Implementation of simultaneous transmission and reception at different frequencies

A Project Report

submitted by

RESHMA AMRUT PAWAR

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2014

THESIS CERTIFICATE

This is to certify that the thesis titled **Implementation of simultaneous transmission and reception at different frequencies**, submitted by **Reshma Amrut Pawar**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the project work done byher under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Radha Krishna Ganti
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Prof. Harishankar Ramachandran
Head
Dept.of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 1st June 2014

ACKNOWLEDGEMENTS

I wish to express my deep gratitude to my guide Dr.Radha Krishna Ganti for his great concern,continual encouragement,invaluable advice and patient guidance throughout the project.I had like to thank him for everything I ever learnt from him and the flexibility he gave me regarding the project. His enthusiasm,knowledge and loyalty towards students has been truly inspirational for me.

Specially,I like to thank Arjun and Praveen for their infinite advise and support.I would also like to thank other lab mates Vipul,sorav and other researchers in the Networks and Stochastic Systems lab. I would like to thank all my friends Reshma K B, Laxmi Prasanna,Sowmya,Vinuta,Sreelaxmi,Supriya,Anusha and my classmates for their support and making my stay memorable.

I would like to thank the Department of Electrical Engineering for providing an excellent laboratory and computational resources for making this project possible.I would like to thank all the professors who taught me during my days at IITM. Specially,I would like to thank Dr. Krishna Jagannathan for his support and advise during I semester,without his guidance i would have not continued my M.tech.

I am very grateful to my parents Amrut Pawar and Tarabai Pawar for their encouragement and the sacrifices they have been doing throughout their lives. I thank all my family members for their constant love and support specially my younger brother shridhar.

ABSTRACT

KEYWORDS: SDR, USRP, SBX, OFDM, SISO and FDD

Full-duplex (FDD) is implemented using universal software radio peripheral (USRP) devices with SBX daughter boards that support full-duplex operation. The interface isUHDAPI.

We use two universal software radio peripheral (USRP) devices to perform full duplex communication by transmitting and receiving randomly generated data simultaneously. This is implemented in C++

In order to run this program we need two host computers connected to two USRP radio with SBX daughter boards that support full-duplex operation.

The first USRP ttransmits at 1.8GHz and simultaneously receives another signal at 1.895GHz .The second USRP sends at 1.8GHz and receives at 1.8GHz using OFDM_TX _RX.cpp c++ program.

This project is an effort towards developing a SDR system implementing OFDM modulation in c++(full-duplex).

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

FDD	Frequency Division Duplexing
SDR	Software Defined Radio
USRP	Universal Software Radio Peripheral
UHD	USRP Hardware Driver
GRC	GNU Radio Campanion
BPSK	Binary Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
SISO	Single Input Single Output
OFDM	Orthogonal Frequency Division Multiplexing
ADSL	Asymmetric digital subscriber line
VDSL	very-high-bitrate digital subscriber line
LNA	Low Noise Amplifier
VA	Voltage Amplifier
PA	Power Amplifier
LPF	Low pass filter
ADC	Analog to digital converter
DAC	Digital to Analog converter
SPDT	single pole double throw
DPDT	double pole double throw
VCO	Voltage-controlled oscillator
IP3	Third-order intercept point
IIP2	second-order Inpput intercept
IIP3	Third-order Inpput intercept
P1dB	Power at 1dB gain.
IF,RF	Intermediate frequencyt,Radio frequency
BALUN	Balanced Unbalanced transformer

CHAPTER 1

INTRODUCTION

In this project,simultaneous transmission and reception at two different frequencies is implemented with the help of the USRP devices using the UHDAPI in C++.

In this chapter, some major topics are briefly introduced.

1.1 SDR(software defined radio)

The concept of SDR defines a radio device that is capable of flexibly reconfiguring its radio interface by software.

SDR, is a communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, encoder/decoder, modulators/demodulators, etc.) are instead implemented by means of software on a personal computer or embedded devices.

SDR Performs the majority of signal processing in the digital domain using programmable DSPs and hardware support, but some signal processing is still done in the analog domain, such as in the RF and IF circuits.In this project, we are developing SDR wireless communication system using USRP hardware.Fig. 1.1 shows the block diagram of the sdr communication system.

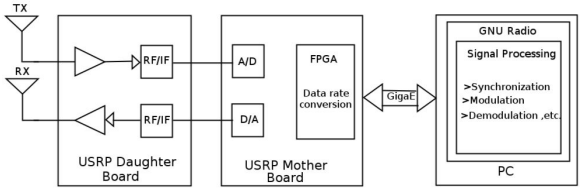

Figure 1.1: Block Diagram of SDR

Free SDR tool kit is available online i.e. GNU. GNU Radio is the development toolkit that handles the signal processing from the SDR hardware. Universal Software Radio Peripheral (USRP) is one of the most popular SDR platforms

developed so far to support the open-source GNU radio software package. Different communication systems have been implemented on the USRP platform .

