FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Electrotécnica e de Computadores
Mestrado integrado em Engenharia Electrotécnica e de Computadores

Implementação de um sistema de RADAR numa plataforma de rádio definido por software

Frederico André Farinha Alves Simões de Carvalho

Júri:

Presidente: Prof. Doutor Henrique José Almeida da Silva

Vogal: Prof. Doutor Luís Alberto da Silva Cruz

Orientadora: Prof. Doutora Maria do Carmo Raposo de Medeiros

Outubro de 2018

· U C ·
FCTUC FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Electrotécnica e de Computadores
Mestrado integrado em Engenharia Electrotécnica e de Computadores

Implementation of a RADAR system on a software defined radio platform

Frederico André Farinha Alves Simões de Carvalho

Jury:

President: Professor Henrique José Almeida da Silva, PhD

Member: Professor Luís Alberto da Silva Cruz, PhD

Supervisor: Professor Maria do Carmo Raposo de Medeiros, PhD

October 2018

AGRADECIMENTOS

Agradeço em primeiro lugar à professora Maria do Carmo, orientadora da minha dissertação de mestrado. Pela oportunidade que me deu em trabalhar num projeto do IT, pela confiança que depositou nas minhas capacidades e pela oportunidade de no seio do laboratório do IT aprofundar e relacionar o que aprendi ao longo do curso. Agradeço-lhe a capacidade analítica que me fez adquirir, e a forma de como se encara uma investigação.

Agradeço à minha namorada Cláudia, pela motivação, amizade, paciência e ajuda que me fizeram encarar a vida de forma diferente, atingir e definir novos objetivos. Agradeço também ao Sr. Augusto e à Anabela pela amizade, por me acompanharem e apoiarem no meu percurso.

Na amizade, agradeço ao meu grupo de convívio e estudo, ao André, ao Pedro e ao Paulo, pelo apoio e pelo contributo que sempre me deram, pelos convívios e pela amizade constante. Agradeço ao Gonçalo e ao Marco pelo companheirismo no laboratório.

Por fim, agradeço também aos meus companheiros de residência universitária, ao meu irmão Luís, ao José, Cristóvão, Luís Henriques, Gabriel, Jorge e ao Elton, que ao longo dos anos viveram comigo e mais me marcaram, toda a amizade, companheirismo e lições.

RESUMO

A crescente necessidade de reconfiguração, readaptação e customização dos sistemas de rádio ao contexto de múltiplo propósito, como por exemplo no ramo automóvel e diagnóstico e tratamento médico, leva a que os tradicionais sistemas de rádio implementados em arquiteturas de hardware muito específicas e dedicadas, com reduzido uso e poder de software e elevado custo, se torne impensável. Como solução, o processamento complexo implementado pelo hardware passa para o domínio do software. Nestes sistemas é usada eletrónica de rádio frequência integrada em circuitos sólidos, circuitos integrados, altamente eficientes, tal como as Field Programmable Gate Array (FPGA).

No mesmo sentido, nesta dissertação pretende-se estudar e avaliar as capacidades do Rádio Definido por Software (Software Defined Radio SDR) para implementar um sistema de radar (RAdio Detection And Ranging RADAR), reduzindo os custos de implementação e permitindo adaptar o sistema a vários contextos, fazendo uso de Plataformas Definidas por Software (Software Defined Platform SDP).

São identificadas as principais limitações de sincronismo das SDPs que normalmente são utilizadas na demonstração de sistemas de comunicação e é demonstrado que os mecanismos de sincronismo básicos não permitem a implementação de um sistema radar. Foi proposta e validada experimentalmente uma estratégia inovadora que baseada em conceitos de sistemas de comunicação Multiple-Input Multiple-Output (MIMO), permite a implementação de um sistema radar Frequency Modulated Continuous Wave (FMCW) numa plataforma genérica SDP.

Palavras-Chave:

RADAR, Rádio Definido por Software (SDR), Plataformas Definidas por Software (SDP).

ABSTRACT

In recent years the commercial aim for reconfigurable, upgradable and customized radio systems for multipurpose context, such as automotive and medical diagnostic, makes unthinkable traditional radio systems, implemented over complex, dedicated and expensive costs, with reduced use and power of software. To solve the problem, the complex processing done with expensive and dedicated hardware passes to the software domain. In these systems is used microwave integrated circuitry, highly efficient, such as Field Programmable Gate Array (FPGA).

In the same direction, in this work the main objective is to study and evaluate the implementation of Radar (RAdio Detection And Ranging RADAR) on Software Defined Radio (SDR), reducing the costs of implementation, allowing the reconfigurability of system to many contexts, by using Software Defined Platforms (SDP).

The major synchronization limitations of the of general purpose SDP platforms are identified. Using Multiple-Input Multiple-Output (MIMO) capabilities, it is proposed a way of triggering signals to compensate the failures of synchronism. This method enables the use of general purpose SDP platforms with MIMO capabilities to be used to implement demonstration concepts of radar, in present case of study Frequency Modulated Continuous Wave (FMCW) radar.

Key-Words:

RADAR, Software Defined Radio (SDR), Software Defined Platform (SDP).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ADC	Analog to Digital Converter
BB	Baseband
BP	Bandpass
CW	Continuous Wave
DAC	Digital to Analog Converter
DDC	Digital Down Converter
DSP	Digital Signal Processor
DUC	Digital Up Converter
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FMCW	Frequency Modulated Continuous Wave
FPGA	Field-programmable gate array
HDL	Hardware Description Language
IF	Intermediate Frequency
IS	International System
LFM	Linear Frequency Modulated
LO	Local Oscillator
LP	Lowpass
LPF	Low Pass Filter
LNA	Low Noise Amplifier
MTU	Maximum Transmission Unit
MIMO	Multiple-Input Multiple-Output
PA	Power Amplifier
PLL	Phased Locked Loop
PRI	Pulse Repetition Interval
RADAR	RAdio Detection And Ranging
RCS	Radar Cross Section
RF	Radio Frequency

RX	Receiver
SDR	Software Defined Radio
SDRadar	Software Defined Radar
SDP	Software Defined Platforms
SFP	Small Form-Factor Pluggable
SNR	Signal to Noise Ratio
STALO	STAble Local Oscillator
TX	Transmitter
VCO	Voltage Controlled Oscillator
UDP	User Datagram Protocol
USRP	Universal Software Radio Peripheral
UHD	USRP Hardware Driver
UWB	Ultra-Wide Band

CHAPTER 1 – Introduction

1.1. Overview and Motivation

RADAR is the acronym for RAdio Detection And Ranging, however the term radar has since entered English and other languages as a common noun losing all capitalization. In a brief historical note, radar systems are based on the work of Heinrich Hertz that demonstrated that radio waves are reflected by metallic objects. However, the term radar was only introduced in 1939 by the United States Signal Corp when working on these systems for the Navy [1]. During the Second World War, major improvements were achieved, the radar capabilities developed during that period are acknowledge as one of the decisive factors for the victory of the Allies. Two decades later radar functionalities were enriched with imaging [2]. Since that time, radar principles have been incorporated in all types of surveillance, control and defense, and other types of applications such as medical imaging, automotive navigation, sensing, and other commercial applications.

Radar applications are very broad and cover different fields that demand diverse radar capabilities, leading today to a high number of highly specialized radar systems on the same platform (airplane, ship, etc). In recent years, radar commercial applications with low implementation cost and high adaptability are required, particularly in medical diagnostics and automotive applications [3]. Custom-made wideband Radio Frequency (RF) frontends using dedicated integrated circuits are a cost effective solution for applications where the benefits of economies of scale applies.

However, conventional fixed purpose radar system development requires long development cycle and high cost because of the dedicated individual components, such as Arbitrary Waveform Generators (AWGs), frequency synthesizers, mixers, wideband Analog to Digital Converters (ADC) and Digital to Analog Converters (DAC), etc. By using Software Defined Radar (SDRadar), radar characteristics such as operating frequency, waveform and bandwidth can be easily modified without the need of changing the system hardware [4].

SDRadar applies the same principles as Software Defined Radio (SDR), and enables building a radar system with flexible operating parameters using a reconfigurable hardware platform. SDR is defined by SDR Forum as [5]: “SDR provides software control of a variety of modulation techniques, wideband or narrow band operation, communications security functions (such as hopping) and waveform requirements of current and evolving standards over a broad frequency range.”

In SDRadar most signal detection and processing are performed by digital signal processing either on dedicated hardware such as Field Programmable Gate Arrays (FPGAs) or in a computer. Since SDRadar is software defined, it can switch between different operation modes by modifying both transmit waveforms and receive signal processing tasks as well as operation frequency bands.

Taking advantage of the waveform reconfigurability allowed by SDR, and since Frequency Modulated Continuous Wave (FMCW) radars are widely used in automotive radar systems [6], this dissertation is focused on the implementation of a FMCW radar in a SDR platform.

Synchronization is a major aspect of radar systems. FMCW based radars rely on the measurement of the delay time and frequency shift between the transmitted and reflected waveforms therefore precise and stable time and frequency references are needed. Commercially available, low cost, SDR platforms present imperfections in their local oscillators, the nominal frequency does not correspond to the real frequency, additionally if different oscillators are used they need to be synchronized by the same master oscillator. In the digital domain the sample clocks must also be synchronized and aligned to ensure timing synchronization.

1.2. Objectives

The main objective of this dissertation is to implement a FMCW radar system for target range detection in the SDRs Ettus X300 and NI2920 platforms. To reach the main objective the following task objectives are identified:

1) Identification of the synchronization issues in the SDRs Ettus X300 and NI2920 platforms.
2) Propose and evaluate solutions to overcome the synchronization problems.
3) Demonstrate the FMCW SDRadar system concept.

1.3. Dissertation outcomes

It was identified the major synchronization limitations of the SDRs Ettus X300 and NI2920 platforms. Using opensource GNU Radio toolkit capabilities on SDRs, MATLAB® signal processing techniques and Multiple-Input Multiple-Output (MIMO) capabilities, it is proposed a way of triggering signals to compensate the failures of synchronism. It was also applied, calibration tests and techniques to compensate the phase offset response of local oscillators and their frequency deviation.

1.4. Structure

Following this brief introduction, Chapter 2 will introduce FMCW radar concepts and theoretical fundamentals. First is presented some general concepts about radars and then is presented the architecture of a FMCW Radar.

In Chapter 3, is showed the implementation of a radar system based on SDR platforms. The SDR architecture is firstly presented to explain its operation and functionalities, then, it is discussed the implementation and operation of FMCW radar on SDR system. To contextualize this work, are presented three different cases of similar implementations that serve as comparison test beds.

Chapter 4 discusses and tests the major synchronism limitations of SDRs platforms available in the laboratory.

In Chapter 5, it is demonstrated that, mainly due to the unpredictable latency of the SDRs used it is not possible to implement a radar system following available literature proposals. In the second part of the chapter, it is discussed a novel implementation of a SDRadar that exploits the MIMO synchronization capabilities of the commercial SDR platform Ettus X300. The chapter ends with the performance assessment of the SDRadar in terms of range profiling.

Finally, in Chapter 6, the conclusions of this work are presented as well as future work suggestions.

CHAPTER 2 –FMCW radar system principles

This chapter discusses the FMCW radar concept. The theoretical fundamentals as well the main advantages of this type of Continuous Wave (CW) radar are introduced. First, an introduction to some basic CW radar definitions is presented. Then, is presented an analysis of a Linear Frequency Modulated (LFM) architecture and signal processing. All analysis was done based on the reference books [7] and [8].

All equations and physical parameters are given in fundamental units, it is assumed from here that all parameters are in International System (IS) units, unless otherwise explicitly indicated.

2.1. Radar Classification

Radars can be classified as pulsed or continuous wave, based on the type waveform used. Continuous wave radar emits continuously electromagnetic field and uses different antennas to transmit and receive [7].

Nowadays, FMCW radars are widely used for industrial and consumer electronics [6] [9] [10], giving reasons to implement this type of radar.

2.2. FMCW radar architecture overview

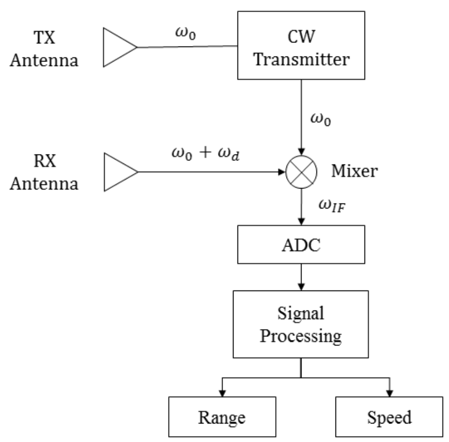

Figure 1- CW Radar block diagram. Image adapted from [7].

