# SDR-BASED TRACKING SYSTEM FOR BIODEGRADABLE IMPLANT SENSOR

by

Diren Erdem Aydın

B.S., Electrical and Electronics Engineering, MEF University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2023

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor Sema Dumanlı Oktar and Faik Baskaya for their exceptional support, guidance, and mentorship throughout the entire research process. Their expertise, invaluable insights, and dedication have been instrumental in shaping this thesis and achieving its successful completion. I am truly grateful for their unwavering commitment and encouragement.

Moreover, I am delighted to extend my appreciation to Tuba Ayhan for joining my thesis committee, her unwavering support and profound understanding have allowed me to concentrate on my thesis with confidence and dedication.

I would also like to express my gratitude to Ahmet Bilir for his contributions to this thesis and his support during the measurement phase.

Finally, I would like to extend my heartfelt appreciation to my parents, Nevin and Hakan, for their patience, encouragement, and support throughout this journey. This thesis is dedicated to them as a token of my gratitude and love.

# ABSTRACT

## SDR-BASED TRACKING SYSTEM FOR BIODEGRADABLE IMPLANT SENSOR

Covid-19 pandemic had an impact on society's priorities towards the end of 2019. In the short term, people had to relinquish their regular social activities and adopt new habits to safeguard their health. Consequently, personal health emerged as the foremost concern in daily life, surpassing all others. This heightened awareness of personal health has led to an increased demand for novel medical systems, irrespective of whether they are specifically related to the Coronavirus or not. Even prior to the recent pandemic, advancements in nano and RF technologies have already hinted at the potential of health applications based on wireless communication. RF medical imaging and sensing has emerged as a promising adjunct to existing diagnostic tools in the field of medical applications, primarily due to its cost-effectiveness, portability, and non-ionizing radiation properties. Software Defined Radios (SDR) are an exceptionally well-suited option for medical applications due to their remarkable flexibility, enhanced RF specifications, and compact hardware dimensions. This thesis introduces a novel low-cost and wearable system as a viable substitute for the Vector Network Analyzer (VNA), a pivotal measurement technology in the RF industry. While several studies have explored the use of SDR as a substitute for VNAs, it is noteworthy that this research represents the pioneering application of an SDR-based system in conjunction with genetically modified bacteria for sensing biological processes and specific chemical outputs generated by molecular communications. The system design and implementation involve the utilization of GNU Radio and Python for Adalm-Pluto SDR development, MATLAB for advantaged signal processing tasks, and the establishment of a measurement setup using biological phantoms. The comparison of SDR results with those from a VNA provides valuable insights into the system's effectiveness.

# ÖZET

## BİYOBOZUNUR İMPLANT SENSORU İÇİN SDR TABANLI TAKİP SİSTEMİ

Toplumun öncelikleri, Covid-19 pandemisiyle 2019 yılının sonlarına doğru değişti. Insanlar kısa vadede sağlıklarını korumak için düzenli olarak yaptıkları bazı aktivitelerden vazgeçmek ve yeni alışkanlıklar benimsemek zorunda kaldılar. Sonuç olarak, kişisel sağlık günlük yaşamda diğer her şeyin önüne geçen en önemli konu haline geldi. Kişisel sağlık bilincinin artmasıyla birlikte, Corona Virüsü ile ilgili olsun veya olmasın yeni tıbbi sistemlere olan talep arttı. Son dönemdeki pandemiden bağımsız olarak, nano ve RF teknolojilerindeki ilerlemeler, kablosuz iletişime dayalı sağlık uygulamalarının potansiyelini göstermiştir. RF tabanlı tıbbi görüntüleme, tıbbi uygulamalar alanında mevcut teşhis araçlarının etkili bir tamamlayıcısı olarak ortaya çıkmıştır. Bunun temel nedenleri arasında maliyet etkinliği, taşınabilirlik ve iyonlaştırıcı olmayan radyasyon özellikleri bulunmaktadır. Yazılım Tabanlı Radyo (YTR), esnek olması, geliştirilmiş RF özellikleri ve kompakt donanım boyutları nedeniyle tıbbi görüntüleme uygulamaları için son derece uygun bir seçenektir. Bu tez, RF endüstrisindeki temel ölçüm teknolojilerinden biri olan Vektör Ağ Analizörünün (VAA) yerine geçebilecek uygun maliyetli ve takılabilir bir sistem sunmaktadır. Bu bağlamda birçok çalışma yapılmış olsa da, bu araştırma, moleküler iletişim tarafından üretilen biyolojik süreçleri ve özel kimyasal çıktıları algılamak için genetik olarak değiştirilmiş bakterilerle birlikte YTR tabanlı bir sistemin öncü uygulamasını temsil etmektedir. Sistem tasarımı ve uygulaması, Adalm-Pluto YTR geliştirme için GNU Radio ve Python'un kullanılmasını, gelişmiş sinyal işleme görevleri için MATLAB'in kullanılmasını ve biyolojik fantomlarla bir ölçüm kurulumunun oluşturulmasını içermektedir. YTR sonuçlarının VAA sonuçlarıyla karşılaştırılması, sistemin etkinliği hakkında değerli bilgiler sağlamaktadır.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xiv
LIST OF SYMBOLS . . . xv
LIST OF ACRONYMS/ABBREVIATIONS . . . xvi
1. INTRODUCTION . . . 1
2. BACKGROUND AND PRELIMINARIES . . . 6
2.1. Software Defined Radio . . . 6
2.2. Sine - Gaussian Signal . . . 13
2.3. Biology-Inspired Communication . . . 16
3. HARDWARE AND SOFTWARE TOOLS . . . 21
3.1. ADALM-PLUTO SDR Hardware . . . 21
3.1.1. Modulation Basics . . . 26
3.1.2. Phase and Quadrature Signals . . . 28
3.1.3. Decimation and Interpolation . . . 36
3.2. Software . . . 38
3.2.1. Industrial I/O Subsystem . . . 38
3.2.2. GNU Radio . . . 41
3.2.3. MATLAB . . . 41
4. IMPLEMENTATION OF THE SYSTEM . . . 42
4.1. Extraction of Transmitted Signal . . . 42
4.2. Transmit and Receive the Extracted Signal with Pluto SDR . . . 48
4.2.1. Tracking Options . . . 50
4.2.1.1. RF DC Correction . . . 51
4.2.1.2. BB DC Correction . . . 51
4.2.1.3. Quadrature . . . 53
4.2.2. Phase Randomization . . . 55
4.3. Final Flowgraph . . . 60

4.4. Measurements and Setup . . . . . 66
4.5. Post Processing with MATLAB . . . . . 71
5. RESULTS AND COMPARISON . . . . . 76
5.1. Results from Selected Points . . . . . 76
5.2. Using SDR as a VNA . . . . . 81
5.2.1. Comparison Among Different Step Sizes . . . . . 82
5.2.2. Comparison of SDR and VNA Results . . . . . 84
6. CONCLUSION AND FUTURE WORK . . . . . 90
REFERENCES . . . . . 92
APPENDIX A: PERMISSIONS FOR VISUAL MATERIALS . . . . . 100

# LIST OF FIGURES

Figure 2.1. SDR block diagram. . . . . 7

Figure 2.2. Multistages of superheterodyne architecture components and filter images, (a) BPF 1 image, (b) BPF 2 image, (c) LPF image. . . . . 8

Figure 2.3. Zero IF architecture components. . . . . 10

Figure 2.4. Sampling of a real signal on the frequency domain, (a) frequency image of the signal, (b) sampled with higher a rate, (c) aliasing due to lower rate. . . . . 11

Figure 2.5. Effects of the sampling rates on a sinusoidal signal. . . . . 12

Figure 2.6. Sine-Gaussian wave in the time domain. . . . . 14

Figure 2.7. Frequency domain representation of Sine-Gaussian components. . . . . 15

Figure 2.8. Schematics of proposed frameworks for molecular communication scenarios [16]. . . . . 17

Figure 2.9. The effect of bacterial activity on biodegradation speed, without e.coli (top), with e.coli (bottom) [16]. . . . . 19

Figure 2.10. Measurement setup for passive in-body sensing system consisting antennas, resonators at two different phases (left), and initial experimental results demonstrating the change in scattering parameter S21 in dB (right). [16]. . . . . 20

Figure 2.11. Demonstration of proposed concepts, reconfiguration using muscle tissue (left) and bacteria (right) [17]. . . . . 20

Figure 3.1. (a) ADALM-Pluto SDR module, (b) analog and digital blocks on the SDR [45]. . . . . 23

Figure 3.2. AD9363 transceiver receiver path [55]. . . . . 24

Figure 3.3. AD9363 transceiver transmitter path [55]. . . . . 25

Figure 3.4. Simplified block diagram of Pluto SDR [45]. . . . . 26

Figure 3.5. 100 Hz sinusoidal signal with 1 V amplitude - s(t). . . . . 27

Figure 3.6. Used signals in modulation. . . . . 28

Figure 3.7. Quadrature modulator. . . . . 29

Figure 3.8. In-phase, quadrature components of a complex signal and amplitude modulated sum. . . . . 30

Figure 3.9. In-phase, quadrature components of a complex signal and amplitude, phase modulated sum. . . . . 30

Figure 3.10. BPSK modulation example. . . . . 31

Figure 3.11. Quadrature representation of signals: (a) $f_0$ Hz sinusoidal signal, (c) continuous time real signal with BW B [47]. . . . . 33

Figure 3.12. Resulting quadrature-sampling spectrums using the in-phase component [47]. . . . . 34

Figure 3.13. Resulting quadrature-sampling spectrums using the Q component [47]. . 35

Figure 3.14. Addition result of filtered I and Q components [47]. . . . . 35

Figure 3.15. Structure IIO command [50]. . . . . 38

Figure 3.16. LibIIO diagram [52]. . . . . 40

Figure 3.17. Pluto SDR system level diagram [50]. . . . . 40

Figure 4.1. TX sinc signal flowgraph. . . . . 42

Figure 4.2. Time domain representation of the pulse. . . . . 43

Figure 4.3. Frequency domain representation of the pulse. . . . . 45

Figure 4.4. Designed low pass filter by using GNU Radio filter design tool. . . . . 46

Figure 4.5. TX signal in the time domain. . . . . 47

Figure 4.6. TX signal in the frequency domain. . . . . 48

Figure 4.7. Initial flowgraph with PlutoSDR sink and source blocks. . . . . 49

Figure 4.8. LO leakage impact to received spectrum. . . . . 51

Figure 4.9. Possible scenarios for self mixing issue . . . . . 52

Figure 4.10. Spike free spectrum by BB DC correction. . . . . 53

Figure 4.11. In-phase (green) and quadrature (red) when quadrature tracking is on. . 54

Figure 4.12. In-phase (green) and quadrature (red) when quadrature tracking is off. 55

Figure 4.13. Effects of phase randomization on received signals, received I and Q pair (top), transmitted I and Q pair (bottom). . . . . 56

Figure 4.14. Pluto SDR TX and RX PLL synthesizer block diagram, (1) reference divider, (2) fractional-n divider, (3) frequency-phase detector and charge pump [46]. . . . . 57

Figure 4.15. VCO divider [46]. . . . . 59

Figure 4.16. Phase ambiguity of 0 or 180 degrees. . . . . 60

Figure 4.17. Flowgraph on GNU Radio with IIO blocks. . . . . 61

Figure 4.18. Final flowgraph with proposed custom Python block. . . . . 64

Figure 4.19. Different statues of SRR and dimensions of biodegradable implant [68]. 66

Figure 4.20. VNA output of S21 scattering parameter in dB for different SRR status and related SRR status for each curve; degraded (red), partially degraded (green), non-degraded (blue) [68]. . . . . 67

Figure 4.21. Simulation model of human tissue mimicking experimental setup [68]. . 68

Figure 4.22. 3D printed versions of the experimental setup, without muscle phantom (left), filled with muscle phantom (right) [68]. . . . . 69

Figure 4.23. The fabricated antenna [68]. . . . . 69

Figure 4.24. SRR included the final version of the experimental setup [68]. . . . . 70

Figure 4.25. Overall system, including host PC, Pluto SDR, and experimental setup. 71

Figure 4.26. Received raw data with Pluto SDR, (a) in-phase, (b) quadrature. . . . . 72

Figure 4.27. Calculated averages from raw data for each resonator status, (a) in-phase, (b) quadrature. . . . . 73

Figure 4.28. Magnitude of received signals for each SRR status in data converter codes. . . . . 73

Figure 4.29. Proposed SDR results for different status and parameters in dBFS, (a) 1 cm - rotation 0 degrees, (b) 4 cm - rotation 270 degrees. . . . . 75

Figure 5.1. VNA output for different SRR statuses at 7 cm Depth for S21 scattering parameter. . . . . 77

Figure 5.2. SDR measurement results in data converter codes for selected frequencies at 7 cm, (a) 1.2 GHz, (b) 1.3 GHz. . . . . 78

Figure 5.3. VNA output for different SRR status at 2 cm depth for S21 scattering parameter. . . . . 80

Figure 5.4. Illustration of each frequency point and fitted curves for no ring case by using different step sizes, (a) 1 cm no ring 2 MHz step size, (b) 1 cm no ring 5 MHz step size. . . . . 83

Figure 5.5. SDR and VNA measurement results for 0 degrees rotation at 1 cm in dBFS, (a) SDR, (b) VNA. . . . . 85

Figure 5.6. SDR and VNA measurement results for 90 degrees rotation at 1 cm in dBFS, (a) SDR, (b) VNA. . . . . 85

Figure 5.7. SDR and VNA measurement results for 180 degrees rotation at 4 cm in dBFS, (a) SDR, (b) VNA. . . . . 86

Figure 5.8. SDR and VNA measurement results for 270 degrees rotation at 4 cm in dBFS, (a) SDR, (b) VNA. . . . . 86

Figure 5.9. SDR measurement results for 0 and 90 degrees rotations at 1 cm with 5 MHz step size in dBFS, (a) rotation 0, (b) rotation 90. . . . . 87

Figure 5.10. SDR results for 180 and 270 degrees rotations at 4 cm with 5 MHz step size in dBFS, (a) rotation 180, (b) rotation 270. . . . . 88

# LIST OF TABLES

Table 3.1. Comparison of classic sampling and complex sampling . . . . . 36

Table 4.1. Magnitudes of the received signal for all possible I and Q cases. . . . . 63

Table 4.2. Comparison between proposed method and previous method. . . . . 74

Table 5.1. Comparison of simulation parameters, 2 MHz and 5 MHz step sizes. . . 84

# LIST OF SYMBOLS

| | |
|---|---|
| $f_s$ | Sampling Frequency |
| $\kappa$ | Angular Frequency |
| $\phi$ | Phase |
| $\omega$ | Angular Frequency |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| ADALM | Analog Devices Active Learning Module |
| ADC | Analog to Digital Converter |
| AM | Amplitude Modulation |
| API | Application Programming Interface |
| BAN | Body Area Networks |
| BB | Baseband |
| BER | Bit Error Rate |
| BPSK | Binary Phase Shift Keying |
| C | Capacitor |
| CM | Centimeter |
| CMOS | Complementary Metal–Oxide–Semiconductor |
| CPU | Central Processing Unit |
| DAC | Digital to Analog Converter |
| dB | Decibels |
| dBFS | Decibels Relative to Full Scale |
| DC | Direct Current |
| DCXO | Digitally Controlled Crystal Oscillator |
| DDC | Digital Down Conversion |
| DDS | Direct Data Synthesis |
| DDR3L | Dual Date Rate 3 Low-Voltage |
| DSP | Digital Signal Processing |
| DSRR | Double Split-Ring Resonator |
| DMA | Direct Memory Access |
| DUC | Digital Up Conversion |
| E. coli | Escherichia coli |
| EVM | Error Vector Magnitude |
| FFT | Fast Fourier Transform |
| FIR | Finite Impulse Response |
| FM | Frequency Modulation |
| FPGA | Field Programmable Gate Array |

| | |
|---|---|
| FT | Fourier Transform |
| GPS | Global Positioning System |
| GRC | GNU Radio Companion |
| HB | Half Band |
| I | Phase |
| IC | Integrated Circuit |
| IF | Intermediate Frequency |
| IIO | Industrial Input-Output |
| IIOD | Industrial Input-Output Daemon |
| IP | Internet Protocol |
| I/Q | Phase and Quadrature |
| kSPS | Kilo Samples Per Second |
| L | Inductor |
| LNA | Low Noise Amplifier |
| LPF | Low Pass Filter |
| LO | Local Oscillator |
| LVDS | Low-Voltage Differential Signaling |
| Mg | Magnesium |
| MATLAB | Matrix Laboratory |
| MC | Molecular Communication |
| MNCN | Molecular Nano Communication Networks |
| MSPS | Mega Samples Per Second |
| OTG | On-The-Go |
| PC | Personal Computer |
| PFD | Phase Frequency Detector |
| PLL | Phase Locked Loop |
| PM | Phase Modulation |
| Q | Quadrature |
| QPSK | Quadrature Phase Shift Keying |
| QSPI | Quad Serial Peripheral Interface |
| RF | Radio Frequency |
| RX | Receiver |
| SD | Standard Deviation |

| | |
|---|---|
| SDR | Software Defined Radio |
| SRR | Split Ring Resonator |
| SSH | Secure Shell |
| SMA | SubMiniature version A |
| SNR | Signal to Noise Ratio |
| TCP | Transmission Control Protocol |
| TIA | Transimpedance Amplifiers |
| TX | Transmitter |
| URI | Uniform Resource Identifier |
| USB | Universal Serial Bus |
| VCO | Voltage Controlled Oscillator |
| VGA | Variable Gain Amplifier |
| VNA | Vector Network Analyzer |
| WCB | Whole Cell Biosensor |

# 1. INTRODUCTION

Radio Frequency (RF) imaging and sensing, especially near-field sensing, is a compelling area of study due to its ability to effectively penetrate optically obscured or opaque media, including clothing, walls, luggage, and living tissues. From a technical standpoint, RF or often referred to as microwave imaging, typically operates within the frequency range of 300 MHz to 300 GHz [1].

The significant advancements and rapid growth in microwave near-field imaging applications can be attributed not only to the demand for such technology but also to the availability of advanced tools and techniques over the past 15-20 years. Notably, there has been a remarkable miniaturization of radar and radio electronics. Presently, there are on-chip radios with surface areas of up to 1 $mm^2$, specifically designed for operation in the low gigahertz frequency range. This development enables researchers to integrate a radio onto each antenna, facilitating the creation of large-scale active antenna arrays.

The applications of microwave imaging span a wide range of fields, including security applications such as whole body scanners [2], mining operations [3], characterization of circuits [4], military applications like underground radar systems [5], software radio [6] and, notably, medical applications as imaging tools [7-11]. However, microwave applications has garnered significant attention in recent years as a promising adjunct to current diagnostic tools in medicine. This can be attributed to its inherent advantages such as cost-effectiveness, portability, and utilization of non-ionizing radiation [12]. These medical applications entail the transmission of a microwave signal into a subject's body, followed by the measurement of the resultant signal alterations. By collecting an adequate number of these modified signals across various input frequencies and positions surrounding the body, it becomes possible to identify the diseases characterized by notable disparities in permittivity or conductivity compared to the surrounding tissues.

The applicability of microwave imaging systems for medical purposes is typically assessed through experimental evaluations employing a vector network analyzer as the primary microwave transceiver. VNA plays a crucial role as a fundamental measurement tool

in the RF and microwave industry. As electronic technologies continue to advance at a rapid pace, the demand for this equipment has significantly increased. However, the acquisition of a VNA presents a significant challenge primarily due to its exorbitant cost, often reaching several thousands of dollars [13]. This financial barrier impedes access to VNAs for many researchers and organizations, limiting their ability to perform essential measurements and analysis in the RF and microwave domain. Moreover, the VNA's large shape poses challenges in achieving portable microwave imaging systems. Therefore, a crucial prerequisite for transforming microwave imaging into a widely accessible mass screening diagnostic tool is the development of a low-cost, portable unit capable of characterizing, generating, transmitting, and receiving signals across a wideband with substantial dynamic range and stability. A substantial body of literature has explored the utilization of SDR as a viable substitute for VNAs.

