Distance to Object Estimation Based on Software Defined Radio USRP using Python

Diego Armando Mora-Huaman
Escuela Profesional de Ingeniería Electrónica
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú
110294@unsaac.edu.pe

Facundo Palomino Quispe
Escuela Profesional de Ingeniería Electrónica
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú
facundo.palomino@unsaac.edu.pe

Roger Jesus Coaquira-Castillo
Escuela Profesional de Ingeniería Electrónica
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú
roger.coaquira@unsaac.edu.pe

Mark Clemente-Arenas
Grupo de Circuitos y Sistemas Electronicos de Alta Frecuencia ECS-HF
Universidad Nacional Tecnológica de Lima Sur UNTELS
Lima, Peru
mclemente@untels.edu.pe

***Abstract*—The use of a Software Defined Radio System SDR is widely used in several critical applications such as wireless communications, airborne, satellite, radar, etc. SDR through Universal Peripheral Radio Software Technology USRP, has allowed to reduce hardware complexity and cost. In this article, the implementation of a Frequency Modulated Continuous Wave FMCW radar is presented to estimate the distance between the transmitter and a target object at 3.3 GHz using open source hardware and software such as: GNU Radio, USRP B210 and Python. Raw signals processing from wall reflections was implemented in Python, and the distance estimation is presented on a spectrogram**

Keywords— radar FMCW, GNU Radio, Software Defined Radio.

I. Introduction

The increasing use of radars in different applications such as: ground penetrating radars [1], weather surveillance radars[2], and synthetic aperture radars require that radar systems be configurable for such applications by simply introducing some software configuration change. This feature is provided by software-defined radio systems allowing to reduce the amount of hardware that includes the large classic radar systems used around the world.

Continuous wave (CW) Radar systems transmit an electromagnetic signal and simultaneously receive scattered signals reflected from objects. CW radar systems are generally used in compact short-range applications. When a stationary object is illuminated, the frequency of the echo signal does not change relative to the transmitted wave. However, when an object is moving, then the frequency of the echo signal is altered due to the Doppler Effect. A disadvantage of CW radar systems is the inability to detect stationary objects or measure the distance between the transmitter and the target. These limitations can be overcome by modulating the transmitted signal, for example, the transmission frequency can be changed linearly with time. This type of system is known as frequency modulated continuous wave radar (FMCW) [3]. Then, this radar system is implemented using GNU Radio and two Vivaldi antennas for transmission and reception of linear FMCW signals. In [8], a implementation of a FMCW radar using GNU Radio and USRP is presented. It was able to detect up to two targets. However, in order to identify the target, long signal pre-processing and processing was required.

This document is organized in Four sections. After this introductory section, the second section presents theory related to Radar used for this implementation. Section three shows details of the design and implementation of the system. Section four deals with result presentations and discussions at the end.

II. Principles of Radar

A. FMCW Radar

According to [4], the linear FMCW signal is also known as the chirp signal, in simplified form, the transmitted signal shown in (1).

$$S_t = e^{j\pi k_r t^2} \tag{1}$$

Where k_r denotes the modulation rate, t_0 is half of the modulation period and B denotes the bandwidth of the chirp signal.

$$k_r = \frac{B}{2t_0} \tag{2}$$

The reflected signal is similar to the transmitted, but delayed by Δ_t as shown in Fig. 1

$$S_t = e^{j\pi k_r (t-\Delta_t)^2} \tag{3}$$

The delay time Δ_t, in (4) between the transmitted signal and the reflected signal, at some instant in time, is proportional to the distance to an object.

$$\Delta_t = \frac{2R}{c} \tag{4}$$

Where: R denotes the distance between the transmitter and the target, c is the speed of light

Measuring the beat frequency f_b, allows us to determine the range to a target because it is directly related to the delay of the reflected signal [3]. Therefore, a relationship in (5) can be formed between the modulation bandwidth, the modulation period, the rate frequency and the delay time of the reflected signal.