Fig. 1 shows a simplified diagram of FMCW radar. In contrast to pulsed radars, FMCW radars transmit and receive simultaneously during acquisition and utilize a STAble Local Oscillator (STALO) which is modulated in frequency (not shown in the figure, is part of transmitter). The FMCW waveform is amplified in the transmitter (not shown in the figure) and the electromagnetic waves are sent through the Transmitter (TX) antenna to the air. The electromagnetic waves will be reflected by the target object. Then the Receiver (RX) antenna receives the reflected waves from the target, which are processed by a Low Noise Amplifier (LNA) (not shown in the figure) and then mixed with a copy of the transmit FMCW waveform. The signal reflected from the target lags the transmitted signal by a frequency proportional to the time delay between the transmitted and reflected signal, and since continuous wave emitted signal has the form of sinewave with center frequency f_0, when the target is moving, the spectra radar echo is shifted by f_d from f_0, and therefore is possible extract the target radial velocity and angular position. The radar received signal is subsequently down-converted by mixing with the transmit signal to produce an Intermediate Frequency (IF) output, which is digitalized and processed to obtain target range and speed, however it is not intended to exploit speed detection in this work. Mixing the transmitted and receive waveforms effectively down-converts the frequency difference to the baseband, allowing the ADC to sample at lower rates. The data samples can be processed in real-time or stored and processed offline.

2.3. Radar Definitions

2.3.1. Range

Considering a two-way path, with radar echo received with a time delay Δt from a target at range R. R is computed by measuring the time delay it takes to a signal to travel from the transmitting antenna be reflected by the target and received by the receive antenna.

$$R = \frac{c\Delta t}{2}, \tag{1}$$

where, c is the speed of light.

To calculate the range is of key importance to have a reference point in the transmitting waveform. For example, if the transmitting waveform is a train of pulses, the time separation between the pulses should be high enough so that the reflected pulse from the target is measured without ambiguity.

2.3.2. Range resolution

Range resolution, ΔR, is a measure of capacity of radar to detect targets near each other as distinct objects,

$$\Delta R = \frac{c}{2B}, \quad (2)$$

where B is the bandwidth of the signal waveform.

2.3.3. Doppler frequency

Doppler frequency is used to determine target radial velocity and then distinguish between moving and stationary targets. According to Doppler phenomenon, if we consider the transmitting waveform a CW carrier with frequency f_0. The frequency deviation of reflected signal from the carrier frequency f_0, i.e. is computed as,

$$f_0' = \frac{c+v}{c-v} f_0 \ , \quad (3)$$

where, v is the target velocity. The frequency shift can be positive or negative depending on target's direction/motion. Doppler frequency is computed as the difference between shifted carrier frequency and original carrier frequency,

$$f_d = f_0' - f_0 \ . \quad (4)$$

Since $v \ll c$, then

$$f_d \approx \frac{2v}{c} f_0 = \frac{2v}{\lambda} \ \Rightarrow v = \frac{\lambda f_d}{2} \ , \quad (5)$$

where, λ is the radar operating wavelength.

Doppler frequency depends on target velocity component in the direction of radar, designated as radial velocity. Thus, to account situations where the target radial velocity with respect to the radar velocity is not in the same direction, and then the relation is not direct, target velocity is described as function of θ, that represents the angle between line of sight defined by radar receiver antenna, and the trajectory of target,

$$f_d = \frac{2v}{\lambda} cos\theta \ , \quad (6)$$

with, $\cos\theta = cos\theta_e \cos\theta_a$, where θ_e and θ_a are respectively the elevation and azimuth angles.

2.3.4. Coherence and time synchronization

Coherence is the capacity of radar to maintain the phase between consecutive pulses, i.e. if the phase between two pulses is consistent, the radar is coherent. If the radar stores in memory the phase reference of transmitted signals, then the radar is classified as coherent-on-receive. This type of radar has the advantage to take care of situations where is not guaranteed constant phase deviation in transmitted signals. To achieve coherency must be used a STALO [7].

As referred in Sections 2.2 and 2.3.1 target range information depends on the measure of time delay it takes to a signal to travel from the transmitting antenna be reflected by the target and received by the receive antenna, for that reason, the only way to extract this information is assuring that both transmitter and receiver share the same clock reference, providing time synchronization.

2.3.5. Radar range equation

The maximum achievable radar range is given by,

$$R_{max} = \left(\frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 S_{min}}\right)^{1/4}, \qquad (7)$$

where P_t is the power transmitted, G_t and G_r are respectively transmitter and receiver antenna gain, λ is the radar operating wavelength, σ is the Radar Cross Section (RCS), and the S_{min} is the minimum power detectable by radar.

2.4. FMCW radar waveform generation

The architecture of FMCW is based on a Voltage Controlled Oscillator (VCO) that generates a RF carrier with frequency controlled by the applied voltage. FMCW waveforms can be generated with a sawtooth, triangular or sinusoidal waveform as input signal $s_i(t)$ of VCO that produces a frequency proportional to the amplitude of the input signal. Fig. 2, illustrates the mathematical model of generation of a frequency modulated signal, where the signal at the output of the VCO is $x(t) = A\cos(\omega_0 t + \theta(t))$ [8].

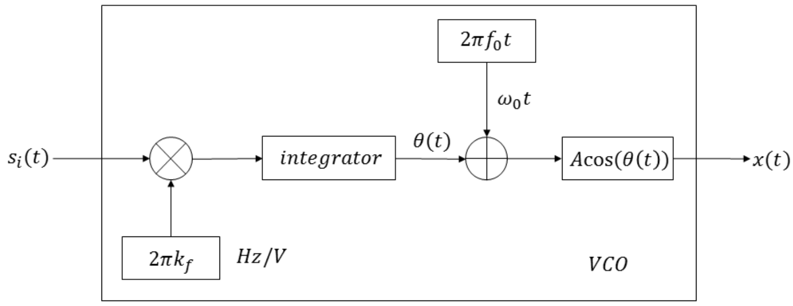

Figure 2- Voltage Controlled Oscillator. Image adapted from [8].

At the output of integrator, the phase, $\theta(t)$, is obtained from the input signal as,

$$\theta(t) = 2\pi k_f \int_0^t s_i(t)dt \ , \tag{8}$$

where, k_f in the modulation sensitivity.

When $s_i(t) = 0$, the output of VCO is a cosine wave with center frequency f_0, i.e the carrier,

$$x(t) = A\cos\,(\omega_o t) \ , \tag{9}$$

where, A is the carrier amplitude.

When $s_i(t) \neq 0$, the output of VCO appears a frequency modulated signal instead of a sinewave with carrier frequency f_0, expressed by,

$$x(t) = A\cos\,(2\pi f_0 t + 2\pi K_f \int_0^t s_i(t)dt) \ . \tag{9}$$

2.5. Linear Frequency Modulated CW Radar

Linear frequency modulation or 'chirp' is a widely used waveform in radars. The amplitude of the signal is kept constant and the frequency varies linearly. Fig. 3 illustrates a sawtooth LFM waveform where the frequency of the carrier is linearly modulated during *T* seconds with a swept bandwidth *B*. The time-bandwidth product of the LFM waveforms equal to *TB* (pulsewidth x Bandwidth). The dashed line in Fig. 3 represents the return waveform from a stationary target at range *R*.

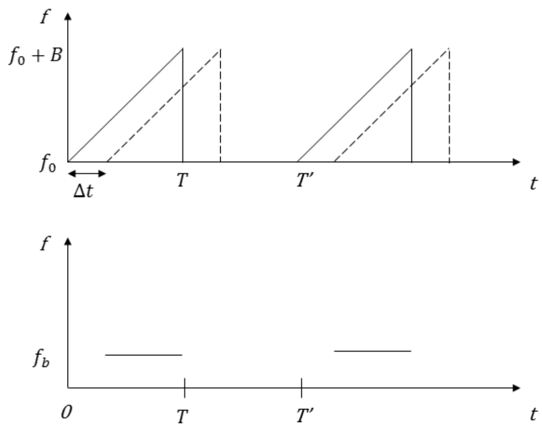

Figure 3- Transmitted and received sawtooth LFM signals and beat frequency from stationary target. Image adapted from [7].

The beat frequency, f_b, is also sketched in Fig. 3. It is defined as the difference between the frequencies of the transmitted and received signals.

$$f_b = f_{received} - f_{transmitted} \cdot \tag{10}$$

The time delay is a measure of target range

$$\Delta t = \frac{2R}{c} \, . \tag{11}$$

Radars that use this type of waveform modulation, sawtooth, that is a type of FMCW radar, just can measure range information, although it does it with a simple and efficient architecture which is an advantage [6].

Each individual frequency sweep is commonly referred to as a chirp. While many FMCW radars do transmit continuously with zero inter-chirp delay, a radar which does not transmit for a period between chirps would still be considered FMCW.

Referring to Fig. 3, T is the duration of the chirp and the time T', also sketched is the Pulse Repetition Interval (*PRI*).

The rate of frequency change also called sweep slope, α, is defined as,

$$\alpha = \frac{B}{T}\,. \tag{12}$$

Thus, the beating frequency that corresponds to the target return signal is given by,

$$f_b = \Delta t\alpha = \frac{2R}{c}\frac{B}{T}\,. \tag{13}$$

In this present work, non-stationary target moving is not analyzed, although some considerations that must be take into account, for both stationary and non-stationary target detection, are presented.

2.6. Waveform band-pass representation

As we seen in Section 2.4, a LFM signal in analogue domain is expressed by,

$$x(t) = A\cos\,(2\pi f_0 t + 2\pi K_f \int_0^t s_i(t)dt)\,. \tag{16}$$

Basis on architecture presented in Section 2.5, where is used a sawtooth waveform, corresponding to an up-chirp, is described as,

$$s_i(t) = \begin{cases} \alpha t, & 0 \le t \le T \\ 0, & otherwise \end{cases}, \tag{17}$$

where, $\alpha = \frac{B}{T}$ is sweep slope parameter.

Solving the integral for $s_i(t) = \alpha t = \frac{B}{T}t$, we have,

$$\theta(t) = 2\pi K_f \frac{B}{T}\frac{t^2}{2},\ 0 \le t \le T\,. \tag{18}$$

Thus, LFM signal for sawtooth waveform is expressed in analogue domain by,

$$x(t) = A\cos\left(2\pi f_0 t + 2\pi K_f \frac{B}{T}\frac{t^2}{2}\right),\ 0 \le t \le T\,. \tag{19}$$

Since that all signal generating and processing will be done in digital domain, assume from here that $K_f = 1\,Hz/V$, and to simplify that $A = 1V$. Thus, LFM signal can be expressed by,

$$x(t) = cos\left(2\pi f_0 t + 2\pi \frac{B}{T}\frac{t^2}{2}\right), 0 \le t \le T\,. \tag{20}$$

The instantaneous phase of signal can be expressed by,

$$\phi(t) = 2\pi(f_0 t + \frac{B}{T}\frac{t^2}{2}), 0 \leq t \leq T \; . \tag{21}$$

Then, instantaneous frequency of signal is given by,

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\phi(t) = f_0 + \frac{B}{T}t, 0 \leq t \leq T \; . \tag{22}$$

Looking at instantaneous frequency equation, we can see that signal contains a significant frequency composition away from origin, hence is called Bandpass (BP) signal, more specifically, narrow bandpass signal, because $f_0 \gg B$. Signals of this type can be described as in-phase and quadrature components,

$$x(t) = x_I(t)\cos(2\pi f_o t) - x_Q(t)\sin(2\pi f_o t) \; , \tag{23}$$

where, $x_I(t)$ and $x_Q(t)$ are Lowpass (LP) components given by,

$$x_I(t) = \cos\left(\pi \frac{B}{T} t^2\right) , \tag{24}$$

$$x_Q(t) = sin\,(\pi \frac{B}{T} t^2) \; . \tag{25}$$

Thus, complex Baseband (BB) signal envelope is given by,

$$\tilde{x}(t) = x_I(t) + j x_Q(t) \; . \tag{26}$$

Where we can see that,

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} real(\tilde{x}(t)) \\ imag(\tilde{x}(t)) \end{pmatrix} = \begin{pmatrix} \cos\,(\pi \frac{B}{T} t^2) \\ sin\,(\pi \frac{B}{T} t^2) \end{pmatrix} . \tag{27}$$

In Fig.4 is represented the quadrature BB signals of an up-chirp, with bandwidth of 20MHz and sweep time of 0.1 ms.

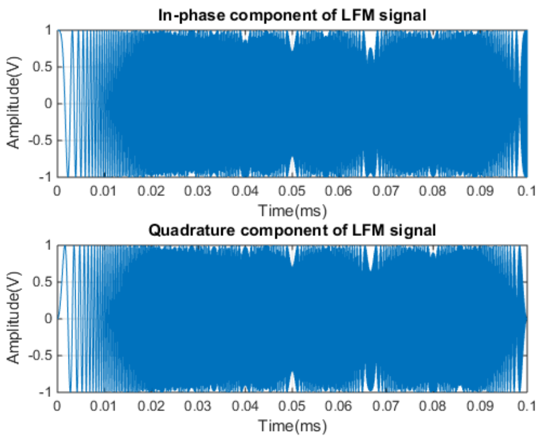

Figure 4- Quadrature baseband signals of an up-chirp with bandwidth of 20MHz and sweep time of 0.1ms.