Microwave signals experience considerable attenuation when propagating through human tissues due to the inherent lossiness of biological materials at microwave frequencies ranging from 300 MHz to 300 GHz. Balancing the desired signal penetration and image resolution necessitates careful consideration. It is worth noting that despite the potential trade-off in resolution, lower frequencies are generally preferred for improved penetration in materials containing water, which can cause significant signal loss. Higher frequency spectra offer improved resolution, while lower frequencies are preferred for their superior penetration capabilities, particularly in media containing significant water content, which is prevalent in various tissues. Thus, microwave frequencies ranging from 0.5 GHz to 4 GHz have been extensively employed in diverse medical applications, including head, torso, and breast imaging, this range is also covering the used frequency band for the proposed system, 0.8 GHz to 1.5GHz. Therefore, an ideal microwave transceiver for medical imaging purposes should encompass this wide frequency band. Furthermore, ensuring accurate detection in complex and cluttered environments, such as those encountered in medical imaging scenarios, demands a transceiver with a substantial dynamic range capable of effectively managing high levels of interference.

When employing a versatile commercial product like the SDR for applications that demand precise selectivity and high sensitivity, it is crucial to configure the product's specifications carefully to achieve optimal performance. Nevertheless, under certain circumstances,

the stringent demands for precise measurements are relaxed, the cost and simplicity factors taken into consideration. Utilizing SDR as a viable alternative to a VNA necessitates hardware and software modifications, as well as calibrations specific to the SDR device being employed. Among the affordable SDR platforms commonly utilized in amateur and academic applications, the Adalm-Pluto SDR stands out as one of the most popular choices. However, when operating in full-duplex mode, where both the transmitter and receiver operate simultaneously, certain peculiarities may arise such as variations of approximately 0.2 dB when conducting repetitive measurements, or encounter fluctuations that can be attributed to non-ideal characteristics within its components [4].

To ensure effective functionality in full-duplex communication scenarios, numerous SDR systems utilize distinct frequency synthesizers for the transmit and receive paths. This design prevents any overlap between the frequency spectrums used for transmission and reception. However, in cases where the same frequency is employed for both transmission and reception, such as with the Adalm-Pluto SDR, the utilization of separate synthesizers introduces a non-deterministic phase offset, which arises each time the carrier frequency is retuned or when any SDR parameter is reset [14]. In order to address the non-deterministic phase offset issue, a proposed solution in reference [14] suggests conducting all spatial measurements before testing each new frequency. This approach involves altering the antenna positions between several settings that mimic the human body. In the context of this thesis, the only distinction is that instead of the antenna, a biodegradable sensor is the moving component. On the contrary, there are certain SDR systems that employ shared frequency synthesizers to mitigate the non-deterministic phase offset issue. By using shared synthesizers, the system can maintain a consistent phase relationship between the transmit and receive paths but it is worth noting that this approach often comes with a notable increase in the overall cost of the system [15].

In this study, a Data Capture Module Design for a Multi-scale Communication System has been developed, utilizing the Adalm-Pluto SDR as the hardware implementation. The software of the SDR is programmed using open-source projects such as GNU Radio and the Python Programming Language.

The project represents an extension of the Bountenna Research Group's endeavors from earlier studies [16-18]. These valuable prior works encompassed diverse aspects, including the utilization of biological processes of biodegradable sensor, antenna design, and the development of a methodology for multi-scale communication within a system that relies on engineered bacteria. The background section provides a comprehensive explanation of the concepts explored in these studies.

The proposed system begins by generating a wide-band arbitrary waveform using software. This waveform is then transmitted to a biodegradable sensor, and the reflected signals from the chosen spectrum are collected by the SDR device. For further signal processing algorithms, MATLAB is employed. By analyzing the results, the system can determine the concentration of the molecules of interest using a low-cost SDR device, comparable to the capabilities of a Vector Network Analyzer.

The main contribution of this work lies in its novelty. While previous studies have explored the use of SDR as a substitute for VNAs, this research represents a pioneering application of an SDR-based system in conjunction with genetically modified bacteria for sensing biological processes and specific chemical outputs generated by molecular communications. Additionally, a software-based solution has been proposed to address the non-deterministic phase offset issue, resulting in reduced measurement times and workload.

The thesis structure begins with Chapter 2, where the fundamental background and theoretical concepts that serve as the foundation for the research are comprehensively explained. This chapter provides a solid understanding of the principles and theories underlying the concept employed in the thesis. In Chapter 3, detailed explanations are provided regarding the hardware and software tools that were utilized in the design of the system. This chapter delves into the specific components, devices, and programming languages used, shedding light on their functionalities and significance within the context of the research. Chapter 4 presents the complete system architecture, providing an overview of how all the individual components and modules of the system are integrated to form a cohesive whole. This chapter offers a comprehensive understanding of the system's structure and organization. Moving on to Chapter 5, this section focuses on the practical aspects of the research, including the measurements conducted using the developed system. The results obtained

from these measurements are presented, and a thorough comparison and analysis of the findings are provided. Finally, in Chapter 6, the thesis concludes with a summary of the key findings and contributions. The conclusion highlights the achievements of the research and provides insights into the implications and potential future directions for further work in the field.

# 2. BACKGROUND AND PRELIMINARIES

In this chapter, a comprehensive summary of the key concepts and theoretical foundations employed in the proposed work is provided. The purpose of this summary is to enhance the reader's understanding of the subsequent chapters and ensure a clear comprehension of the entire document.

## 2.1. Software Defined Radio

The evolution of communication systems over time has been remarkable, with significant advancements observed from the invention of Samuel Morse's electric telegraph in 1837 to the emergence of Bluetooth devices in the early 2000s, owing to the continual enhancement of technological capabilities. During this progress, traditional radio systems have been replaced with newly emerged technologies, one of which is Software Defined Radio devised by Joseph Mitola and defined as a class of radios that could be programmed and reconfigured by using software [19]. The conventional radio systems that have been widely utilized are characterized by their hardware-based architecture, which imposes rigid, pre-determined operational parameters. Specifically, the modulation type and the permissible range of received input signal frequency are typically pre-specified and not readily amenable to alteration. Therefore, radio functionality is limited with the implemented hardware and it is not possible to modify hardware for other modulation types or employ performance upgrades in case of any error [20]. On the other hand, as defined by The Institute of Electrical and Electronic Engineers (IEEE) P1900.1 Working Group, SDR refers to a type of radio system in which the infrastructure necessary for data transfer and processing is software-based, whereby certain or all physical layer functionalities are implemented via software [21]. The physical layer is formed by hardware components of SDR that are adaptable to software configuration, encompassing key elements such as filters, modulators, demodulators, mixers, and amplifiers, enabling greater flexibility and functionality to end-users.

A simplified SDR block diagram is shown in Figure 2.1. The first block is referred to as the "analog" and includes the Radio Frequency front-end, which is a general term

for all components between an RX/TX antenna up to the mixer stage [22] and some IF amplifiers and filters. Considering the constituent elements incorporated in this particular segment of the SDR architecture, the RF front-end part is software controlled rather than software-defined [23], which means cut-off frequency for tunable analog filters and gain parameters for analog amplifiers are controlled through software whereas in traditional radio architecture, these components are fixed and have a predetermined functionality. On the contrary, its digital counterpart given in Figure 2.1 can be fully defined by software due to general-purpose hardware blocks such as Field Programable Gate Array (FPGA) to handle signal processing burdens. The integration of software-controlled RF front-end and software-based signal processing culminates in the emergence of the SDR paradigm.

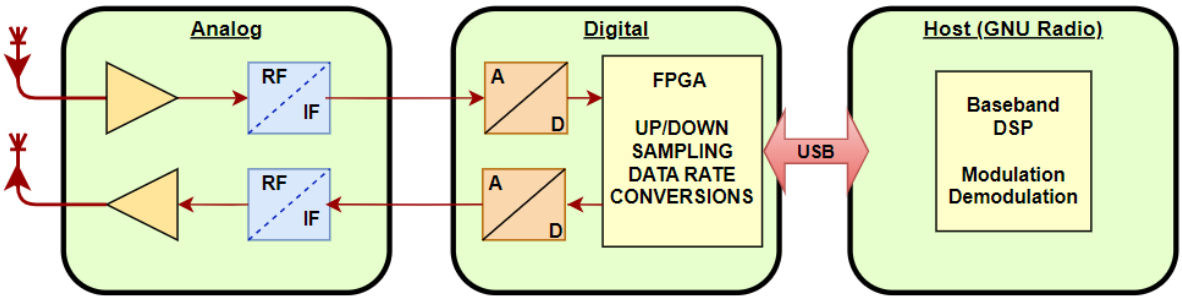



Figure 2.1. SDR block diagram.

The superheterodyne architecture has persisted as the predominant method for designing the RF front-end for over a century, representing an advanced iteration of the heterodyne technique [24]. This approach addresses inherent challenges associated with image frequencies occurring in heterodyne architecture, thereby enhancing the overall performance of the system. Figure 2.2 depicts the RX and TX paths of a multistage superheterodyne architecture. The receiver path of this architecture incorporates two down-conversion stages, hence it is referred to as multistage. In the first stage, incoming RF signals are mixed with a local oscillator (LO) signal, resulting in an intermediate frequency (IF) signal. By adjusting the local oscillator frequency, the IF signal is always produced at a fixed frequency, regardless of the input frequency of the receiver. This allows for the IF filters and amplifiers to be designed for a specific frequency, with commonly used IF values for FM and AM receivers being approximately 10.7 MHz and 455 kHz, respectively. Otherwise, a tuning mechanism should be implemented to control filter elements like variable capacitors and

inductors, which are hard to implement with adequate selectivity. This old RF architecture where a turning mechanism exists is known as tuned radio frequency and has not been used since the invention of superheterodyne architecture. The second conversion takes the IF signal to the baseband or the second IF value, which is lower than the first one for Digital Signal Processing (DSP) blocks. The advantages of superheterodyne architecture are; a satisfactory level of sensitivity (due to image reduction by BPF1, interferers from image frequencies are largely removed thus RF signals can be captured at lower power), sharp selectivity (bandwidth of interest can be separated efficiently due to the BPF2 and LPF given in Figures 2.2.2 and 2.2.3, as long as IF frequency is known the most suitable filter can be designed in terms efficiency and power), and immunity to DC problems (at least in the first stages) such as LO leakage since the received signal is not directly converted to baseband.

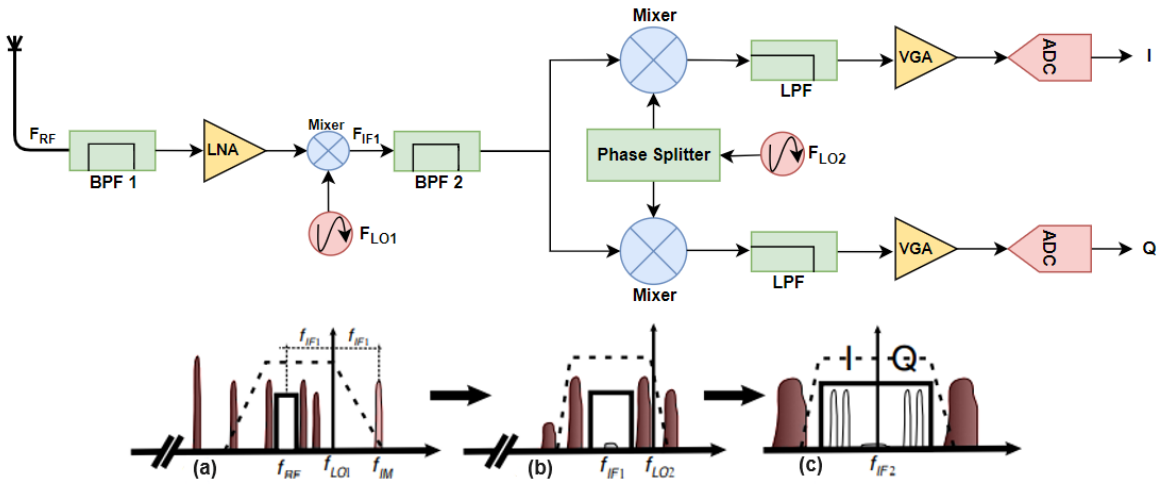



Figure 2.2. Multistages of superheterodyne architecture components and filter images, (a) BPF 1 image, (b) BPF 2 image, (c) LPF image.

Prior to delving deeper into the topic, it is crucial to acknowledge the issue of image frequency that is associated with heterodyne architecture. Furthermore, it is imperative to explicate the dissimilarities between superheterodyne and heterodyne. The conversion of frequencies from RF to IF or RF to baseband (0 Hz) is achieved through RF mixers, which are located between the components of a low-noise amplifier (LNA), BPF2, and LPF as illustrated in Figure 2.2. The mixer takes the RF signal at $f_{RF}$ frequency and mixes it

with the LO signal at frequency $f_{LO}$ and produces an output signal that contains sum and differences frequencies [28] from the given trigonometrical equation below

$$\cos(f_{RF}) \times \cos(f_{LO}) = \frac{1}{2}[\cos(f_{RF} - f_{LO}) + \cos(f_{RF} + f_{LO})]. \tag{2.1}$$

With the proper filtering, either of the outputs can be selected and the IF is determined in this way. Although it seems that math fits faultlessly to this operation, there may raise some issues because of the image response. Consider the case given in Figure 2.2, according to the leftmost frequency axis (2.2.a), which is captured by the RX antenna, the first intermediate frequency will be equal to $f_{LO1} - f_{RF}$ and the RF component is going to be downconverted to $f_{IF1}$. However, there is one more image located at $f_{IM}$, which is $f_{IF1}$ Hertz away from $f_{LO1}$. The image is downconverted to $f_{IF1}$ as well and corrupts the RF signal, diminishing receiver sensitivity, since the cosine function is an even function as shown in the equation:

$$\cos(f_{LO} - f_{RF}) = \cos(f_{RF} - f_{LO}) = \cos(f_{IF1}). \tag{2.2}$$

In order to overcome image issues BPF1 is placed in the superheterodyne architecture and the image is rejected by filtering techniques. Moreover, selecting $f_{IF1}$ high makes the image rejection process more straightforward due to unrestricted filter requirements such as longer transition widths. However, to maintain proper selectivity from the interferences, BPF2 given in Figure 2.2.b should have a sharp transition width to separate bandwidth of interest and this is only possible by using high quality (Q) components since $f_{IF1}$ is high. Integration of these high Q filters (SAW or ceramic) full on-chip is unpractical in current integrated circuit (IC) technologies (cheap, scalable, and power-efficient in digital CMOS tech is the current trend.) [25] since they remain physically large compared to other radio frequency (RF) components. Additionally, these analog filters are often custom designs so it is also a disadvantage in terms of cost [23].

The superheterodyne architecture's issues have led to the emergence of a solution known as the zero-IF (ZIF) architecture, as shown in Figure 2.3. The ZIF architecture has gained popularity due to its simplicity, making it more suitable for full chip integration than the superheterodyne architecture. Additionally, the reduced number of components in ZIF architectures makes them more cost-effective than the superheterodyne architecture [26]. In the zero-IF architecture, the need for multiple intermediate frequencies or bandpass filter stages is eliminated. Instead, the received RF signal is directly downconverted to

DC by an RF mixer, and then the bandwidth of interest is filtered by a low-pass filter (LPF) at the baseband. This approach allows for easier and less expensive analog filtering, as baseband filters are simpler to design than custom high-frequency filters used in the superheterodyne architecture. The relaxed filtering conditions in the zero-IF architecture enable full integration of analog and digital parts on the same silicon, facilitating the design process [23]. Despite the aforementioned advantages, the zero-IF architecture has some drawbacks compared to its heterodyne counterparts. One of these drawbacks is that some components, such as the LNA, need to cover a wider frequency range in the order of gigahertz due to the lack of an RF prefilter. This wider range requirement can make it challenging to achieve full chip integration. Furthermore, when there is no prefilter in place, the mixer is subjected to stringent linearity requirements [27]. In this scenario, the mixer's output is expected to either increase or decrease in the same proportion as the RF signal [28]. Otherwise, receiver sensitivity is affected in the presence of interferers if they are moved to the band of interest, as occurs in any nonlinear system [29]. In addition to the challenges in the design and implementation of the zero-IF architecture, there are also some other significant issues that need to be addressed, including but not limited to DC offset, I/Q mismatch, and flicker noise. The origins of these issues can be traced back to various sources, such as component non-idealities and environmental factors, and they can have a detrimental impact on the performance of the receiver. Therefore, it is crucial to thoroughly investigate these problems and devise effective countermeasures to mitigate their effects. These topics will be discussed in greater detail in the subsequent sections.

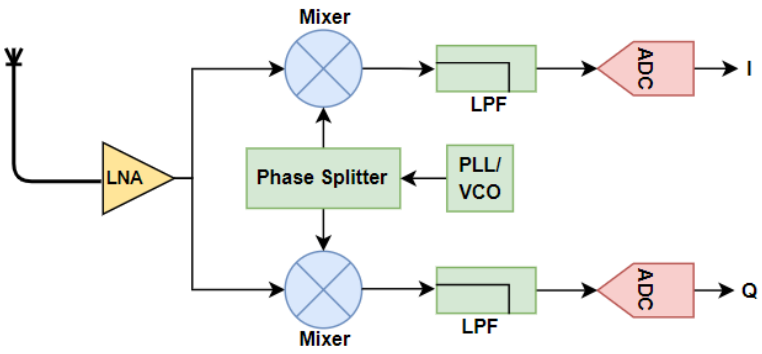



Figure 2.3. Zero IF architecture components.

The second functional block depicted in Figure 2.1 is referred to as the digital block, which encompasses data converters and the digital back-end components such as FPGA or Application Specific Integrated Circuit (ASIC), and DSP blocks. It is crucial to select the RX or TX signal bandwidth based on the data converters' sampling rates to comply with the Nyquist criteria. Nyquist sampling theorem states that in order to reconstruct a real signal, which is bandlimited to $f_h$ Hz, samples should be taken uniformly at least $2f_h$ samples per second, thus minimum sampling frequency $f_s$ is defined as $f_s > 2f_h$ [30]. In Figure 2.4 there are two examples of sampling; at b, sampling frequency $f_s$ is selected at least 2 times higher than the highest frequency component in the sampled signal $f_s$, therefore $H(f)$ can be reconstructed as $h(t)$ without any corruption. However, in Figure 2.4.c, selected $f_s$ does not satisfy Nyquist Criteria, and the replica at $f_s$ overlaps with the image at baseband, which is called aliasing and indicated with dashed circles in the figure, so it is not possible to reconstruct $h(t)$ without any error. In applications that require data conversions, it is essential to select an appropriate ADC sample rate, typically at least twice the highest frequency present in the signal to satisfy the Nyquist criterion. However, in many cases, increasing the sample rate beyond the minimum required can result in improved data transfer from the analog domain to the digital one, as demonstrated in Figure 2.5.

