TECHNICAL UNIVERSITY OF CRETE
Department of Electronic and Computer Engineering

Software-Defined Radio Impementation of OFDM

Author:
Stylianos Leonidakis

Supervisor:
Prof. Athanasios Liavas

August 2014

Acknowledgements

First of all, i would like to thank my supervisor, Professor Athanasios Liavas for suggesting me with such an interesting topic and for his overall guidance and suggestions throughout my thesis. I would also like to thank my family for being so understanding and supportive. Last but not least, i would like to thank my friends for their invaluable help and support throughout my student years.

Abstract

Wireless communications is by any measure the fastest growing segment of the communications industry. Cellular phones, Wireless Local Area Networks (WLANs), Smart Homes and telemedicine are some applications that make extensive use of wireless communications.

Orthogonal Frequency Division Multiplexing (OFDM) is a modulation that has been applied to a wide variety of applications and is adopted in many recently standardized broadband communication systems. OFDM is an appealing technique for achieving high-bit-rate data transmision, due to its robustness against the multipath fading and Inter Symbol Interference. Since the OFDM transmission is vulnerable to time and frequency offsets, accurate estimation of these parameters is one of the most important tasks of the OFDM receiver. In this thesis, we implement a time synchronization algorithm based on correlation of two identical parts of an OFDM pilot symbol. In addition, knowledge of the channel in the transmitter is very useful for effective transmission and/or interference mitigation (especially in static environments). Thus, channel feedback is utilized throughout the experiments conducted for this thesis.

The experiments were conducted on a software-defined-radio (SDR) testbed based on the Universal Software Radio Peripheral (USRP).

Contents

List of Figures

Chapter 1

Introduction

Wireless Communication

Wireless communication is the transfer of information between two or more points that are not connected by an electrical conductor. The most common way of wireless communication is by electromagnetic waves.

In the past decades, wireless communication systems gained enormous popularity over wired systems, mainly because they are in many applications much cheaper to implement. Besides, in places where the environment hampers cable deployment, wireless connection remains the only means of communication.

1.1 Multicarrier Modulation

Digital bandpass modulation techniques can be broadly classified into two categories. The first is single-carrier modulation, where data is transmitted by using one radio frequency carrier. The other is multicarrier modulation, where data is transmitted by simultaneously modulating multiple radio frequency carriers. By using multicarrier modulation, we divide the stream of data into several bit streams, each of them having a lower bit rate.

During the early stages of multicarrier modulation, the signal bandwidth was divided into non-overlapping frequency subchannels, with each transmitting a different data stream from a common source. The fact that there was no overlap between the data streams helped eliminating interference among different data streams (interchannel interference). On the other hand, it resulted in a very inefficient use of the available spectrum. These systems involved a high hardware complexity since parallel data transmission was implemented using a large number of oscillators, each tuned to a specific subcarrier.

Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation technique where data symbols are modulated on a number of regularly spaced subcarriers. These subcarriers have the minimum frequency separation required to maintain orthogonality (of their corresponding time domain waveforms) yet the signal spectra of the different subcarriers overlap in the frequency domain. The spectral overlap results in a waveform that utilizes bandwidth very efficiently. The most important feature of OFDM is that, by choosing the subcarrier spacing properly in relation to the channel coherence bandwidth, we can convert a frequency selective channel into a number of parallel frequency flat subchannels. Techniques that are appropriate for flat fading channels can then be applied [8].

OFDM has a number of advantages compared to single carrier transmission. Firstly, OFDM has the ability to cope with severe channel conditions such as narrowband interference and frequency-selective fading due to multipath. In addition, wideband channel equalization is attained through many narrow band channel equalizations. Also, Inter Symbol Interference (ISI) is easily eliminated by adding a comparatively short Cyclic Prefix at the start of each packet. OFDM is a signaling technique that is widely adopted in many recently standardized broadband communication systems, mainly due to its ability to cope with frequency selective fading which is common in high data rate communications.

1.2 Software Defined Radio (SDR)

SDR is a rapidly evolving technology that receives enormous atttention and generates interest in the telecommunication industry. SDR is a revolution in radio design due to its ability to create radios that can change on the fly, creating new choices for users.

An SDR system is a radio communication system where standard components that are implemented in hardware (e.g. filters, amplifiers, mixers modulators/demodulators, detectors etc.) are instead implemented using software in a personal computer or an embedded system. Instead of changing the circuitry to handle different types of radio signals, we can just change the software. This makes SDR very flexible because we can just replace the software and operate with completely different protocols.

A basic SDR system may consist of a personal computer and the necessary special-purpose hardware to transmit and receive electromagnetic waves (a device such as a USRP). A significant amount of signal processing is handed over to the general-purpose processor of the personal computer rather than being done in the special-purpose hardware.

Chapter 2

OFDM Structure

2.1 OFDM Transmitter

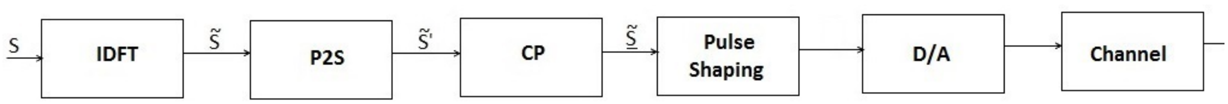

Figure 2.1: OFDM Transmitter

As we mentioned earlier, the difference between OFDM and other FDM systems is orthogonality. Orthogonality indicates that there is a precise mathematical relationship between the frequencies of the carriers of the system. While in normal FDM systems carriers are spaced apart and guard bands are introduced between adjacent carriers, in OFDM, carriers are arranged in such a way that they are overlapping, but because of the orthogonality, the received signal is not affected by Inter Carrier Interference.

In Figure 2.1, we depict the operation of an OFDM transmitter. In the first place, a training sequence is added in the beginning of each OFDM symbol (trainings' format will be addressed in detail in Time synchronization chapter) and then data and trainings are passed through an Inverse Discrete Fourier Transform. Then a Cyclic Prefix is added at the beginning of the symbol as a guard interval to cope with Inter Symbol Interference. Once Cyclic Prefix is added, a Pulse Shaping Filter (in our case, Square Root Raised Cosine) is applied to the transmitted sequence. Afterwards, data is converted to Analog with the use of the FPGA Digital to Analog converter and then transmitted through the channel.

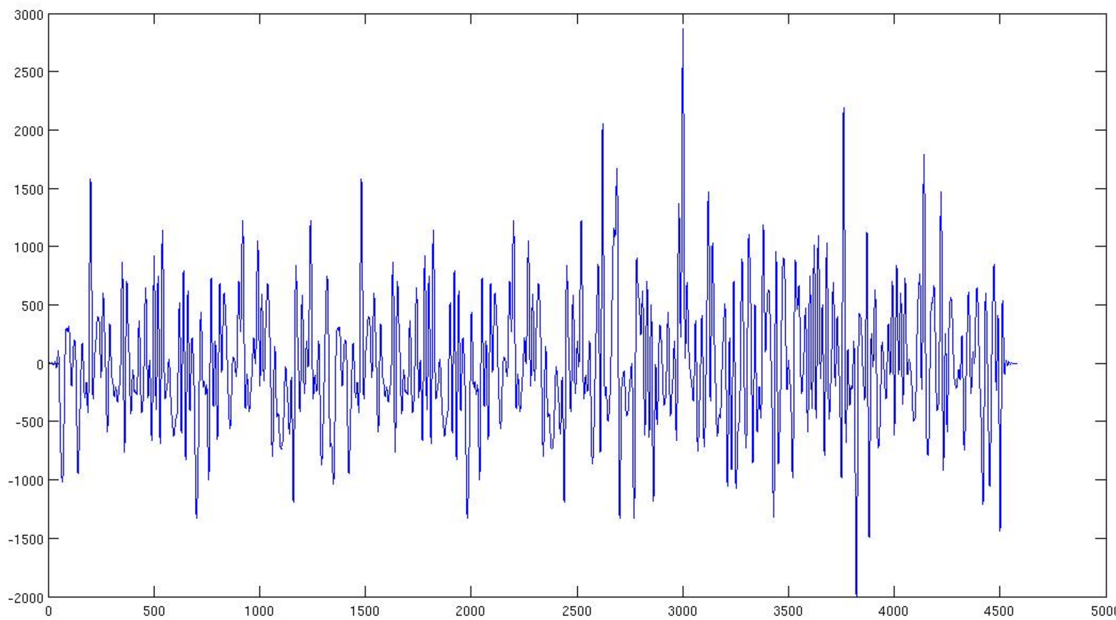

Figure 2.2: An OFDM frame in the transmitter

2.2 OFDM Receiver

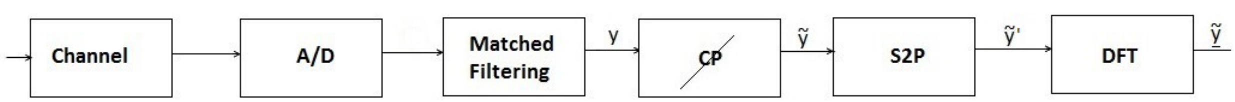

Figure 2.3: OFDM Receiver

Receiver's operations are roughly the opposite of the transmitter. Once the signal is received, the FPGA Analog to Digital converter converts data into a digital signal. A Matched Filter is applied to the signal (usually a Square Root Raised Cosine filter). Then the Cyclic Prefix is removed and the rest of the received signal is passed through a Discrete Fourier Transform. Time synchronization, Carrier Frequency Offset and channel estimation will be adressed later in this chapter.

