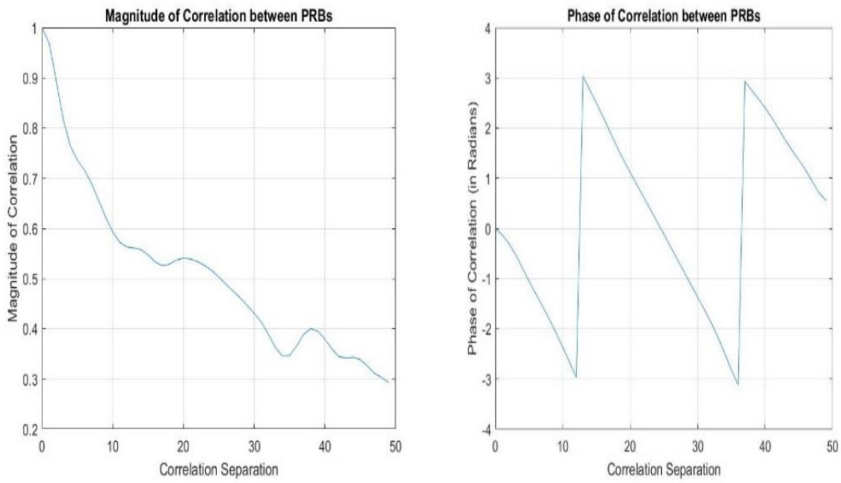


**Figure 3.7: Magnitude and phase of correlation between PRBGs for UE in NLOS conditions**

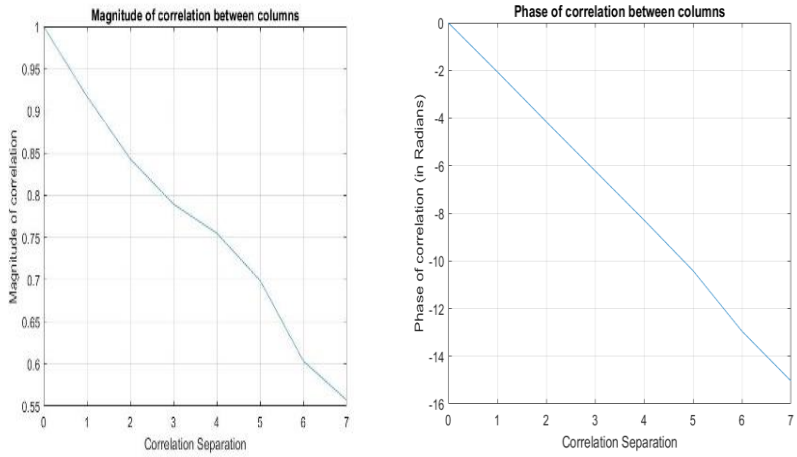


**Figure 3.8: Magnitude and phase of correlation between columns of antenna array for UE in NLOS conditions**

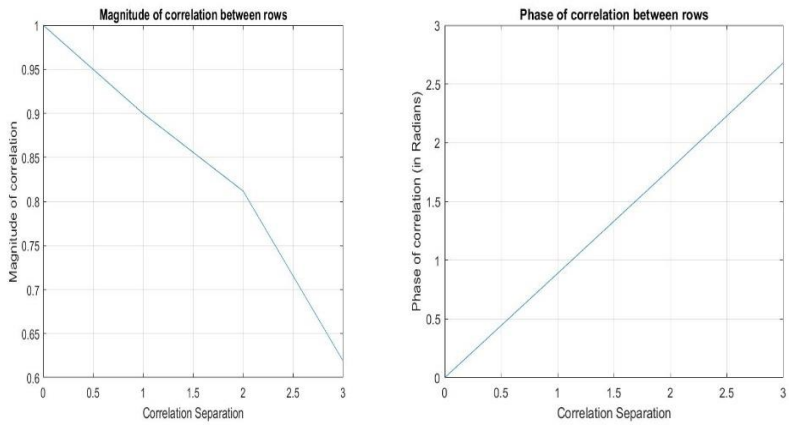


**Figure 3.9: Magnitude and phase of correlation between rows of antenna array for UE in NLOS conditions**

It can be observed from the plots that magnitude of correlation is low compared to LOS UEs. This implies that there is less similarity between the received signals corresponding to different PRBGs and across the antenna array which implies that the received signals are more random compared to that of LOS UE. However, phase of correlation is linear which implies nearly uniform change in phase.

## 3.3 DATA COMPRESSION

The process of representing data with fewer number of bits is called data compression and the process of recreating the original data from the compressed data is called as decompression [9]. Compression can be lossless or lossy based on the output of reconstruction algorithm. In lossless compression, the original data can be retrieved from the compressed data without any loss of information. On the other hand, lossy compression algorithm involves some loss of information. The latter technique helps to achieve better compression compared to the former [10].

A compression algorithm is implemented in two phases [9]:

Modeling: This phase involves extracting the information about parameters like redundancy or probability present in data and representing it as a model.

Coding: In this phase, input data bits are mapped to a sequence of fewer number of bits based on the model generated in phase 1.

The performance of a compression algorithm can be measured in different ways like amount of compression, speed at which data is compressed and decompressed, complexity of the algorithm, memory requirement of the algorithm and so on [9].

The level of compression can be specified by compression ratio or space savings. Compression ratio is defined as [10]

$$Compression\ ratio = \frac{size\ of\ original\ data}{size\ of\ compressed\ data}.$$

Space savings term is defined in [10] as

$$Space\ savings = \left(1 - \frac{size\ of\ compressed\ data}{size\ of\ original\ data}\right) \times 100\ \% \qquad (3.6)$$

Another important performance measure is the time required to compress and decompress the data, namely compression and decompression times [11].

## 3.4 Lossless Compression

In lossless compression, the data recreated from the compressed data is identical to the original data, that is, there is no loss of information [9]. This technique is used in scenarios where the integrity of the data must be preserved.

A scenario where lossless compression is widely used is text compression as a small difference between original and reconstructed texts may result in statements with entirely different meanings.

Chapter 4 of this thesis describes the implementation of lossless compression of SRS based channel estimates.

## 3.5 Lossy Compression

In lossy compression, the recovered data is not identical to the original data, that is, there is loss of information [10]. The advantage of this technique is that one can achieve higher compression compared to the lossless counterpart. The difference between reconstructed data and original data is called distortion [9].

Fidelity and quality are other terms that are used when describing the difference between reconstructed and original data streams. A high fidelity or quality means that the difference between the original and the reconstructed is small.

This technique can be employed in cases where loss of some information can be tolerated. Examples of such cases include compression of speech, image and video.

Chapter 5 of this thesis describes the implementation of lossy compression of SRS based channel estimates.

## 3.6 Entropy

Entropy is described as the average information required to determine the outcome of a random variable. The entropy of a random variable X can be defined as [16]:

$$H(X) = E_X[-\log p(X)] = -\sum_x p(x).log_2 p(x) \quad (3.7)$$

where x denotes the possible values of random variable X and p(x) is their corresponding probability.

The best a lossless compression algorithm could do is to encode the outcome of a source with an average number of bits equal to its entropy

[9]. Hence, entropy is of great importance to this thesis as it gives an idea about the maximum compression possible.

SRS based channel estimates for the two UEs are available in Hexadecimal format. A hex digit (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) can be represented using 4 bits.

Channel estimates corresponding to first SRS occasion of UE1 in LOS were selected and entropy was calculated using equation 3.7, which was found to be 3.4011. This means that each digit can be represented using 3.4011 bits rather than 4 bits. This translates to a space savings as shown below:

$$Space\ savings = \left(1 - \frac{compressed\ size}{uncompressed\ size}\right) \times 100\ \%$$

$$= \left(1 - \frac{3.4011}{4}\right)\% = 14.97\ \%$$

This tells us that maximum space savings possible in this case is 14.97 %.