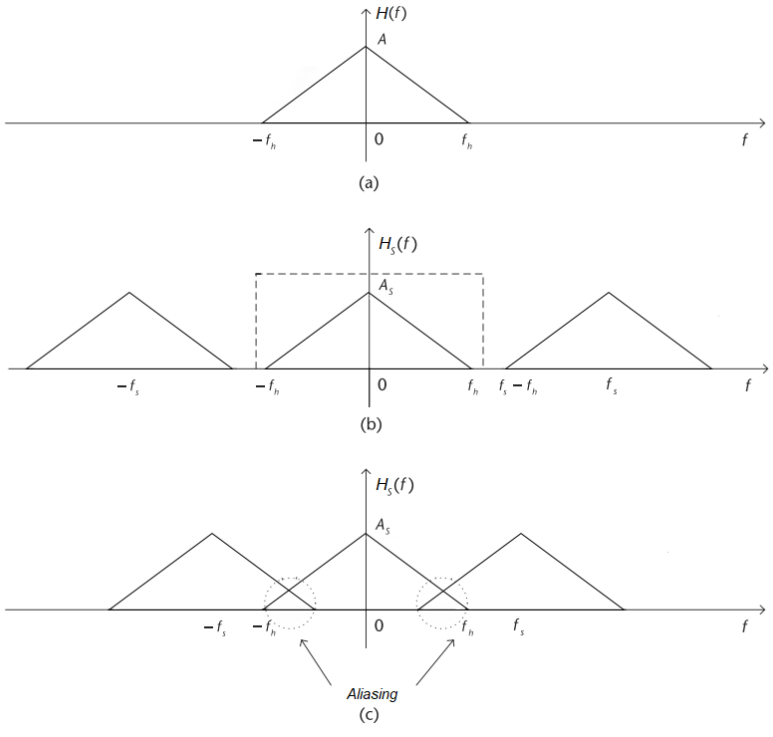



Figure 2.4. Sampling of a real signal on the frequency domain, (a) frequency image of the signal, (b) sampled with higher a rate, (c) aliasing due to lower rate.

It is evident from the figure that the digital signal becomes more reliable with the increasing sampling rate, resulting in more accurate frequency and amplitude information, as shown in Figure 2.5.c. Therefore, selecting a higher ADC sample rate than the minimum requirement can lead to better overall system performance.

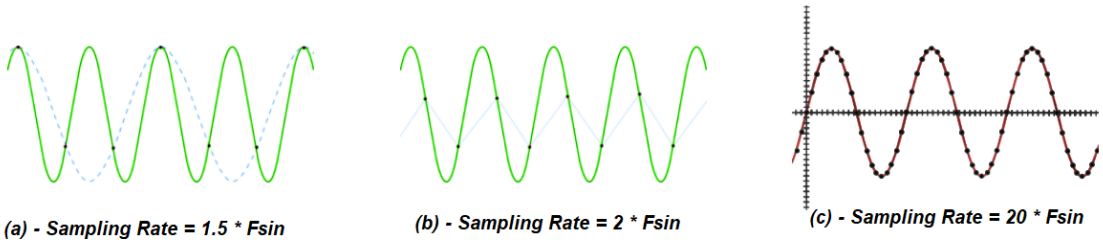



Figure 2.5. Effects of the sampling rates on a sinusoidal signal.

As analog data is converted to the digital domain, various components are utilized in accordance with the implemented architecture for digital data processing. General-purpose processors (GPPs) are employed for stand-alone field applications with software-defined radio and control the data flow between hardware and the host PC in cases where data is passed to the PC for further analysis.

However, GPPs are ill-suited for signal processing tasks, as they process instructions sequentially and can be inefficient in terms of time and power consumption when used for mathematical computations. To handle these mathematical computations efficiently, digital signal processors (DSPs) are typically employed, though DSPs may be inadequate for certain computationally intensive processes that require custom design. In such cases, FPGAs are selected due to their reconfigurability and parallel computation capabilities. As such, FPGAs are the most suitable hardware for computationally intensive tasks such as digital down conversion (DDC), digital up conversion (DUC), digital filtering, and channelizing (including sample buffers and JESD interfaces).

The final block in the "Basic SDR Block Diagram" pertains to software environments that enable the configuration of various radio parameters, including bandwidth, local oscillator frequency, sampling rate, etc., via a USB interface. Furthermore, modulation and demodulation can be performed using software environments such as MATLAB and GNU

Radio. If the SDR hardware possesses the capability to execute modulation and demodulation, then these algorithms can be implemented onboard.

The technical background of SDR has been explained in brief thus far. This state-of-the-art technology has found extensive use in numerous academic and commercial applications [31]. The applications of SDR are diverse and include Spectrum Regulation and Cost Reduction [32], GPS Signal Reception [33], Driver Assistance [34], Interpretation of Cellular Technology [35, 36] and Radio Frequency Emissions [37], Cognitive Radios [38, 39], Evaluation of Multi-Path Communications [40], Aviation Tests [41], and Cooperative Wireless Network Diversity [42].

## 2.2. Sine - Gaussian Signal

In the initial phases of the thesis, a Sine-Gaussian waveform, specifically the Morlet wavelet, was selected as the analog signal for the proposed system. The Morlet wavelet can be described as a point-wise multiplication of a Sine wave and a Gaussian envelope in the time domain, as depicted in Figure 2.6. The Sine wave has a period of 20 ns and a frequency of 50 MHz, while the Gaussian envelope has a duration of 375 ns, resulting in an image band between 0 Hz to 10 MHz in the frequency domain. This choice of signal is based on its desirable time-frequency characteristics, making it suitable for various signal processing applications.

Sine-Gaussian waves are characterized by two important parameters, which are the central frequency and the quality factor Q. The central frequency is determined by the sinusoidal component, which has a frequency of 50 MHz in this particular case.

The frequency domain representation of the three given signals is shown in Figure 2.7, where the peak of the spectrum, which corresponds to the center of the ”sine” and ”sine-gaussian” waveforms, indicates the central frequency value of 50 MHz. It is worth noting that the image of the Sine-Gaussian wave in the frequency domain can be shifted by adjusting the frequency of the sinusoidal component.

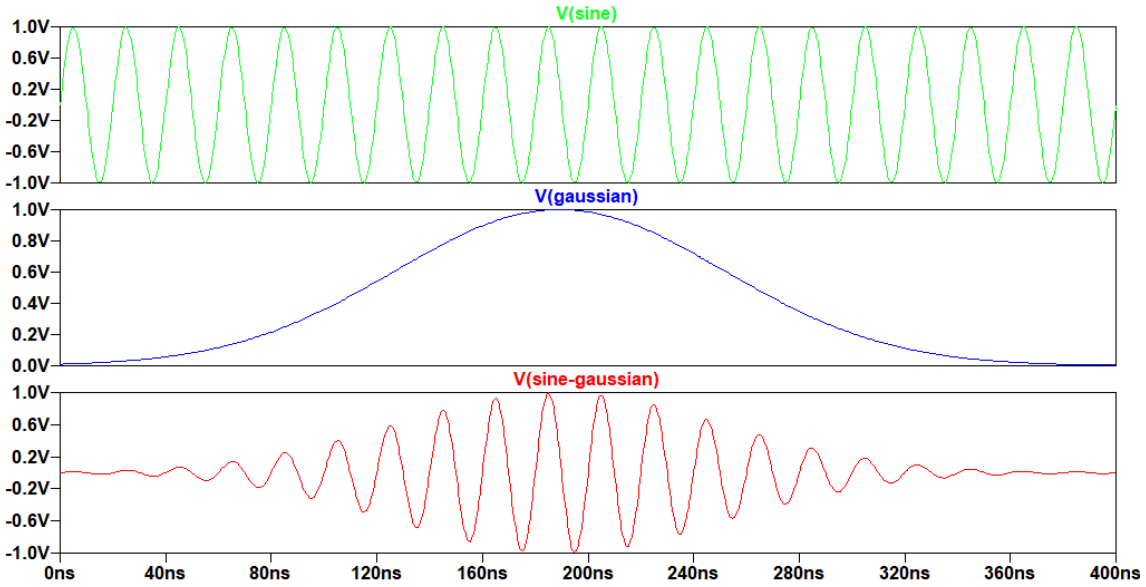



Figure 2.6. Sine-Gaussian wave in the time domain.

The frequency representation of the Sine-Gaussian signal is much wider than the sinusoidal component, which can be expressed as a spike at an exact signal frequency in Figure 2.7. The reason for this dissimilarity is that the Sine-Gaussian signal is a sin wave that is tapered by a Gaussian envelope so amplitude and duration in time will not be the same as the sine wave. This envelope introduces non-stationary behaviors in terms of amplitude. As a result of these amplitude differences in signal, the shape will be wider instead of a spike because the Fourier transform (FT) requires energy at other frequencies in addition to the central frequency of a sine wave for representing amplitude variations.

Moreover, the frequency domain representation of Sine-Gaussian can be expressed as a shifting version of the Gaussian envelope peak to the center frequency of the sinusoidal component. Since the Gaussian envelope is a real-valued signal, the Fourier Transform's negative half is equal to the mirrored version of the positive frequency part. Although the negative part is redundant for most applications, for this particular case, it doubles the bandwidth and might be useful for analyzing frequency spectra. When the Gaussian envelope image is shifted to the central frequency, which is 50 MHz, the negative frequent images will be visible and the Sine-Gaussian image will be between 40 MHz and 60 MHz. Therefore, the Sine-Gaussian bandwidth is 2 times the Gaussian envelope bandwidth.

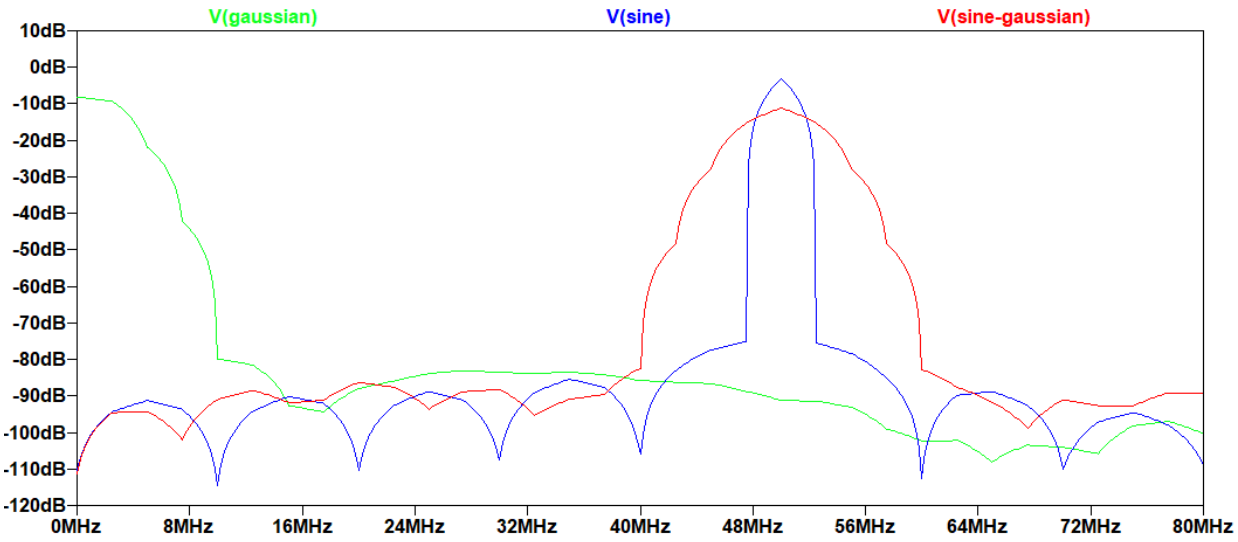



Figure 2.7. Frequency domain representation of Sine-Gaussian components.

And the other critical parameter Quality factor, Q is a relationship between the central frequency and bandwidth of a Sine-Gaussian wave, it can be defined in several ways such as the number of sinusoidal cycles within the Gaussian envelope or the ratio of central frequency to bandwidth as known from filters. Thus, the Sine-Gaussian bandwidth is related to the Q factor. Increasing the Q factor while keeping the central frequency constant will lead to a reduction in the bandwidth of the Sine-Gaussian wave, resulting in higher spectral resolution. On the other hand, decreasing the Q factor will broaden the bandwidth of the waveform, reducing its spectral resolution. In this work, the interested bandwidth is between 800 MHz and 1.5 GHz. Along with this given range, several signals that had different specs such as bandwidth used to examine the range.

While a circuit has been developed to generate Sine-Gaussian Signals and validated through simulations, the inherent structure of the employed data capture device is unsuited for high-frequency signals. Consequently, this particular design is not integrated into the current system. Nevertheless, there is potential for its application in a custom-designed system tailored for the same objective, as detailed in the forthcoming sections on future work.

## 2.3. Biology-Inspired Communication

In this section, a comprehensive summary of the research conducted by the Bountenna Research Group is presented. The focus is on two publications [16, 17] directly relevant to the scope of this thesis, providing insights into the type of system that SDR will be part of.

In recent years, there has been a significant focus on biology-inspired communication, driven by advancements in nanotechnology. Two specific areas of interest are molecular communications (MC) at the nano-scale and body area networks (BANs) at the macro-scale, both of which show great promise in communications research, especially for health applications. However, a crucial challenge lies in establishing a connection between MC and BAN, acting as a bridge to seamlessly transfer information gathered at the nano- or micro-scale to the macro-scale BAN and its extensions [16]. This link is vital for unlocking the full potential of these technologies in various practical applications, especially in the context of improving healthcare systems. By addressing this gap, researchers in [16] introduced two distinct sensing concepts that leverage genetically modified bacteria to detect biological processes and specific chemical outputs generated by molecular communications.

The first design proposed involves an active implantable sensor that necessitates a power source and requires surgical removal after the monitoring period is completed. While active implantable sensors offer high potential and performance, their widespread use is hindered by several drawbacks. One of the primary concerns is biocompatibility, as many in-body sensors require surgical extraction to ensure that no harmful residues are left behind, posing potential risks to the patient's health. To address these limitations, a second design is introduced by the researchers, featuring a passive and biodegradable device. Unlike the active implant, this passive device operates without the need for an external power source. Instead, it utilizes the surrounding environment or external stimuli to function. The biodegradable nature of the device allows it to naturally degrade over time within the body, eliminating the necessity for surgical removal. By opting for a passive and biodegradable approach, this second design aims to overcome the challenges associated with active implants, making the monitoring process safer, more practical, and less invasive for patients.

In Figure 2.8, the schematics of both systems proposed by [16] are depicted. This work revolves around the concept of utilizing the biodegradation process of the sensor after its designated operation time has elapsed. To achieve a passive implant, a backscatter communications-based approach is employed.

Backscatter communications involve the use of a remote reader that emits an electromagnetic wave towards the passive device. The passive device, in this case, is the sensor that operates without any active power source. Instead, it relies on the energy received from the incident electromagnetic wave to carry out its functions. The passive device collects data during its operation and modulates it into the reflected electromagnetic wave. The remote reader then receives and interprets the modulated data from the reflected electromagnetic wave, effectively extracting the information collected by the passive device. This backscatter method enables communication with the passive device without the need for any internal power source, making it a feasible and energy-efficient approach for passive implants.

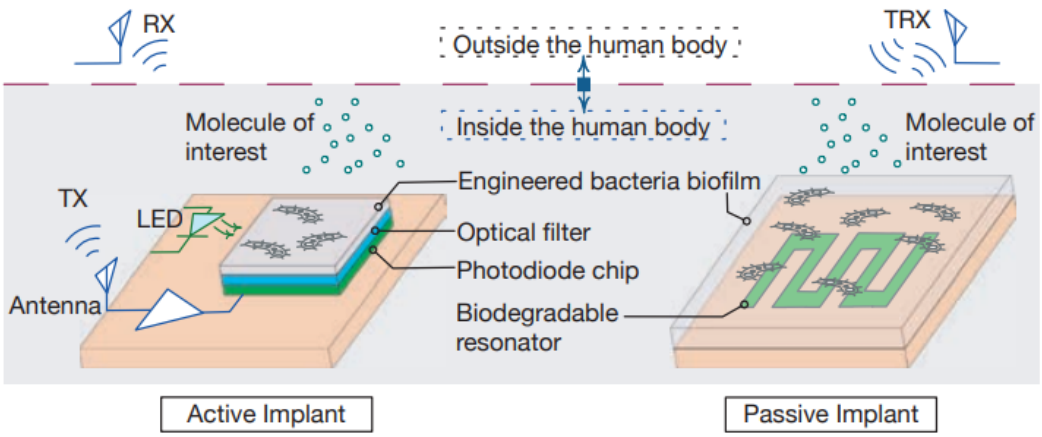



Figure 2.8. Schematics of proposed frameworks for molecular communication scenarios [16].

Both proposed mechanisms in [16] are making use of genetically engineered Escherichia coli (E. coli) bacteria. E. coli is a naturally occurring bacteria with a straightforward genomic and metabolic structure, making it highly amenable to growth and manipulation. As a result, it serves as the most abundant model and working organism in molecular biology research.

The passive sensing mechanism is designed to be battery-free and consists of a fully biodegradable magnesium (Mg) implant and a whole cell biosensor (WCB) of engineered E. coli embedded within. The E. coli bacteria act as sensors and detect the target molecule of interest. When the molecule of interest is detected, the engineered E. coli initiates a response, such as deactivating a gene that encodes a particular protein or metabolic pathway. This response leads to an increase in the degradation rate of the Mg implant. By monitoring the degradation speed of the magnesium implant, real-time in-body monitoring of biological processes can be achieved. The molecule of interest effectively triggers an elevated bacterial activity, resulting in faster biodegradation of the magnesium implant. The on-body transceiver, using backscatter communications, tracks the biodegradation speed and retrieves the data from the implant.

The primary goal of the passive sensing mechanism is to achieve real-time observation of the arrival of the molecule of interest from outside the human body. The concept relies on the fact that heightened bacterial activity leads to an increased biodegradation rate of the magnesium (Mg) implant. To achieve this, the Mg implant needs to be characterized through step-by-step degradation in an environment simulating the conditions inside the human body. The degradation process was observed wirelessly, taking into account scenarios with and without the presence of E. coli, which is illustrated in Figure 2.9. By comparing the electromagnetic responses of the Mg implant in these two conditions, the researchers can establish the correlation between bacterial activity and biodegradation rate. This correlation is critical for accurately monitoring the arrival of the molecule of interest and providing real-time in-body observations.

An implantable double split-ring resonator (DSRR) as a biodegradable implant and an on-body meandered slot loop antenna are designed in the proposed work [16]. A pair of these on-body reader antennas are located outside the setup given in Figure 2.10. The degradation of the Mg implant leads to an anticipated change in its resonant frequency, which can be estimated beforehand. This shift in resonant frequency will be detected by the on-body transceiver. Throughout the degradation process, the resonator's geometry will be partially preserved, resulting in a measurable shift in the resonance frequency, as illustrated in Figure 2.10.

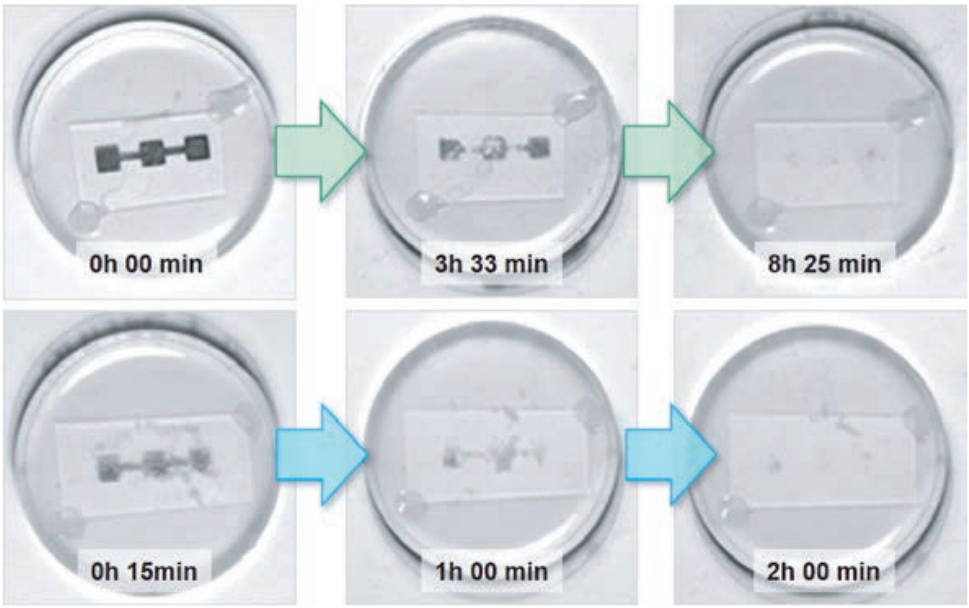



Figure 2.9. The effect of bacterial activity on biodegradation speed, without e.coli (top), with e.coli (bottom) [16].

