Technische Universität München

Experimental Evaluation of Machine Learning based Wireless Communication Algorithms

Master Thesis

Karthik Sukumar

Supervisor: Prof. Wolfgang Utschick

Submission: Sep 07, 2020

TUM Department of Electrical and Computer Engineering
Professur für Methoden der Signalverarbeitung
Univ.-Prof. Dr.-Ing. Wolfgang Utschick

List of Figures

List of Tables

Contents

Acronyms

AFW	Application Framework.
AWGN	Additive White Gaussian Noise.
BPSK	Binary Phase Shift Keying.
BS	Base Station.
CFO	Carrier Frequency Offset.
CRS	Cell Specific Reference Signal.
DCI	Downlink Control Indicator.
DFT	Discrete Fourier transform.
DL	Downlink.
DMA	Direct Memory Access.
FDD	Frequency Division Duplexing.
FFT	Fast Fourier Transform.
FIFO	First In First Out.
FPGA	Field Programmable Gate Array.
GSM	Global System for Mobile communication.
H2T	Host to Target.
IEEE	Institute of Electrical and Electronics Engineers.
iFFT	inverse Fast Fourier Transform.
INN	Inverse Neural Network.
LTE	Long Term Evolution.
LUT	Look-up Table.

MCS	Modulation and Coding Scheme.
MIMO	Multiple Input Multiple Output.
NI	National Instruments.
OCXO	Oven Controlled Crystal Oscillator.
OFDM	Orthogonal Frequency Division Multiplexing.
PCIe	Peripheral Component Interconnect Express.
PDCCH	Physical Downlink Control Channel.
PDSCH	Physical Downlink Shared Channel.
PLL	Phase Locked Loop.
PPS	Pulse Per Second.
PRB	Physical Resource Block.
PSS	Primary Synchronisation Signal.
QPSK	Quadrature Phase Shift Keying.
RE	Resource Elements.
RF	Radio Frequency.
SNR	Signal to Noise Ratio.
SSS	Secondary Synchronisation Signal.
T2H	Target to Host.
TB	Transport Block.
TDD	Time Division Duplexing.
UE	User Equipment.
UL	Uplink.
VCO	Voltage Controlled Oscillator.

1 Introduction

Channel estimation is an important part not only in LTE but for any Orthogonal Frequency Division Multiplexing (OFDM) system in general. It is neccesary to invert the channel propogation effects to reduce bit error rate and improve data throughput [2, 3]. Channel estimation in OFDM systems is based on estimating the Channel Frequency Responce (CFR). For a perfect estimation every frequency and time resource block must have a reference symbol (pilot), but this leads to a high pilot overhead. To save on the bandwidth used for pilot transmission LTE inserts them sparsly in the 2D OFDM grid both in the time and frequency axis. The CFR for the complete grid is obtained by a 2D interpolation. The spacing in the time domain of the pilot symbols is called the coherence time, which is the minimum time for which the channel is expected to remain constant. The complexity increases when there are multiple antennas and in this case the channel needs to be estimated for each antenna i.e, pilots are needed for every antenna. This a challenge for massive MIMO applications where several antennas are used and each antenna needs to have pilots inserted in the 2D time frequency grid. There is a limited bandwidth for pilots to be filled in the grid and with massive MIMO the pilot overhead scales linearly with the number of antennas used.

Furthermore the complexity of Downlink is higher than in the Uplink case as the UEs are not expected to carry a massive MIMO transmitter or receiver. Current alternatives to pilot based channel estimation are model based channel estimation, where the environment is built in software and ray tracing employed to determine the channel. This is very resource intensive and needs to be updated in a dynamic environment. Research is also looking into machine learning applications for channel estimation.

In parallel to this thesis an inverse neural network (INN) based machine learning algorithm has been developed [4], and simultaed data was used to train the network. The output results of the trained network was impresive and the next step was to use real world MIMO data to evaluate the performance of the neural network.

This thesis aims to setup a functioning experimental MIMO jig in order to use the empirical data to evaluate the above mentioned machine learning algorithm that inverts the channel propogation effects. This is done by training an inverse neural

network (INN) using experimental data from a MIMO test jig.

This work has the following structure:

Chapter 2 contains the system model description and the basics MIMO transceivers.
Chapter 3 contains the fundamentals of channel estimation and the different detaied algorithms descriptions.
Chapter 4 contains the different possible solutions for setting up a MIMO testjig with the advantages and disadvantages of every approach.
Chapter 5 contains the detailed description of the final chosen approach for the MIMO setup.
Chapter 6 discusses the results of the experiments that were run with the chosen approach.
Chapter 7 concludes with potential improvements and tweaks to the existing setup.

2 System Model

As mentioned in Chapter 1, the goal of this thesis is to obtain measurement data from the MIMO setup. This collected experimental data will be applied to the machine learning model, that has been developed in parallel with this thesis. This chapter explains the relevant fundamentals of the LTE standard, which will be used as a basis for the experiments are described on a high level.

2.1 LTE

LTE stands of Long Term Evolution and is a successor standard to UMTS. LTE was chosen as a standard to use for the communication system as it is currently widespread in the telecommunications industry and because of the integrated support of simulation environments like MATLAB, Labview and Simulink. LTE is a multicarrier approach for multiple access which uses Orthogonal Frequency-Division Multiple Access (OFDMA) in the physical layer. OFDMA uses multiple carriers (known as sub carriers) spaced equally apart and can transmit independent data streams on each sub carrier [5].

A LTE frame is commonly represented as a 2D time frequency grid, where the vertical axis represents the sub-carriers (Frequency) and the horizontal axis represents time. LTE also comes in 2 flavours Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD). In this thesis the focus will be on FDD systems, which use seperate frequency bands, for uplink and for downlink data respectively. The advantage of an FDD system is that the uplink and downlink transmittion can happen simultaneously.

LTE has a certain predefined signalling symbol structure according to the standard [6]. In the following sections the frame structure and the different symbols in the LTE Frame shall be introduced.

2.2 LTE Waveform Processing

2.2.1 Transmission

A dual antenna transmitter using a USRP software defined radio as mentioned in Section 4.1 is used as a transmitter. A modified version of the LTE Application framework with MIMO 2x2 extension is used to log the data. The transmitter USRP is connected to a Host PC using a 4-lane PCIe connection. This is necessary for the high data throughput exchanged between the host and the devices.

LTE has a standardized frame structure which is commonly represented as a 2D time frequency grid, where the vertical axis represents the sub-carriers and the horizontal axis represents time. LTE also uses a predefined signalling structure according to the standard [6]. In the following sections the frame structure and the symbols relavent to channel estimation shall be introduced.

2.2.1.1 LTE Frame

A single frame is $10\mathrm{ms}$ long and consists of 10 smaller units called subframe, each $1\mathrm{ms}$ long. A symbol is the smallest unit of time for an LTE system and one subframe has 14 such symbols each approximately 66.7μs long. Scheduling is normally done on a subframe basis for both uplink and downlink communication.

LTE's time frequency grid contains many different signals each performing specific funtionality like broadcasting, control channel information, user data, among other functions.

For the purpose of channel estmation and the application as outline in this thesis, the most important signals are Primary Synchronisation Signal (PSS), Secondary Synchronisation Signal (SSS) and Cell Specific Reference Signal (CRS) which are described in detail in the subsequent sections.

2.2.1.2 Primary Synchronisation Signal (PSS)

OFDM is extremely time and frequency sensitive, hence it is very important to know the exact start of every frame. The PSS helps to achieve the synchronisation of the frame by using a specific sequence called the Zadoff-Chu sequence [6]. The Zadoff-Chu sequence has the property of constant amplitude zero autocorrelation waveform (CAZAC sequences) such that cyclically shifted versions of the waveform are orthogonal to original waveform. The sequence is described in Equation 2.1 where u can be 25,29 or 34 depending on the cell ID. The PSS is broadcast twice every radio frame and the symbols are identical each time.

$$d_u(n) = \begin{cases} e^{-j\frac{\pi un(n+1)}{63}} & n = 0, 1, ..., 30 \\ e^{-j\frac{\pi un(n+1)(n+2)}{63}} & n = 31, 32, ..., 61 \end{cases} \tag{2.1}$$

2.2.1.3 Secondary Synchronisation Signal (SSS)

The SSS is a 62 bit BPSK modulated pseudo random sequence [6]. It is broadcast twice in a frame once in subframe 0 and once in subframe 5, one symbol before the PSS. The 2 sequences of transmission in a frame are different so that the UE can identify which position in the frame the synchronisation happens.

2.2.1.4 Cell Specific Reference Signal (CRS)

As mentioned in Chapter 1 the channel needs to be estimated in order to reverse the channel propogation effects. With the help of CRS the channel can be estimated by placing equally spaced reference symbols every 6 subcarriers starting from subcarrier 2 on symbols 1, 8, 15, etc... and every 6 subcarriers starting from subcarrier 5 on symbols 5, 12, 19, etc...[6]. The signals received by the UE and the channel effects are inferred based on amplitude damping and phase shift. Placing the signals in the above defined spacing gives the best coverage to interpolate over in time and frequency.

The signals to be placed on the grid are decided by the Equation 2.2 as shown below, with the parameters defined in Table 2.1 [6].

$$r_{l,n_s}(m) = \frac{1}{\sqrt{2}}(1-2\cdot c(2m)) + j\frac{1}{\sqrt{2}}(1-2\cdot c(2m+1)),\ m = 0, 1, ..., 2N_{RB}^{max,DL} - 1 \tag{2.2}$$

Parameter	Description
$c(m)$	Pseudo Random Sequence defined in [6]
$N_{RB}^{max,DL}$	Maximum number of Downlink resources blocks
n_s	Slot number in the frame
l	OFDM Symbol number in the frame

Table 2.1: Parameter definitions for evaluating CRS Symbols

In the case of multiple antennas, each antenna port has a dedicated CRS slots and have their corresponding positions on the OFDM time/frquency grid as shown in Figure 2.1. These CRS signals for different antennas are signaling overheads,

which implies that they cannot be used for data transmission. On the receiver end this particular symbol is decoded and the channel estimate is calculated for the respective transmit antenna.

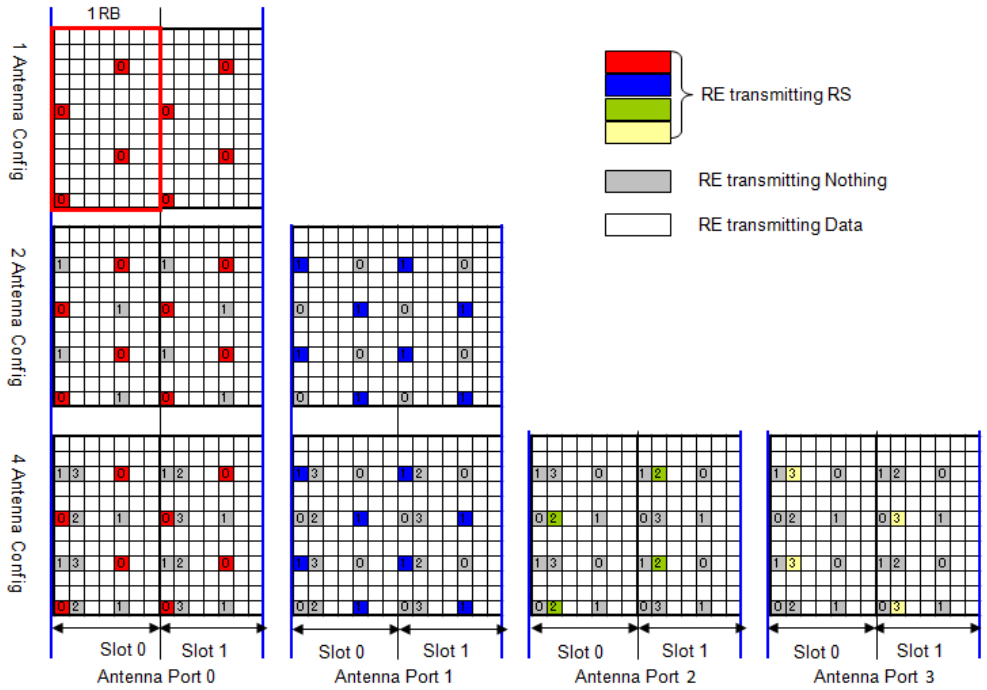


Figure 2.1: Reference Signal layout for multi antenna configurations of 1,2 and 4 antenna systems in LTE

2.2.1.5 Final Waveform

All the signals mentioned in Section 2.2.1.1 are generated in Labview along with other signals that are not mentioned here due to the complexity are generated using the LTE Application Framework Software and arranged in a 2D grid of frequency and time as shown below in Figure 2.2. The CRS for a 2 antenna system are placed every 6 subcarriers in the frequency domain and every 3 or 4 time symbols apart, according to the position on the grid. Where as the PSS and SSS only repeat every 5 subframes.

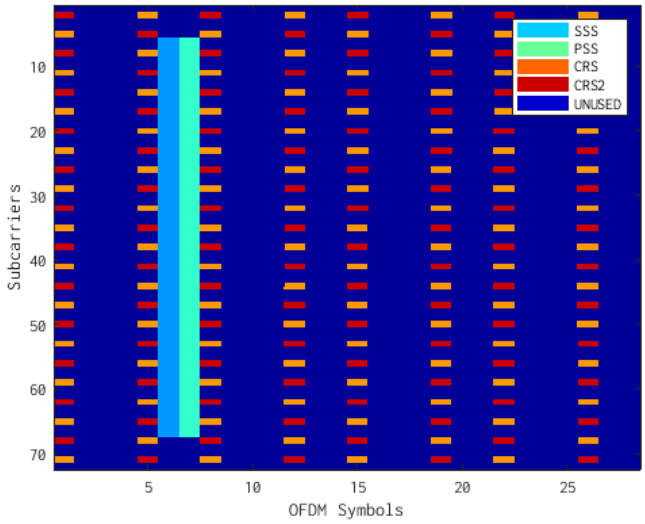


Figure 2.2: Resource Block Grid for 2 sub frames

One of the disadvantages of using OFDM in the physical layer is the high peak to average power ratio (PAPR) of the signal. High PAPR translates to high fidelity requirements of power amplifiers on the transmitter side. It also induces non-linear distortions to the signal [5]. To ensure that the PAPR of the signal is as low as it can be, random QPSK symbols are transmitted on the unused slots instead of 0s.

2.2.2 Reception

For the LTE receiver processing a USRP with 2 antennas acquires the RF data from both ports and processes the data in the FPGA in real time. The data requested by the host is then forwarded to it using the high speed PCIe interface. A more in depth explaination of the receiver processing is explained in Chapter 5.

The channel estimation data is forwarded from the FPGA IQ processing engine to the host which is then logged to a TDMS file (NI propriotory format) and post processed to extract samples for the purpose of the experiments. The main steps of the DL receiver processing once enough samples have been captured are the following.

2.2.2.1 Carrier Offset Estimation and Correction

OFDM is extremely senstive to frequency shifts in the received signal. In the case that the receiver or transmitter center frequency clock was not accurate enough, the frequency shift needs to be estimated and corrected. This is the very first step before processing the baseband waveform.

2.2.2.2 Frame synchronisation

As mentioned in Section 2.2.1.1 the synchronisation is a very important part of knowing where the LTE frame begins. This is done by correlating the received signal with the known Zadoff Chu Sequence and to look for the peak. Figure 2.3 shows an example correlation of a Zadoff Chu sequence and a received signal containing 307200 samples in total amounting to 2 frames of a 10MHz Bandwidth LTE signal. 4 peaks are to be expected here as there are 2 frames, each containing 2 PSS sequences.