In order to avoid a large number of modulators and filters at the transmitter and complementary filters and demodulators at the receiver, it is desirable to use digital signal processing techniques such as Discrete Fourier Transform (DFT). The sinusoids of the DFT form an orthogonal basis set which is necessary to eliminate Inter Symbol Interference. The OFDM transmitter is implemented using the IDFT and the receiver using DFT. Both IDFT and DFT would usually require N^2 multiplications and additions. However, by calculating the output using the Fast Fourier Transform (FFT), we reduce the number of computations to the order of $N \log N$.

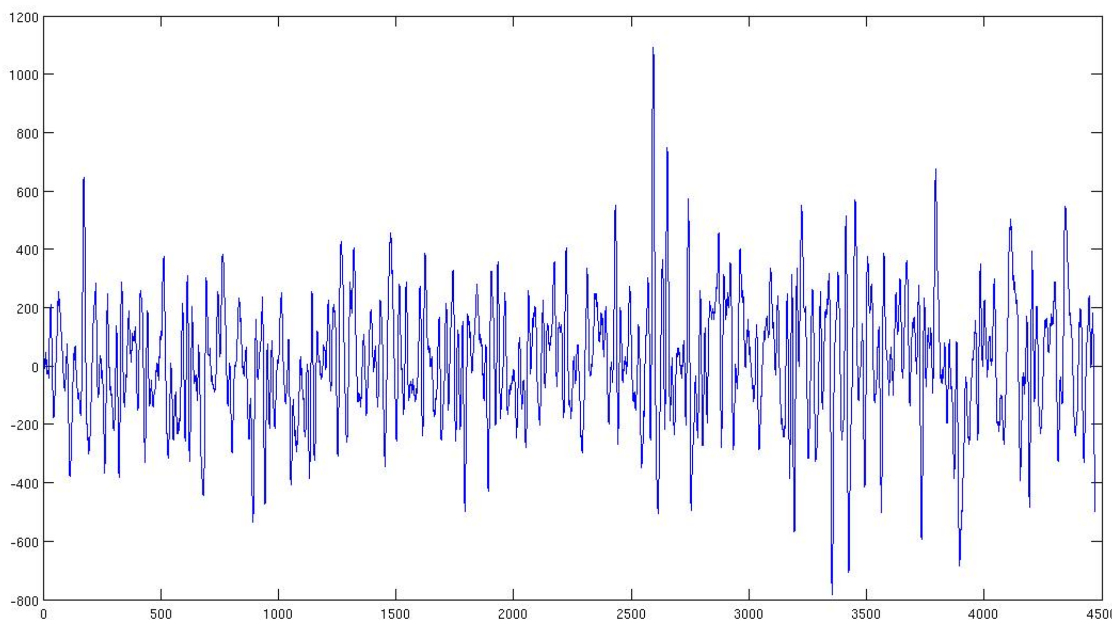

Figure 2.4: An OFDM frame in the receiver

2.3 Time synchronization

A number of time synchronization algorithms for the OFDM systems have been proposed. Most of these algorithms are based on the correlation of identical parts of the OFDM symbols. One way of identifying the start of the packet is to use the correlation between the cyclic prefix in the beginning and in the end of the packet. Another way is to create two identical sequences of training symbols in the start of each OFDM frame.

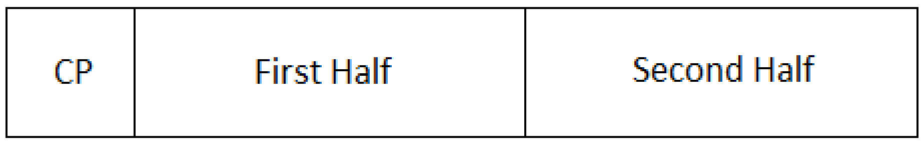

Figure 2.5: Structure of an OFDM synchronization symbol

The transmitter inserts a synchronization symbol in the beginning of each frame of OFDM data symbols. The synchronization symbol consists of two identical parts. In the frequency domain, this means that symbols are only transmitted in every even frequency (tone). Before each synchronization symbol we insert a (CP) Cyclic Prefix.
The length of the cyclic prefix must be as big as that of the channel impulse response in order to keep the two halves of the training symbols almost[1] identical at the output of the channel.
The synchronization symbol is placed in the start of each frame. The time synchronization process is carried out in the time domain, searching for identical symbols in the first and the second half of the packet. In Figure 2.6, we can observe the two identical halves of the Synchronization symbol in the time domain.

[1]The two halves are not exactly identical because of the phase difference caused by carrier frequency offset.

Let S be the training sequence used as an input to the IDFT module. Then, the output of the IDFT in the transmitter is denoted by $\tilde{S} = [\tilde{S}_0, \tilde{S}_1, ..., \tilde{S}_{N-1}]^T$, where

$$\tilde{S}_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k e^{j\frac{2\pi kn}{N}}. \tag{2.3.1}$$

The training symbol sequence to be transmitted is expressed as

$$\underline{\tilde{S}} = \Big[\underbrace{S_{\frac{N}{2}-L+1}, S_{\frac{N}{2}-L+2}, ..., S_{\frac{N}{2}-1}}_{CP}, \underbrace{S_0, S_1, ..., S_{\frac{N}{2}-1}}_{FirstHalf}, \underbrace{S_0, S_1, ..., S_{\frac{N}{2}-1}}_{SecondHalf} \Big]^T. \tag{2.3.2}$$

The transmitted signal after pulse shaping will be

$$X(t) = \sum_n \underline{\tilde{S}}_n g_T(t - nT). \tag{2.3.3}$$

At the channel output the received signal will be

$$Y(t) = c(t) * X(t) + W(t) = \sum_n \underline{\tilde{S}}_n h(t - nT) + W(t) \tag{2.3.4}$$

where $h(t) = c(t) * g_T(t)$.

With Carrier Frequency Offset and phase offset, we have

$$Y(t) = e^{-j(2\pi\Delta Ft+\phi)} \sum_n \underline{\tilde{S}}_n h(t - nT) + W(t). \tag{2.3.5}$$

If we sample with period $T_s = \frac{T}{over}$, we obtain

$$\begin{aligned} Y[k] = Y(kT_s) &= e^{-j(2\pi\Delta FkT_s+\phi)} \sum_n \underline{\tilde{S}}_n h(kT_s - nT) + W(kT_s) \\ &= e^{j\phi} e^{-j2\pi\Delta fk} \sum_n \underline{\tilde{S}}_n h_{k,n} + W[k] \end{aligned} \tag{2.3.6}$$