To track this shift in the transmission spectra, a pair of on-body antennas is employed. These antennas establish a wireless link with the Mg implant and interact with it during its degradation. This wireless interaction allows for real-time monitoring of the molecule of interest inside the body. By observing the changes in the resonance frequency of the implant over time, the system can provide continuous in-body monitoring of the targeted molecule, enabling real-time observation of biological processes. Moreover, in addition to passive implant reconfigured by bacterial activities, researchers have found a second approach [17], a bio-hybrid device is utilized to establish a connection between MNCN (Molecular Nano Communication Networks) and BAN. The device incorporates engineered muscle tissue that responds to the molecule of interest by contracting [43]. This muscle tissue is employed to reconfigure the implant antenna in the proposed work.

Structurally, the bio-hybrid device consists of three main components: the engineered muscle tissue, a 3D-printed flexible scaffold, and the implant antenna. The muscle tissue, which is engineered to respond to specific stimulants, contracts upon detecting the molecule of interest. The sensing mechanism in the bio-hybrid device operates similarly to the bacterial passive implant. However, the method of response to molecular activities differs. Two proposed concepts for the bio-hybrid device are presented in Figure 2.11.

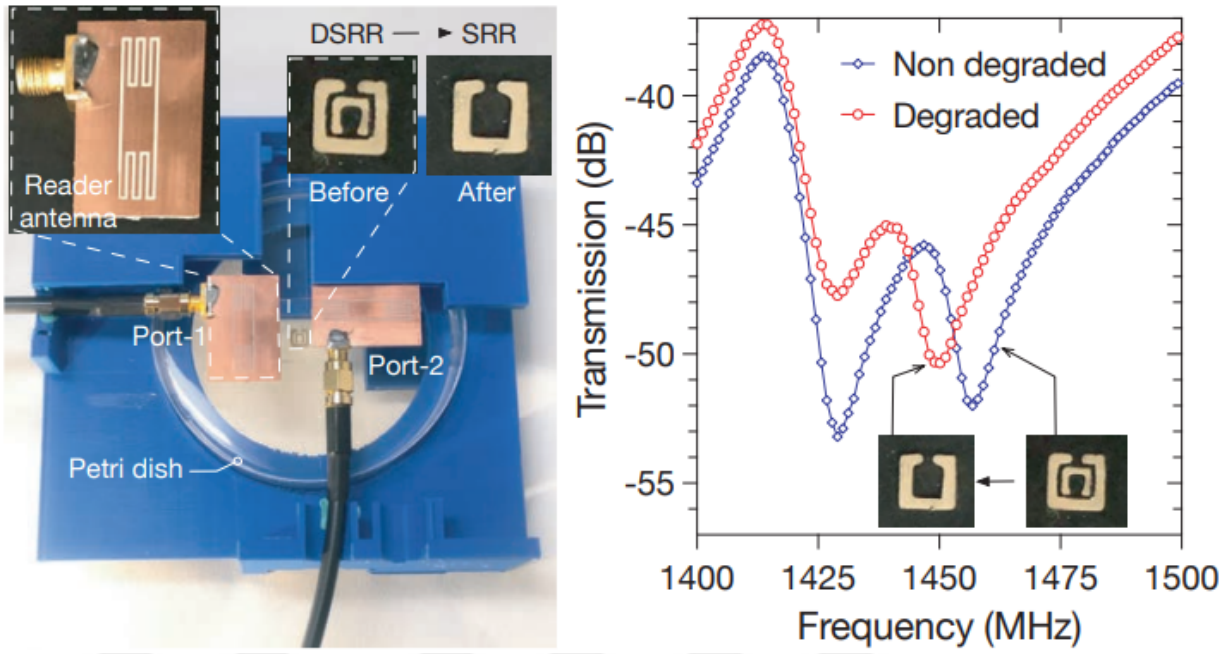



Figure 2.10. Measurement setup for passive in-body sensing system consisting antennas, resonators at two different phases (left), and initial experimental results demonstrating the change in scattering parameter S21 in dB (right). [16].

During the measurement phase of each experiment, a vector network analyzer is utilized to test the proposed designs. However, using VNA in a wearable medical diagnosis system poses challenges due to its shape and high cost. As a solution, this thesis integrates Software-Defined Radio, a portable and cost-effective device, into the existing systems. This integration allows for more practical and accessible implementation of the proposed sensing mechanisms in wearable medical diagnosis applications.

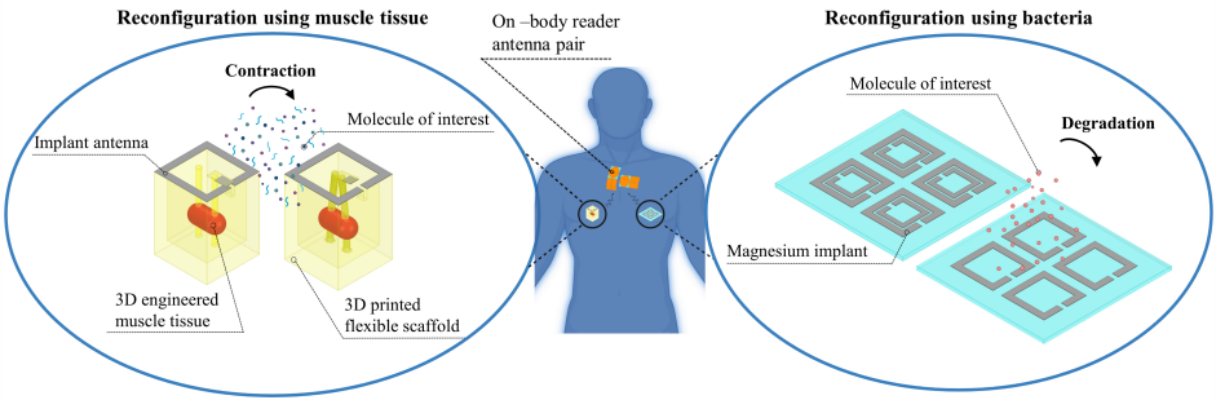



Figure 2.11. Demonstration of proposed concepts, reconfiguration using muscle tissue (left) and bacteria (right) [17].

# 3. HARDWARE AND SOFTWARE TOOLS

This chapter serves as a resource by providing detailed information about the SDR hardware used in the research, specifically the Adalm-Pluto device. It offers an in-depth explanation of the hardware specifications, capabilities, and functionalities of the Adalm-Pluto SDR, allowing readers to gain a comprehensive understanding of the chosen implementation platform.

Additionally, the chapter covers the software tools utilized to configure and program the Adalm-Pluto SDR. It provides insights into the specific software environments, such as GNU Radio, MATLAB, and Python programming language, which were employed in the development and customization of the SDR system. This information enables readers to comprehend the software aspects of the research and the tools utilized to optimize and control the functionality of the SDR device.

Furthermore, the chapter delves into the essential details of the post-processing tools employed in the research. It sheds light on the specific techniques, algorithms, and methodologies used for analyzing and manipulating the data collected by the SDR device. This information is crucial for understanding the data processing steps performed in the research and the subsequent analysis and interpretation of the results.

## 3.1. ADALM-PLUTO SDR Hardware

Pluto SDR has gained significant attention as a low-cost, medium-performance solution for a wide range of applications, spanning from amateur radio to GPS receivers. The underlying architecture of modern SDR devices is commonly structured around a basic block diagram, comprising three main components, as elaborated upon in the background section. Two of these components are depicted in Figure 3.1.b, namely the AD9363 block, which denotes the analog part, and the Xilinx Zynq block, which pertains to the digital domain. Meanwhile, the USB connection illustrated in Figure 3.1.a facilitates communication between the SDR module and the third component, which is the PC. Notably, the manufacturer Analog Devices Inc. provides datasheets that highlight Pluto SDR's key spec-

ifications. These include factors such as its maximum RF bandwidth, signal-to-noise ratio, and transmission power, among others, some of which are listed below;

- TX - RX Specs (AD9363) [44]
  (i) Wide bandwidth: 325 MHz to 3.8 GHz (can be extended to 70 MHz to 6 GHz due to software modifications)
  (ii) 2.4 Hz LO step size
  (iii) Tunable channel bandwidth (BW): up to 20 MHz (can be extended to 56 MHz due to software modifications)
  (iv) Tx Power Output 7 dBm
  (v) Rx Noise Figure $< 3.5$ dB
  (vi) Rx and Tx Modulation Accuracy (EVM) –34 dB (2%)
- Data Converter Specs (AD9363) [44]
  (i) 65.2 kSPS to 61.44 MSPS (CMOS - LVDS Interface Bottleneck) 12-bit ADC and DAC
  (ii) 5 Hz sample rate step size
- Digital Specs [45]
  (i) USB 2.0 On-the-Go (OTG) 480 Mbits/seconds
  (ii) ARM Cortex -A9 @ 667 MHz Processor
  (iii) 28k FPGA Logic Cells
  (iv) 80 DSP Slices
  (v) DDR3L 4 Gb (512 MB)
  (vi) QSPI Flash 256 Mb (32 MB).

The AD9363 transceiver is a crucial component in managing various operations within the RF front-end and analog baseband processes. It handles tasks such as data conversion and certain signal processing operations. Figure 3.1.b illustrates the components involved in these procedures, which can also be observed in the receiver path presented in Figure 3.2.

The Pluto SDR employs a homodyne architecture, eliminating the need for an intermediate frequency stage. Instead, received signals are directly converted to baseband after the low-noise amplifier stage. The LNA gain is software-controlled, allowing for the adjust-

ment of multiple gain regimes based on the received signal type, as discussed in subsequent sections.

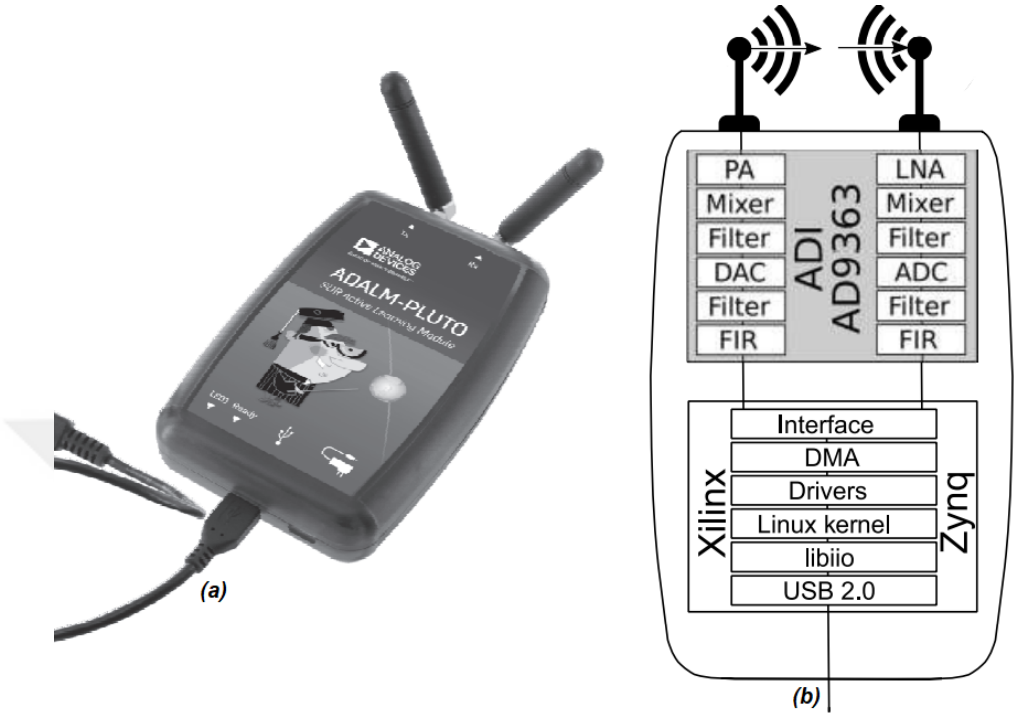



Figure 3.1. (a) ADALM-Pluto SDR module, (b) analog and digital blocks on the SDR [45].

Once the signal passes through the LNA stage, it is split into two parallel paths to process the in-phase (I) and quadrature (Q) signals separately. Each path includes two mixers, responsible for direct frequency conversion and extraction of the I/Q signals. This process involves multiplying the RF signals, ranging from 325 MHz to 3.8 GHz, with a local oscillator signal at a frequency equal to the RF signal frequency. The mixers are followed by transimpedance amplifiers (TIA) and low-pass filters to determine the bandwidth to be digitized by the ADC. The TIA acts as a single pole filter, while the LPF is a third-order Butterworth filter [46]. This combination effectively mitigates aliasing effects and noise interference.

The Analog to Digital Conversion process in the Adalm-Pluto SDR is performed by 12-bit ADCs, as depicted in Figure 3.2. Although these ADCs can operate at higher speeds (e.g., ADC 640 MSPS and DAC 320 MSPS), the data sheet specifications indicate the sample rate at the LVDS/CMOS interface to be 61.44 MSPS. It should be noted that

during signal reception, the user can access the samples at the output of the decimation filters. Therefore, the ADC output sample rate is not significant at this stage.

The final stage of the receiver side process involves decimation, which is achieved through digital filters such as half-band (HB) and Finite Impulse Response (FIR) filters. Each filter has its own decimation factor, which can be controlled by software. This enables the reduction of the ADC data rate to more manageable rates within the digital domain.

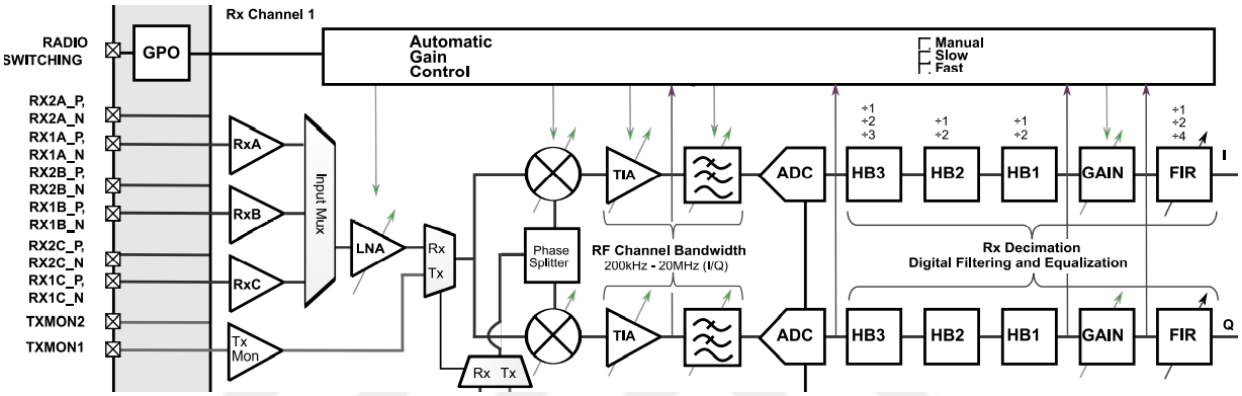



Figure 3.2. AD9363 transceiver receiver path [55].

On the transmitter side, the signal processing steps follow a slightly different sequence compared to the receiver side. However, a comparison of Figures 3.2 (receiver path) and 3.3 (transmitter path) reveals that the same fundamental operations are performed, but in reverse order.

In the transmitter path, interpolation is employed instead of decimation. Interpolation is used to increase the data rate of the digital-to-analog converter (DAC) input. This higher data rate allows for the generation of signals with higher frequencies at the output of the DAC. By increasing the data rate through interpolation, the transmitted signal can cover a wider frequency range.

Additionally, an attenuator is placed at the end of the transmit chain to regulate the strength of the transmitted signal. This attenuator allows for control over the transmitted signal power. To achieve the optimal signal-to-local oscillator ratio, the DAC is typically operated as close to full scale as possible [55]. This approach ensures that the transmitted

signal strength can be adjusted effectively using the attenuator. By employing interpolation and utilizing an attenuator, the transmitter side of the system can generate signals with adjustable power levels and cover a wide range of frequencies.

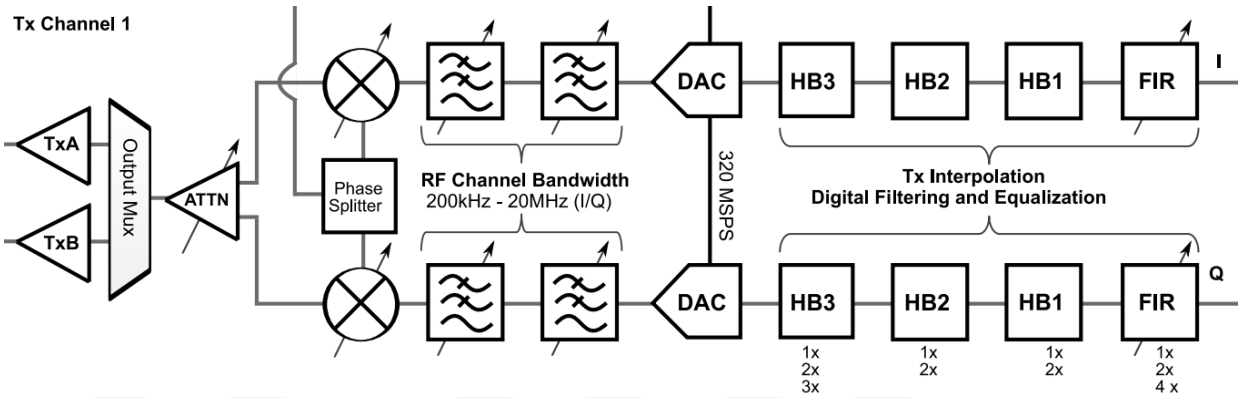



Figure 3.3. AD9363 transceiver transmitter path [55].

Once the received data is converted to the digital domain, it undergoes processing by the Xilinx Zynq Z-7010 System on Chip (SoC), which serves as the main controller in the system. The SoC performs various operations on the digital samples, including interpolation and decimation, DC correction, and I/Q correction, among others. These processes are designed to enhance the quality and accuracy of the received data.

The processed samples are then stored in the external DDR3L memory, which acts as a buffer for data storage and facilitates the operation of the Linux operating system. The DDR3L memory provides a reliable and efficient means of storing digital samples for further analysis and processing.

Additionally, the system includes a flash memory, which holds important information such as the root password and SSH key. This memory serves as a secure storage location for critical system configurations and authentication credentials. Moreover, the Pluto automounts USB storage devices, it looks like shell scripts or files that can be run on an ARM processor. By providing power and a binary file, Pluto can be used remotely.

The simplified block diagram in Figure 3.4 illustrates the connections between the analog and digital domains within the overall system architecture. It highlights the flow

of data and the interaction between different components. It is important to note that the configuration of the digital part is typically handled by software programs such as MATLAB or GNU Radio. These programs automatically configure the digital components based on the parameters provided by the user, simplifying the setup and operation of the system.

