Johannes Wågen

Implementing downlink channel measurements in the ReRaNP massive MIMO test platform

Master's thesis in Electronic Systems Design

Supervisor: Nils Torbjørn Ekman

July 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis describes how the labview software ReRaNP radio test platform been expanded to support channel measurements of the downlink channel of a multi-user massive MIMO system. The added functionality has been verified by making several channel measurements, and comparing them to some of their expected properties. It will be shown that the measurements corresponds well with the theory. The transfer and storage of these channel measurements have also been implemented. The details necessary to read the stored data are also described in detail. The work presented in this thesis is a continuation of the work presented in the project report on the same subject.

Sammendrag

Denne masteroppgave rapporten beskriver hvordan Labview programvaren til ReRaNP test platformen har blitt utvidet til å kunne måle og lagre nedlink kanalen i et flerbruker massive MIMO system. Funksjonaliteten som er lagt til er verifisert ved å sammenligne egenskapene til målingene med det som er forventet utfra teorien. Det vill bli vist at målingene stemmer godt overens med teorien. Hvordan disse målingene er overføringen og lagret or også dokumentert her. De nødvendige detaljene som trengs for å lese og interagere med systemet er dokumentert. Arbeidet som presenteres her er er fortsettelse på en prosjektoppgave med samme tema.

Table of Contents

List of Tables

List of Figures

Acronyms

DTP Data Transfer Protocol. 18

eNB eNodeB - Evolved Node B - (LTE speak for part of BS that connects to UE). 6

FFT Fast Fourier Transform. 9, 10

FPGA Field programmable gate array. 7, 8, 14

IFFT Inverse Fast Fourier Transform. 10

LTE Long-Term Evolution. 8

MIMO Multiple Inputs Multiple Outputs. 27

MMSE Minimum Mean Squared Error. 6

MRC Maximum Ratio Combining. 6

NI National Instruments. 12, 20

OFDM Orthogonal Frequency Division Modulation. 10

PPS Pulses Per Second. 7

ReRaNP Reconfigurable Radio Network Platform. 1, 6

SDR Software Defined Array. 7

SNR Signal to Noise Ratio. 6, 10

UDP User Datagram Protocol. 19

ZF Zero Forcing. 6

Glossary

LabView Graphical data flow programming environment from National Instruments.. 8

massive MIMO The use of multiple antennas along with a measured channel estimate to precode signal to archive spacial diversity.. 1

Chapter 1

Introduction

As technology progresses, the need for higher data rates for lower power consumption with higher quality of service requirement is apparent. Traditional single antenna systems are starting to show their limitations, as the available spectrum is a limited shared resource. Many of the limitations can be solved with multi user massive MIMO, where multiple antennas are coordinated in an array.[1] Enabling the system to use channel inversion to transmit data to several users on the same frequency at the same time.

Reconfigurable Radio Network Platform (ReRaNP) is a multi year research project at NTNU aimed at exploring some uses for massive MIMOmimo systems. The project has founding from Research Council of Norway. Some of the areas of research are coastal communications and wireless sensor networks. The hardware of the system consist of a large radio hardware platform with accompanying software that enables it to create an LTE-like multi user massive MIMO system. Nearly all of the software is open to modification, making it possible to adapt the platform for different experiments.

1.1 Problem Description

The task given for this thesis was to expand the functionality of the ReRaNP testbed to allow for measurement and storage of the downlink radio channel measurement data. The existing testbed has functionality for viewing this channel measurement, but not for storing it. The channel measurements are not tagged with timing data either. This makes it hard to correlate the data with measurements done on the base station. In addition the interference channel is inaccessible in the original system. The work in this report aims to solve both these issues, by storing both the intended channel and interference channels, along with the data timestamps. Some channel measurements will be done to demonstrate the functionality of the improved system.

Chapter 2

Theory and system description

2.1 Channel theory

A central property of a channel is the length of the channel. That is if a pulse is sent, how long does it take before the received signal is below the noise floor. This property directly correlates to the channel bandwidth. That is how wide a channel can be and still be considered flat, meaning it only have an amplitude and phase shift. Such a channel will therefore have no frequency dependency.

2.1.1 Channel modeling

Every radio channel is essentially unique, but for the purposes of working with them effectively they are modeled with statistics. The choice of model depend on the environment of the channel. The simplest channel model is a free space model, where there is only an amplitude and phase shift for the entire spectrum. More advanced models such as Rice and Rayleigh[2] takes into account the effects of multipath signals and the effects of fading. These models can further be extended by empirical data such as Okumura[3] and the Hata model[4].

2.1.2 Multipath

A multipath channel is a radio channel where there are multiple routes for the radio signal to take. Where the multiple routes have different length, and thereby different propagation delay. These differently delayed signals will be added together in the receiver, causing the channel to have delay spread. The waves of the different paths will be added in the receiver causing destructive and constructive interference. This will change if the receiver is moving or other parts of the channel are moving. This phenomenon is commonly called fading.

2.1.3 Fading

Due to the physical changes in the channel it is expected to observe fading in a channel. That is, the received power drops for a limited time and the frequency below the average frequency. The cause of fading can be roughly divided into two categories, large scale fading and small scale fading. Fading that occurs over an area of multiple wave lengths. Common causes for this are objects blocking or interfering with the transmitted signal, sharp edges causing diffraction or an object being within the Fresnel zone. Small scale fading on the other hand is mostly caused by multipath interference.[2]

The probability of a fading event can be modeled statically, and should be taken into account when designing a radio system. For simple link budgets a factor called L_{50} can be used. Which is how much lower the power is expected to be in 50 percent of fading instances. The system design needs to take into account what the explicable probability of an outage are, often called quality of service. In a single antenna system it is generally not possible to guarantee 100% quality of service, as the required extra power that is needed in the deepest fading dips will make the system prohibitively expensive.

Fading can affect different frequencies differently, particularly in the case of small scale fading. For a sufficiently narrow range of frequency the bandwidth can be considered frequency independent, also known as a flat channel. This bandwidth is called the coherence bandwidth, B_c and is determined by the rms delay spread, ρ. It can be estimated by:

$$B_c = \frac{1}{k\rho} \tag{2.1}$$

The factor k is dependent on how stringent the requirement for channel flatness? is. For example, the requirement for a frequency correlation above 0.9 k is 50. For a more relaxed frequency correlation requirement of 0.5, k is set to 5. [2]

This is only an estimate as the channel spectrum is determined by random processes. However it can still be useful for designing radio systems.

Fading is also time variable. If the channel can be considered stable within one symbol it is considered slow fading. Such channels can use simpler receivers. The stable channel time can be estimated by

$$T_c \approx \frac{1}{f_m} \tag{2.2}$$

Where f_m is the Doppler spread. As with the other equation this equation is also dealing with a statistical channel and so is only an approximation.

Generally it is desirable to work with channels that are both time and frequency coherent. If the available channel does not have these properties, it can often be divided into frequency and time slots that can be considered coherent. This can greatly simplify the complexity of the receiver.

2.2 Multi User Masive MIMO

Massive MIMO is a technology that utilises a large number of coordinated antennas in a radio system to increase performance.

In a multi user massive MIMO system there are up to K users communicating with a base station with up to M antennas. Some systems can also use multiple antennas per user. This report will only focus on single antenna users, although much of the theory presented here still applies to multi antenna user terminals. Each user will then have M uplink channels and M downlink channels. The uplink channel is called $H^{UL}_{k \to m}$ and the downlink channel is called $H^{DL}_{m \to k}$. The base station has to measure and combine these multiple uplink and downlink? channels in order to create one effective channel to each user.

Before sending or receiving any data the base station has to estimate both the uplink and downlink channel to each user. The base station can easily estimate the $H^{UL}_{k \to m}$ by user k sending a known sequence. The base station can then compare the received signal on each antenna with the expected signal. The base station can however not directly measure the downlink channel, as that would have to be measured on the user terminal. For a time division duplexed system as investigated in this report, the downlink channel does not necessarily have to be measured. The reciprocity property of a radio channel can be used to assume that $H^{UL}_{k \to m}$ and $H^{DL}_{m \to k}$ are the same within a sufficiently short time window.