We use C/C++ as software environment and USRP as hardware platform to transmit data from one place to other over wireless medium.

A software-defined radio receiver uses an analog-to-digital converter to digitize the analog signal in the receiver and convert it to an intermediate frequency (IF). Once digitized, the signals are filtered, demodulated, and separated into individual channels.Similarly, a software-defined radio transmitter performs coding, modulation, etc. in the digital domain. In the final IF stage, a digital-to-analog converter (DAC) is used to convert the signal back to an analog format for transmission.

1.2 USRP(Universal software radio peripheral)

The Universal software radio peripheral products are computer-hosted software radios. USRPs connected to a host computer through a Gigabit Ethernet link or high speed USB.

The USRP product family includes a variety of models that use a similar architecture. A motherboard provides the following systems:clock generation and synchronization,FPGA,ADCs,DACs,host processor interface,and power regulation.These are the basic components that are required for baseband processing of signals.

A front-end called a daughter board is used for analog operations such as up/down-conversion,filtering,and other signal conditioning. This permits the USRP to serve applications that operate between DC and 6GHz.

The USRP hardware driver(UHD) is the device driver which supports Linux,MacOS,and windows platforms.Several software frameworks including GNU radio,Labview,MATLAB and simulink use UHD. Any other language that can import c++ functions can also use UHD. In our experiment we are using networked series USRP called USRPN210 with SBX daughter board and two differnt antennae called VERT900 and VERT2450 dual band.Fig. 1.2 shows the hardware(usrpn210 with sbx daughter board) and antennae used in this project.

Figure 1.2: USRP hardware,SBX daughterboard and two different antennae used in the experiment

1.2.1 SBX Daughter board

The SBX is a wide bandwidth transceiver that provides up to 100 mW of output power, and a typical noise figure of 5 dB. The local oscillators for the receive and transmit chains operate independently, which allows dual-band operation. The SBX is MIMO capable, and provides 40 MHz of bandwidth. The SBX is ideal for applications requiring access to a variety of bands in the 400 MHz-4400 MHz range. Example application areas include WiFi, WiMax, S-band transceivers and 2.4 GHz ISM band transceivers.Fig. 1.3 shows the block diagram of the SBX daughter board.

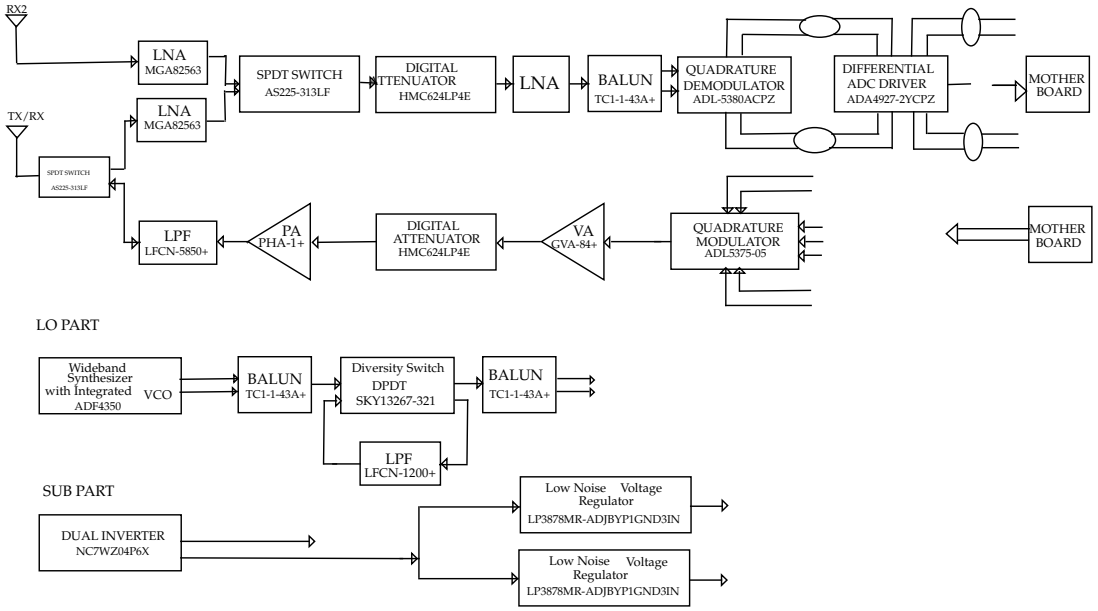

Figure 1.3: Block Diagram of SBX Daughter Board

In this section, we have listed out features/parameters of the each component/chip in present in the SBX board.