2.7. Digital signal processing

2.7.1. Sample rate

Since LFM is a limited and known bandwidth signal, to satisfy sampling theorem the sample rate must respect,

$$f_s \geq 2B \,. \quad (28)$$

The time duration of sweep T is related with the number of samples N, by,

$$f_s = \frac{N}{T} \geq 2B \,. \quad (29)$$

Thus, we can define the frequency resolution as,

$$\Delta f = \frac{1}{T} = \frac{fs}{N} \geq \frac{2B}{N} \,. \quad (30)$$

To ensure that analysis has an appropriated resolution and determined frequencies are correct in signal processing, should be adopted a total number of samples $N_{FFT} \geq N$.

2.7.2. Mixer

The beat frequency is generated in the receiver of the FMCW radar by a mixer[1] or 'multiplier' as illustrated in Fig.1.

Mixing transmitted $x_T(t) = A_T cos\Phi_T$ with the received signal $x_R(t) = A_R cos\Phi_R$, the resulting signal is,

$$x_T(t).x_R(t) = A_T cos\Phi_T . A_R cos\Phi_R = \frac{A_T A_R}{2}[cos(\Phi_T - \Phi_R) - cos\Phi T{+}\Phi R, \quad (31)$$

the phase-sum term represents an oscillation at twice the carrier frequency, which is generally filtered out, or more usually in radar systems it is filtered because it is beyond the cut-off frequency of the mixer and subsequent receiver components. We are therefore left with the phase-difference which is called the 'dechirped', 'beat', or 'Intermediate Frequency' (IF) signal. Considering a time delay τ, between the transmitted and received waveform

[1] A mixer is a three-port device that uses a nonlinear or time-varying element to achieve frequency conversion (Pozar 2005). In its down-conversion configuration, it has two inputs, the radio frequency (RF) signal and the local oscillator (LO) signal. The output, or intermediate frequency (IF) signal, of an idealized mixer is given by the product of RF and LO signals.

$$x_{IF}(t) = co\,s\left(2\pi f_0\tau + 2\pi\frac{B}{T}\frac{t^2}{2} - 2\pi\frac{B}{T}\frac{(t-\tau)^2}{2}\right) = cos\left[2\pi\left(\frac{B}{T}\tau t + f_0\tau - \frac{B}{2T}\tau^2\right)\right] \quad (32)$$

where, $f_0\tau$, is the total number of cycles of that occur during the round trip propagation time from the transmitter to the target, $f_b = \frac{B}{T}\tau$, is the beat frequency that is used to calculate the time delay between the transmitted and reflected waveforms and $\frac{B}{2T}\tau^2$ is a range dependent phase term.

2.7.3. Low pass filter

After mixing transmitted and received signals and, if processing is done in digital baseband domain or if the mixer has a cut-off frequency greater than the maximum bandwidth of radar, its necessary filter resulting signal containing beating frequencies to guarantee that all targets are in the range of radar. The Low Pass Filter (LPF) which is not shown in Fig. 1, is an important part of signal processing. This filter is of type Finite Impulse Response (FIR), with bandwidth and cutoff frequency B, and f_{stop} equal to f_s, and with linear phase.

2.7.4. Target range calculation

The range can be calculated from the beat frequency, by using equation (13). The beat frequency is evaluated, digitally, by calculating the power spectral density of $x_{IF}(t)$, and finding its maximum. Fig. 5 illustrates, a simulated power spectral density of a dechirped signal, corresponding to a static target located 191.235 meters away from radar. An FMCW waveform with 20 MHz bandwidth and chirp duration of 0.1ms, as illustrated in Fig. 4 is considered.

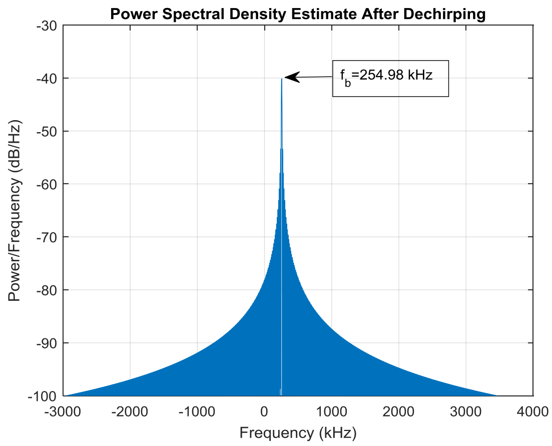

Figure 5- Power spectral density of dechirped signal of a simulated target at range 191.235 m.

CHAPTER 3 – Implementation of a SDRadar on SDR platform

This chapter discusses the implementation of a Radar system based on a SDR platform. The chapter starts with the general overview of SDR architectures. The second part of the chapter introduces how SDR platforms can be used to implement SDRadar. The third part of the chapter presents a brief state of the art of SDRadar. A range of software defined radios designed and sold by Ettus Research and its parent company, National Instruments adopted the nomination of Universal Software Radio Peripheral (USRP), since in this dissertation the SDRs platforms used are of these brands the name SDRs and USRP are used interchangeably.

3.1. SDR Architecture with IF up/down conversion

The objective of using SDR is to decrease the costs associated with dedicated hardware systems, by using standard systems and letting the hardest part of processes be done with software. SDR enables user's applications development for multipurpose, high performance, small size, low cost and possibility of improving system performance, improving signal processing algorithms. The main benefits are upgradeability, customization and adaptability [11] [12]. To accomplish that, standard hardware that implement SDR and a Digital Signal Processor (DSP), usually a host computer with Signal Processing software, must be capable to support some basic physical capabilities that make feasible multipurpose implementations. Fig. 6 illustrates a basic SDR architecture.

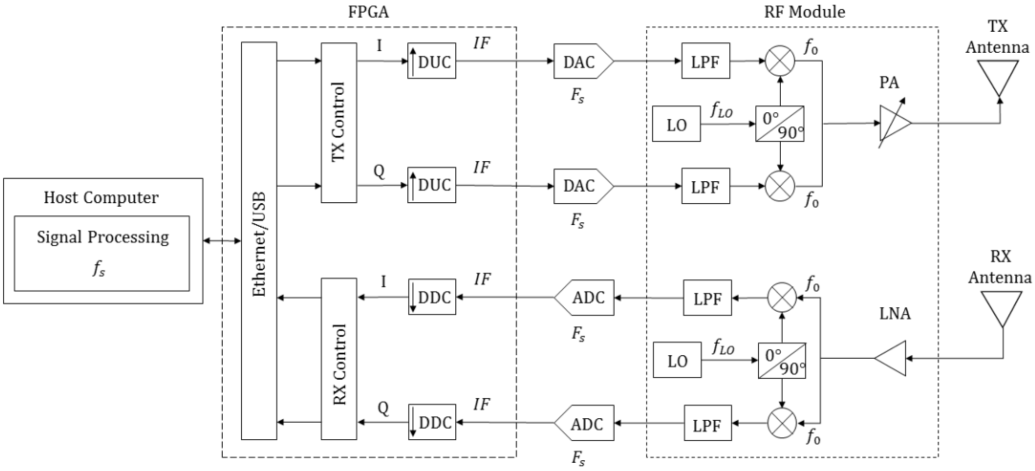

Figure 6 - SDR Architecture with Heterodyne operation. Image adapted from [13].

This architecture can be divided in 8 main groups:

A. Host Computer

Host computer assures the connection to USRP and perform baseband signal processing, including modulation. Software tools such as MATLAB®, GNURadio, etc... can be used. It also controls some hardware basic parameters, such as, operating frequency f_0, gain of transmitter/receiver (RF Module), sample rate f_s, dimensions of transmitter and receiver buffers and size of Maximum Transmission Unit (MTU) in case of Ethernet connection. It sends and receives signals to/from USRP and does signal processing.

B. Interfaces

The connection of the host computer to the USRP can be done in four different ways:

- Bus Series – Connected to a host computer via a USB connection;
- Network Series – Connected to a host computer via an Ethernet connection;
- High Performance Series – Connection can be Ethernet (1Gbit or 10 Gbit) or a x4 PCI-Express connection;
- Embedded Series – Meant to run stand-alone (without a host computer).

C. FPGA

This module is responsible to sample rate conversion and channelization. On the transmitting side this process is done by Digital Up Converter (DUC) that translates baseband signal, sampled by f_s, to the operating sampling rate of the hardware, and sends it to DAC. On receiving side, the inverse process is done with Digital Down Converter (DDC) that receives a signal from ADC at a given sample rate re-sample it down, extracting baseband signals [14]. Next, signal is sent through Ethernet, USB or etc.., connections to the signal processor. As note, the FPGA is programmed with code written in a Hardware Description Language (HDL). Most USRP makers provide pre-built images, therefore the user does not need to program the FPGA unless different functionalities are required.

D. DAC/ADC

Following DUC in transmitting side, DAC converts the digital samples into the analog signal, after the signal is up-converted to RF frequency in the RF module. Similarly, on receiving side, signal that came from RF Module is digitized by the ADC and then sent to DDC.

The ADC/DAC sample the full bandwidth of the signal. If the signal is localized at an IF and occupies the bandwidth B, the ADC/DAC must assure a sample rate of [15],

$$F_s \geq 2(IF + 0.5B). \tag{33}$$

This equation expresses a practical limitation of resolution achievable in SDR, by using this architecture. Although, typical CW radars use this architecture [7].

E. RF Module

RF Module performs up and down conversion of the IF or baseband signal to/from the RF operating frequency, it also couples that circuitry to the antenna or its feeder [14]. In the transmission path, signal is equalized with a LPF, converted to the operating frequency, by mixing with a Local Oscillator (LO), amplified with a Power Amplifier (PA) with adjustable gain, and finally transmitted over the air through antenna. In the reception path, received signal from antenna is amplified with LNA, translated to baseband frequency and filtered with LPF before sent to ADC.

F. Antenna

The chosen antenna should ensure the optimum performance in terms of, adaptability gain of frequency band, correct polarization, maximum power orientation, maximum power efficiency, directivity, and beam-steering if needed. All these parameters should be considered taking into account the application.

G. Hardware Drivers

Depending on the USRP hardware there are different ways to access it from several software's. The most common option is using drivers provided by USRP developers. For the platforms that are considered in this dissertation it is a C^{++} interface. Ettus Research™ [2] provides a driver, the USRP Hardware Driver (UHD).

H. Software Tools

The two software tools considered in this dissertation is GNU Radio software and MATLAB®. GNU Radio is a free, open-source toolkit that provides DSP functions and most importantly GNU radio provides a SDR interface library that supports a very large number of commercially available platforms. GNU radio can be used to write applications to receive data out of digital streams or to push data into digital streams, which is then transmitted using hardware. GNU Radio can also work alone without real USRPs as a simulation environment by using pre-recorded or generated data to create a simulation environment. GNU Radio applications are commonly written in Python as a wrapper and combine DSP blocks integrated in GNU Radio. The DSP blocks implemented in C++ perform critical performance DSP tasks such as modulations, demodulations, filters, mixers, and signal operators.

MATLAB® and Simulink® are well known powerful numerical mathematical software platforms with a huge number of toolbox applied to different fields. Recently MATLAB® and Simulink lunched a Support Package for USRP. This software package provides the interface between MATLAB® and the USRP. This software was initially developed by Institute for Communications Engineering at Technical University of Munich. It can set the basic configuration options of the USRP and send and receive

[2] Ettus Research™, a National Instruments (NI) brand since 2010, is the world's leading supplier of software defined radio platforms, including the Universal Software Radio Peripheral (USRP™) family of products.

samples to and from a vector. Actually, MATLAB® uses the UHD drivers, a UHD Binary License is required for each USRP device to be used with MATLAB® or Simulink. Every USRP device includes one free license when the device is purchased directly from National Instruments or authorized distributors.

In summary:

- MATLAB® is widely used for off-line analysis (even with USRP), while GNU radio is often used for real-time analysis.
- To perform real-time analysis, MATLAB® code must be translated into C-code and be compiled, using tools like MATLAB® Coder or Simulink Coder. This supports only a subset of core MATLAB® language features. In GNU radio there is no such restriction.
- The GNU radio flow graph is built in python code, which do not need compilation before execution, for real-time analysis.
- GNU radio has better driver support for USRPs. MATLAB® doesn't support legacy USRP.
- Graphical flow graph GNU radio clearly indicates data type of each link in different color.

Summarizing, SDR architecture ensure the possibility of use the appropriate modulation, operating frequency, sample rate, gain on transmission and reception, upgrade software on host computer and then improving both approaches and signal processing algorithms, all of this with possibility of reconfiguring system based in new approaches.

3.2. SDR architecture with baseband up/down conversion

SDR systems that translate directly baseband signals to radio frequency signals have a more simplified hardware architecture, are conceptually simpler and closer to theoretical fundamentals and requires DACs and ADCs with lower sampling rate when compared with architectures that employ IF up/down conversion.