The work in this paper has been funded by Cienciactiva-UNSAAC from contract N021 -2017-UNSAAC-Proyectos de investigación.

978-1-7281-9377-9/20/$31.00 ©2020 IEEE

$$\frac{f_b}{\Delta_t} = \frac{B}{t_0} \quad (5)$$

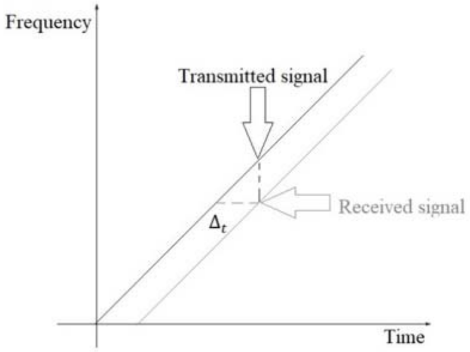

Fig. 1. The Relationship between Frequency and Time for a FMCW Radar System [3].

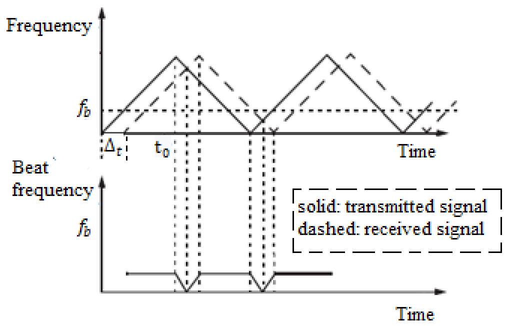

Fig. 2. Transmitted and received triangular Lineal FMCW signals and beat frequency for stationary target [5].

Replacing (4) in (5) we obtain (6):

$$f_b = \frac{4BR}{t_0 c} \quad (6)$$

An important parameter for the design of the Radar System is the range resolution Δ_R, defined in (7), that refers to the minimum distance between two objects that allow them to be distinguished as different object.

$$\Delta_R = \frac{c}{2B} \quad (7)$$

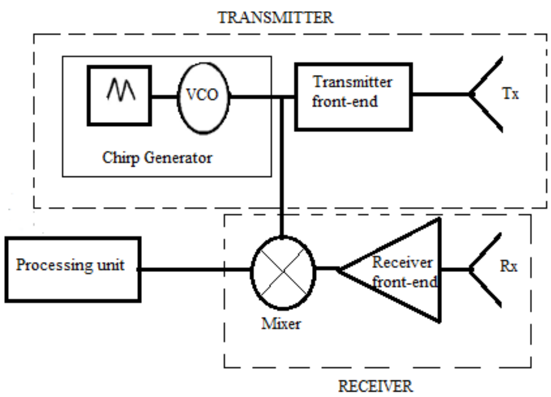

Fig. 3. Block Diagram of FMCW Radar [6].

The FMCW radar transmits a continuous signal that is modulated by a periodic signal that could be: ramp, sinusoidal and triangular. In this paper, a triangular signal is used. In Fig. 2 it is noticed that the triangular signal and the VCO (Voltage Controlled Oscillator) produce chirp signal, a signal that at the same time works as a signal source in the transmitter and as the receiver mixer.

B. Software Defined Radio (SDR)

The USRP B210 is a fully integrated Radio system, Universal Software Radio Peripheral (USRP), which is characterized by being a platform with continuous frequency coverage from 70 MHz to 6 GHz. It is a low-cost implementation, since it combines a direct conversion transceiver on a single spartan6 FPGA chip that provides up to 28 MHz of real-time bandwidth for radar applications. It is fully open-oriented, reprogrammable, and has a fast connection USB 3.0. It has full support for the USRP Hardware Driver ™ (UHD) software for better performance

USRP B210 presents two software configuration alternatives, through MATLAB and GNU RADIO. For the software configuration through MATLAB, there are packages developed by ETTUS that make communication between MATLAB and USRP B210 possible. However, MATLAB® is powerful software dependent on computational resources and requires USB 3.0 connectivity, becoming a disadvantage due to computational load. For reduce the computational load caused by the radar system SDR, GNU Radio is an option, because it has a set of tools free and open-source software, reducing the use of computational resources [7].