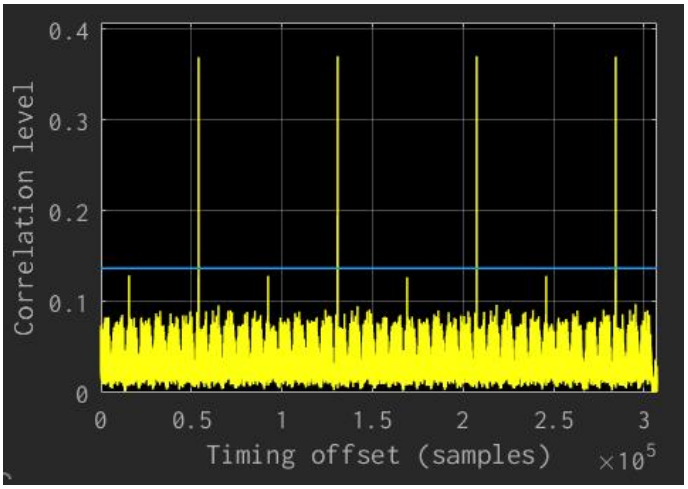


Figure 2.3: PSS Correlation

2.2.2.3 Channel Estimation

Once the frame has been demodulated and the 2D OFDM grid has been obtained, the channel can be estimated based on the original pilot symbols structure known to the receiver, hence the phase and amplitude of the particular sub carrier can be obtained. A linear interpolator is applied in the frequency axis to interpolate the channel estimate from 200 subcarriers to 1200 subcarriers(for a full bandwidth system) and a zero

order hold is used in the time axis until the next pilot symbol in the time domain is encountered. The details are explained in the Chapter 3.

2.2.3 Antenna

The analog time domain signal is transmitted from the USRP (Section 4.1) over the air using a Triband antenna. For the setup an omni directional Antenna from NI capable of transmitting and receiving around frequencies of 144, 400 or 1200 MHz is used for the transmitter and receiver as shown in Figure 2.4.

Figure 2.4: Narrowband Omnidirectional Antenna used for transmitting and receiving the LTE Signals

3 Channel Estimation

LTE was chosen as the standard to use here as it is very mature and has readily available MATLAB/Labview based implementation. In the case of this thesis the aim is not to reinvent standard by redesigning pilot symbol placements, instead existing standards were used in order to collect experimental data. This reduces design time and focusses on the issue at hand which is channel estimation data of a MIMO Channel.

3.1 OFDM

LTE is based on OFDMA in the physical layer which is a multi carrier communication scheme [7]. As the name suggests OFDM uses orthogonal sub carriers from an orthonormal system to form the basis for independent data streams. For band limited transmission systems with finite access time per channel use the dimension of the parallel data stream is given by the Equation 3.1 [1].

$$N = BT \tag{3.1}$$

Parameter	Description
N	Dimension of system
B	Signal Bandwidth
T	Channel access time

Table 3.1: Parallel data streams parameter description

The orthonormal basis function can be mathematically modelled as the Equation 3.2 [1].

$$\begin{aligned} \psi_{b,q} &= p_{T_b}(t) exp(j2\pi q \frac{t}{T}) \\ p_{T_b}(t) &= \begin{cases} 1; & t \in T_b \\ 0; & otherwise \end{cases} \end{aligned} \tag{3.2}$$

Parameter	Description
$\psi_{b,q}$	normalized orthogonal basis functions
b	channel access slot
q	sub carrier index
T	channel access time
T_b	$bT \leq t < (b+1)T \subset \mathbb{R}$

Table 3.2

The transmitted data can hence be modelled as the following

$$x_b(t) = \sum_{q=0}^{N-1} \underbrace{X_{b,q}}_{\text{data}} \psi_{b,q}(t) \tag{3.3}$$

For a given ideal AWGN Channel, where there is no delay spread or multipath propogation, the corresponding received data is modelled as

$$y_b(t) = \sum_{q=0}^{N-1} x_b(t)\psi_{b,q}(t) + \eta_b(t) \tag{3.4}$$

where $\eta_b(t)$ is the additive noise

The demodulation is based on the same set of orthonormal basis vectors as that of the transmitter, hence we have

$$\hat{x}_{b,q} = \langle y, \psi_{b,q}(t)\rangle + \langle \eta_b, \psi_{b,q}(t)\rangle = x_{b,q}(t) + \eta_{b,q} \quad \forall q = 1, ..., N \tag{3.5}$$

In the case of a channel with delay spread (also referred to as multipath propogations) we model it as the following

$$h(t) = \sum_{l=0}^{L-1} h_l \delta(t - l\frac{T}{N}) \tag{3.6}$$

where h is the time domain channel taps. The term $h_l x_{b,q}\psi_q(t - l\frac{T}{N} - bT)$ leads to interference between adjacent channels T_{b-1}, T_b, T_{b+1}, etc. This has be avoided to have faithful reproduction of transmitted data on the receiver end.

3.1.1 Cyclic Prefix

As a result of Equation 3.6 we have to compensate for the interference of the symbols of the b_{th} channel with the other channels. A simple solution to this interference is a Cyclic prefix. As the name suggests it is a cyclic addition of the signal at the transmitter end, which means copying the beginning of the signal and adding it to the end. To this end $T = T + T_p$, where T_p is the duration of the cyclic prefix. In the LTE standard the cyclic prefix lengths are predefined based the normal or extended mode [6]

$$\begin{gathered} \psi_{b,q}(t) = p_{T_b}(t) exp\left(j2\pi q \frac{t}{T}\right) \\ \text{where } \psi_{b,q} \in T_b = [-T_p + bT_s, bT_s + T[\end{gathered} \tag{3.7}$$

This is illustrated in the Figure 3.1

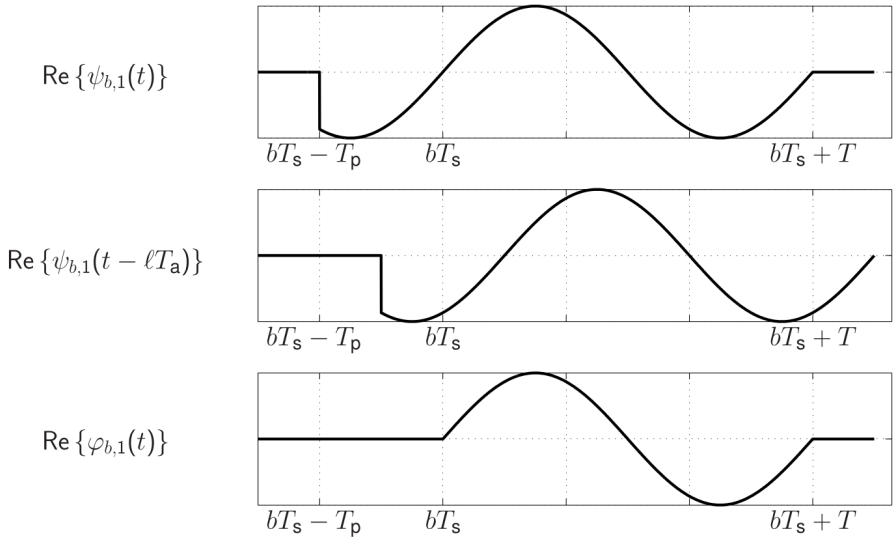


Figure 3.1: Cyclic prefix addition [1]

On the receiver end

$$y_{b,n} = \sum_{l=1}^{L-1} h_l \delta_{n-l} * \sum_{l=-Np}^{N-1} x_{b,k} \delta_{n-k} = \sum_{l=1}^{L-1} \sum_{k=-Np}^{N-1} h_l x_{b,k} \delta_{n-l-k} \tag{3.8}$$

subject to $L-1 \leq N_p$,where N_p is the number of samples of the cyclic prefix

$$\begin{aligned} y_{b,n} &= \sum_{l=1}^{L-1} h_l x_{b,n-l} = \sum_{l=1}^{L-1} h_l x_{b,mod_N(n-l)}, \quad n,q \in 0,\cdots,N-1 \\ y_{b,n} &= h_n \circ x_{b,n} \Leftrightarrow \frac{1}{N} Y_{b,q} = H_q X_{b,q}, \qquad n,q \in 0,\cdots,N-1 \end{aligned} \tag{3.9}$$

$$\begin{bmatrix} y_{b,-2} \\ y_{b,-1} \\ y_{b,0} \\ y_{b,1} \\ y_{b,2} \\ y_{b,3} \end{bmatrix} = \begin{bmatrix} h_0 & & & & & \\ h_1 & h_0 & & & & \\ h_2 & h_1 & h_0 & & & \\ & h_2 & h_1 & h_0 & & \\ & & h_2 & h_1 & h_0 & \\ & & & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_{b,-2} \\ x_{b,-1} \\ x_{b,0} \\ x_{b,1} \\ x_{b,2} \\ x_{b,3} \end{bmatrix} + \begin{bmatrix} h_2 & h_1 & & \\ & h_2 & & \\ & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} x_{b-1,2} \\ x_{b-1,3} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \eta_{b,-2} \\ \eta_{b,-1} \\ \eta_{b,0} \\ \eta_{b,1} \\ \eta_{b,2} \\ \eta_{b,3} \end{bmatrix} \tag{3.10}$$

With a cyclic prefix chosen such that $x_{b,-n} = x_{b,N-n}, n \leq N_p$, then $x_{b,-1}$ and $x_{b,-2}$ can be deleted and equation 3.10 can be rearranged as the following

$$\begin{bmatrix} y_{b,0} \\ y_{b,1} \\ y_{b,2} \\ y_{b,3} \end{bmatrix} = \begin{bmatrix} h_0 & & h_2 & h_1 \\ h_1 & h_0 & & h_2 \\ h_2 & h_1 & h_0 & \\ & h_2 & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_{b,0} \\ x_{b,1} \\ x_{b,2} \\ x_{b,3} \end{bmatrix} + \begin{bmatrix} \eta_{b,0} \\ \eta_{b,1} \\ \eta_{b,2} \\ \eta_{b,3} \end{bmatrix} \tag{3.11}$$

Taking the DFT of equation 3.11 gives

$$y_b = \hat{H} x_b + \eta_b \Leftrightarrow Y_b = \frac{1}{N} F \hat{H} F^H X_b + F \eta_b \tag{3.12}$$

Where F is the fourier matrix. Since $\hat{H}$ is a circulant matrix (also a convolutional matrix) it can be broken down as equation 3.13 [8]. The eigenvalues of the circulant matrix are the fourier transformed coefficients of the first column while the eigenvectors are the columns of the inverse fourier transform matrix. Equation 3.13 presents a eigenvector decomposition of matrix $\hat{H}$ where F^H is the inverse DFT matrix and Λ is the diagonal matrix with entries being the fourier transform terms of the first column of the matrix $\hat{H}$.

$$\begin{aligned} \hat{H} = U \Lambda U^{-1} \quad &= \frac{1}{N} F^H \Lambda F \\ \frac{1}{N} F \hat{H} F^H &= \Lambda \end{aligned} \tag{3.13}$$

Therefore equation 3.12 simplifies to

$$\begin{aligned} Y_b &= \Lambda X_b + F\eta_b \\ &\text{or} \\ Y_b &= diag(X_b)H + F\eta_b \end{aligned} \tag{3.14}$$

where Λ, is a diagonal matrix and H is the vector of channel coefficients in the frequency domain and decouples the subcarrier distortions with each other. Because of the cyclic prefix converting linear convolution into cyclic convolution, the received signal in the frequency domain is linearly dependent **only** on the corresponding frequency component of the channel propogation effect.

3.1.2 Pilot assisted channel estimation

In order to determine the channel propogation effects known pilot symbols are inserted in the transmit frames as mentioned in Section 2.2.1.4 and the channel effects are inferred from the demodulated symbol on the receiver side. Although populating all the subcarriers with pilots is the best approach, this leads to high channel reference signal overheads and less data bandwidth. Hence in practical system implementations only P out of N subcarriers carry the pilots and the rest are interpolated.

$$Y = XSH + \eta \tag{3.15}$$

$Y \in \mathbb{C}^P, H \in \mathbb{C}^N, \eta \in \mathbb{C}^P$ are vectors
$X = diag[X_1, ..., X_P] \in \mathbb{C}^{PxP}$
and $S : \mathbb{C}^N \rightarrow \mathbb{C}^P \Rightarrow H \mapsto H_P = [H_{s1}, ..., H_{sp}]^T$. where S is the selector matrix which selects P out of N subcarriers which carry the pilot symbols.

For non fully loaded OFDM systems which is typical in the real world the Equation 3.15 reduces to the equation below.

$$Y_P = XH_P + \eta \tag{3.16}$$

where $H_P = SH$. A least squares estimate of the H_P gives

$$\hat{H}_{P,LS} = X^{-1}Y_P \tag{3.17}$$

H_P is then subsequently interpolated across all the N subcarriers. This can be done in many ways using either

- Linear interpolation

- Sinc interpolation
- Cubic/Spline interpolation
- Wiener filters

3.2 MIMO Channel Estimation

For the MIMO channel estimate the basics is the same as described in the Section 3.1.2, except in a $N_t \times N_r$ system we have $N_t N_r$ possible channels per subcarrier. This is explained as follows, from Figure 3.2 it can be seen that there are $N_t N_r$ channels per subcarrier in the frequency domain. Equation 3.17 is used to calculate each channel. This is enabled by non interfering pilots for different antennas. This **must** be followed for faithful channel estimation and is a necessary criteria which has been followed in the LTE standard.

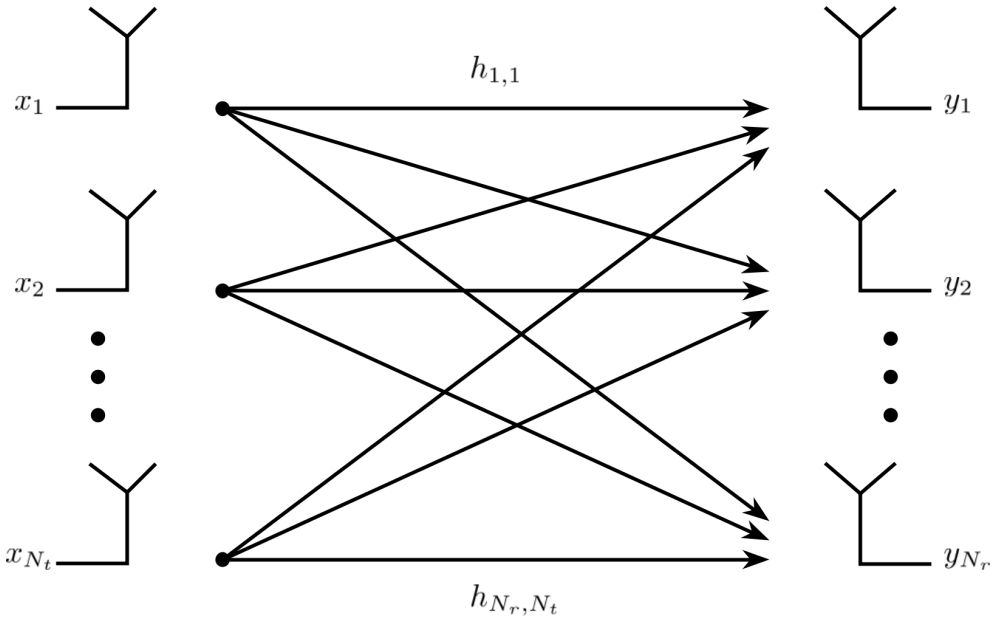


Figure 3.2: MIMO channel model

$$\hat{x} = Wy \tag{3.18}$$

Once the channels propogation effects have been calculated each symbol has to be

equalized to compensate the distortion effect caused by the dispersive channel. In the case of a MIMO channel there are 3 main algorithms which are as follows

3.2.1 Maximum Ratio Combiner

$$W_{MRC} = C\hat{H}^H \tag{3.19}$$

where C is a diagonal matrix with

$$C_{k,k} = \left(\sum_m |H_{m,k}|^2 \right)^{-1} \tag{3.20}$$

and can be considered as a normaliser for the columns of the matrix H.