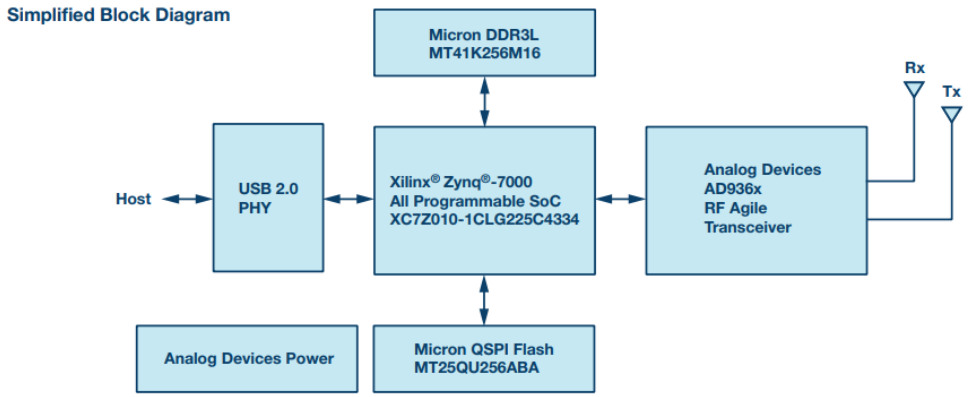



Figure 3.4. Simplified block diagram of Pluto SDR [45].

In the subsequent subsections, important processes utilized in both the transmitting and receiving operations are examined in detail. These processes play a critical role in ensuring the accurate and efficient transmission and reception of signals.

### 3.1.1. Modulation Basics

Quadrature or in-phase/quadrature (I/Q) signals form a fundamental basis for the proper operation of software-defined radios with various modulation types. To delve into the modulation process in SDRs, let's consider a sinusoidal wave, denoted as s(t), as shown in Figure 3.5. Depending on the chosen modulation type, we can modify the attributes of this signal, including amplitude (A), frequency ($f$), or phase ($\phi$). These attributes can be altered according to the picked modulation type since all modulation types are just functions of these three qualities that can be seen in the following general time expressions for sinusoidal waves and the given signal respectively,

$$y(t) = A \times sin(2\pi ft + \phi) \qquad s(t) = sin(200\pi t). \tag{3.1}$$

For example, taking the constant amplitude term from the general equation (3.1) and making it a function of a time as A(t), which will be used in amplitude modulation as an

envelope. If A(t) is multiplied by a high-frequency sinusoidal function whose amplitude is modulated according to A(t) as shown in the following equation

$$m(t) = (A(t) + offset) \times cos(2\pi f_c t + \phi). \tag{3.2}$$

In this expression, the offset term represents a DC bias added to the envelope, which helps in avoiding the phase reversals. When the envelope crosses zero, it can cause a 180-degree phase shift, known as over-modulation. This can lead to distortions in the received signal, such as reversed peaks. To address this issue, the offset term is introduced to provide a DC bias to the envelope. By introducing a non-zero offset, the envelope stays above zero, preventing phase reversals and avoiding distortions in the received signal. This is particularly important when using a basic envelope detector, as the diode used in the detector cannot sense negative peaks when no DC bias is present.

In equation (3.2), the cosine component represents the RF carrier signal, which is modulated with the time-varying envelope A(t). The carrier signal carries the high-frequency information, while the envelope modulation encodes the desired information in the amplitude variations of the carrier signal.

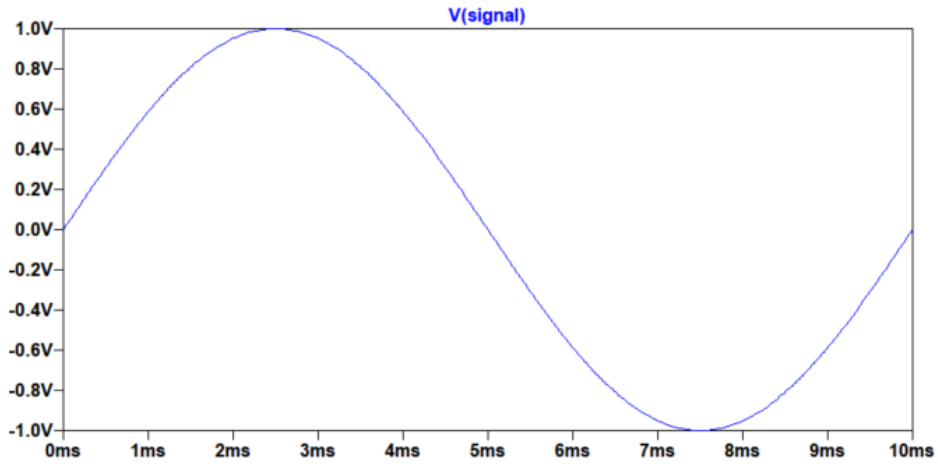



Figure 3.5. 100 Hz sinusoidal signal with 1 V amplitude - s(t).

Fundamentally, the high-frequency carrier's amplitude, in this particular case cosine component's amplitude, changes in accordance with the low-frequency message signal, A(t), which forms an envelope for high-frequency signals to fit in. As shown in Figure 3.6, the 1000 Hz carrier signal is modulated with 100 Hz message signal (1.2 V DC bias is added) and results in AM signal as given modulated.

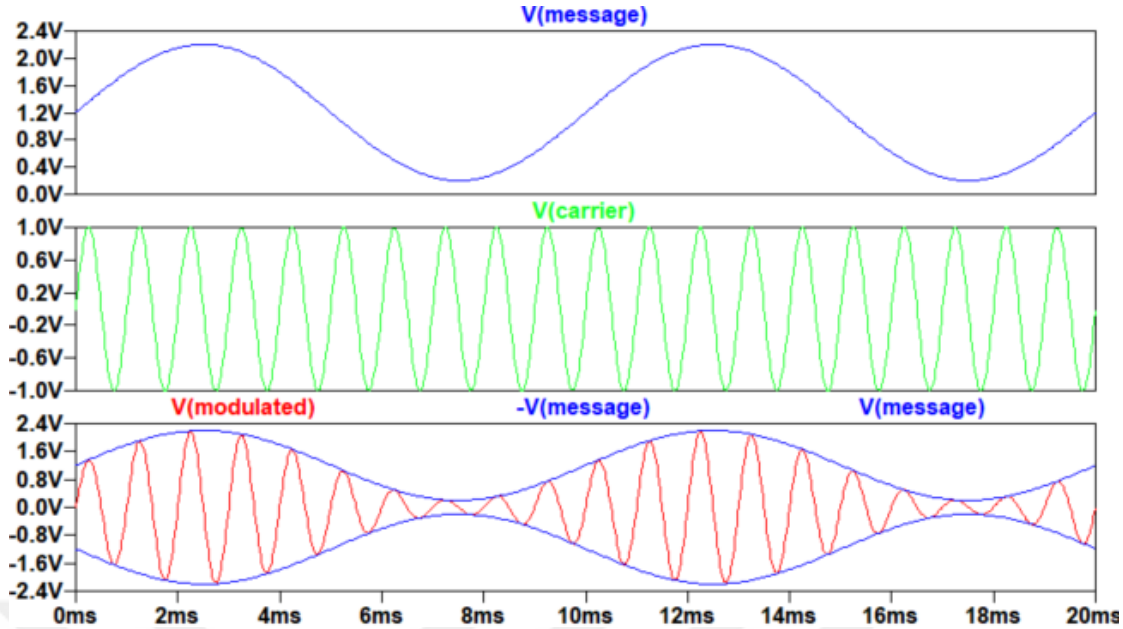



Figure 3.6. Used signals in modulation.

### 3.1.2. Phase and Quadrature Signals

The phase difference between two quadrature signals is 90 degrees, quarter cycle. Cosine and sine signals are quadrature waveforms where cosine leads sine by 90 degrees; therefore, the cosine component of a signal is called as in-phase and sine component is called as quadrature. The general form of I and Q signals are expressed as

$$I = I(t) \times cos(2\pi ft) \qquad Q = Q(t) \times sin(2\pi ft). \tag{3.3}$$

In the quadrature modulation process, the I(t) and Q(t) signals serve as inputs to the I/Q modulator, as shown in Figure 3.7. The I signal is generated by multiplying the input signal with a local oscillator signal, typically a cosine waveform, to obtain the in-phase component. The Q signal is obtained by applying a 90-degree phase shift to the local oscillator signal, resulting in a sine waveform.

The I and Q signals represent low-frequency signals that carry the desired information to be modulated onto the RF carrier. These signals can be any analog or digital signals that encode the desired information. By manipulating the amplitude and phase of the I and Q signals, different modulation types can be implemented. The adder block in Figure 3.7 combines the I and Q signals, resulting in the modulation of the RF carrier signal. The output of the adder represents the modulated RF signal, which can be any desired RF

signal based on the characteristics of the input I and Q signals. This modulation process allows for the generation of various RF signals, enabling the transmission of different types of information over the airwaves.

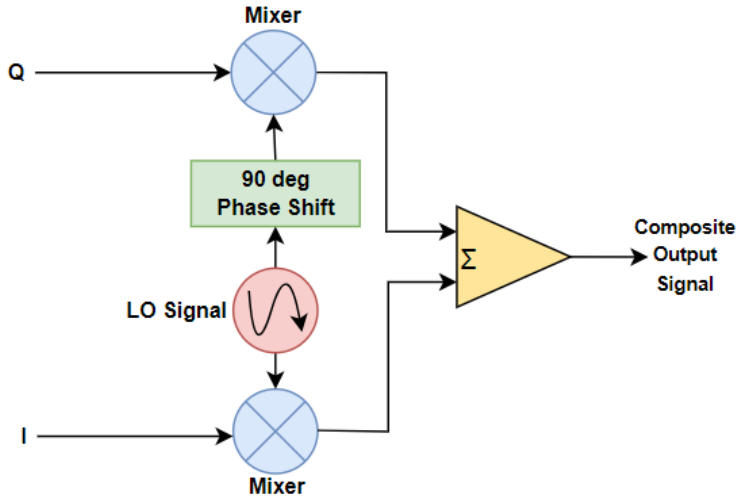



Figure 3.7. Quadrature modulator.

Figure 3.8 demonstrates the concept of amplitude modulation using quadrature signals. In this example, both the I(t) and Q(t) signals have the same amplitude, which is 1 according to equation (3.3). The output signal, obtained by adding the I and Q signals, exhibits a phase shift of 45 degrees compared to the original quadrature signals.

When the amplitudes of both I and Q signals are varied identically, the resulting output signal's amplitude will also vary accordingly. This means that the amplitude of the output signal can be modulated by simultaneously varying the amplitudes of I and Q signals in a synchronized manner. This modulation technique corresponds to amplitude modulation, where the information is encoded in the variations of the signal's amplitude.

Figure 3.9 illustrates the concept of phase modulation (PM) and frequency modulation (FM) using quadrature signals. In this example, the I(t) value is set to 0.5, while the Q(t) value is set to 2, according to the given equation in (3.3). Due to the higher amplitude on the Q side, the resulting signal is shifted towards the Q signal, introducing a phase difference, which corresponds to phase modulation.

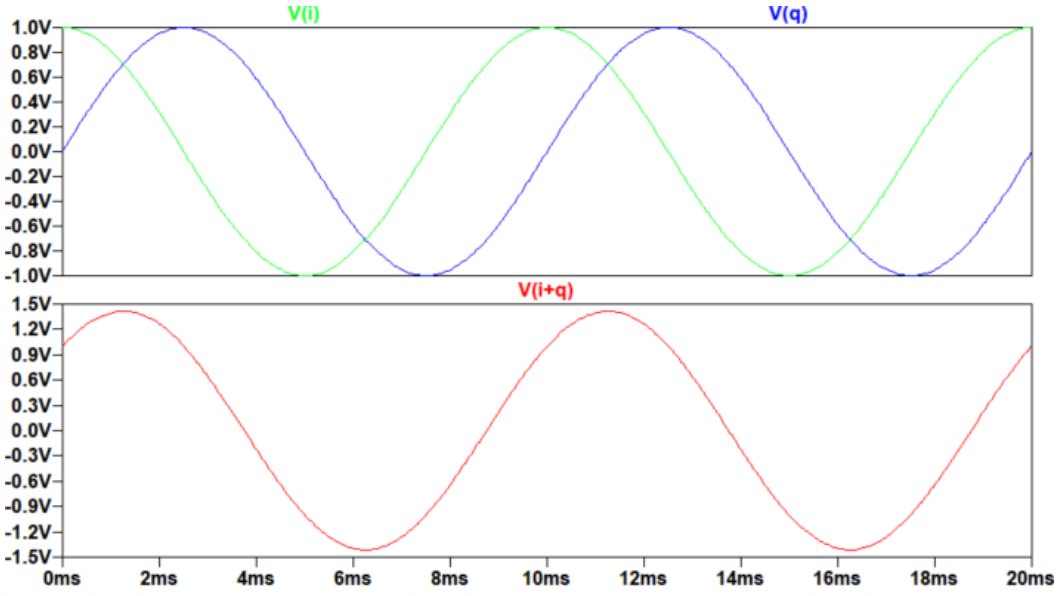



Figure 3.8. In-phase, quadrature components of a complex signal and amplitude modulated sum.

To achieve phase modulation, the I(t) and Q(t) values need to be varied differently. By setting one of the values to 0 and the other to 1, the resulting signal will have a phase shift of either 0 or 90 degrees, depending on the values of I and Q. This phase shift can be interpreted as phase modulation, where the phase of the resulting signal is modulated according to the values of I and Q.

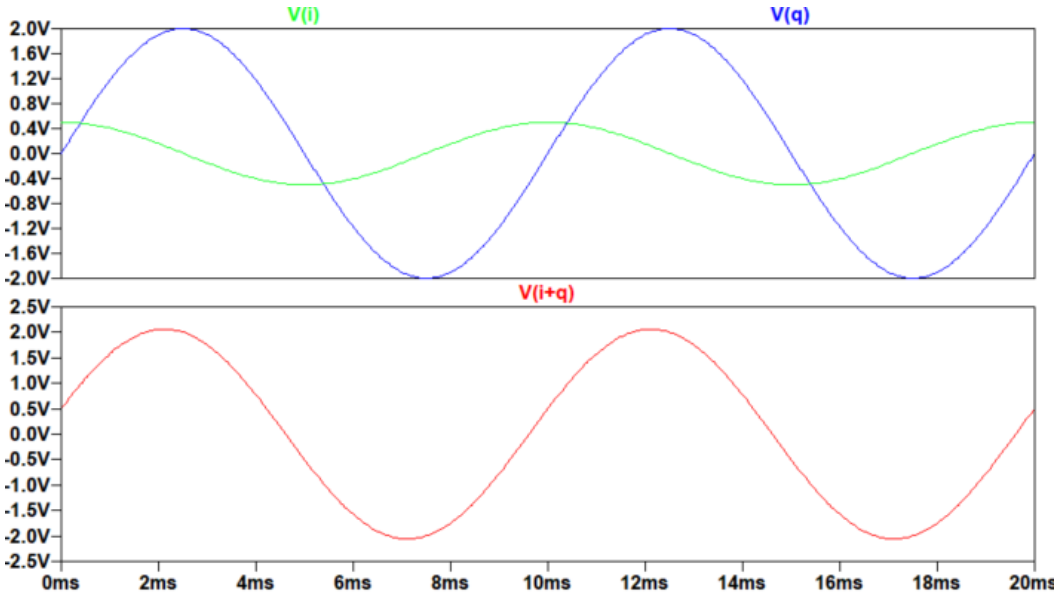



Figure 3.9. In-phase, quadrature components of a complex signal and amplitude, phase modulated sum.

Similarly, frequency modulation operates based on the concept of phase modulation. Both PM and FM are forms of angle modulation, where the modulated signal's phase or frequency, respectively, is varied in accordance with the modulating wave. By modulating the phase of the resulting signal, the instantaneous frequency of the signal can be varied. In FM, the modulated signal experiences compression during the positive half-cycle of the modulating sinusoidal wave, which leads to an increase in the instantaneous frequency of the resulting signal during that period. In PM, the modulated signal is compressed when the modulating sinusoidal wave completes one period of oscillation, the instantaneous angle is changed in accordance with the modulating wave. I/Q signals are indeed used for digital modulation, and one example is Binary Phase Shift Keying (BPSK). BPSK is a simple digital modulation technique where the phase of the carrier signal changes between 0 and 180 degrees. By manipulating the I/Q signals, BPSK modulation can be achieved.

In BPSK modulation, either the I(t) or Q(t) component in equation (3.3) is set to 0, while the other component is represented as a pulse signal ranging between -1 and 1. The BPSK modulated signal can be obtained at the output of the adder. Figure 3.10 provides a visualization of the signals involved in BPSK modulation. In the first plot pane, the I(t) and Q(t) signals are shown, where I(t) is set to 0 and a pulse signal is given for Q(t). The next plot pane displays the outputs of the local oscillator, with cosine denoted as "$lo_i$" and sine denoted as "$lo_q$".

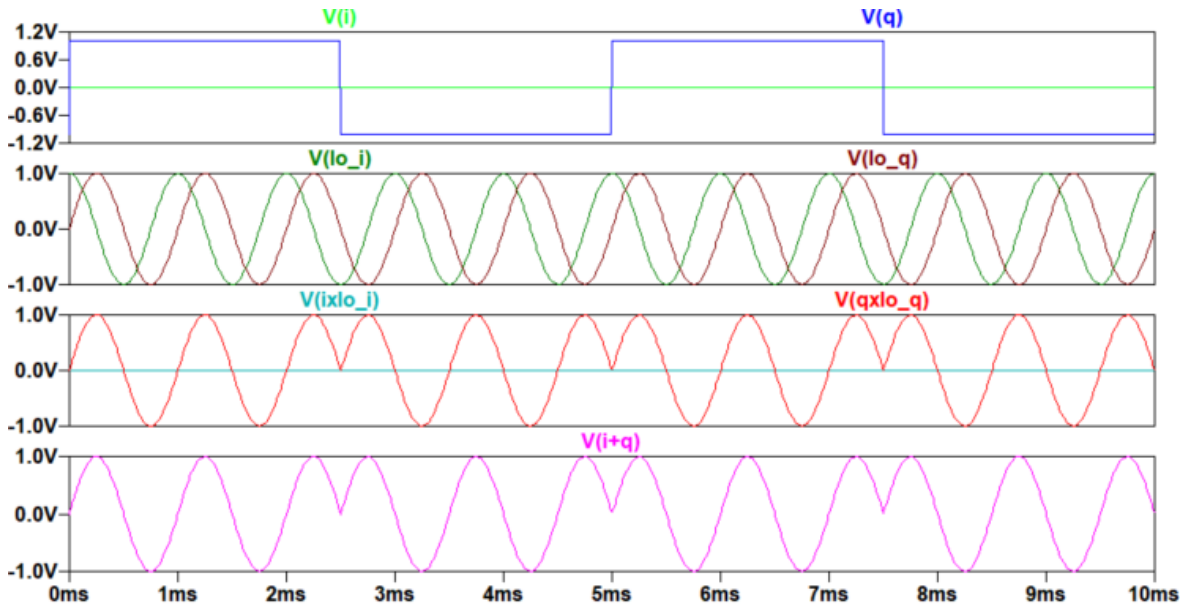



Figure 3.10. BPSK modulation example.

In the third plot pane, the pulse signal is multiplied by the sine signal. During the edges of the pulse, both positive and negative phase reversals can be observed. When the pulse's amplitude is 1, the resulting signal maintains the same phase as the local oscillator's sine signal. However, when the pulse's amplitude is -1, the resulting signal is 180 degrees out of phase compared to the "$lo_q$" signal.

Finally, the last plot pane shows the sum of the I and Q signals. Since I(t) is equal to zero, the resulting signal is identical to the red signal. This represents the BPSK modulated signal. Furthermore, a more complex digital modulation type Quadrature Phase Shift Keying (QPSK) in comparison to BPSK can be generated in the same way if the I(t) component is varied between -1 and 1 concurrently with the Q component.