Specification	Typical Values
Wide supply operation	+5v to ± 5v
Low input voltage	0.3 mv
-3dB small signal BW	2000MHz
-3dB large signal Bw	1300MHz
2nd harmonic	-86 dBc(@100MHz)
3rd harmonic	-76 dBc(@100MHz)
power supply Operation range Quiescent Current per Amplifier	4.5-11.0V 20-21.6mA

Table 1.1: ADA4927-2YCPZ(Differential ADC Driver)

Specification	Values
low voltage control	0/+3v
Insertion loss	0.6 dB
Isolation	20 dB
Return loss	25 dB
P1 dB	30dBm
2nd harmonic	70 dBc
3rd harmonic	68 dBc
Input third order intercept	53 dBm
Frequency range	0.1-6 GHz

Table 1.2: AS225-313LF(SPDT Switch)

Specification	Typical Values
Frerquency range	DC to 1200 MHz
RF Power input	10w max
DC Current I/p to o/p	0.5A max
Insertion Loss	31.88 dB(@2GHz)

Table 1.3: LFCN-1200+(LPF)

Specification	Typical Values
Frerquency range	DC to 5850 MHz
RF power input	8 w
DC current I/P to o/P	0.5A
Insertion loss	0.28 dB(2GHz)

Table 1.4: LFCN-5850+(LPF)

Specification	Typical Values
Frerquency range	0.05 to 6 GHz
Reverse Isolation	19.9 dB
Input Return loss	11.3 dB
Output Return loss	17.1 dB
High Pout,P1dB	22 dBm
Output IP3	42 dBm
Noise Figure	2.2 dB
Gain	13.5 dB

Table 1.5: PHA-1+(Amplifier))

Specification	Typical
power supply	3v
output power at 1dB gain compression	+17.3 dB
P_sat	+20 dBm
Noise Figure	2.2 dB
Gain	13.2 dB
Output third order intercept	+31 dB

Table 1.6: MGA82563(Low Noise Amplifier)

Specification	Typical Values
Input supply voltage	2.5v to+16v
Output voltage line regulation	0.007% per v
Min. Intput voltage requred to maintain output regulation	2.5v
Output noise voltage(RMS)	18micro v
Quiescent current	<10 μA
Output	1 to 5.5v

Table 1.7: LP3878MR-AD(Low noise voltage regulator)

Specification	Typical Values
power supply	+3v or 5+v
Insertion loss	1.8dB@(Dc-3GHz) 2.8dB@(3-6GHz)
Attenuation range	31.5dB
Return loss	15 dB
I/p power @0.1dB	30dBm
Input IP3	55 dBm
Frequency range	DC-6GHz
Switching speed tRise,tFall tON,tOFF	100ns 150ns

Table 1.8: HMC624LP4E(Digital Attenuator)

Specification	Typical
Frequency range	650-4000MHz
Input return loss	15.36 dB@(2GHz)
Insertion loss	.53 dB@(2GHz)
Reverse Isolation	19.9 dB
Noise Figure	2.2 dB
Output third order intercept	42 dBm
Maximum RF power	24 dBm(250mw)

Table 1.9: TC1-1-43A+(Balun transformer)

Specification	Values
Frerquency range	DC to 7GHz
Saturated O/p power	22.3 dBm
Input Return loss	18.5 dB
Output Return loss	7.7 dB
High Pout, P1dB	20.6 dBm
Output IP3	36.6 dBm
Noise Figure	5.5 dB
Gain	24 dB

Table 1.10: GVA-84+(Amplifier)

Specification	Typical Values
Output carrier frequency	.4 to 6 GHz
Modulator Voltage gain	-3.4 dB
Output Return loss	-19.3 dB
Reverse Isolation	19.9 dB
Sideband suppression	-48.3 dB
second harmonic power	-60.9 dBc
Third harmonic power	-51.3 dB
Baseband I/P BW	95MHz
LO I/P power	13 dBm
RF I/P voltage	500mv p-p
P1dB	9.4 dBm(min.)

Table 1.11: ADL5375-05[Broadband Quadrature Modulator])

Specification	Values
o/p freq. range	.1375 to 4.4GHz
Single supply	3.0 to 3.6 v
vco sensitivity	33MHz/V
Phase dector frequency	32 MHz
2nd harmonic content	-19 dBc
3rd harmonic content	-13 dBc
VCO Phase noise performance	-89 dBc/Hz
Normalised In-Band phase noise floor	-213 dBc/Hz
In-Band phase noise	-97 dBc/Hz
RF output power	-4 to 5dBm
vco tunning voltage	0.5-2.5v
O/p power variation	+1/-1

Table 1.12: ADF4350BCPZ Wideband synthesizer with Integrated VCO

Specification	Typical Values
power supply Voltage Current	4.75-5.25v 245mA(ENBL pin low) 145mA(ENBL pin high)
RF Input Return loss	-10 dB
Input P1dB	11.6 dBm
second-order Inpput intercept(IIP2)	68dBm
Third-order input intercept(IIP3)	29.7dBm
LO and RF Frequency Range	0.4-6GHz
Voltage conversionn gain	450 ohm differential load on I and Q outputs 6.9 dB 200 ohm differential load on I and Q outputs 5.9 dB
Demodulation bandwidth	390 MHz
Quadrature phase error	0.2degrees
I/Q amplitude imbalance	0.07 dB
Output DC Offset(Differential)	+/-10mv
0.1 dB Gain flatness	37MHz
Off Isolation	-70dB
Noise Figure	10.9dB
IQ Magnitude Imbalance	0.07dB
IQ Phase Imbalnce	0.2degrees

