Politecnico di Torino

Faculty of Engineering
Master Thesis

Implementation of a complete radar system on the NI USRP-2944R software defined radio platform

Supervisor:
Prof. Riccardo Maggiora

Candidate:
Redon Rukaj

April 2, 2018

Abstract

Software defined radios are one of the latest trends in the research world on communications systems. These systems allow implementation of various applications from WiFi, LTE to radars. In particular SDR can be implemented as radar systems where many complex signal processing procedures have to be employed to achieve high detection probability. This master thesis project discusses the implementation of a radar system by means of an USRP (Universal Software Radio Peripheral) 2944-R of NI (National Instruments). The state of the art of both radar waveform processing and also SDR projects are explored to have a better insight of what has been done so far in the area. Taking into account this objective, we implemented a complete radar system able to transmit pulsed LFM (Linear Frequency Modulated) signals and receive the echos from targets.

LabVIEW Communications System Design Suite 2.0 software and also the sample project that comes with it for USRP 2944-R device were used for the implementation of the system. The programming part was done by means of LabVIEW Comms. 2.0 using 'G' dataflow programming language. More specifically, we implemented '.gvi' functions in LabVIEW Comms. able to run the radar prototype. We devoloped functions able to perform waveform generating, radar signal processing methods and real-time data acquisition. In the software panel we put parameters such as number of samples, sampling frequency, start and end frequency for setting up the waveform pulses in transmission. The software front panel is very user friendly and the configuration of the parameters can be done easily depending on the user requirements/application. Furthermore, we implemented the radar signal processing part for range processing in a function able to take the transmitted and received waveforms, mix them together, filter in a lowpass and perform a FFT (Fast Fourier Transform) to the signal at the filter outcome. According to LFM radar theory, the beat frequency of the FFT peak corresponds to the radar to target distance. All this process happens at real-time, thus letting the radar prototype track the targets' positions at anytime.

In order to verify the correctness of the range processing FFT algorithm implemented, we tested the device with a loopback. We connected the transmitter on TX1 with the receiver on RX2 in the RF0 module via 2 coaxial cables of different lengths. The lengths of each cable were $0.8m$ and $40.75m$. The time delay between the transmitted and received signal would depend on the cable length. The results of the FFT algorithm, considering the range resolution we

had for the signal transmitted were satisfying. The tests showed that a radar system could be implemented with a SDR device.

Furthermore, we implemented a method for targets' Doppler estimation by using another FFT. The Doppler processing method is done by saving FFTs coming from many pulses in a matrix, taking the range bins of interest (where targets are previously detected from range processing), keeping the columns corresponding to those range bins and then performing the second FFT to those columns. By doing this, the resulting FFT peaks would correspond to targets' Doppler frequency, thus giving velocity information in correspondence with the carrier frequency used. Also, in this case we simulated Doppler frequency to the received data by mixing them with a sine wave at a constant frequency. Afterwards, we verified the correctness of the implemented processing method.

The USRP 2944-R device has a frequency span from 10 MHz to 6 GHz and 160 MHz of real time bandwidth. It also has an internal oscillator with a reference clock of 10 MHz used for synchronization. These hardware limitations influenced the overall radar system performance. Radar performance metrics such as range resolution, Doppler resolution, maximum Doppler frequency detectable, maximum unambiguous range had also been bounded due to these device limits. In Fig. 1 is shown the block diagram of NI USRP 2944-R device.

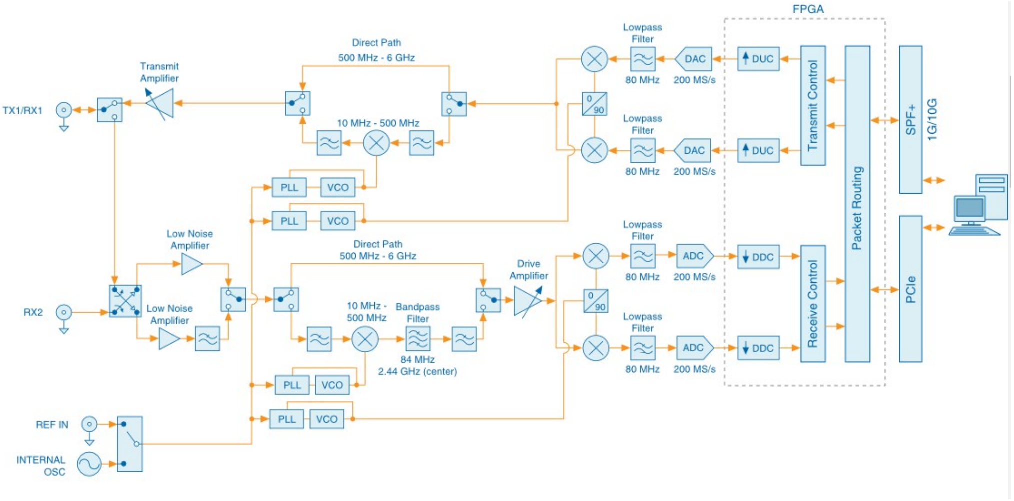


FIGURE 1: USRP 2944-R block diagram

This block diagram gives a better picture of all the components in the software defined radar prototype system. The system consists of 3 main parts, the host PC where the software is runned, the Field Programmable Gate Array (FPGA) and the RF module.

In conclusion, the project showed that theoretical performance metrics could be achieved also in practice. The prototype has shown some very good achievements in range profiling and also some drawbacks on other parts such as Doppler frequency detectability due to synchronization limits on transmit and receive chain. This limits were demonstrated to come from LabVIEW software calculations speed. These limits could be overcame by implementing the algorithms directly in the SDR FPGA. Moreover, this project work showed also that complex and expensive communications systems such as radars could be implemented with a comparable low-cost software defined device, two antennas, a desktop and a software at university laboratory conditions.

Preface

This project is carried on by Redon Rukaj and concludes the Master Thesis as the last requirement for the Master's degree in Communications and Computer Networks Engineering at the Department of Electronics and Telecommunications, Faculty of Engineering, Polytechnic University of Turin. The thesis project is developed at the RADAR Laboratory. The basis for the project is the theme; RADAR Signal Processing and Computing. References made before a full stop regard the sentence and after full stop regard the paragraph. The LabVIEW Communications Design Suite 2.0 version used during the project work was installed using academic license.

Acknowledgements

This project is carried on at the Department of Electronics and Telecommunications, Polytechnic University of Turin from November until April. It fulfills the last requirement for the Master of Science degree in Communications and Computer Networks Engineering. I am grateful to a lot of people for their practical and psychological support during my studies at PoliTO. I would like to mention some of them.

First of all, I would like to thank my supervisor Prof. Riccardo Maggiora, for his guidance, sharing of ideas and technical support into the area of radar signal processing. He taught me to think deeply, to be creative and taught me skills and methods to present ideas and research. I am thankful for his patience and help.

Furthermore, I would like to express my gratitude to my fiancee Jona for her love and caring which she constantly has shown towards me during these months. Finally, my most sincere gratitude goes to my parents and my brother without whom this would have never possible. Their encouragement and constant support mixed with their endless love has given me the confidence to overcome difficulties and complete the Master's degree studies. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	**A**nalog to **Digital** **C**onverter
AF	**A**mbiguity **F**unction
AWGN	**A**dditive **W**hite **G**aussian **N**oise
BIOS	**B**asic **I**nput **O**utput **S**ystem
CPI	**C**oherent **P**rocessing **I**nterval
CPU	**C**entral **P**rocessing **U**nit
DAC	**D**igital to **A**nalog **C**onverter
DDC	**D**igital **D**own **C**onverter
DFT	**D**iscrete **F**ourier **T**ransform
DMA	**D**irect **M**emory **A**ccess
DOD	**D**irection **O**f **D**eparture
DOA	**D**irection **O**f **A**rrival
DVB-T	**D**igital **V**ideo **B**roadcasting - **T**errestrial
DUC	**D**igital **U**p **C**onverter
FFT	**F**ast **F**ourier **T**ransform
FIFO	**F**irst **I**n **F**irst **O**ut
FMCW	**F**requency **M**odulated **C**ontinuous **W**ave
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GS	**G**iga **S**amples
GPS	**G**lobal **P**ositioning **S**ystem
LFM	**L**inear **F**requency **M**odulation
LNA	**L**ow **N**oise **A**mplifier
LO	**L**ocal **O**scillator
LTE	**L**ong **T**erm **E**volution
MIMO	**M**ultiple **I**nput **M**ultiple **O**utput
MTD	**M**oving **T**arget **D**etector
NI	**N**ational **I**nstruments
OFDM	**O**rthogonal **F**requency - **D**ivision **M**ultiplexing
OS	**O**perating **S**ystem
PC	**P**ersonal **C**omputer
PLL	**P**hased **L**ocked **L**oop
PRF	**P**ulse **R**epetition **F**requency
PRI	**P**ulse **R**epetition **I**nterval

RADAR	**RA**dio **D**etection **A**nd **R**anging
RAM	**R**andom **A**ccess **M**emory
RF	**R**adio **F**requency
SAR	**S**ynthetic **A**perture **R**adar
SDR	**S**oftware **D**efined **R**adio
SISO	**S**ingle **I**nput **S**ingle **O**utput
SLL	**S**ide**L**obe **L**evel
VCO	**V**oltage **C**ontrolled **O**scillator
VF	**V**elocity **F**actor
VI	**V**irtual **I**nstrument

Dedicated to my family

Chapter 1

Introduction

1.1 Radar Fundamentals

RADAR is the acronym for RAdio Detection And Ranging that is now used as an English noun. It is a system used to detect the presence of vehicles, aircrafts, ships, terrain, space vehicles in an unknown area, to investigate on the atmosphere, meteorology, ground penetrating etc. The underlying principle behind radars is to collect different electromagnetic waveforms reflected from objects and to process them in order to gain as much information as possible about those objects. Such information includes the object's presence, position, direction, velocity, distance. Fig. 1.1 shows a single input single output radar target scenario.

In Fig. 1.1, R is the distance between the radar and the target. It is a parameter which radars should estimate with low error in order to find the exact location of the target. In most of radar applications, the purpose of their usage is to find the target location. Therefore it is a very critical parameter to be estimated. The true location and the radar's estimate, if not the same should be really close to each other. In estimation theory, that means that the estimator accuracy should be very high. Signal processing techniques are used in radar systems to make that possible.

The relation between the radar range and time of propagation is given by Eq. 1.1:

$$R = \frac{c\tau}{2} \tag{1.1}$$

where c is the speed of propagation (speed of light) and τ is the time delay between the transmitted and the received signal (echo).

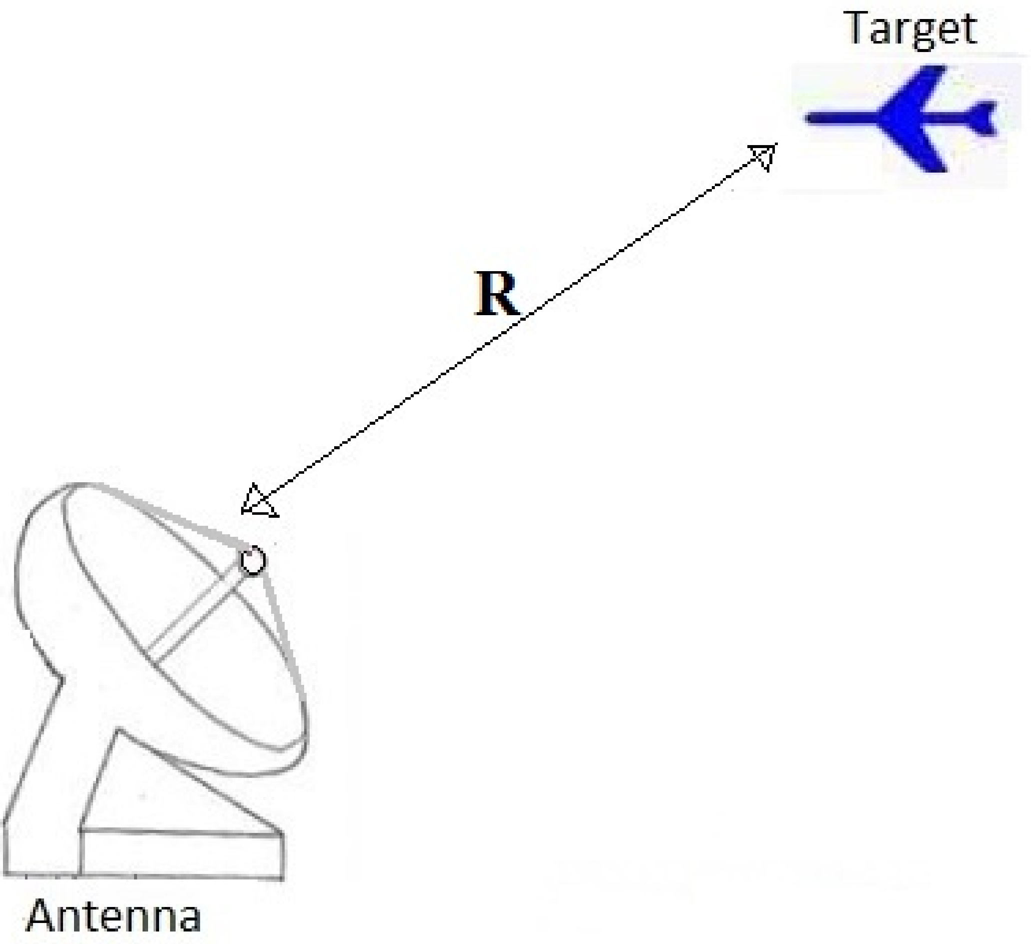


Figure 1.1: Single input single output monostatic radar scenario [1]

Eq.?? states that the terms range and delay are the same and they are used to imply each other in radar terminology. When we know one of them the other is also known. Range is one of the two parameters which we want to estimate by using the radar prototype implemented with a SDR. The other one is its velocity with respect to the radar. In order measure the velocity we need a model on how R changes with time. Let us assume we transmit the signal

$$s(t) = cos(2\pi f_c t) \tag{1.2}$$

and receive a delayed version of it

$$s(t-\tau) = s(t - \frac{2R_c}{c}) \tag{1.3}$$

Let us suppose the target has a constant velocity and therefore the following relation stands:

$$R = R_c + vt \tag{1.4}$$

By substituting Eq. 1.4 into Eq. 1.3 we get:

$$f_D = \frac{2f_c v}{c} = \frac{2v}{\lambda_c} \tag{1.5}$$

where λ is the wavelength corresponding to f_c the carrier frequency and f_D is the Doppler frequency. Eq. 1.5 states that the Doppler frequency is used to imply the target's velocity or the other way around.

So, we can conclude that the two key concepts in radar are the time of flight (time delay) and Doppler frequency. If we have information on these two parameters then we know the target to radar distance and the target relative velocity with respect to the radar.

The term radar was first introduced by the US Navy in 1941. Since then the radar has been a very fascinating developing field of research for its application in a wide variety of uses such as in military, remote sensing, air control, ships and space safety, highway safety etc. The two main functionalities of a radar are detection and the above mentioned target parameters' estimation.

The external environment under surveillance most of the time is not perfect and the noisy components usually are clutter and jamming. Clutter is every object that produces an echo which is none of the radar's interest. radar jamming is defined as radio frequency signals originating from sources outside the radar, transmitting in radar's frequency and thereby masking targets of interest. Jamming can be intentional or unintentional when friendly forces operate their equipments transmitting at the same frequency range. The ability of the radar system to surpass these unwanted signals is defined by its SINR (signal to interference plus noise ratio).

1.2 Classification of Radars

Radars can be classified in a number of different categories. According to the working principles of their antennas they are classified into Phased array or MIMO radar, according to the way of operation they are classified into passive or active radars, according to the type of transmitted signals they are classified into Continuous wave or Pulsed radar and finally according to the position of receivers and transmitters with respect to each other they are classified into monostatic or bistatic radar. In the following subsection we would like to emphasize the differences between a continuous wave and a pulsed radar as it is in our interest to do so because in this project we will prototype a monostatic pulsed radar system. In Subsection 1.2.1 and 1.2.2, we emphasize the differences between a CW or pulsed radar and monostatic or bistatic radar, respectively.

1.2.1 Continuous Wave or Pulsed Radar

Continuous Waveform (CW) radar systems use modulated or unmodulated continuous signals for transmission. The main advantage of CW radar is that energy is not pulsed so these are much simpler to manufacture and operate. They have no minimum or maximum range, although the broadcast power level imposes a practical limit on range. CW radar systems maximize total power on a target because the transmitter is broadcasting continuously. Transmit receive switch (circulator) or separate antennas are used for transmission and reception. Some simple CW radars are not able to measure the range but the Doppler frequency shift of echo signal is useful for indication. One example application is its use in aircraft navigation for speed measurement.

Pulsed radar systems use modulated pulses as signals for transmission. They are able to indicate the range of the target and also achieve much higher peak power for transmitting. A duplexer is used for transmission and reception. The performance can sometimes be affected by stationary targets. The circuits used in these systems are comparatively complicated.[2]

Compared to CW radar it the pulsed radar uses higher transmitting power, more complicated circuits and the performance is affected by stationary targets. Anyway, the number of applications of pulsed radar is bigger compared to CW radar.

The factor which determines whether a radar is CW or pulsed is the duty cycle. In CW radars the duty cycle is 100% whereas is modern pulsed radars it usually goes in the interval from 10% – 20%.