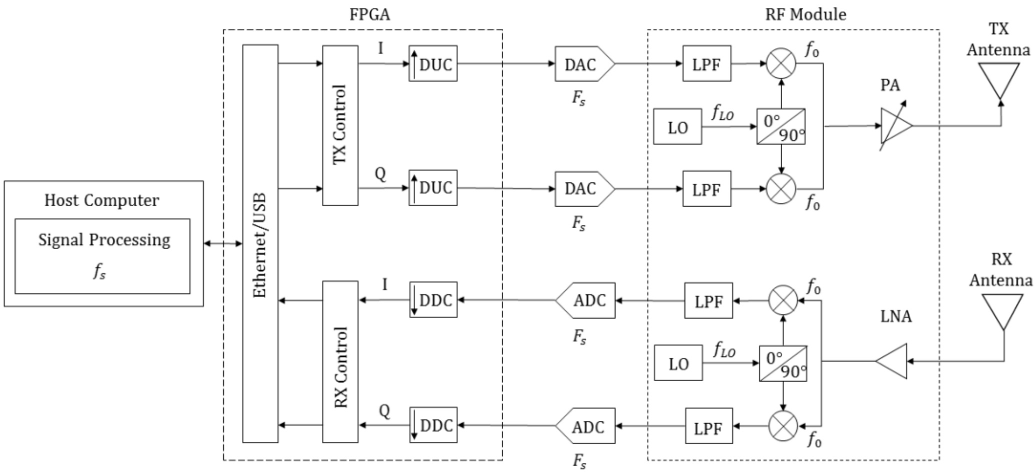

Figure 7- SDR with Homodyne Architecture. Image adapted from [13].

As shown in Fig. 7 baseband signals are directly converted to operating frequency after DAC. For this reason, this architecture is also called zero-IF or direct conversion [16]. This process is possible and assures the integrity of signals since is used a pair of ADC s in-phase and quadrature in both paths. The LO is shifted by 0° and 90° for mixing the in-phase and quadrature components respectively. To guarantee best performance, bandwidth of baseband signal, B, must respect ADCs sample rate F_S that is given by,

$$F_S \geq 2B. \tag{34}$$

In comparison to previous architecture, Homodyne architecture allow higher signal bandwidths.

3.3. SDRadar system overview

The architecture design of software-defined FMCW radar system is illustrated in Fig. 8.

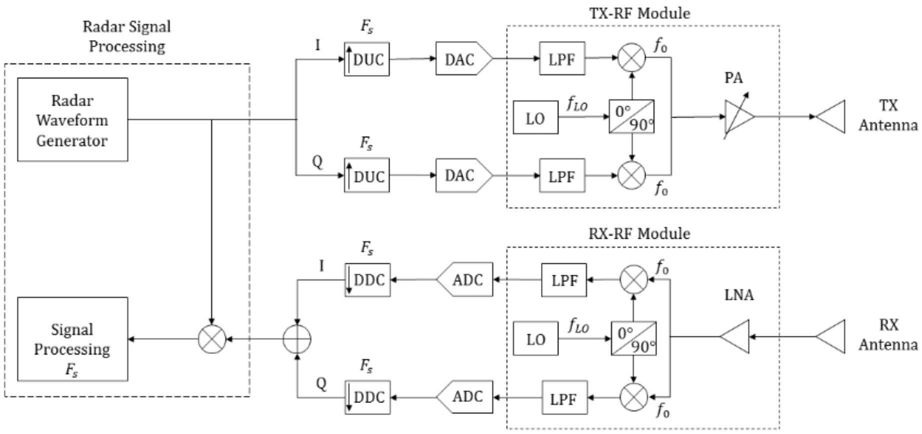

Figure 8-FMCW Radar architecture based on SDR with Homodyne architecture. Image adapted from [13].

The radar waveform generator is a baseband signal processor that generates the complex envelope of the radar waveform, described in equation (26) by,

$$\tilde{x}(t) = x_I(t) + jx_Q(t),$$

Where quadrature components of envelope are described by equations 24, 25 and 27,

$$\begin{pmatrix} x_I(t) \\ x_Q(t) \end{pmatrix} = \begin{pmatrix} real(\tilde{x}(t)) \\ imag(\tilde{x}(t)) \end{pmatrix} = \begin{pmatrix} \cos(\pi\frac{B}{T}t^2) \\ sin(\pi\frac{B}{T}t^2) \end{pmatrix},$$

The in-phase, $x_I(t)$ and quadrature, $x_Q(t)$ components are sent from the host computer to the SDR. Then, after DUC, DAC and LPF steps, in RF Module, the analog baseband I/Q components are mixed respectively with the component in phase and quadrature of the local oscillator, as illustrated in TX-RF Module in Fig. 8.

In the output of RF module, signal is described as original bandpass signal by equation 23 as,

$$x(t) = x_I(t)\cos(2\pi f_o t) - x_Q(t)\sin(2\pi f_o t),$$

Substituting quadrature components described in equations 24 and 25,

$$x(t) = \cos\,(\pi \frac{B}{T} t^2) \cos(2\pi f_o t) - \sin\,(\pi \frac{B}{T} t^2) \sin\,(2\pi f_o t), \tag{35}$$

Equivalently,

$$x(t) = \frac{1}{2}\left[\cos\left(\pi \frac{B}{T} t^2 - 2\pi f_o t\right) + \cos\left(\pi \frac{B}{T} t^2 + 2\pi f_o t\right)\right] - \frac{1}{2}\left[\cos\left(\pi \frac{B}{T} t^2 - 2\pi f_o t\right) - cos\left(\pi \frac{B}{T} t^2 + 2\pi f_o t\right)\right]. \tag{36}$$

Simplifying equation, we obtain a RF modulated signal, corresponding to designed LFM signal.

$$x_{RF}(t) = co\,s\left(2\pi f_0 t + \pi \frac{B}{T} t^2\right). \tag{37}$$

This signal corresponds to the signal that is sent by the USRP, it is equivalent to equation 20 in Section 2.6. The inverse process takes part on receive path, as shown in RX-RF Module in Fig. 8. The reflected signal from target is received, amplified with LNA, mixed with radar operating frequency components, extracting into baseband analog quadrature components, which have now additive noise, resulting from channel propagation. Are then, filtered with LPF to eliminate high frequency components, digitized and sampled down to original sample rate, finally complex envelope is formed and sent to signal processor, where signal processing is done, in order to obtain target range.

3.4. SDR architecture limitations

Comparing the architecture of a SDRadar with the architectures of SDR overviewed in this chapter, the most evident drawback of SDR platforms is the lack of a STALO. Conventional SDR platforms use two distinct oscillators for the transmitter and receiver RF chains. For radar operation it is fundamental that the receiver and transmitter share the same oscillator, if different oscillator are used they should be synchronized. The system must maintain a known phase relationship between each RF input or output. Another important aspect that must be taken into account is the delay introduced by the signal processing and delays introduced by the hardware. Low cost SDR platforms architectures comprise of multiple heterogeneous processing units with interconnecting buses leading to a large amount of delay and a low throughput.

3.5. State of the art of SDRadar Implementation

In [17] it is presented a FMCW radar for weather surveillance based on SDR using GNU Radio Software and MATLAB®. It is a simulation work that uses an USRP N210 [18] in a loopback configuration and Ultra-Wide Band (UWB) antennas. The received signal is the transmitted signal with and added delay (simulating a reflection from a target). The transmitted and received signals are mixed, low pass filtering is applied followed by a Fast Fourier Transform (FFT) on the resultant signal. The computed range is according to the added delay. The system was evaluated with a sawtooth waveform, with bandwidth 0.75 MHz, sweep time of 1 ms (covering of up to 150 m) and sample rate of 6 MS/s. The signal was processed using MATLAB®, and GNU Radio was used to interface with the SDR. The possibility of speed measurement was not tested.

In [19] it was demonstrated experimentally the implementation of a FMCW surveillance radar for drone detection on SDR using GNU Radio software, and USRP B210 [20]. The radar waveform signal was generated with a dedicated block in GNU Radio and written to a binary file. After, GNU Radio open the file containing the signal and with USRP, transmits and receives signals through directional antennas. After, GNU Radio sent both signals via User Datagram Protocol (UDP) protocol connection to another computer were signal processing is performed with Python. To deal with the incapacity to determine both range and Doppler, because of use an up-chirp, in signal processing the double FFT is taken for multiple targets. In experimental tests, radar detected a stationary target at range of 60 m, although was found Doppler drift effect on Doppler map, concluding this effect is caused by phase drift in LO of USRP. This error was compensated with phase correction algorithm. Finally, was tested a moving target (car) moving at 20 m/s, identified correctly in Doppler map. The system was designed with sawtooth waveform, with bandwidth of 28 MHz and sample rate of 28 MS/s, that is no guarantees of non-ambiguous detections operating frequency 5.5 GHz. Resulting in a resolution of 5.35 m, maximum unambiguous velocity of 30 m/s with resolution of 1.2 m/s. The system was tested with directional antennas with a gain of 10 dB, a PA and LNA respectively in the output and input of USRP. The system has 63.9 % probability of detection.

In [3] a system was implemented based on USRP NI2920 supported by LabView Software and is divided in two approaches, outdoor and indoor to verify the capacity of system to distinguish the same scenario when multipath is present or not. All signal processing was done with LabView that generates an up-chirp signal. USRP transmits and receives signals, then, after LabView perform mixing operation and low pass filtering, coherent detection is done to increase SNR in resulting signal, increasing the capacity to accurately extract target range. In outdoor test the previous target position was set using a foil plate. Proving that system is capable to measure accurately the target range in different ranges. Indoor approach was tested into anechoic chamber using same scenario. After calibration eliminates false alarm peaks and proving the capability to detect targets, but don't prevent incorrect scenario interpretation. The system was designed with sawtooth waveform, operating frequency 1.8GHz (L-Band) and bandwidth of 25 MHz, resulting in a resolution of 6 m, that is possible using 1GbE connection. Outdoor system was tested with directional antenna as transmitter and a log periodic antenna as receiver. Indoor utilizes a broadband logarithmic antenna as both transmitter and receiver, controlling both paths with a circulator.

CHAPTER 4 – Assessment of the USRPs capabilities and limitations

This chapter discusses and tests the major synchronism limitations of SDRs platforms available in the laboratory.

4.1.Characteristics of USRPs NI2920 and X300

In this Section are discussed characteristics of USRPs available on the Optical Laboratory at the Institute of Telecommunications, the X300 and the NI2920. The X300 [21] equipped with CBX-120 and SBX-120 daughterboards, and NI2920 [22] equipped with WBX-40 daughterboard. Since both platforms have similar architecture and will be used with same resources, in the figure below we present a general architecture of both.

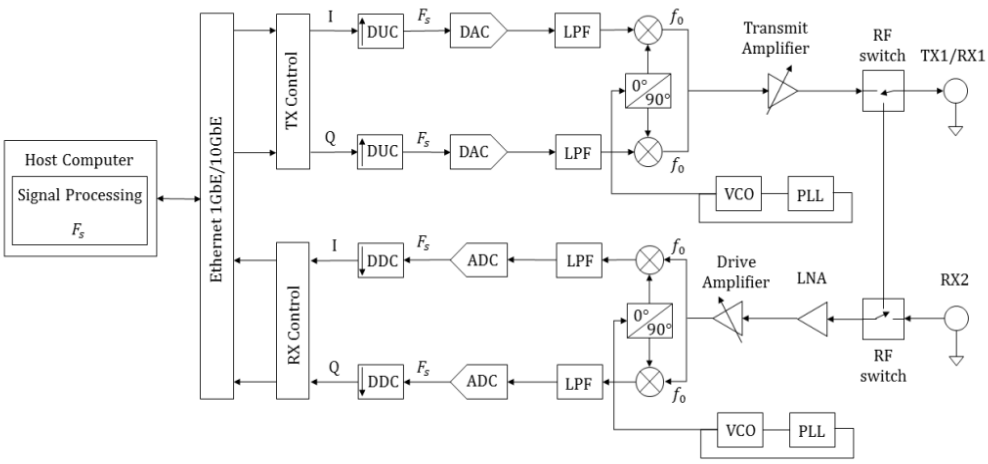

Figure 9- USRPs NI2920 and X300 basic architecture. Image adapted from [13].

The major difference between these USRPs is that X300 has better performance than NI2920 in terms of FPGAs, sample rate of the DAC/ADC, increasing signal resolution and supporting daughterboards with higher bandwidths. X300 has two bidirectional channels each of them supported with 1GbE interfaces and/or 10GbE interfaces Small Form-Factor Pluggable (SFP+) that can be configured based on requirements. While NI2920 has one bidirectional channel only supported with 1GbE interface.

An important characteristic of the USRP X300 is that its two channels can be synchronized. The two transmitters can transmit data simultaneously and synchronously and the same applies for the receivers. Table 1 summarizes the most important characteristics of the two USRPs.

Table 1- Characteristics of USRP X300 equipped with CBX-120 and SBX-120 and NI2920 equipped with WBX-40, [21] [22].