Table 1.13: ADL5375-05(Quadrature Demodulator)

Specification	Typical Values
Broadband	LF-6HZ
Very low Insertion loss	0.8 dB
P1 dB	+30 dBm
Low distortion:IP3	44 dBm
Low current consumption	$<15\ \mu$ A
RF input power @0/3v	32dBm
RF input power @0/5v	34dBm
Control voltage rannge	$-0.2 \leq Vc \leq 8V$
Input third order intermodulation intercep	49 dBm
Insertion loss	0.7 dB(ANT1,ANT2 to TX,RX ports)
Isolation	32dB(ANT1,ANT2 to TX,RX ports) 23dB(ANT1 to ANT2,Tx to Rx ports)
Return loss	22dB(ANT1,ANT2 to TX,RX ports)

Table 1.14: SKY13267-321(Diversity Switch)

1.3 Orthoganal Frequency Division Multiplexing(OFDM)

OFDM is a multicarrier, wideband modulation scheme, i.e. the data is modulated not by just a single frequency, rather the data is sent on a number of frequencies and these frequencies bare a special relationship with each other(i.e orthogaonality property). OFDM is therefore thought as serving a dual purpose; it is a multiplexing and modulation technique.

OFDM is a frequency domain multiplexing scheme,where the available frequency bandwidth (frequency selective channel) is divided into large number of orthogonal sub-carriers with a flat fading channel. Each of these sub-carriers can be modulated with data. Here the orthogonal sub-carriers are chosen such that the inter-carrier spacing is less than or equal to the coherence bandwidth. Hence, the channel can be considered as a flat-faded one for each subcarrier.

Also, since the sub-carrier frequencies are multiples of each other, they become orthogonal and the cross talk between adjacent sub-carriers in minimised. Let fs be the sub-carrier spacing. Hence the RF bandwidth of the signal is approximately Nf _s,where N denotes the number of sub-carriers. Hence the sampling time should be smaller than $T\_s = \frac{1}{Nf\_s}$

Hence, $f\_s = \frac{1}{NT\_s}$ is required to maintain orthogonality among the sub-carriers at the receiver.

Since the data is sent in parallel, the amount of data sent per one OFDM symbol is higher which helps to maintain a lower symbol rate. Hence we can afford to use slower algorithms in real-time. A low symbol rate also helps to maintain a bigger guard interval thereby reducing inter-symbol interference. Hence OFDM is a preferred modulation scheme.

1.4 SISO

Traditional wireless communication systems use a single antenna for transmission and a single antenna for reception. Such systems are known as single input single

output (SISO) systems.

1.5 FDD

Frequency division duplex (FDD) is a technique where separate frequency bands are used at the transmitter and receiver side.

Because the FDD technique uses different frequency bands for send and receive operations, the sending and receiving data signals don't interfere with each other. This makes FDD a better choice than Time Division Duplex (TDD) for symmetric traffic such as voice applications in broadband wireless networks.

In this technique,we are using two SISO systems for establishing two-way link.

CHAPTER 2

USRP Interface With PC

Once the basic interfacing was developed, the system was made flexible by allowing some parameters to be set during the run-time. These include the flexible parameters for the baseband processing of the samples also apart from the flexible parameters for the USRP device.

2.0.1 Flexible parameters

Parameters that can be set at the transmitter end are listed in the Table. 2.1 and those available at the receiver end are mentioned in Table. 2.2

Table 2.1: Runtime flexible parameters available at the transmitter end

Parameter	Description
Help	lists the various transmit parameters available.
Args	transmitter USRP device address.
scale	The scaling factor for the transmitted data.
duration	Decides how long you want to transmit and receive concurrently in seconds
timediff	time delay between the frames
tx _cpu	specify the host/cpu sample mode for TX("fc64")
tx _bps	number of bits per symbol (decides the constellation)
tx_freq	Centre frequency for the link in Hz
tx_rate	Rate of the outgoing samples in Hz
tx_gain	Gain for the Tx antennas (only for SBX) in dB
tx_ant	antenna to be used for the transmitter(TX/RX).

Table 2.2: Runtime flexible parameters available at the receiver end

Parameter	Description
Help	help message, lists the various receive parameters available for setting .
Args	receiver USRP device address.
rx _cpu	specify the host/cpu sample mode for RX("fc32")
rx _bps	number of bits per symbol (decides the constellation to be used)
rx_bw	daughter IF filter bandwidth in Hz
rx_freq	Centre frequency for the link in Hz
rx_rate	Rate of the incoming samples in Hz
rx_gain	Gain for the RX antennas (available for RFX and SBX) in dB
rx _ant	antenna to be used for the receiver(RX2 or TX/RX)

It has been found out that the allowed sampling rates should produce an even integer when it divides 100 MSPS. This is so because the sampling rate of the DSP [in FPGA] is 100MSPS and even order decimation filters are present.