3.2.2 Zero Forcing

$$W_{ZF} = (\hat{H}^H \hat{H})^{-1} \hat{H}^H \tag{3.21}$$

Zero forcing is an algorithm that is avoided although given its simplicity, the reason being that the noise can be amplified in the case of a low SNR situation. Hence in practice this algoritm is avoided. The implimentation is a Least squares estimate of the vector $\hat{x}$.

3.2.3 MMSE

$$W_{MMSE} = (\hat{H}^H \hat{H} + \sigma^2 I)^{-1} \hat{H}^H \tag{3.22}$$

where σ is the inverse SNR.

MMSE algorithm performs the best under all situations. For high SNR scenarios it converges to an ZF algorithm and for a low SNR situation it converges to the MRC algorithm.

4 Potential Hardware Setups

MIMO channel measurements can be achieved using multiple methods. This chapter discusses some of the potential approaches which were tested and elaborates each of their advantages and disadvantages. A detailed final implementation is described in Chapter 5 where the setup configurations are discussed. The following are some possible realisation for a MIMO Setup.

4.1 Software Defined Radios USRP

USRP is a Software Defined Radio (SDR) designed by National Instruments that enables quick prototyping of different wireless applications. It is aimed at hobbists, research labs, universities, etc... or anyone interested in evaluating custom algorithms. The SDR used in this masters thesis is a USRP2940, specifications of which are described in Table 4.1.

Model	USRP2940
Baseband Bandwidth	40MHz
RF-Operating Frequency	50MHz-2200MHz
FPGA	Kintex-7 410T
No of Transmitters	2
No of Receivers	2
Connectivity	MXIe, Ethernet
Oscillator	Internal Crystal
ADC/DAC	14 (For Rx)/16 (For Tx) bit
Frequency Accuracy	2.5 ppm
Maximum Power Output	20dBm
Maximum I/Q Sample Rate	200MHz

Table 4.1: USRP2940 SDR Product details

4.1.1 PCIe-8371

The USRP SDRs are PXIe communication based devices that connect to a Host/PC using a 1/4/8 way PCIe slot. For this setup a 4 lane PCIe daughter card (*PCIe-8371*) was installed on the motherboard of the host PC to provide a dedicated PCIe communication port for the USRP.

4.1.2 Host

The host chosen to be used here is a Fujitsu Celcius M770 with the following specifications.

OS	Microsoft Windows 10 Education
Version	10.0.18363 Build 18363
OS Manufacturer	Microsoft Corporation
System Name	TUEIMSV-PRAKT01
Manufacturer	FUJITSU
System Model	CELSIUS M770
System Type	x64-based PC
System-SKU	S26361-Kxxx-Vyyy
Processor	Intel(R) Xeon(R) W-2123 CPU @ 3600 MHz 4 Cores, 8 logical Processors
BIOS-Version/-Date	FUJITSU // American Megatrends Inc. V5.0.0.13 R1.8.0 for D3498-A1x, 08.10.2019
SMBIOS-Version	3.2
BIOS-Mode	UEFI
BaseBoard-Manfacturer	FUJITSU
BaseBoard-Product	D3498-A1
BaseBoard-Version	S26361-D3498-A1
Platform Type	Workstation
Installed Physical Memory	16,0 GB
Total Physical Memory	15,6 GB

Table 4.2: Fujitsu Celcius M770 Configuration

Streaming a Custom LTE Waveform into the 2 seperate USRP transmitters, would have been the simplest and the ideal solution. However since the 2 transmitters on a single device are not time synchronised, which is essential for MIMO transmission

and reception, this was not an option. The details of the issues are described in the Appendix A.2.

4.2 MIMO Application Framework (MIMO AFW)

MIMO Application Framework (MIMO AFW) is a Software developed by National Instruments, that offers a comprehensive plug and play MIMO setup. This setup requires a host of additional hardware which are required for the functioning of the MIMO AFW [9]. When setup with all the required Hardware MIMO AFW can support a Multi-user system with a maximum of 128 Antennas on the Base Station (BS) side and upto 12 Antennas on the User Equipment (UE) side.

A list of different Hardware parts needed to run this MIMO AFW is listed in Table 4.3. There are many different MIMO configurations possible and the hardware requirements of each of these configurations are mentioned in Table 4.4. A high level system overview of the main features of MIMO AFW is as follows

- Multi-User MIMO transmission between one Base Station (BS) with up to 128 Antennas and up to 12 single antenna Mobile Stations (MS)
- Single-user MIMO transmission between one BS with up to 128 antennas and one MS with up to 12 antennas
- Modulation Schemes from QPSK to 256 QAM
- Automatic gain control (AGC) at the BS and MS
- FPGA based real time signal processing such as modulation, over-the-air synchronization, MIMO equalization and MIMO precoding
- Scalable number of antennas (multi-antenna MS: between 2 and 12; BS: between 2 and 128). Interfaces and configuration adapt automatically
- Fully reconfigurable LTE like radio frame structure
- Bi Directional TDD and FDD functionality transmission of 20MHz bandwidth

Part Number	**Description**
USRP-2940	SDR
PXIe-7976	FPGA Module for FlexRIO
CDA-2990	Clock Distribution Device
CPS-8910	Switch Device for PCI Express
PXIe-6674T	Synchronization Module
PXIe-1085	Chassis
PXIe-8135	Controller

Table 4.3: Additional Hardware for required for MIMO AFW to function

	128-antenna BS 8 subsystems	**64-antenna BS 4 subsystems**	**32-antenna BS 2 subsystems**	**16-antenna BS 1 subsystems**	**8-antenna BS 1 subsystems**
USRP-29xx SDR Reconfigurable Device	64	32	16	8	6
PXIe-1085 Chassis (18-Slot, 24 GB/sSystem Bandwidth (BW))	1	1	1	1	1
PXIe-8135 Controller	1	1	1	1	1
PXIe-7976 FPGA Module for FlexRIO	5	3	2	2	2
PXIe-6674T Synchronization	1	1	1	1	1
CDA-2990 Clock Distribution Device	8	5	3	1	1
CPS-8910 Switch Device for PCI Express	8	4	2	1	1

Table 4.4: MIMO Configurations and HW requirements

4.2.1 USRP 2940

As mentioned in Section 4.1, this is the backbone of the architecture. The Software defined radio (USRP2940) is used as an air interface for over the air transmission. There are host of other options that can be used here instead of the USRP2940. Table 4.5 lists the alternatives with an overview of the functionality of each of the parts.

4.2.2 PXIe-7976

MIMO has very demanding operations that are quite compute intensive such as precoding, equalization as well as channel estimation in the frequency domain. In addition to the aforementioned processing tasks this FPGA card also perform the *bit processing*. This PXIe communication based FPGA card contains a Xilinx Kintex-7 FPGA and moves data in and out using an 8 lane PCIe slot.

4.2.3 CDA-2990

This device also known as the Octoclock is a clock distribution accessory. It can either receive an input reference clock and distribute the clock to 8 other devices synchronously along with a PPS (Pulse Per Second) trigger. The CDA-2990 also contains an input for a GNSS Antenna which uses the GNSS signal to generate a PPS signal. In the absense of a GNSS Antenna the device generates its own internal clock based on an internal oven controlled oscillator (OCXO). The the inputs terminations are 50 Ohms as can be seen from Appendix B.

4.2.4 CPS-8910

The CPS-8910 is a 8 way PCIe data aggregator and has 2 upstream ports. With the 8 downstream ports it can aggregate large amounts of data from a maximum of 8 USRPs and send them out to a PC/Controller via the 2 other upstream ports. This is essential for Massive MIMO applications.

4.2.5 PXIe-1085

The PXIe-1085 is an 18 slot chassis, which can be populated by many different daughter cards suitable to the customers needs. Out of the 18 slots, 16 are hybrid that can be populated with various add ones, and one slot is reserved for a timing and synchronisation slot as described in Section 4.2.6 and the other reserved for a PXI-controller which is define in Section 4.2.7. The chassis is capable of supporting a throughput of upto 24GBps.

4.2.6 PXIe-6674T

The PXIe-6674T generates and routes clocks and trigger signals (PPS Signals) between PXI devices or chassis. This timing and synchronisation card not only generates an accurate clock but can also shift levels of an input signal according to the user's settings. Although the octoclock distributes the clock, the signal is synchronised and level shifted by the PXIe-6674T.

4.2.7 PXIe-8135

The PXIe-8135 is a PXI Controller needed to handle the different slot daughter cards installed in the *PXIe-1085* (Section 4.2.5). Its an Intel Core i7 based embedded controller for PXI express systems. The controller also has a variety of ports to support the 10/100/1000BASE-TX Gigabit Ethernet, 2 SuperSpeed USB ports and four Hi-Speed USB ports, as well as an integrated hard drive, serial port, and other peripheral I/O.

4.2.8 Implementation Advantages and Disadvantages

The advantages of using MIMO AFW is that it reduces time to prototype as the functionality is readily available in the form of Hardware and Software, it is also a scalable model providing a solution with up to 128 BS antennas and 12 MS antennas. But given the amount of hardware required to have a minimum viable product, it is quite an expensive undertaking (over €100.000) and is hence cost preventative. Apart from that there is also the time cost to setup the instrument and check it functionality which is not to be neglected.

Model	RF-Frequency Range	RF-Frontend Bandwidth	FPGA	Inputs	Outputs	Communication	GPS Osillator
USRP-2940	5 MHz - 2.2 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2940	50 MHz – 2.2 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2942	400 MHz - 4.4 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2942	400 MHz - 4.4 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2943	1.2 GHz - 6 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2943	1.2 GHz – 6 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2944	10 MHz - 6 GHz	160 MHz	Kintex-7 410T	2	2	MXIe Ethernet	No
USRP-2945	10 MHz - 6 GHz	80 MHz	Kintex-7 410T	4	0	MXIe Ethernet	No
USRP-2950	50 MHz - 2.2 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2950	50 MHz - 2.2 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2952	400 MHz - 4.4 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2952	400 MHz - 4.4 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2953	1.2 GHz - 6 GHz	40 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2953	1.2 GHz - 6 GHz	120 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2954	10 MHz - 6 GHz	160 MHz	Kintex-7 410T	2	2	MXIe Ethernet	Yes
USRP-2955	10 MHz - 6 GHz	80 MHz	Kintex-7 410T	4	0	MXIe Ethernet	Yes

Table 4.5: List of alternative Software defined radios offered by National Instruments

4.3 LTE Application Framework

LTE Application Framework is a Software that National Instruments designed and offers to provides us a **Downlink ONLY** 2x2 LTE setup. This setup does not require the additional hardware which are required for the functioning of the MIMO AFW.

4.3.1 Hardware Requirements

This setup is quite a simplified version of the MIMO AFW and only requires a pair of USRPs (Section 4.1), a pair of the PCIe daughter cards(Section 4.1.1) and a host PC(Section 4.1.2). One USRP acts as a eNodeB with 2 antennas (Base Station) and the other USRP acts as a UE with 2 antennas.

4.3.2 Software Requirements

The LTE AFW Software template that NI provides in its example libraries only has a LTE SISO implementation. But an internal development version of a 2x2 MIMO Extension was issued to TUM for the sake of expediting the experimental setup since MIMO AFW was not a financial viable option. However the NI internal development version of the code was meant to be run on a version 2.1 of Labview Communication Suite which is not supported on Windows 10. Hence the project had to be ported to the latest version of Labview Communications Suite (now called Labview NXG). Details of the software project are described in Chapter 5.

4.3.3 Implementation Advantages and Disadvantages

The advantages of using LTE AFW is that it works with minimum hardware, albeit with a limited capability of providing only a 2x2 MIMO system. Unlike MIMO AFW which is capable of bidirectional MIMO communications, LTE AFW can only perform unidirectional communications, namely from the eNodeB(Base station) to the UE. However given all the time and budget constraints, LTE AFW was chosen as the preferred solution. The architecture of the software and the modifications are explained in the following chapter.

5 Experimantal Setup

Chapter 4 discussed the possible options to implement a MIMO setup. The first option (Section 4.1) was not viable due to technical limitations as it is described in Appendix A.2.

The second option (Section 4.2) was financially unfeasible as the additional hardware for the modular MIMO set was to cost over €100.000. The only viable option was to use the LTE AFW with a MIMO Extension, although limited to a 2x2 MIMO system, was sufficient in this case.

5.1 LTE Application Framework MIMO Extension

LTE AFW is a SISO LTE Release 10 implementation, where as LTE AFW 2x2 MIMO Extension was developed internally by NI as a plugin to the SISO framework as a proof of concept for their MIMO Application Framework product. The main goal of LTE 2x2 MIMO Extension was to demonstrate a functioning 2x2 MIMO System. It is implemented for **DL only** without the UL feedback. The software is not readily available for customers to purchase, but it was given to MSV as a workaround for the MIMO implementation.

For the implementation of the LTE AFW MIMO Extension as outlined in Chapter 4.3 the equipment required is minimal. A set of 2 USRPs, one acting as a UE (Receiver) and the other acting as a Base Station (eNodeB) completes the transceiver chain and the Host PC does the graphing, data visualisation, user data communications and device setup. Figure 5.1 illustrates the simple HW connections required to make the device communicate with the host. For details on the part descriptions refer to Section 4.3.1.