X300		NI2920	
Specification	Value		Unit
Conversion Performance			
FPGA	Xilinx Kintex-7 XC7K325T	Xilinx® Spartan® 3A-DSP 3400	
Master Clock	200.0 MHz and 184.32 MHz	100MHz	
ADC Sample Rate	200	100	MS/s
ADC Resolution	14	14	bit
DAC Sample Rate	800	400	MS/s
DAC Resolution	16	16	bit
SBX – 120 Daughterboard		WBX-40 Daughterboard	
Frequency range	400 to 4400	50 to 2200	MHz
Bandwidth	Up to 160 MHz bandwidth per channel	40	MHz
TX/RX Gain	0 to 31.5	0 to 31.5 (0.5 gain step)	dB
Maximum Input Power	0	0	dBm
Power Sensitivity	-98.67	-98.52	dBm
Noise Figure	-	5	dB
CBX-120 Daughterboard		-	
Frequency range	1.2 to 6	-	GHz
Bandwidth	120	-	MHz
TX/RX Gain	0 to 31.5	-	dB
Maximum Input Power	0	-	dBm
Power Sensitivity	-98.91	-	dBm
Noise Figure	-	-	dB

4.2. Bandwidth limitation

The radar resolution is directly proportional to the bandwidth of the FMCW waveform. Therefore, the maximum achievable bandwidth provided by the USRP limits the radar bandwidth. The bandwidth of the USRP is limited mainly by three functional blocks, the analog bandwidth, FPGA processing bandwidth and the host interface bandwidth.

The analog bandwidth is the amount of useful bandwidth (3 dB) between the RF port and IF/baseband interface of an RF channel. This bandwidth is set by RF frontend or by the baseband filters on the daughterboard, which are designed to avoid aliasing.

The FPGA processing bandwidth sets the maximum digital bandwidth of a system based on the USRP. For example, the FPGA of the USRP X300 sends and receives samples at maximum 200 MS/s from the DACs and ADCs while the USRP NI2920 only sends and receives samples at 25 MS/s.

The host bandwidth is mainly set by the host host computer interface and the USRP. The host interface streams data between the FPGA of a USRP device, and a host host computer. Most applications stream I/Q data to and from the USRP device, sampled and transmitted using 16-bit or 8-bit. The data rate depends on the interface used as well as on the host computer and the processing load. The table below summarizes achievable sample rates referred by the equipment suppliers, for the USRP X300 and NI2920.

Table 2- Achievable sample rate by interface and USRP.

Interface	USRP	Host Sample Rate (MS/s @ 14-bit I/Q)	Mode
1 Gigabit Ethernet	X300/ NI2920	25	Full Duplex
10 Gigabit Ethernet	X300	50	Full Duplex

At the beginning of the dissertation project, both USRPs were equipped with 1 Gigabit Ethernet interfaces and the host computer used had an Intel Core i5 processor and 8 Gb RAM. To improve the sample rate the USRP X300 was equipped with a 10

Gigabit Ethernet interface, a host computer with an Intel Core i7 6700 processor and 8 Gb RAM equipped with a 10 Gigabit Ethernet was used as host. The host computer was running MATLAB® R2018b and uses UHD 3.11.1.0-vendor. All the experimental data reported in this dissertation was obtained with the last experimental conditions. Fig. 10 shows the experimental setup used in the following sections.

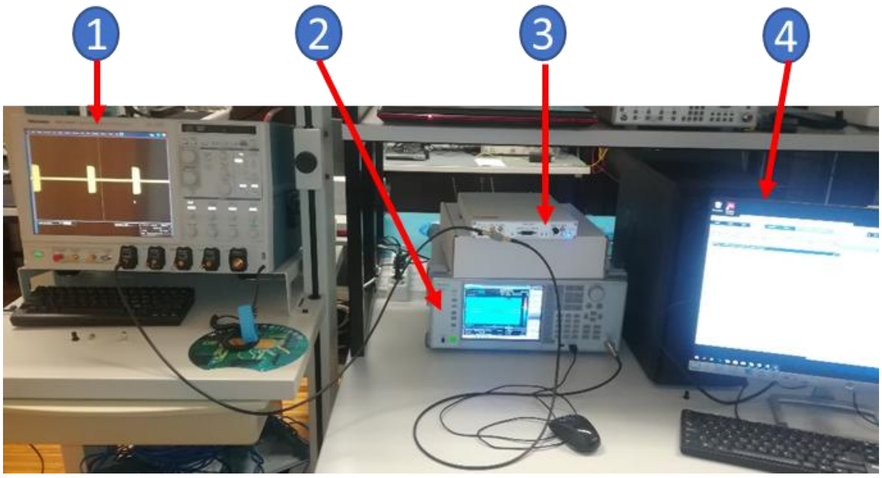

Figure 10- Experimental setup. 1) Oscilloscope, 2) Spectrum analyzer, 3) USRP, 4) Host computer.

4.3. Bandwidth bottleneck assessment in the transmitter path

To assess the bandwidth bottleneck in the transmission path the USRP X300 was programmed to send continuously a radar FMCW signal. The USRP X300 TX/RX port was used for transmission and the transmitted RF signal was measured by the Tektronix DPO70404. A FMCW signal was generated in MATLAB® with a bandwidth of 25 MHz, sample rate 50 MS/s and seep time of 0.5 ms. Since the Master Clock Rate (MCR) of the USRP X300 is 200 MHz the DAC operates at 200 MS/s, the transmitted FMCW samples needs to be up-sampled by a factor (*InterpolationFactor*), this operation is accomplished by the USRP DUC block. The valid interpolation factors are between 1 and 1024. For the MCR of 200.0 MHz, the achievable sampling rates are 200.0, 100.0, 50.0, 33.33, 25.0, 20.0, 16.67, 14.286 MS/s, ... 195.31 kS/s.

The connection between the host computer and the USRP transmitter was established by the MATLAB® object “comm.SDRuTransmitter”, that identifies the USRP platform X300, with the IP address “192.168.10.2”

```
radiotx = comm.SDRuTransmitter('Platform','X310','IPAddress','192.168.10.2')
```

This object enables the setting of several hardware parameters of the USRP. For our application the most relevant ones are:

- Frequency of operation: `radiotx. CenterFrequency=1.8 GHz, Gain`
- Overall gain of the USRP® hardware transmitter data path: `radiotx.Gain=20 dB`
- Upsample: `radiotx.InterpolationFactor=4; %This parameter can be adjusted`
- Test if packets are lost: `radiotx. UnderrunOutputPort=1;`

The underrunOutputPort set to “1”, when data is transmitted using the “step” method indicates that data is lost. This output allows the verification of real-time operation of the system. When transmitting, the USRP consumes samples at a constant rate. Underruns occurs when the host host computer does not produce data fast enough. The FMCW signal sent to the USRP consists of a data array (*FMCWData*) with 12500 complex samples. This data was continuously transmitted 10000 times using a *for* cycle. The number of lost frames was calculated as the average value over 10 runs of the for cycle, as summarized in Table 3.

```
for iavg=1:10;
for iFrame = 1:10000;
        underrun=step(radioTx,FMCWdata);
        if (underrun);
              Framelost=[Framelost iFrame];
              msg=['frame# lost',int2str(iFrame)]
        end;
end;
iFramelost(i)=length(Framelost);
end;
Framelost_avg=mean(iFramelost);
```

Table 3- Percentage of lost frames for a given interpolation factor.

Interpolation Factor	4	8	16	32
% Lost Frames	88.21	61.32	18.42	0

From this test it was verified that until interpolation factor of 16, which corresponds to a real time sampling rate of 200/16 MS/s= 12.5 MS/s, there were some underruns that might compromise real time operation. However, measuring the RF signal at the oscilloscope, even with an interpolation factor of 4, as shown in Fig. 11 the FWCW radar signal is continuously transmitted, with gaps between the different runs that we interpret as lack of data in the transmission buffer. The time intervals between transmissions is not constant and varies randomly, therefore we concluded that the latency of the system is unpredictable.

It is interesting to note that the signal amplitude is frequency dependent, higher frequencies lead to smaller signal amplitude.

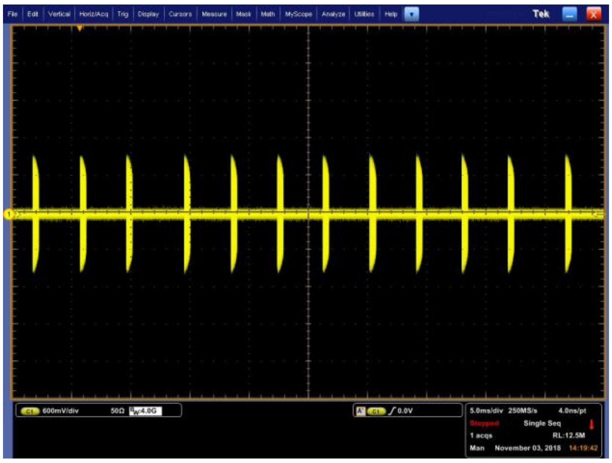

Figure 11- Oscilloscope trace, with interpolation factor of 4.

4.4.Bandwidth bottleneck in the receiver path

One of the major drawbacks of USRPs is bottleneck in the receive path, where the sample rate of ADCs is in most of times lower than the sample rate supported by the interface as we can see from Table 2. For the X300, it is suggested by the vendor that the problem is solved with 10GbE interface to match the sample rate of ADCs to host computer, however this is not the case.

To assess bandwidth limitations in the receiver a connection between the host computer and the USRP transmitter was established by the MATLAB® object "comm.SDRuReceiver".

```
radioRx = comm.SDRuReceiver ('Platform','X300','IPAddress','192.168.10.2')
```

The object parameters identify the USRP platform X300, with the IP address "192.168.10.2". This object enables the setting of several hardware parameters of the USRP. For our application the most relevant ones are:

- Frequency of operation 1.8 GHz: `radioRx. CenterFrequency=1.8 GHz, Gain`
- Overall gain of the USRP® hardware receiver data path: `radioRx.Gain=20 dB`

- Upsample: `radioRx.InterpolationFactor=4; %This parameter can be adjusted`
- Number of samples to read: `radioRx.Framelength=6000`
- `[RxData, len] = radioRx();`

The RxData contains the data and the “len” flag indicates if the data read data is valid. When trying to read data from the object “radioRx”, it is possible that the host has not yet received any data from the USRP® hardware.

A cycle *for* was used to step continuously the “radioRx” object 10000 times. In each “for” cycle it is issued a command to send data from the host computer to the USRP followed by a command to read data from the host computer to the USRP. The number of invalid frames for different interpolation factors was measured. The percentage of invalid frames was calculated averaging over 10 runs. The test was performed for only one receiving channel active and with the two receiving channels active.

Table 4- Percentage of lost transmitted frames and valid received frames for a given interpolation factor when 1 and 2 receiving channels are active.

Interpolation Factor	4	8	16
(%of lost transmitted frames) /(% of valid received frames (1 channel))	96.5/50.8	96.1/64.6	95.1/70.5
(%of lost transmitted frames) /(% of valid received frames (2 channels))	96.6/44.5	96.4/52.8	95.7/63.1

From the results presented in Table 4 we can see that real time operation is not possible even with an interpolation factor of 16, which corresponds to a sample rate of 12.5 MS/s, which corresponds to an achievable bandwidth of 7.5 MHz.

At this point we note that the bottleneck on the transmitter side is not a major problem, since the radar signal that is transmitted is always the same and even if some frames are lost there is continuous transmission. Although, we notice that radar chirped signals are not transmitted at a constant rate. In the receiver side, some frames are lost.

4.5. Oscillator stability

According to Fig. 8 and Fig.12 there are two RF local oscillators, one for transmission and other for reception. These RF oscillators are generated by a VCO and a Phase Looked Loop (PLL). Although, not shown in the schematic the USRP X300 also provides a clock signal to the daughterboards which is used as the reference clock for the frequency synthesizers and other components that require clocks. Fig. 12 shows the oscilloscope trace of the two RF outputs, with the RF output set to 1.8 GHz. It visible that the two RF signal have the same frequency but there is a constant phase shift between them. This phase shift is kept constant during operation and when the USRP is switched ‘off’ and ‘on’ and therefore can be compensated.

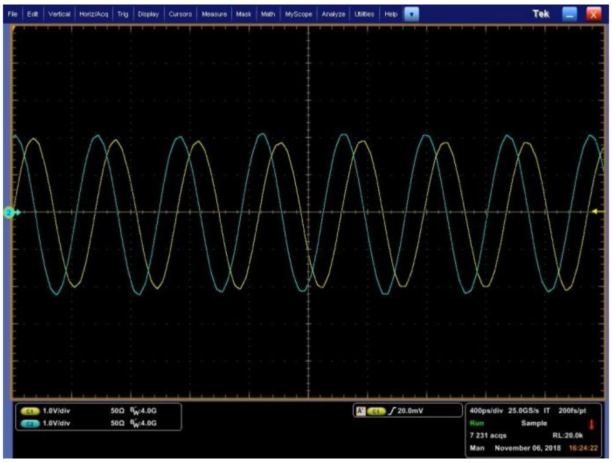

Figure 12- Oscilloscope trace of the two RF outputs.