Similarly, for LOS UE2 SRS, entropy was found to be 3.4185. This translates to a maximum space savings of 14.54 %.

For NLOS UE SRS, entropy was found to be 3.0563. This tells us that maximum space savings possible in the case of NLOS UE is 23.59 %.

It can be seen that entropy corresponding to NLOS UE SRS is lower than that LOS UE SRS. The received signal strength is low in case of NLOS UE, and many of the hexadecimal digits were '0' and 'f' corresponding to very low positive or negative value respectively. We believe, these unused bits in case of NLOS UE cause its entropy to be lower than that of LOS UE and translates to higher space savings.

# CHAPTER 4

## *Lossless compression of SRS based channel estimates*

As mentioned in chapter 1, lossless compression of channel estimates is of high importance to Ericsson. Hence this chapter describes the implementation of lossless compression of SRS based channel estimates. Performance of multiple lossless compression formats such as BZIP2, GZIP, ZIP, LZMA and LZO was evaluated. The best method is the one with higher compression ratio along with shortest compression and decompression times and lowest implementation complexity. However, in reality there is a trade-off among these, which is also analyzed. Before explaining the compression formats this chapter begins with the basic explanation of dictionary-based algorithms such as LZ77 and Deflate which form the basis for the above mentioned compression formats. This chapter also discusses about entropy-based encoding such as Huffman coding and Arithmetic coding.

## *4.1 Entropy Coding*

Entropy coding is a lossless data compression technique that compresses data by replacing fixed-length input symbols with variable-length prefix-free output codeword [22]. Arithmetic coding and Huffman coding are the most commonly used lossless entropy coding schemes.

## *4.2 Huffman Coding*

The objective of this algorithm is to compress data by generating prefix codes, which refers to uniquely decodable codes. Let $p(x_i)$ be the probability for a finite source $x=x_1,x_2,.....,x_N$. Codewords that are longer should be assigned to symbols with lower probability and codewords that are shorter should be assigned to symbols with greater probability by the binary code

Let each node represent each symbol with respective probability which are all arranged in descending order. Node with smaller probability is chosen where there are multiple nodes and binary 1 or a 0 is assigned randomly after which two nodes must be combined to form a new node with a probability

equivalent to the sum of two previous nodes. Every symbol's codeword must be traced from last node left which is also known as the root node. An example of Huffman encoding calculation is shown in figure 4.1:

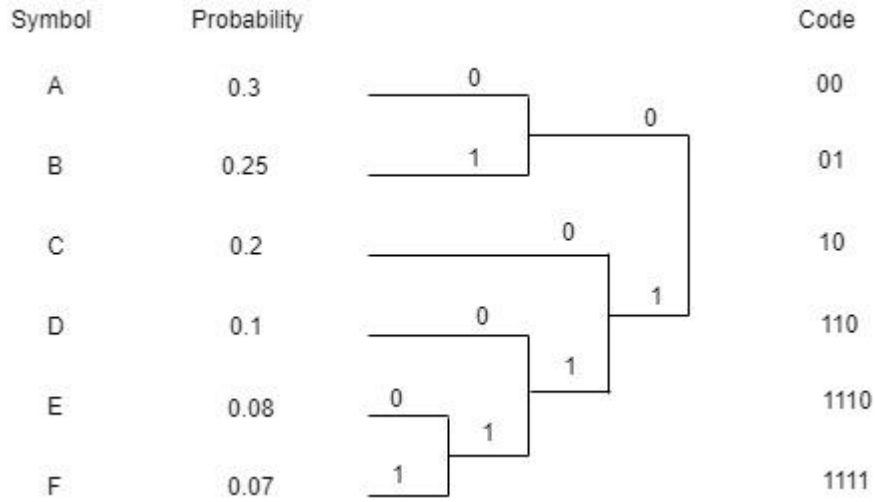


**Figure 4.1: Huffman coding example according to probability of symbols be coded**

Huffman coding performed on LOS UE1 channel estimates resulted in a space savings of 14.67 % and a space savings of 14.40 % for LOS UE2.

Huffman coding performed on NLOS channel estimates resulted in a space savings of 22.66%.

## *4.3 Arithmetic Coding*

One of the biggest downsides of Huffman coding is that a minimum of one bit is required to represent a symbol. This issue can be resolved by using blocks of symbols to enhance the efficiency of the code but becomes extremely complicated with increasing length of blocks.

Therefore, in arithmetic coding, series of symbols are represented in a range between 0 and 1 which spans based on the symbol probability. First range is decided based on the probability of first symbol to get coded. Similar range allotment takes place for subsequent symbols as well, symbols with larger

probability would be allocated larger range by which less numbers of bits can be used to represent large probability symbols.

Arithmetic coding is relatively better than Huffman coding because in reference to code from the preceding series of symbol, new symbols can be coded. This removes the need to retain long codebook as symbols can be coded constructively with existing information. Only synchronized update in probabilities of encoder and decoder is required. Also, the source data in this coding scheme can also dynamically reconstruct to statistical changes. An example of Arithmetic encoding calculation is shown in figure 4.2:

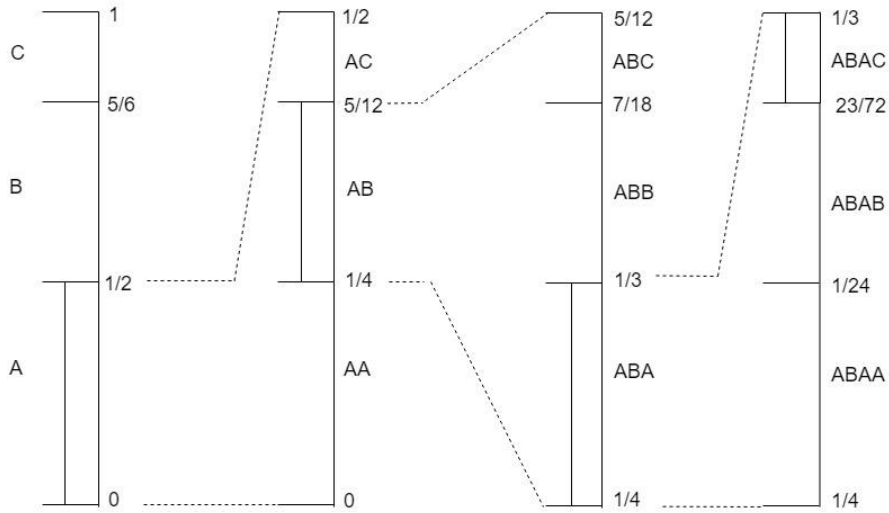


**Figure 4.2: Arithmetic Coding example for sequence of source 'ABAC' based on the probability of each symbol.**

Arithmetic coding was performed for LOS UE1 channel estimates and resulted in a space savings of 14.96% and a space savings of 14.53% for LOS UE2.

Arithmetic coding performed for NLOS channel estimates resulted in a space saving of 23.58%.

| | Space Savings for UE1 in LOS | Space Savings for UE2 in LOS | Space Savings for UE in NLOS |
|---|---|---|---|
| Entropy | 14.97 % | 14.54 % | 23.59 % |
| Huffman Encoder | 14.67 % | 14.40 % | 22.66 % |
| Arithmetic Encoder | 14.96 % | 14.53 % | 23.58 % |

**Table 4.1: Table comparing space savings specified by entropy and that achieved by Huffman and Arithmetic encoders.**

## *4.4 LZ77*

It is a sliding window technique which consists of two buffers namely search buffer which stores previous characters of length S and look ahead buffer which stores characters of length B.

Algorithm:

Firstly, search buffer must be loaded with S characters of the text and n needs to be set as S+1.Proceed till n in the text of search buffer to identify the offset $X_n$.