1.2.2 Monostatic or Bistatic Radar

Radar systems can be divided in two different groups: monostatic or bistatic radars, by considering the positions of the transmitter and receiver. In monostatic radar systems, the transmitting and receiving antenna are at the same position. This fact implies that the transmitting and receiving signal follow the same path and so the DOD (direction of departure) and DOA (direction of arrival) are also the same as well as the target distance to transmitter and receiver. The radar system in Fig. 1.1 is an example of a monostatic system. Whereas in a bistatic radar the transmitter and receiver are separated by a considerable distance and thus, the DOD and DOA are not the same and also the target distance to Tx and Rx. These facts imply the target localization in a bistatic radar is more complicated because of the number of parameter for each target are more than the ones used for monostatic radar targets.

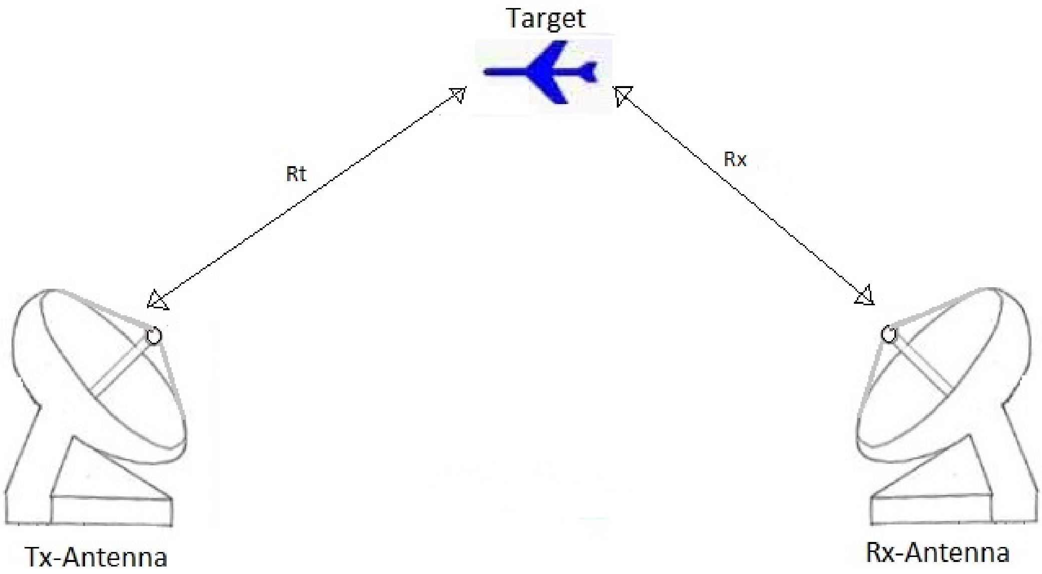


Figure 1.2: Single input single output bistatic radar scenario [1]

In Fig. 1.2, Rt is the distance from the transmitting antenna to thetarget and Rx is the distance from the target to the receiving antenna inthe case of bistatic radar. These distances are non necessarily equal and most of the times they differ from each other. Therefore, these parameters are both to be estimated in bistatic radar whereas in monostatic radar the parameter R is the only one to be estimated. This is a great advantage together with the versatility since one location is required for monostatic radar compared to the two locations needed for bistatic. Throughout this thesis only monostatic radar systems will be considered.

1.3 Software Defined Radio Intro

SDRs are one of the latest trends for many applications in the radar field. Researchers are using them in order to devolop computer softwares capable of implementing components which traditionally were implemented in hardware. This drives to innovative, low-cost, versatile solutions for many applications. In the radar applications components such as mixers, filters, modulators and demodulators are implemented in hardware. In this project, these components will be implemented in software programs. By means of SDR we can implement a radar system capable of tracing objects, which is the most important mission of a radar, with a low cost. From our point of view this is very profitable considering that most of the hardware used to implement the above mentioned parts could cost up to many thousands or even millions of Euros. Furthermore, the software defined radar can be multipurpose, meaning that one could implement functions able to make the system achieve not only one goal. The hardware we used, the NI USRP 2944-R, for which we will talk about later in the Section 2.2 can be reconfigurable and therefore reused for other applications. Last but not least, many complex signal processing algorithms can be very easily implemented in software to make the software defined radar work. Considering that SDR is not a new concept, there has been a lot of previous works on the development of software defined radar systems and we would like to mention in Section 1.5 some interesting works on what has been done so far in this field.

1.4 Problem Statement

> *What are the characteristics of a complete radar system implemented in a software defined radio with a universal software radio peripheral 2944-R of National Instruments? Which are the limits imposed from hardware and software to the overall radar system?*

The purpose of this project is to implement a complete software defined pulsed LFM radar on the NI USRP 2944-R radio platform. All the components of the system such as mixer, filters, decimators are implemented by software in LabVIEW Communications System Design Suite 2.0, whereas the target device (USRP-2944R) is the hardware device able to transceive radar signals for target location. We will investigate on which limitations in terms of performance the radar has and also their source. It is evident that the sources of these limits will be coming from the hardware specifications and also the software processing.

1.5 State of the Art Overview

SDR platforms are widely used from researchers all over the world to investigate on different applications. One of the broad fields investigated by employing SDRs is also the radar. Software defined radars have been implemented to investigate on many radar topics, such as waveform design, radar hardware, radar signal processing, cognitive radar, radar network systems etc. In this section we make a state of the art overview on software defined radars.

Researchers have presented attempts to implement different types of radars in SDRs, such as synthetic aperture radar (SAR) [3] or medical imaging radar in [4] where the authors present a low-cost reconfigurable microwave transceiver which according to them paves the way to replace costly and bulky vector network analyzer used in the research of microwave-beased medical systems. Other researchers have investigated the theoretical radar performance limited from SDR device capabilities and also demonstrating improved target resolution on the USRP NI2920 with respect to the first generation platform [5].

In many other works FMCW waveforms have been employed for software defined radar implementation. In [6] the authors have proposed a GNU-radio based software defined FMCW radar for weather surveillance application working at 2,1 GHz center frequency with a 750 kHz bandwidth whereas in [7] they propose a 24 GHz software defined measurement system for automotive applications by using FMCW for ranging and pseudorandom (PN) coded pulse radar for intervehicular communications. FMCW radar for drone detection and surveillance purposes on SDR using GNU Radio on the USRP B210 is presented in [8]. In [9] they present a project work on the implementation of a FMCW radar on a SDR for range and speed estimation to be used as an anti-collision radar for automotive applications.

Apart from FMCW radar in [10] a software defined passive radar system based on the use of DVB-T signals is proposed and live data detection results are presented in order to demonstrate the detection capabilities.

In [11], polyphase codes are compared to multitones in simulations. The predictable conclusion was that multitones was superior for target detection.

Researchers have not only investigated on SISO software defined radars but also on MIMO ones. In [12] the authors discuss design, implementation and validation of a software defined radar test bed to derive MIMO waveform adaptive radar research. Also in [13], the authors have implemented a real-time 2-Dimensional localization and velocity measurements using simultaneous 4×4 transmit and receive antenna configuration OFDM Radar on a SDR.

Chapter 2

System Setup

2.1 System Configuration

The system setup consisted of a USRP-2944Rio device (the target), one connector for transmission and one for reception, a PC (the host) and a PCIe express x4 connector between the target and the host. We also used a digital storage oscilloscope from Agilent Technologies working up to 60 MHz and 1 GS/s for signal visualization. A schematic representation of the system setup is shown in Fig. 2.1.

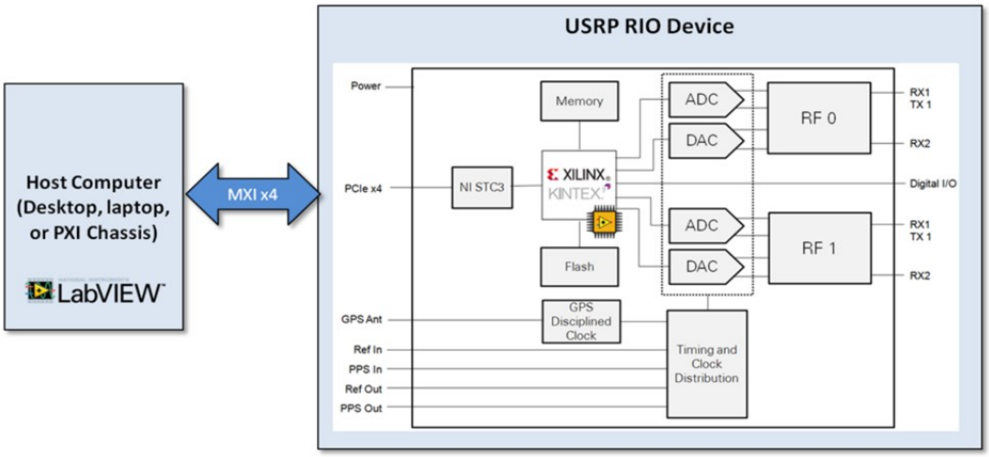


Figure 2.1: System components

The NI USRP 2944-R incorporates Xilinx Kintex-7 FPGA for local, real-time signal processing. The Kintex-7 FPGA implements a reconfigurable LabVIEW FPGA target that incorporates DSP48 coprocessing for high-rate, low-latency applications.[14]

2.2 Hardware Equipments

The USRP 2944-R device has a frequency span from 10 MHz to 6 GHz and 160 MHz bandwidth on carrier frequencies, with two full duplex channels. In our system, the device was connected to a desktop with a PCIe express x4 which allows up to 800 Mb/s streaming data. The specifications of the USRP-2944R device together with the connector to the host PC are provided in the Table 2.1.

Specifications
USRP NI-2944R
Frequency Range 10 MHz to 6 GHz
LabView Communications Design Suite 2.0
Kintex-7 FPGA
Up to 800 MS/s PCI Express x4 connection to the host
Up to 160 MHz real-time bandwidth in Tx and Rx

TABLE 2.1: Characteristics of software and hardware used to simulate the software defined radar

Fig. 2.2 shows the actual hardware devices used for this project. The cost of the USRP 2944-R device is about 8000 Euros. Summing together the prices of all the other hardware equipments used in this project to implement the radar prototype, the overall cost goes up to 10000 Euros.

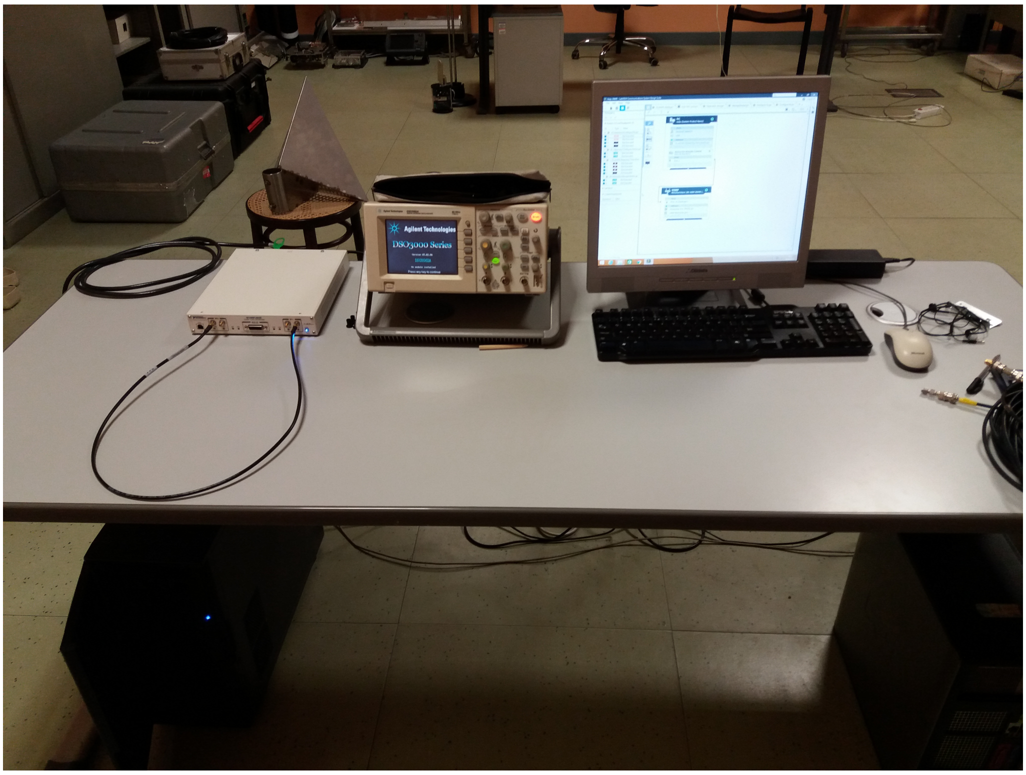


FIGURE 2.2: View of all system components

The NI USRP-2944R device specifications are crucial to the overall radar prototype system. As we will discuss in the next chapters the specifications highlighted at Table 2.1 will be seen as upper bounds or constraints which will limit the system performance with respect to the radar metrics.

2.3 Architecture of SDR Platform

For the implementation of the radar one USRP 2944-R device was used. The architecture block diagram of the USRP 2944-R is presented in Fig 2.3. The scheme was taken from online resource at National Instruments webpage. [15]

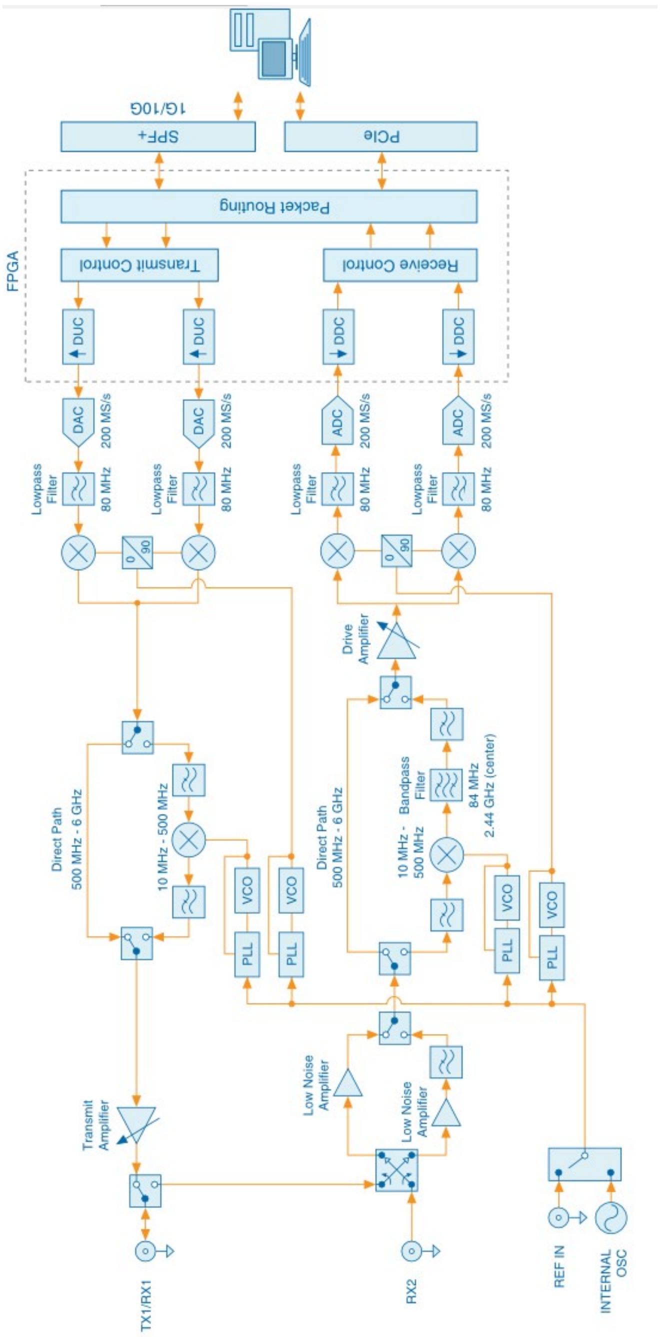


FIGURE 2.3: USRP 2944-R block diagram

In the following two paragraphs is given an explanation of individual block functionalities.

Starting from right to left at the transmit path the PC synthesizes I and Q signals and transmits them over a PCIe connection to the USRP 2944-R device. At the FPGA the packets are routed and controlled before the DUC mixes, filters and interpolates the signal at $400MS/s$. After the DUC, the signal passes at the RF module whare a DAC converts the signal into analog at $200MS/s$ sample rate. The lowpass filter reduces noise and high frequency components in the signal. After that, the mixer upconverts the signal to a software panel controlled RF carrier frequency. The PLL and VCO blocks make sure that the device clocks and LO are frequency locked to a reference signal which in our case is internally generated at $10MHz$ but is can also be external, for example a GPS antenna. In the end of the transmit path the signal is amplified to be ready to be sent at the transmit connector.

After the transmission, starting from left to right in the receive path the signal is received at the receive connector. Depending on the frequency, it is either amplified from a LNA or a LNA plus lowpass filter combination. Depending on the frequency the signal travels at the upconvert path whereas the PLL and VCO have the same function for the receive chain this time. The signal is once again amplified from a drive amplifier before the quadrature mixer downconverts the signal to baseband I an Q signals. The lowpass filter reduces noise and high frequency components in the signal. After the the ADC converts the signal to enable its employment at the digital world. The DDC mixes, filters and decimates the signal and then the samples are passed through a PCIe connection at the host PC.

The software used to program the device was Labview Communications Design Suite 2.0 of National Instruments. It uses a graphical programming language called "G" and which was designed to highlight the dataflow in the programs. The type of files created have ".gvi" extension name.