In each “for” cycle the oscillator takes some time to be tuned. This settling time depends on the RF board that is used. Fig. 13 shows the measured transmitted signal spectrogram measured in the spectrum analyzer, where it is visible a small transient at the beginning of each signal. In Fig. 14 where a single chirped pulse is shown, the setting time is visible and is measured to be 29.495 μs. In order to avoid frequency oscillation at the beginning of each pulse the oscillator should already be running continuously. Therefore, the radar pulses should be generated after the settling time of the LO. In Fig.15 it is shown a spectrogram of a radar waveform with smooth frequency transitions.

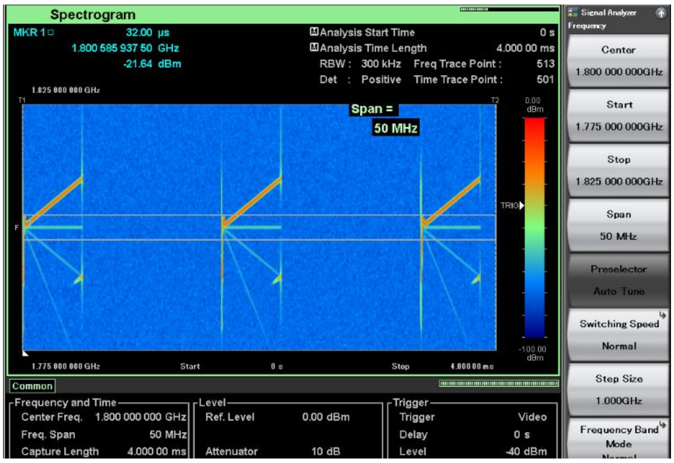

Figure 13- Measured spectrogram with interpolation factor of 4.

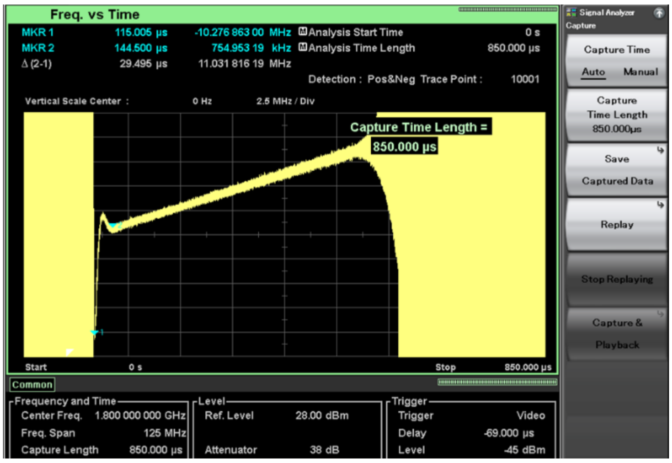

Figure 14- Frequency versus time for a single chirped pulse.

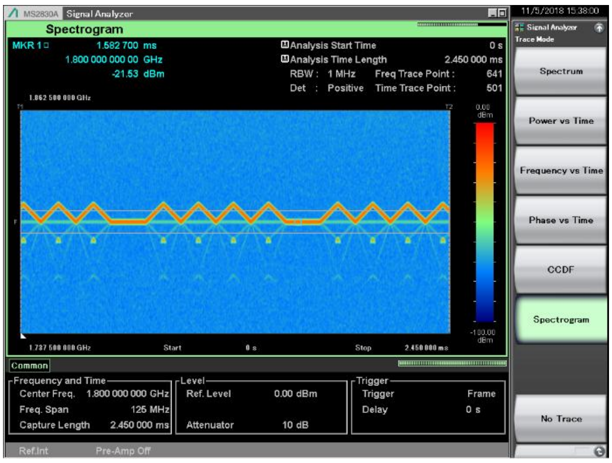

Figure 15- Measured radar waveform spectrogram showing smooth frequency transitions.

Another drawback of the system is its occasional random frequency instability as is shown in Fig. 16. These instable operations occurred in our experimental setup after operation over a long-time period, usually longer than 30 minutes. Under normal operation these instable pulses should be discharged.

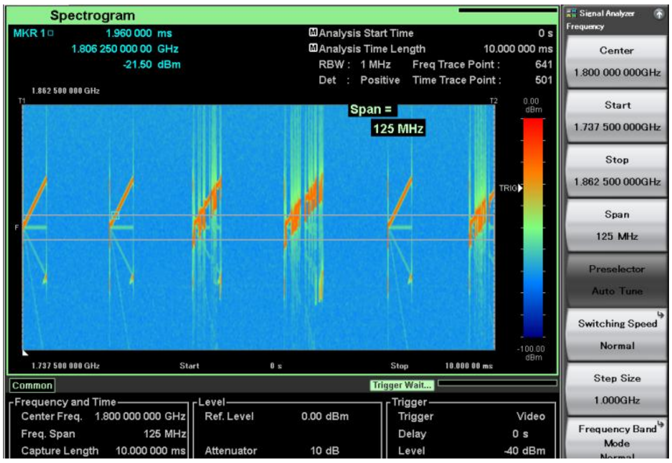

Figure 16- Frequency instability.

4.6. MATLAB® versus GNU radio

MATLAB® has lower interface speed in comparison to GNU Radio when using 1GbE interface, which affects radar range resolution. In fact, with GNU Radio it's achieved a sample rate of 25MS/s with the USRP NI2920 and is achieved 50MS/s for X300 using 1GbE interface too, and that can be increased to 200MS/s using 10GbE interface SFP+. Using MATLAB®, sample rate imposed for NI2920 is 2MS/s using 1GbE and 50MS/s for X300 with 10GbE. Interface 1GbE was set with MTU of size 1500bytes, and 9000bytes for 10GbE, recommended by Ettus Research. PRI was decreased to 1ms since receiver buffer is limited when using this architecture with MATLAB®, and the test parameters are adapted for maximum sample rate achievable. The major advantage of MATLAB® is its signal processing capabilities.

CHAPTER 5 – SDRadar implementation

The first part of this chapter presents the implementation of a SDRadar as suggested in [17]. It is demonstrated that, mainly due to the unpredictable latency of the USRP platform used, this implementation is not feasible. The second part of the chapter, discusses the implementation of a SDRadar that exploits the multiple input multiple output (MIMO) synchronization capabilities of the USRP X300. The chapter ends with the performance assessment of the SDRadar in terms of range profiling.

5.1 USRP X300 initial SDRadar implementation.

Following [17], a loopback system was established by using a loopback connection between the transmitting chain TX1 in RF0 with the receiving chain RX2 in RF1. A 30 dB attenuator (MiniCirc 15542 VAT-30 30 dB [23]) was used in the loop to limit the RF power reaching the RX2. In order to avoid extra delays introduced by MATLAB®, GNU radio was used. In Fig. 17 is represented GNU Radio functional diagram used in this experiment. Basically, this program sends continuously a FMCW signal to the USRP that is transmitted through the TX1 RF0, simultaneously the down-converted signal received by the RX2 RF1 is acquired.

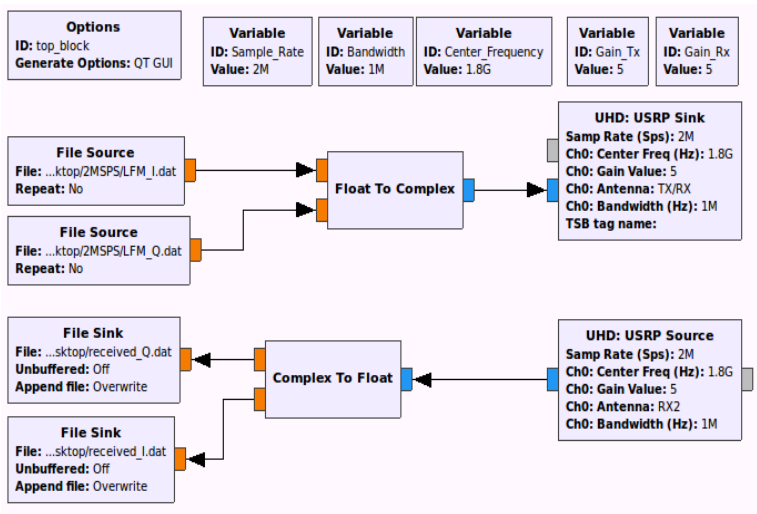

Figure 17- Functional diagram of radar system of experimental setup in GNU Radio.

Although, it was verified inconsistent delays between transmission and acquisition, during system operation, which compromise target range determination. Fig. 11, suggest it, and to prove that real time processing is not possible. The illustrated tests was done. Firstly, the system was powered off to erase the USRP buffers, then data is transmitted and received, to verify if there is some relation between delays and buffer states, as a limitation of hardware. In the second test, during transmission system was interrupted, to verify if the connection to the host computer compromise the delay between pulses. This process was repeated three times for both tests. In-phase received components can be seen, respectively, for both tests, in Fig.18 and Fig.19.

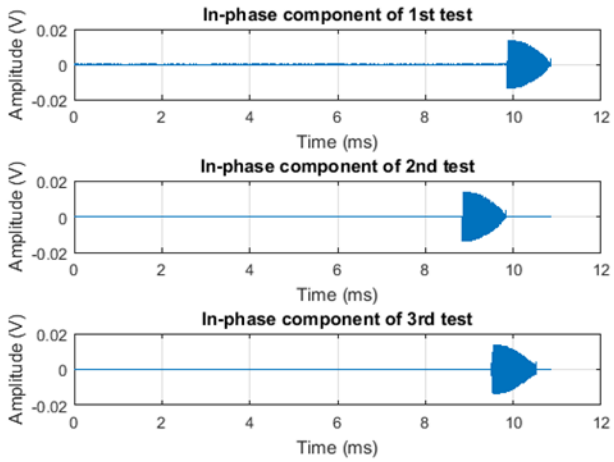

Figure 18- Inconstant time delay when system is powered off.

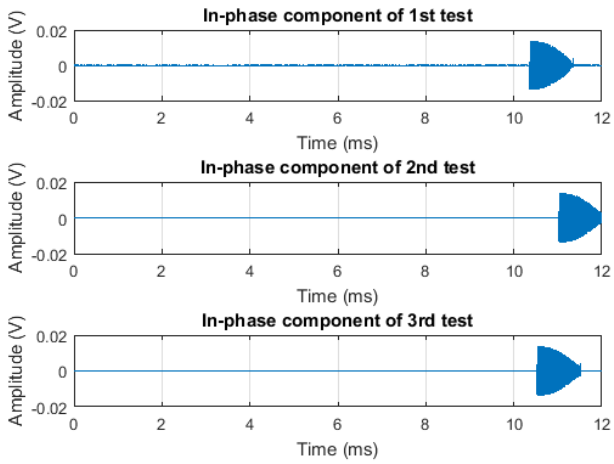

Figure 19- Inconstant time delay when system is interrupted.

From Fig. 18 and Fig. 19 we conclude that the delay introduced by the system, both hardware and software, is not constant, it changes from run to run. This fact means that's, is not possible assure the exact time when USRP starts to send and receive signals, despite of both channels be set at the same time. It depends on USRP buffer states, proved by the test when USRP is powered off, and host availability, proved when system is interrupted.

We can also observe that the amplitude of signal is frequency dependent, and this also applies to the transmitted signals (in the output of USRP). To solve that, and following recommendations of Ettus Research, operating frequency (f_o) should be adapted to sample rate (f_s), i.e f_o=Nf_s, with N an integer. After that adjustment the results did not improve. At this point, we conclude that it is impossible to implement software defined radar using this setup as suggested in [17] because of the unpredictable latency of the system.

5.2 SDRadar implementation using USRP X300 MIMO capabilities

An important characteristic of the USRP X300 is that its two channels can be synchronized. The two transmitters can transmit data simultaneously and synchronously and the same applies for the receivers. This capability enables the USRP X300 to be used to implement MIMO communication systems. It is important to note that transmission channels and reception channels are not synchronized.

Our approach is based on the following:

1) All the transmitter and receiver channels share the same clock and oscillator reference.

2) One of the transmitter channels, transmits continuously the FMCW radar signal.

3) The two receiving channels, receive simultaneously and in phase.

4) One of the receiving channels is used as a reference. It is connected to the transmitter channel in a loopback configuration.

5) The other receiving channel is used to receive a reflected signal from the target.

Fig. 20 shows the schematic diagram of the proposed experimental setup.

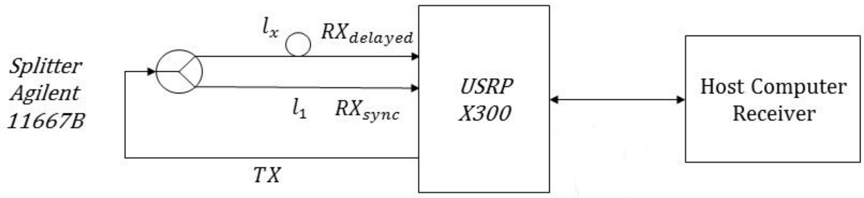

Figure 20- Experimental setup using one receiving channel as reference.