The received downlink signal, Y_k on one user antenna k at one channel at one time instance is given by[5]

$$Y_k = \sqrt{\beta_k} \sum_{l=1}^{K} \left(\sum_{m=1}^{M} H^{DL}_{m \to k} F_{ml} \right) X_l + e_k \tag{2.3}$$

Where β_k is the large scale fading, $H^{DL}_{m \to k}$ is the downlink channel, F_{ml} is the precoding factor for antenna m on the base station and layer l. The signal intended for user l is given by X_l and the noise is given by e_k. This equation can be simplified by splitting it up into an intended effective channel $\widetilde{H}_{kk}$, and interference from other users on the effective interference channels. This is given by

$$Y_k = \sqrt{\beta_k} \widetilde{H}_{kk} X_k + \sum_{\substack{l=1 \\ l \neq k}}^{K} \widetilde{H}_{kl} X_l + e_k \tag{2.4}$$

The effective channel is given by

$$\widetilde{H}_{kl} = \sum_{m=1}^{M} H_{ml} F_{kl} \tag{2.5}$$

As is evident from Equation 2.5 the effective channel to one user antenna from the base station array is the superposition of all signals at the base station. Note that this is all given in terms of the frequency domain signal. The same could be derived for the time domain.

The precombining algorithm chosen for the system will determine how the precoding factors are calculated. This gives the relation between the intended channel and the interference channel. Some of the most common linear frequency domain precombining algorithms are Maximum Ratio Combining (MRC), Minimum Mean Squared Error (MMSE) and Zero Forcing (ZF).[5]

The MRC algorithm can be represented by

$$F' = H_{DL}^{H} \tag{2.6}$$

Where F' is the non normalized precoding matrix, H_{DL}^{H} is the hermatian transformed downlink channel. This will effectively optimize the system for optimal Signal to Noise Ratio (SNR). From an implementation perspective this algorithm requires very little processing to calculate the precoding factors. In addition the processing can be done on each individual receiver, it does not need to know about the other antennas.

The MMSE algorithm is given by

$$F' = H_{DL}^{H}(H_{DL}H_{DL}^{H} + \sigma^2 I)^{-1} \tag{2.7}$$

where σ is an adjustment factor. Setting this to zero yields a ZF algorithm which optimizes for minimum interference.

2.2.1 Channel hardening

As previously described, multipath channels suffers from small scale fading. This can cause a significant impact in the link budget if a high quality of service is required. With massive MIMO the probability of an outage caused by small scale fading is reduced.[6][5] This can be of particular importance in systems that are power limited, such as wireless sensor networks. As the the maximum required output power the system needs can be lower, thus reducing cost.

2.3 ReRaNP hardware system description

The ReRaNP used in this study consists of two main systems, the base station and the user terminal, sometimes also known as mobile terminal. They play similar roles to the eNodeB - Evolved Node B - (LTE speak for part of BS that connects to UE) (eNB) and user equipment in LTE systems. There can only be one base station operating at one time. Up to 12 user terminals can operate simultaneously. All of the equipment is based on modules and products from National Instruments.

The system exists in two main base station configurations, a 128 antenna system and a 2 antenna system. The smaller system is the smallest supported configuration that the software framework supports. This was set up to enable testing of programs without the uncertainty of hardware errors that the larger system could be prone to. The expected MIMO effects with only two antennas is expected to be small so it can not be used for MIMO channel measurement. It is however a good platform for debugging programs.

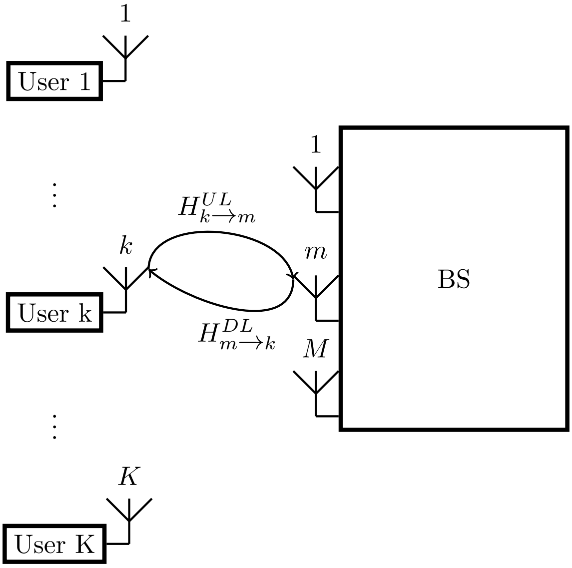


Figure 2.1: The different parts of a massive mimo system

Each user terminal consists of two main parts, an USRP 2953R SDR radio and a host computer. Each SDR radio implement two users. The majority of processing is done on the Field programmable gate array (FPGA) inside the Software Defined Array (SDR). The host is used for setting the system up, feeding it with data, and displaying diagnostics. The host and SDR are connected through a PXIe bus. This is a proprietary extension to PCIe by National Instruments. Since the load on the host is relatively low, all user terminals can be run by one host. They are connected to an MXI switch.[7]

The base station is larger and more complicated than the user terminal. It consists of 64 USRP 2952 SDR radios, multiple FPGA processing modules, and a PXI host computer. All of this is connected through PCIe switches. Synchronization is handled by distributing a Pulses Per Second (PPS) signal, and a 10MHZ signal to all the units. The SDR radio is responsible for modulation and demodulation. Four large FPGA modules are responsible for doing the channel inversion based on the measurement from the sdr radios. The last FPGA processing module does the bit processing, where it reassembles the data packets to their original format.

The USRP 2952 radio that is used in both the transmitter and receiver has two independent radios. Enabling the processing of two antennas or two user terminals. The radio front end has a bandwidth of 40 MHz. After downsampling, the sample rate is 30.72MHz. The processing on the FPGA is done at a faster 290MHz. Enabling the system to use multiple clocks to process a single sample.

The ReRaNP project has several frequency ranges available to be used for testing. These are licensed to the project for use in testing of the system by Ncom, the Norwegian authority on frequency allocation. The available frequencies are

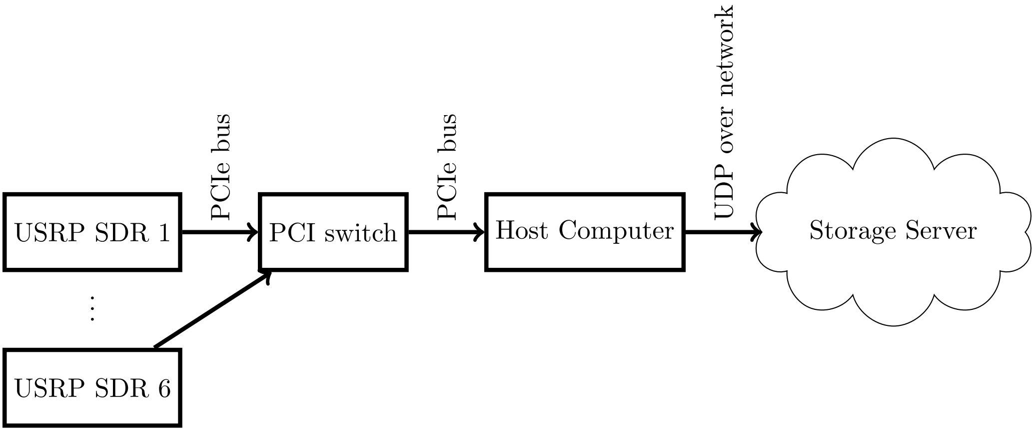


Figure 2.2: User Terminal Data flow

given in Table 2.1. These frequencies cover the range of frequencies encountered in an LTE system.