If number of offsets is greater than 0

1. then largest match of offset is to be found which gives length l and index j
2. Codeword has to be initialized to (j,l,c) where c = $X_{n+l}$
3. Initialize n to n+l+1.

Else

1. Initialize codeword to (0,0,c) where c=$X_n$.
2. Initialize n to n+1.

After all this, look ahead buffer and search buffer need to be updated. Search buffer and look ahead buffer are illustrated in figure 4.3:

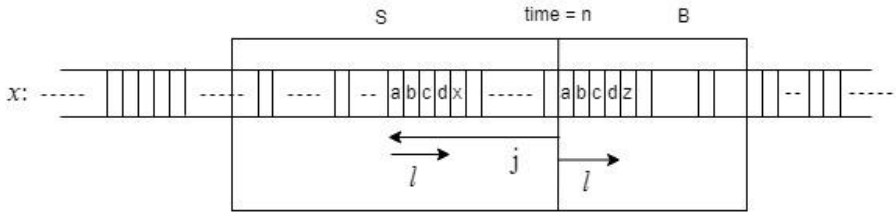


**Figure 4.3: Search buffer and look ahead buffer diagram for LZ77**

In the look ahead buffer the algorithm looks for maximum match of $X_n$ and its successors to find if there is a match or not. When there is a match algorithm would set the length as l and make the start of the match as index j which is the offset between look ahead buffer and search buffer. When no match is found, j,l will be set to 0,0. Successive character needs to be appended from the string with the codeword in order for it to proceed further when no match is found [16]. Therefore, codeword after a match identified is (j,l,c).

## *4.5 Deflate*

In Deflate, input data is separated into blocks and then compressed. Based on the encoder memory and prefix codes, these blocks will have variable lengths. Blocks of any size should be decoded by the decoder in deflate. Three different approach would be followed by deflate for compression. [10]

Approach 1 Without compression: Data files that cannot be compressed, or uncompressed files that require compression software to segment file, use this approach. Specific header is used to indicate that this approach is used for compression. Maximum data length should be 65,535. Also, code tables are not used in this approach.

Approach 2 Fixed code table compression: Data will be used to build 2 code tables in this approach. This results in higher compression and decompression speeds. However, the speeds may get affected to some extent as the data used to build code table is statistically different from that data getting compressed. In the first code table, match lengths of data are present. Second code table consists of distances. Just like the first approach specific header to indicate this approach is used for compression.

Approach 3 Individual code table compression: Encoder for data produces individual code table. Highly advanced Deflate encoder traverses from one

block to another and builds code table based on the statistics of data obtained when compressing blocks. Block starts with a specific header followed by compressed Huffman code table and then each of two code blocks compressed by Huffman code. Finally, single prefix code after compressed data is used to indicate the end of block.

## 4.6 Lossless compression file formats

Lossless compression file formats use a combination of data transformation algorithms (like Burrows-Wheeler Transform, Move-to-front algorithm) and lossless compression algorithms or combinations of different compression algorithms (like Deflate) to implement compression. The use of data transformation algorithms or combination of compression algorithms help in transforming the data to have lower entropy. This helps in achieving higher compression compared to that achieved by a compression algorithm on its own. The compression file formats that we implemented in this thesis are listed in sections 4.6.1 – 4.6.5. The performance of these formats was analyzed based on their compression ratio and compression and decompression times.

### 4.6.1 LZMA

Lempel-Ziv-Markov chain (LZMA) algorithm is a dictionary based lossless compression algorithm. An encoder same as arithmetic encoding is used at the output known as range encoder. Encoding is done faster by this encoder with minuscule effect in the efficiency of compression. Highly advanced data structure is used to detect likelihood expectation of bits by the compressor to choose the most optimal one. In general, LZMA offers higher compression than other formats because the encoder uses bit based model for compression as opposed to byte based compression model and memory allocation for dictionary is also higher. Hence unconnected bits are not combined making it easier for compression. [10]

### 4.6.2 ZIP

It is a file compression format which uses Deflate as the algorithm for lossless compression. 32-bit CRC algorithm and dual archive directory structure help in data from getting lost. This file format which is archive may contain multiple files or folders inside the main zip directory. Advantage with using this format is because it gets easier to add or remove data files

separately from a zip archive as there is no requirement to apply compression or decompression to the entire directory as this facility is not possible with other data compression formats [15].

### 4.6.3 GZIP

Gzip is another compression algorithm that uses Deflate. It contains a header of size 10 bytes, payload that is compressed as body using Deflate and footer size of 8 bytes which includes information on actual uncompressed data's length and 32-bit CRC checksum. Generally, gzip compresses files one after the other even though the format is capable to compress multiple files simultaneously [16]. Data files get collected into a single archive with tar extension which later gets compressed with gzip. Although gzip uses Deflate just like zip, gzip requires external archive such as tar to hold data file whereas zip does not have such requirements as files gets compressed separately and does not exploit redundancy between one file and another file.

### 4.6.4 BZIP2

Bzip2 is a compression program that uses Huffman coding and Burrows Wheeler Transform (BWT), which is a sorting algorithm for text. Compression ratio achieved in general is better than that obtained from compressor based on LZ77/LZ78. Data gets encoded based on run length encoding initially, after which sorting algorithms namely Burrows Wheeler and Move-to-front (MTF) are applied. Finally, Huffman coding is applied. Bzip2 does compress in multiple stages and decompresses in reverse order. It consists of 4 Byte header and 32-bit CRC at the end [18]. It is difficult to extract data because of syntax errors; however, it is possible to compress and decompress damaged archives.

### 4.6.5 LZO

LZO is a lossless compression format and is described in detail in section 4.9.

## 4.7 Compression and Decompression times

The time required to compress data (or a file) is called compression time. Similarly, the time required for decompression is called as decompression time. These two parameters are of paramount importance in case of lossless compression in this thesis, as the requirement at Ericsson was to implement an algorithm that takes a maximum of 10 ms as the sum of compression and decompression times. The performance of the above mentioned compression programs was simulated on Linux-based servers. A Linux-based server provides multiple time metrics that help us to evaluate the execution time of a program and are listed below [23]:

**Real** is wall clock time, that is the time from the beginning to the end of the call. This is all elapsed time including time slices used by other processes and time the spent by the process when it is blocked that is for instance if it is waiting for I/O to complete.

**User** is the amount of CPU time spent in user-mode within the process. This is the actual CPU time used in executing the process outside the kernel. Other processes and time the process spends blocked do not count towards this figure.

**Sys** is the amount of CPU time spent in the kernel-mode. This means executing CPU time spent in system calls within the kernel. Like 'user', this is only CPU time used by the process.

**User + Sys** time is the actual CPU time a process takes and is used to evaluate the execution time of a code.

Therefore, **User** time + **Sys** time was considered to evaluate time for compression and decompression.

## 4.8 Evaluating the performance of compression file formats on SRS based channel estimates on a Linux-based server

The SRS based estimates corresponding to one occasion of a UE were selected and copied to a text file. The compression programs were executed on a Linux-based server, and their compression ratio and speed were found

using Linux terminal commands implemented as a Shell script. The SRS based estimates in this system are in hexadecimal format and the CSI size corresponding to one SRS occasion of a user amounts to 25600 bytes (as described in section 1.1). However, when this CSI is stored in a file, hexadecimal digits are considered as characters and each character in a file is represented by 8 bits. This implies that each hexadecimal digit in the file is represented by 8 bits rather than 4 bits. This results in a file size double that of the CSI it stores, that is, 51200 bytes. Hence, when compression ratio or spacing savings is evaluated, a factor of 2 is removed from the result in order to remove the redundancy due to hexadecimal digits being treated as characters in file.