In this project, all the transmission, reception and processing is coherent. Coherence in a radar system is defined from the phase angles of transmitted and received signals. The phase angles of the signal pulse's are all defined from a reference in a coherent radar.

2.4 Software Application

The software application used to define the radar prototype on the NI USRP 2944-R radio platform was LabVIEW Communications System Design Suite 2.0. This software application is provided from NI, released on January 2016 and its compatible to run on most of the NI hardware which are software defined. Its filesize is around $8Gb$ and the PC where it will be installed have to fulfill some requirements. The OS must be Windows 8.1 64-bit or Windows 7 64-bit and the software requires at least $8Gb$ of RAM.

2.5 PC Requirements

The host PC used to communicate with the SDR device has to fulfill some requirements to be eligible to do so. Apart from requirements for software application mentioned in the above Section 2.4, it needs to comply with some others to communicate with the USRP-2944R. The PC we used for this project had the following properties:

- Installed RAM of 24,0 GB
- Intel(R) Core(TM) 940 i7 CPU 2,93 GHz dual core
- Windows 8.1 Enterprise 64-bit OS

We have to point out that the PC is very critical for the ongoing work flow of the project. At the beginning of this project, we firstly started working with another PC which did not recognize and so could not communicate with the USRP-2944R. Apart from the requirement of 8 GB of RAM the PC must also be fast enough and have graphical video driver slot available. The missing connection between that PC and the device came as a result of BIOS incompatibility. One solution may be to consider installing the BIOS compatibility driver software from NI but if even that does not work than from our experience without doubts we could say that the PC in use can not be part of that particular system you are working on.

So, in order to be certain of the eligibility of the PC one must also look at the requirements when downloading the hardware drivers for installation at NI webpage. Also, during the test measurements we noticed that the Labview Comms. software used most of the CPU part. It is crucial that if one

wants to run complex signal processing algorithms on the processor by using this software-hardware setup the CPU of the PC must be one of the latest and fastest in the market.
One PCIe x4 port was physically installed to the PC in order to have a faster connection with the device through an MXI Express four-lane cable. The manual for installation of the connector can be found at: `http://www.ni.com/pdf/manuals/371976c.pdf`. The cable in usage was a 'NI 19599A-03 MXI-Express X4 3 Meters 1628 MLX'. PCIe Express x4 connection back to the system controller allows up to 800MB/s of streaming data transfer to the PC.

Chapter 3

Software Implementation

For the implementation of the software programming, we started by using the NI USRP-2944R ($160MHz$) sample project provided together with LabVIEW Comms. application software. We used the sample project as a basis on where to build the programming for our radar prototype system. The sample project could perform basic signals transceiving functionalities in different modes. It could transceive radio signal in continuous or finite mode. Also there are implemented different trigger choices for the transmitter or receiver. Dependent on user or application requirements the system triggers choice decides when the signal transmitting or receiving will happen. We kept these functionalities provided by the NI USRP-2944R sample project but also provided some essential modifications which would enable the SDR to work as a radar system prototype.

3.1 Project panel

The sample project came along with many 'G' dataflow programming files which are runned at LabVIEW Comms. and have extension file names '.gvi'. From now on we will refer to these files only as gvi. There were many gvis runned at the host. In our modified project we have only one upper-level gvi which triggers all the software and calls the necessary subvis (a function written in gvi block diagram) when it runs. In Fig. 3.1 is shown the project panel at the programming software.

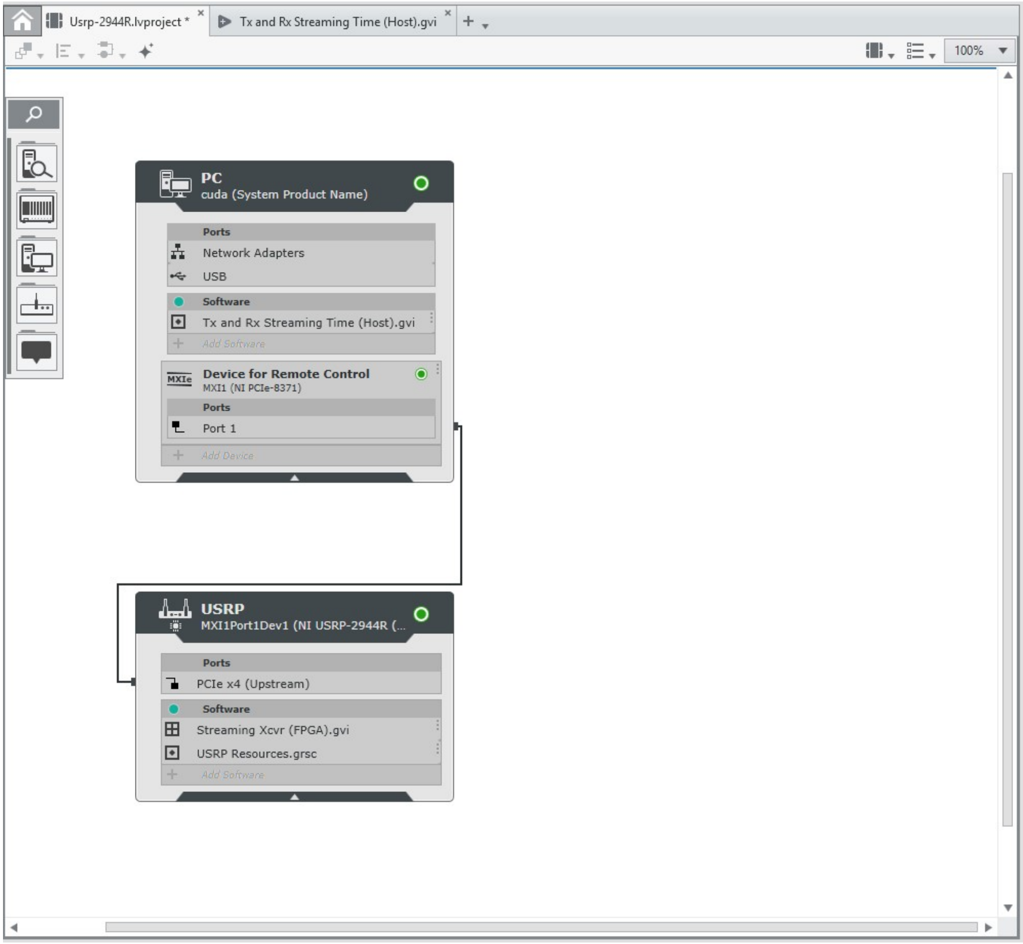


FIGURE 3.1: The project panel at LabVIEW Comms software

As shown in the above Fig. 3.1, the host PC is connected to the USRP 2944-R device through a PCIe connection. At the host PC and at the FPGA are runned the two main programs which trigger all the system to work. The USRP resources.grsc file contains all the input and output DMA FIFO and the FPGA derived onboard clocks information. In Fig. 3.2 is shown the content of the file used in our project.

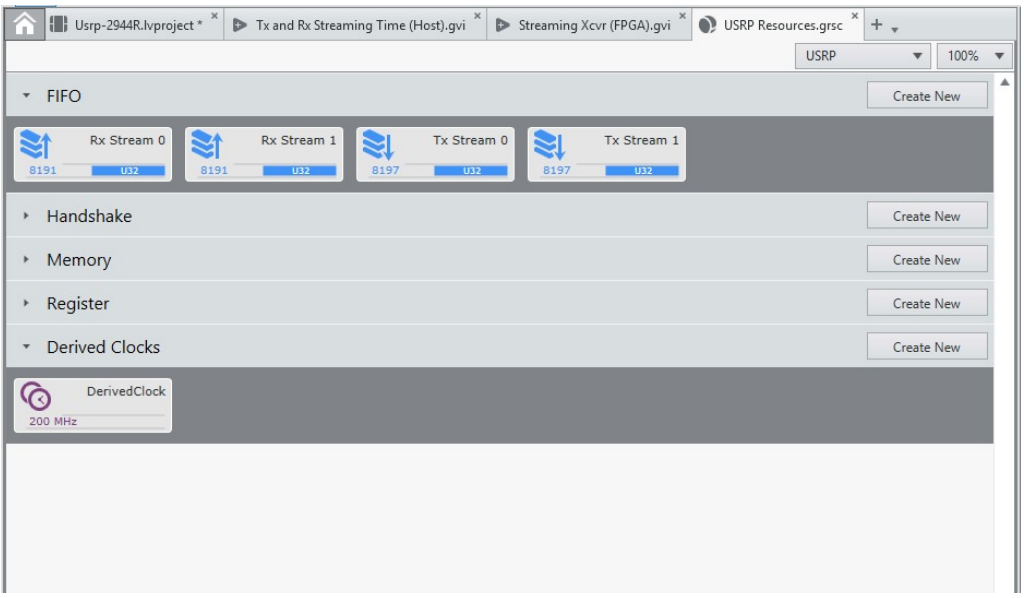


FIGURE 3.2: The project panel at LabVIEW Comms software

With difference from the initial sample project, we increased the transmission and reception FIFOs for each channel from 1024 to 8192 as shown in Fig. 3.2 and also derived a $200MHz$ FPGA onboard clock with the intention to increase the FPGA throughput. The default clock was set at $40MHz$. The process of rebuilding the FPGA took about 1 hour and a half. Fig. 3.3 shows the Streaming Xcvr (FPGA).gvi main block diagram.

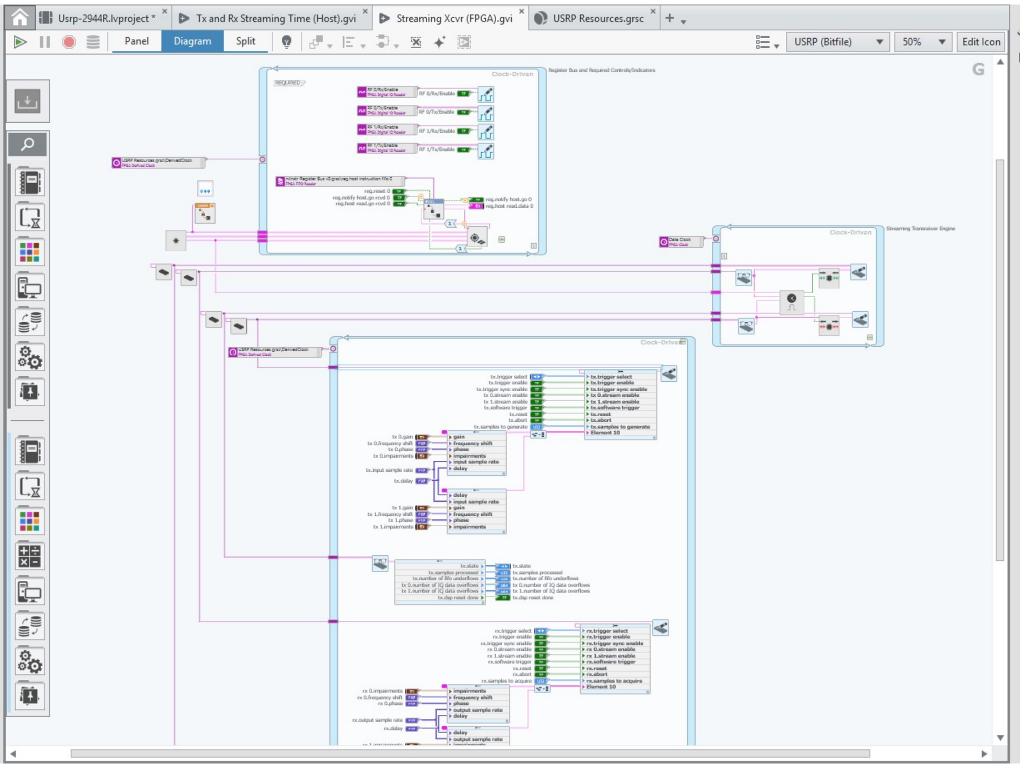

Usrp-2944R.lvproject *
Tx and Rx Streaming Time (Host).gvi
Streaming Xcvr (FPGA).gvi
USRP Resources.grsc
Panel
Diagram
Split
USRP (Bitfile)
50%
Edit Icon
Clock-Driven
Register Bus and Required Controls/Indicators
Streaming Transceiver Engine
Clock-Driven
Clock-Driven

FIGURE 3.3: Block diagram of the FPGA main gvi

The process of rebuilding the FPGA takes as target file the main FPGA gvi file shown in the above Fig. 3.3 and before running the FPGA building process it was necessary to save all the previous modifications to the project, close it and then restart again. Before or during the rebuilding process the FPGA file absolutely should have not been modified.

3.2 Operations at host PC

There are two ways in which the radar signal processing part can be implemented. One method is to implement the processing at the FPGA whereas the other is at the PC. As we stated in Section 1.5 both ways are explored from researchers on software defined radars. The method we used in this project was the implementation of all the radar processing blocks at the host PC. In Fig. 3.4

is shown the system flow of all the prototype components where the functions at the host PC are highlighted.

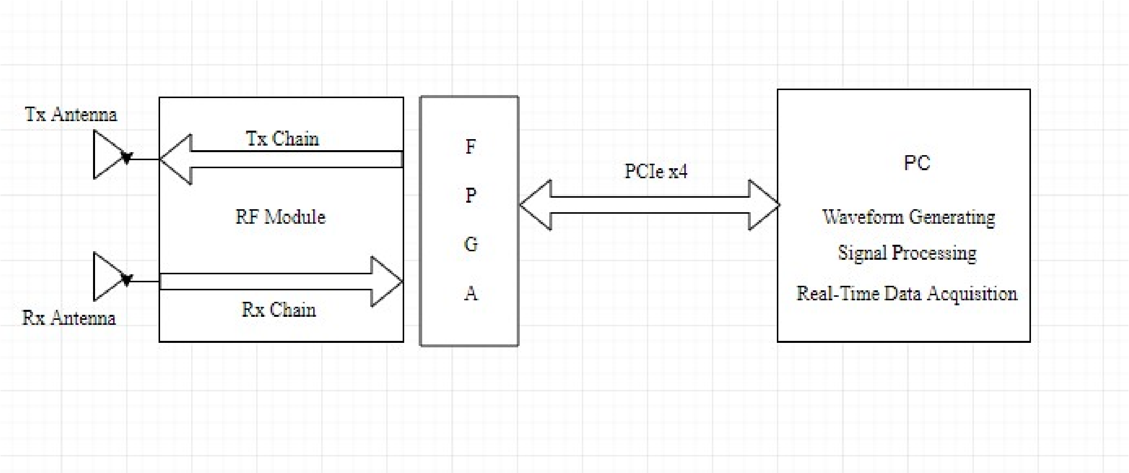


FIGURE 3.4: Block diagram of system flow highlighting the processing at the PC

At the host PC the LFM pulsed waveform is generated for transmission. Then, after reception the transmitted and the received signals are processed to estimate the range and Doppler frequency of the target. Therefore, all the radar signal processing algorithms which enable target detection and sensing are implemented at the host. During the testing tentatives we could implement functions to the software able to provide us with real-time data acquisition for further offline processing also in other programming languages.

3.3 Main Host Gvi

As discussed, we used the Labview Communications Systems Design Suite sample project for USRP-2944R data transceiving as a basis and modified it to be able to implement radar signal processing algorithms and methods. More specifically, we implemented one waveform generating gvi able to generate pulsed linear frequency modulated "chirps" signals for transmission. We added new parameters such as start and end frequency controller on the panel, so the user could control the transmitting pulse specifications. Furthermore, we modified

Tx and Rx Streaming Time (Host) .gvi program in the sample project to be able to transmit from TX1 on RF0 and receive the echo at RX2 on RF1. The Tx and Rx Streaming Time (Host).gvi is the main program which triggers all the system and its block diagram is shown in Fig. 3.5.