5.1.1 LTE AFW MIMO Extension Architecture

Figure 5.2 shows the block diagram of the system in the DL eNodeB and UE operation modes. Data streams that require high data rates for data transfer between host and FPGA are implemented as DMA FIFOs. These streams include the payload and UL

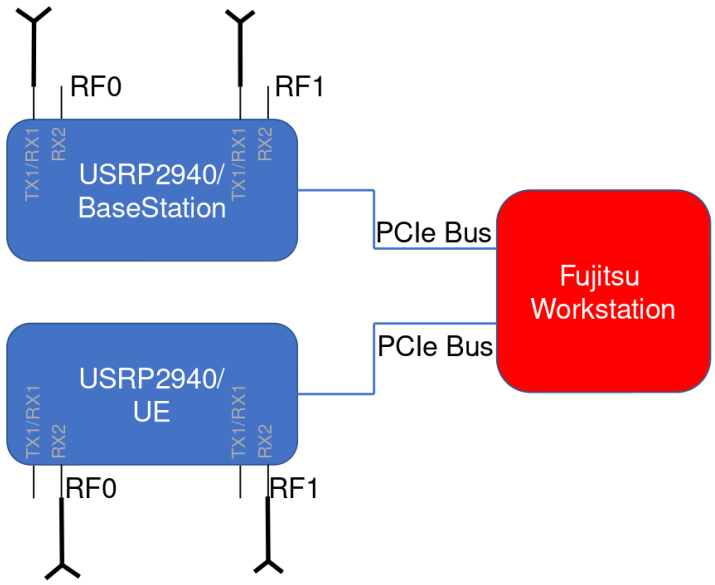


Figure 5.1: Overview of the LTE 2x2 MIMOHW Connections

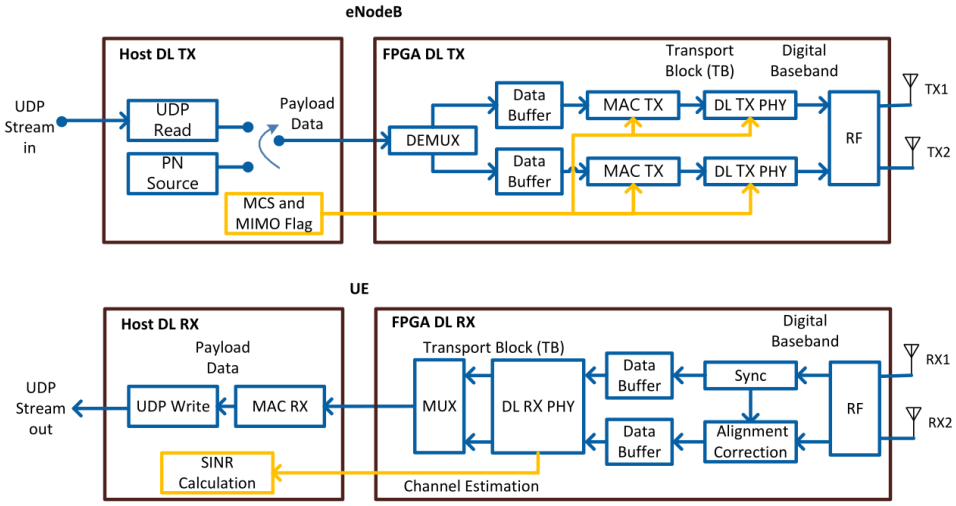


Figure 5.2: Overview of a 2 device Tx-Rx setup as intended in the final implementation

data from host to FPGA and the received PDSCH transport blocks from FPGA to host. In-phase/quadrature (I/Q) samples for constellation and spectrum display as

well as the channel estimation (**ONLY** diagonal coefficients h_{11} and h_{22}) values are also transferred from FPGA to host using DMA FIFOs. Further status information is transferred to the host by reading the indicator values [10]. As it can be seen both RX and TX components are implemented in the Host Software as well as in the FPGA Hardware. Hence the same application can be used as an *eNodeB* as well as *UE*. The *eNodeB* mode uses the DL TX processing units and *UE* uses the DL RX processing units which are explained in Section 5.1.3.

The following lists the various processing blocks in the application and a brief description of the corresponding tasks of each of the modules as shown in Figure 5.2.

- **UDP read** - Reads data, provided by an external application, from a UDP socket. The data is used as payload data in the transport block (TB). This data is then encoded and modulated as an LTE DL signal by the downlink transmitter (DL TX PHY).
- **UDP write** - Writes the payload data, which was received and decoded from the LTE DL signal by the downlink receiver (DL RX PHY), to a UDP socket. The data can then be read by an external application.
- **MCS and MIMO Flag** - Creates a cluster including the required information for the downlink control information (DCI) message such as the modulation and coding scheme (MCS) and the MIMO flag. The MIMO flag is a control that is used to activate MIMO or SISO in the eNodeB.
- **DEMUX** - A demultiplexer is used to split the data stream to Data Stream 1 and Data Stream 2 if the transmitter is configured to use both RF chains for TX.
- **Data Buffer** - Two buffers are used to store data coming from the Payload H2T FIFO into two local FIFOs for TX chain 1 and TX chain 2. In the RX mode, two buffers are used to store the received data coming from the synchronization module to synchronize the signal processing in RX chain1 and RX chain 2.
- **MAC TX** - A simple MAC implementation that adds a header to the TB containing the number of payload bytes. The header is followed by the payload bytes, and the remaining bits of the TB are filled with padding bits.
- **DL TX PHY** - Physical layer (PHY) of the downlink (DL) transmitter (TX). Encodes the physical channels and creates the LTE downlink signal as digital baseband I/Q data. This code includes encoding of the control channel (PDCCH), encoding of the data channel (PDSCH), resource mapping, and orthogonal

frequency-division multiplexing (OFDM) modulation. The white paper Labview communications LTE Application Framework 19.5[10] has the details of the implementation of the TX Bit Processing and IQ Signal Processing.

- **Sync** - The primary synchronization sequence (PSS) is used for synchronization. The synchronization is done using the received signal on RF0/RX2. The synchronization results such as frame alignment and frequency offsets estimation (integral and fractional) are used to adjust the received signals on both ports RF0/RX2 and RF1/RX2.
- **DL RX PHY** - Physical layer (PHY) of the downlink (DL) receiver (RX). Demodulates the LTE downlink signal and decodes the physical channels, OFDM demodulation, resource demapping, MIMO channel estimation and MIMO equalization, decoding of the control channel (PDCCH), and decoding of the data channel (shared channel, PDSCH). For MIMO equalization, three algorithms have been implemented namely: simple algorithm, minimum mean squared error (MMSE), and zero-forcing (ZF). The simple algorithm means that the inverse of the estimated channel parameters is used directly in the equalizer. There is also an option of disabling the equalization using a switch to get raw data.
- **MUX** - A multiplexer is used to combine the received data from both RX chains to be transmitted to the host.
- **MAC RX** - Disassembles the TB and extracts the payload bytes.
- **SINR calculation** - Calculates the signal-to-interference-noise-ratio (SINR) based on the channel estimation that was used for PDSCH decoding. Channel estimation is either based on cell-specific reference signals (CRS), or on UE-specific reference signals (UERS).
- **RF** - In the transmit path, it performs digital up conversion (DUC), RF impairments correction, and writes the transmit data to the RF. In the receive path, it reads the received data from the RF, performs digital down conversion (DDC) and performs RF impairments correction.

5.1.2 LTE AFW Host Software

Each host implementation interfaces with the bitfile that was built from the corresponding FPGA implementation. It demonstrates the main functionalities for each

implementation. This functionality includes configuration of the FPGA target, exchanging payload data, and monitoring the system status. The LTE AFW 2x2 MIMO application uses a UDP stream to transfer data between the host and the device or vice versa. Figure 5.3 shows the end to end data flow starting from the host, going through the data FIFOs set up in the FPGAs and back to the host again where the received data is displayed.

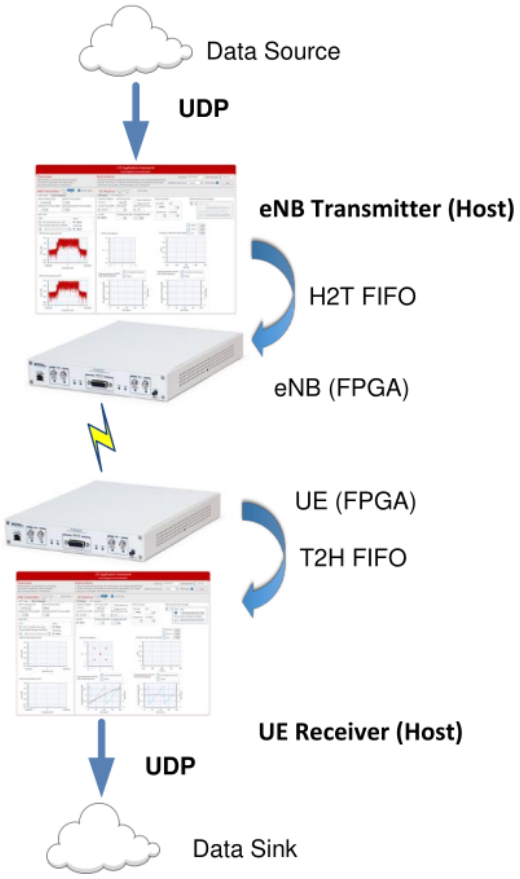


Figure 5.3: UDP Host-Device Data transfer

5.1.2.1 Host Software Modifications

The host software was heavily modified for the following reasons

- The version of the software given to MSV by NI was intended for an older

version *LabVIEW Communication System Design Suite 2.0* which was only supported for all OS releases until Windows 8.1/7 64-bit.

- The FPGA source was not issued to MSV instead a pre compiled FPGA bit file was delivered, but this corresponded to the **old** version namely *Labview Communications System Design Suite 2.0.*
- The project was not forwards compatible with the latest version of the Labview Communications System Design Suite.

As a result of the above mentioned points and to guarentee future compatibility, the project had to be ported to the latest version of *LabVIEW Communication System Design Suite 4.0*, which was renamed as *Labview NXG 4.0*. Missing dependencies had to be manually included and many driver related functionalities had to be modified as the newer version had a different driver implementation compared to the older version.

5.1.2.2 DL Transmitter

The host side of the DL transmitter acts as the setup interface for the eNodeB and initiates the FPGA, the RF Frontend and monitors the status of the transmitter as well provides visual feedback of the TX power spectrum. Figure 5.4 shows a screencapture of the TX Host panel.

The Table 5.1 shows the setup parameters used on the host.

Parameter	**Value**
Center Frequency	1.2GHz
TX RF Antenna Port On USRP	TX1/RF0
Cell ID	0
LTE Duplexing Mode	FDD
Transmitter Gain	10dBm

Table 5.1: Transmitter USRP Parameters Setup

5.1.2.3 DL Receiver

The DL Receiver implements the end to end receiver chain from receiving the raw data from the RF Frontend to the OFDM processing, demodulation and LTE processing steps. Most of the complex resource intensive data processing is done onboard the

Figure 5.4: A Screencapture of the DL TX(eNodeB) in the application window

FPGA as mentioned in Section 5.1.3. Certain parameters of the receiver can be manually configured according to user specifications. Two important features were added to the receiver processing on the Host, namely *data logging interface* and *Equalizer Bypass switch*. These are custom requirements for the purposes of the experiment in this thesis. The chosen settings for this experiment are listed in Table 5.2.

Parameter	**Value**
Center Frequency	1.2GHz
Transmit IP Address	127.0.0.1
UDP Transmit Port	60000
Receiver RX Antenna Port On USRP	RX2/RF1
Cell ID	0
LTE Duplexing Mode	FDD
Manual Receiver Gain	20dB

Table 5.2: Receiver USRP Parameters Setup

The DL RX (UE) mode screens are shown in Figures 5.5 and 5.6.

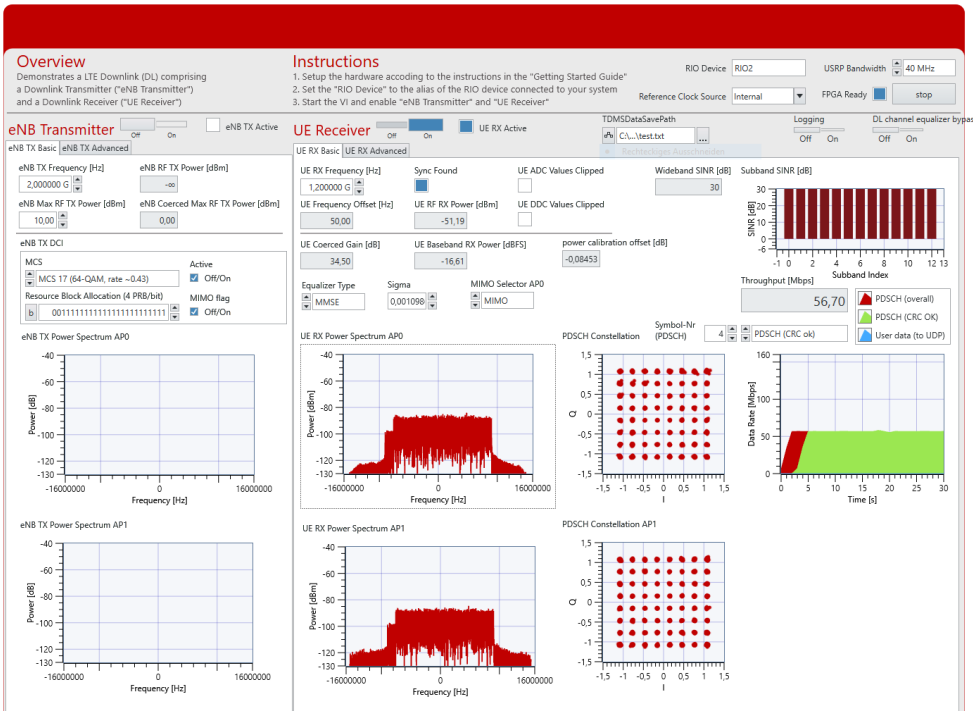


Figure 5.5: A Screencapture of the DL RX(UE) side application window

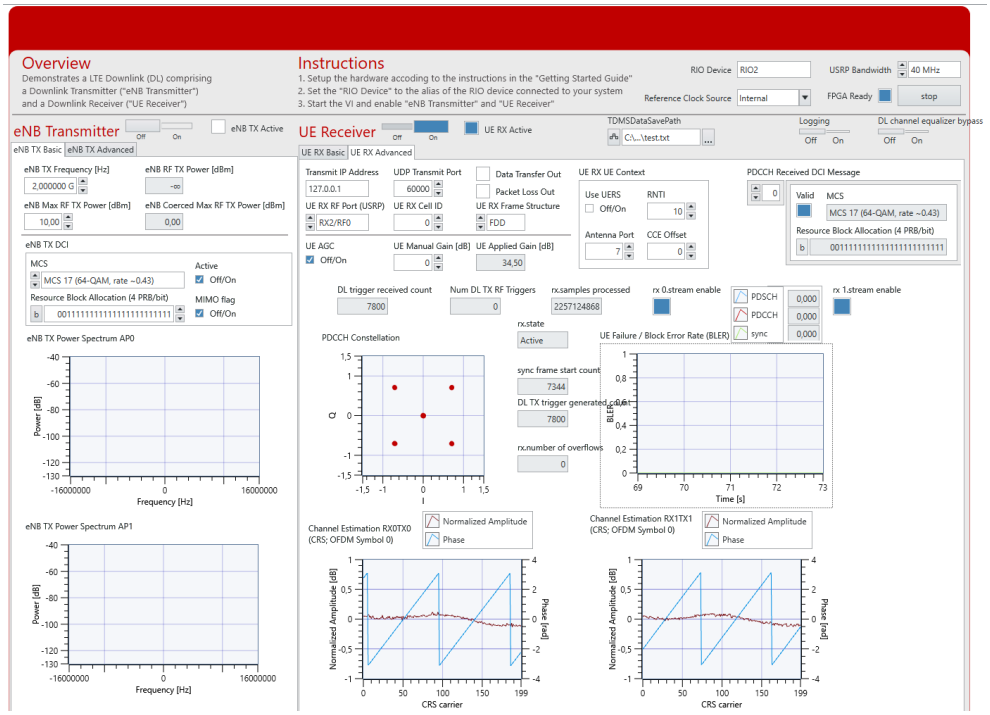


Figure 5.6: A Screencapture of the DL RX(UE) Advanced Options in the application window

Table 5.3 describes the various graphs and indicators shown on the RX screen of the application.