This process was performed on LOS UE1, LOS UE2 and NLOS UE. Results of testing the performance of compression file formats on Linux-based server are provided below:

**LOS UE1**

| Compression Format | Compression Time (ms) | Decompression Time (ms) | Compression Ratio | Space Savings (%) |
|---|---|---|---|---|
| BZIP2 | 0m0.009s | 0m0.005s | 1.75 | 35.71 |
| GZIP | 0m0.009s | 0m0.004s | 1.4 | 32.14 |
| ZIP | 0m0.010s | 0m0.007s | 1.4 | 32.14 |
| LZMA | 0m0.049s | 0m0.006s | 2.33 | 39.28 |
| LZO | 0m0.005s | 0m0.004s | 1.29 | 22.48 |

**Table 4.2: Performance comparison of various lossless compression formats for LOS UE1.**

**LOS UE2**

| Compression Format | Compression Time (ms) | Decompression Time (ms) | Compression Ratio | Space Savings (%) |
|---|---|---|---|---|
| BZIP2 | 0m0.009 | 0m0.006s | 1.4 | 32.14 |
| GZIP | 0m0.009s | 0m0.004s | 1.4 | 32.14 |
| ZIP | 0m0.009s | 0m0.007s | 1.4 | 32.14 |
| LZMA | 0m0.049s | 0m0.007s | 1.75 | 35.71 |
| LZO | 0m0.005s | 0m0.003s | 1.27 | 21.72 |

**Table 4.3: Performance comparison of various lossless compression formats for LOS UE2.**

**NLOS UE**

| Compression Format | Compression Time (ms) | Decompression Time (ms) | Compression Ratio | Space Savings (%) |
|---|---|---|---|---|
| BZIP2 | 0m0.009s | 0m0.005s | 1.75 | 35.71 |
| GZIP | 0m0.009s | 0m0.003s | 1.75 | 35.71 |
| ZIP | 0m0.011s | 0m0.007s | 1.75 | 35.71 |
| LZMA | 0m0.052s | 0m0.006s | 2.33 | 39.28 |
| LZO | 0m0.005s | 0m0.004s | 1.36 | 26.87 |

**Table 4.4: Performance comparison of various lossless compression formats for NLOS UE.**

As can be seen in the results above, LZO has the lowest compression ratio. However, it outperforms all the other compression file formats in terms of compression and decompression speeds. As compression and decompression times are of high importance to Ericsson and LZO is the method that offers the least sum of compression and decompression times, we chose to proceed with LZO as the algorithm to implement on Ericsson's test environment. The next task was to obtain the source code of LZO and implement it on the test environment. This process is discussed in following sections.

## *4.9 Lempel-Ziv-Oberhumer (LZO)*

LZO is a lossless compression format that compresses and decompresses blocks of data, where the size of each block must be similar for both compression and decompression. However, for incompressible data, this compression format mitigates the problem by extending the input block to a maximum size of 64B every 1024B input data. In LZO, similar to LZ77 discussed earlier, blocks of data get compressed based on matches and runs of non-matching literals. LZO is extremely cautious with lengthy matches and literals in order to exploit highly redundant data.

An important property of this compression format is that it offers high speed compression and decompression without any memory requirement for decompression. It is also possible to achieve higher compression ratios with small compromise on compression speed, however, decompression speed can still be high. This algorithm and implementations are copyrighted and opensource distributed under GNU General Public License. There are multiple algorithms in LZO based on compression and decompression speeds

and compression ratio, and the default algorithm is LZO1X-1 which offers a good trade-off between compression ratio and compression and decompression speeds.

The source code of LZO was then obtained. LZO is written in ANSI C. The source code was then analyzed and modified to compress and decompress SRS based estimates corresponding to one occasion of a UE. The input to compression process and output of decompression process were checked to make sure compression implemented was lossless. The process was performed on SRS based channel estimates corresponding to LOS UE1, LOS UE2 and NLOS UE. The default compression algorithm of LZO source code, LZO1X-1, was used in our tests. Results of the tests are presented in the table below:

| UE | CSI Size (bytes) | Compressed CSI Size (bytes) | Space Savings (%) | Sum of compression and decompression times (ms) |
|---|---|---|---|---|
| LOS UE1 | 25600 | 19844 | 22.4844 | 7 |
| LOS UE2 | 25600 | 20040 | 21.7187 | 7 |
| NLOS UE | 25600 | 18720 | 26.8750 | 7 |

**Table 4.5: Table displaying CSI size, compressed CSI size, space savings and sum of compression and decompression times for different UEs using LZO source**

As can be seen from the results, LZO takes only 7 ms to compress and decompress the data while enabling a good amount of space savings.

However, the source code of LZO consists of numerous C source files and header files, making it difficult to implement on Ericsson's DSPs. MiniLZO, a very lightweight subset of LZO library [17], was thus considered as a replacement for LZO and is discussed in section 4.10.

## 4.10 MiniLZO

MiniLZO as mentioned in section 4.9 is a lightweight subset of LZO. It implements the LZO1X-1 compressor and both the standard and safe LZO1X decompressor. Apart from fast compression it also useful for situations where there is a need to use pre-compressed data files. MiniLZO consists of only one C source file: minilzo.c and three header files: minilzo.h, lzoconf.h, lzodefs.h making it easier to implement on Ericsson's DSPs. MiniLZO works similar to LZO as the source file minilzo.c is automatically produced from the LZO sources [17].

The source code of miniLZO was analyzed and modified to compress and decompress SRS based estimates of all UE's. MiniLZO uses LZO1X-1 algorithm for compression, which is also the default algorithm of LZO. Hence, the performance of MiniLZO source code in terms of space savings and time for compression and decompression is the same as that of LZO (as shown in section 4.9). However, miniLZO is more efficient compared to LZO in terms of ease of implementation on Ericsson's DSP.

# CHAPTER 5

## *Lossy compression of SRS based channel estimates*

This chapter describes the implementation of lossy compression of SRS based channel estimates. As was discussed in chapter 3, lossy compression involves some loss of information, so data integrity is not guaranteed. However, such an approach will help in achieving higher compression compared to lossless compression.

In Chapter 3, correlation between different PRBGs and between columns and rows of the antenna array at eNodeB were studied. In case of LOS UEs, it can be observed that the magnitude of SRS based channel estimates corresponding to different PRBGs was fairly the same and change in phase of SRS values when moving from one PRBG to the next remained nearly same across the PRBGs. Similar relationships were observed in case of columns as well as rows of antenna array at the eNodeB.

This led us to the thought that if we know the phase shift between SRS values corresponding to adjacent pair of PRBGs of a UE TX - eNodeB RX antenna pair, phases of SRS values corresponding to all PRBGs can be obtained from a reference SRS value (which is a complex number).

Since magnitude remains fairly the same across PRBGs, magnitude of the reference SRS value can be used as magnitude of SRS values corresponding to all other PRBGs. With the phase and magnitude obtained, SRS values corresponding to all PRBGs of UE TX - eNodeB RX antenna pair can be generated.

Since change in phase of SRS values when moving between columns or rows in the antenna array is nearly uniform, the same approach can be extended to antenna array as well. If the change in phase of SRS values corresponding to adjacent columns and adjacent rows of a PRBG is found, phases of all SRS values of the antenna array of the PRBG can be calculated using phase of a reference SRS value. Since magnitude of SRS values remains fairly the same

across the antenna array, magnitude of the reference SRS value can be used obtain the magnitude of SRS values corresponding all antennas in the antenna array.

However, this type of relationship between PRBGs or elements of the antenna array exist only in case of LOS UEs and not in case of NLOS UE (as shown in section 3.2). Hence, this chapter of the thesis focuses on implementation of lossy compression of SRS based channel estimates of LOS UEs and is described in subsequent sections.

This approach aims at storing only a reference SRS value (complex number) for each polarization and change in phase of SRS values corresponding to PRBGs and columns and rows of antenna array to generate all the SRS based channel estimates corresponding to 50 PRBGs and 128 beams. Since there are antennas of two polarizations at eNodeB, a reference value corresponding to each polarization will be stored in this thesis.