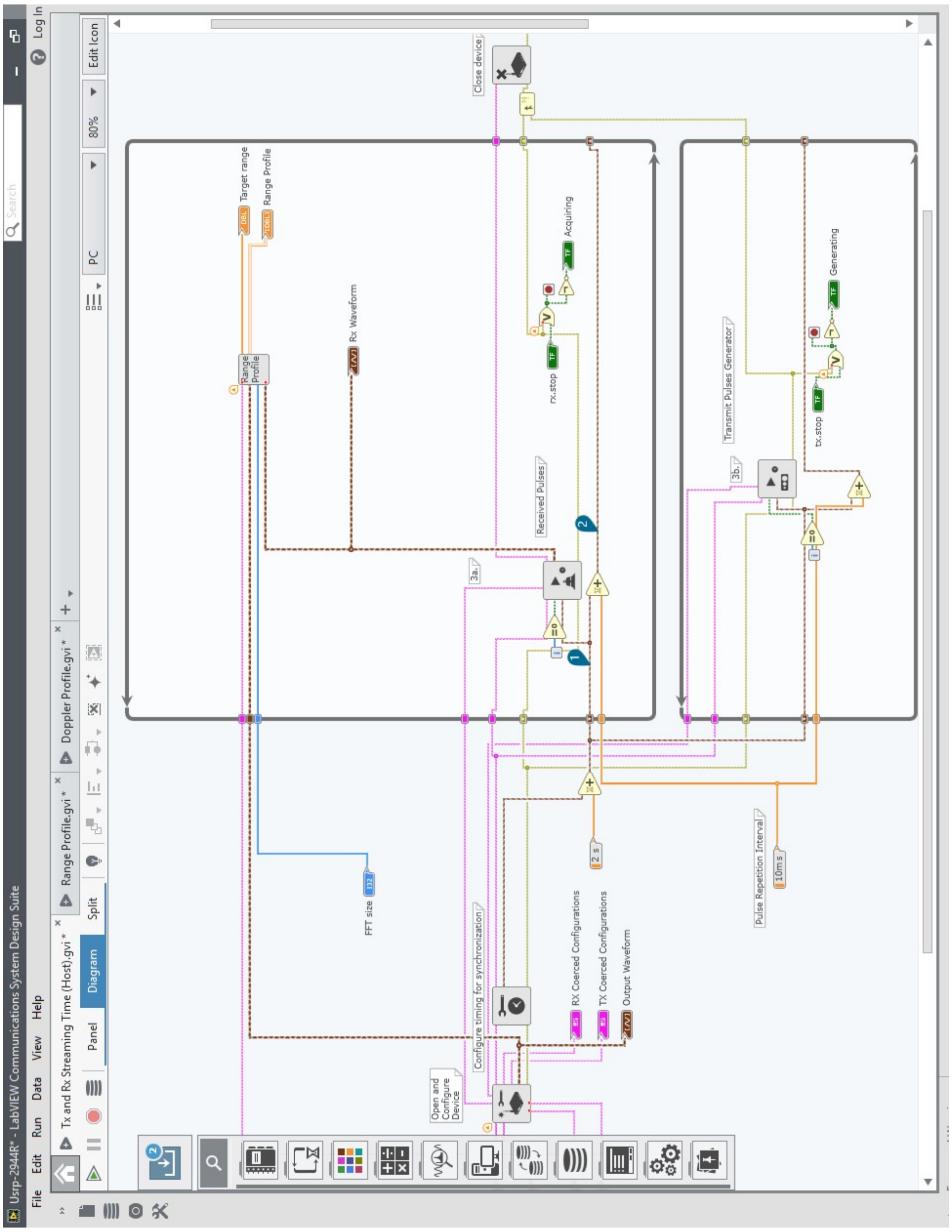

Usrp-2944R* - LabVIEW Communications System Design Suite
File Edit Run Data View Help
Log In
Search
Tx and Rx Streaming Time (Host).gvi *
Range Profile.gvi *
Doppler Profile.gvi *
Panel
Diagram
Split
PC
80%
Edit Icon
Close device
Target range
Range Profile
Rx Waveform
Acquiring
rx.stop
Received Pulses
3a.
Transmit Pulses Generator
3b.
Generating
tx.stop
FFT size
2 s
Pulse Repetition Interval
10m s
Configure timing for synchronization
RX Coerced Configurations
TX Coerced Configurations
Output Waveform
Open and Configure Device

FIGURE 3.5: Block diagram of the Tx and Rx Streaming Time (Host) gvi

This program is runned at the PC and so all its functionalities are calculated on the host. This gvi is the upper-level program but it has tens of subvis. One subvi file is also a gvi program with a specific function that is called on the main host gvi anytime this one is runned. Considering that we want to estimate the range and Doppler profile of targets, the time syncronization between the transmitter and receiver of the radar set was a requirement. The system radiates each chirp pulse during transmit time, namely the pulse width T_p, waits for returning echoes during listening time and then radiates the next pulse. The pulse width plus receiveing time is equal to the pulse repetition interval (PRI) which is a very critical parameter in radar operation. Eq. 3.1 states that the PRI is the inverse of PRF.

$$\text{PRI} = \frac{1}{\text{PRF}} \tag{3.1}$$

where PRF stands for Pulse Repetition Frequency. Depending on the PRF pulsed radar systems provide different performances in terms of Doppler and range. In Table 3.1 there is shown the relation between PRF and performance characteristics.

PRF	**Doppler**	**Range**
Low	Ambiguous	Unambiguous
Medium	Ambiguous	Ambiguous
High	Unambiguous	Ambiguous

Table 3.1: Doppler and Range ambiguities relation to PRF values

Looking back at the main host gvi the block diagram is subdivided in 4 main parts. The first blocks opens and configures the device whereas the second configures the time. The third block which is the two while loops in the diagram makes sure the signals are transmitted and received synchronously and peridically. The system is synchronous because the timing is set for both in transmission and reception in the same time. It is periodic because it radiates pulses every while loop. And the retrigger delay of every loop is controlled by the pulse repetition interval controller shown in the block diagram. At the reception loop, we have implemented the signal processing part which is performed in subvi functions called when the reception loop runs. Everytime, the system transceives a pulse it also performs targets range estimation. In Section 4.2 is discussed the range estimation method and the block diagram function used to do that. After a certain number of pulses the FFTs coming from each pulse are saved in a chart to calculate the Doppler frequency of targets. The number of pulses is defined by CPI. In Section 4.4 is discussed the Doppler estimation method.

3.4 Waveform Generating Subvi

One important subvi called everytime we run the system is the waveform generating gvi. It generates pulses to transmit every PRI seconds. We called this function 'Wavegenerator.gvi' and its block diagram flow is shown in Fig. 3.6.

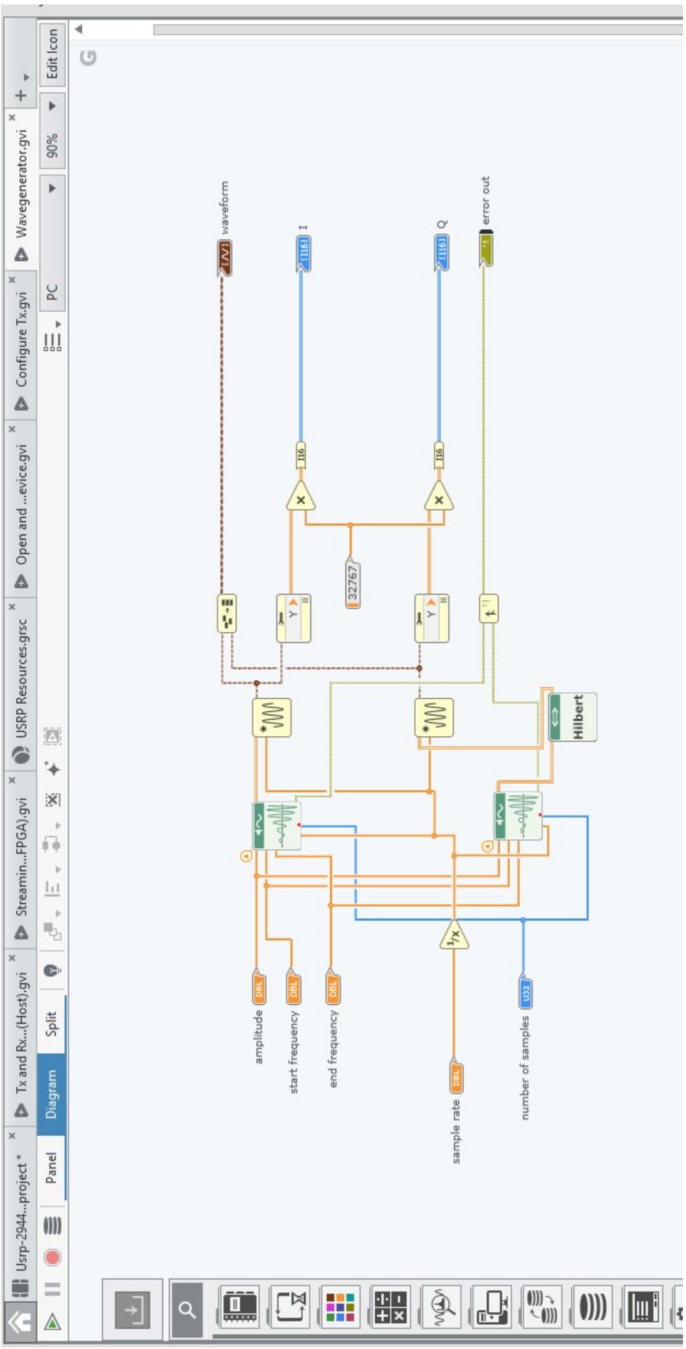


FIGURE 3.6: The block diagram of the transmit signal generator subvi

This subvi creates an LFM signal pulse for transmission. We used the LabVIEW Comms. built-in function 'chirp pattern'. It takes as input 5 parameters which are shown in Fig. 3.6. All these parameters are user-defined as they will also be called to appear in the main host gvi. The reason why we used two of the 'chirp pattern' built-in functions is to be able to get I and Q signals which are then passed to the FPGA. In order to get the quadrature signal we needed a chirp pattern with an initial phase at 90°. To impart a 90° shift to the initial phase of the chirp we used the Hilbert transform. The motivation behind the usage of this specific transform is provided in Appendix A.5.

After the chirp pattern creation we used another function in the block diagram which takes the samples of the chirp and the time interval between each sample as input and gives the waveform as output. The function is called 'build waveform' and it is built-in. Parameters used to generate the pulses are specified in Appendix A.6. Those pulse parameters are used for all test measurements unless otherwise stated.

Chapter 4

Waveform Design and Processing

4.1 Linear Frequency Modulated Waveform

The linear frequency modulation, or 'chirp' is a widely used waveform in radars. It has a rectangular amplitude modulation with pulsewidth T_p and a linear frequency modulation with a swept bandwidth B applied over the pulse. The time-bandwidth product of the LFM waveforms equal to TB (pulsewidth x Bandwidth). The LFM waveform is a single-pulse bandpass signal defined as:

$$x(t) = Acos(2\pi f_c t + \pi \alpha t^2), \qquad |t| \leq \frac{T_p}{2} \tag{4.1}$$

where A is the amplitude, T_p is the pulsewidth, f_c the carrier frequency, α the LFM slope which is $\alpha = \pm\frac{B}{T}$ where the plus sign is for a positive LFM slope (up-chirp) and minus stands for a negative LFM slope (down-chirp). The phase modulation is a quadratic function of time:

$$\phi(t) = \pi \alpha t^2 \tag{4.2}$$

Whereas the frequency modulation which is the instantaneous frequency deviation from the carrier frequency f_0 is expressed in terms of the phase modulation by:

$$f_i(t) = \frac{1}{2\pi}\frac{d\phi}{dt} = \alpha t = \pm\frac{B}{T}t, \qquad |t| \leq T/2 \tag{4.3}$$

The following plot shows the linearity relation between time and frequency in a up-chirp signal:

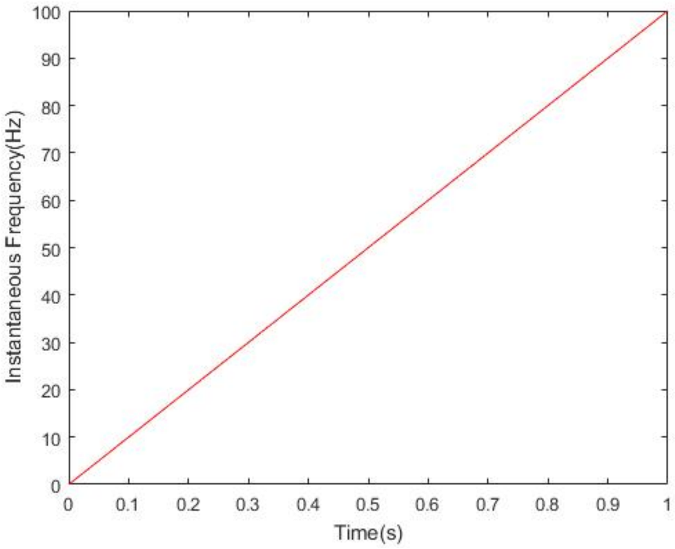


FIGURE 4.1: Instantaneous frequency for up-chirp signal with T $= 1$, $\alpha = \frac{B}{T} = 100$, and $t = [0, T/2]s$

The complex envelope of a LFM pulse, with instantaneous frequency $f(t) = \alpha t$, is:

$$u(t) = \frac{1}{\sqrt{T}} exp(j\pi\alpha t^2), \qquad |t| \leq \frac{T_p}{2} \tag{4.4}$$

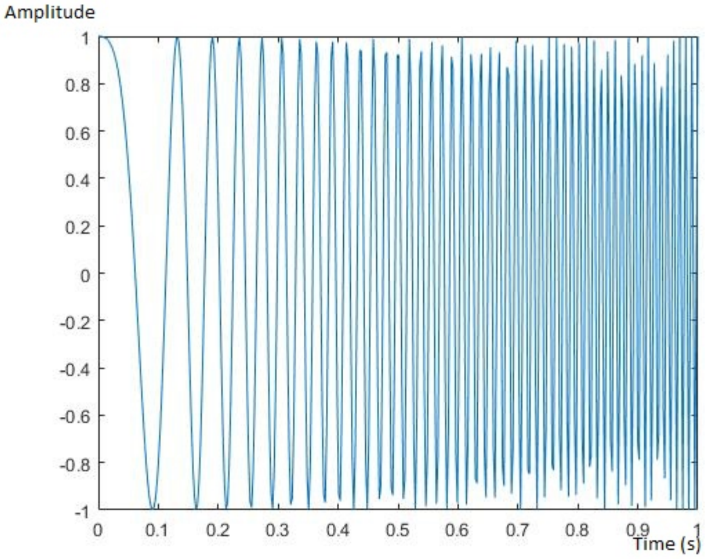


FIGURE 4.2: LFM waveform complex envelope for $\alpha = 100$, $T = 1$

In pulsed radar signal processing the ambiguity function is a two dimensional function of time delay and Doppler frequency $\chi(\tau, f_D)$ showing the distortion of a returned pulse due to the receiver matched filter (commonly, but not exclusively, used in pulse compression radar) and also due to the Doppler shift of the return from a moving target. The ambiguity function of a signal whose complex envelope is denoted by u(t) is defined as follows [16]:

$$|\chi(\tau, f_D)| = \left| \int_{-\infty}^{\infty} u(t) u^*(t - \tau) \exp(j 2\pi f_D) dt \right| \tag{4.5}$$

where * denotes the complex conjugate and j is the imaginary unit. In Fig. 4.3 is shown for visualization purposes the ambiguity function of a LFM pulse plotted in Matlab.

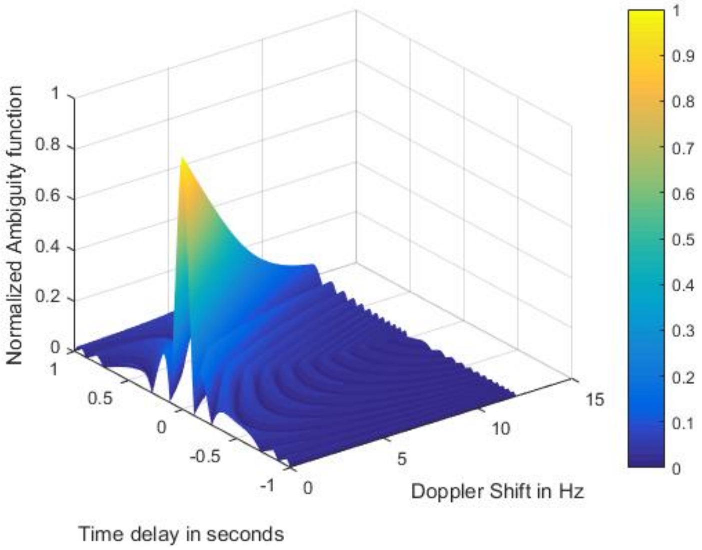


FIGURE 4.3: LFM waveform Ambiguity function for time delay $\tau = [-1, 1]s$ and Doppler frequency $f_D = [0, 12]Hz$

Note that the function is the modulus of a matched filter output when the input is the Doppler shifted version of the original signal to which the filter is applied. It follows that (0,0) coincides with the output when the input signal is matched to the nominal delay and Doppler of the filter and non-zero values of τ and f_D indicate a target from other range and/or velocity. A more concise way of representing the ambiguity function consists of examining the one dimensional zero-delay and zero-Doppler cuts. The cut of $\chi(\tau, f_D)$ along the delay axis by setting $f_D = 0$ is $R(\tau)$ the autocorrelation function of u(t) whereas the cut along the Doppler axis by setting $\tau = 0$ is independent of any frequency or phase modulation of the signal and is just defined as zero-delay cut. Therefore, again we plotted in Matlab the autocorrelation function and the zero-delay cut of a LFM complex envelope shown respectively in Fig. 4.4 and Fig. 4.5.