Besides its usefulness in the modulation process, I/Q signals are used in complex or quadrature sampling, which is utilized in most SDRs to improve Nyquist criteria. Conventional sampling is actualized with the real signals, and sampling frequency should be at least 2 times greater than the highest frequency component of the signal to avoid aliasing.

If the Nyquist Criteria is not satisfied, negative frequency components may overlap with the positive spectral component, eradicating the original signal's reconstruction. Negative frequency components occur because of the nature of FT, which is defined as

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt, \tag{3.4}$$

where $e^{-j2\pi ft}$ can be written as $cos(2\pi ft) - j \times sin(2\pi ft)$ from Euler's formula and translates the equation given in (3.4) to

$$X(f) = \int_{-\infty}^{\infty} x(t)cos(2\pi ft) - j \times sin(2\pi ft)dt. \tag{3.5}$$

In complex or quadrature sampling, the Fourier transform produces a mirrored image of the real component at the negative center frequency due to the even nature of the cosine function. However, the lack of an imaginary component prevents the removal of this mirrored image.

As illustrated in Figure 3.11, the real component displays a symmetrical counterpart on the negative frequency side relative to the zero-frequency point, which is a noteworthy property of real-valued signals and arises from the inherent nature of real signals. Con-

versely, the negative frequency counterpart of the imaginary component exhibits symmetry with respect to the frequency axis but is reversed compared to its positive frequency counterpart due to the mathematical properties of trigonometric functions, as sin(x) is not equal to sin(-x), resulting in the unique appearance of the negative frequency image.

In the context of complex sampling, the presence of the imaginary component enables the elimination of the mirrored image of the real component. The receiver architecture, depicted in Figure 3.2, is a common design employed in most software-defined radios available in the market. This architecture comprises two identical signal paths for processing the in-phase and quadrature signals, which facilitates complex sampling. Let's assume that x(t) is received by the antenna. The frequency domain representation of x(t) is depicted in the initial components of Figures 3.12 and 3.13.

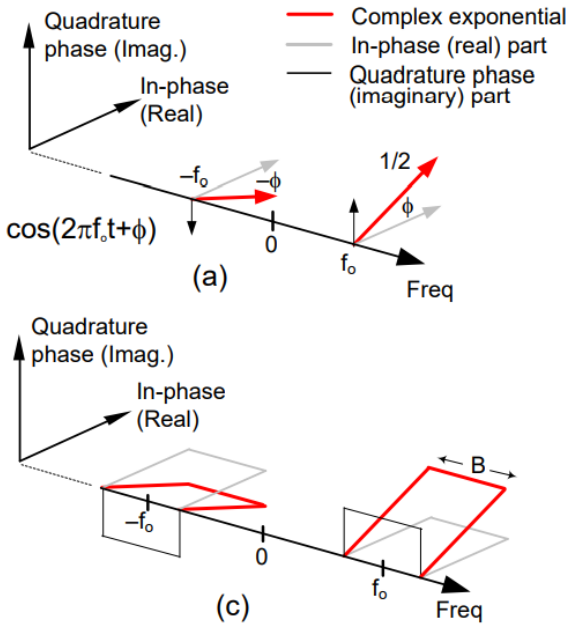



Figure 3.11. Quadrature representation of signals: (a) $f_0$ Hz sinusoidal signal, (c) continuous time real signal with BW B [47].

Before sampling, the received signal needs to be down-converted to the baseband in order to employ a more reasonable ADC sample rate. This down-conversion is achieved using a complex mixer, which can be observed in Figures 3.2 and 3.7. During the down-conversion phase, the received signal x(t) is mixed with cosine and sine signals, resulting in

the in-phase and quadrature signals presented as

$$X_I(t) = x(t) \times cos(2\pi fct) \qquad X_Q(t) = x(t) \times sin(2\pi fct). \tag{3.6}$$

As a result of mixing, the received signal's in-phase image is shifted down to ($f_c$ - $f_c$ = 0) baseband and shifted up to $2f_c$ ($f_c$ + $f_c$). Since there is also mirrored image of the received signal at -$f_c$ because of the cosine component in (3.6), it is also shifted up and down, which forms the second spectrum of Figure 3.12. After proper filtering, the remaining components are shown in the last spectrum, negative and positive frequency images are overlapping each other at the baseband.

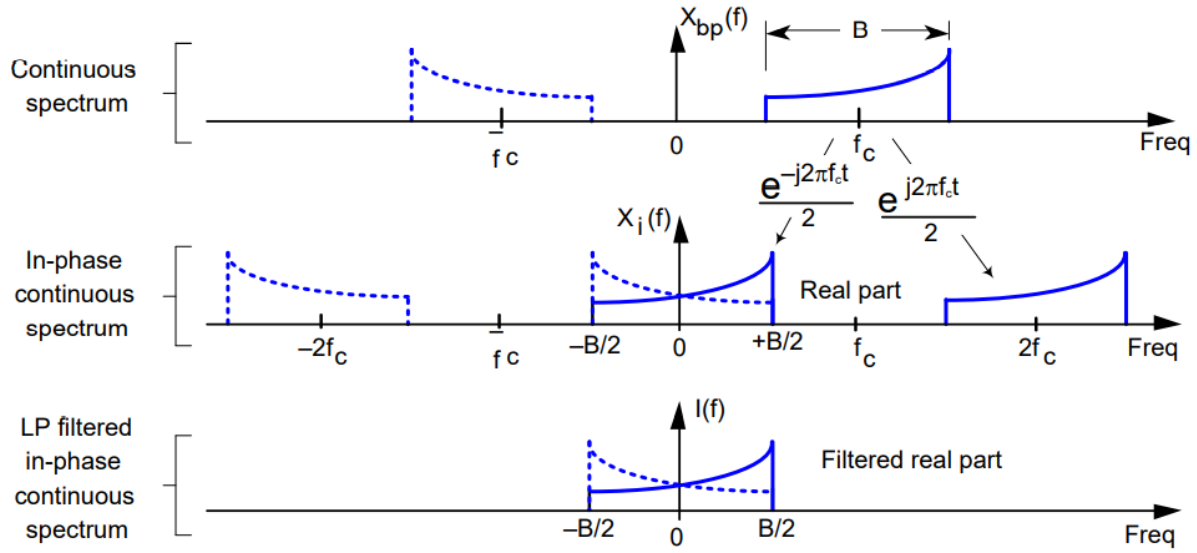



Figure 3.12. Resulting quadrature-sampling spectrums using the in-phase component [47].

However, for quadrature down conversion, the resulting spectrums are different than what the in-phase component produces. It is better to explain the process with mathematical arguments. The quadrature equation given in (3.6) can be converted to

$$X_Q(t) = x(t) \times \frac{1}{2j}(e^{j2\pi ft} - e^{-j2\pi ft}). \tag{3.7}$$

If both sides in (3.7) are multiplied by j, the result will be equal to

$$j \times X_Q(t) = x(t) \times \frac{1}{2}(e^{j2\pi ft} - e^{-j2\pi ft}). \tag{3.8}$$

Fourier transform by using duality and frequency shifting property of the right-hand side will produce the equation given below

$$X_Q(F) = \frac{1}{2}(X(f - f_0) - X(f + f_0)), \tag{3.9}$$

which creates the images that are given in the second spectrum given in Figure 3.13. The image at the negative frequency is now reversed in terms of amplitude.

After the frequency shifting process, a low-pass filter with a cut-off frequency of B/2 Hz is applied to both the in-phase and quadrature signals. This filter is designed to remove unwanted images and retain only the baseband components. The outputs of the low-pass filters are shown in the third spectrum of Figures 3.12 and 3.13.

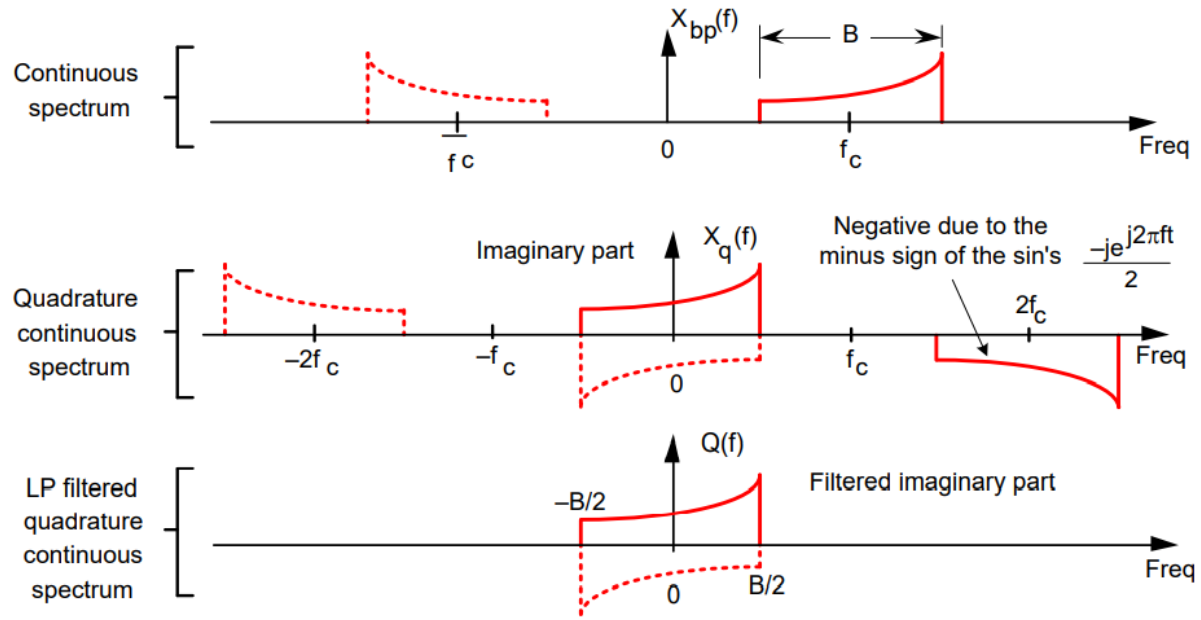



Figure 3.13. Resulting quadrature-sampling spectrums using the Q component [47].

When these spectra are added together, which occurs at the end of the receiver path, the mirrored images of the in-phase and quadrature components will cancel each other out. As a result, only the positive frequency component will remain, with its initial amplitude, as depicted in Figure 3.14.

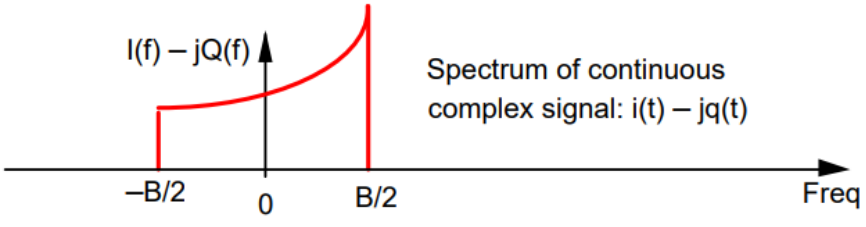



Figure 3.14. Addition result of filtered I and Q components [47].

The utilization of complex sampling in SDR systems brings several advantages and improvements compared to conventional sampling. These improvements are summarized in Table 3.1 as follows:

- Increased Unique Bandwidth: With complex sampling, the unique bandwidth is effectively doubled compared to conventional sampling. This is because there are no mirrored images in the negative frequency spectrum, allowing signals to be shifted down to negative frequencies.
- Higher Effective Sample Rate: Due to the removal of conjugate symmetric images, the sample rate can be set equal to the highest frequency component of the signal without causing aliasing. This allows for a more efficient and optimized sample rate selection.
- Improved Spectrum Efficiency: Complex sampling enables the representation of both positive and negative frequency components, resulting in a more compact and efficient representation of the signal spectrum. This leads to better spectrum utilization and improved signal processing efficiency.
- Enhanced Flexibility in Signal Processing: By working with complex-valued signals, complex sampling provides more flexibility in signal processing algorithms and techniques. It allows for a wider range of modulation schemes, demodulation techniques, and digital signal processing operations to be implemented.

Table 3.1. Comparison of classic sampling and complex sampling [48].

| **Parameter** | **Classic Sampling** | **Complex Sampling** |
|---|---|---|
| **Unique Frequency Range** | $0 \leq f \leq fs/2$ | -fs/2 $\leq f \leq fs/2$ |
| **Symmetry** | Conjugate Symmetric | None |
| **Min. Sampling Frequency** | fs | fs/2 |

### 3.1.3. Decimation and Interpolation

Sample rate and data rate are indeed specifications that can cause confusion, but they have distinct meanings in the context of data converters and signal processing. Sample rate refers to the speed at which an analog-to-digital converter samples the analog input signal

or a digital-to-analog converter produces analog samples at the output. It is typically measured in samples per second (SPS) or its multiples (kilo samples per second, mega samples per second, etc.). The sample rate determines the frequency range that can be captured or reproduced by the ADC or DAC.

The data rate, on the other hand, refers to the rate at which digital data is processed or transmitted. For an ADC, the data rate represents the rate at which the digital output samples are produced, and for a DAC, it represents the rate at which digital input samples are consumed. The data rate is usually expressed in bits per second (bps) or its multiples (kilobits per second, megabits per second, etc.). It reflects the amount of information being processed or transmitted per unit of time.

In many systems, including the Adalm-Pluto, the sample rate, and data rate are not the same due to the use of decimation (Digital Down Conversion - DDC) and interpolation (Digital Up Conversion - DUC) operations. The purpose of decimation is to reduce the data rate by removing samples from the original data, allowing the signal to be represented using fewer samples. This simplifies processing in the digital domain. On the other hand, interpolation increases the data rate by inserting zeros between the existing samples, without affecting the signal itself. DUC enables easier filtering by shifting images further in the spectrum and allows for higher output frequencies at the DACs, as the output frequency is proportional to the input data rate.

In the case of the Adalm-Pluto, the ADC operates at a sample rate of 640 MSPS (Mega Samples Per Second), and the DAC operates at a sample rate of 320 MSPS [55]. However, the data rate is limited to 61.44 MSPS for both the ADC and DAC. This reduction in data rate is primarily due to the constraints imposed by digital hardware components such as processors, FPGAs, or ASIC processes. Additionally, I/O devices or specifications, such as the LVDS/CMOS interface in the case of the Adalm-Pluto, may further limit the achievable data rate.

## 3.2. Software

### 3.2.1. Industrial I/O Subsystem

The Linux Industrial I/O subsystem (IIO) provides support for devices that can be considered as ADCs or DACs [49]. The IIO subsystem, which is part of the kernel, functions as a framework for input and output operations and is responsible for managing the drivers for each component on the AD9363 transceiver.

The drivers for the AD9363 components are implemented within the IIO subsystem. This allows users to control the transceiver by manipulating its parameters using simple Linux commands. These commands are typically executed in a shell environment. Figure 3.15 illustrates a standard command structure for manipulating a parameter, specifically for a variable gain amplifier (VGA). The Linux commands used, such as "cat" and "echo," modify the contents of the targeted registers on the hardware. These register contents control the behavior and settings of the device. File indicators associated with the VGA provide information to the user about the attributes of the device.

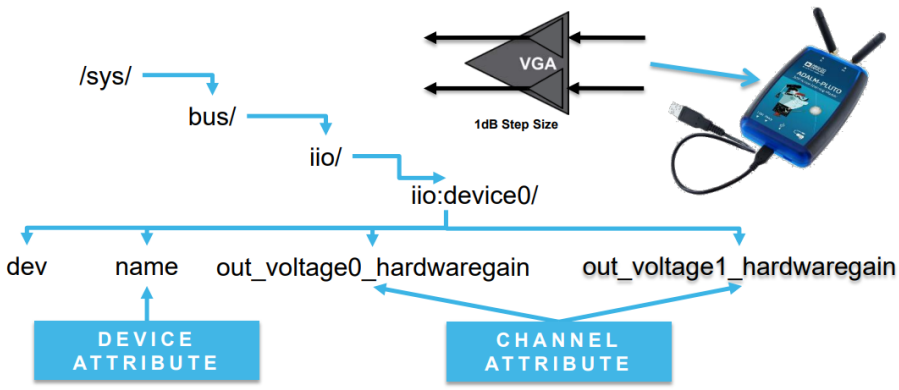



Figure 3.15. Structure IIO command [50].

The "device" attribute indicates the parameter that will be set, while the "channel" attribute specifies whether the component is used for receiving or transmitting. By leveraging the IIO drivers and using simple Linux commands, users can easily configure and control the AD9363 transceiver by modifying its parameters.

LibIIO is a library developed by Analog Devices Inc. that provides high-level APIs in C, C++, C#, and Python for controlling low-level attributes, making the process of interacting with devices more user-friendly [51]. This library allows direct communication with kernel drivers from the ARM processor on the Zync platform.

In addition to the high-level API, LibIIO offers several backends that enable remote communication between the library, the drivers, and the host computer. The local backend interfaces directly with the Linux kernel, while the network backend interacts with the IIO Daemon (IIOD) server. The IIOD server acts as a central hub where the context of the local backend is shared with any client application that connects to the server [52].

The architecture of LibIIO is depicted in Figure 3.16, illustrating the communication flow between the local backend, the network backend, the IIOD server, and the client applications. This architecture allows for remote control and monitoring of devices through the host computer, enhancing the flexibility and accessibility of the LibIIO library. In this architecture, the left side represents the system's System-on-Chip (SoC) part. The IIO devices block represents the FPGA and the AD9363 transceiver in the case of the Pluto platform. LibIIO interacts with the kernel directly and indirectly with the IIO devices.

On the board, there is a running daemon called IIOD, which is a background process. This daemon serves as an intermediary between the host application and the device. The host application, which can be software such as MATLAB or GNU Radio, is shown on the right-hand side of the diagram. The embedded ARM, which is the device itself, runs LibIIO and enables proper communication between the host and the device. The IIOD daemon performs API translation, bridging the communication between the host machine and the embedded ARM. It receives commands and values from the host application and updates the driver on the device with the provided values, allowing for control and configuration of the device from the host side.

In Figure 3.17, a comprehensive diagram is presented that illustrates the system-level process. Assume a scenario that the user generates a waveform using GNU Radio and intends to transmit it using the transceiver. The data needs to be pulled from the host (GNU Radio) to the transceiver. To facilitate this communication, GNU Radio utilizes

LibIIO to communicate with the IIOD server. LibIIO provides a high-level API that allows GNU Radio to interact with the server over various transmission protocols such as TCP/IP, Ethernet, and serial USB. Through this communication, the IIO drivers in the kernel are manipulated.

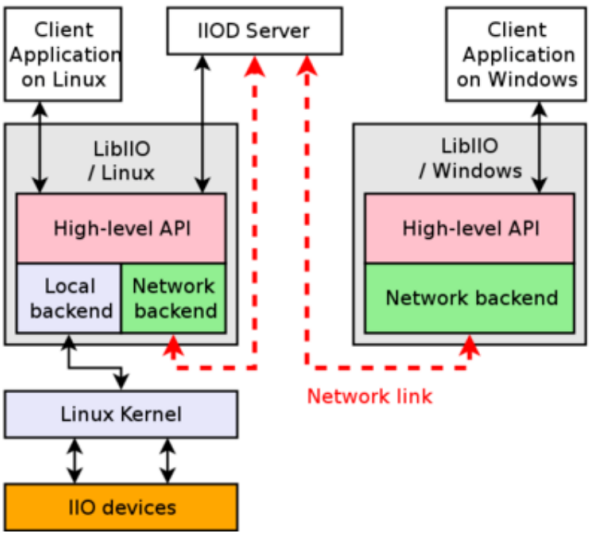



Figure 3.16. LibIIO diagram [52].

The kernel, in turn, communicates with the transceiver or FPGA, executing the commands received from the IIOD server. This process allows for the implementation of the desired functionality and configuration on the transceiver or FPGA based on the user's commands and waveform generated in GNU Radio.