Start	End	Bandwidth
1452,0 MHz	1492,0 MHz	40 MHz
3800,0 MHz	3840,0 MHz	40 MHz
4140,0 MHz	4180,0 MHz	40 MHz
5905,0 MHz	5925,0 MHz	20 MHz

Table 2.1: Licensed frequencies for ReRaNP

2.4 NI massive MIMO framework

The previously described hardware is programmed in LabView Communications, a graphical programming environment from National Instruments. The NI massive MIMO framework is a complete software package that implements a massive MIMO Long-Term Evolution (LTE) like system on the hardware. Nearly every part of the system can be modified for doing custom experiments or accessing information that is normally hidden in the system.

The user interface allows to configure the system as needed. It provides a lot of debugging information and system performance data. One section of these can bee seen in Figure 2.3 and Figure 2.4.

Labview is a visual programming environment where blocks are connected with wires instead of writing traditional code. The environment is nearly identical between FPGA development and host code development. Although the environments are similar only some simple blocks can be reused in both.

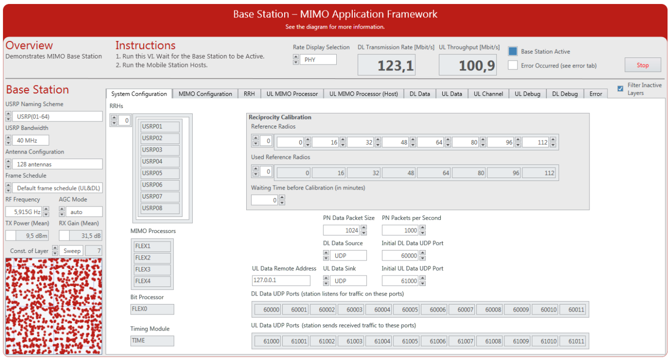


Figure 2.3: Base station user interface

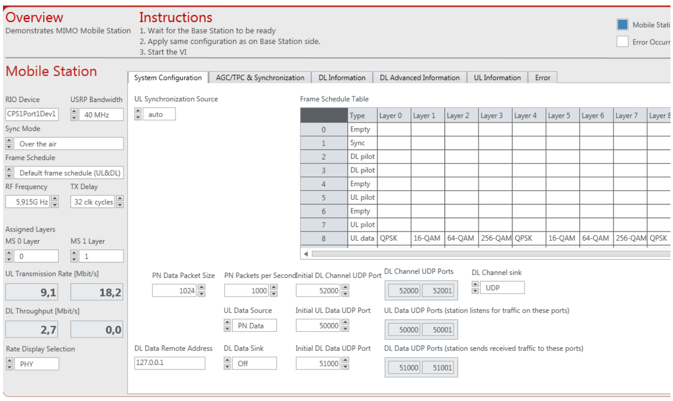


Figure 2.4: User terminal user interface

For this work the receive structure of the user terminal is the most relevant. The larger blocks of the receive structure can be seen in Figure 2.6. It consists of three main blocks, synchronization, IQ processing (both time and frequency domain) and bit processing. Prior to the synchronization there is also a digital front end responsible for receive and transmit switching along with buffering the data to be read for the synchronization block.

The synchronization block compensates for frequency offset between the transmitter and receiver. It then finds the start on the frame based the position of the synchronization pulse in the radio frame. The start of frame is then calculated and passed on to the next block.

The samples are then processed first in time domain, then converted to to frequency domain with an Fast Fourier Transform (FFT). The majority of the processing happens in the frequency domain. Before the fft is done the cyclic prefix is removed from the data. The fft itself takes in 2048 complex numbers in the time domain and outputs 2048 complex numbers in the frequency domain. The guard channels are removed and the remaining 1200 channels are sent further. The symbol type is determined based on the position of the symbol in the radio frame. If it is a pilot sequence the channel is estimated with this. The estimate is used to equalize the channel, and to provide input for the frequency and time tracking blocks. From the channel equalizer the IQ data is ready to be demapped from QAM symbols to bits.

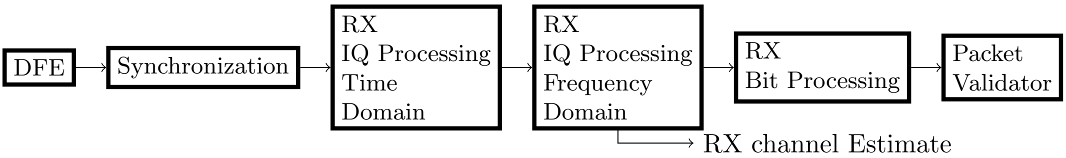


Figure 2.5: Recieve chain block diagram (Recreation of [7, Fig 6-2])

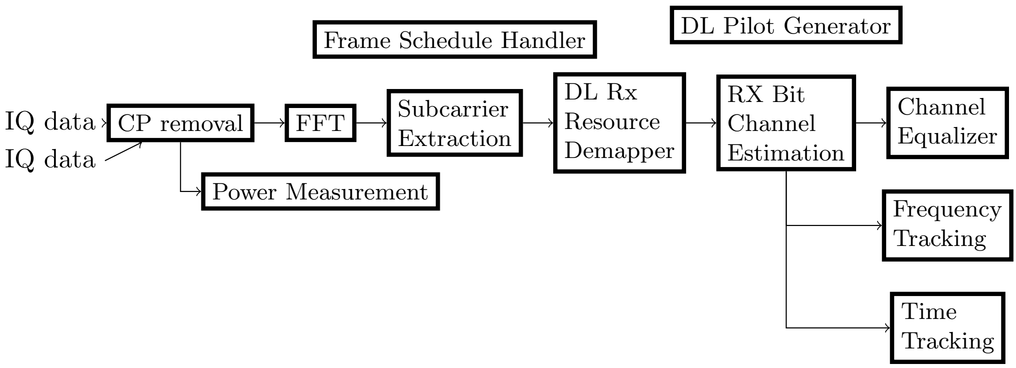


Figure 2.6: Receive IQ processing block (Recreation of [7, Fig 6-4])

2.4.1 Modulation

The system uses Orthogonal Frequency Division Modulation (OFDM) modulation with a bandwidth of 20MHz. In OFDM modulation a FFT and Inverse Fast Fourier Transform (IFFT) is used to split a single wide channel into many narrow band channels. Keeping the channels narrow band will increase the probability that they are flat fading. This in turn greatly simplifies the receiver.

In this system there are 1200 used channels and 848 guard channels, totaling 2048 total channels. Which is a power of two which the fft algorithm is dependant on. Each of the 1200 channels can transmit symbols of varying complexity. From 4 QAM up to 256 QAM. The processing of each channel is independent on the neighbouring channel so each channel can be optimized for the available SNR on that particular channel.

The symbol rate is approximately 14k symbols per second. Before each symbol, some guard bits are sent. These are used as a protection against the time it takes to switch transmission direction. Each OFDM symbol lasts for 66.67 microseconds excluding the guard bits.

The 1200 subchannels are divided into 100 resource blocks with 12 consecutive channels in each. This is done such that one resource block will contain 12 layers. This helps with the implementation, and makes it possible to split up the processing by resource block.

2.4.2 Layers

A massive MIMO system transmits to multiple users simultaneously on orthogonal channels. These orthogonal channels are called layers in the Labview Massive MIMO framework. The system supports up to 12 layers. Each user terminal can use one or multiple layers for transmitting data.

2.4.3 Frame schedule

The individual OFDM symbols are organized into larger structures called frames. The structure of these is based on LTE[7]. Each frame is divided into 10 subframes, which in turn is divided into two slots. Each slot consists of 7 OFDM symbols. Giving a total of 140 symbols per frame. A single frame lasts for 10 ms. As can be seen in Figure 2.7.