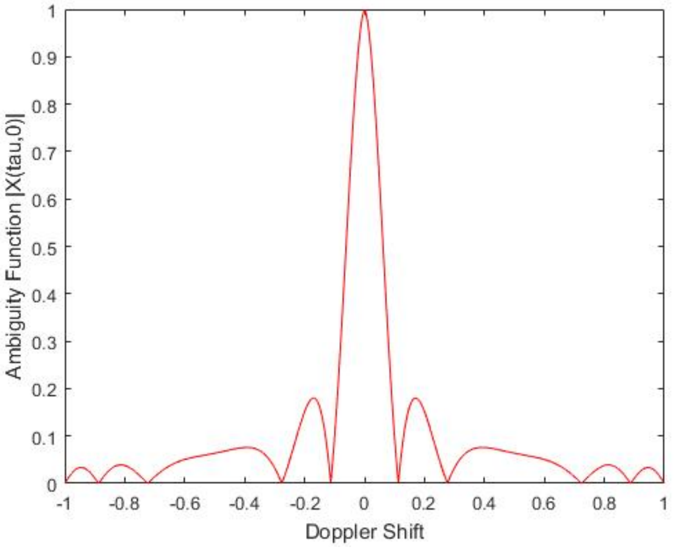


FIGURE 4.4: Autocorrelation function of LFM signal

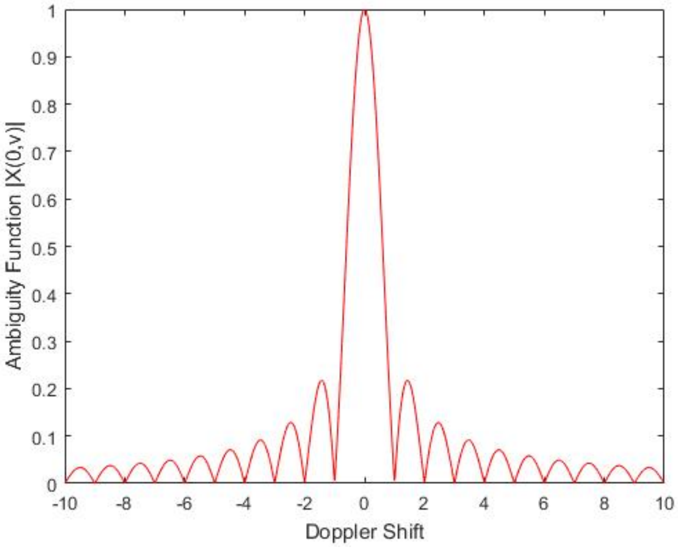


FIGURE 4.5: Zero-delay cut of LFM signal

The sidelobe level (SLL) of the autocorrelation function of LFM shown in Fig. 4.4 is -14.85 dB whereas the zero-delay cut of LFM has a SLL of -17.98 dB.The numerical evaluation of SLL is performed using the Matlab function discussed in Appendix A.2. Both these numbers are important for the detection process in radars. The sidelobes must be kept low in order not to have false alarms (sidelobes that surpass a defined threshold).

In Section 4.2 and Section 4.4, we discuss algorithms on how to perform separately range and Doppler processing respectively. The processing will be done in LabView Communications Design Suite software.

4.2 Range Processing

As mentioned, in many applications radars are used to detect objects far away from them. Location finding is probably the most important functionality of a radar. Therefore, in radar systems there is implemented a way to do that. Many signal processing procedures have been developed by engineers and researchers to find the time of flight or range profile for object location tracing. One of them is the discussed AF evaluation. Anyway, in practice the calculation of the AF is a very computationally heavy procedure and therefore it needs very powerful machines and softwares able to calculate it in real-time. So, in this project we developed an algorithm with LabView Communications Design software able to calculate the range profile and Doppler profile separately. In Fig. 4.6 is shown the function block diagram implemented in LabVIEW Comms. for range profile calculation.

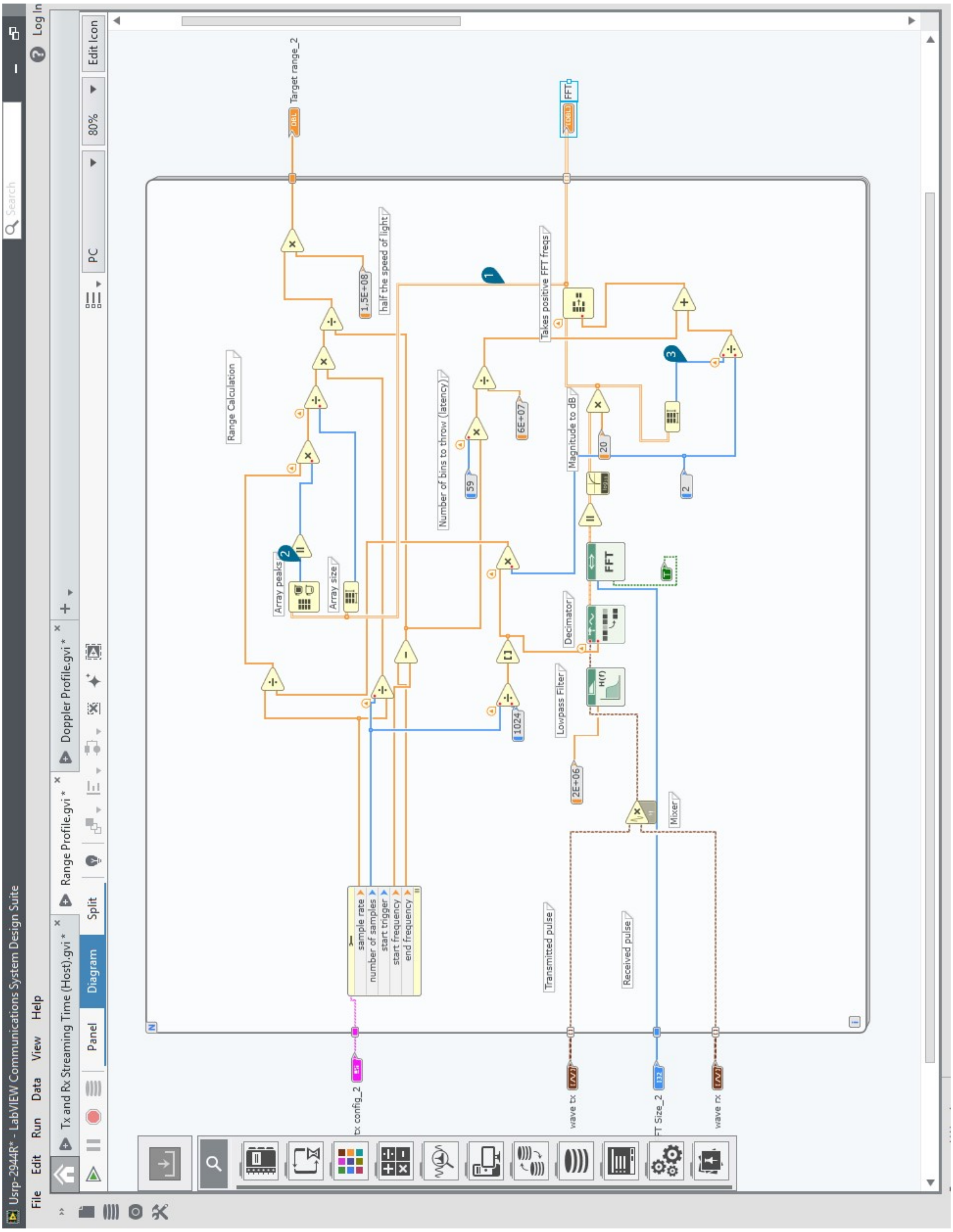


Figure 4.6: Block diagram of Range processing function

The block diagram takes as input the transmitted signal pulses with the attenuated and delayed Doppler shifted received echos. The process happens at the host PC in digital. Both the input signals are mixed together. The signal at the mixer will have frequency dependent on the echo delay and Doppler shift. After that, the signal passes through a lowpass filter at cutoff frequency of $2MHz$. In order to save on computation time and complexity we use a decimator at the output of the filter. The decimator has a factor of 32, meaning that it takes 1 sample every 32. The choice of cutoff frequency and decimation factor are discussed in Appendix A.7. Afterwards, we put a block taking the mixed, filtered and decimated signal as input and giving out a N-points FFT. The FFT and its advantages are discussed theoretically in Appendix A.3. The number N is equal to the number of samples of the decimated signal and in our test simulations it is equal to $N = 1024$.

The FFT function output is a graph of frequency bins to the amplitude and the beat frequencies of the mixer is at the frequency bins which correspond to an amplitude higher than a threshold. Of course each frequency bin corresponds to a specific time delay and therefore a target to radar distance. After some calculations we translated each frequency bin in a range bin. In Fig. 4.7 is shown the part in the block diagram which performs the calculation. So, the FFT procedure would give as a result the target detection and location which is what we were looking for. In Appendix A.1 there is a theoretical explanation of why this algorithm is used to perform target ranging and also some calculations on how to get the range profile from the FFT. In 5.1 we discuss the correctness of the algorithm.

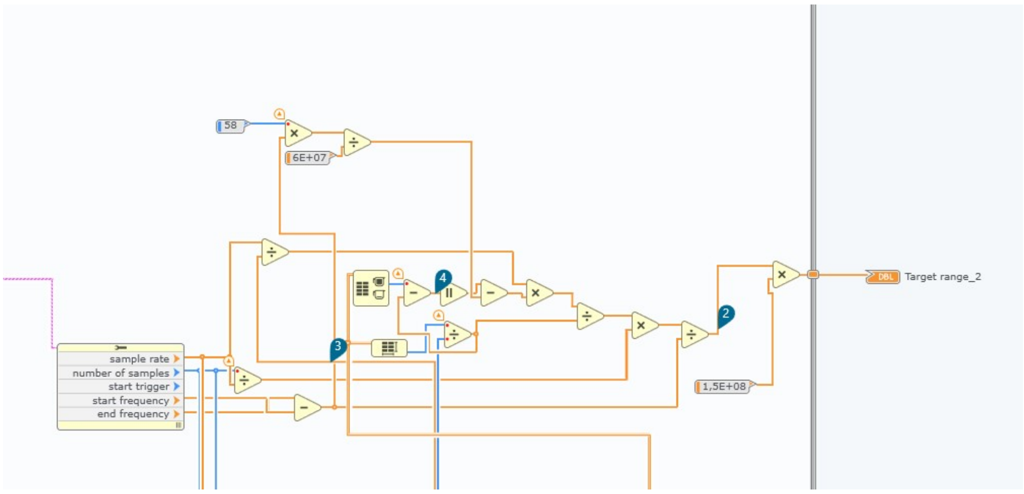


FIGURE 4.7: Block diagram used for range information retrieving from FFT

The input to the range calculation function in the above block diagram is the FFT vector. In order to calculate the range as defined in Eq. A.1 the function in the block diagram initially finds the frequency bin of the peaks and then throws away the number of frequency bins which come from hardware latency. The latency was calculated from different tests via loopback as discussed in Sec. 5.1. After that beat frequency f_b is calculated as in Eq. 4.6.

$$f_b = \frac{freqbin}{N/2} * \frac{f_s}{Dec - factor} \tag{4.6}$$

where f_s is the sampling frequency, N is the FFT number of points, $Dec-factor$ is the decimation factor. After calculating the beat frequency f_b according to the Eq. A.1 there are 3 other operations performed from the diagram block to come up with the range. It divides the beat frequency with the chirp bandwidth B, multiplies the result with the pulse width T_p and then multiplies with half the speed of light $1.510^8 m/s$. The outcome of this function is the range of targets detected from the FFT algorithm.

4.3 Range Resolution

Range resolution in radar system is a very critical performance metric. It defines the ability of a radar to detect two objects close to each other as two distinctive targets. The theoretical range resolution in systems transmitting 'chirp' waveforms is defined as in Eq. 4.7.

$$\Delta R = \frac{c}{2B} \tag{4.7}$$

where c is the speed of propagation and is assumed to be equal to the speed of light and B is the transmitted signal bandwidth. As mentioned, the USRP 2944-R device is connected to the host PC with a PCIe x4 Express connector that goes up to 800 MS/s and considering oversampling 4 times higher than the bandwidth for accuracy then we could have a real-time bandwith of up to 200 MHz. But on the other hand the USRP 2944-R device can go up to 160 MHz of real-time transmitting and receiving bandwidth for two channels. So, finally we can conclude that the best theoretical range resolution we can achieve using two channels with this system is $93,75$ cm which is appropriate for many radar applications. Although, it is inappropriate for those radar applications which are used to provide very detailed information on the scanned area. In Fig. 4.8 is shown the graph of FFT outcome. In the x-axis is represented each frequency bin of the N-point FFT whereas at the y-axis its amplitude expressed in dB.

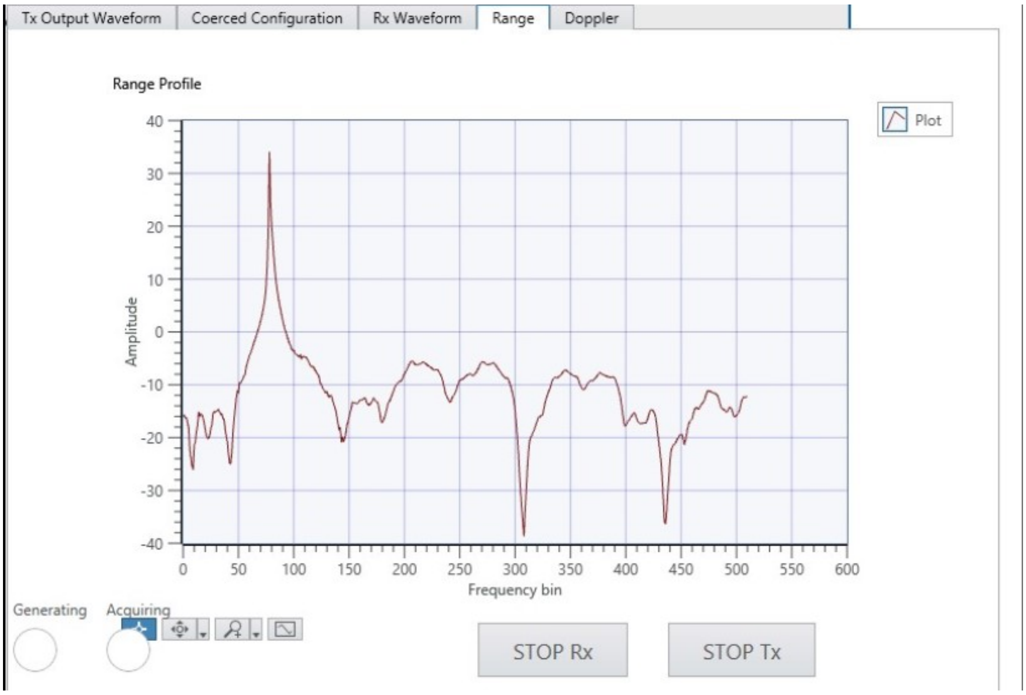


FIGURE 4.8: FFT graph of a simulated target

In this project, we used one channel of $80MHz$ and therefore the theoretical limit we could reach in terms of range resolution was $1.875m$. To have a better understanding of what that means in Fig. 4.8, each of the frequency bins are separated by $1,875m$. Therefore, if we suppose that there is another target at a distance less or equal to $1,875m$ from the target than it would not be tracked because the FFT would result in only one peak at that frequency bin.

4.4 Doppler Processing

Doppler processing of the target is required for moving targets. Moving target have a location which changes with time and which need to be updated to know excatly where the target is at different times. Therefore, they have a relative velocity with respect to the radar which is not equal to zero (stationary targets).

In order to determine exactly the target profile we need to have information on its velocity and it is where the Doppler processing comes into the game. In order to perform Doppler processing, we need a number of pulses that is usually more than one. Also, we need that at transmission each pulse is sent with a specific PRF which must be $\frac{PRF}{2} \geq f_D$ where f_D is the maximum Doppler frequency detectable.
In Fig. 4.9 is shown the Doppler processing method used in this project. Each matrix row corresponds to a range profile coming from the m^{th} pulse where $m \in [1, 2, ..M]$. Whereas, each matrix column gives information on how the amplitude of each range bin changes with the returns of the upcoming pulses. Therefore it gives information on how the magnitude changes with time. If we take the FFT to the n^{th} range bin, where $n \in [1, 2..N]$, then as a result we would get the Doppler frequency of the target situated at that specific range bin. Therefore, that is what we actually did. We found the indice of the first row of the matrix corresponding to the peak (range bin) and took the column of the matrix corresponding to the indice as shown in Fig. 4.9

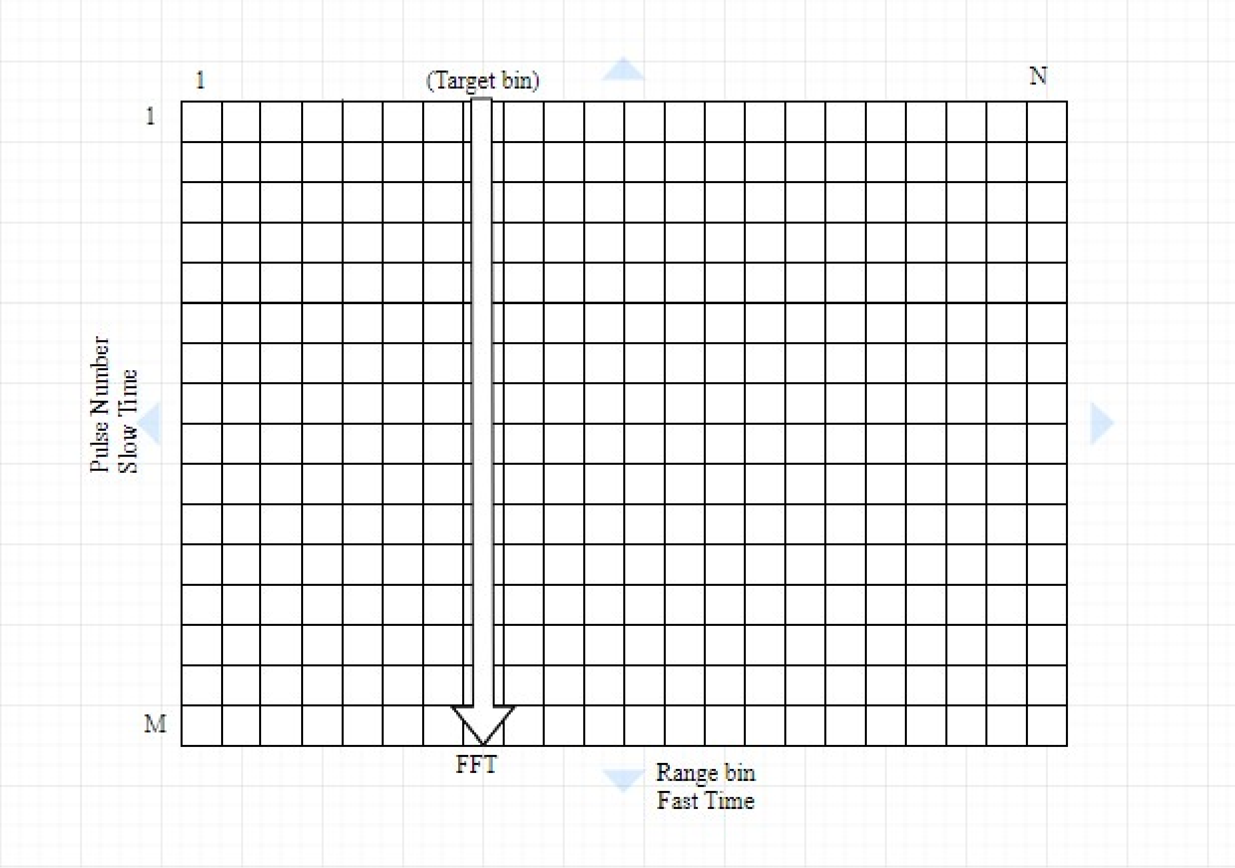


Figure 4.9: Pulsed Doppler signal processing

As discussed in Section 1.1 the Doppler frequency of a moving target indicates its relative velocity with respect to the radar system. The Doppler effect is the change in frequency or wavelength of a wave for an observer who is moving relative to the wave source [17]. This phenomenon is named after the austrian physicist Christian Doppler who described it in 1842. In pulsed radar, moving target detector (MTD) systems are employed for Doppler processing. The MTD is a N point FFT performed on a time sequence for each range bin. The N points for pulsed Doppler processing usually go up to 1000 points. The coherent processing interval (CPI) is equal to $T_{CPI} = N * T_{PRI}$. The CPI should be carefully chosen in order not to have the target move from the range bin at the end of it. Therefore, the choice of the number N is dependent on the PRI and the range bin width. A MTD system enables the creation of range and Doppler maps for targets and clutter. In Table 4.1 are defined the parameters used for taking data from non moving target simulated via loopback.