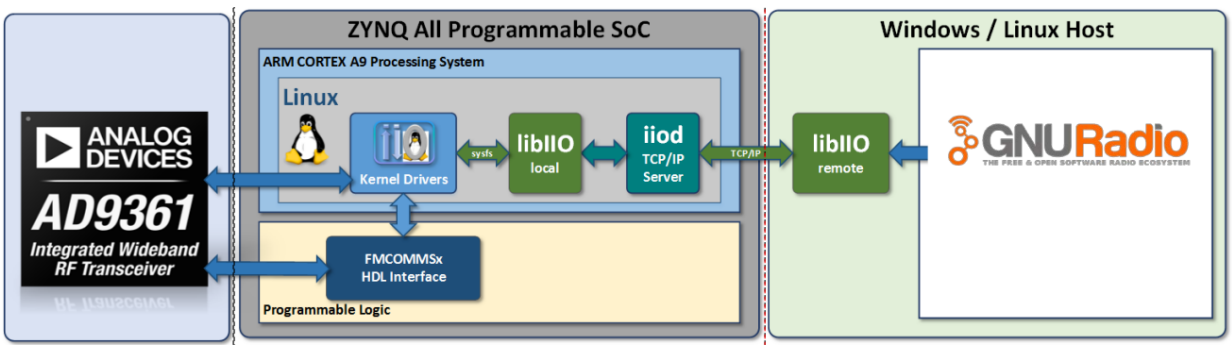



Figure 3.17. Pluto SDR system level diagram [50].

### 3.2.2. GNU Radio

GNU Radio is a popular and widely used software development toolkit for implementing software radios. It is a free and open-source platform that provides a comprehensive set of signal processing blocks for SDR development [53]. In GNU Radio, the user works with these signal processing blocks to design and build their desired system. The blocks are categorized into various types, including waveform generators, modulators, math operators, channel models, and filters. The design process takes place in the graphical tool called GNU Radio Companion (GRC), where the user can visually connect these blocks to construct the overall system. The user establishes connections between the blocks, defining the flow of data and signal processing operations. Once the design is complete, it can be deployed to the hardware using the LibIIO library, which enables communication with the underlying hardware components.It's worth mentioning that GNU Radio is not solely dependent on hardware components. It can also perform signal processing operations on existing data within the host computer, making it versatile for simulation and testing purposes.

In terms of programming languages, GNU Radio predominantly uses C++ for modeling the signal processing blocks that execute the workload [49]. However, users have the flexibility to choose Python as the output language thanks to GRC's support. Additionally, users can implement custom blocks using both C++ and Python, providing further flexibility and extensibility to the toolkit.

### 3.2.3. MATLAB

MATLAB is a widely-used software environment for advanced signal processing of received signals in SDR. It offers a range of tools and functions specifically designed for signal processing, communications, and data analysis, making it an ideal choice for SDR applications. With MATLAB, users can perform various signal processing operations such as filtering, demodulation, decoding, channel estimation, and synchronization. In this work, curve fitting and the signal processing toolbox are widely used.

# 4. IMPLEMENTATION OF THE SYSTEM

## 4.1. Extraction of Transmitted Signal

The original plan was to employ a Sine-Gaussian signal for capturing internal changes within the body. However, utilizing a Sine-Gaussian signal with a frequency order in the gigahertz range is not feasible in the context of a Software-Defined Radio system. Digital signal processing operations require higher data rates when working with gigahertz-frequency output signals than baseband signals. Moreover, the SDR architecture, particularly in devices like the Adalm-Pluto SDR, is ill-suited for processing signals beyond the baseband range due to the presence of down-conversion and up-conversion stages between the analog and digital blocks. As a result, it becomes necessary to generate a baseband signal within the GNU Radio Companion software. This baseband signal is designed to have a bandwidth of 10 MHz, adhering to the constraints of the SDR system. The process involves utilizing a flowgraph, illustrated in Figure 4.1, which comprises four distinct blocks. These blocks will be briefly elucidated in the subsequent paragraph to provide a better understanding of the methodology employed.

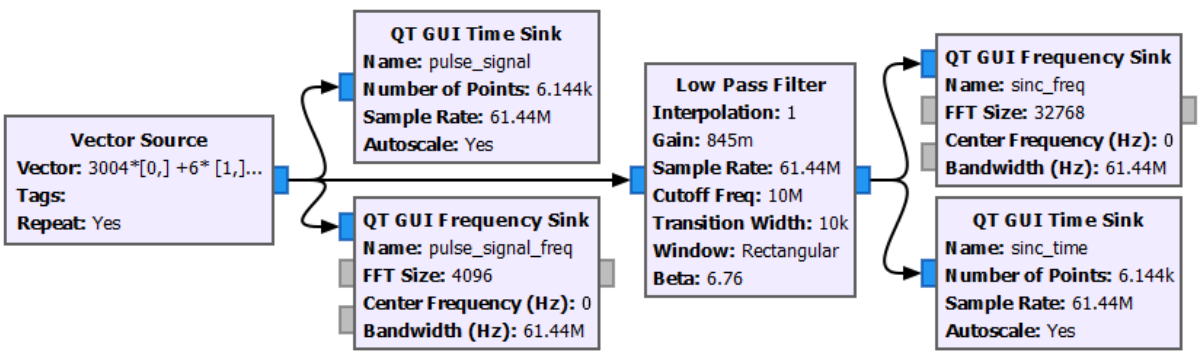



Figure 4.1. TX sinc signal flowgraph.

In this specific case, the Vector Source block is utilized to extract a single pulse burst. It operates by streaming items based on one or more given vectors [54]. To generate the desired signal type, three 1-element vectors are employed. Each vector is multiplied by a specific coefficient, resulting in the formation of 6144 complex samples. Although the

samples are complex, their imaginary component is set to 0, meaning that the vector values modify only the real part of the signal.

In the context of Adalm Pluto, which accepts complex signals, the Vector Source block directly streams complex samples to avoid the need for a float-to-complex converter block. This choice is made to simplify the flowgraph. The first coefficient used is 3004, which is multiplied by the first vector consisting of a single 0 element. The subsequent coefficient is 6, and it is multiplied by the second vector, which has a value of 1. Finally, the last vector, which contains 0 as its sole element, is multiplied by 3144, resulting in a total of 6144 samples.

The length of the vector is chosen to be a multiple of Adalm-Pluto's sample rate, which is 61.44 MSPS. This selection ensures that the device buffer is properly filled and can handle the streaming data effectively.

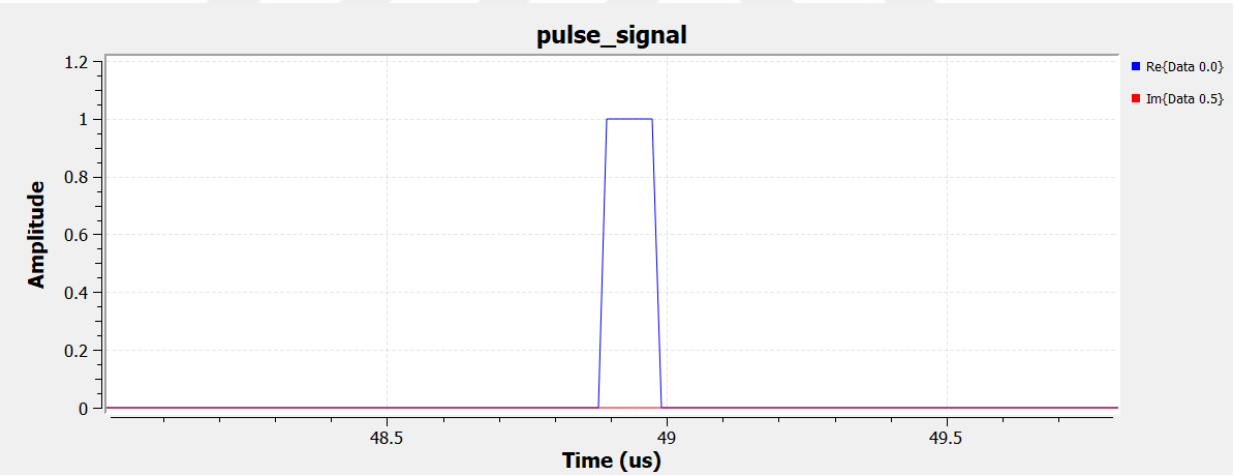



Figure 4.2. Time domain representation of the pulse.

Figure 4.2 displays the time domain representation of the pulse signal using the "QT GUI Time Sink" block. The real part of the signal is represented by the blue waveform, while the imaginary component is depicted by the red constant line at 0. Several parameters need to be configured for the time sink prior to executing the flow graph, but only four of them are observable in Figure 4.1, and two are explained in the following sentences.

The "Number of Points" parameter determines the number of samples of the signal that are plotted on the screen. In this case, it is set to 6144 to plot only one pulse burst. If the value is increased, the signals can be concatenated. For example, setting it to 61440 would result in 10 pulse peaks being plotted.

Another significant parameter is the "Sample Rate", which defines the time interval between each signal point, essentially determining the unit of the time axis. The sample rate of the time sink is set equal to Adalm-Pluto's sample rate, which is 61.44 MSPS. This corresponds to a time difference of 0.0163 microseconds between each set of 6144 samples, and the unit of the time axis is denoted as microseconds in Figure 4.2. Therefore, based on the given information, the pulse signal's on/high period can be calculated as

$$T_{ON} = (0.5 \times 0.0163) + (5 \times 0.0163) + (0.5 \times 0.0163) = 0.0978\,\mu s, \tag{4.1}$$

where the first and the last elements represent zero-to-one and one-to-zero transitions respectively. The other element expresses the duration between 6 one/high valued samples.

By using $T_{ON}$ from (4.1), the pulse signal image's bandwidth can be calculated from the frequency relationship as follows

$$BW = \frac{1}{0.0978\,\mu s} = 10.225MHz \tag{4.2}$$

and the result matches with the frequency representation of the signal given in Figure 4.3 where the fundamental image is between 0 to 10 MHz approximately with its harmonics. Note that the image in the negative frequencies is mirrored version of the positive frequency component because of the constant 0 value of the imaginary component of the pulse signal. Moreover, its power is shown relative to 10 dB by default.

The QT GUI Frequency Sink block plays a crucial role in generating frequency plots. Figure 4.1 presents the primary parameters associated with this block: FFT Size, Center Frequency, and Bandwidth. While the parameter names themselves provide a basic understanding of their functionalities, it is essential for users to carefully consider the Bandwidth parameter, as it determines the unit of the x-axis in the frequency plot.

In the specific example mentioned, the Bandwidth parameter is set to match the sample rate of the Pluto device, which is 61.44 MHz. Consequently, the frequency axis unit

is defined as MHz. If the Bandwidth were to be reduced to 20 MHz, only the fundamental frequency image would be visible in the plot, while the harmonics would be excluded. It is important to note, however, that the plot itself would remain unchanged, except for the frequency axis. The modified frequency axis would range between -10 MHz and 10 MHz, leading to an erroneous indication of the signal's bandwidth.

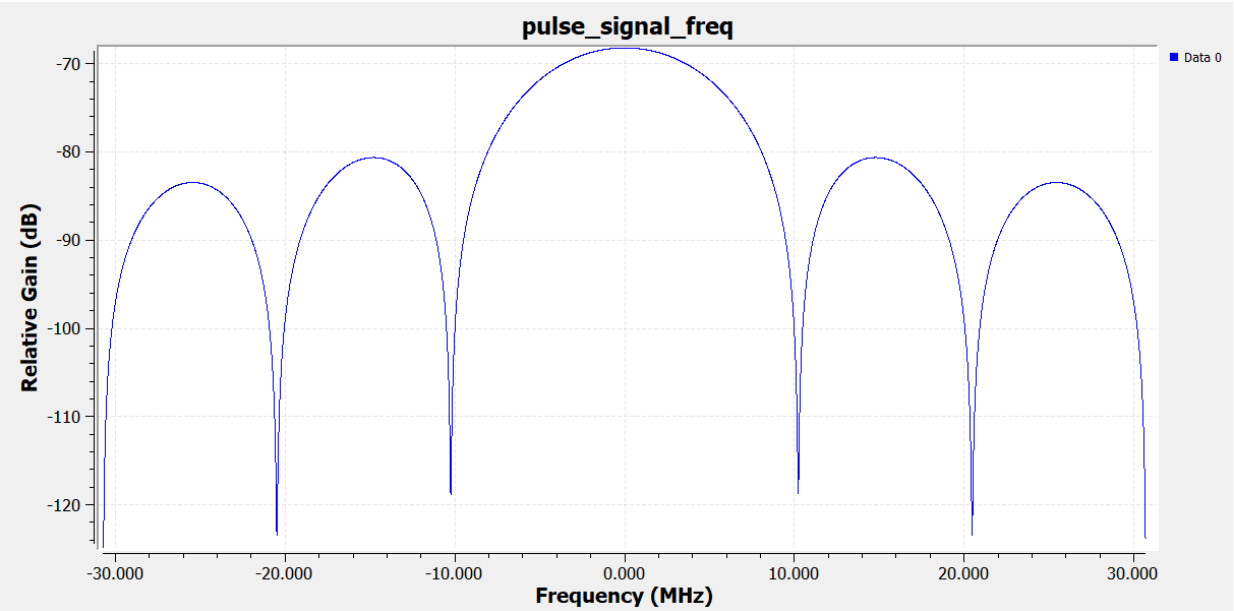



Figure 4.3. Frequency domain representation of the pulse.

To eliminate high-frequency harmonics, a "Low Pass Filter" block is introduced in the flowgraph, as depicted in Figure 4.1. Similar to other blocks, the filter block possesses specific parameters that must be set by the user according to desired specifications.

Low pass filter block offers decimation and interpolation capabilities, however, the frequency of the input signal is not changed during run time in this work, there is no need for both. The gain parameter is set as 0.845 to scale the output as close as the vector source's output level, which is 1, this value for the scaling factor is chosen to get the best Signal to LO ratio by running I and Q DACs at full-scale, Pluto SDR Sink and Source block alter the signal's power anyway. The sample rate is selected as 61.44 MSPS which allows the cut-off frequencies up to 30 MHz due to Nyquist criteria. The cut-off frequency is set as 10 MHz where the pass-band of the low pass filter ends. Finally, the last parameter to be explained is the transition width, it is set as 500 Hz, which can be defined as the

frequency difference between the cut-off frequency (pass-band endpoint) and starting point of the stop-band.

By using the filter design tool given parameters should be checked to determine whether GNU Radio can implement the filter with computational CPU capabilities or not. The designed low-pass filter with the explained parameters is shown in Figure 4.4. The low pass filter has a remarkably sharp transition due to its short transition width value. Frankly, the transition width is selected according to the resolution of the Fourier Transform or the frequency axis of the filter, which can be denoted as

$$Transition\ Width > \frac{F_S}{N}, \tag{4.3}$$

where $F_S$ is 61.44 MSPS and N is the number of FFT points, which is selected as 200000, can be seen in the bottom right corner of Figure 4.4. Thus, the sharpest value of transition width can not be less than 307.2 Hz for this case. This value indicates the frequency difference between each filter sample and provides frequency domain definitions for the samples at 10 MHz and 10.0005 MHz, which are boundaries of transition width.

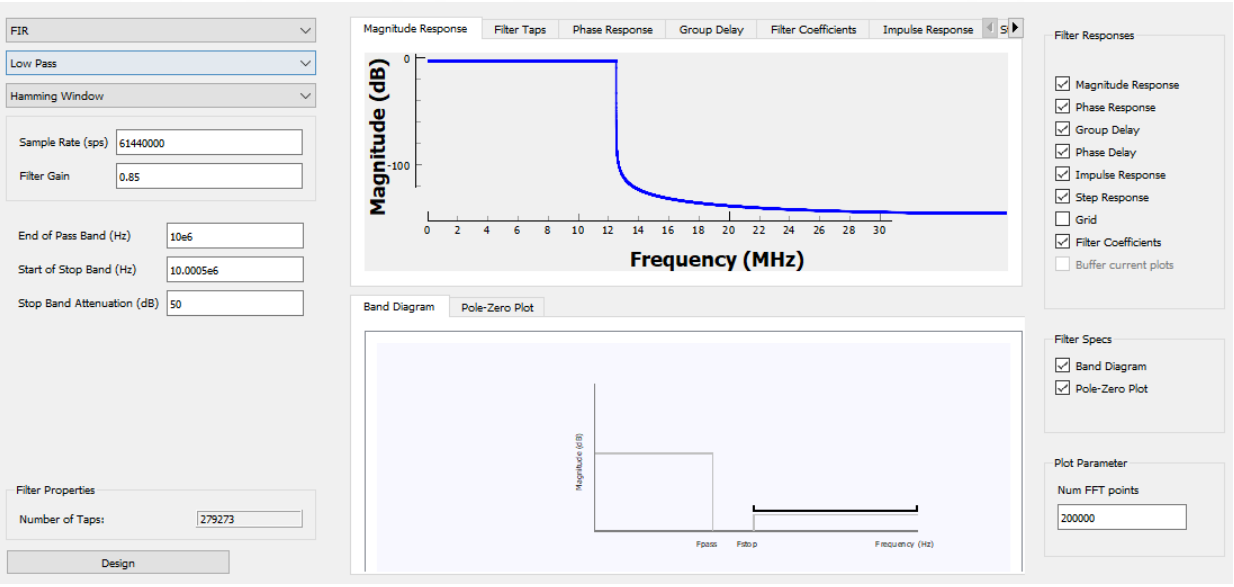



Figure 4.4. Designed low pass filter by using GNU Radio filter design tool.

Moreover, the filter design tool also calculates the number of taps, which is important for the time domain representation of the given transfer function in Figure 4.4 namely the finite impulse response filter. Since frequency domain multiplication equals time domain

convolution, the output of the FIR filter is acquired by the convolution of the input signal and impulse response of the filter, which is also drawn by this piece of software. In terms of the samples on the frequency and time axis, there is one to one relationship. Therefore, the number of FFT points (N) is used will be used in the following expression of FIR weighted sum that derived from the convolution

$$y[n] = \sum_{i=1}^{N-1} b_i \times x[n-i], \tag{4.4}$$

where weight b and input x form output y signal. From (4.3) N-1 value, which is used in (4.4) can be roughly estimated by dividing the sample rate by transition width, this value is called filter taps, the amount of computation required for each sample and it is calculated by the filter design tool. Filter taps can be also referred to as time domain response or the impulse response of the frequency domain.

The number of taps is calculated as 273279, this means that each output sample requires 273279 multiplication in the time domain approach as well as in the frequency domain FFT approach. Although the CPU is loaded with huge computation processes by selecting the transition width narrow, the used device can handle this load and produce a sharp filter.

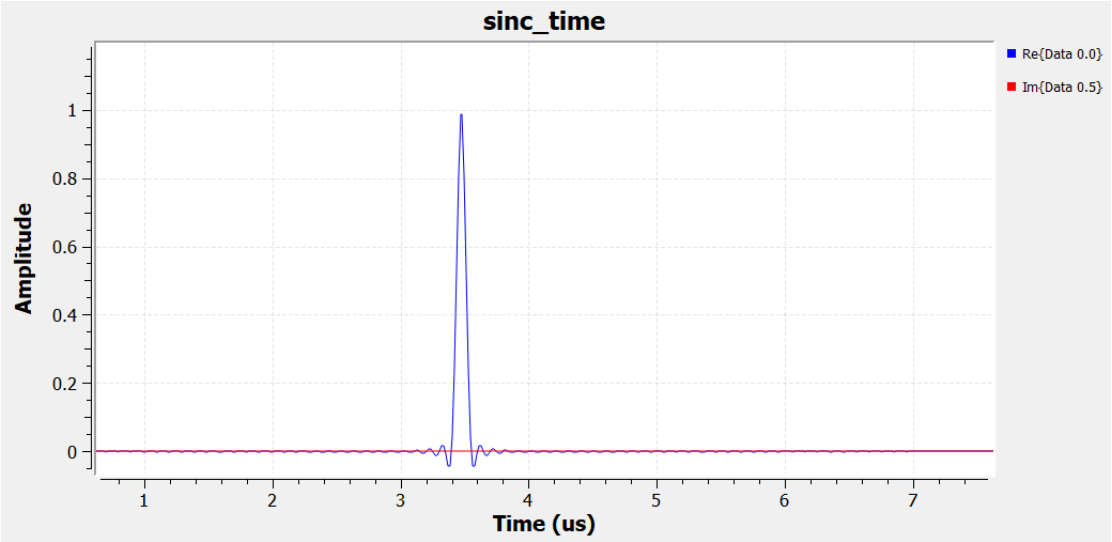



Figure 4.5. TX signal in the time domain.