USRP B210 cards work through the USB 2.0 port, but the speed (480 Mbps equivalent to 8 MHz of maximum bandwidth of the FMCW signal) is lower compared to the USB 3.0 port, so it is recommended to use the USB 3.0 port for radar applications in SDR

III. Design and Implementation

A. Implementation

The implementation of the linear FMCW radar system used a PC, two vivaldi antennas, two 1 m cables with SMA adapters, the GNU Radio software and the USRP B210. The transmit and receive stage was implemented using the GNU Radio blocks as shown in Fig.4. and Fig. 5.

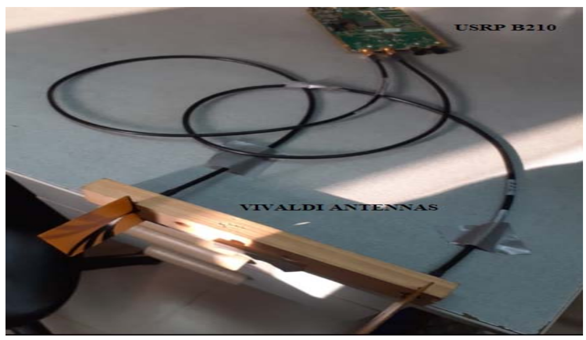

Fig. 4. USRP B210 and Vivaldi antennas for experiments

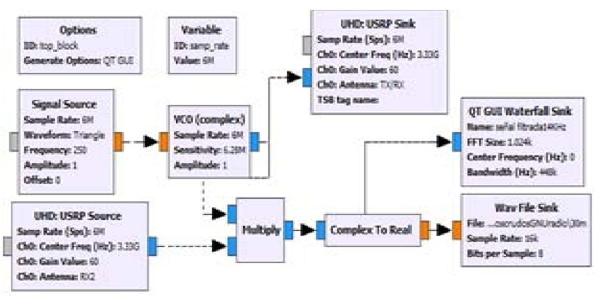

Fig. 5. GNU Radio implementation of transmission and reception

For the transmission stage, the "Signal Source" block is used, choosing as a waveform a triangular signal of 250 ms period. The output of the Signal Source block is connected to the "VCO (Complex)" block where the important parameter is the sensitivity which determines the bandwidth of the chirp signal. The sensitivity is equivalent to $2\pi B$. Also, for the transmission to the wireless medium through the Tx antenna connected to the USRP card, it is necessary to send a real and imaginary signal, for this process the "UHD: USRP Sink" block is used. In the block for transmission to the wireless medium, 3.33 GHz is set as "Center Freq", the options "Clock Rate", "Clock Source" and "Time Source", must be configured in case of having an external clock that is physically connected to the card, in this case the option is in "Default" because the internal clock of the card is being used, for the USRP B210 there are two channels RX and TX in this case only one channel is used to transmit, the gain is configured with a value of 60 dB.

On the other hand, for the implementation of the reception stage, at the beginning the "UHD: USRP Source" block is used, placing the same values as in the "UHD: USRP Sink" block. Later, the received signal is multiplied by the chirp signal using the "Multiply" block. Finally, the multiplied signal is stored in the "Wav File Sink" block.

The value of "Sample Rate" of all the transmission and reception blocks is fixed and its value depends on the USRP model available, it is not recommended to modify it by lower or higher values, according to this, the maximum number of samples that the card can process according to specifications.

B. Radar Processing

Raw radar signal is stored in an audio file with a 16KHz sampling rate whose pulse repetition interval, also known as pulse duration, is 20ms. The audio file has two channels: one is used to store a square and periodic signal, and the other is used to store radar echoes.