Parameter	**Value**
Transmitted Pulse Length (T_p)	$205\mu s$
Bandwidth (B)	$80MHz$
Carrier Frequency (f_c)	$2GHz$
Pulse Repetition Interval (PRI)	$30ms$
Number of Pulses processed (N)	114
Coherent Processing Interval (CPI)	$3.42s$

Table 4.1: Doppler Processing Parameters

As mentioned earlier, to calculate the Doppler frequency of the target we must perform a N point FFT to the column highlighted in Fig. 4.9. The FFT gives information on Doppler frequency. We implemented in LabVIEW Comms. the function showed in Fig. 4.10 to provide Doppler information on saved data by means of charts.

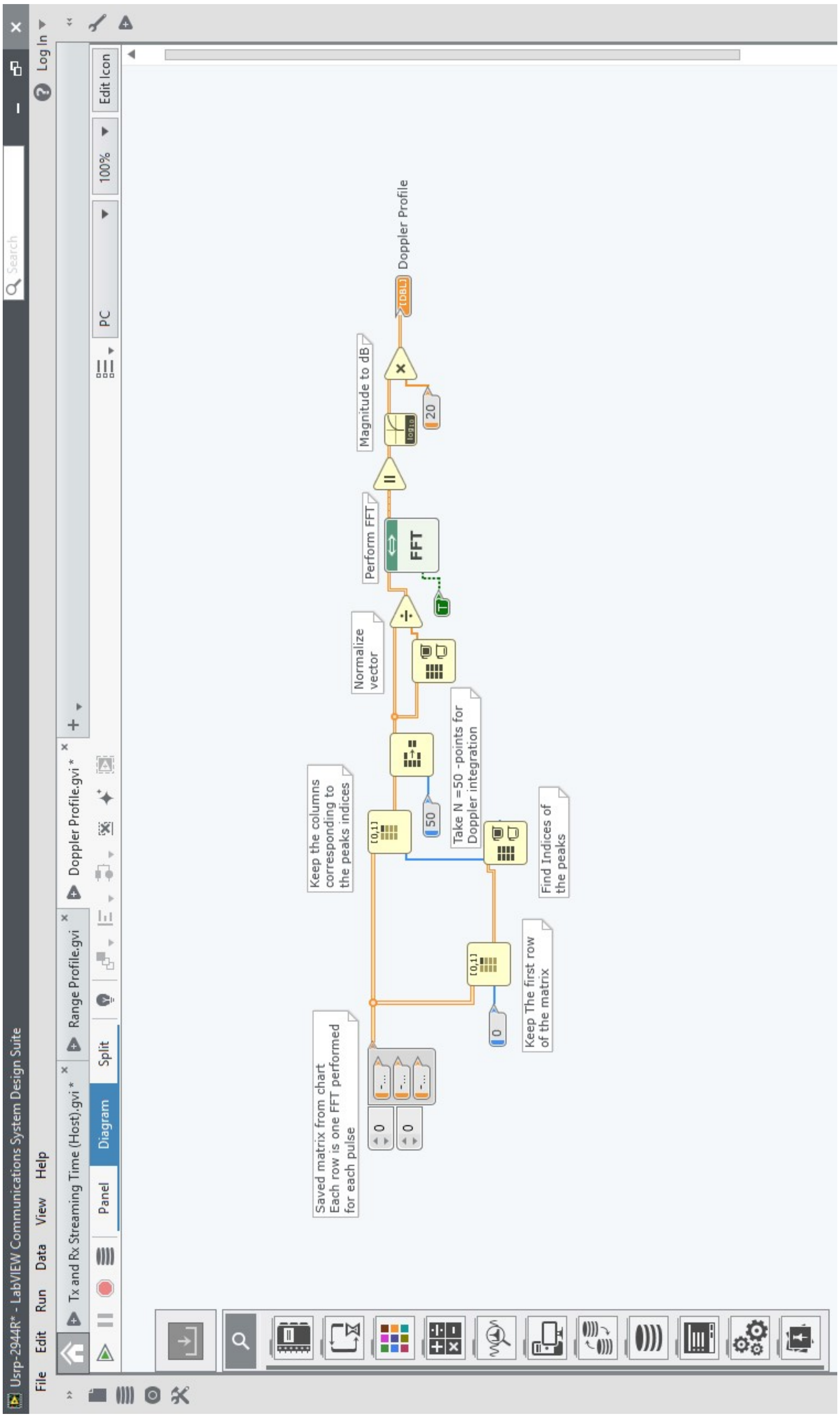

Usrp-2944R* - LabVIEW Communications System Design Suite
File Edit Run Data View Help
Log In
Search
Tx and Rx Streaming Time (Host).gvi*
Range Profile.gvi
Doppler Profile.gvi*
Panel
Diagram
Split
PC
100%
Edit Icon
Saved matrix from chart Each row is one FFT performed for each pulse
Keep The first row of the matrix
Keep the columns corresponding to the peaks indices
Find Indices of the peaks
Take N =50 -points for Doppler integration
50
0
[0,1]
Normalize vector
Perform FFT
FFT
Magnitude to dB
log10
20
[DBL]
Doppler Profile

Figure 4.10: Block diagram of Doppler processing function

From left to right, in the diagram there is shown the saved matrix of N range profiles (each range profile is a row compounding the matrix), two block functions which are able to take the row of interest, the FFT block, block which translate the FFT output amplitude in dB and in the most right the Doppler profile graph output.

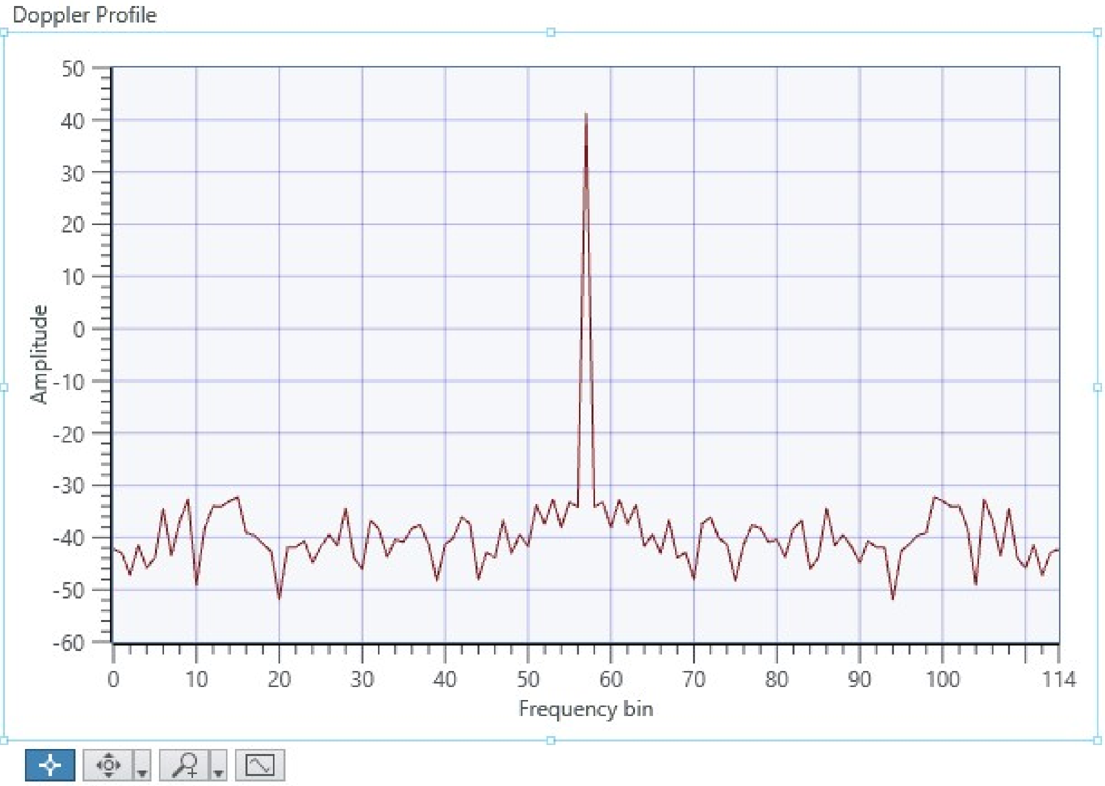


FIGURE 4.11: Doppler profile for non moving target simulation via loopback cable with length $40.75m$

The peak of the FFT result as could be seen from the Fig. 4.11 is at the middle of FFT points. This is due to the fact that we used complex FFT which would provide information even on negative Doppler frequencies (targets going away from radar). As expected the Doppler bin of a non moving target should have been at the middle of its graph which is the $0Hz$ Doppler frequency bin.

Chapter 5

Performance Results

5.1 Range Testing

In order to validate and correct the implemented signal processing procedure, we took many test measurements and runned the system with all the possible specifications which made the hardware work at its limits. When running these test simulations the very first thing we noticed by visualizing the receivel signal pulses was that they had a time delay in the range of $1\mu s$. The target range corresponding to $1\mu s$ is equal to $R = \frac{c\tau}{2} = \frac{3\times10^{8}m/s\times10^{-6}s}{2} = 150m$. Considering that the validation tests we did were done by creating loopback systems at the USRP device by using cables of length $0,8m$, $12m$ or $40,75m$, this delay reasonably was unfeasible. The loopback system create by connectd the transmitting chain TX1 in RF0 with the receiving chain RX2 in RF1 is showed in Fig. 5.1.

Therefore, the first question we asked ourselves was: why the received signal pulses had this unexpected delay? And the asnwer is somehow obvious. We needed to look at the flow path in which the signal passes. The two sources of this latency could be either coming from software or hardware. Considering that, the processing of generating and receiving the signals at software is done simultaneously at the same host PC gvi, the we could definitely say that this delay was mostly compounded from hardware latency and of course the time needed for the signal to travel through the cable.

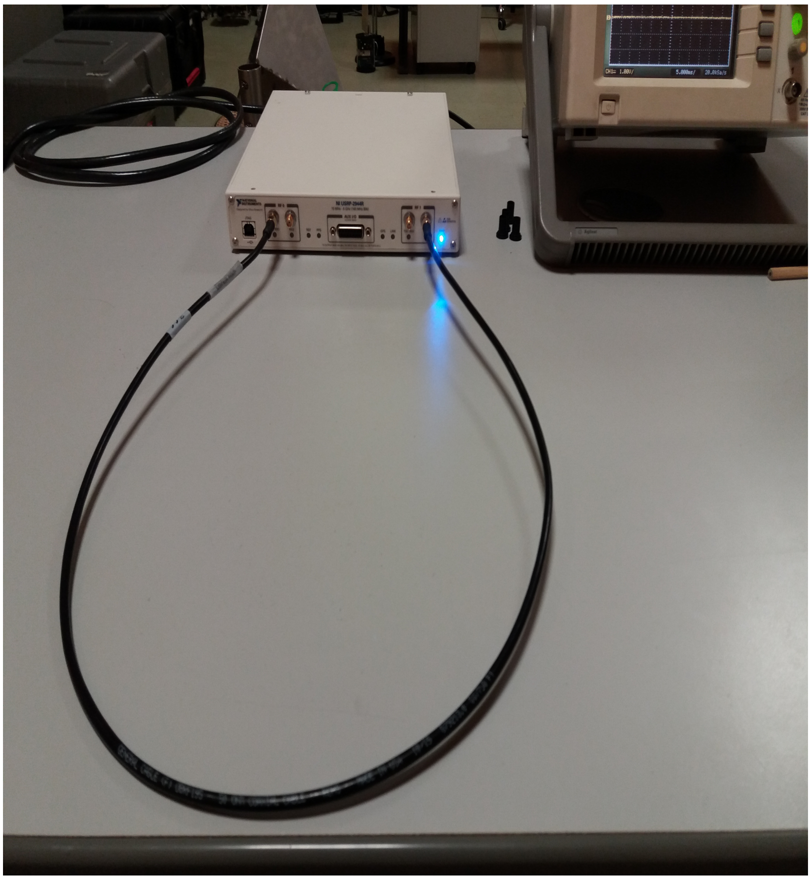


Figure 5.1: USRP 2944-R loopback system via 0.8m coaxial cable

Afterwards we asked if this hardware latency was constant or variable and most importantly how much is it and how to calculate it. To find out if the latency was variable or not we runned the system hundreds of times with the same cable but changing panel controlled specifications, such as carrier frequency, pulse length, bandwidth, sample rate, every time we runned the software. The delay of the received signal did not react to these changes. It neither reacted to the restarting of the USRP device. Therefore, we concluded that it is constant and due to the hardware processing.

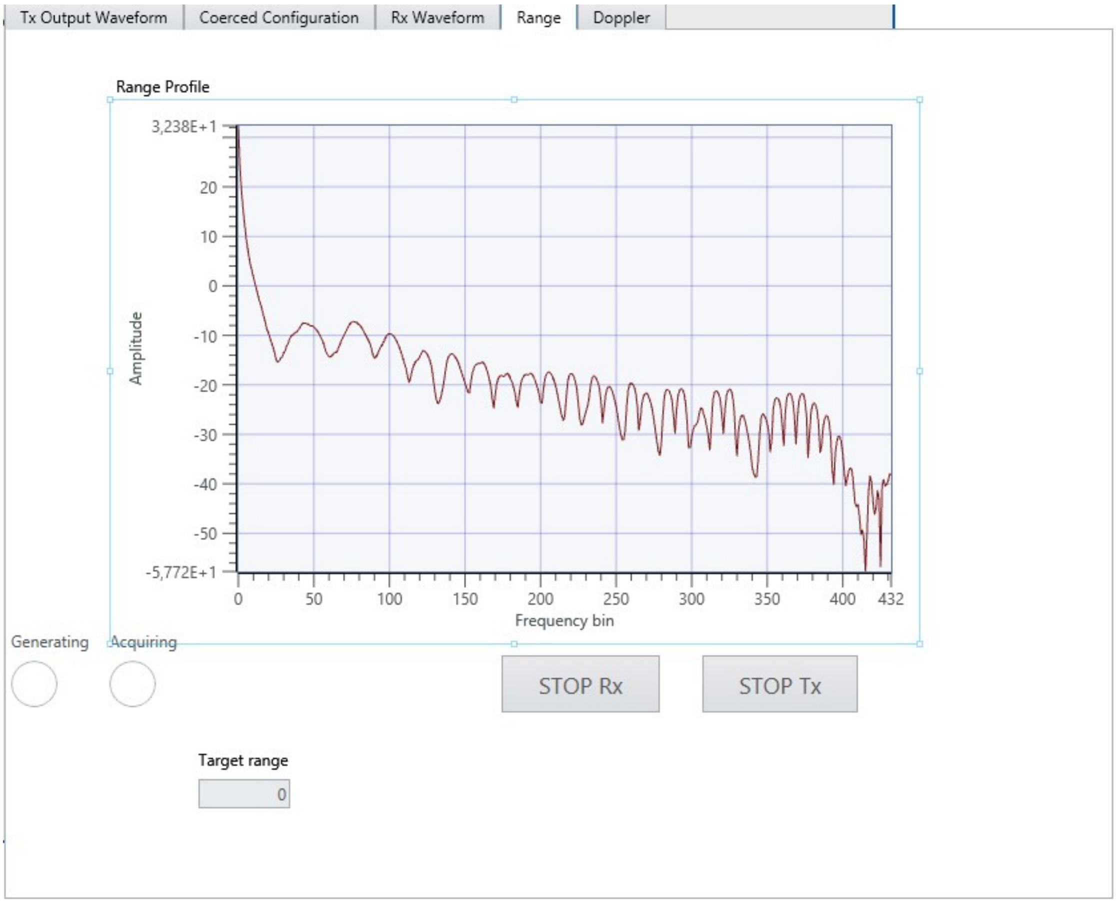
Then, we wanted to know how much it actually is and so we thought to calculate it again using the loopback system. This time, we kept panel controlled specifications constant but changed loopback cables. We visualized the received signal pulses when using the 3 different cables. The received signal delays changed proportional to the cable lengths as it was expected. Then, again by visualization of the received pulses with graph tool, we calculated the the difference in time $\tau_1 - \tau_2$ where τ_1 stands for time delay of received signal with loopback system formed by $40,75m$ cable and τ_2 is that of the $0.8m$ long cable, was $0.16\mu s$. That means that the received signal was delay by $0.16\mu s$ more to travel $39.95m$. In a vacuum the signal would require $\tau_{vac} = \frac{39.95m}{3\times10^8 m/s} =\approx 0.13\mu s$ to travel the same distance.