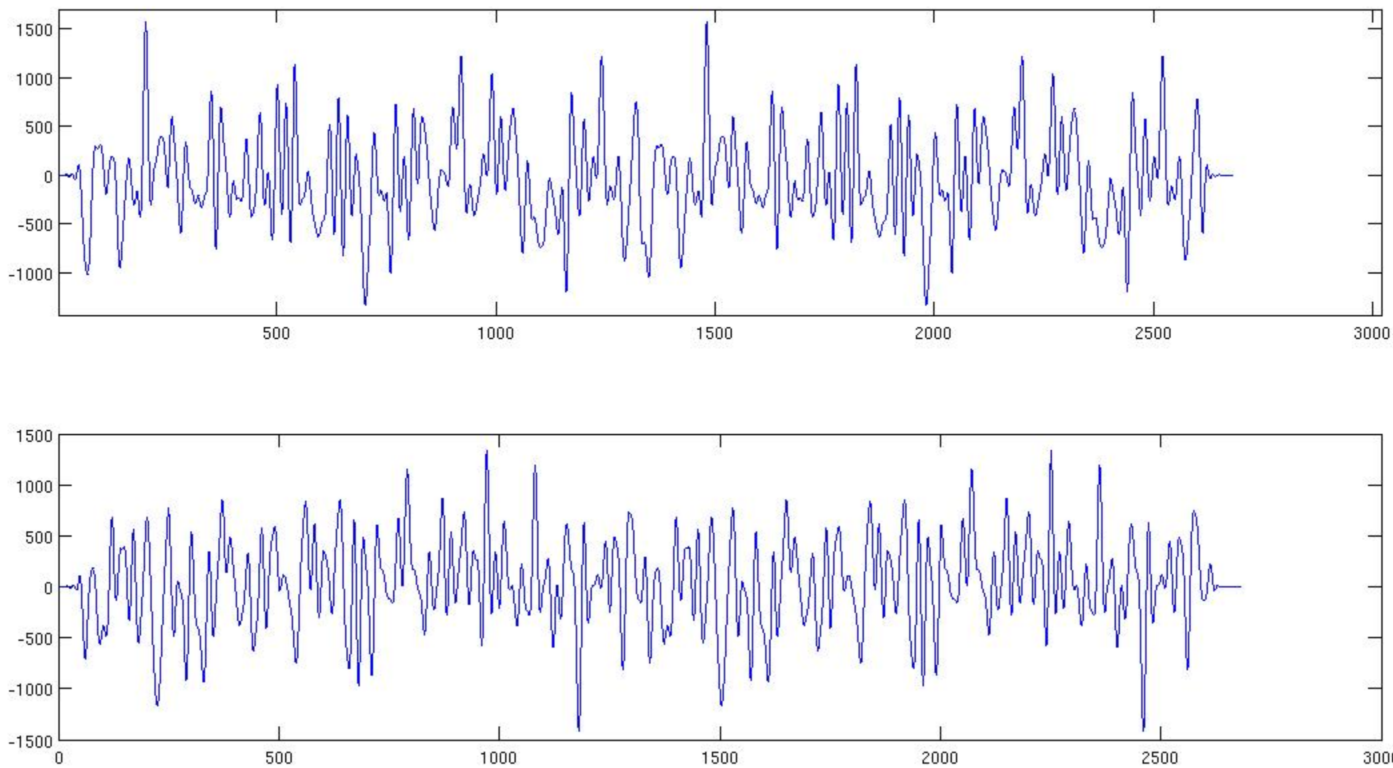

Figure 2.6: Real and imaginary part of Synchronization Symbol.

where $\Delta f := \Delta F T_s$. If we incorporate $e^{j\phi}$ into h we get

$$Y[k] = e^{-j2\pi\Delta f k} r[k] + W[k] \tag{2.3.7}$$

where $r[k] = e^{j\phi} \sum_n \underline{\tilde{S}}_n h_{k,n}$ or, in a slightly different notation,

$$y_n = e^{-j2\pi\Delta f n} r_n + w_n. \tag{2.3.8}$$

At the receiver, we need to find the correct start of the packet. A way to do this, using the training sequence $\tilde{S} = \left[S_0, S_1, ..., S_{\frac{N}{2}-1}\right]^T$, is to compute the statistic [2]

$$P(d) = \sum_{m=0}^{\frac{N}{2}-1} \left(y_{d+m}\tilde{S}_m\right)^* \left(y_{d+m+\frac{N}{2}}\tilde{S}_m\right) \tag{2.3.9}$$

where d is a time index corresponding to the first sample in a window of N samples and m is a time index that runs for each d. By incrementing d, we slide the window in order to find the start of the packet. We define the optimal d as

$$d_* = \text{argmax}|P(d)|. \tag{2.3.10}$$

For an analysis of this estimator see [6]

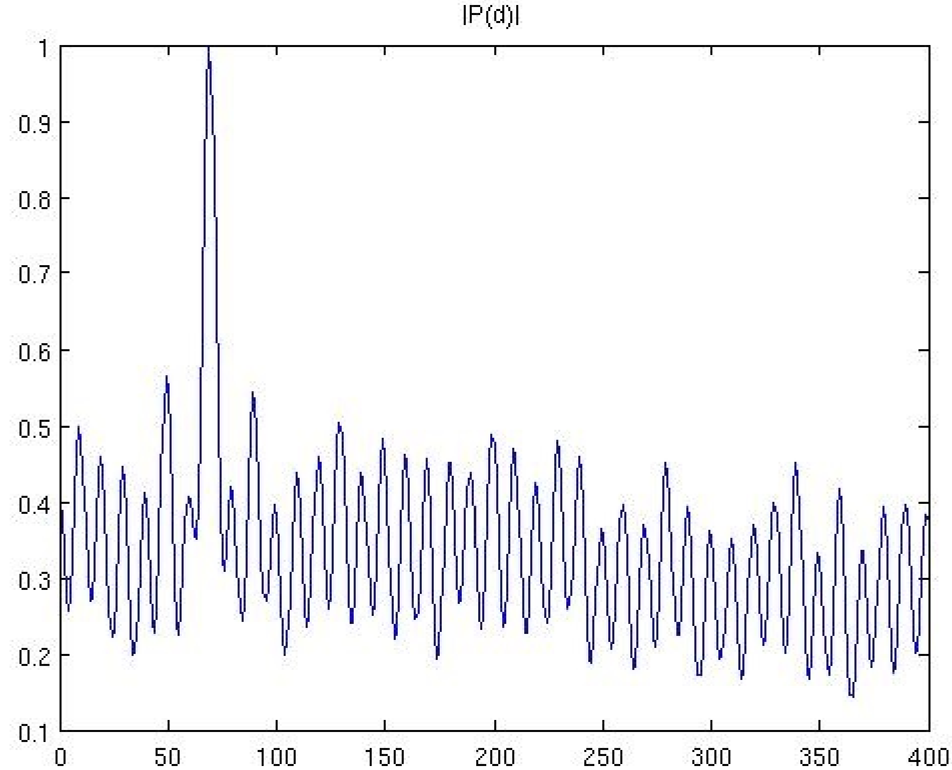

Figure 2.7: Pilot based method for high SNR (10 dB).

In Figure 2.7 we plot a realization of $|P(d)|$. As we can see, the main peak is approximately twice the amplitude of the other peaks of $|P(d)|$.

This method for time synchronization in OFDM systems is robust and is effective in low SNR as well. In Figure 2.8, we see that even when the SNR is low the peak of the correct timing synchronization is still observable and once again approximately twice the amplitude of the other peaks of $|P(d)|$.

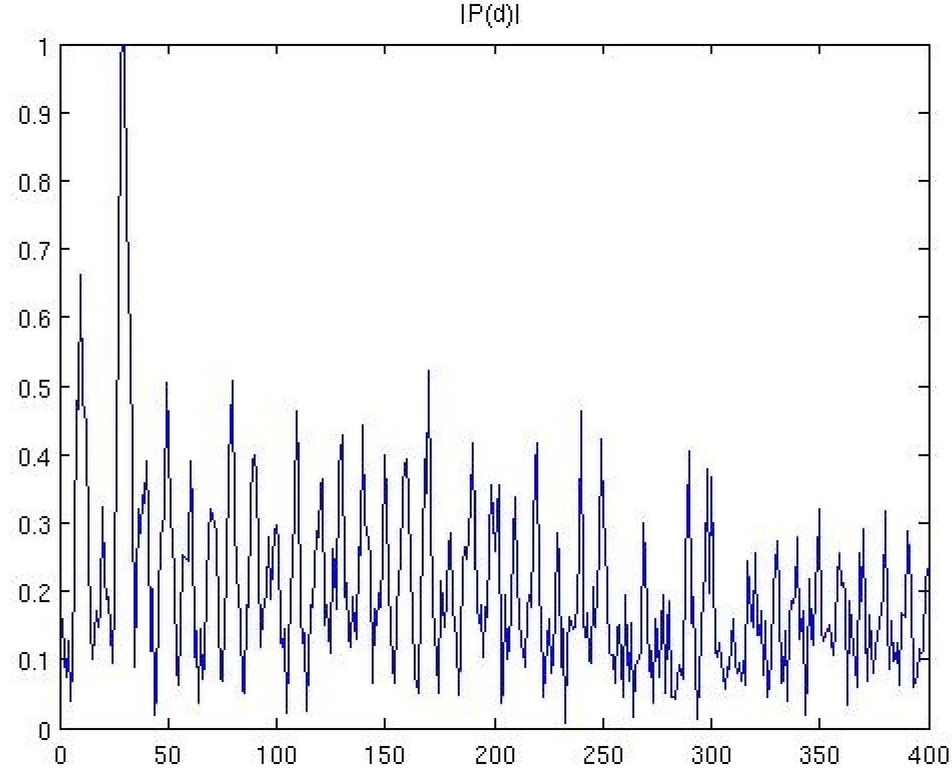

Figure 2.8: Pilot based method for low SNR (3 dB).