Indicator	**Description**
MIMO flag	Activates the eNodeB to run in MIMO setup. The MIMO flag is included in the DCI message to be deciphered at the RX part of UE.
Equalizer Type	Configures the algorithm that is used for equalization. The enumeration contains the following values: MMSE, ZF, and Simple.
Sigma	Selects the scaling factor that is used in the MMSE channel equalizer.
MIMO Selector AP0	Determines the PDSCH Constellation of AP0 that has been derived from SISO equalizer or MIMO equalizer.
eNB TX Power Spectrum AP0	Shows the power spectrum of the DL TX baseband signal transferred to the RF0/TX1.
eNB TX Power Spectrum AP1	Shows the power spectrum of the DL TX baseband signal transferred to the RF1/TX1.
UE RX Power spectrum AP0	Shows the power spectrum of the DL RX baseband signal received from RF0/RX2. If synchronization is successful based on the received signal from RX1 (RF0/RX2), frame timing and frequency offset correction are applied on the received signals from both ports RF0/RX2 and RF1/RX2.
UE RX Power spectrum AP1	Shows the power spectrum of the DL RX baseband signal received from RF1/RX2.
PDSCH Constellation of AP0	Constellation of RX IQ samples of AP0 allocated for PDSCH transmission after SISO or MIMO equalization. Only samples for the configured OFDM Symbol-Nr (PDSCH) are displayed.
PDSCH Constellation of AP1	Constellation of RX IQ samples of AP1 allocated for PDSCH transmission after MIMO equalization. Only samples for the configured OFDM Symbol-Nr (PDSCH) are displayed.
Channel Estimation of AP0	Graphical representation of the normalized channel amplitude and phase estimated based on the cell specific reference signals of AP0
Channel Estimation of AP1	Graphical representation of the normalized channel amplitude and phase estimated on the cell specific reference signals of AP1.

Table 5.3: Receiver USRP indicators description

5.1.3 LTE AFW FPGA

LTE has demanding and resource intensive processing requirements and in the application framework, the processing blocks for the DL and UL TX and RX are implemented directly on the FPGA. They exchange the baseband data with the RF interface using target-scoped FIFOs. The processing on the FPGA has advantages because it provides lower latency and therefore enables real-time physical layer processing. The following section describe the steps involved in the DL TX and DL RX processing done on the FPGA.

5.1.3.1 DL TX IQ Processing

The PDCCH contains the DCI(Downlink Control Indicator) message which describes the DL data transmitted to the UE. Due to the criticality of this message it is highly redudant and repeated multiple times in a frame and has a low modulation, inorder to recover the signal on the receiver side. PDCCH contains information like resource blocks allocated, MCS scheme, MIMO Mode, etc . . . which is vital for correct frame decoding.

PDSCH is the user data which is to be encoded into the frame and mapped to appropriate resource blocks. It includes the resource mapping that assembles all 1200 subcarriers of the current symbol. The TX IQ Processing module is triggered after the PDCCH and PDSCH I/Q samples for the current subframe are generated.

The index generator generates the timing information for each sample of the current OFDM symbol, such as subcarrier, resource block, OFDM symbol and subframe index. Depending on the current timing information, the index-to-channel mapping decides for each subcarrier, which reference signal or physical channel is mapped to it. The PSS sync sequence, the CRS and the UERS are precalculated and stored in a LUT. The PDCCH and PDSCH I/Q samples are read from a FIFO that was filled with all I/Q samples for the current subframe before the TX IQ Processing module was triggered. After combining all channels, the DC gap is inserted, and whitespace is added so that the resulting number of samples equals the FFT size of 2,048. The IFFT converts the frequency domain data into the time domain. Finally, the cyclic prefix is attached to the output of the IFFT. The resulting time-domain signal is transferred to the RF loop using a FIFO.

The PSS, CRS and the UERS are stored in the FPGA LUTs and fed into the iFFT block based on the symbol and subcarrier index.

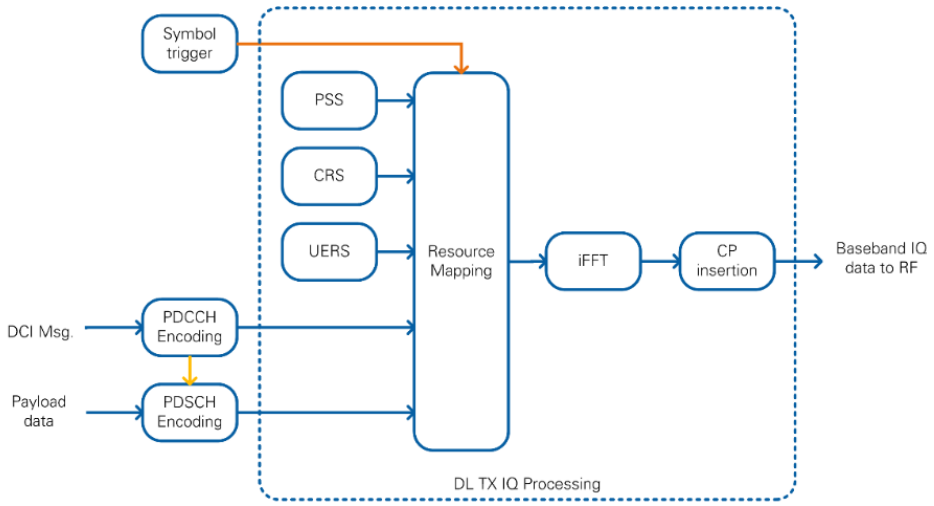


Figure 5.7: Simplified implementation of the DL TX processing block of the FPGA

5.1.3.2 DL RX IQ Processing

The RX IQ processing is more complex than the TX IQ processing chain as it performs very resource intensive calculations. This module reads the radio-frame aligned signal in time domain and outputs the channel-equalized subcarriers that are associated to the physical channels. As shown in Figure 5.8, it includes the following functional blocks

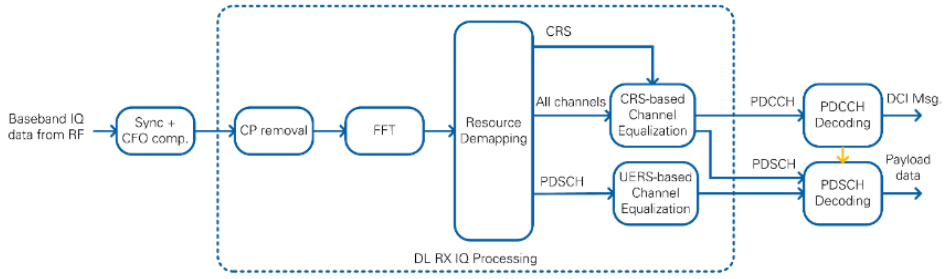


Figure 5.8: Simplified implementation of the DL RX processing block of the FPGA

- **Radio Frame Synchronization** - Synchronization is vital for the time alignment of the Frames as OFDM is extreamly time sensitive. CFO (Carrier Frequency Offset) correction is important to correct for any error/inaccuracies in the local oscilators. This could inadvertently move the frequency grid up or down based on the offset. Synchronization and CFO compensation are achieved by continuous

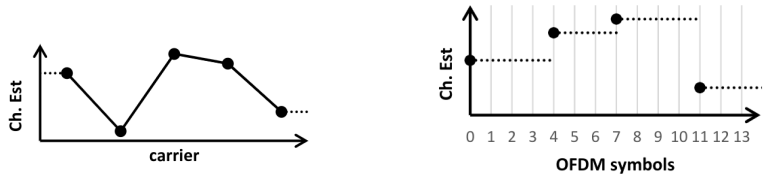
measurement of both an autocorrelation and a cross correlation. LTE signals contain a PSS, which is detected by two finite impulse response (FIR) filters (real and imaginary part) that calculate the cross correlation. This operation is executed on a reduced sample rate of 1.92 MS/s, which is the result of a decimation by 16. For each radio frame, the cross correlation peak is detected. To avoid misdetection, a validation unit checks that the peak amplitude is eight times higher than the average energy of the cross correlation. Additionally, three consecutive peaks are required, and the peak position may not drift more than five samples.

- **CP removal** - An internal FIFO is used to decouple the incoming samples from the rest of the processing chain. The throttle control module waits until enough samples for one complete OFDM symbol (FFT size + CP) are available before it passes them as a consecutive stream to the next modules. The CP removal removes the valid flag from the samples belonging to the cyclic prefix. The 2,048 remaining samples are sent to a Xilinx FFT. For a full 20 MHz BW LTE system the number of samples in a subframe is 2048 plus the CP samples that are added during the TX process. Once the CP is removed the remaining 2048 Samples are forwarded onto the FFT Conversion module.

- **FFT Conversion** - The output of the FFT are 2048 the subcarriers in frequency domain.

- **Resource demapping** - The resource demapper selects the 1200 allocated subcarriers by removing the surrounding whitespace and the DC carrier in the center. Then it generates the timing information for each sample and the resource grid by marking each sample for its corresponding channel by using a Boolean cluster. The resource mapping is based on a fixed frame structure configuration described in the LTE specifications. All subsequent modules use this Boolean cluster with elements for each LTE channel to determine if this sample is relevant.

- **CRS based channel estimation and equalization** - The FFT output data is fed into two separate channel estimation blocks running in parallel. The first channel estimation is based on the CRS. The channel estimate values are calculated by conjugate complex multiplications. A linear interpolation is applied in frequency domain between adjacent reference symbols as explained in Section 3.1.2. On the edges of the symbol, the nearest estimated value is replicated (zero order hold). OFDM symbols not containing CRS sequences rely on the last channel



(a) Channel Estimation Interpolation over Frequency

(b) Channel Estimation Zero order hold over time

Figure 5.9: Illustration of channel estimation interpolation over time and frequency

estimation (zero order hold in time) as shown in Figure 5.9. One important flaw inherent to the hardware is the phase synchronisation between the RF front ends. The problem is explained extensively in Section A.2. This phase difference although existing on the transmitter side, can be corrected on the receiver side as the hardware based phase offset is constant for every run of the application. The phase offset compensation happens within FPGA in the channel estimation sub section. Once calculated the phase correction is applied to all the subcarriers and time symbols of the frame.

- **UERS based channel estimation and equalization** - This signalling scheme is available for use but not actively used in this thesis.

5.2 Application Example

Once the above mentioned apparatus was setup, it was ready to be used to log data. At the time of writing this thesis there was another Bachelor thesis being conducted in parallel to use machine learning to reverse the effects of the Channel distortions.

One of the main topics of communication technology is to reconstruct data at the receiver that was transmitted over a non-trivial channel. Figure 5.10 shows the ML model being used to train the inverse neural network to retrive the sent data. It is only necessary to feed the trained network with the received signal y and it then computes the original signal x. The transmitted signals x are mapped to the received signals y in the natural way and then the origial signals are reconstructed by inverting this mapping in the online phase. The mapping is done with the help of invertible and composable transformation.

The standard way of using Neural Networks for this task is to map the received

signals y to the transmitted signals x [4]. For training the neural network, transmitted and received symbols and their corresponding Subband Noise are required.

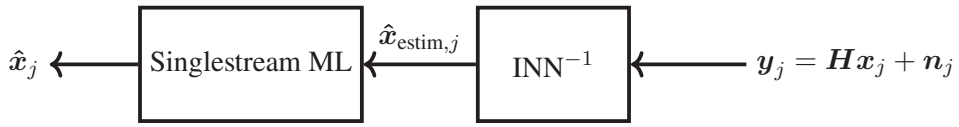


Figure 5.10: ML Model for learning the network

5.2.1 Transmitted and Received Data

With the Host applications modifications mentioned in Section 5.1.2.1, the unequalized data was received and recorded by the RX USRP. Due to Host performance degradation (details in Section 6.1) custom user data was susceptible to data loss.

Given the time constraints modifying the FPGA bit file was not an option as this step involved recompiling the whole FPGA project. Once compiled the bitfile had to be tested and verified and given the tight timeframe. Therefore a decision was made to use the CRS symbols as the (x,y) pairs that were transmitted and received. This was the only signal that was forwarded from the FPGA to the host, that was easy to use, already frame synchronised and spanned the whole bandwidth.

Figure 5.11 shows that the CRS Ant 1 symbols are intended for the frame to be sent on TX Port 1 where as the CRS Ant 2 symbols are intended for the frame to be transmitted on TX Antenna 2. The greyed out resource blocks are meant as placeholders where no data can be sent on them.

On the receiver side RX antenna 1 receives the frame, once decoded the distortion of the CRS Ant 1 symbols can be inferred from h_{11} channel estimation parameters. Similarly from the decoded frame on RX antenna 2, the distortion of the CRS Ant 2 symbols can be inferred from the h_{22} channel estimation parameters. This functions as our (x,y) pair as the equalization can be bypassed.

5.2.2 Subband and Wideband Noise

The SINR estimation algorithm is based on filtering the potentially noise channel estimates derived for the CRS subcarrier or the UERS subcarrier, respectively.The noisy least squares (LS) channel estimates obtained for the reference symbol carriers are filtered by a de-noising low-pass filter to obtain LS channel estimates with reduced

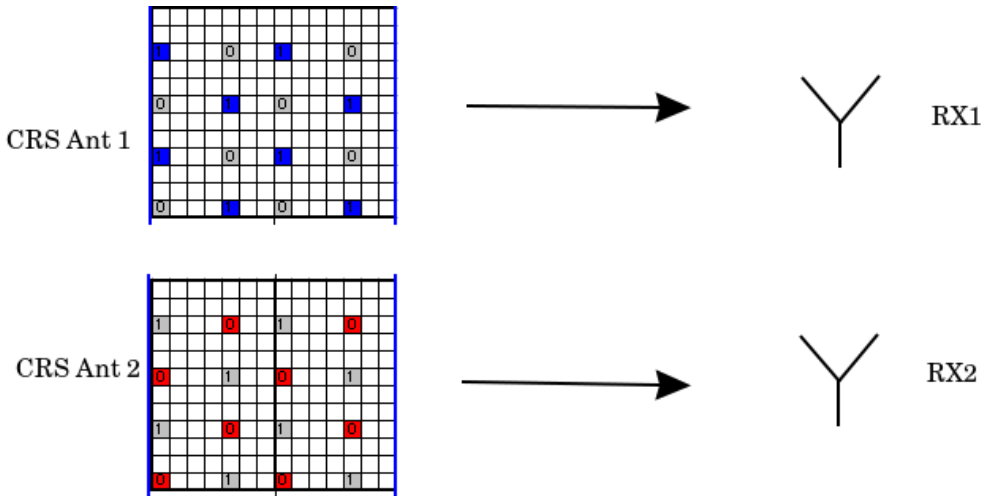


Figure 5.11: Illustration of the CRS signals and channel estimation coefficients used as the (x,y) pair

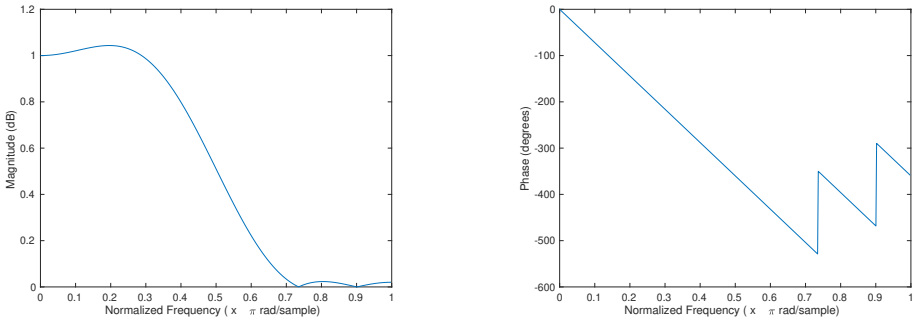
noise. The implemented prototype de-noising filter is a raised cosine filter with nine taps and the filter coefficients in the following Table 5.4.

The SINR value is calculated per subband, each of which occupies 8PRBs. so SINR0 is subband SINR for PRBs 0. . . 7, SINR1 is for PRBs 8. . . 15, etc. . . It has a fixed point format with 8 bits signed fixed-point number with 6 integer and 2 fractional bits (range -32.00 dB to +31.75 dB)

According to NI, to further improve the provided SINR estimates an additional look-up table based fine-calibration stage was implemented in the application framework. The underlying look-up table has been empirically derived by fine-calibration measurements.