Radar signal processing is implemented in Python, using libraries such as: Matplotlib, Numpy and Scipy to process the backscattered echoes that are stored in the audio file.

The initial parameters for processing FMCW radar signals are shown in the following table.

TABLE I. FMCW RADAR PARAMETERS

Initial parameters	
Parameter	***Value***
PRI	20 ms
F_s	1600 KHz
F_0	3.33 GHz
B	28 MHz
Δ_R	5.3m

Where: PRI is the Pulse repetition interval or pulse duration. F_0 is the chirp signal frequency. Fs is the sampling rate and B chirp signal bandwidth.

The audio file is read with the scipy library, where the sampling rate and raw signal is captured. The number of samples is determined by multiplying the pulse duration and the sampling frequency, obtaining 320 samples in pulse repetition interval, which defines the matrix column.

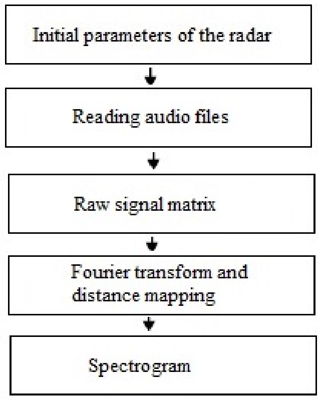

Fig. 6. Signal processing workflow.

Then, the 2292 row and 320 column "mat" matrix is constructed. The rows of the matrix are obtained from the sampling of the baseband signal "signal" in each logical 1 of the square signal "logic", and the average of 11 samples of the square signal must be equal to logical 0. When these conditions are true, then the samples are stored in the "m" row of the matrix as described in the flowchart of Fig. 6. In Python the command "insert" is used to add elements to the list and it is converted into an array with the command "numpy.append" using conditionals and loops as shown in Fig. 6.

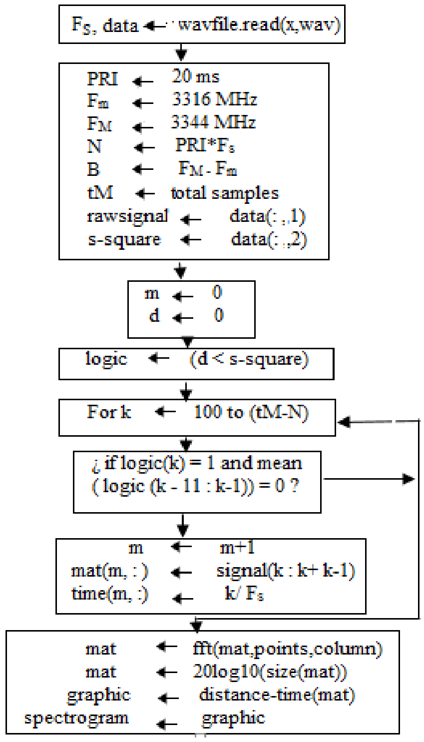

Fig. 7. Flow chart of the implemented algorithm.

The Fourier transform is applied to all columns of the matrix at N points. The spectrogram and graphs are implemented using the maplotlib library in Python. The distance is calibrated according to (8) and the number of points N of the fourier transform. The spectrogram shows the power of signals whose range dimensions are expressed in the frequency domain.

$$R = \frac{c f_b t_o}{4B} \tag{8}$$

IV. RESULTS

After the whole systems was implemented, final tests were carried out in a closed environment. The radar system was located 2 meters from the wall.

In Fig. 7 and Fig. 8, the spectrogram is observed whose yellow intensities represent the maximum value of the powers that are described by the color bar, where it indicates the location of the wall. The spectrogram shows that the radar system is located 2m from the wall.

The error in the calculation of distance is due to the resolution in range, this is due to the limitation of bandwidth of the USRP B210 card, for this it is suggested to develop an algorithm that takes care exclusively in the optimization of the resolution of the card.