2.4 Carrier Frequency Offset

In wireless communications, baseband signals are often modulated with a high frequency carrier in the transmitter, making them more resilient in long distance transmissions, and demodulated with the same frequency in the receiver so as the signal to be proccessed in the baseband.

A difference in the frequency between the oscillators in the receiver and the transmitter and/or the Doppler effect causes a shift in frequency domain. This is called Carrier Frequency Offset (CFO) and needs to be estimated and eliminated because a demodulated signal with even a small shift in the frequency domain can result in a large Bit Error Rate (BER).

First of all, time synchronization needs to be achieved in the receiver in order to identify the beginning of the block. As shown earlier, once we have detected the start of the packet the received signal affected by CFO and channel's impact can be expressed as

$$y_n = e^{-j2\pi\Delta fn} r_n + w_n. \tag{2.4.1}$$

2.4.1 Flat Fading Channel

Assuming a flat channel for the duration of the block, then

$$y_n = e^{-j(2\pi\Delta fn)} e^{-j\phi} h_0 \tilde{S}_n + w_n. \tag{2.4.2}$$

Denoting $h_0' = e^{-j\phi} h_0$, then the received samples can be expressed as

$$y_n = e^{-j(2\pi\Delta fn)} h_0' \tilde{S}_n + w_n. \tag{2.4.3}$$

If we multiply y_n with the complex conjugate of the known training sequence $\tilde{S}_n^*$ we denote z_n

$$\begin{aligned} z_n &= y_n \tilde{S}_n^* = e^{-j(2\pi\Delta f n)} h_0' \tilde{S}_n \tilde{S}_n^* + w_n' \\ &= e^{-j(2\pi\Delta f n)} h_0' |\tilde{S}_n|^2 + w_n' \end{aligned} \tag{2.4.4}$$

z_n is a noisy complex exponential with frequency $f' = -\Delta f$.

We can estimate the frequency f' if we use Fourier Transform on z_n and find the maximum in the digital frequency domain $F_* = \text{argmax}|\mathcal{F}[\{z_n\}]|$.

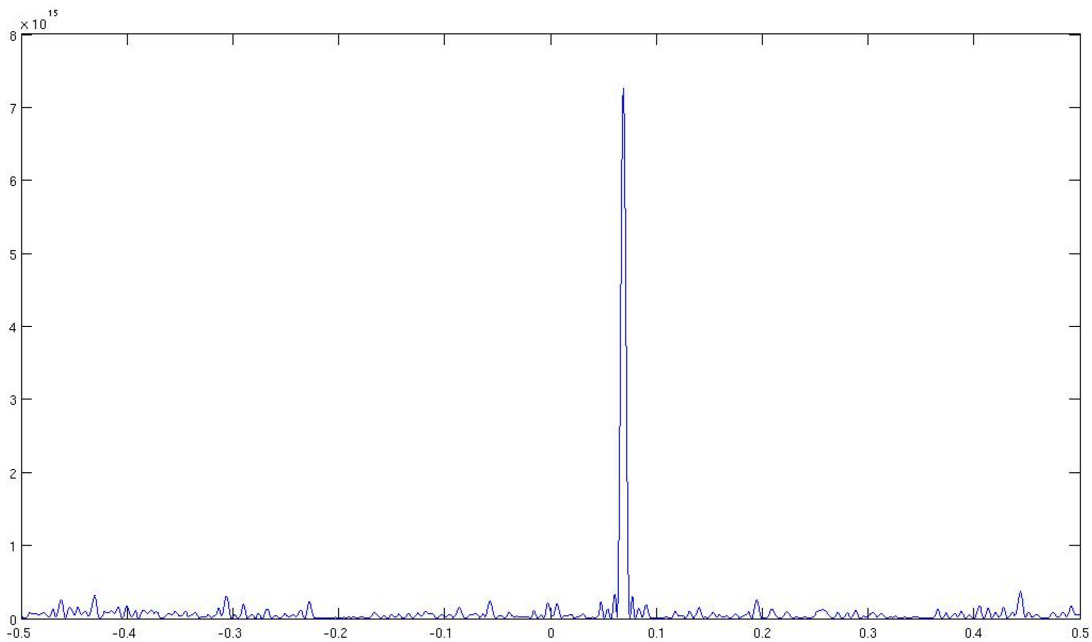

Figure 2.9: A realization of $|\mathcal{F}[\{z_n\}]|$

Once F_* is estimated, it can be used to eliminate the effect of the Carrier Frequency Offset by creating a sequence

$$m_n = e^{-j2\pi F_* n} y_n \tag{2.4.5}$$

m_n is the training sequence without CFO and affected only by channel's impact.

2.4.2 Frequency Selective Channel

In order to estimate the carrier frequency offset, the receiver exploits the special structure of the OFDM synchronization symbol. So we assume that time synchronization has already been achieved. As shown earlier, the received signal can be expressed as

$$y_n = e^{-j2\pi\Delta f n} r_n + w_n, \text{ for } n = 1, 2, ...N \tag{2.4.6}$$

then the received samples in the first half can be expressed as

$$y_n = e^{-j2\pi\Delta f n} r_n + w_n, \text{ for } n = 1, 2, ...\frac{N}{2} \tag{2.4.7}$$

and the second half of the received samples as

$$\begin{aligned} y_{n+\frac{N}{2}} &= e^{-j(2\pi\Delta f(n+\frac{N}{2}))} r_{n+\frac{N}{2}} + w_{n+\frac{N}{2}} \\ &= e^{-j(2\pi\Delta f(n+\frac{N}{2}))} r_n + w_{n+\frac{N}{2}}, \quad \text{for } n = 1, 2, ...\frac{N}{2} \end{aligned} \tag{2.4.8}$$

because of the structure of the Synchronization symbol we know that $r_{n+\frac{N}{2}} = r_n$.

As a consequence,

$$
\begin{aligned}
y_n y^*_{n+\frac{N}{2}} &= \left(e^{-j2\pi\Delta f n} r_n + w_n\right)\left(e^{-j2\pi\Delta f (n+\frac{N}{2})} r_n + w_{n+\frac{N}{2}}\right)^* \\
&= e^{-j2\pi\Delta f n} r_n e^{j2\pi\Delta f (n+\frac{N}{2})} r_n^* + e^{-j2\pi\Delta f n} r_n w^*_{n+\frac{N}{2}} + \\
&+ w_n e^{j2\pi\Delta f (n+\frac{N}{2})} r_n^* + w_n w^*_{n+\frac{N}{2}} \\
&= e^{j\pi\Delta f N} |r_n|^2 + \tilde{w_n}
\end{aligned} \tag{2.4.9}
$$

where

$$
\tilde{w_n} = e^{-j(2\pi\Delta f n)} r_n w^*_{n+\frac{N}{2}} + w_n e^{j(2\pi\Delta f (n+\frac{N}{2})} r_n^* + w_n w^*_{n+\frac{N}{2}}.
$$

Ignoring the noise part, we can estimate the difference in phase Δf between the two halves of the synchronization symbol. Using all the samples of the training sequence, the carrier frequency offset can be estimated by [6]

$$
\arg\left(\sum_{n=0}^{\frac{N}{2}-1} y_n y^*_{n+\frac{N}{2}}\right) = \arg\left(e^{j\pi\Delta f N}\sum_{n=0}^{\frac{N}{2}-1} |r_n|^2\right) = \pi\Delta f N. \tag{2.4.10}
$$

Thus, an estimate of the carrier frequency offset can be derived as [6]

$$
f_* = \frac{\arg\left(\sum_{n=0}^{\frac{N}{2}-1} y_n y^*_{n+\frac{N}{2}}\right)}{\pi N}. \tag{2.4.11}
$$

Once f_* is calculated, it can be used to eliminate the effect of CFO by creating a sequence

$$
m_n = e^{j2\pi f_* n} y_n \tag{2.4.12}
$$

m_n is the training sequence without CFO and affected only by channel's impact.