Therefore, assuming that both tested cables have the same VF attenuation compared to the speed of light, the signal travelled for $\approx 0.03\mu s$ more than it would in vacuum for the same distance. The assumed equal VF for the cables would be $VF = \frac{0.13\mu s}{0.16\mu s} = 0.8125$. So, the speed of the signals travelling in those copper cables is approximately 0.8125 times that of the speed of light in vacuum. Of course, it is an approximation and we can not be sure of that value. Anyway, the main conclusion we came up to from the above described tests was that the latency coming from hardware and 'corrupting' the received signal is constant and it is according to visualization and FFT measurements approximately $0.9833\mu s$. We will explain how we calculated this result also by using the range processing function discussed in Section 4.2. Both received signal graph visualizations and FFT method were complementary. The processing by FFT also provided a time delay of $0.16\mu s$ between delays of $40.75m$ and $0.8m$ cables.

The FFT computation function calculated and overall $1.1454\mu s$ time delay for the $40.75\mu s$. That means that the latency delay due to hardware $\tau_{hardware} = 1.1454\mu s - 0.16\mu s - 0.0021\mu s = 0.9833\mu s$ where 0.0021 is the time needed to travel $0.8m$ cable with $VF \approx 0.8125$.

Why is this hardware latency component important? Obviously, because we need to account for it at the FFT processing result. So when the FFT computes its result, in our calculations we need to throw away the FFT bins corresponding to the hardware delay.

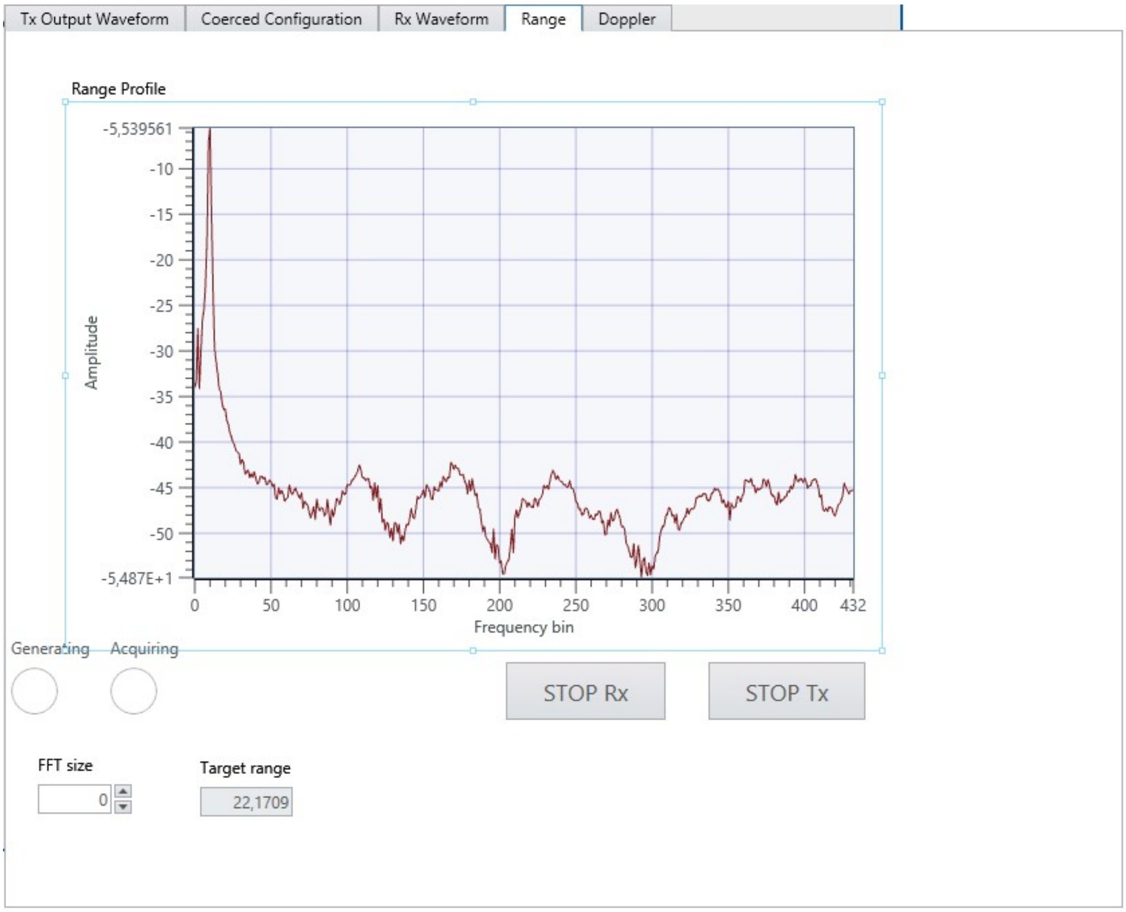
Getting back to the range profiling algorithm verification and validation topic, we want to emphasize that is was done by means of loopback systems similar to the one demonstrated in Fig. 5.1. The loopback system is implemented to verify the signal processing range estimation method computed from FFT because it takes off the unknown parameter uncertainty. We know the lengths of cables and we we think of a target positioned at the middle of those cables then we have simulated in practice a real radar target scenario. This affirmation is also supported from the true fact that the cables input at the received signal AWGN and attenuation. In Fig. 5.2 and Fig. 5.3 are shown the FFT graphs for both $0.8m$ and $40.75m$ cables respectively.



Target range

0

FIGURE 5.2: Range profile for target simulation via loopback cable with length 0.8m



FFT size 0

Target range 22,1709

FIGURE 5.3: Range profile for target simulation via loopback cable with length $40.75m$

Furthermore, the indicator target range calculated as discussed in Section 4.2 determines the targets ranges (in case there are more than one it displays a column of ranges). The resulting range of FFT for $0.8m$ cable was $0m$ whereas for $40.75m$ cable was $22.1709m$ (almost half the length of the cable). We need to point out that the transmitted pulse bandwidth was $80MHz$ resulting in range resolution $\Delta R = 1.875m$. Therefore the FFT results were near to what was the expectation. The reflected the simulated target distance. We run the simulations for all user-controlled specifications in a number of times. The only parameter which would change the FFT target's range estimation was the chirp bandwidth. Certainly, it is due to the fact that is defines range resolution as stated in Eq. 4.7.

What was the error of this estimation and the accuracy? The absolute, relative and percent errors are given in the equations as follows. For some estimated range values R_{approx} and the true value R the following relations stand:

$$\epsilon = |R - R_{approx}| \tag{5.1}$$

$$\eta = \frac{\epsilon}{R} \tag{5.2}$$

$$\delta = 100\% \times \eta \tag{5.3}$$

where ϵ is the absoulte error, η is the relative error and δ is the percent error. For our test simulation with $40.75m$ cable described in the previous paragraphs this values stand as follows:

- $\epsilon = 1.7959m$
- $\eta = 0.088$
- $\delta = 8.8\%$

In general in medium the estimator will have a percent error equal to :

$$\delta_{est} = \frac{\Delta R}{R_{max}/2} \times 100\% \tag{5.4}$$

5.2 Doppler Testing

As discussed in Section 4.4, in this radar prototype implementation we performed also target's velocity profiling by processing their Doppler. During the testing of the implemented Doppler processing method we noticed a very crucial limitation from the software in use. The software was found to have an upper limit PRF, thus limiting also the maximum velocity detectable. After performing a number of tests, we concluded that the minimum PRI of pulses that the radar prototype implemented was equal to $30ms$. Therefore resulting in a maximum $PRF_{max} \approx 33Hz$. As stated also in Section 4.4, the maximum theoretical detectable Doppler frequency from such a pulsed radar would be $f_D^{max} = \frac{PRF}{2} = 16.5Hz$. In practice this number is even lower.

Apart from the non moving target simulation and its Doppler profile showed in Fig. 4.11, in our simulations we devoloped a program which would simulate a moving target. Therefore, the simulated target would have a specific range bin and also a specific Doppler bin at every moment of radar processing. We choosed to simulate a target with a constant Doppler frequency and so a constant relative velocity with respect to the radar. The Doppler simulation consisted in mixing the column at the input of the second FFT. In Fig. 5.4 is shown the LabVIEW implementation of the simulation.

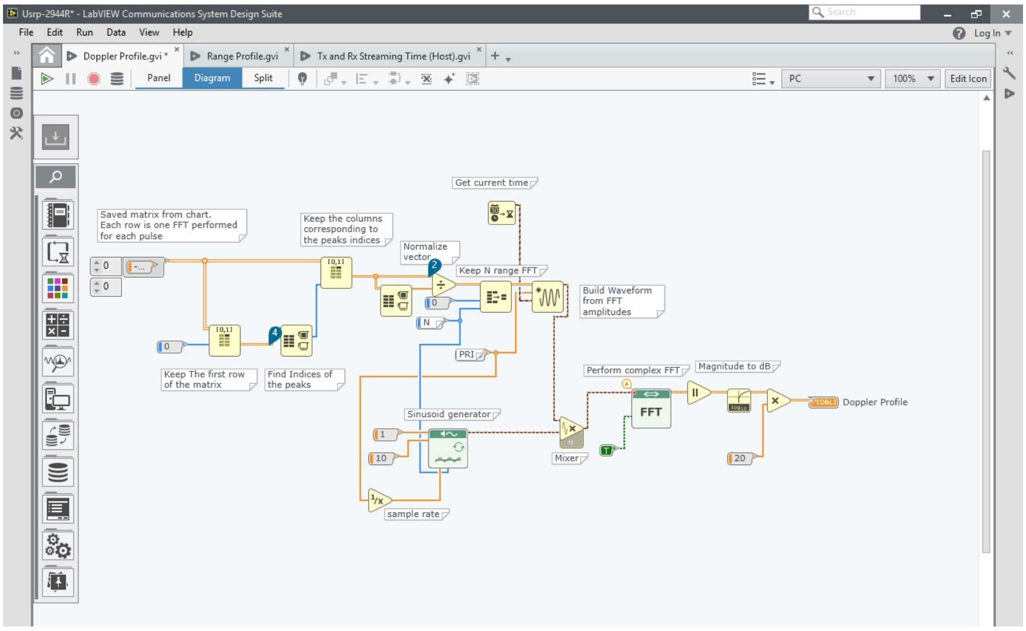

Usrp-2944R* - LabVIEW Communications System Design Suite
Search
File Edit Run Data View Help
Log In
Doppler Profile.gvi *
Range Profile.gvi
Tx and Rx Streaming Time (Host).gvi
Panel
Diagram
Split
PC
100%
Edit Icon
Get current time
Saved matrix from chart. Each row is one FFT performed for each pulse
Keep the columns corresponding to the peaks indices
Normalize vector
Keep N range FFT
Build Waveform from FFT amplitudes
0
0
10,11
2
4
0
0
N
PRI
Keep The first row of the matrix
Find Indices of the peaks
Perform complex FFT
Magnitude to dB
FFT
Doppler Profile
Sinusoid generator
1
10
Mixer
20
sample rate

FIGURE 5.4: LabVIEW block diagram for Doppler simulation and processing

The simulation consisted in Doppler shifting the signal at the input of the FFT by mixing it with a sinusoid of $10Hz$. In Fig. 5.5 is shown the resulting FFT.

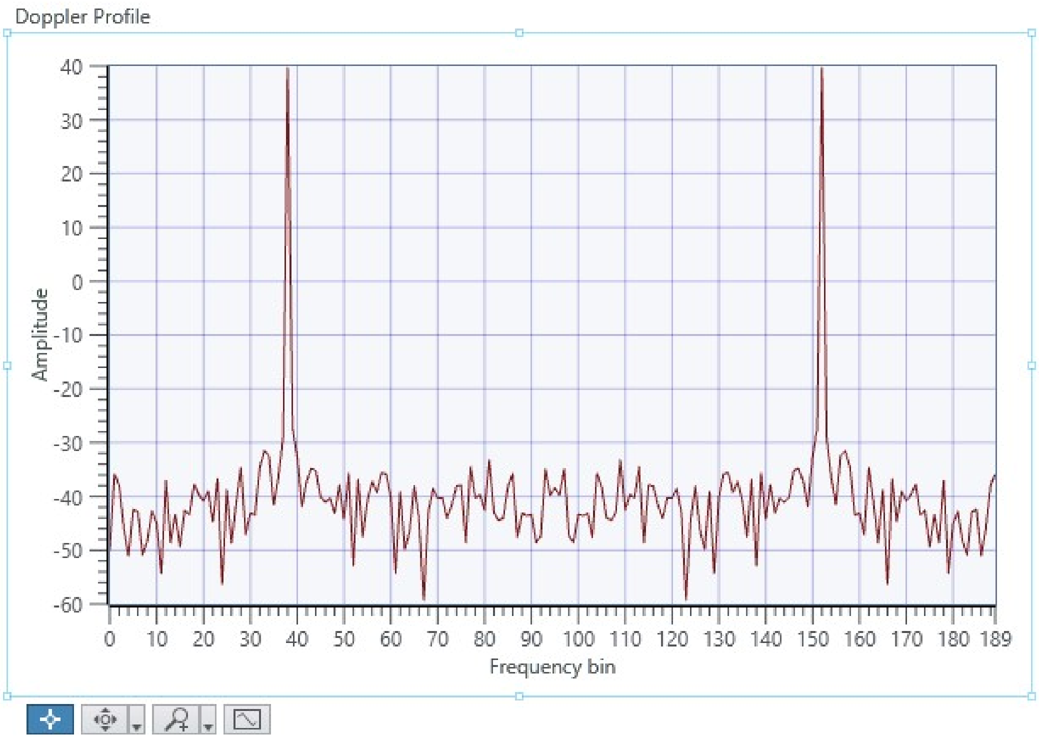


FIGURE 5.5: Doppler profile for a simulated target at $f_D = 10Hz$ with pulses of $PRI = 30ms$

From the above Fig. 5.5 it can be noticed the double peaks resulted from the complex FFT. The two peaks are situated at f_D equal to $-10Hz$ and $10Hz$. Due to the inability of the LabVIEW Comms. software to rearrange the graph axis we let them as in the figure but here we provide an explanation. If we think of the x axis of the FFT going from $-f_D^{max}$ to f_D^{max} where $f_D^{max} \approx 16.5Hz$, then the resulting Doppler profile is correct. The two peaks are situated at bin 38 and at bin 152 whereas the $0Hz$ frequency components is located at the center bin number 95. Therefore both peaks are 57 bins away from the center and their corresponding frequency components would be defined as in Eq. 5.5

$$f_{peaks} = \frac{binnumber - 95}{95} \times f_D^{max} \tag{5.5}$$

In this test case, the resulting frequency components would be as expected at $-10Hz$ and at $10Hz$. Anyway in order to be sure of the correctness of our Doppler processing method implemented, we took another Doppler simulation scenario. The resulting FFT for this simulation is shown in Fig. 5.6.

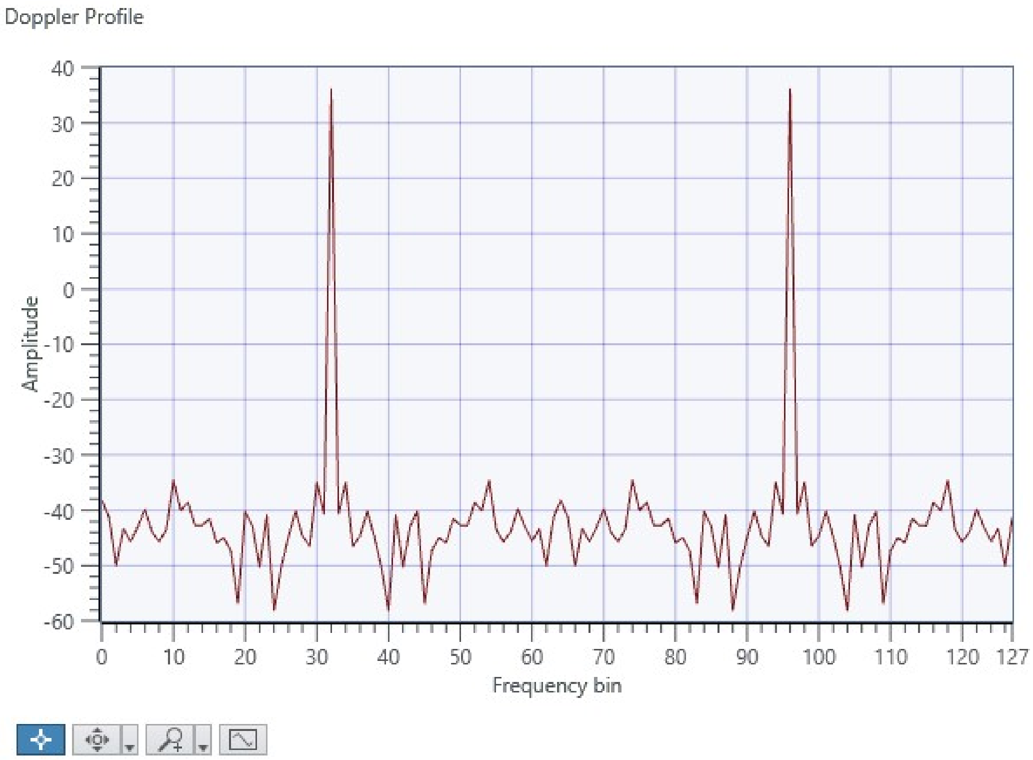


FIGURE 5.6: Doppler profile for a simulated target at $f_D = 5Hz$ with pulses of $PRI = 50ms$

This time we changed some of the parameters. We used specifications as in Table 5.1 for this test simulation.