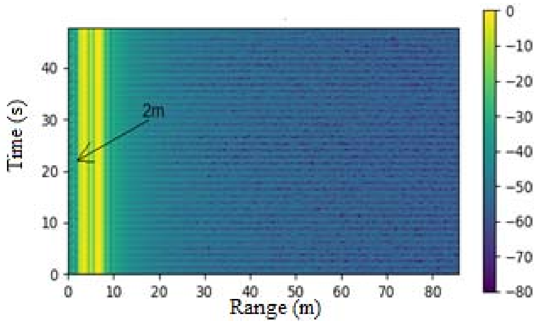

Fig. 8. Spectrogram of the radar signal plotted using Python implementation of the algorithm.

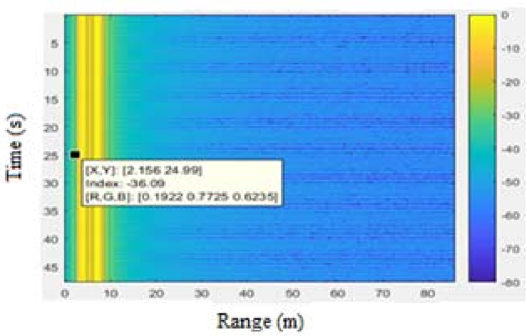

Fig. 9. Spectrogram of the radar signal plotted using Matlab implementation of the algorithm

The setting for this test is a place with a good amount of obstacles and signs of interference. The purpose of testing at this location is to see performance and possible processing errors. During the processing that was done in Matlab and Python you can see that there is similarity, although there is some difference in calculation time of the algorithm, being Matlab the fastest. In both programs it was possible to reduce the intensity of the signals coming from the obstacles that were between the radar and the wall. The proposed system presents the pre-processing in Python and Matlab and differs form [8], both programs present very similar results. The images obtained will allow, by means of signal processing, to identify the targets. Achieving a greater number of applications.

V. CONCLUSIONS

The use of SDR software defined radio systems is beneficial, due to the low cost and ease of configuration. For different applications the only thing that has to be modified is the center frequency in the "UHD: USRP Sink" and "UHD: USRP Source" block. The implemented system was able to detect objects up to 30 m distance. It was possible to detect a wall 1.9 m away from the radar with an error of 0.1 m.

VI. FUTURE WORK

As future work it is recommended to use a Raspberry Pi 4 instead of a PC, thus turning the whole static FMCW radar system into a portable system. Due to the weight that does not exceed 900 g, the entire system could be mounted on a drone, achieving the implementation of synthetic aperture radars, achieving greater applications, always taking into account the working frequency of the B210 card.

REFERENCES

[1] Liang, Huaqing, and Zhiqiang He. "Study on signal processing of FMCW ground penetrating radar." *2009 International Conference on Measuring Technology and Mechatronics Automation*. Vol. 1. IEEE, 2009.

[2] Prabaswara, Aditya, Achmad Munir, and Andriyan Bayu Suksmono. "GNU Radio based software-defined FMCW radar for weather surveillance application." *2011 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*. IEEE, 2011.

[3] Melvin, William L. and James A. Scheer. “Principles of modern radar : advanced techniques.” (2012).

[4] Hein, Achim. “Processing of Sar Data: Fundamentals, Signal Processing, Interferometry.” (2003).

[5] Mahafza, Bassem R.. “Radar Systems Analysis and Design Using MATLAB.” (2000).

[6] Charvat, Gregory L. et al. “The MIT IAP radar course: Build a small radar system capable of sensing range, Doppler, and synthetic aperture (SAR) imaging.” *2012 IEEE Radar Conference* (2012): 0138-0144.

[7] Ettus. [Online]. Available: https://www.ettus.com/all-products/UB210-KIT/. [Last access: 2 January 2020].

[8] S. Sundaresan, C. Anjana, T. Zacharia, and R. Gandhiraj, “Real time implementation of FMCW radar for target detection using GNU radio andusrp,” in 2015 *International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015, pp. 1530–1534.