Coefficient Index	Coefficient Values
0	0
1	-0.061235467
2	0
3	0.306177333
4	0.510116268
5	0.306177333
6	0
7	-0.061235467
8	0

Table 5.4: FIR filter Coefficients for filtering out the Channel Estimation values



(a) Magnitude Response

(b) Phase Response

Figure 5.12: Frequency response of the filter in Table 5.4

All the code to this project can be found on a publicly visible the Github page [1]. The instructions on the structure of the the project and how to run the applications can be found in the README.

[1] https://github.com/karthick1987/MIMOTestJig

6 Results

Over the course of the internship many different parameters had to be determined and set up for the final demo. This chapters documents the results of all the experiments performed as well as the final demo of the working setup.

6.1 Over-the-air User Data Transmission

Following the data transmission through over the air using LTE AFW, benchmarks were run to check the loss of data over the wireless medium. Data loss tests were profiled using *iperf*, a software used to profile UDP traffic.

A server and client of the version of the iperf application was used to test the wireless interface. Where the data was produced by the client on UDP port `50000` and received on the localhost `127.0.0.1` on UDP port `60000` once a SISO and MIMO Tx-Rx wireless interface was setup for the respective tests. The version of iperf used is shown below.

```
iperf version 2.0.9 (1 June 2016) pthreads
```

The command used to start up a Client is shown below.

```
iperf.exe -c 127.0.0.1 -u -i 1 -b 10M -t 10 -p 50000
```

The command used to start up a server is shown below.

```
iperf.exe -s -u -B 127.0.0.1 -i 1 -p 60000
```

6.1.1 SISO

A SISO communications channel was setup with the Tx and Rx settings as mentioned in Table 5.1 and 5.2 using the LTE AFW template. On conducting UDP data flow tests, it was observed that there were no packet drops with a UDP data block loss rate of 0%.

As a next step a short video clip was fed into the UDP stream of the transmitter and received with no video and audio frame drops at all. The demo video can be seen here [1].

[1] `https://drive.google.com/file/d/1PFWkfSUSSI4UC-m7nrF5EjYPyTy7iTYG/view?usp=sharing`

6.1.2 MIMO

In the second test a MIMO communications channel was setup with the Tx and Rx settings as mentioned in Table 5.1 and 5.2 using the LTE MIMO AFW 2x2 extention software. On conducting UDP data flow tests, it was observed that there were large packet losses with a UDP data block loss rate of around 25%. Below is an output of the server client iperf application.

```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\ge69mog>cd Downloads\iperf-2.0.9-win64\iperf-2.0.9-win64

C:\Users\ge69mog\Downloads\iperf-2.0.9-win64\iperf-2.0.9-win64>iperf.exe -s -u
-B 127.0.0.1 -i 1 -p 60000
------------------------------------------------------------
Server listening on UDP port 60000
Binding to local address 127.0.0.1
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 127.0.0.1 port 60000 connected with 127.0.0.1 port 49721
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec   996 KBytes  8.16 Mbits/sec   2.017 ms  166/  860 (19%)
[  3] 0.00-1.00 sec  50 datagrams received out-of-order
[  3]  1.0- 2.0 sec   904 KBytes  7.41 Mbits/sec   2.426 ms  227/  857 (26%)
[  3] 1.00-2.00 sec  70 datagrams received out-of-order
[  3]  2.0- 3.0 sec   902 KBytes  7.39 Mbits/sec   2.148 ms  219/  847 (26%)
[  3] 2.00-3.00 sec  61 datagrams received out-of-order
[  3]  3.0- 4.0 sec   916 KBytes  7.50 Mbits/sec   1.904 ms  222/  860 (26%)
[  3] 3.00-4.00 sec  69 datagrams received out-of-order
[  3]  4.0- 5.0 sec   871 KBytes  7.14 Mbits/sec   4.137 ms  206/  813 (25%)
[  3] 4.00-5.00 sec  56 datagrams received out-of-order
[  3]  5.0- 6.0 sec   953 KBytes  7.81 Mbits/sec   4.710 ms  226/  890 (25%)
[  3] 5.00-6.00 sec  86 datagrams received out-of-order
[  3]  6.0- 7.0 sec   828 KBytes  6.79 Mbits/sec   2.786 ms  253/  830 (30%)
[  3] 6.00-7.00 sec  84 datagrams received out-of-order
[  3]  7.0- 8.0 sec   843 KBytes  6.90 Mbits/sec   4.467 ms  250/  837 (30%)
[  3] 7.00-8.00 sec  90 datagrams received out-of-order
[  3]  8.0- 9.0 sec   970 KBytes  7.95 Mbits/sec   1.648 ms  200/  876 (23%)
[  3] 8.00-9.00 sec  61 datagrams received out-of-order
[  3]  0.0-10.0 sec  8.86 MBytes  7.43 Mbits/sec   2.888 ms 2183/ 8505 (26%)
[  3] 0.00-10.00 sec  689 datagrams received out-of-order
```

The same short video clip that was used in the SISO case was fed into the UDP stream of the transmitter and received with large distortion and missing several frames of the video and audio.

It was inferred as a result of the tests that this loss stems from the CPU overhead in processing 2 simultaneous UDP streams of data, or the handling of UDP data by the

LTE AFW 2x2 MIMO extension application.

6.2 CRS Data Transmission

As seen from the results of the previous section, the data loss in the user data sent via UDP was not a feasible option for the purposes of this experiment as data had to be received any losses. As a result, it was decided to use the CRS symbols as sent and received data pairs. This was done by utilising h_{11} and h_{22} as received unequalized symbols. Figure 5.11 shows that the h_{11} channel coefficients correspond to the CRS signal arranged in the grid for antenna port 0 and h_{22} channel coefficientes correspond to the CRS signals arranged in the grid for antenna port 1. Since this data is processed in the FPGA fabric, it is real time and there is no lost CRS symbol being forwarded to the Host.

The sending pattern is fixed as it is defined in the standard and the corresponding received CRS symbols can be obtained from the h_{11} (Channel estimate of channel between transmit antenna 1 and receive antenna 1) and h_{22} (Channel estimate of channel between transmit antenna 2 and receive antenna 2) signals respectively.

A sequence of 8342 time CRS symbols were recorded over the complete $20\ \mathrm{MHz}$ bandwidth of the LTE spectrum. This corresponds to 200 subcarriers which bear the reference pilot symbols. Hence a matrix of size 200×8342 was generated for the training of the inverse neural network.

6.3 Inverse Neural Network Detection

Following the generation of experimental data, one particular subcarrier was chosen as training data set ,i.e. one row of the above mentioned Matrix. The first CRS subcarrier was chosen

And the related wideband SINR for each time symbol was also recorded, this was the second input needed by the training network (as mentioned in Section 5.2.2).

6.3.1 INN Constellation

Figure 6.1 shows the input CRS signals added with random received noise wideband noise for the different time instants. If the signal was added with the wideband noise, then there are infinite mapping possibilties between the input and the output, as one point could map to a set of infinite continuous point on the output. And this would be a NP Hard problem. To avoid this the approach of adding the wideband noise was taken to create a clusters of inputs and corresponding outputs to form a bijective mapping.

This was computable and could be used by the network. x_1 and x_2 are the real parts of the CRS signals for antenna port 0 and antenna port 1 respectively. These are QPSK signals that can take the possible values $\pm\frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}}$ each. Just using the real parts we have 4 possibilites where x_1 and x_2 can assume the value of $\pm\frac{1}{\sqrt{2}}$ each. But due to the inherent structure of the CRS symbols in the LTE standard, only one of the possible above combinations is found namely, +$\frac{1}{\sqrt{2}}$,-$\frac{1}{\sqrt{2}}$.

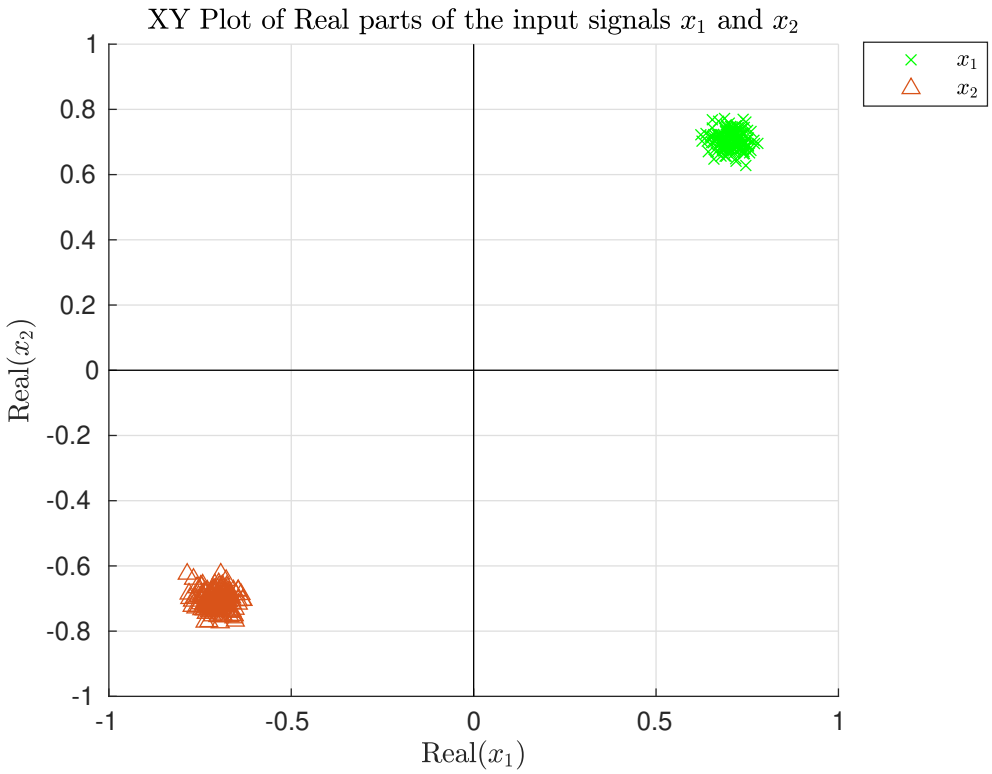


Figure 6.1: XY plot of the real parts of the input signals x_1 and x_2 with noise

The Figure 6.2 shows the received channel estimates $y_1 = h_{11}$ and $y_2 = h_{22}$. As power of this signal is highly attenuated as the air interface leads to a lot of losses. The phase shift in the various samples, represents the channel distortion effects.

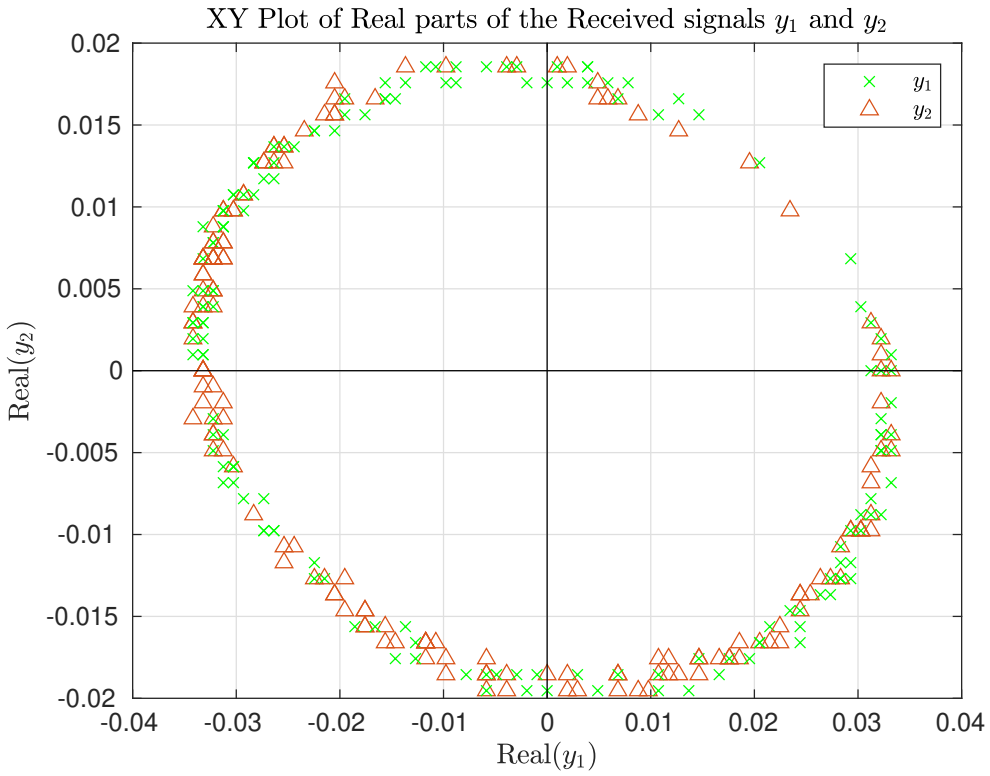


Figure 6.2: XY plot of the real parts of the received signals y_1 and y_2

The Figure 6.3 is the output of the INN which estimates the signals y based on the training data the neural network was fed with. On clustering the input and the estimated data in each of the 4 quarters of the XY plot, it was found that the detection accuracy was close to **22**%.

It is to be noted that only one pair of input data namely $+\frac{1}{\sqrt{2}}$,$-\frac{1}{\sqrt{2}}$ was available instead of all 4 possibilities namely $\pm\frac{1}{\sqrt{2}}$,$\pm\frac{1}{\sqrt{2}}$. This was due to the fact that the CRS symbols chosen as the (x, y) transmit and receive pair (as mentioned earlier) are in fact a predefined sequence as per the LTE standard.

One particular reason for the low detection rate could potentially be because of the complexity of the channel on the one hand and the complexity of the neural network on the otherhand. It is possible to achieve better results with some fine tuning of the neural network.

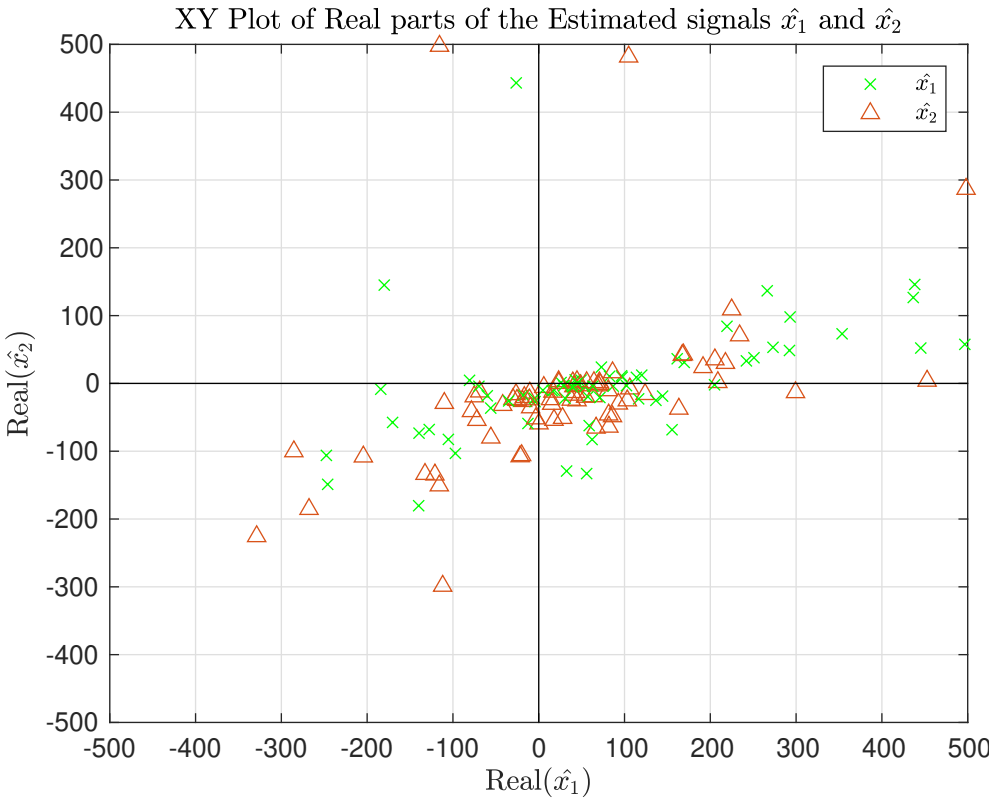


Figure 6.3: XY plot of the real parts of the INN Estimated signals $\hat{y_1}$ and $\hat{y_2}$

7 Conclusion and Outlook

A partially functioning MIMO experimental setup was achieved in this thesis. Firstly it is partial because of the dropped user data packets and secondly as its not a truely synchronised transmitter system. As explained in Section 5.1.3.2, the hardware is not a true synchronous transceiver meaning that the 2 transmitting antennas are not time synchronised due to the phase differences in the RF paths. This is an extremely limiting factor in the ability to run custom MIMO experiments. Also given the limitations in the software and hardware delivered to MSV from NI, this was the best possible outcome with the hardware at hand. However with the modification listed below, better results could be achieved.