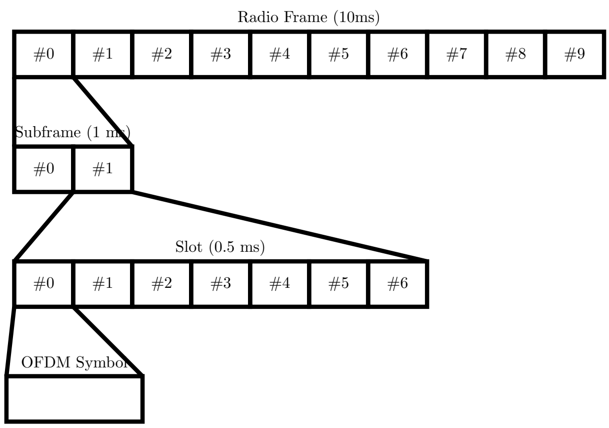


Figure 2.7: Radio Frame Structure

The signal and type of modulation that is sent within a frame is predetermined in this system. Both the base station and user terminal have be set up with the same frame schedule for the system to work. In a commercial product this configuration would have to be shared automatically. The information could be shared on a side channel. The ReRaNP system has a set of predefined frame schedules, but more can be added if needed.

Each symbol within a frame can be chosen to be either, syncronisaiton, uplink pilot, downlink pilot, uplink data or downlink data. For successful communication some requirements must be met by the frame schedule. It is required that each frame contains at least one sync symbol, two consecutive downlink pilots, and one uplink pilot. No data can be sent before the two consecutive downlink pilots are sent. They are needed for phase and frequency tracking. The sync symbol can be placed anywhere in the radio frame, but must only be sent exactly once. When uplink pilots are sent, the precoding is updated. There can be an arbitrary additional number of downlink and uplink pilots in the frame. The higher the density of pilots the more robust the system will be to changes in the channel. The downside to this is of course that there is less space for data in the frame, thereby reducing the effective throughput of the system.

2.4.4 Pilot sequence

The pilot sequence is a sequence of 300 pseudo random qpsk symbols repeated 4 times to get the 1200 samples necessary.[7, p. 7] In the pilot sequences the 1200 channels are mapped to the 12 layers such that layer 1 gets channel 1, 13, 25 and layer 2 gets channel 2, 14, 26 and so on for the remaining 1200 channels as can be seen in Figure 2.8. This means that the channel which is estimated from the pilot sequences is only measured for every 12 channels. This is done to achieve orthogonal channel measurements for each layer in the system. By making the channel measurements orthogonal no further processing in necessary to get the channel between all the antenna pairs. Only the channel estimation uses this system of only using certain channels for each user, the data is transferred on every channel on all layers.

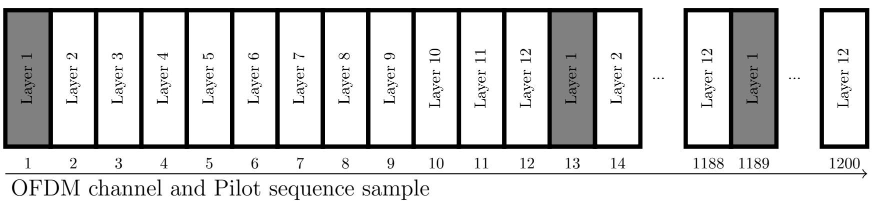


Figure 2.8: Pilot sequence layer mapping

/

The system therefore has to implement interpolation for the channels that have not been measured. On the base station for the uplink pilot the estimate for channel 1 on layer on is simply used for channel 2 to 12 too. This effectively increases the channel bandwidth by a factor 12. On the user terminal linear interpolation is used for the channels in between measurements.

2.4.5 Channel inversion

The National Instruments (NI) massive MIMO application framework implements three different channel inversion algorithms, MRC, ZF and MMSE. The computation of the inverted channel is done on up to four processing fpgas. Each FPGA

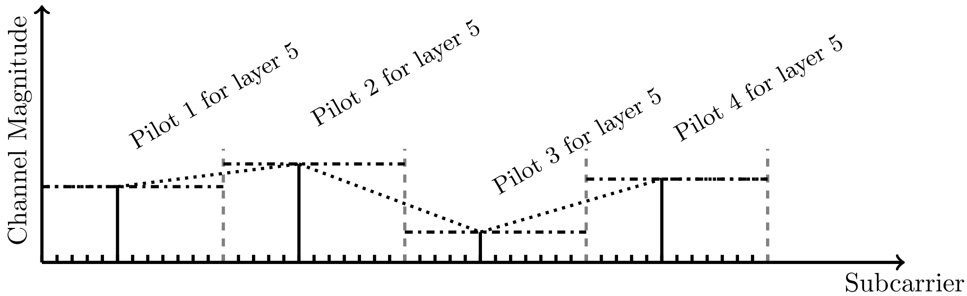


Figure 2.9: Channel Estimation and Interpolation

is able to process the data from 32 antennas. When the system is running with 128 antennas, the workload is split between the FPGA MIMO processors by channel. So that a quarter of the channels from every antenna is sent to each MIMO processor.

2.4.6 Reprocity calibration

Radio channels themselves can be considered are reciprocal in nature. Meaning they display the same properties in both directions. Entire radio system can however not be considered reciprocal. The path a signal takes from the transmitter and receiver is different in each direction, as it goes through a different pair of transmitter and receiver. For some systems this difference can be neglected, but for some massive MIMO they has to be taken into account. In particular in TDD massive MIMO systems, where the channel estimate from the uplink is used to calculate the precombining for the downlink. Any difference between the channel in uplink and downlink would manifest itself as reduced system performance.

The difference in each transmitter and receiver can be calibrated through a reciprocity calibration. This calibration does not need to be done for every sample as the parameters for that is measured in the calibration is relatively stable.

2.4.7 DTP protocol

The system wraps the data to be transferred over the air with a head and a tail. This protocol is called Data Transfer Protocol(DTP). It has a start sequence and end sequence, making it possible to extract a package from a stream of bits. This is done so that each packet can be split over multiple OFDM symbols, allowing for more flexibility for the data to symbol mapping.

2.5 Channel measurements

In order to measure the channel a received signal has to be compared with a known send signal. This could be an impulse on the transmitter. The received signal would then directly be the channel impulse response. This is not practical due to the bandwidth requirement on the transmitter. A better approach is to send a known symbol from the transmitter. The received signal can then be compared with the known sequence to calculate the channel response.

In the ReRaNP system channel measurement is implemented in the downlink for the receiver to be able to track phase and frequency of the receiver. This is all done in the frequency domain after decoding of the OFDM symbol into individual channels.

Each of the received symbol of each OFDM channel is compared to the reference symbol. The difference in phase and amplitude is channel at that particular frequency.

Each layer in the system occupies every twelfth channel. Meaning the channel measurement for the current layer is only available for every twelft OFDM channel.

The measurement is implemented as a complex multiplication between the known sequence and the received signal.

2.6 Channel estimation

A radio channel can be estimated by comparing properties of the received signal with known properties of the sent signal. In the downlink this is done with the base station sending a downlink pilot sequence. For channel n this can be represented as

$$W_n = \frac{Y_n}{P_n} \tag{2.8}$$

Where W is the channel estimate, Y is the received signal and P is the reference pilot signal. Division like this takes longer than multiplication when implemented on an FPGA [8]. It is therefore advantageous to convert this to a multiplication if possible. By using only 4 QAM symbols with an amplitude of one in the pilot sequence, this can be written as

$$W_n = Y_{\frac{P_n^*}{|P_n|^2}} = P_n Y^*. \tag{2.9}$$

Where Y^* is the complex conjugate. This makes the calculation faster and more applicable for use in the system.

The Labview MIMO Framework implements this in order to track frequency offset and time of the downlink. The system implements an estimate for all 1200 channels, even though only 100 is relevant for each user. This is most likely due to the piplelined nature of the FPGA making it simpler to calculate the estimate for all layers, and ignore the non relevant ones.

Chapter 3

Implementation

3.1 FPGA system changes

The FPGA code had to be modified to allow for the channel estimate to be send to the host. There where two main changes that were made, extraction of the channel estimate and sending the channel estimate to the host.