2.5 Channel Estimation

2.5.1 Flat Fading Channel

Once Carrier Frequency Offset has been cancelled the next step is to estimate the channel using the corrected training sequence

$$m_n = h_0' \tilde{S}_n + w_n.$$

One way to estimate the channel is [3]

$$\underline{h} = \frac{1}{N} \sum_{l=1}^{N-1} \frac{m_n}{\tilde{S}_n}. \tag{2.5.1}$$

The channel estimation used above works only for flat fading channels. Another process is followed in case of a frequency selective channel.

This estimate can be used as the actual value of the channel to recover the transmit signal as

$$\tilde{y_n}' = m_n \frac{\underline{h}^*}{|\underline{h}|^2} \simeq \tilde{S}_n + \tilde{w_n}'. \tag{2.5.2}$$

Chapter 3

Software Tools and USRP

3.1 Universal Software Radio Peripheral (USRP)

The Universal Software Radio Peripheral (USRP) products are computer hosted software radios. They are designed and sold by Ettus Research, LLC and its parent company, National Instruments. The USRP product family is intended to be a comparatively inexpensive hardware platform for software radio, and is commonly used by research labs and universities all around the world. USRPs connect to a host computer through a high-speed USB or Gigabit Ethernet link. Some USRP models integrate the general functionality of a host computer with an embedded processor that allows the USRP Embedded Series to operate in a standalone fashion.

3.2 USRP 1

The USRP1 is the original Universal Software Radio Peripheral hardware (USRP) that provides entry-level RF processing capability. It is intended to provide SDR development capability for cost sensitive users and applications. The architecture includes an Altera Cyclone FPGA, on 64 MS/s dual ADC on 128 MS/s dual DAC and USB 2.0 connectivity to provide data to host computers. A modular design allows the USRP1 to operate from DC to 6 GHz. The USRP1 platform can support two complete RF daughterboards. This feature makes the USRP ideal for applications requiring high isolation between transmit and receive chains, or dual-band transmit/receive operation. The USRP1 can stream up to 8 MS/s to and from host applications, and users can implement custom functions in the FPGA fabric. [4]

RFX2400 daughterboard

The RFX2400 daughterboard is a full duplex trasceiver designed specifically for operation in 2.4 GHz band. The daughterboard has a Transmitter/Receiver antenna plug and a receiver antenna plug. The transmitter inputs the baseband signal which comes from the mainboard and it is modulated in the central frequency chosen by the software (gnu radio). Pulse shaping is implemented through software as well. The output of the transmitter goes to a two-sided switch which is connected to the input of the transmitter from the one side and to the antenna plug from the other side. This switch is controlled through software and it depends whether data is trasmitted or received. The receiver consists of an oscillator whose frequency is controlled by software, and a mixer that mixes the signal coming from the antenna with a sinusodial signal coming from the oscillator. This is how the receiver demodulates the signal and it is sent to the mainboard for sampling.

We must point out that RFX2400 daughterboards do not have an embedded clock and FPGA's clock is used instead. So their use is required when we need perfect synchronization (e.g. when transmitting from both antennas simultaneously).

Mainboard

The USRP has a 4 high-speed analog to digital converters (ADCs) each with 12 bits per sample with 64 MS/s and 4 high-speed digital to analog converters (DACs) each with 14 bits per sample and 128 MS/s. These 4 buses are connected to an Altera cyclone FPGA and in turn the FPGA is connected through a USB2 port to the computer. So we have 4 input channels and 4 output channels available for use.

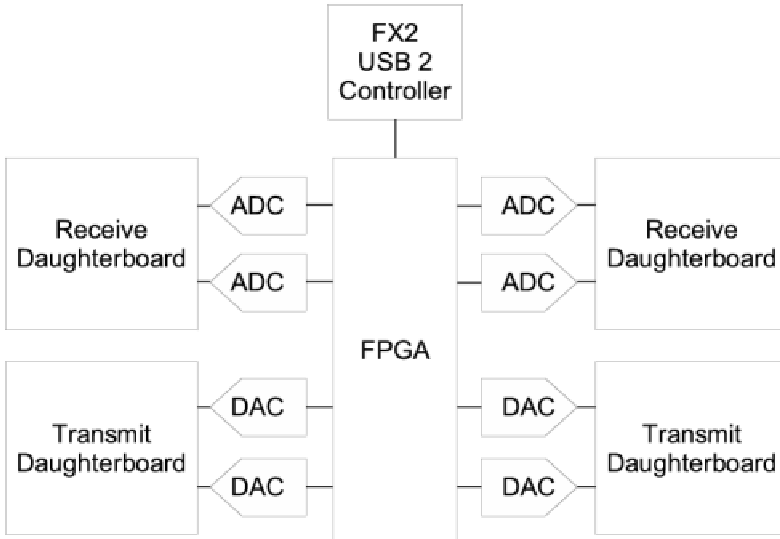

Figure 3.1: The USRP block diagram

FPGA

According to the above, the maximum rate that a USRP sends and receives at the same time is
$(64MS/s * 12bit/Sample + 128MS/s * 14bit/Sample) * 2 = 640Mbyte/s$

But the data rate that USB2 supports is 32Mbyte/s. On the FPGA there are two digital filters for decimation and interpolation. The decimation filter's input is the flow of the samples from the ADC which comes with a rate of 64MS/s. The decimation filter subsamples its input flow by a decimation rate factor, which is chosen through the software. Accordingly, the interpolation filter oversamples the data flow by an interpolation rate factor, which is also chosen through the software, so that the final sampling rate is 128MS/s (sampling rate of the DAC). Finally, as the data rate of the USB is not stable, we need to use a buffer before the interpolation filter in which the samples coming from the PC are stored and then they are read at a rate of (128MS/s)/interpolation so that in the end we have a 128MS/s. [4]

3.3 GNU Radio

USRP is specifically designed to be used with GNU radio. GNU radio is a free and open source signal processing package for building SDRs. It is widely used in academic and commercial environments to support wireless communication research. All the signal processing blocks are written in C++ and Python is used to create the network or graph and glue these blocks together. Every block has a predefined number of input/output interfaces and performs one or more communication functions in the software domain. Each block can be edited, upgraded or even implemented independently, without interfering with the whole communication chain.

3.3.1 GNU Radio Companion

The GNU Radio Companion (GRC) is a graphical programming interface that allows to generate GNU Radio flow graphs without writing the Python code. All the GNU Radio blocks can be accessed in the GRC. This graphical interface has been implemented in XML language. When a new block is generated and if this is required in the GRC, then definition of the block has to be implemented in the XML language. The definition includes: which Python function or class should be called; what are the arguments for the function or class; what is the type of the input/output data streams. The GRC facilitates the user to make the flow graphs by dragging and dropping the blocks from the library. When the GRC runs a model, it generates a Python file in the same directory. This Python file contains the flow graph which the Python interpreter runs. [7]

Chapter 4

Experimental part

4.1 Single Antenna Point to Point System with channel feedback

This is the most basic system we implemented. We have one transmitter and one receiver, both with one antenna. The basic aspects of our protocol are the following:

- We have 2 distinct USRPs so we denote them as Node A and Node B. Each of these nodes is a transceiver, that is, it alternates between a Transmitter (Tx) and a Receiver (Rx) mode.
- When one of these nodes is in Tx mode, it sends N($\simeq$ 50) packets and then turns to Rx mode.
- When a node is in Rx mode, it waits until it detects a packet. Once a packet is detected, after processing, the training sequence is checked and if the Bit Error Rate is zero we assume that a correct packet has been detected.
- After the correct packet has been detected, if it is node B, the flat fading channel h_{est} is estimated and coded into data, if it is node A, h_{est} the flat fading channel is decoded from the received packet and it turns to Rx mode.