7.1 FPGA File Port

Currently the FPGA Hardware design is where all the data is being received from the RF Front end and processed in real time. This FPGA hardware file is a pre compiled bit file delivered by NI to work with the older version of the LTE AFW Software. To adapt the design to suit the needs of the current research, the FPGA bit file has to be ported over to the latest version of Labview NXG to suit the design needs of the team.

7.2 Transmitting User Defined Data

Once the FPGA bit file has been modified, a user defined pattern can be transmitted through the shared data channel (PDSCH) instead of the CRS signals. The CRS signals were chosen as they were only a work around, and they enabled quick prototyping of the MIMO system without FPGA bitfile modification.

To be able to use the PDSCH channel, the implementation has to be changed to eliminate the data loss on the MIMO systems. Once the source of the packet drop has been identified, a pseudo random pattern with a fixed seed can also be generated and sent over the air. Finally there is also the possibility of sending custom bit streams over the air.

7.3 Experiments with structured Channel

Structured channels could first be built using reflectors, instead of having a single Line of Sight (LOS) channel. The structure of the channel can be verified (using a Singular Value Decomposition of the channel matrix) if and only if all the channel coefficients are available. Once this channel has been created, experiments can be run for with this channel.

7.4 Wideband Antennas

Figure 7.1 shows an omni directional Antenna from Huber and Suhner (SWA-2459/360/4/45/V), which is capable of transmitting and receiving in the frequency range of 2400MHz to 5875MHz. The antenna also has an antenna gain of 4dBi which is important to compensate for some losses. These experiments can be useful potentially in the higher frequency bands. Although the USRP2940 can only transmit upto 2.2GHz, it can be replaced with other USRPs as defined in Table 4.5 that can transmit up to 6GHz

Figure 7.1: Huber and Suhner SWA-2459/360/4/45/V widebadn Antenna

Troubleshooting A

A.1 Boot Order

The communications protocols between the USRP and the host follows the PCIe standard. It is essential that the drivers are loaded up during boot up. It is highly advised against hot swapping or unplugging PCIe cables while the host is booted up and running. Secondly care must be taken while booting up the system by following the sequence recommended by NI in order to avoid synchronisation issues.

1. Connect the PCIe cables from the PCIe port of the USRPs to the PCIe cards mounted on the Host (section 4.1.1)
2. Power up the USRPs once the above step has been completed
3. Power up the PC and wait for the device to boot up
4. Check the NI MAX (NI's software for managing NI devices) Software to see if the devices have been recognised by the PC
5. When all the USRPs appear on the NI MAX they are ready to be used

A.2 Synchronisation of the USRPs

USRPs are complex devices which combine analog tranceivers and digital Xilinx Kintex-7 FPGA, which does all the compute intensive tasks. Figure A.2 shows a block diagram of a **single** RF channel with the internal components of the USRP. There are two such channels in a given USRP 2940.

The figure shows that a single RF channel can be used either as a single antenna transmitter (**TX1**) or a single antenna receiver (**RX1/RX2**). RX1 and RX2 **cannot** be used simultaneously.

The 2 radio channels within a USRP are either synchronised using the REF IN signal which is a 10 MHz supplied by an external clock distributer (section 4.2.3) or by using the internal 10MHz oscillator. They share the same reference clock for all the

PLLs and the VCOs as illustrated in figure A.2. It was inferred from tests that the RF path delay for the Low Noise Amplifiers (LNA) are not matched between the channels in the USRP and therefore leading to phase shifts between the 2 RF channels on the single USRP. This results in unsynchronous transmit and receive data between the channels which is not the preffered mode of operation for a MIMO setup as receive and transmit synchronisation is key. Furthermore the delay is not a constant for every new run of the system, rather its a new delay for every rerun. This is the limitation of the hardware and caused by delays in the RF analog path. The delay does remain constant within the duration of a run.

The Octoclock provides a common 10 MHz sampling clock which all the USRPs synchronize to. The Octoclock also has a PPS Trigger which is a pulse per second trigger which is used as a sampling trigger for different USRPs.

Figure 3. Synchronizing Multiple USRP RIO Devices with the CDA-2990

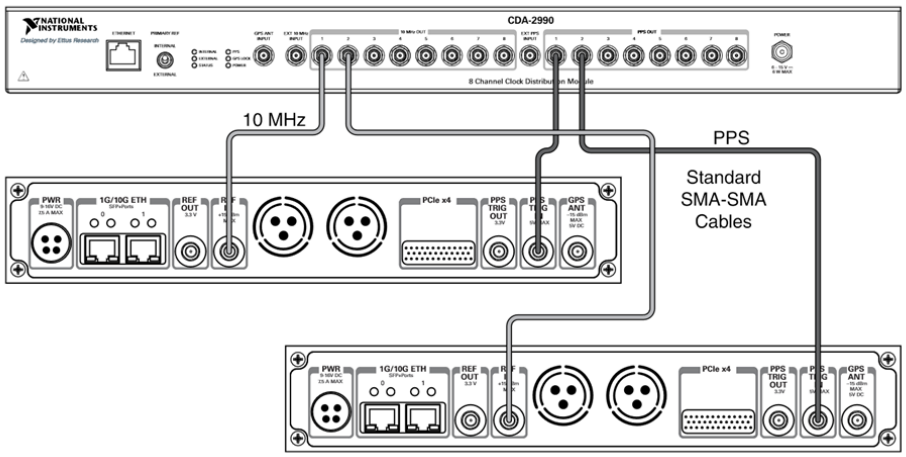


Figure A.1: The connections from the Octoclock to the USRP

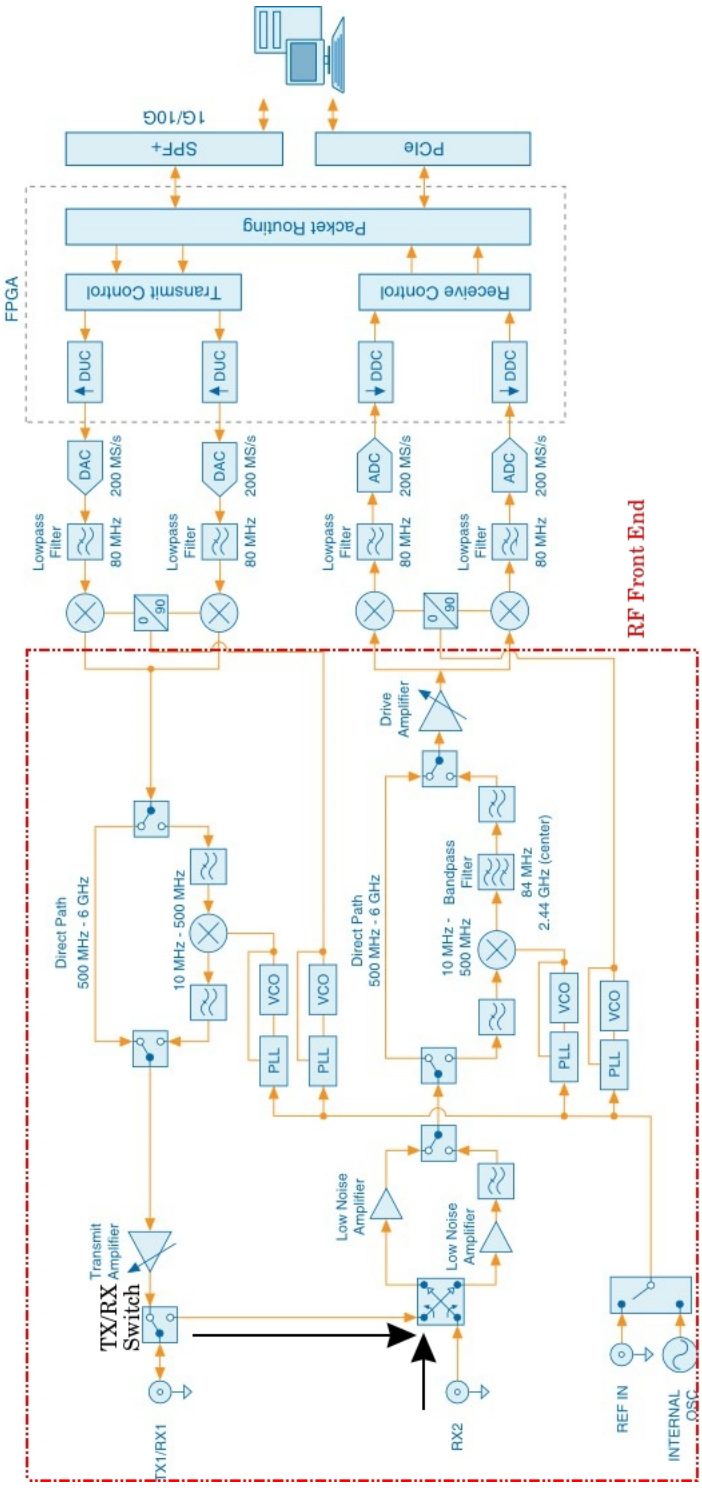


Figure A.2: USRP 2940 Internal Components

The following section describes the test setup used to verify the functional specification of the Octoclock. Figure A.3 show the REF IN coming in from 2 different ports of the octoclock each terminated with a 50 Ohm resistor and measured with an oscilloscope. The experiment was carried out to verify the synchronicity of the reference clocks from two different ports of the octoclock. The FFT of the signal also shows that the tone lies at 10 MHz.

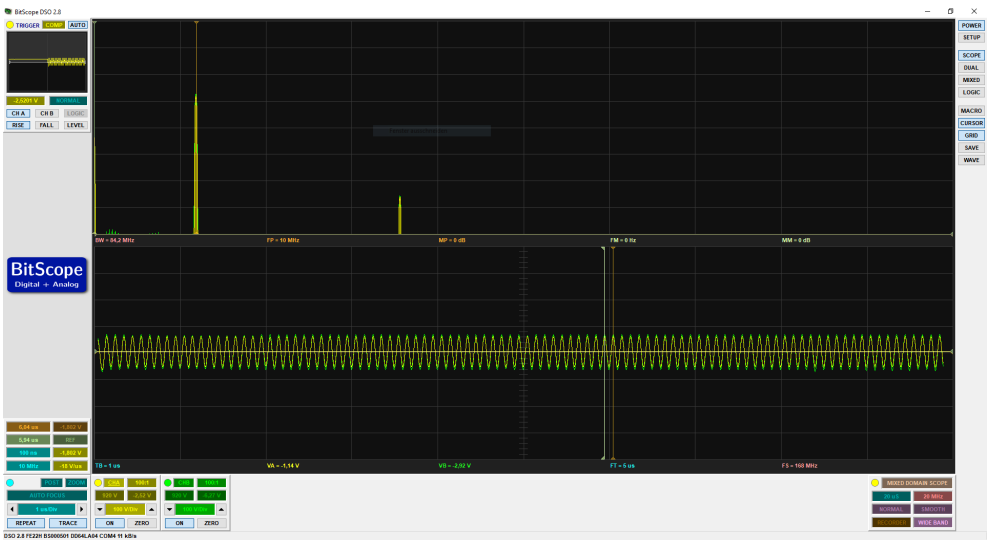


Figure A.3: The 10 MHz REF IN clock as seen at the output of two respective channels of the Octoclock. The yellow is from ch1 and the green is from channel 2 which are connected to USRP1 and USRP2 respectively. The top half is the Fourier transform and the bottom half is the time domain signal

Figures A.4 and A.5 show 2 PPS (Pulse per Second) triggers received from 2 ports of the octoclock each terminated with a 50 Ohm resistor and measured using an oscilloscope. The signals found to be aligned and have a rising edge every second as we expect.

It can be inferred from the design of the Octoclock B that the REF IN is internally generated by the Octoclock and distributed among all the ports. Alternatively there is an option to feed the an external reference input to the octoclock and distribute this reference among the other ports.

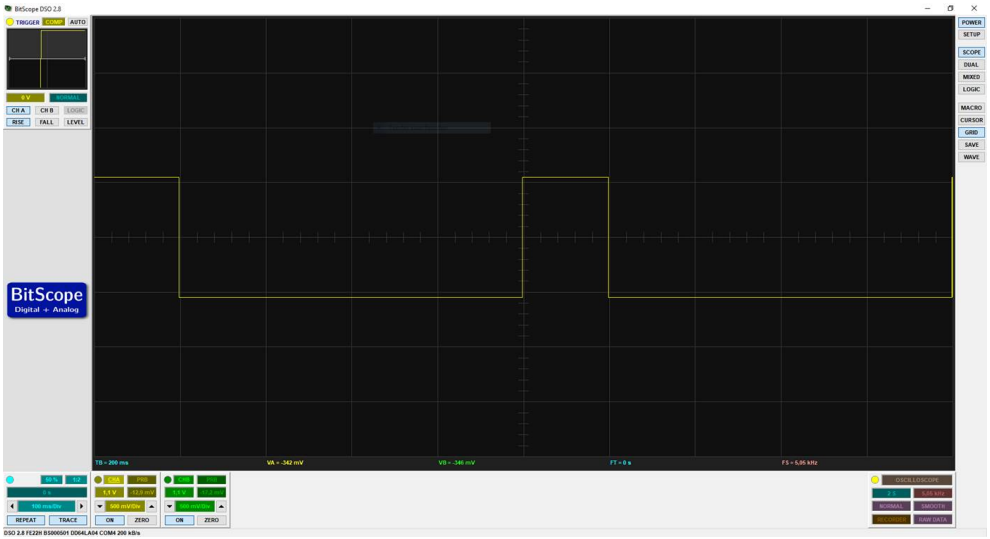


Figure A.4: Figure shows a zoomed-out view of the PPS Trig Out which occurs every 1s and the high pulse is 200ms long.