Even though, change in phase of SRS values between adjacent PRBGs (when moving across PRBGs) or between adjacent columns or adjacent rows (when moving across the antenna array) is similar, it is not the same. This approach thus requires finding an optimal change (separately for PRBGs, columns and rows) in phase that lies closest to changes between all pairs of adjacent PRBGs, columns and rows. We propose to use linear regression to help find this optimal change in phase.

## *5.1 Linear Regression*

A statistical model is defined as a simple description of a state or process in [12]. Regression analysis is a statistical technique for analyzing and modeling the relationship between variables. Regression has a wide range of applications and used in numerous fields like engineering, physical and chemical sciences, life and biological sciences, management, economics, social sciences and so on [13].

There are three types of regression [12] and [13]:

i. Simple linear regression:

Simple linear regression is used for modeling linear relationship between two variables, where one of them is the predictor or regressor variable and the other is the response variable [13]. More details are provided in section 5.3.

ii. Multiple linear regression

Multiple linear regression is a linear regression model with one response variable and more than one regressor variable [13]. The general form of multiple linear regression is shown below:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \qquad (5.1)$$

where y is the response variable, $x_1, x_2, \ldots, x_k$ are the regressor variables, $\beta_0, \beta_1, \ldots, \beta_k$ are the regression coefficients and $\varepsilon$ is the regression error. The amount by which an observation differs from its expected value is defined as regression error [12]. It is assumed that $\varepsilon$ is normally distributed with $E(\varepsilon) = 0$ and a constant variance $\text{Var}(\varepsilon) = \sigma^2$.

iii. Nonlinear regression

Nonlinear regression is a regression model in which relationship between response variable and regressor variable is not linear in regression parameters [12]. A nonlinear regression model is intricate with respect to estimation of model parameters, selecting the model, selection of variables, diagnosis of model and so on.

## *5.2 Goals of regression analysis*

Goals of regression analysis include [12]:

- Establish causal relationship between response variable and regressor variables $x_1, x_2, \ldots, x_k$.
- Predict the value of y based on values of $x_1, x_2, \ldots, x_k$.
- Analyze variables $x_1, x_2, \ldots, x_k$ to identify which of them have more importance than the others with respect to response variable y. This

would help in determining causal relationship more accurately and efficiently.

## 5.3 Simple linear regression

As mentioned in the previous section, simple linear regression is used for modeling linear relationship between regressor and response variables. A simple regression model can be represented as [13]:

$$y = \beta_0 + \beta_1 x + \varepsilon \qquad (5.2)$$

where y is the response variable, $\beta_0$ is y intercept, $\beta_1$ is the slope or gradient of regression line, x is the regressor variable and $\varepsilon$ is the error term. In simple linear regression, it is assumed that $\varepsilon$ is normally distributed with $E(\varepsilon) = 0$ and a constant variance $\text{Var}(\varepsilon) = \sigma^2$ [12]. It is also assumed that the errors are uncorrelated, that is value of an error does not depend on value of any other error [13].

A regression model can also be represented as [12]:

$$y = E(y) + \varepsilon \qquad (5.3)$$

where $E$(y) is the mathematical expectation of response variable.

An example of simple linear regression is shown in figure 5.1 [13]:

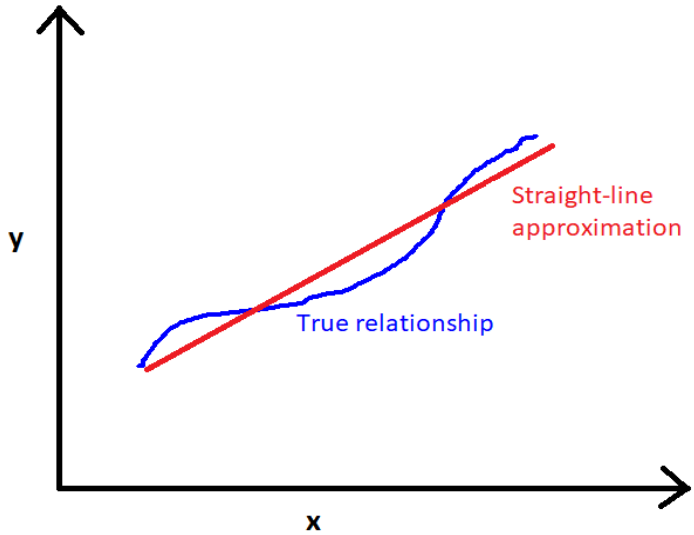


**Figure 5.1: Simple linear regression approximation of a complex relationship**

The slope $\beta_1$ is the change in mean of y for a unit change in x. Parameters $\beta_0$ and $\beta_1$ are called as regression coefficients [13].

In this thesis, we propose to use simple linear regression to model the phases of SRS values corresponding to different PRBGs, columns or rows of antenna array to a straight-line as per the approach shown in figure 5.1.

The next step after modeling is to find good estimates of $\beta_0$ and $\beta_1$ for simple regression model that can best describe the data [12]. We propose to use the method of least squares to estimate $\beta_0$ and $\beta_1$.

## *5.4 Least Squares Estimation*

The method of least squares can be used to obtain estimates of $\beta_0$ and $\beta_1$. Least squares principle involves finding estimates $\beta_0$ and $\beta_1$ such that sum of squared distances between actual observations and modeled straight-line is minimum [13].

In an experiment modelled using simple linear regression, there are n pairs of data $(x_1,y_1)$, $(x_2,y_2),\ldots,(x_n,y_n)$. A model can be represented in terms of n pairs of data as [12]:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad \text{for i= 1,2,…,n} \qquad (5.4)$$

Least squares estimation aims to find the line that is closest to all data points $(x_i,y_i)$. The least square estimates $\widehat{\beta_0}$ and $\widehat{\beta_1}$ are the estimates that satisfy the condition [12]:

$$(\widehat{\beta_0},\widehat{\beta_1}) = \arg\min_{(\beta_0,\beta_1)} \sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_i)]^2 \qquad (5.5)$$

The solution to the above equation 5.5 is obtained by solving the below system [12]:

$$\frac{\partial}{\partial\beta_0}\sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_i)]^2 = 0 \qquad (5.6)$$

$$\frac{\partial}{\partial\beta_1}\sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_i)]^2 = 0 \qquad (5.7)$$

The line derived by obtained estimates of slope and gradient is called fitted regression line [12]. The difference between observed value and corresponding fitted value is called as residual [13].

Regression error is not observable; however, regression residual is observable [12]. Regression residual can be viewed as observable estimate of unobservable error.

## *5.5 Implementation of simple linear regression model by least squares estimation in this thesis*

As mentioned earlier, we propose to use simple linear regression by least squares estimation to model the change in phase between PRBGs and columns and rows of antenna array.

$$(\hat{\beta}_0,\hat{\beta}_1) = \arg\min_{(\beta_0,\beta_1)} \|\boldsymbol{y} - \beta_1\boldsymbol{x} - \boldsymbol{\beta_0}\|^2 \qquad (5.8)$$

where $\boldsymbol{y}$ is vector containing the phases of SRS based channel estimates (corresponding to PRBGs, columns of antenna array or rows of antenna array), $\boldsymbol{x}$ is a vector of corresponding indices of PRBGs, columns or rows of antenna array, $\beta_1$ is the slope of the model, $\beta_0$ is the y-intercept of the model and $\boldsymbol{\beta_0}$ is the product of $\beta_0$ with a vector of ones with same size as $\boldsymbol{x}$ and $\boldsymbol{y}$. $\hat{\beta}_0$ and $\hat{\beta}_1$ are the y-intercept and slope of the model respectively with "closest" fit to observed values.