The channel estimate where taken by extending the functionality of some of the existing blocks in the system. The channel estimator orginaly used for time and frequency tracking contained the necessary functinality to do the channel measurement for the all sub channels. The measurement is done in the channel estimator block which is part of the IQ processing block. The channel estimate where routed so it where and output of the IQ processing block. As can be seen in red in **??**.

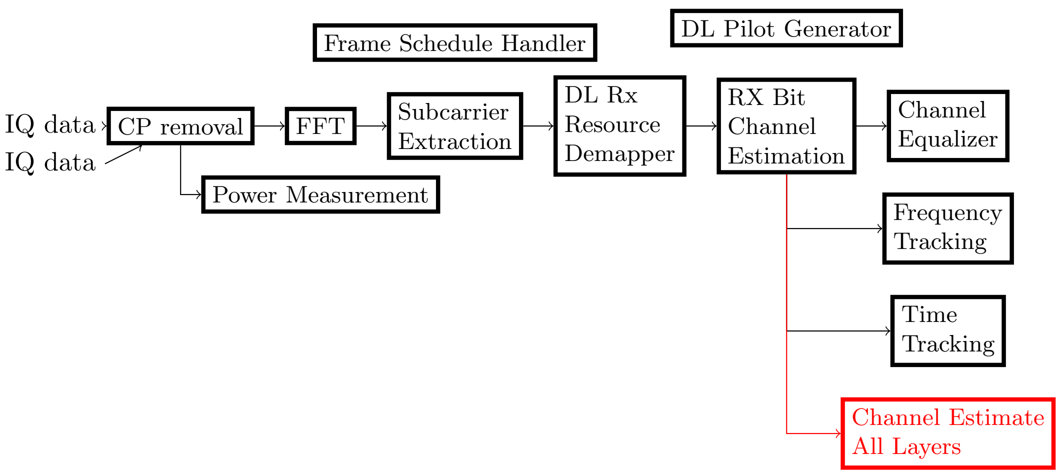


Figure 3.1: Receive IQ processing block (Recreation of [7, Fig 6-4])

3.2 Data packing

3.2.1 Data tagging

The collected channel estimate where tagged with timing data to make it possible to correlate it with other measurements in the future. The timing data where gathered from existing timing information in the system. The symbol index locates the position within a subframe. Its values can be between 0 and 13. To locate a subframe within a whole radiframe a subframe index is used. It can be in the range 0 to 9. In addition a counter that counts the number of recived pilot sequeneces where added. This counter is 64 bit. This size where chosen so overflow would not be a concern for long recordings.

A 64 bit timestamp where also added to support adding a gps timestamp. For the time being the input data to this is not implemented.

3.3 Data transfer

Different methods of transfering the channel estimate between the FPGA and the host computer where explored. The system uses multiple ways to transfer data, depending on the data rate. A worst case for the amount of data to transfer can be calculated based every symbol being a downlink pilot sequence. The system would not function under these conditions but it provides a upper limit for data rate.

$$R_{upper} = N_{sub} \times b_{sub} \times R_{sym} = 1200\text{subcarriers} \times 64\text{bits} \times 1400\text{symbols/s} = 1075 Mb/s \tag{3.1}$$

Where R_{upper} is the absoulute upper rate for data for the system to generate from channel estimates, N_{sub} is the number of OFDM subcarriers in one OFDM symbol, b_{sub} is the number of bits a channel estimate takes for a single subcarrier. The total number of OFDM symbols per second is given by R_{sym}. The spesific data rate is dependant on the particular frame schedule used. A functioning frame schedule has to contain uplink pilots and sync sequenes in addition to downlink pilots. Effectively reducing the required data transfer rate. The default frame schedule has sends a pilot on average about every 8th symbol. This reduces the acctual data rate by a factor of 8 to about 134 Mb/s. This is well within what 1 Gbit networking can handle easily.

Labview provides support for dma buffers for transferring large amount of data between the FPGA and the host. It consists of a buffer that can be written to on the FPGA side. Labview will transfer this data to the host through PCIe packets to the host. For users of this system it acts as a black box where data is pushed in at the FPGA side and it appears at the host side of the system. In reality the actual implementation of this is far more complex.

On the host the data can then be read. If the buffer filled faster than it is read it will overflow. Ideally a overflow would never occur, but in case it does happen the channel measurements should continue to function. A synchronization mechanism in the data packets is therefore required.

The LabVIew framework already implements a system for this called DTP, (Data Transfer Protocol). This protocol is used to transfer and reasemble data that is transfered between the base station and the user terminal. In the DTP the payload is wrapped in a header containing a start syncronisation word, crc, and a length. It is terminated with a different synchronisation word. The syncronisation words makes it possible to extract a payload from an unsyncronised datastream.

The channel estimate and timing information is packaged as show in 3.2. The 4 and 8 bit number are transfered most significant byte first also known as big-endian. The complex numbered channel estimates is in the form Q10.15 complex fixed point numbers. This means there is 10 bits for the integer part for both the real and imagenary part. With 15 bits for the fractional, as can be seen in 3.3. Note that this brings the total number of used bits to 50, which is mapped to a 64 bit number. Meaning there is some wasted capacity in the data transfer. This where done as it simplified the data packaging as it where not necessary to split the data over byte boarders.

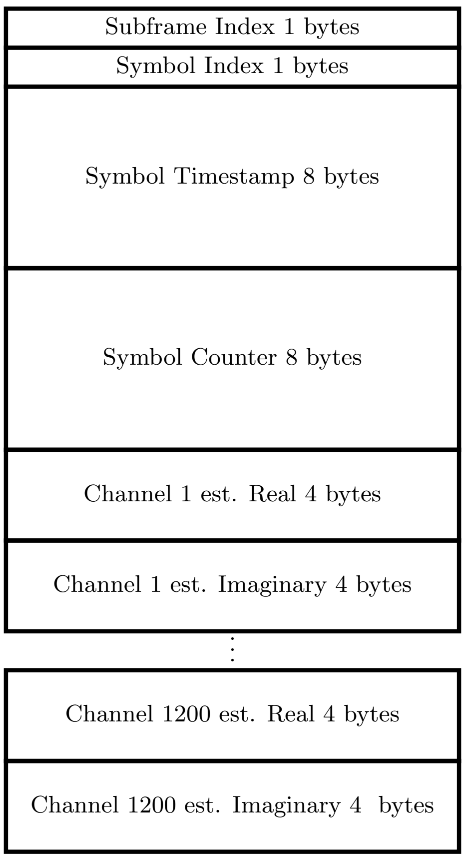


Figure 3.2: Pacet structure of data send over udp

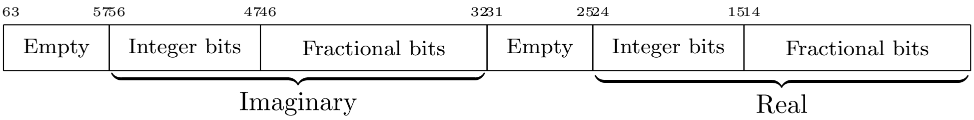


Figure 3.3: Fixed point complex number format

3.3.1 FPGA packet generation

The channel estimate output from the channel estimator where connected to a block that packaged the data into the previously described Data Transfer Protocol (DTP) package with all the required meta data. This block where made specifically for this purpose. It where structured as a state machine where each state where a different field in the package.

The incoming channel estimate where buffered on the fpga before being packaged This where done as the the packaging state machine would only be alerted that there where data ready and it has to store the incoming data. Channel estimate arives at the packaging block as a Q10.25 complex fixed point number at every fourth clock cycle after a downlik pilot is detected. This is effectively 50 bits of data and as the data is sent to the host one byte at the time. There is about two incoming bytes for every outgoing byte. A buffer capable of holding multiple channel estimates where therefore used. An overview of the circuit can be seen in Figure 3.4. The large block on the left is the state machine itself and the buffer can be seen on the right.