2.0.2 Parameter Setting

A class is declared to store variables of arbitrary type and variables from the command line are stored into the class above from which they are assigned to the regular variables. An USRP device handler named usrp is createdwith the address obtained from the boost library, which is used to set other parameters using the pre-defined functions available in UHDAPI. Table 2.3 lists the functions at the transmitter end that are required to set parameters to the USRP device. Similar function calls are available at the receiver end too. UHDAPI(ref) [3](give ref link) provides a detailed description of the same.

Table 2.3: Functions for setting parameters to USRP at the transmitter end

Function	Description
usrp → getpp _string	gives the address of the usrp.
usrp→ *set_tx_rate*	sets the trassmit sample rate
usrp→ *get_tx_rate*	gets the actual value of the transmit sample rate used by the device
usrp→ *set_tx_freq*	this is run in loop for all transmit channels and centre frequency is set
usrp→ *get_tx_sensor*	checks if local oscillator is locked
usrp→ *set_tx_gain*	this is run in loop for all transmit channels and gain is set
usrp→ *set_tx_bandwidth*	set the TX bandwidth on the front end set
usrp → *set_tx_antenna*	select the TX antenna on the front end

2.0.3 send and receive

We create a transmit/receive streamer in the main thread and define a device buffer for storing the data to be sent to/from the USRP.After this,we fill the send packet buffer with the data to be transmitted and send it using send() pre-defined UHD function. Transmission and reception in USRP happens in bursts. So, the UHD function send() splits the data to be sent into chunks and sends them each at a burst. Also, the UHD function recv() collects a defined number of samples (set during run-time through the variable spb) at once and stores them in the recv buffer.

During the transmission and reception of a single frame (which was the first step for communication), the recv buffer should be sized appropriately and the transmitter and receiver have to timed properly such that receiver should be able to capture the packet.

parameter values used for experiment are listed below.Table. 2.1 lists the frequensy,gain and other parameter values usedfor theexperiment.

	Freq(GHz)	Gain	Rate(MSPS)	BW(MHz)	Scale
TX1	1.8	15	1	2	0.2
TX2	1.895	15	1	2	0.2
RX1	1.8	15	1	2	---
RX2	1.895	15	1	2	---

Figure 2.1: Different parameter values

2.1 Frame structure

In this project same framework is used throughout the transmission, which is discussed in this section. The frame structure can be represented broadly as shown in Fig. 2.2

Figure 2.2: Frame structure.

Each frame has a prefix attached to it known as the preamble, followed by a specified number of data blocks.These frame structure received at both the receiver ends is shown in Fig. 2.3. Each part of the frame has a specific purpose and structure. The structure and function of these parts is discussed in this section.

The preamble is used to synchronize timing between the transmitter and receiver. The structure of the preamble depends on the method of synchronization used. The Schmidl and Cox method is used in this project. Only a part of the original preamble is used in the project which has two identical halves in time domain as shown in Fig. 2.4.

This preamble is generated by taking a frequency domain sequence with good correlation properties of half of the length required and inserting zeros between the samples. This gives a sequence of required length with zeros in the even positions. Then the sequence is converted into time domain by taking an IFFT of length equal to the preamble length. Thus we get a sequence which has two identical halves in the time domain. A sequence which gives good performance is selected and used for all the frames. This sequence is also known at the receiver.

The next part of the frame is the data block. All the data blocks follow a common and uniform structure. Each is made of three types of elements. First

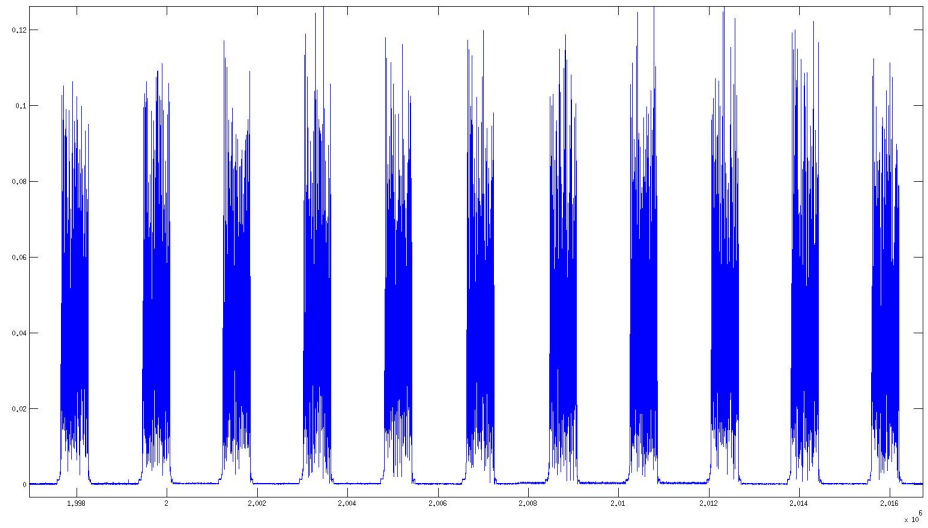

Figure 2.3: Frame structure obtained at the receiver.