Solution to the above equation is obtained by solving the system below:

$$\frac{\partial}{\partial \beta_1} \|\boldsymbol{y} - \beta_1 \boldsymbol{x} - \boldsymbol{\beta_0}\|^2 = 0 \quad (5.9)$$

$$\frac{\partial}{\partial \beta_0} \|\boldsymbol{y} - \beta_1 \boldsymbol{x} - \boldsymbol{\beta_0}\|^2 = 0 \quad (5.10)$$

To find closest fit to phases of SRS values across PRBGs, phases of SRS values of all PRBGs of a UE TX – eNodeB RX pair was taken in $\boldsymbol{y}$ and indices of PRBGs in $\boldsymbol{x}$**.** The system described by equations 5.9 and 5.10 was solved in MATLAB to find $\hat{\beta}_0$ and $\hat{\beta}_1$. The process was repeated for all UE TX – eNodeB RX pairs and all $\hat{\beta}_0's$ and $\hat{\beta}_1's$ were found and the mean value of $\hat{\beta}_0's$ and $\hat{\beta}_1's$ were found.

$$\hat{\beta}_{0_{PRBG}} = mean\left(\hat{\beta}_0's\ of\ PRBGs\ corresponding\ to\ all\ UE\ TX - eNodeB\ RX\ antenna\ pairs\right) \quad (5.11)$$

$$\hat{\beta}_{1_{PRBG}} = mean\left(\hat{\beta}_1's\ of\ PRBGs\ corresponding\ to\ all\ UE\ TX - eNodeB\ RX\ antenna\ pairs\right) \quad (5.12)$$

The phases of the SRS based estimates were recreated using the straight line defined by y-intercept $\hat{\beta}_{0_{PRBG}}$ and slope $\hat{\beta}_{1_{PRBG}}$. Figure 5.2 shows a comparison between recreated phases and actual phases of SRS based channel estimates corresponding to all PRBGs of a UE TX – eNodeB RX pair.

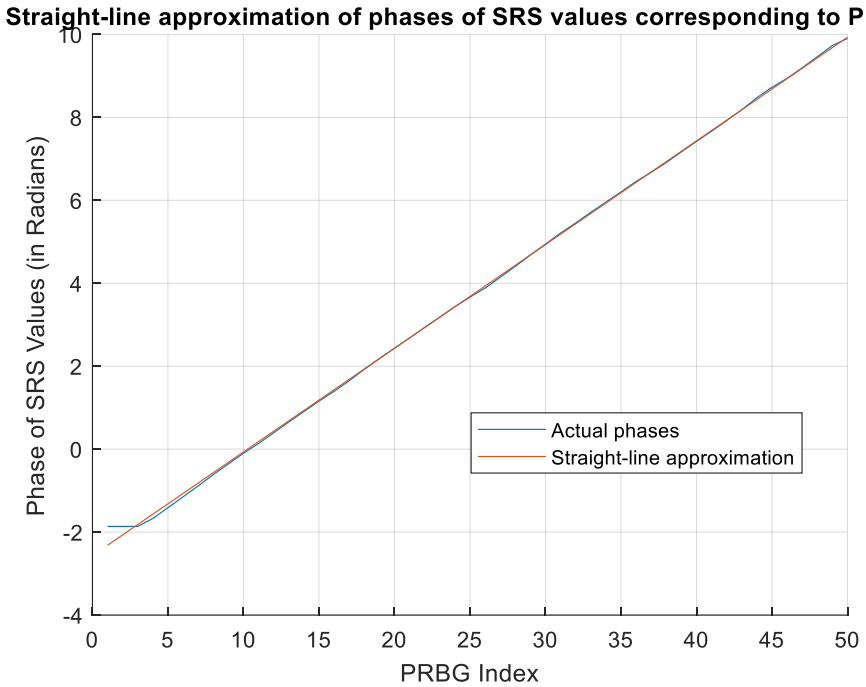


**Figure 5.2: Straight-line approximation of phases of SRS values corresponding to PRBGs**

Similarly phases of SRS values corresponding to all columns of one row of the antenna array at eNodeB corresponding to a PRBG was taken in $\boldsymbol{y}$ and corresponding column indices in $\boldsymbol{x}$ and solved for $\hat{\beta}_0$ and $\hat{\beta}_1$. The process was repeated to all rows and all PRBGs and mean of $\hat{\beta}_0's$ and $\hat{\beta}_1's$ were calculated.

$$\hat{\beta}_{0_{Column}} = mean\left(\hat{\beta}_0's\ of\ columns\ corresponding\ to\ all\ rows\ and\ PRBGs\right) \quad (5.13)$$

$$\hat{\beta}_{1_{Column}} = mean\left(\hat{\beta}_1's\ of\ columns\ corresponding\ to\ all\ rows\ and\ PRBGs\right) \quad (5.14)$$

Phases of SRS based estimates corresponding to columns of the antenna array were recreated as in the case of PRBGs. This was compared with actual phases and is shown in figure 5.3:

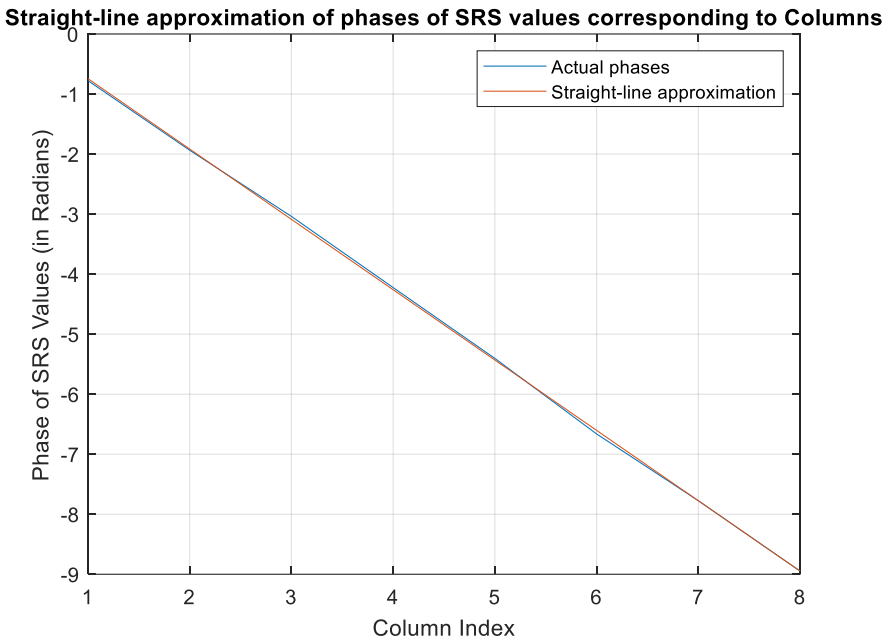


**Figure 5.3: Straight-line approximation of phases of SRS values corresponding to columns of antenna array**

Likewise phases of SRS values corresponding to all rows of one column of the antenna array at eNodeB corresponding to a PRBG was taken in $\boldsymbol{y}$ and corresponding row indices in $\boldsymbol{x}$ and solved for $\hat{\beta}_0$ and $\hat{\beta}_1$. The process was repeated to all columns and all PRBGs and mean of $\hat{\beta}_0's$ and $\hat{\beta}_1's$ were calculated.

$$\hat{\beta}_{0_{Row}} = mean\left(\hat{\beta}_0's\ of\ rows\ corresponding\ to\ all\ columns\ and\ PRBGs\right) \quad (5.15)$$

$$\hat{\beta}_{1_{Row}} = mean\left(\hat{\beta}_1 's\ of\ rows\ corresponding\ to\ all\ columns\ and\ PRBGs\right) \quad (5.16)$$

Phases of SRS based estimates corresponding to rows of the antenna array were recreated as in the cases of PRBGs and columns. This was compared with actual phases and is shown in figure 5.4:

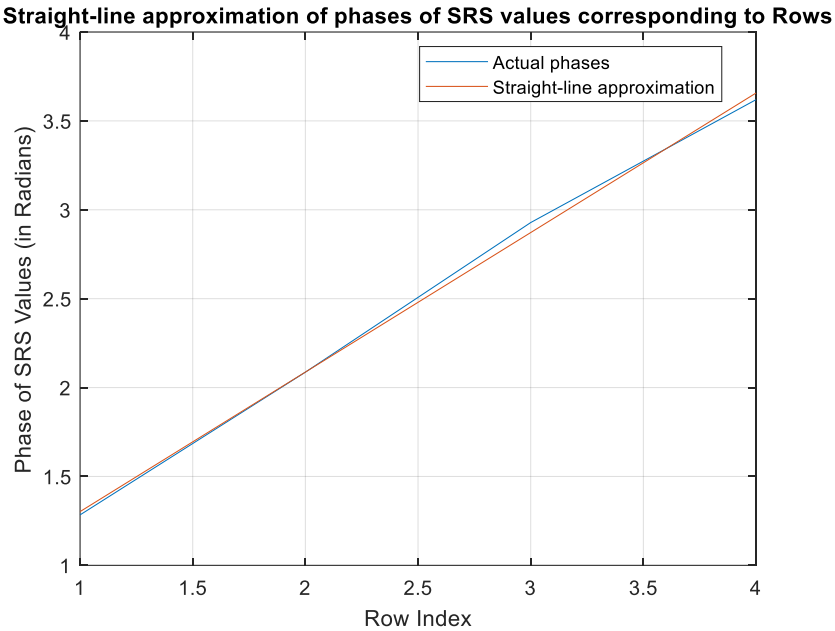


**Figure 5.4: Straight-line approximation of phases of SRS values corresponding to rows of antenna array**

There are two polarizations of antennas at the eNodeB as well as the UE's. As a result, there are four possible combinations of polarizations of UE and eNodeB antennas. To generate SRS values, a reference value (an SRS value which is a complex number) is chosen from each of the four possible combinations. We decided to select the reference value as the SRS value corresponding to first PRBG, first column and first row of a combination. All the SRS values corresponding to remaining PRBGs, columns and rows of that combination can be generated using equation 5.17:

$$SRS_{Value}(x,y,z) = |ReferenceSRS_{Value})| \times e^{jx\hat{\beta}_{1_{PRBG}}} \times e^{jy\hat{\beta}_{1_{Column}}} \times e^{jz\hat{\beta}_{1_{Row}}} \times e^{j\hat{\beta}_{0_{PRBG}}} \quad (5.17)$$

where $SRS_{Value}(x,y,z)$ is the SRS value that needs to be reconstructed in which x, y and z denote the index of PRBG, column and row respectively and $|ReferenceSRS_{Value})|$ is the magnitude of reference SRS value.

So by storing a reference value and $\hat{\beta}_{0_{PRBG}}$ corresponding to each possible combination of UE and eNodeB antenna polarizations, $\hat{\beta}_{1_{PRBG}}$, $\hat{\beta}_{1_{Column}}$ and $\hat{\beta}_{1_{Row}}$, all 6400 SRS values of an occasion (corresponding to 2 UE antennas, 64 antennas at eNodeB and 50 PRBGs) can be reconstructed. This results in extremely high compression and space savings achieved is shown below:

$$CSI\ size\ corresponding\ to\ one\ occasion\ of\ a\ UE = 25600\ bytes\ (section\ 1.1)$$

$$Compressed\ CSI\ Size = 4 \times reference\ SRS\ values\ (4 \times 32\ bits) + slopes\ corresponding\ to\ PRBGs, columns\ and\ rows\ (3 \times 16\ bits) + offset\ corresponding\ to\ PRBGs\ (4 \times 16\ bit) = 30\ bytes$$

$$Space\ savings\ (calculated\ as\ per\ equation\ 3.6) = 99.8828\ \%$$

However, these reconstructed SRS values have some errors compared to the original SRS values. The errors in reconstructed SRS values are due to two reasons:

1. The magnitude of SRS values is assumed to be the same as that of reference SRS value.
2. The straight-line approximation of phases using linear regression by least squares estimation produces regression residual as explained in Section 5.4.

The next task is to evaluate degradation in performance as a result of loss in information due to the compression technique employed. We designed a system in MATLAB for this purpose and details are provided in section 5.6.

## 5.6 Bit Error Rate (BER) Simulator

In this part of the thesis, there are two UEs with one antenna each and an eNodeB with 64 antennas. Hence, there are 64 antennas at RX side and 2 antennas at TX side, resulting in a 64 x 2 MIMO scenario. A system was designed in MATLAB with the following specifications:

$10^6$ random bits were generated, where each UE transmits $5x10^5$ bits and the sequences are independent. The bits were mapped to Quadrature Phase Shift Keying (QPSK) symbols and transmitted through the TX antennas of two UEs. The transmit signal vector is represented by $\boldsymbol{s}$. Channel matrix, $\boldsymbol{H}$, is a $N_r$ x $N_t$ Matrix where $N_r$ and $N_t$ are number of receiver and transmitter antennas respectively [4]. Hence in this scenario, $\boldsymbol{H}$ is a 64 x 2 matrix.

Received signal vector $\boldsymbol{r}$ is defined as [4]:

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{n} \tag{5.18}$$

where $\boldsymbol{n}$ is the noise vector. Noise vector $\boldsymbol{n}$ was generated using MATLAB function 'randn' to generate a vector of complex numbers and normalized by multiplying with a factor of $1/\sqrt{SNR\ (linear)}$.

SNR (linear) refers to the signal to noise ratio in linear scale at RX antenna port, that is, before combining the signals from 64 antennas at eNodeB. This SNR was used in plotting BER versus SNR in figure 5.5. SNR at detector stage will be higher (by approximately 18 dB), owing to the array gain.

There are 50 realizations of the channel corresponding to 50 PRBGs and this is denoted by N. Channel matrix $\boldsymbol{H}$ is normalized as shown below [7]:

$$\boldsymbol{H_{norm}} = \boldsymbol{H} \times \left[\frac{1}{N \times N_r \times N_t} \sum_{n=1}^{N} \|\boldsymbol{H}_n\|_F^2\right]^{-1/2} \tag{5.19}$$

where $\|\boldsymbol{H}_n\|_F^2$ is the squared Frobenius norm of $n^{th}$ realization of channel matrix $\boldsymbol{H}$. Squared Frobenius norm of H is defined as [7]:

$$\|\boldsymbol{H}\|_F^2 = Tr(\boldsymbol{H}\boldsymbol{H}^H) \tag{5.20}$$

where Tr() is the trace operator and $\boldsymbol{H}^H$ is the Hermitian transpose of $\boldsymbol{H}$ matrix. $\|\boldsymbol{H}\|_F^2$ can be interpreted as the total power gain of the channel [7].

SRS based channel estimates from two UEs were stored in $\boldsymbol{H}$ and then normalized as per equation 5.19 to obtain $\boldsymbol{H_{norm}}$, which was then used as the channel matrix described in equation 5.18.

Next task was to design a receiver for this MIMO system. We designed a Zero Forcing (ZF) receiver. A ZF equalizer compensates for channel response by using an inverse filter and it can be defined as [8]:

$$\boldsymbol{G}_{ZF} = \sqrt{\frac{N_t}{E_s}} \times \boldsymbol{H}^{\dagger} \tag{5.21}$$

where $E_s$ is the symbol energy and $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose pseudo inverse. $\boldsymbol{H}^{\dagger}$ is defined as [8]:

$$\boldsymbol{H}^{\dagger} = (\boldsymbol{H}^{H}\boldsymbol{H})^{-1}\boldsymbol{H}^{H} \tag{5.22}$$

The output of the ZF filter is only a function of symbol vector to be detected and the noise, and the output is given by [8]:

$$\boldsymbol{z} = \boldsymbol{G}_{ZF}\boldsymbol{r} = \boldsymbol{s} + \boldsymbol{G}_{ZF}\boldsymbol{n} \tag{5.23}$$

QPSK demodulation was then implemented on the received symbol vector to obtain corresponding bits. Bit Error Rate (BER) was calculated as shown below:

$$BER = \frac{Number\ of\ bits\ received\ incorrectly}{Total\ number\ of\ bits\ transmitted} \tag{5.24}$$

This process was performed for SNRs between -30 dB to -10 dB in steps of 5 dB. BER was calculated in each case and plotted against SNR.