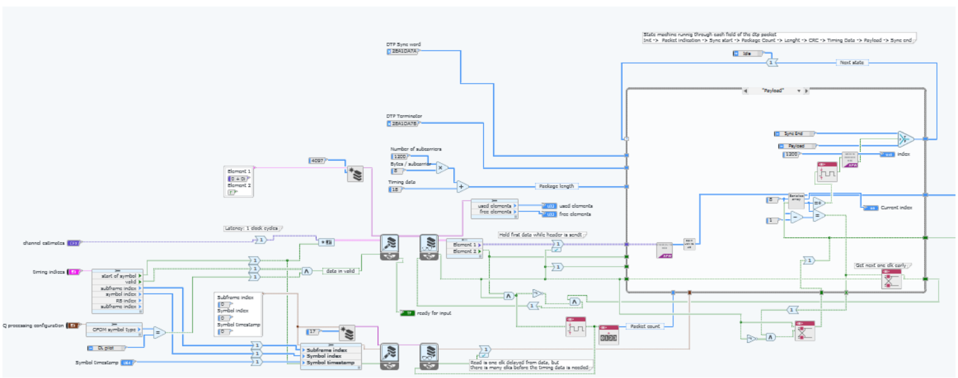


Figure 3.4: The circuit creating the packages on the FPGA

3.3.2 Host data transfer

On the host computer the data path where in large parts copied from the existing functionality used for transferring data packages to the host. It was copied and slightly modified. Some parts that were specific for data packages where removed. Existing code where used to unpack the payload of the DTP packet and send it over the network with the User Datagram Protocol (UDP) protocol.

3.3.3 Reception and storage

A new labview program where created to receive and store the channel estimate data from the UDP packages. The blocks opens a network socket and lisens to it. It will decode the in comming packages and write them to a clear text file. In cases where it is desirable to store the info on a server that is unable to run labview, the same functionality can be implemented in another language. Based on the description provided in 3.2 and 3.3.

The received data is stored as a tab separated file. Each row is one channel estimate. Row corresponds to Figure 3.2. It is further specified in Table 3.1. The channel estes are stored as decimal numbers. This is not the most efficient way of storing data, but it can easely be read by either a script or excel. A more efficiant storage mecanism would be to store the packages in a binary format similar to how they are sent with over the network.

The user interface of the program saving the channel estimates can be seen in Figure 3.5.

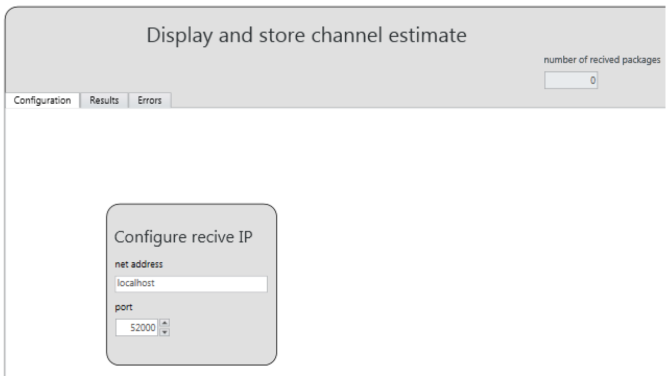


Figure 3.5: User interface of the Capture Packet program

There where also created a small application to view the saved channel estimates. It is able to open the previously mentioned tsv file. In the application it is possible to select a particular channel estimate to inspect further. This program can be used for viewing the channel estimates and it can be used for a starting point if the data is to be further processed using labview.

3.4 Testing

Due to the long compilation times and low visibility for debugging involved with fpga development, testbenches for the most important modules where made. These makes it possible to simulate the functionality of a fpga module. The testbench

Column number	Content
1	Date
2	Time
3	Subframe Index
4	Symbol Index
5	Symbol Timestamp
6	Symbol Counter
7	Channel 1 Estimate Real
8	Channel 1 Estimate Imaginary
9	Channel 2 Estimate Real
10	Channel 2 Estimate Imaginary
	⋮
2406	Channel 1200 Estimate Real
2407	Channel 1200 Estimate Imaginary

Table 3.1: TSV file columns

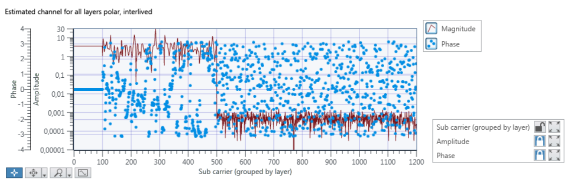


Figure 3.6: Simulation of a channel estimate measurement

provides the FPGA block with the input that it expects, and checks the output for correctness.

The Channel estimate packet block where connected to a channel estimate provided by the channel generator in the link simulaton testbench. This channel estimate where sent into the packaging block. The output where then observed. By using exising DTP blocks from the framework the package where decoded. The output where then compared and would signal a passing test if they are equal. The decoding bolcks used where the same as the ones used for the real system.

The testbench simulating the entire radio signal chain in the user terminal where used to also verify ineroperability with the other modules in the receive chain. This testbench where also used to verify the channel measuring block. Much of this work where done in the project presented in previeus project report on the same subject.[9]. Which was based on work done by NI.

From this previous work some simulations where done. One such simulation can be seen in Figure 3.6. Note that each layer is grouped together and it is not one continuous spectrum.

Chapter 4

Results and discussion

4.1 Measurement Procedure

Before each measurement the settings for frequency and frame schedule where set to the same on both the user terminal and the baste station. The ip address and port for the receiver where added. For small test this where the same as the host so the address where set to `localhost` which the local computer. In addition to this the precoding algorithm where set. The base station where then started. When it had fully started up, the user terminals where enabled. The measurements where done on layer 1. The choice of layer is arbitrary. Three different parameters where varied, frequency, number of users and precombining algorithm. Totaling 12 different configurations. For each configuration more than 1000 channel estimates where saved.

The base station antennas and the user terminal antennas where pointed in the same direction as can be seen in Figure 4.1. The distance to the wall where approximate five meters. This setup where chosen such that the measured channels would have stronger multipath components, and lower line of sight components of the channel.

The different system settings used for measurements can be seen in Table 4.1 As MRC perhaps most relevant for wireless sensor systems [5], to majority of measurements where with MRC precoding.

4.2 Measurement results

The raw channel estimate can bee seen in Figure 4.2. This includes both the intended channel and all the interference channels. The spikes clearly visible in the amplitude plot of every subfigure is the intended channel. As one would expect this has a higher power than the interference channel. The data show in these plots follows the layer division that where shown in Figure 2.9, where every layer uses

Frequency [MHz]	Active Users	Precoding Algorithm
1472	12	MRC
1472	6	MRC
1472	2	MRC
3820	12	MRC
3820	6	MRC
3820	2	MRC
5915	12	MRC
5915	6	MRC
5915	2	MRC
5915	12	MMSE
5915	6	MMSE
5915	2	MMSE

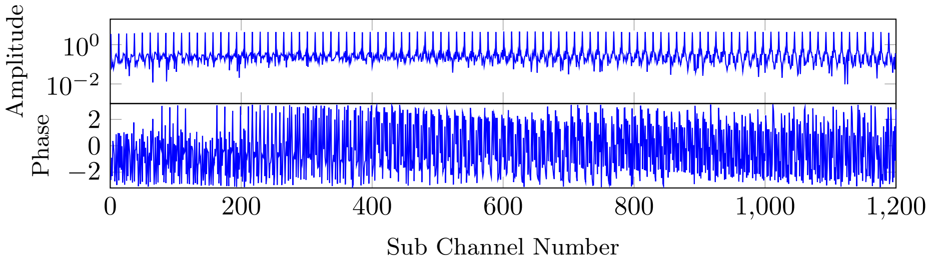
Table 4.1: Measurement configurations

Figure 4.1: Test setup, Base station antennas in background and user terminals in for ground.