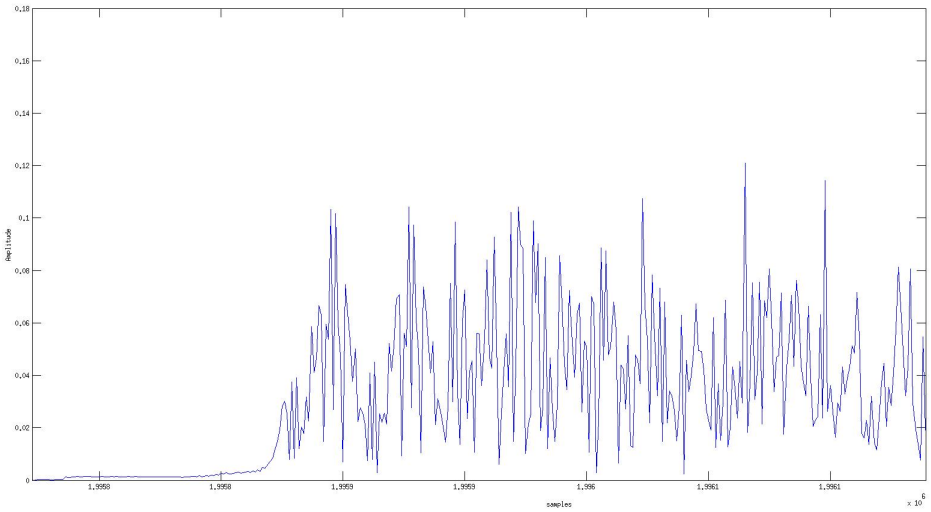

Figure 2.4: Preamble structure obtained at the receiver.

of them is the encoded data corresponding to the message that needs to be transmitted. The second is pilot symbols that are introduced at specified locations in the data symbol. The last is nulled out sub-carriers in the data block.

CHAPTER 3

Frequency Division Duplexing

Frequency Division Duplexing is a technique where the transmitter and receiver operate at different carrier frequencies. For instance, in mobile wireless networks, one block of the electromagnetic spectrum is allocated for uplink, which carries data from mobile phones to a base station. A different block of the spectrum is allocated to downlink, carrying data from a base station to mobile phones.

In the same way,in our experiment there is one link for transmitting data from one usrp(usrp1)to other(say usrp2) and another link for transmitting data from usrp2 to usrp1 simultaneously.

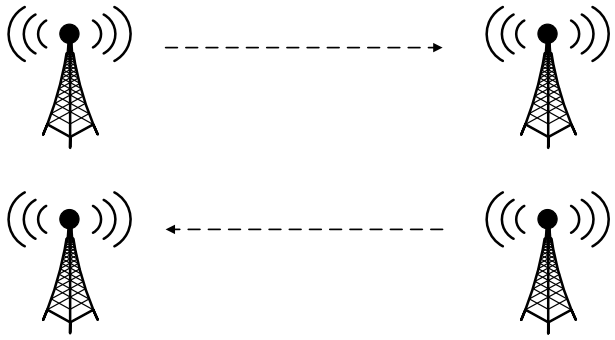

Figure 3.1: Full-Duplex(Frequency Division multiplexing)

A duplex communication system is a point-to-point system composed of two connected parties or devices that can communicate with one another in both directions, simultaneously.Fig. 3.1 shows the duplex communicaton system.

An example of a duplex device is a telephone.Examples of FDD systems include the following: ADSL, VDSL, Cellular systems, including the UMTS/WCDMA Frequency Division Duplexing mode, the CDMA2000 system and IEEE 802.16 WiMax Frequency Division Duplexing mode

3.1 Hardware setting

In this section, how two way link can be established is explained. Two USRP's connected to two seperate PC's(each usrp to one PC) using gigabit Ethernet. Two antenna's are connected to each usrp's. There are different antennae tuned for different frequencies viz. VERT900 Vertical Antenna Dualband and VERT2450 Vertical Antenna Dualband.

Experiment is performed using both the type of antenna.we are operating the USRP N210 in combination with the SBX board in a Full Duplex mode (establishing two way link connection).To do that we are using both the RF connections(TX/RX(used for transmitting) and RX2(used for receiving)).