To evaluate the performance of lossy compression algorithm we implemented in this thesis, another ZF equalizer was designed using the reconstructed channel matrix in place of the original one and the steps repeated. BER corresponding to different SNRs was plotted in this case as well and compared it with ZF equalizer using original channel. BER versus SNR plot for the two cases is shown in figure 5.5:

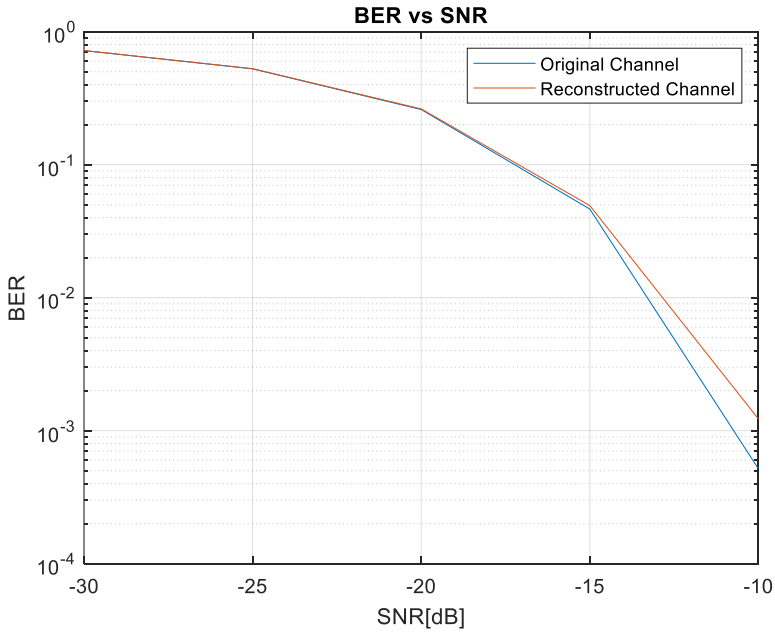


**Figure 5.5: Comparison of BER versus SNR plots when using original channel and reconstructed channel**

As can be seen in figure 5.5, BER in case of reconstructed channel after lossy compression is higher than that of original channel. Even though, there is a slight degradation in performance as per this Simulator, an extremely high space savings of 99.8828 % can be achieved by using this approach.

# CHAPTER 6

## *Conclusion*

We investigated the redundancy in SRS based channel estimates corresponding to PRBGs and elements in the antenna array in an SRS occasion for LOS and NLOS scenarios.

We analyzed various lossless compression algorithms and compression formats based on their compression ratio, time required for compression and decompression and complexity of implementation. MiniLZO was chosen as it provided lowest compression and decompression times and was relatively easier to implement.

Source code of MiniLZO was obtained, analyzed and modified to suit Ericsson's DSP architecture. Mini LZO also offered space savings of 22.4844 % for UE1 in LOS, 21.7187 % for UE2 in LOS and 26.8750 % for UE in NLOS.

Lossy compression of SRS based channel estimates was implemented using linear regression by least squares estimation. This approach helps in achieving extremely high space savings (99.8828 %) with a small degradation in performance.

Core essence of this invention is to compress SRS based channel estimates, so that lesser space is occupied by the channel estimate data in the shared memory of the eNodeB. Benefit from this invention is that more space will be available in the shared memory after compression which can facilitate higher capacity for reciprocity-based beamforming. Hence this will prolong the lifetime of the existing hardware and improve efficiency. This may also lead to higher cost optimization.

# CHAPTER 7

## *Future work*

Mini LZO was chosen as the compression format for lossless compression as the time required to implement compression and decompression was least when simulated on Linux based server, compared to other formats – BZIP2, LZMA, GZIP and ZIP. However, the process of compressing or decompressing the data using these four compression formats could be faster or slower when implemented on the DSPs. Nevertheless, implementing all these compression formats in DSP is a time-consuming task as the source code of each compression format must be analyzed and modified according to the DSP's architecture at Ericsson. The implementation of the other formats on Ericsson's DSP and analysis of their performance could be considered as future work.

Lossy compression of SRS based channel estimates using linear regression by least squares estimation has been described in Chapter 5. However, this technique will yield a set of reconstructed data that is close to original data only when there is high correlation in magnitude and phase both between elements of antenna array and between PRBGs. This typically happens in LOS scenario. Extension of a similar approach to NLOS scenario is considered as a future work.

## *References*


[1] Johnson, C. (2012) *Long Term Evolution IN BULLETS.* 2nd edition. Northampton: Johnson.

[2] Dahlman, E. and Parkvall, S. and Sköld, J. (2011) *4G LTE/LTE-Advanced for Mobile Broadband.* 1st edition. Oxford: Elsevier Ltd.

[3] Dahlman, E. and Parkvall, S. and Sköld, J. and Beming, P. (2008) *3G EVOLUTION HSPA and LTE for Mobile BroadbandLTE/LTE-Advanced for Mobile Broadband.* 2nd edition. Oxford: Elsevier Ltd.

[4] Andreas F.Molisch (2011) Wireless Communications 2nd edition. Wiley Publishing ISBN: 0470741864 9780470741863

[5] Erik Dahlman Stefan Parkvall Johan Skold (2018) 5G NR: The Next Generation Wireless Access Technology 1st edition, Academic Press Elsevier.

[6] Christopher Cox (2012), An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications 1st Edition, Wiley.

[7] Arogyaswami Paulraj, Rohit Nabar, Dhananjay Gore (2008) Introduction to Space-Time wireless Communications, Cambridge University Press.

[8] Bruno Clerckx, Claude Oestges (2013), Mimo Wireless Networks: Channels, Techniques and Standards for Multi-Antenna, Multi-User, Multi-Cell 2nd edition Academic Press Elsevier.

[9] Khalid Sayood (2012), Introduction to Data Compression 4th edition Morgan kaufmann Elsevier.

[10] Saloman, David (2007), Data Compression 4th edition Springer.

[11] Komal Sharma, Kunal Gupta(2017), Lossless data compression techniques and their performance, 2017 International Conference on Computing, Communication and Automation (ICCCA).

[12] Xin Yan, Xiao Gang Su, Linear Regression Analysis: Theory and Computing, World Scientific

[13] Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining (2012), Introduction to Linear Regression Analysis 5th Edition, Wiley

[14] https://www.itu.int/osg/spu/imt-2000/technology.html

[15] Mark Nelson, Jean-Loup Gailly (1996), The Data Compression Book 2nd Edition Press New York, NY, USA.

[16] Stefan Höst (2017), Information Theory and Communication Engineering, compendium, Department of Electrical and Information Technology, Faculty of Engineering, LTH, Lund University.

[17] http://www.oberhumer.com/opensource/lzo/

[18] http://www.bzip.org/

[19] Simon O. Haykin , Michael Moher (2005), Modern Wireless Communication International Edition, Pearson.

[20] A K Sharma (2005), Textbook of correlations and regression.

[21]https://www.eit.lth.se/fileadmin/eit/courses/eitn21/Lecture5_OFDMSynch.pdf

[22]https://pdfs.semanticscholar.org/4253/7898a836d0384c6689a3c098b823309ab723.pdf

[23] https://linuxize.com/post/linux-time-command/