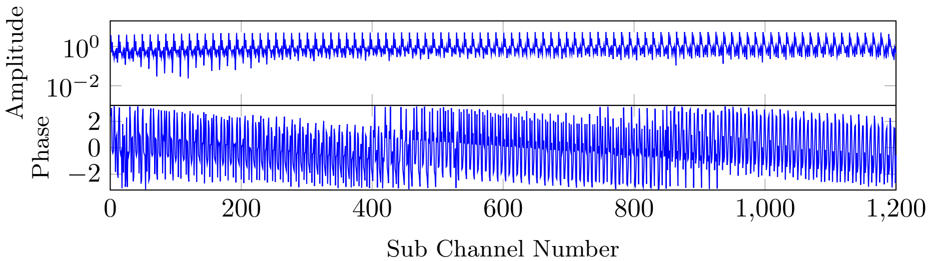
every 12 channel. The radio channel can therefore not be estimated with more than 100 samples per layer.

The frequency response for all the layers is relatively flat and has few deep fading dips. As would be expected from channel hardening the intended channel is more or less completely flat. The interference channels is also reasonably flat. This can passably be caused by the relatively small room with a channel that has low delay spread.

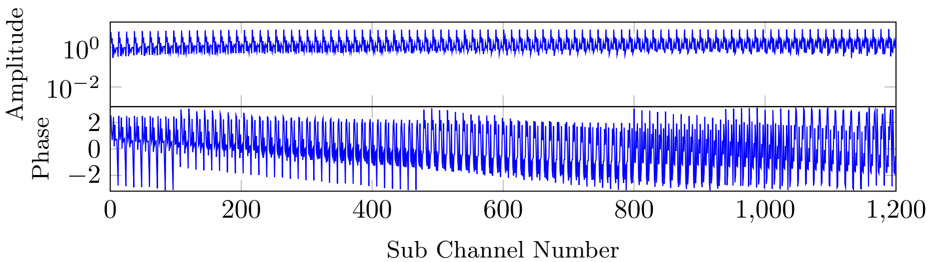
One interference channel and the main intended channel from each measurement with 12 users is shown in Figure 4.3. From this one can see that the intended channel has a higher amplitude and less noisy phase compared to the interference channel. It is worth noting here that this particular interference channel is quite



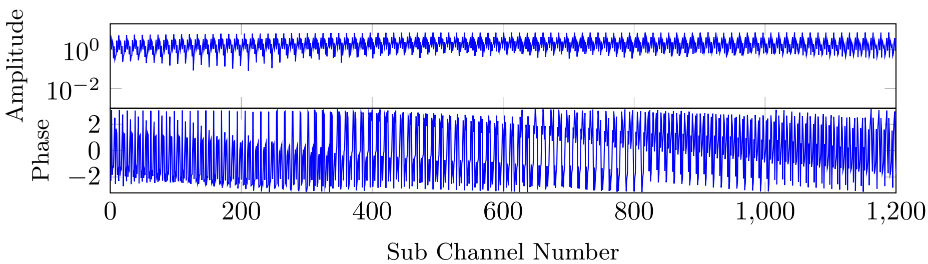
(a) MMSE precoding 12 Users 5915Mhz



(b) MRC precoding 12 Users 5915Mhz



(c) MRC precoding 12 Users 3820Mhz



(d) MRC precoding 12 Users 1472Mhz

Figure 4.2: Entire spectrum, all 1200 channels

similar to the intended channel. The interference can therefore not be simply thought of as extra noise. One possible reason for this is that the length of the channel involved is relatively short, and the user terminal antennas are placed close together. Further measurement and analysts wound have to determine the reason for the channel similarities.

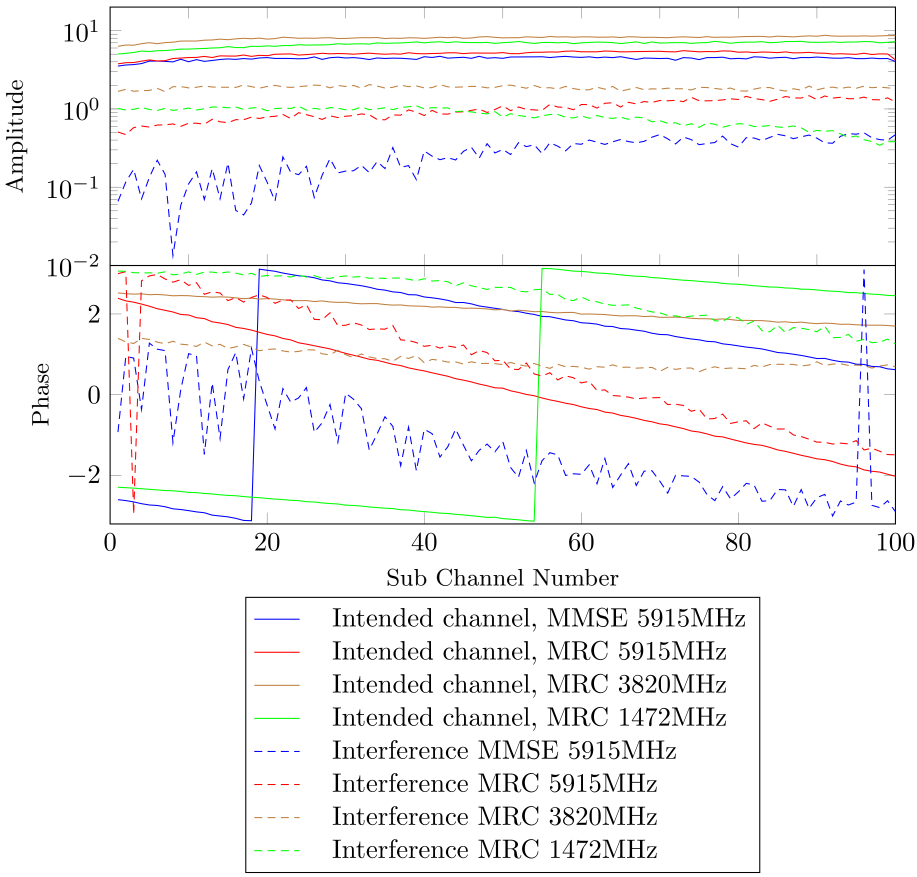


Figure 4.3: Intended and one interference channel at different frequencies and precoding with 12 users

In Figure 4.4 the intended channel and one interference channel is plotted for different number of users. It is clear that the intended channel is between 7 and 10 dB higher than interference channel. Note that the vertical axis is logarithmic. The full scale value is 512, which is the highest value the system can output. One can also see that as the number of users goes down the channel estimate goes up. This is likely due to the limited power of the system is spread across fewer layers. The sudden jump in the phase plot is caused by the phase wrapping around π radians.

Both the intended channel end the interference channel shares the x axis. This is not strictly correct the interference channel is not estimated for the same channel or frequency as the intended channel. This is shown in Figure 2.9. For the porpoises of these plots it is sufficiently accurate as the channels are next to each other. It is also accurate in terms of how the system uses these channel estimates.