3.2 Overall Experimental Setup

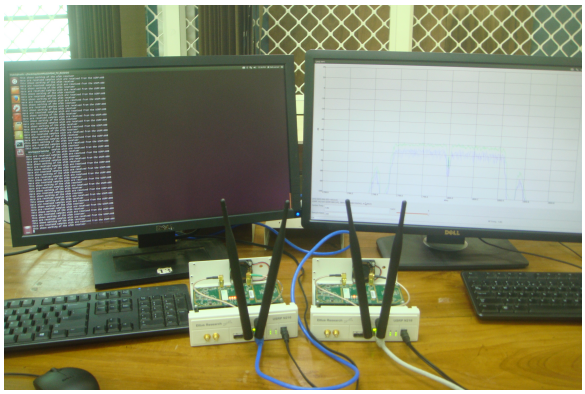

Figure 3.2: Hardware setup.

The first USRP transmits at 1.8GHz and simultaneously receives another signal at 1.895GHz.The second USRP transmits at 1.895GHz and receives at 1.8GHz.

3.3 Block diagram of Experimental setup

Digital data(string of bits) which can be mapped onto constellations during communication. There are two basic types of encoding schemes namely, Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM).

In this project, QAM was used. The choice of encoding scheme is a tradeoff

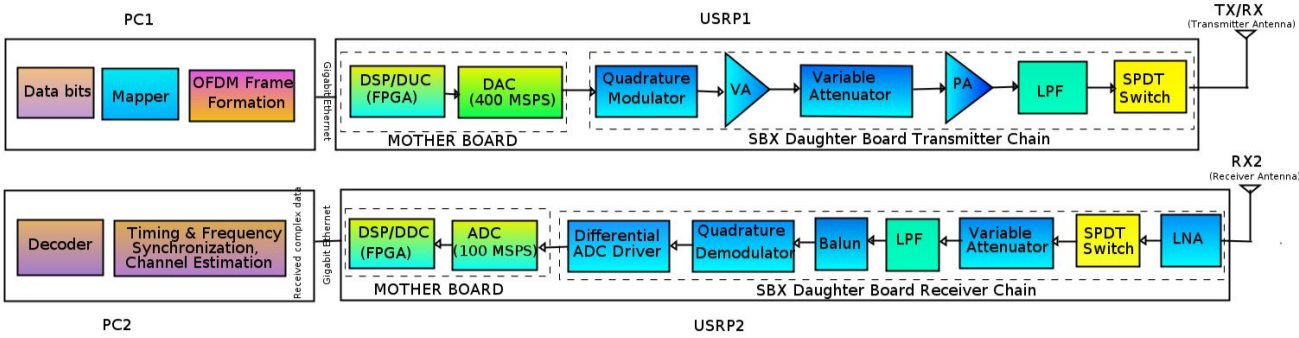

Figure 3.3: Block Diagram.

between the amount of data transmitted per symbol and the symbol error at the receiver. For example BPSK has fewer constellation points than QPSK so the amount of data represented by each constellation point of BPSK is lower than QPSK, but since the number of constellation points are fewer, the chance to mistake a given constellation point for another at the receiver is also lower, i.e. the symbol error rate and hence BER is lower. However, these issues can be overcome by increasing the SNR. So, depending upon the channel conditions and transmit power, decision can bemade during run-time as to which constellation to use. The code provides such flexibility.

The modulation scheme used in this project is orthogonal frequency division multiplexing (OFDM). Implementation of the OFDM modulation is done using IFFT.. All the mapped data points are taken in the frequency domain in the form of a vector and an N-point IFFT is performed to get the time domain signal, where N is the OFDM symbol length.

Once the ofdm frame has been formed, it is send from PC to USRP through Gigabit ethernet.In USRP there are two parts as shown in the above Fig. 3.3 one is mother board where DUC,DDC,DAC and ADC chips resides and other is daughter board where all other chips Ouadrature modulator,voltage amplifier,voltage attenuator,power amplifier,LPF,LNA,SPDT switch,variable attenuator,LPF,BALUN,Quadrature demodulator and differential ADC driver are present.

When we transmit data from one end to other, then the data will go through all these stages(through both the TX and RX chains). One such one-way link (from USRP1 to USRP2) is shown in the Fig. 3.3 .And other link will also be in the same way (from USRP2 to USRP1). In the mother board,USRPN210 has

two high speed ADCs.

In mother board, FPGA gives data to DAC and takes from ADC at sample rate of 100 MSPS. DAC has internal converter of 4x to convert data to 400 MSPS. FPGA digitally interpolates and decimates the hardware sample rate of 100MSPS to sample rate according to user specification (given by set_tx_rate and set_rx _rate functions).This is the only sample rate control available on USRP N210 and is achieved by changing interpolation and decimation factors according to user specification.

Speifications of the antennae used in the experiment are given in Fig. 3.4.

Type	Frequency range
VERT900	824-960,1710-1990
VERT2450	3400-2480,4900-5900

Figure 3.4: Antenna specification

DDC performs two functions:

- Frequency shifting takes place:It down converts the signal from the IF band to the baseband.
- Decimation of rx stream: It decimates the signal so that the data rate can be adapted by the Gigabit ethernet(or USB) and reasonable for the computer's computing capability.