In this setup the host computer running MATLAB®, sends FMCW baseband signal to X300, which transmits the signal. Next, the signal is split into two different paths, and received in channel RX_{sync}, that is the reference channel for all received pulses, and received in channel $RX_{delayed}$, that delays signal, acting as reflected signal from target.

5.3 MATLAB® MIMO capabilities and SDRadar system calibration

For a transceiver to be considered MIMO-capable, each channel in the system must meet two basic requirements. The sample clocks must be synchronized and aligned and DSP operations must be performed on samples aligned in time, i.e. from the same sample clock edge. In addition to sample time and sample clock alignment, for radar applications, the system must maintain a known phase relationship between the reference channel and the measuring channel.

The following MATLAB® code enables the use both channels USRP X300 receiver channels.

```
radioRx.ChannelMapping=[1 2];
```

When the object "radioRx" is called the output is a matrix "rxSig", with two columns with complex samples that correspond to the received signal on channel1 and channel2.

```
[rxSig, len] = radioRx();
```

The variable ‘len’, indicates if the data if valid. The following code was used to send 10000 FMCW signals and toand receive and data on both USRP channels.

```
iFrame=1
for iFrame = 1:10000;
        underrun=step(radioTx,FWCW);% transmits FWCM signal
         if (underrun); %verifies if the frames was effectively
         transmitted
            Framelost=[Framelost iFrame];
                msg=['frame# lost',int2str(iFrame)]
              end;
        [rxSig, len ] = radioRx();
       if len > 0; %Verifies if the received data is valid
            RrxSig1(:,iFrame_valid)=rxSig(:,1);%Saves the received
of channel 1
             RrxSig2(:,iFrame_valid)=rxSig(:,2);%    );%Saves    the
received of channel 1

        iFrame_valid=iFrame_valid+1;
       end
 end
```

Because there is no synchronization between transmission and reception, the number of points to be received is set to be three times the number of points to be transmitted in this way we are sure to receive a complete transmitted frame. However, not all the received frames are acceptable for signal processing, several situations of invalid data can occur has is illustrated in Fig.21b.

Fig. 21a illustrates a valid frame. Fig 21b shows invalid data since the FMCW data samples were not all received. Fig. 21c shows invalid data, not any FMCW signal was being transmitted. Fig. 21d shows a situation when the FMCW signal was not transmitted continuously.

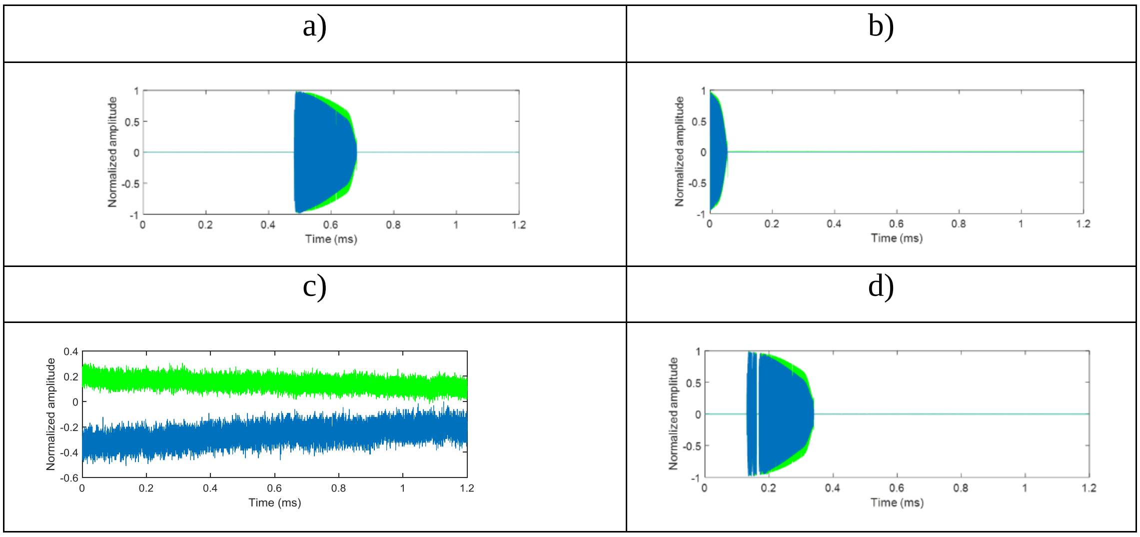

Figure 21- Example of received traces. Green curve is the channel 1 and blue curves correspond to channel 2. a) valid data. b), c) and d) invalid received data.

Due to phase ambiguities of the hardware, in a configuration with 15 cm cables connecting both the $RX_{delayed}$ and RX_{sync} inputs, there is some phase dealignment between the two inputs, as is shown in Fig. 22. However, the delay between the two received is very small comparing with the resolution of our system that it can be neglected. It is also important that this delay is kept constant even after the USRP is switched off and on.

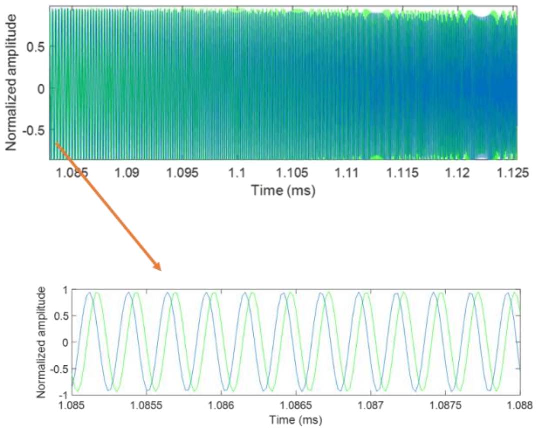

Figure 22- Received traces for 15 cm cables connecting both the $RX_{delayed}$, and RX_{sync}, inputs . Green curve is the channel 1 and blue curves correspond to channel 2.

5. 4 SDRadar signal processing

After receiving reference RX_{sync} and target $RX_{delayed}$ signals, signal processing is applied based on schematic present in Fig. 23.

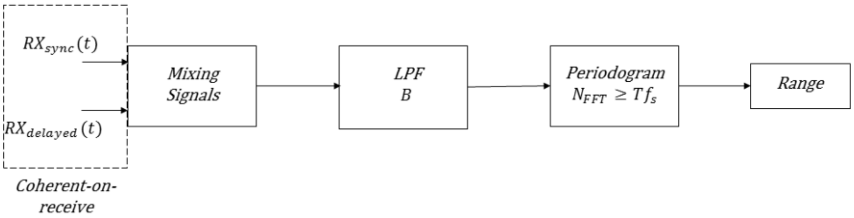

Figure 23- Signal processing of second approach.

This signal processing architecture prevents failures of synchronism and phase deviation, since phase of received pulses is mixed with correspondents transmitted pulses, which corresponds to coherent-on-receive processing. After mixing signals and apply low pass filtering, periodogram is done to get beat frequency associated to the target, and then range is taken basis on it.

5. 5 SDRadar parameters

Table 5 summarizes radar parameters used in the performance evaluation.

Table 5- Characteristics of designed radar used in the experimental setup.

Type	FMCW-LFM
Operating frequency	2 GHz (L-Band)
Sweep Time	0.5 ms
Bandwidth	10 MHz
PRI	0.5 ms
Range resolution	15 m
Minimum detectable range	15 m
Maximum range	75 km
Maximum unambiguous range	37.5 Km
Transmitted Power	1.0498 mW (0.2109 dBm)
Power sensitivity of reference channel	-98.67 dBm
Power sensitivity of fictitious target channel	-98.91 dBm
Maximum achievable range	35.9318 m for RCS of 0.5 m^2

5. 6 SDRadar performance evaluation

As in the above experiments, we start by verifying the synchronism of system. The reason of 2 GHz instead of 1.8 GHz is a better performance on signal phase. Next, the length of cables of delayed channel is tuned according to 4 lengths. Table 6 summarizes parameters used in test.

Table 6- Parameters of experimental setup using a fictitious target.

Transmitter Sample Rate	20 MS/s
Transmitter gain	10 dB
Receiver gain	10 dB
Reference signal, l_1	1 m
Fictitious target, l_1, l_2, l_3, l_4	1, 15, 30, 45 m

i. Test using $l_1 = l_2$:

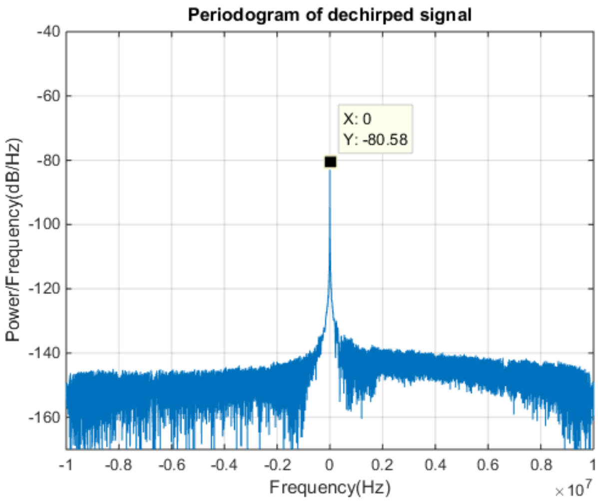

Figure 24- Periodogram of dechirped signal for $l_1 = l_2$.

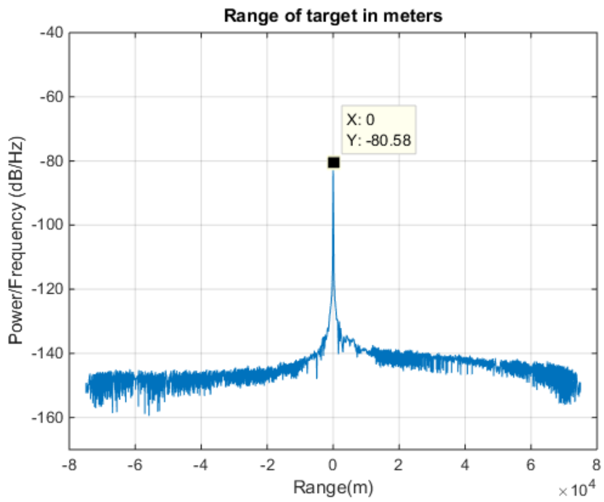

Figure 25- Range of target in meters for $l_1 = l_2$.

From periodogram of *dechirped* signal in Fig.21 we can see that system is synchronized, since the corresponding beat frequency is 0 Hz which corresponds to 0 m. Despite the length of the cables, both cables the one used for synchronism and fictitious target (delayed channel) have 1m. However, since the minimum detectable distance is 6 m it is not possible to resolve this cable length.

ii. Test using $l_2 > l_1$:

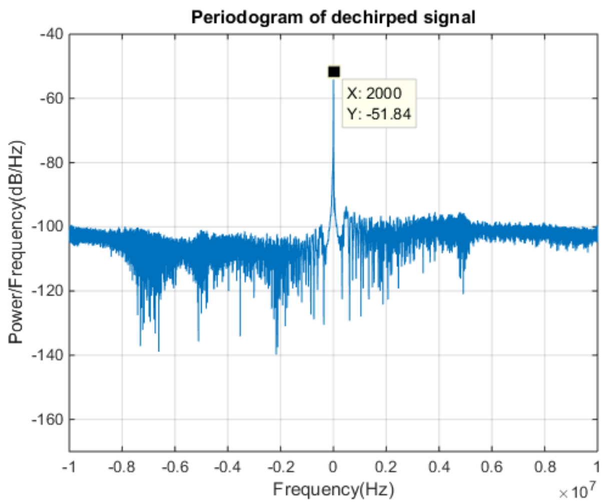

Figure 26- Periodogram of dechirped signal for $l_2 > l_1$.

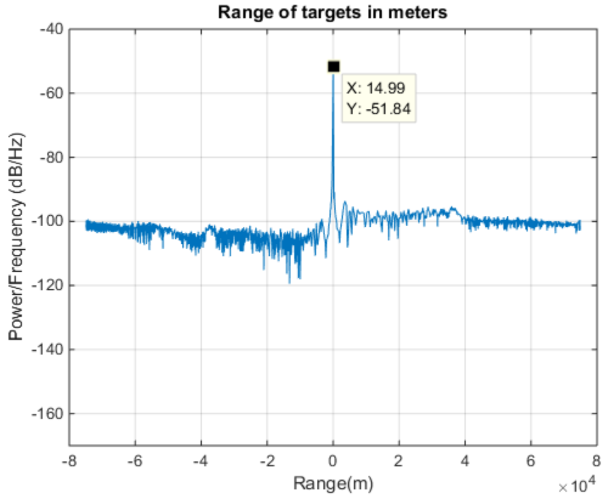

Figure 27- Range of target in meters for $l_2 > l_1$.