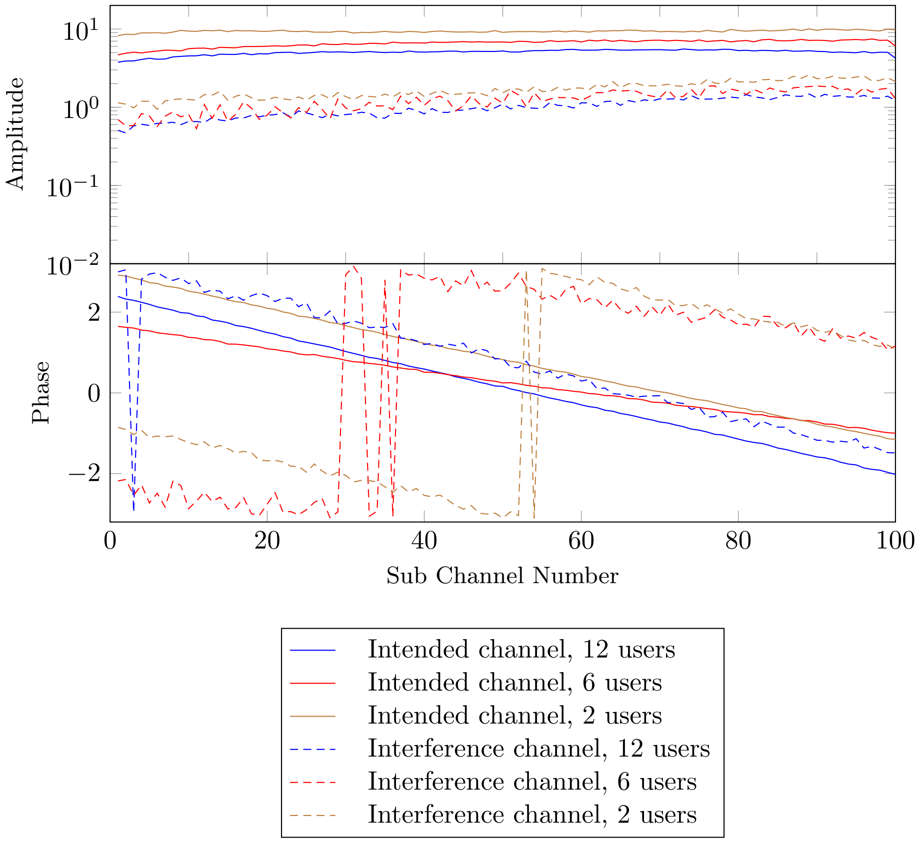


Figure 4.4: Intended and one interference channel at 5915 Mhz with MRC for 2, 6 and 12 users

In the previous figures only one particular interference channel is plotted. In Figure 4.5 the average of all the interference channels is plotted. This is much lower that most of the individual channels, which is expected if the correlation between each interference channel is low. The separation of the intended and the interference is the signal to interference ratio and is about 13dB. It is this average of the interference channels that would be measured as interference when the system is in use and transmitting on all subchannels and layers. Here it is worth noting a difference in the amplitude of the different interference channels of over 17dB.

It almost every measurement the amplitude of the channel estimate is lower at each end of the spectrum. This could be caused the band pass filter used in the radio. However the precombining in the base station should have compensated for it. It can perhaps be caused by an insufficient or inaccurate reciprocity calibration.

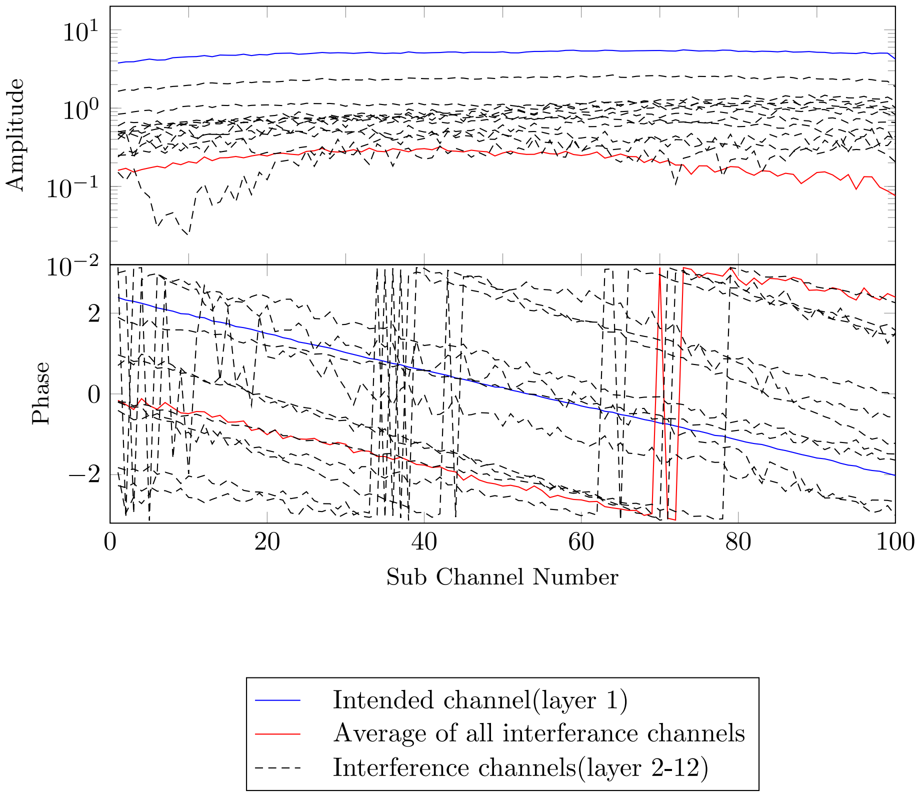


Figure 4.5: All layers, MRC at 5915MHz

4.3 Future work

The system contains some bugs causing certain places in the frame schedule to be off by one sample. In the current design these are simply filtered out. The root cause of this bug should be found and fixed.

Currently, only a LabView application has been written to receive the signal. This can only run on a Windows system with Labview installed. For storage of more data over a longer period of time, a storage server could be used. This will likely run Linux, so an application to receive and store the data has to be written. This should be possible based on the information provided in this report.

The data analysys show above only consideres a single time instance from each measurement. The data could be further analysed to answer how the radio channel changes with time.

Chapter 5

Conclusion

The work presented in this thesis has shown how a downlink channel can be measured in a multi user massive Multiple Inputs Multiple Outputs (MIMO) system by extending the functionality of the testbed. It has also shown what the challenges are with implementing such functionality in a large complex system.

By utilizing existing functionality that was previously only available inside the FPGA it has been show how channel downlink measurements can be done on the ReRaNP system. Tagging of the data with time stamps makes it possible to correlate it with data recorded on other systems.

All the necessary data formats created for this study have been documented such that it should be possible to interface this with other systems. This also gives future users of the system the ability to create their own programs to easily read the stored data.

As the system is presented, more measurements can easily be made, and all the data stored in a reasonable manner, using a common protocol such as UDP making it simple to create a program to store the data on nearly any device, such as a storage server. As the data rate from the channel measurements were reasonable for modern computer systems, an uncompressed clear text format was chosen to emphasize ease of use over storage density.

Bibliography

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, “Scaling up MIMO : Opportunities and challenges with very large arrays,” *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 40–60, 2013, ISSN: 10535888. DOI: `10.1109/MSP.2011.2178495`.

[2] T. S. Rappaport, *Wireless communications : principles and practice*. Prentice Hall PTR, 2002, p. 707, ISBN: 0130422320. [Online]. Available: `https://books.google.no/books?id=TbgQAQAAMAAJ&hl=no&source=gbs_book_other_versions`.

[3] Y. Okumura, “Field Strength and Its Variability in UHF and VHF Land-Mobile Radio Service,” *Review Electrical Communication Laboratory*, vol. 16, no. 9-10, pp. 825–873, 1968.

[4] M. Hata, “Empirical formula for propagation loss in land mobile radio services,” *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, Aug. 1980, ISSN: 0018-9545. DOI: `10.1109/T-VT.1980.23859`. [Online]. Available: `http://ieeexplore.ieee.org/document/1622772/`.

[5] J. Abraham and T. Ekman, “Power Inversion of the Massive MIMO Channel,” May 2019. [Online]. Available: `http://arxiv.org/abs/1905.07555`.

[6] B. Hochwald, T. Marzetta, and V. Tarokh, “Multiple-Antenna Channel Hardening and Its Implications for Rate Feedback and Scheduling,” *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 1893–1909, Sep. 2004, ISSN: 0018-9448. DOI: `10.1109/TIT.2004.833345`. [Online]. Available: `http://ieeexplore.ieee.org/document/1327795/`.

[7] “LabVIEW Communications MIMO Application Framework 1.1 Manual,” National Instruments, Tech. Rep., 2017.

[8] D. M. Harris and S. L. Harris, *Digital design and computer architecture*, ISBN: 9789382291527.

[9] J. Wågen, “Modifications to ReRaNP massive MIMO testbed for effective channel measurements in ReRaNP,” Tech. Rep., 2018.

NTNU
Norwegian University of
Science and Technology