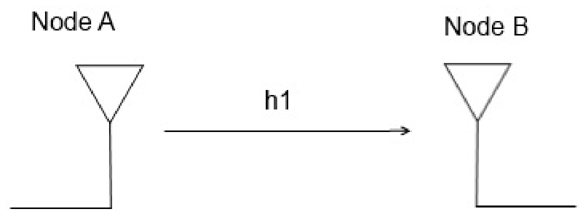

Figure 4.1: A point to point system

We assume that CFO correction and time synchronization was carried out at the receiver so using 2.5.1, we get a complex number $h = h_r + jh_i$ for the actual value of the channel. Figure 4.2 shows the steps after channel estimation.

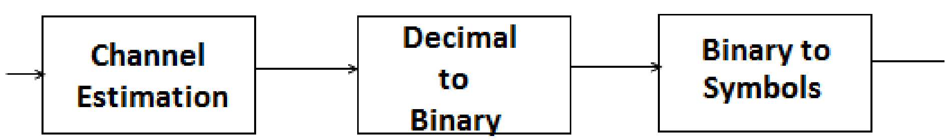

Figure 4.2: Process carried out in Node B

The real and the imaginary part of the channel, h_r and h_i, are converted into binary numbers. Then each of the binary numbers is converted to a 4-QAM sequence using Gray Code. Then both symbol sequences are concatenated and used as a data packet ready to be transmitted.

Now both nodes' function has switched from Transmitter to Receiver and vice versa. Node B now sends the same number of packets as Node A now adding the channel feedback in the end of the packet.

The process that Node A carries out, once the channel is estimated and eliminated, is the inverse procedure as shown in Figure 4.3

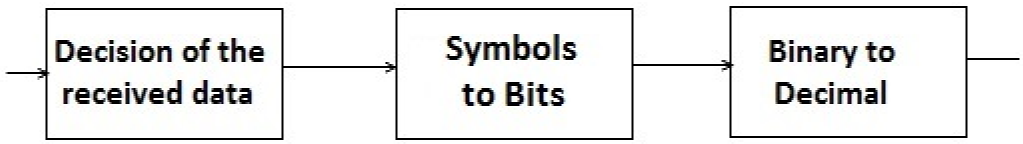

Figure 4.3: Process carried out in Node A

In order to see how the channels look like in a static environment, we estimated the channels for 100 consecutive packets. In Figure 4.4, we plot the channel estimates. We observe that the channel estimates are uniformly distributed on a circle of the complex plane. This happens because their magnitude does not change due to the static environment. However, each channel has a random phase, which seems to be uniformly distributed in $[0, 2\pi)$.

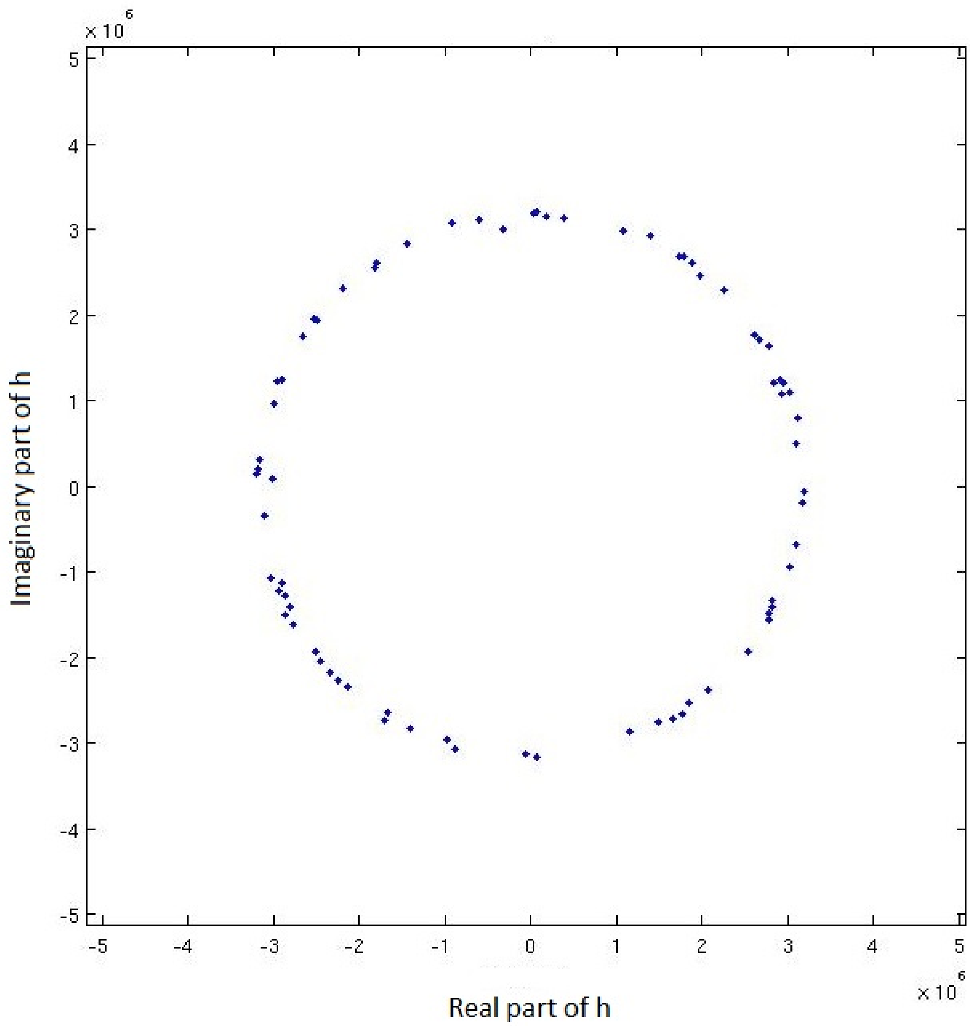

Figure 4.4: Channel Estimates

4.2 Single to Dual Antenna Point to Point System with channel feedback

This system is quite similar to the previous one with one difference. Our Receiver, Node B, has two antennas instead of one. Node A sends again a number of packets and both antennas in Node B are receiving until they detect a packet. Once packets are sent, Node A turns to a receiver.

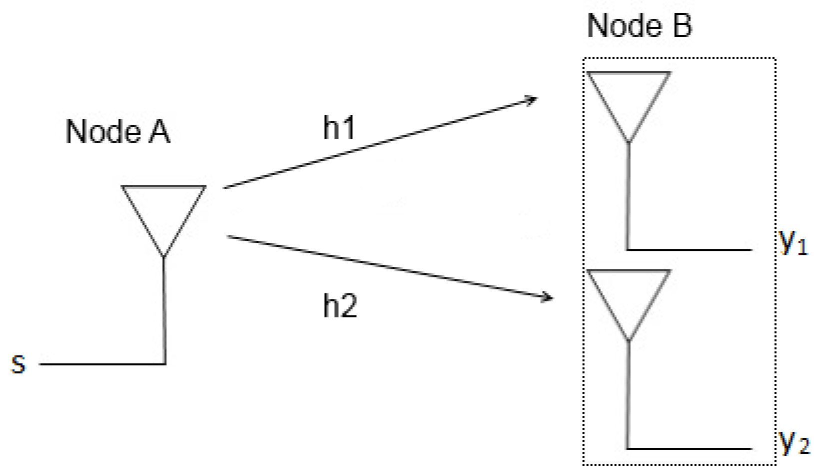

Figure 4.5: Single to Dual Antenna point to point

The received signal after CFO-cancellation and assuming perfect time synchronization for each node is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} s + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{4.2.1}$$

or, in vector notation:

$$\mathbf{y} = \mathbf{h}s + \mathbf{w}.$$

Both antennas of Node B are looking for a packet. Once one of them detects a packet both inputs are being processed. With a similar way as shown previously, Node B estimates both channels (h_1 and h_2) and are coded as data in order to be sent back to Node A.

Channel feedback can be sent back to Node A from either or both of the antennas of node B.

Once Node A has the channel estimates (h_1 and h_2) we calculate the ratio of the magnitude of these channels

$$c = \frac{\max(|h_1|, |h_2|)}{\min(|h_1|, |h_2|)}. \tag{4.2.2}$$

This ratio shows us how stronger is one channel compared to the other. This ratio can be quite big, if for instance there is no line of sight between Node A and one of the antennas on the receiver.