In this test, the cable corresponding to fictitious target has 15m. Analyzing periodogram we detect a beat frequency of 2 kHz which corresponds to 14.99m, concluding that the system has a certain margin of error. Although, we conclude again that system is operating correctly.

iii. Test using $l_3 > l_1$:

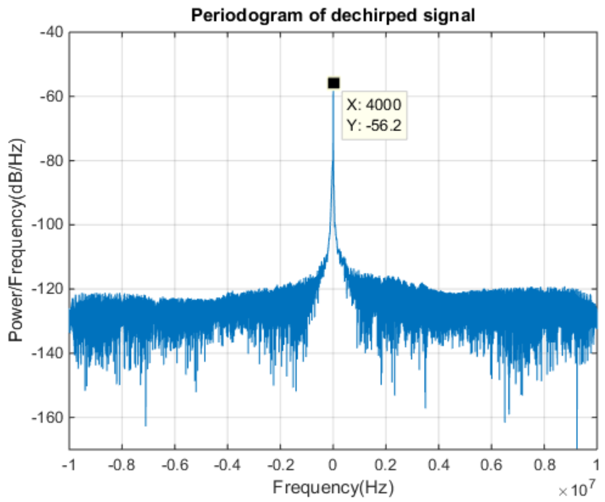

Figure 28- Periodogram of dechirped signal for $l_3 > l_1$.

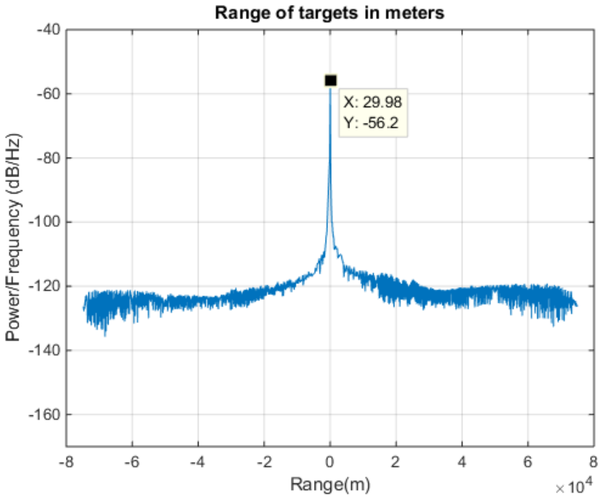

Figure 29- Range of target in meters for $l_3 > l_1$.

iv. Test using $l_4 > l_1$:

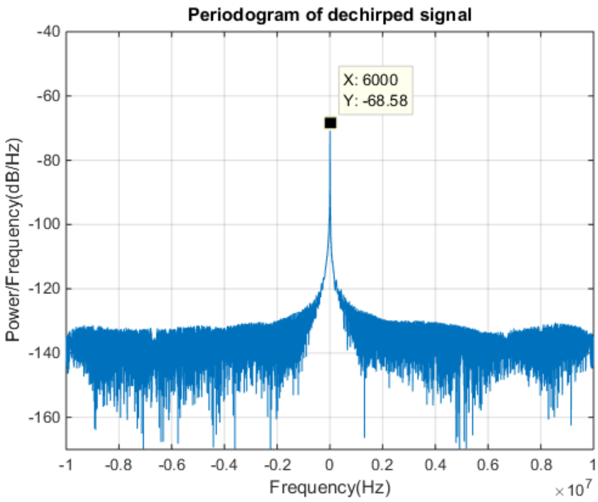

Figure 30- Periodogram of dechirped signal for $l_4 > l_1$.

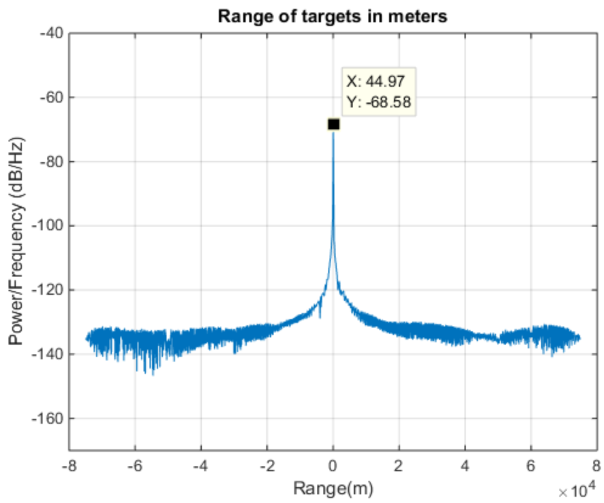

Figure 31- Range of target in meters for $l_4 > l_1$.

For tests iii and iv we conclude the same as for test ii. In test iii where was used a length of 30m for fictitious target (delayed channel), corresponding to a beat frequency of 4 kHz which corresponds to a 29.98m. The same way, in test iv for a length 45m again for fictitious target, corresponding beat frequency is 6 kHz and a distance of 44.97m.

This experimental setup proves the capacity of SDR to implement radar system, using the copy of transmitted pulses to overcome phase deviation related with unpredictable latency of the system, as well as, and most important time synchronization.

CHAPTER 6 – Conclusions

In this work, the implementation of an FMCW radar system operating in L-Band over SDR was studied and evaluated. The parameters of signal modulation were changed to adapt to the system, such as, sample rate, bandwidth, sweep time and operating frequency. To accomplish that GNU Radio and MATLAB® software's capabilities were evaluated at each experiment to improve the performance and to test the ability of SDR to implement system, at the same time, signal processing techniques such as coherent detection were applied.

Both host computer and USRP are relevant components on this system. As Chapter 3 suggests, the communication between these two components is a limiting factor to achieve a high-resolution radar system. Presented in Chapter 4, characteristics of USRP such as sample rate of ADC's and maximum speed interface connection impose limitations on this resolution, proved by experimental results in the same chapter. Sample rate was adapted to the maximum achievable speed interface connection, and it was verified that the real speed interface is lower than expected, because of host availability and buffers state, which affects general performance of system.

It was concluded that deviation on beat frequency is in fact phase deviation and is caused by unpredictable latency response of RF Module of USRP. To solve the problem, coherent-on-receive detection was applied as described in section 2.3.4 on Chapter 2. It was concluded too that there is some leakage between transmitter and receiver paths, for that reason, before start use the system, should be done verification on hardware of USRP to prevent touch or proximity of cables of each channel, and the gain of each must be adapted empirically.

After accounting all these restrictions, a final radar system capable of detect static targets have been reached, with a bandwidth of 10 MHz and range resolution of 15 m, operating at 2 GHz, by using MATLAB® software and USRP X300.

Contrary referred in [17], it was not obtained the same results, suggesting that described implementation is not straightforward implemented in general purpose SDR platforms. Although, it contributes to explore the capacities of GNU Radio to implement SDRadar, and as the same way [19], where description of SDR limitations help to improve the desire system. Finally, [3] shows and helps to confirm the validity of obtained results, the approximation of system architecture was an attempt to understand the implications of each parameter in system.

After studying the synchronization limitation of the USRP X300, we realized that the system, in its present hardware and software version cannot be used to implement a radar. Recently Ettus has new RF cards that share the same local oscillator for the RF transmitter and receiver chains. This hardware will help the implementation of a radar system, however, latency of the system is as well a major issue, to cope with that, it is necessary to understand and quantify the latency of the several functional blocks of the system. However, the objective of this dissertation is to use the USRP a high level, taking profit of its capabilities and avoiding its limitations.

Following this work, since is implemented static target detection and is proved the correct operation, we suggest firstly the implementation with dedicated directive antennas and high gain, and after that, implement radar detection for non-stationary targets, and of course, with different waveform to accomplish that.

REFERENCES

[1] [Online] Available: https://en.wikipedia.org/wiki/Radar

[Accessed: 28-Ago-2018].

[2] [Online] Available: https://en.wikipedia.org/wiki/Imaging_radar

[Accessed: 28-Ago-2018].

[3] S. Costanzo, F. Spadafora, A. Borgia, O. H. Moreno, A. Costanzo, and G. di Massa, “High resolution software defined radar system for target detection,” *Advances in Intelligent Systems and Computing*, vol.206, pp. 997–1005, 2013.

[4] Debatty, T., “Software defined RADAR a state of the art,” *Proc. of 2nd International Workshop on Cognitive Information Processing (CIP)*, 253–257, Bruxelles, 2010.

[5] Software Defined Radio Forum.

[6] M. Schneider, “Automotive radar—Status and trends,” in Proc. IEEE German Microw. Conf. (GeMiC), pp. 144–147, April 2005.

[7] B.R. Mahafza, Radar systems analysis and design using MATLAB®, 3rd ed. Huntsville, Alabama, USA: CRC Press, 2013, pp. 3-141.

[8] K. Barlee, D. Atkinson, R.W. Stewart, L. Crockett, “Software Defined Radio Using MATLAB® & Simulink and the RTL-SDR”, Strathclyde Academic Media, 2015.

[9] Y. Wang et. al., “A Ku-band 260mW FMCW synthetic aperture radar TRX with 1.48GHz BW in 65nm CMOS for micro-UAVs,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2016, pp. 240–241.

[10] Y. Wang and Y. Zheng, "An FMCW Radar Transceiver Chip for Object Positioning and Human Limb Motion Detection," in *IEEE Sensors Journal*, vol. 17, no. 2, pp. 236-237, 15 Jan.15, 2017.

[11] Y.-K.Kwag, J.-S. Jung, In-S. Woo, M.-S. Park, “ Modern Software Defined Radar (SDR) Technology and Its Trends”, *Journal of Electromagnetic Engineering and*

Science, vol. 14, no. 4, pp. 321-328, Dec. 2014. [Online] Available: http://koreascience.or.kr/article/ArticleFullRecord.jsp?cn=E1ELAT_2014_v14n4_321 [Accessed: 22-Fev-2018].

[12] T.Ulversoy, "Software Defined Radio: Challenges and Opportunities", *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531-550, May 2010.

[13] [Online]Available:
https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates

[Accessed: 20-Fev-2018].

[14] D. Sinha, A. Verma, S. Kumar, "Software Defined Radio: Operation, Challenges and Possible Solutions", in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Jan. 2016.

[15] C. Rossler, E. Ertin, R. Moses, "A software defined radar system for joint

communication and sensing", in *2011 IEEE RadarCon (RADAR)*, Jul. 2011.

[16] [Online] Available: https://kb.ettus.com/UHD

[Accessed: 20-Jun-2018].

[17] A. Prabaswara, A. Munir, A. Suksmono, "GNU Radio Based Software-Defined FMCW Radar for Weather Surveillance Application" in *2011 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Oct. 2011.

[18] [Online] Description available: https://www.ettus.com/product/details/UN210-KIT. [Accessed: 4-Apr-2018].

[19] T.Mathumo, T.Swart, R. Focke, "Implementation of a GNU Radio and Python FMCW Radar Toolkit", in *2017 IEEE AFRICON*, Sept. 2017.

[20] [Online] Description available: https://www.ettus.com/product/details/UB210-KIT. [Accessed: 4-Apr-2018].

[21] [Online] Description available: https://www.ettus.com/product/details/X300-KIT. [Accessed: 16-Apr-2018].

[22] [Online] Description available: http://www.ni.com/pt-pt/support/model.usrp-2920.html. [Accessed: 27-Fev-2018].

[23] [Online] Description available: https://www.minicircuits.com/WebStore/dashboard.html?model=VAT-30%2B

[24] [Online] Available: https://github.com/gnuradio/pybombs [Accessed: 26-Mar-2018].

APPENDIX

Installation of GNU Radio based on GitHub repositorium and forums in [24]

Install git:

$sudo apt-get install git

Assure that there is a version of cmake:

$sudo apt-get install cmake

Install a compatible version of cmake for GNURadio (was done in 29/03/2018):

$sudo apt-get purge cmake

$version=3.10

$build=3

$mkdir ~/temp

$cd ~/temp

$wget https://cmake.org/files/v$version/cmake-$version.$build.tar.gz

$tar -xzvf cmake-$version.$build.tar.gz

$cd cmake-$version.$build/

$./bootstrap

$make -j4

$sudo make install

$cmake –version

Install boost lib:

$sudo apt-get install libboost-all-dev

Install pip:

$sudo apt-get update && sudo apt-get -y upgrade

$sudo apt-get install python-pip

$pip -V

$sudo -H pip install --upgrade pip

Install pybombs:

$sudo pip install pybombs

$pybombs auto-config

Add recipes for pybombs, must have to choose the pretended recipes:

$pybombs recipes add-defaults

$pybombs recipes add gr-recipes git+https://github.com/gnuradio/gr-recipes.git

$pybombs recipes add ettus-pybombs git+https://github.com/EttusResearch/ettus-$pybombs.git

Install gnuradio and gr-osmosdr:

$pybombs prefix init ~/prefix -a myprefix -R gnuradio-default

Run gnuradio with pybombs assuring that all libraries are inicialized:

$pybombs run gnuradio-companion

Lib share update:

$sudo ldconfig

Configure GNU Radio IQ correction:

$sudo pybombs install gr-iqbal

Run GNU Radio:

$sudo pybombs run gnuradio-companion