Parameter	**Value**
Transmitted Pulse Length (T_p)	$205\mu s$
Bandwidth (B)	$80MHz$
Carrier Frequency (f_c)	$2GHz$
Pulse Repetition Interval (PRI)	$50ms$
Number of Pulses processed (N)	128
Coherent Processing Interval (CPI)	$4.12s$
Simulated Doppler Frequency (f_D)	$5Hz$

Table 5.1: Doppler Processing Parameters

The two peaks shown in Fig. 5.6, after following the same reasoning as above for interpreting the FFT result and doing calculations using Eq. 5.5, are frequency components at $-5Hz$ and $5Hz$. This is what we expected and therefore we can conclude that also the Doppler processing method provided correct results.

Chapter 6

Conclusions

6.1 Project Outcome

In conclusion, in this chapter is presented a final draft with facts on what this thesis project concluded and what could be some of the future works on software defined radars.

This thesis work concluded four main points. The first and most obvious conclusion is that a full radar system can be implemented on the NI USRP 2944-R software defined radio device using LabVIEW Communications System Design Suite 2.0, despite its pros and cons discussed. The prototype radar system implemented showed some very good results on profile ranging of the targets. The range resolution which determines the minimum distance between two different targets so that the radar could detect them both, was in line with the theoretical minimum range resolution bounded from hardware specifications. In this radar prototype we developed, the range resolution was $1.875m$ which is good enough for many radar applications. On the other hand, the system demonstrated some limitations in terms of velocity profiling. The processing speed of the used software was the key factor limiting the Doppler estimation algorithm. The software resulted in a $30ms$ PRI threshold, and therefore enabling the estimation of a maximum Doppler frequency $f_D = 15Hz$. Of course dependent on the carrier frequency this maximum Doppler frequency results in a maximum target's velocity detectable which is too low for many radar applications.

Secondly, we concluded that the USRP 2944-R software defined radio in use has a hardware processing latency due to all its physical connections and processing. The latency was demonstrated to be at the order of $1\mu s$, exactly $0.9833\mu s$. By performing many tests using different parameters we concluded that the latency was constant, or if not constant it varied by an unconsiderable amount.

Third, all complex radar processing methods could be implemented via LabVIEW Communications System Design 2.0 software. The main advantage of using this software is that it provides a very user friendly panel for setting up system parameters and an easily understandable block diagram flow programming. Also, the software has many built in functions which could perform complex tasks. On the other hand, when dealing with complex data manipulators such as for example charts used for Doppler processing the software showed slow processing speed. In real-time applications the software would limit the workflow.

Lastly, we concluded that both hardware and software components of the radar prototype somehow influenced the system. Theoretical performance metrics such as range resolution were bounded from USRP 2944-R device characteristic of real-time bandwidth. The best performance was achieved when running the device at its limits. Whereas, Doppler frequency estimation was limited from the pulses PRF. The PRF on its side was limited by the processing speed of the software in use at around $30Hz$.

6.2 Future Work

In this section, we continue by listing some of the future works in the software defined radar topic.

Software defined radios could be of great interest in a number of radar research topics from radar waveform design to radar signal processing, MIMO radars and radar prototyping. Due to their reconfiguration property and the low costs, SDRs could be used to investigate on optimizing critical radar performance metrics such as interference mitigation, estimation accuracy etc., in a set of signals which could be predefined or not. By predefined, we mean that the set from which to choose the optimal signal, is compounded from well known radar signals such as LFM, Costas signals, Frank codes, Barker codes etc. Whereas non-predefined are pseudo random signals from which the software implemented optimizer would pick one in the set.

Furthermore, considering that USRP devices have the property to be used in serial multi mode one suggestion for further investigations would be to carry out the implementation of a system compounded from many of them. This system could increase the real time bandwidth by a factor equal to the number of devices used. Thus, resulting in a better performing system. Also, the implementation of such a system could simulate MIMO radars scenario which apart from ranging and Doppler profiling, could calculate the azimuth of targets.

Finally, another field of investigation using SDR devices could be cognitive radar. They seem to be the next generation of radars with adaptivity and diversity properties. Considering the ability to be reconfigured and its definition from software, SD radars could be used for research on intelligent radar prototype implementation and characterization.

Appendix A

Appendix

A.1 Range Measurement of Stationary Targets

If one considers to find the location of a stationary object using frequency modulated signals for transmission then the following method could be applied. If we assume that the transmitted signal is a ramp sent periodically and the received signal is also a ramp which is time delayed proportional to target-radar distance then we can assume the scenario shown in Fig. A.1

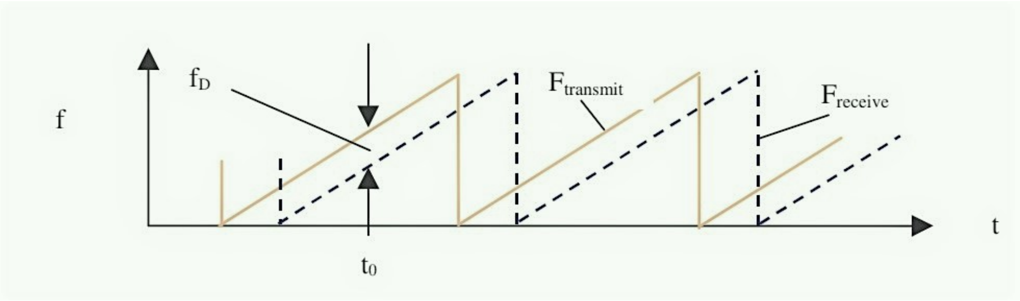


FIGURE A.1: Transmitted and received signal representation for FMCW radar

If we compare the two signals at specific time t_0 we can look that they differ in frequency by f_b. This difference is called beat frequency. If we mix the transmitted and received signal then we get a signal which has a constant frequency equal to the beat frequency. In order to get information about the beat frequency of the mixed signal we perform FFT as explained in Section 4.2 and therefore we get the desired range information using the following relation which stands:

$$R = \frac{c}{2} T_p \frac{f_b}{B} \tag{A.1}$$

where R is the distance of the reflecting object, c is the speed of propagation, T_p is the pulse width in time and B is the chirp bandwidth.

A.2 Procedure for Range Sidelobe Level Computation in Matlab

Range sidelobe level is a critical radar performance metric which tells how good a waveform in range resolution of different targets. It is worthy to know what is the sidelobe level at the output of the FFT for each type of waveform. Therefore, we implemented in Matlab a function which gives the highest range sidelobe level. The technique for deriving such result is performed as follows:

Listing A.1: Matlab Calculation of the Sidelobe Level

```
function [ SLLevel,Location ] = SidelobeLevel( x )

x = mag2db(x);
[peaks,locs] = findpeaks(x);
peaks_s = sort(peaks,'descend');
peaks_n = peaks_s(2:length(peaks_s));

SLLevel = max(peaks_n);
Location = find(x == SLL);
```

In the above script the input x to the function may be either the zero doppler normalized ambiguity function or the range profile vector at the FFT output. The outputs SLLevel and Location are respectively the resulting sidelobe level and its indice in the correlation function, used to come up with the specific time delay (therefore the range) of the highest sidelobe.

A.3 Fast Fourier Transform

The Fast Fourier Transform (FFT) is a very efficient way to calculate the DFT of a vector signal. If the vector has the length equal to n, then the complexity of the FFT algorithm is $O(n \log n)$ compared to $O(n^2)$ which is the DFT complexity, it is more efficient way to compute the DFT. The FFT algorithm samples the input vector signal over a period of time and divides it into its frequency components. The FFT computes the DFT and calculates the same result as evaluating the DFT definition. Let $x_0, x_1, ..., x_{N-1}$ be complex numbers. Then the DFT is defined by the formula in Eq. A.2.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \qquad k = 0, 1, ..., N-1 \tag{A.2}$$

Even though one form of FFT was also discussed by Gauss in the 19th century, the modern generic FFT is accredited to Cooley and Tukey which published the method in 1965. The Cooley-Tukey is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N1N2$ into many smaller DFTs of sizes N1 and N2, along with O(N) multiplications by complex roots of unity traditionally called twiddle factors [18]. Considering that the programming of the USRP project is computationally heavy, the usage of FFT which provides the fastest way to perform the DFT is very advantageous for us. Therefore, as discussed in radar signal processing for range and Doppler profiling the FFT algorithm is one of the most used mainly for its efficiency.

A.4 Device Power

The USRP 2944-R device according to its specifications manual has a maximum output power $P_{out} = 20dBm$ and maximum input power $P_{in} = -15dBm$ [19]. The following relation in Eq. A.3 stands to convert the power expressed in dBm to power in mW.

$$P_{mW} = 1mW10^{\frac{P_{dBm}}{10}} \tag{A.3}$$

The conversion Table is true for $50Ohm$ impedance systems.

Power (dBm)	**Power (mW)**	**Voltage (rms)**	**Voltage (peak-to-peak)**
20	100	$2.236V$	$6.324V$
10	10	$0.707V$	$2V$
0	1	$0.2236V$	0.6323
-3	0.501	$0.1583V$	$0.4476V$
-15	0.031	0.0397 V	$0.1124V$

TABLE A.1: Power to Voltage conversion table for 50 Ohm impedance systems

The table can serve as a reference to find out which was the output power or voltage at the test measurements.

A.5 Hilbert Transform

The Hilbert transform is used in mathematics or signal processing to perform a linear operation by taking a function of one real variable $u(t)$ and producing another function of a real variable $Hu(t)$. It is defined from Eq. A.4.

$$H[u(t)] = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{u(\tau)}{t-\tau} d\tau \qquad \text{(A.4)}$$

In Table A.2 are shown some of the most used Hilbert transforms

Signal $u(t)$	**Hilbert Transform** $H[u(t)]$
$\cos(\omega t)$	$sgn(\omega)\sin(\omega t)$
$\sin(\omega t)$	$-sgn(\omega)\cos(\omega t)$
$e^{i\omega t}$	$-isgn(\omega)e^{i\omega t}$
$\frac{1}{t^2+1}$	$\frac{t}{t^2+1}$
$\frac{sin(t)}{t}$	$\frac{1-\cos(t)}{t}$
δt	$\frac{1}{\pi t}$

Table A.2: Hilbert Transforms

As stated at the above table the following relation exists for the Hilbert transform:
If $x(t) = cos(\omega t)$ then $Hx(t) = sin(\omega t)$
In order to implement this transform, in LabVIEW Comms. exists a built-in function able to do that with good accuracy.

A.6 Transmitted Pulses' Parameters

The waveform generating subvi discussed in Sec. 3.4 is runned when the main host gvi calls it. All the inputs to the signal generating function are available at the main host front panel and are easily configurated from the software user according to his needs bounded from hardware limitations. Parameters of the pulses we used in our test simulations are specified in Table A.3.

Parameter	**Value**
Carrier Frequency (f_c)	$2GHz$
Sample Rate (f_s)	$160MHz$
Number of Samples (N_s)	32756
Pulse Width T_p	$204,725\mu s$
Start Frequency	$0Hz$
End Frequency	$80MHz$
Bandwidth (B)	$80MHz$
Output Power	$10dBm$

Table A.3: Transmitted Pulse Parameters

The choice of the parameters came as result of hardware limitations and they were chosen at hardware threshold limit. For example the choice of LFM pulse instantaneous bandwidth came as it is limited from the device which when working with one RF channel could go up to $80MHz$. It is shown in Fig. 2.3 that the RF module has a lowpass filter with cutoff frequency at $80MHz$.

A.7 Decimator

The decimator used to implement the range processing discussed in Section 4.2 had a specific decimation factor. In this section, we will discuss the calculation of that factor. But first, we want to emphasize the importance of the decimator in the overall signal processing part at the host. The importance stays behind the fact that the decimator factor also defines the number of FFT points. Therefore, the processing computational complexity is divided by the decimator factor resulting in a faster and wiser signal processing implementation. To calculate the decimation factor we supposed we wanted to detect a target at a maximum distance of $2km$. The time of flight for a target at $2km$ is equal to $\tau = \frac{2R}{c} = \frac{4000m}{310^8 m/s} \approx 13,3\mu s$. Therefore the sinusoid after the decimation must have a maximum bandwidth of up to $B_{dec} = \frac{B}{T_p}\tau = \frac{80MHz}{204{,}725\mu s}13,3\mu s = 5,2MHz$. For conventionality, we took this bandwidth to be $5MHz$. In conclusion the decimation factor used in the processing is equal to $D_f = \frac{f_s}{B_dec} = \frac{160MHz}{5MHz} = 32$.

Bibliography

[1] *Monostatic radar vs Bistatic radar-Difference between Monostatic radar and Bistatic radar*. 2012. URL: `http://www.rfwireless-world.com/Terminology/Monostatic-radar-vs-Bistatic-radar.html`.

[2] *Comparison between Pulsed Radar and CW Radar*. 2011. URL: `http://www.indiastudychannel.com/resources/135325-Comparison-between-Pulsed-Radar-and-CW-Radar.aspx`.

[3] C. Prathyusha and et al.. "Implementation of a low cost synthetic aperture radar using software defined radar". In: *booktitle*. Internation Conference on Computing Communication and Networking Technologies (ICCCNT). Ed. by editor. 2010, pp. 1–7.

[4] J. Marimuthu, K. S. Bialkowski, and A. M. Abbosh. "Software-Defined Radar for Medical Imaging". In: *IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES* Vol. 64.No. 2 (2016), pp. 643–652.

[5] S. Costanzo et al. *POTENTIALITIES OF USRP-BASED SOFTWARE DEFINED RADAR SYSTEMS*. In: ed. by EMW Publishing. Progress In Electromagnetics Research B. 2013.

[6] A. Prabaswara, A. Munir, and A. B. Suksmono. "GNU Radio based software-defined FMCW radar for weather surveillance application". In: *Int. Conf. Telecommun. Syst. Services Applic.* (2011), pp. 227–230.

[7] H. Zhang, L. Li, and K. Wu. "24GHz Software-Defined Radar System for Automotive Applications". In: *Proc. 10th Eur. Conf. Wireless Tech.* (2007), pp. 138–141.

[8] T. W. Mathumo, G. T. Swart, and R. W. Focke. "Implementation of a GNU Radio and Python FMCW Radar Toolkit". In: *booktitle*. IEEE Africon Proceedings. Ed. by editor. 2017.

[9] editor, ed. *FMCW Software Defined Radar for Range and Speed Estimation*. First International Undergraduate Research Conference. (Cairo).

[10] A. Capria et al. "Ship detection with DVB-T software defined passive radar". In: *booktitle*. IEEE Gold Remote Sensing Conference. Ed. by editor. 2010.

[11] N.N.S.S.R.K. Prasad et al. "Improvement in target detection performance of pulse coded Doppler radar based on multicarrier modulation with fast Fourier transform (FFT)". In: *IEEE Proc.-Radar, Sonar Navigat.* 151 (2004), pp. 11–17.

[12] S. Baskar and E. Ertin. "A software defined radar platform for waveform adaptive MIMO radar research". In: *Proc. IEEE Radar Conf.* (2015), pp. 1590–1594.

[13] Y.L. Sit et al. "Real-Time 2D+velocity Localization Measurement of a Simultaneous-Transmit OFDM MIMO Radar using Software Defined Radios". In: *Proc. 13th European Radar Conf.* (2016), pp. 21–24.

[14] *NI announces two new USRP SDRs with widest frequency ranges, highest bandwidth, best RF performance in the family.* 2017. URL: `https://forums.xilinx.com/t5/Xcell-Daily-Blog/NI-announces-two-new-USRP-SDRs-with-widest-frequency-ranges/ba-p/750786`.

[15] *USRP-2944 Block Diagram.* URL: `http://www.ni.com/documentation/en/software-defined-radio-reconfigurable-device/latest/usrp-2944/block-diagram/`.

[16] M. Skolnik. *Radar Handbook.* 2008. ISBN: 978-0-07-148547-0.

[17] N. Giordano. "College Physics: Reasoning and Relationships". In: *Cengage Learning* (2009), pp. 421–424.

[18] W. Morven and G. Sande. "Fast Fourier transforms—for fun and profit". In: *Proceedings of the AFIPS* (1966), pp. 563–578.

[19] *Specifications USRP 2944-R.* 2017. URL: `http://www.ni.com/pdf/manuals/375724b.pdf`.