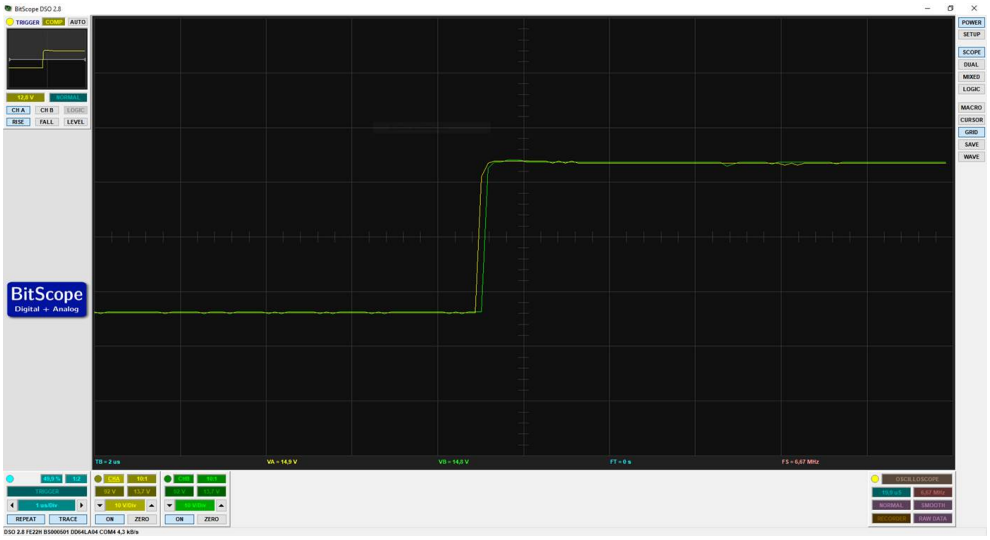


Figure A.5: Figure shows the PPS trigger signal from the Octoclock. The slight shift is potentially an artifact from the measurement instrument as the bandwidth of the measurement tool was limited (20MHz).

Finally after verifying the Octoclock functionality, a test was devised to measure

the synchronisity of the Tranceiver. For this 2 USRP devices were connected as shown in A.1 and tests were run in loopback mode where the TX1 of each RF channel was connected to the respective RX2 of the same RF channel. A pure tone is sent as IQ samples from each TX1 port of each USRP 1 and USRP 2. This test was conducted by using 1m 50 Ohm cables to eliminate the air interface. Matched length cables avoid phase delays and are critical in this measurements.

The function sent at the transmitter side is described as follows

$$y(t) = cos(\omega t) + j * sin(\omega t) \tag{A.1}$$

where the I Signal is $cos(\omega t)$ and the Q Signal is $sin(\omega t)$.

Ideally the same signal should be received on all RX ports, where as the signals are received with a relative random phase shift. Figure A.6 below shows the results of the tests showing the phase difference when one USRP transmits the same sine tone on the 2 respective TX1 channels of the 2 USRPs, while the signals are received on the respective RX2 channels of each RF channel.

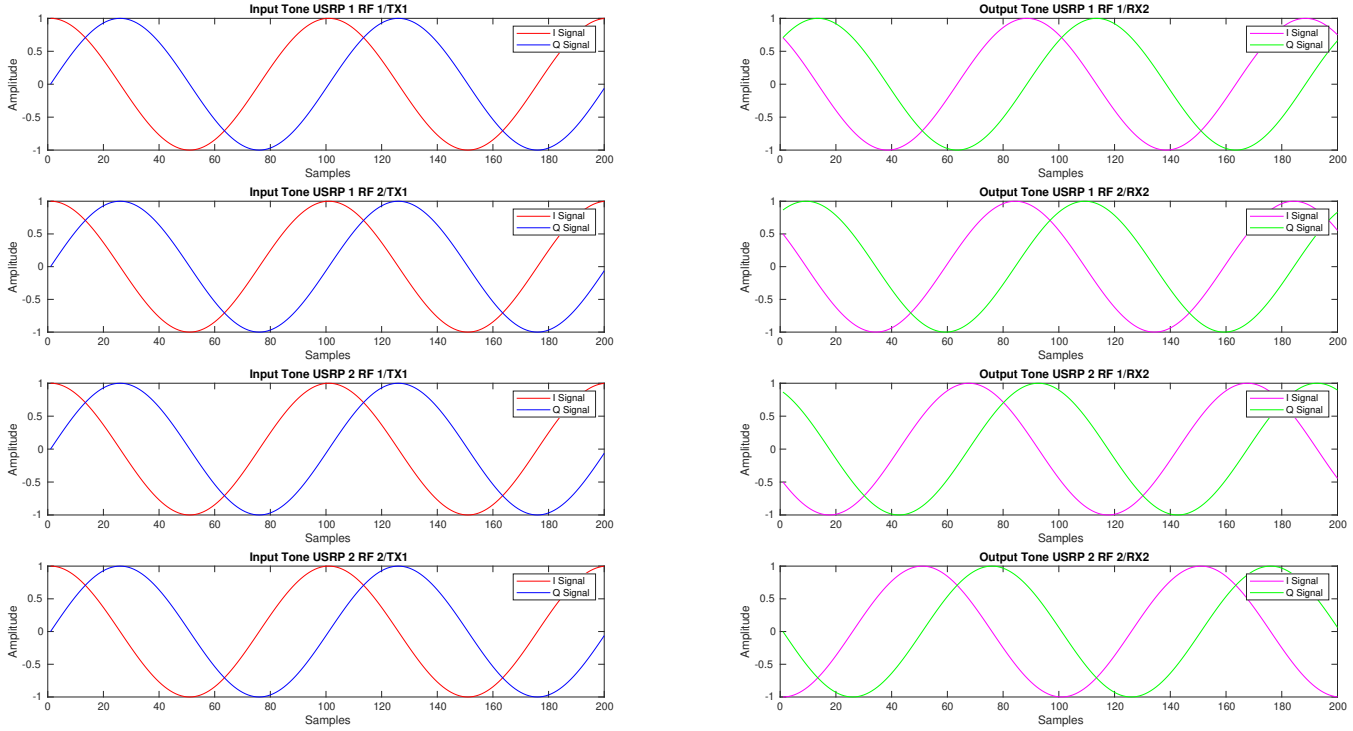
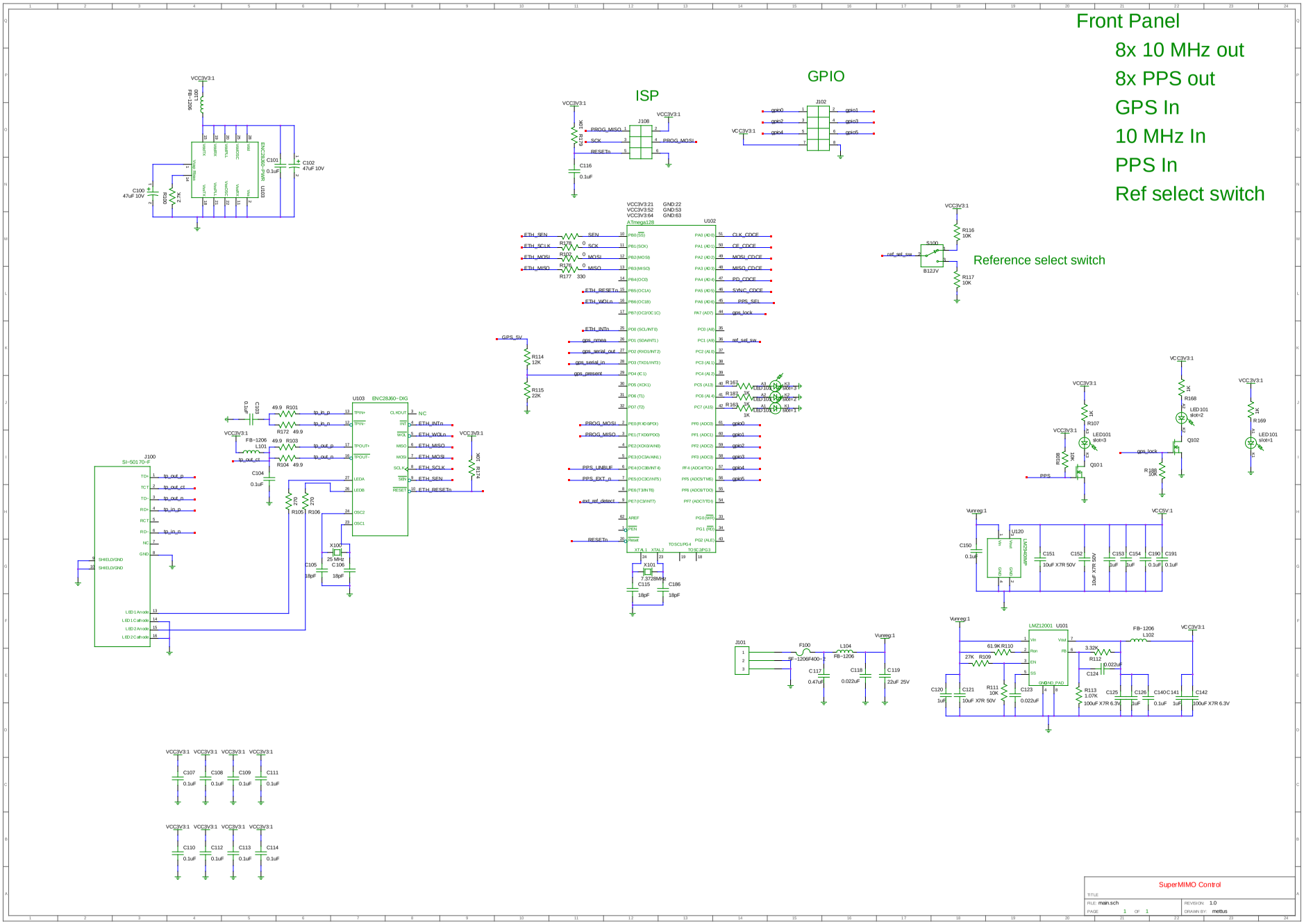


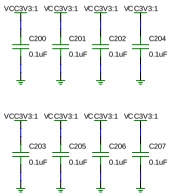
Figure A.6: The left waveforms are the TX waveforms and the right waveforms are the RX waveforms. It can be seen that the received IQ waveforms are unsynchronised and clearly have a phase shift

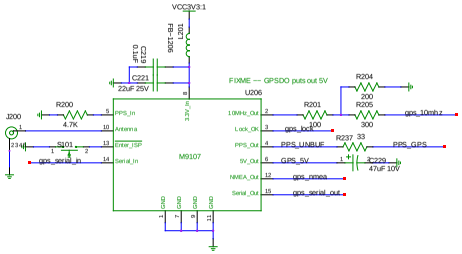
Schematic Octoclock B

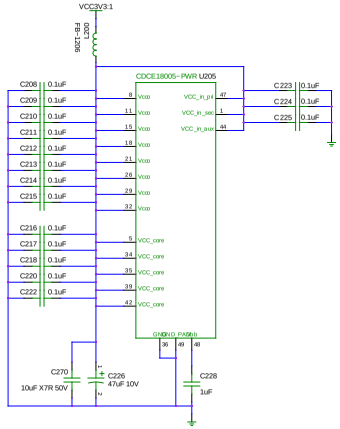

Front Panel
8x 10 MHz out
8x PPS out
GPS In
10 MHz In
PPS In
Ref select switch
ISP
GPIO
Reference select switch
U102
ATmega128
U103
ENC28J60-DIG
J100
SI-50170-F
U120
LM2936MP
U101
LMZ12001
SuperMIMO Control
TITLE
FILE: main.sch
PAGE 1 OF 1
REVISION: 1.0
DRAWN BY: mettus

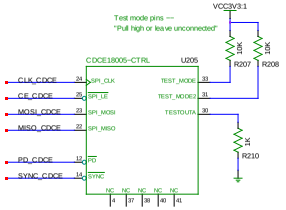
Front Panel

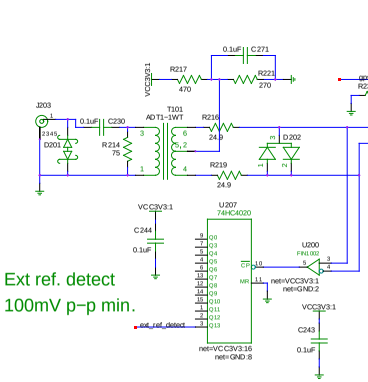
- 8x 10 MHz out
- 8x PPS out
- RS232 out
- GPS In
- 10 MHz In
- PPS In

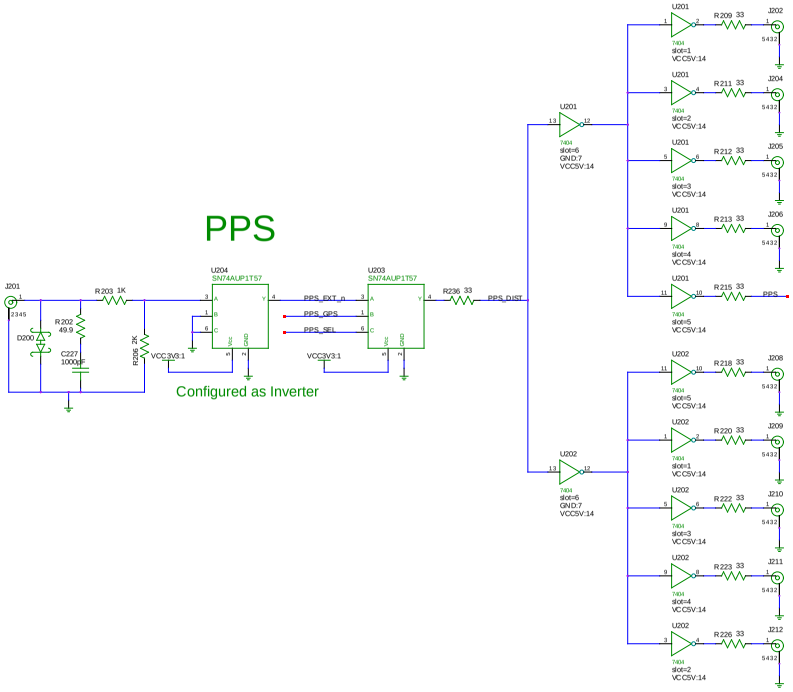


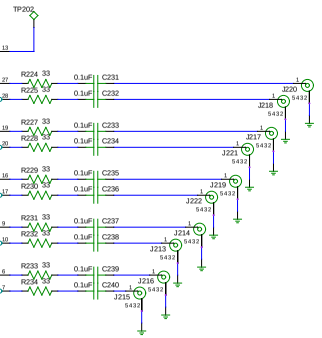












SuperMIMO Control

TITLE	
FILE: main.sch	REVISION: 1.0
PAGE 1 OF 1	DRAWN BY: mattus

Bibliography

[1] W. Utschick, "Anwendungen in der Informationstechnik – Grundlagen der Mehrfachzugriffsverfahren I und II."

[2] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li., "Channel Estimation for OFDM," Jun. 2014.

[3] S. Coleri, M. Ergen, A. Puri, and A. Bahai., "Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems," Jan. 2002.

[4] J. Maas, "Invertible Neural Networks for MIMO Detection," Master's thesis, Technical University of Munich, 2020.

[5] H. Rohling, *OFDM - Concepts for Future Communication Systems*, 3rd ed. Springer, Jun. 2011.

[6] 3GPP, "TS 36.211 Technical Specification Group Radio Access Network; Physical Channels and Modulation (Release 13), V13.2.0," vol. 13, Jun. 2016.

[7] K. Fazel and S. Kaiser, *Multi-Carrier and Spread Spectrum Systems*, 2nd ed. Wiley, Jun. 2008.

[8] R. M. Gray, "Toeplitz and CirculantMatrices: A review," Deptartment of Electrical Engineering, Stanford University, Stanford 94305, USA, Tech. Rep.

[9] *MIMO Prototyping System Getting Started Guide*, 2016.

[10] *LabVIEW Communications LTE Application Framework 19.5*, 2016.