  Decimator can be treated as a LPF followded by a down sampler.

  In the TX path ,we need to send a baseband I/Q complex signal to the USRP board.
- DUC will interploate this complex signal,upconvert it to the IF and finally send it through the DAC.

  The receiver chain consists of a wideband RF front-end(nothing but SBX daughter board) that firstly preamplifies an incoming RF passband signal with a low-noise amplifier (LNA) and then converts it directly down to a complex BB signal. The BB signal is then amplified with a baseband amplifier and ready to enter the digital back-end.

  Note that The Maximum input power to receiver should be less than -20 dBm and shuld not exceed -10 dbm for safe use[Ref:internet]

CHAPTER 4

Implementation Part

We combine two codes OFDM TX.cpp and OFDM RX.cpp such that simultaneous transmission and reception is possible at different frequencies.(Ex1:transmitting freq=2.4GHz and receiving freq=2.45GHz and ex2:TX freq=1.895GHZ and RX frq= 1.8GHz).

We are using boost thread to combine them and such that both threads should run parallely(Ref: benchmark_rate.cpp source code).

In this section,software part required for acheiveing FDD is explained.

Two threads are created using create_thread _group, one is ofdm_rx_thread and other is ofdm_tx thread .First receiver thread will be called and then txansmitter thread.

Threads are discrete processing sequences that allow execution of different functions within a given application at the same time. Threads actually allow the simultaneous execution of two functions without having one waiting for the other.

The name of the function that should execute within the thread is passed to the constructor of boost:: thread_group().

Once the variables ofdm_rx , ofdm _tx in the main program are created, then these two functions starts executing in its own thread immediately. At this point, ofdm_rx ,ofdm_tx executes concurrently with the main() function. These two threads does not share the data.

Sleep() either expects a period of time or a specific point in time indicating how long or until when the current thread should be stalled. By passing an object of type boost::posix_time::seconds, a period of time is specified in this program as boost::posix_time::seconds comes from the Boost.DateTime library that is used by Boost.Thread to manage and process time data.

Calling interrupt() on a thread object interrupts the corresponding thread. This instruction "thread_group.interrrupt_all()" does the same for all the threads used in the main program.

In our program, interrupted means that an exception is thrown inside all the threads.However, this only happens once the thread reaches an interruption point.Simply calling interrupt() does not cause anything if the given thread does not contain any interruption points. Whenever a thread reaches an interruption point it will check whether the interrupt() method has been called.Only if it has called,then only exception is thrown.

This instruction in the program "thread _group _join_all " will wait all threads to complete and then join. It blocks the current thread(main() function) until both the threads, has been terminated.

The following are a few things to keep in mind before running the code for FDD .

- The daughter board frequency ranges must be checked and used appropriately.For example, RFX daughter board does not support 900MHz center frequency. Only SBX daughter board must be used at this frequency.
- The antennae frequency range must be checked before using them.
- The Gigabit ethernet cable must be checked.
- Remove received data and transmitted data files if exists (frame.txt and data.txt respectively) before running the code.
- At one end i.e usrp1(Do this while the running the c-code FDD).

CHAPTER 5

CONCLUSION

5.1 Conclusion and future work

In this project, two-way link(full-duplex) have been developed on SDR using the OFDM modulation scheme.This is the basic model for other systems.

Channel feed back can also be implemanted by using this model.In case of channel feedback,initially one transmitter(say TX1) will transmit the randomly generated data and at the receiver side(say RX1) channel can be estimated and then phase of the channel can quantised. This quantised channel phase information will be fedback(TX2 will transmit this channel information back to transmitter TX1).

Due to CSIT available at the transmitter end, transmitter(TX1) will allocate the constellation/power to the different channels according to their status. For example, allocate more power for the good channel and less power for the bad ones.

By knowing CSIT at the transmitter end, techniques like water filling,precoding etc. can be implemented.

REFERENCES

[1] ://files.ettus.com/uhd docs/doxygen/html/index.html

[2] ://code.google.com/p/microembedded/downloads/list?q=label:SBX

[3] ://code.ettus.com/redmine/ettus/projects/uhd/repository/revisions/171e46cebe9c661246d6e313a40f38822d90bbb7/entry/host/examples/benchmark$_r$*ate.cpp*

[4] Michele Morelli and Umberto Mengali, *âĂİA Comparison of Pilot- Aided Channel Estimation Methods for OFDM Systems,âĂİ Signal Processing, IEEE Transactions, vol.* 49, no. 12, 3065 - 3073, Dec 2001.

[5] Schmidl, T.M. and Cox, T.M *.,âĂİRobust frequency and timing synchronization for OFDM,âĂİ Communications, IEEE Transactions, vol.* 45, no. 12, 1613 - 1621, Dec 1997