We suppose that we need to transmit different data to each antenna in Node B. So all future packets will have the data sequence split into two halves, each half for a different antenna. So we will use this ratio c to compensate for channel's attenuation in the weak channel antenna.

We assume $|h_1| > |h_2|$ so we need to amplify the symbols sent to the bottom antenna as shown in figure 4.5. So the data packet will have the following form

$x_1, x_2, x_3, \ldots, x_n$	$c \cdot y_1, c \cdot y_2, c \cdot y_3, \ldots, c \cdot y_n$

Figure 4.6: Data symbols for both antennas

This way, we manage to make both antennas receive data symbols with the same energy regardless of the magnitude of each channel.

4.3 Single Point to multi Point System with channel feedback

This system assumes one antenna transmitter that needs to receive the channel estimates from a number of independent nodes.

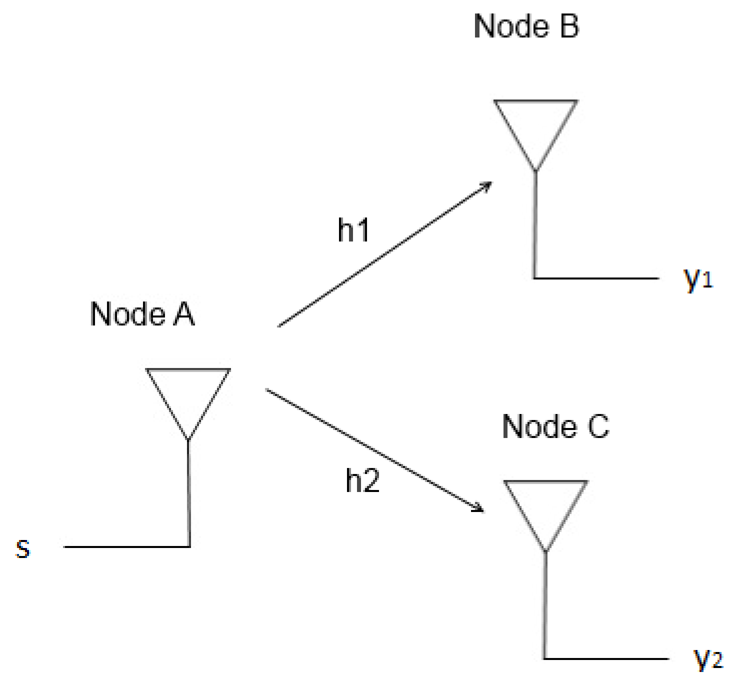

Figure 4.7: Point to Multi point system

This system can be generalized for a bigger number of nodes. The difference of this experiment is that the receivers have to send back their feedback seperately. So now for this system we have to add a header. When Node A receives a packet this header will work as an identification of the Node that sent its channel feedback. Thus Node A will have to receive a number of packets (depending on the nodes) in order to receive the channel estimates of each node.

Once all channel estimates have been received our system follows the same procedure as in the previous system amplifying appropriately the data packets that correspond to nodes with smaller channel absolute value.

4.4 Interference cancellation

In this experiment we study the interference introduced from another transmitter through the same frequency band in a simple Point to Point system. Let Tx_1 with Rx_1 be a communication pair which needs to communicate without causing interference to Rx_2 node.

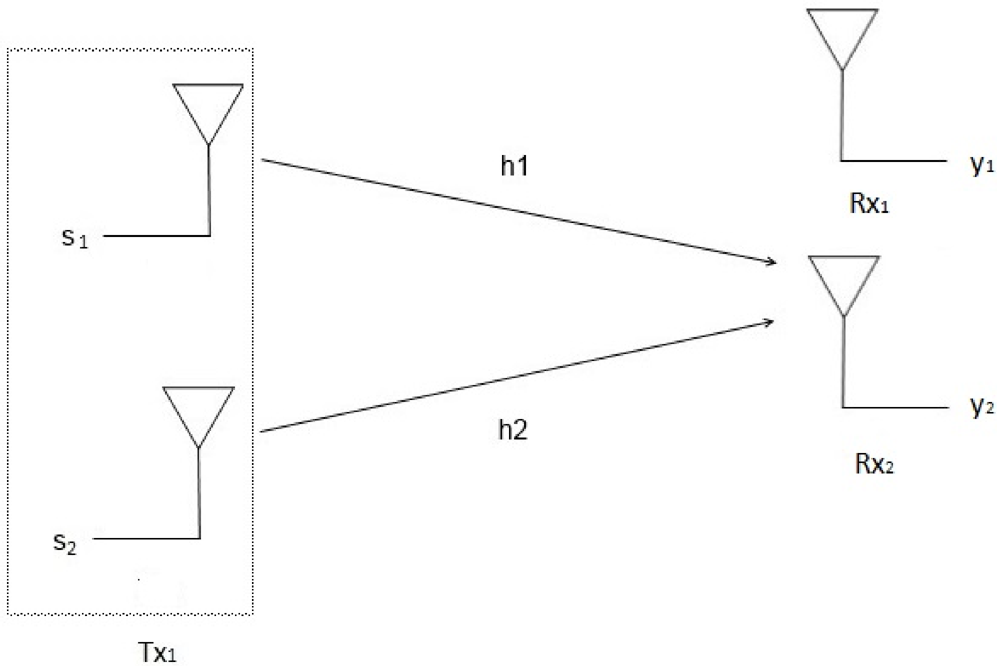

Figure 4.8: Channel estimation of the receiver that will not receive any future packets

As shown in Figure 4.8 the received signal y_2 for Rx_2 will be

$$\boldsymbol{y}_2 = \begin{bmatrix} y_{2,1} \\ y_{2,2} \\ \vdots \\ y_{2,n} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{21} \\ s_{12} & s_{22} \\ \vdots & \vdots \\ s_{1n} & s_{2n} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} \tag{4.4.1}$$

or, in vector notation

$$\boldsymbol{y}_2 = \boldsymbol{S}\boldsymbol{h} + \boldsymbol{w} \tag{4.4.2}$$

where $\boldsymbol{w}$ is Additive White Gaussian Noise

Using the above equation we need to estimate both channels and then send their estimates back to $\mathrm{T_{x1}}$ as feedback. The Least Squares channel estimates can be found by minimizing the following square error quantity

$$\boldsymbol{h}_{LS} = \underset{\boldsymbol{h}}{\operatorname{argmin}} ||\boldsymbol{y}_2 - \boldsymbol{S}\boldsymbol{h}||_2^2. \tag{4.4.3}$$

The solution is given by [1]

$$\boldsymbol{h}_{LS} = \left(\boldsymbol{S}^H \boldsymbol{S}\right)^{-1} \boldsymbol{S}^H \boldsymbol{y}_2. \tag{4.4.4}$$

4.4.1 Orthogonal Beamforming

Beamforming is a signal processing technique used for directional signal transmission. In our case we use the channel feedback in order to cancel the interference of our transmitter at the specific node. So between Tx_1 and Rx_2 if we estimate the channel, as showed earlier, all future packets will pass through the same channel just with a different random phase. Assuming a static environment, for every future packet we will have

$$\begin{bmatrix} h_{1*} \\ h_{2*} \end{bmatrix} = e^{j\phi} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}, \tag{4.4.5}$$

where h_1 and h_2 are the initial channels as shown in Figure 4.8 and h_{1*}, h_{2*} all future channels. We need to beamform the transmit signal appropriately in order to arrive at Rx_2 very weak. Tx_1 will transmit $w_1 s$ and Tx_2 will transmit $w_2 s$. So we beamform the signal sent from Tx_1 with $w_1 = h_2$ and signal sent from Tx_2 with $w_2 = (-h_1)$.
Thus after beamforming Rx_2 will receive

$$\begin{aligned} y_{2,n} &= (h_2)h_{1*}s_n + (-h_1)h_{2*}s_n + w_{2,n} \\ &= h_2 h_1 e^{j\phi} s_n + (-h_1)h_2 e^{j\phi} s_n + w_{2,n} \\ &= 0 + w_{2,n} \\ &= w_{2,n}, \end{aligned} \tag{4.4.6}$$

where $w_{2,n}$ is Additive White Gaussian Noise

For Rx_1 the received signal will be

$$\begin{aligned} y_{1,n} &= (h_2)h_3 s_n + (-h_1)h_4 s_n + w_{1,n} \\ &= h_2 h_3 s_n - h_1 h_2 s_n + w_{1,n} \\ &= (h_2 h_3 - h_1 h_2) s_n + w_{1,n} \\ &= h' s_n + w_{1,n}, \end{aligned} \tag{4.4.7}$$

where $w_{1,n}$ is Additive White Gaussian Noise

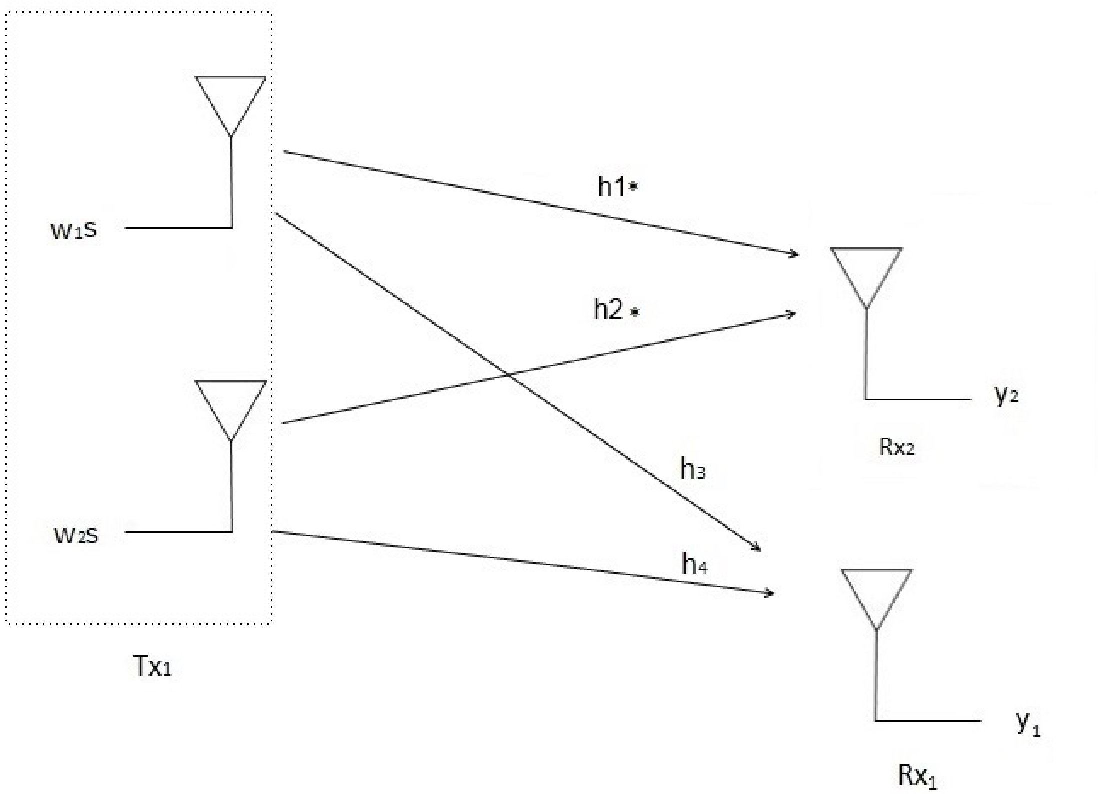

Figure 4.9: 2x2 system, Rx_1 only receiving data signals

In Figures 4.10 and 4.11 we can see two beamformed packets received from Rx_2 and Rx_1 respectively.

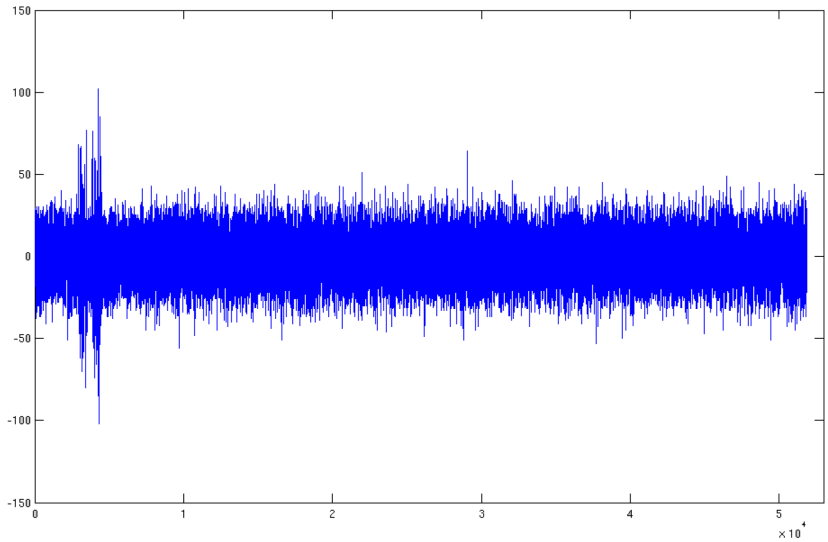

Figure 4.10: Received packet in Rx_2

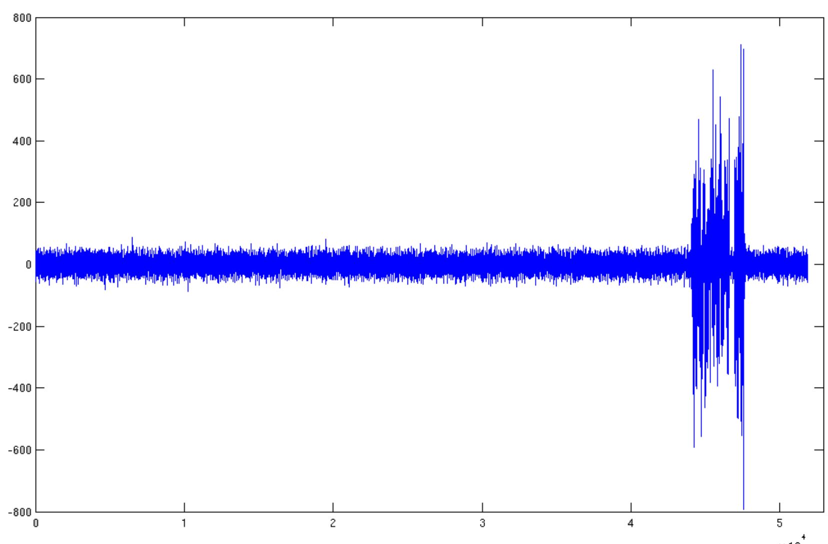

Figure 4.11: Received packet in Rx_1

In this experiment, we observe that indeed the packet received at Rx_2 is utterly attenuated compared with the packet received from Rx_1.

This way we have managed to establish a communication between Tx_1 and Rx_1 without Rx_2 receiving any data. Exploiting the channel's feedback sent, even though both receivers are tuned in the same frequency just one of them receives data. This can be generalized for a greater number of receivers and transmitters that communicate over the same frequency bandwidth. More specifically, we can achieve interference cancellation at $n-1$ receiving antennas with n transmit antennas. [5]

Chapter 5

Conclusion

The importance of multi-carrier communication systems has been established in the present communication era. Multi-carrier communication systems, especially OFDM, have evolved as one of such potential candidates which are bandwidth efficient and robust to multipath channel conditions. In this thesis, a number of experiments were conducted in an SDR testbed using OFDM. In many cases, knowledge of the channel at the transmitter can be useful thus channel feedback was exploited throughout the experiments. A number of transceivers were used in order to make different communication systems using the same hardware and loading different programmes.

Bibliography

[1] S. M. Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory", Prentice-Hall, 1998, Chapter 8.

[2] Ch. N. Kishore and V. U. Reddy,"A Frame Synchronization and Frequency Offset Estimation Algorithm for OFDM System and its Analysis," EURASIP journal on Wireless Communications and Networking.

[3] Athanasios P.Liavas, Lectures of Telecommunication Systems II 2010.

[4] USRP-Ettus Research, a National Instruments Company.

[5] Nikolaos Sapountzis, Bachelor Thesis, "Interference Cancellation in Wireless Communications," 2013.

[6] Ioannis Kardaras, Master thesis "Software-Defined Radio Implementation of an OFDM Link," 2010.

[7] Omair Sarwar, Master Thesis "Software Defined Radio (SDR) for Deep Space Communication," 2013.

[8] Li, Ye, Gordon L. (Eds.), Orthogonal Frequency Division Multiplexing for Wireless Communications, Springer, 2006.