The output signal from the low-pass filter is presented in both the time and frequency domains in Figures 4.5 and 4.6, respectively. By applying the low-pass filter and removing

high-frequency components, the signal is shaped into a sinc function in the time domain. This shaping results from the frequency response characteristics of the low-pass filter.

The sinc function, characterized by its oscillatory behavior and a central lobe resembling the shape of the sinc(x) function, is often encountered when dealing with low-pass filtering operations. The sinc function exhibits zero crossings at multiples of the sampling period, leading to a periodic pattern.

The shaped signal, represented as a sinc function, is intended for transmission purposes. It is important to consider the spectral properties and the bandwidth of the signal, as they determine the information-carrying capacity and the fidelity of the transmitted signal.

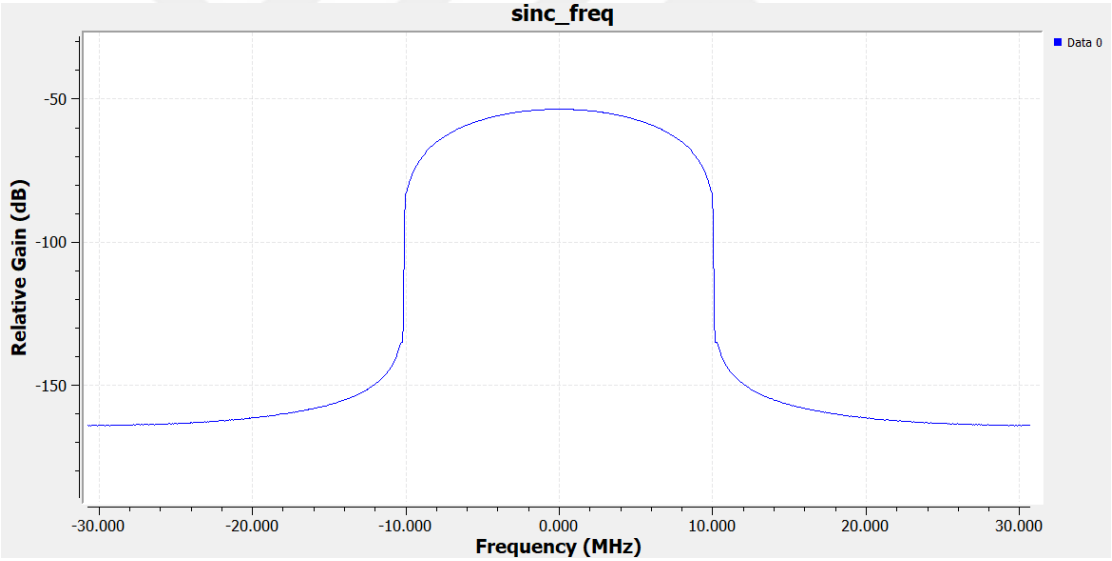



Figure 4.6. TX signal in the frequency domain.

## 4.2. Transmit and Receive the Extracted Signal with Pluto SDR

Once the TX signal has been generated, the next step in the flowgraph is to incorporate the Pluto SDR transmitter and receiver blocks. In the loopback configuration, where the RX port is connected to the TX port using an SMA cable, the Pluto SDR Sink and Pluto SDR Source blocks are utilized. These blocks facilitate the transmission and reception of signals using the PlutoSDR device.

In Figure 4.7, the PlutoSDR Sink block represents the transmitter (TX), responsible for sending the generated signal to the PlutoSDR device. On the other hand, the PlutoSDR Source block symbolizes the receiver (RX), which captures the received signal from the PlutoSDR device.

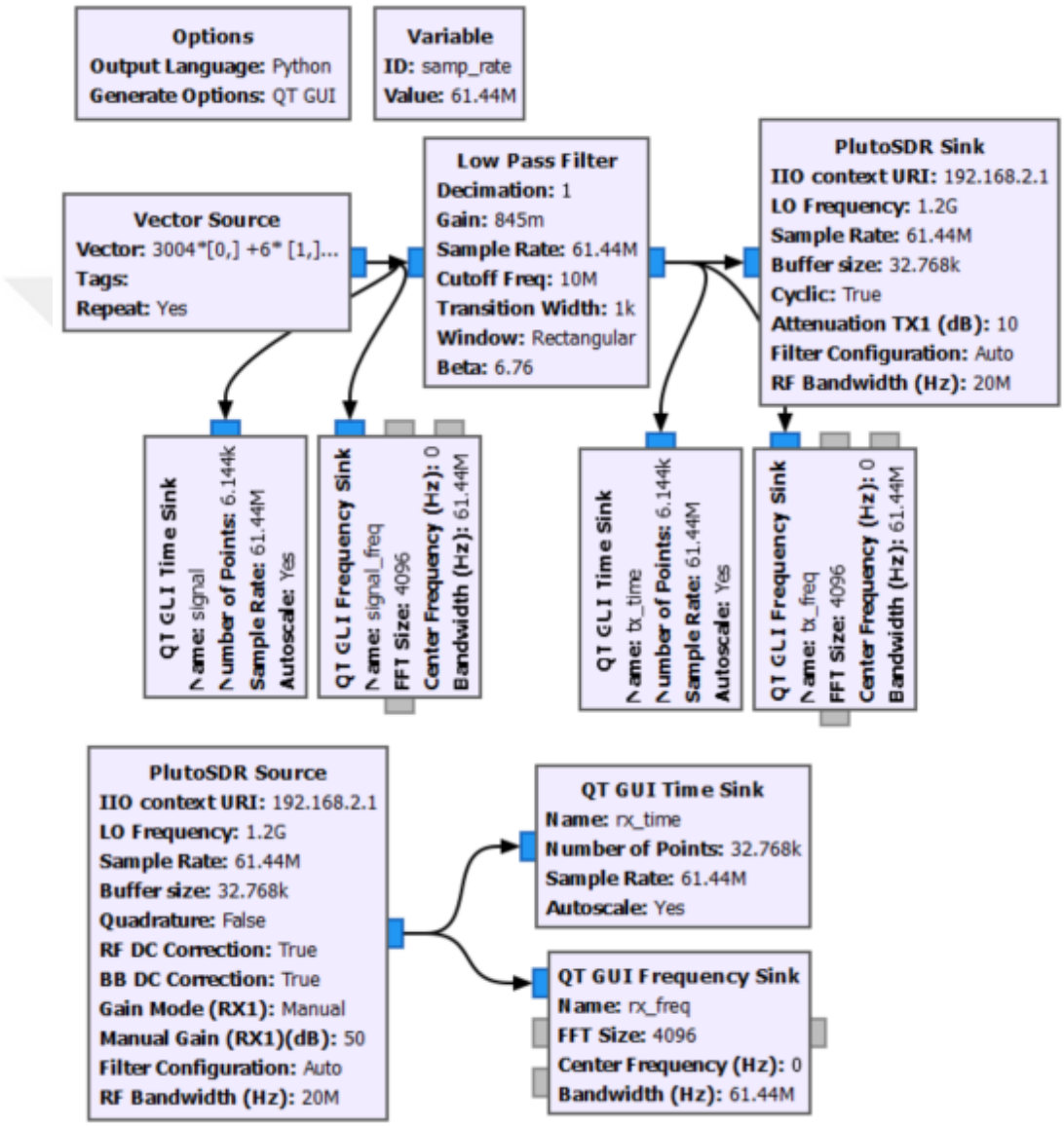



Figure 4.7. Initial flowgraph with PlutoSDR sink and source blocks.

By incorporating these blocks into the flowgraph, the signal generated by the TX block can be transmitted through the PlutoSDR Sink and received by the PlutoSDR Source. This loopback configuration allows for measurements and testing within the system, as the signal is transmitted and received internally through the connected RX and TX ports of the PlutoSDR device.

The following section contains the explanation of each parameter set in these blocks from [55, 56];

- IIO context URI: Set as 192.168.2.1 is the reserved IPv4 address for the Pluto devices.
- LO Frequency: Selects RX and TX local oscillator frequencies that will be used in mixing operations for up-down conversions of RF signals. There are two identical RFPLL synthesizers inside the AD9363 to generate the demanded LO signals [46].
- Sample Rate: Frequency at which the hardware will input or output samples. On the RX side, it is equal to data rates after decimation filters whereas it is equal to input data rates at the interpolation filter on the TX path. It defines how much of the bandwidth is received at once.
- Buffer Size: Size of the internal buffer in samples, one buffer of samples is processed at a time by the IIO blocks. When a stream is started four buffers are used for capturing in hardware. Then each refill request, a single buffer is transferred to them, which makes room for the hardware to save more data.
- Tracking Options (RX): Quadrature, RF DC Correction, BB DC Correction.
- Gain Mode (RX): Slow attack, fast attack, manual, hybrid.
- Filter Configuration: Auto, loads a default filter, which enables easier processing with lower sample rates
- RF Bandwidth: Configures TX analog filters, TX BB LPF, and TX Secondary LPF, which are set by libiio automatically according to the assigned value on the software.
- Attenuation (TX): Controls the attenuator at the end of the TX path, maximum output in terms of is generated at 0 dB attenuation.

### 4.2.1. Tracking Options

In order to address the specific issues that arise in the implemented application on the SDR, appropriate tracking options need to be configured. Due to the architecture of the SDR, which is direct conversion or zero intermediate frequency, there are certain common issues that need to be accounted for, including I/Q mismatch and DC offsets [57]. To mitigate these issues, three tracking option algorithms have been implemented in the FPGA. These algorithms are designed to monitor and compensate for variations in I/Q mismatch and DC offsets, ensuring the accuracy and quality of the received signal.

4.2.1.1. RF DC Correction. In the RX path, as described in section 3.2, certain ports require regular DC bias values to achieve optimal performance. For example, the LNA input pins should have a DC bias around 0.6 $V_{DC}$ [46]. This DC biasing ensures proper operating conditions for the LNA and helps achieve desired performance levels.

RF DC correction is employed to compensate for any DC offsets in the RF path that may be present internally within the chip. This correction is achieved by calibrating the gain indexes in the RX gain table. By adjusting the gain indexes, the RF DC correction process aims to minimize or eliminate any DC offsets that could degrade the performance of the RX path.

4.2.1.2. BB DC Correction. DC offsets are one of the most influential drawbacks in direct conversion architecture, this malfunction manifests itself as a considerable spike in the received spectrum center as shown in Figure 4.8.

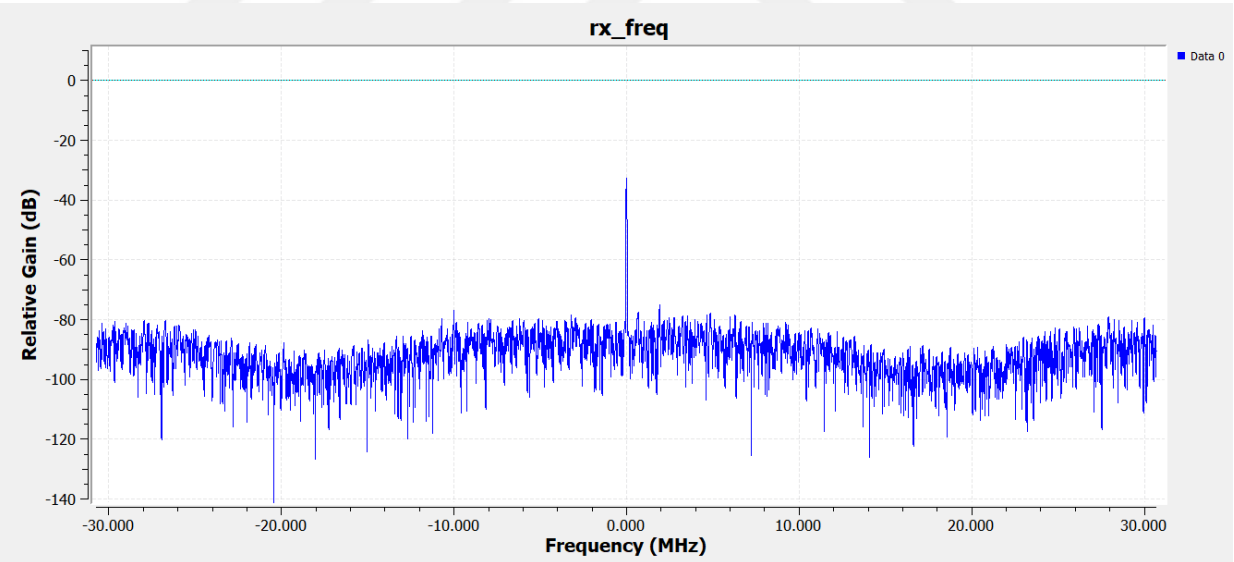



Figure 4.8. LO leakage impact to received spectrum.

It is important to note that the plot in Figure 4.8 is a result of a simulation where the RX and TX ports are shielded with insulating covers. In this scenario, no signal is actually transmitted or received by the device. Therefore, it can be concluded that the spike observed in Figure 4.8 is caused by LO leakage. Local oscillator leakage refers to the leakage of the Local Oscillator signal into the received signal path. In direct conversion

architecture, where the LO frequency is closely aligned with the desired signal frequency, LO leakage may have serious consequences.

The LO leakage given in Figure 4.8 referred to as "self-mixing" is a phenomenon that arises due to finite isolation between the LO and RF ports of the mixer in silicon-based Integrated Circuits (ICs) [58]. The level of isolation is limited by various factors, including low-resistivity substrate coupling, bond wire radiation, and capacitive and magnetic coupling [57].

Figure 4.9 illustrates potential scenarios where the LO signal can leak through. The most destructive scenario is when the LO signal leaks to the input of the LNA, which is a low-impedance node, through the substrate. In this case, the LO signal gets amplified before mixing with itself, leading to a significant spike in the received spectrum.

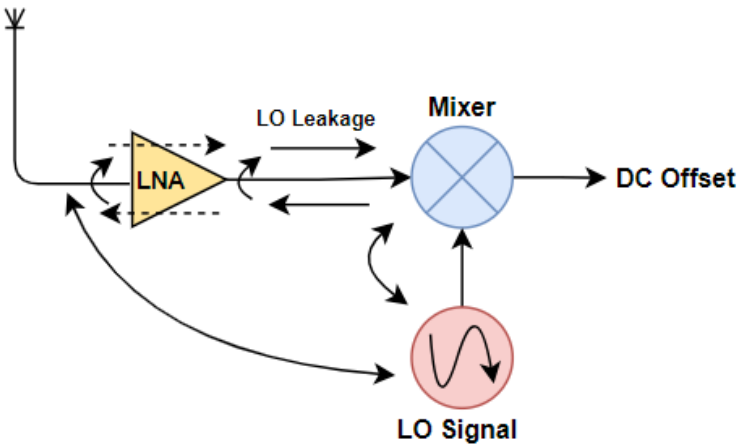



Figure 4.9. Possible scenarios for self mixing issue

Furthermore, if the reverse gain of the LNA is not sufficient, a leakage signal at the output of the LNA may propagate back to the input. This can result in the same detrimental effect as the first scenario, where the amplified leakage signal mixes with itself. Alternatively, the leakage signal may be reflected off the mixer input directly, causing a less pronounced spike at the center of the baseband.

These various scenarios of LO leakage and self-mixing emphasize the importance of addressing isolation issues and optimizing the performance of the RF front-end components

in direct conversion architectures. Proper design techniques, such as substrate isolation, shielding, and careful selection of component characteristics, are crucial in mitigating LO leakage and minimizing its impact on system performance.

By setting the BB DC Correction tracking option to true before the simulation changes the output plot as given in Figure 4.10, the spike caused by the leakage is now removed by the DC offset correction values that are applied to the RX gain table index [46].

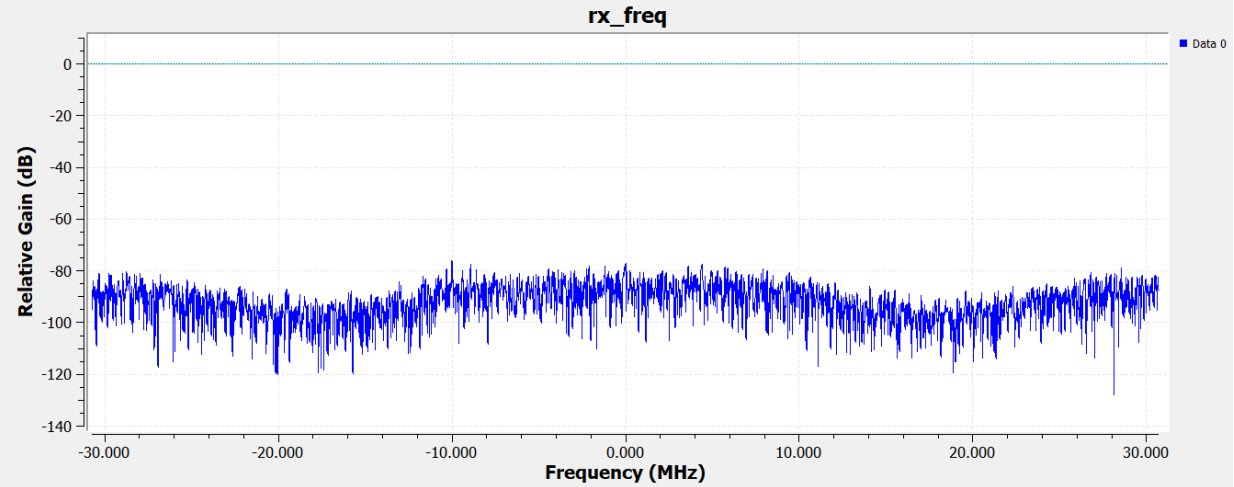



Figure 4.10. Spike free spectrum by BB DC correction.

4.2.1.3. Quadrature. In the most modern zero intermediate frequency software defined radios, it is common to encounter in-phase versus quadrature imbalance, which refers to the deviations in amplitude and phase between the in-phase and quadrature components of the signal. This imperfection can have a detrimental impact on the Bit Error Rate (BER) performance, particularly for systems that employ digital modulation schemes [59]. The main cause of the I/Q imbalance is the deviation from the ideal 90-degree phase relationship between the I and Q components. This ideal phase relationship ensures excellent orthogonality between the I and Q components, which is crucial for accurate signal demodulation and decoding. To mitigate I/Q imbalance, modern SDR devices often incorporate quadrature calibrations. These calibrations aim to reduce phase and gain mismatches in the RX and TX I/Q channels, effectively improving the balance between the in-phase and quadrature components. By enabling quadrature calibrations, the device can minimize the impact of I/Q imbalance on system performance.

However, it is important to note that quadrature calibrations can pose challenges when dealing with non-quadrature signals. Non-quadrature signals are generated by setting either the I or Q component to zero, resulting in an inherent amplitude difference between them. In such cases, the calibration process may introduce amplitude perturbations, as shown in Figure 4.11 where in-phase signal amplitude is varying without any input. These perturbations can affect the accuracy of the signal representation and potentially degrade system performance.

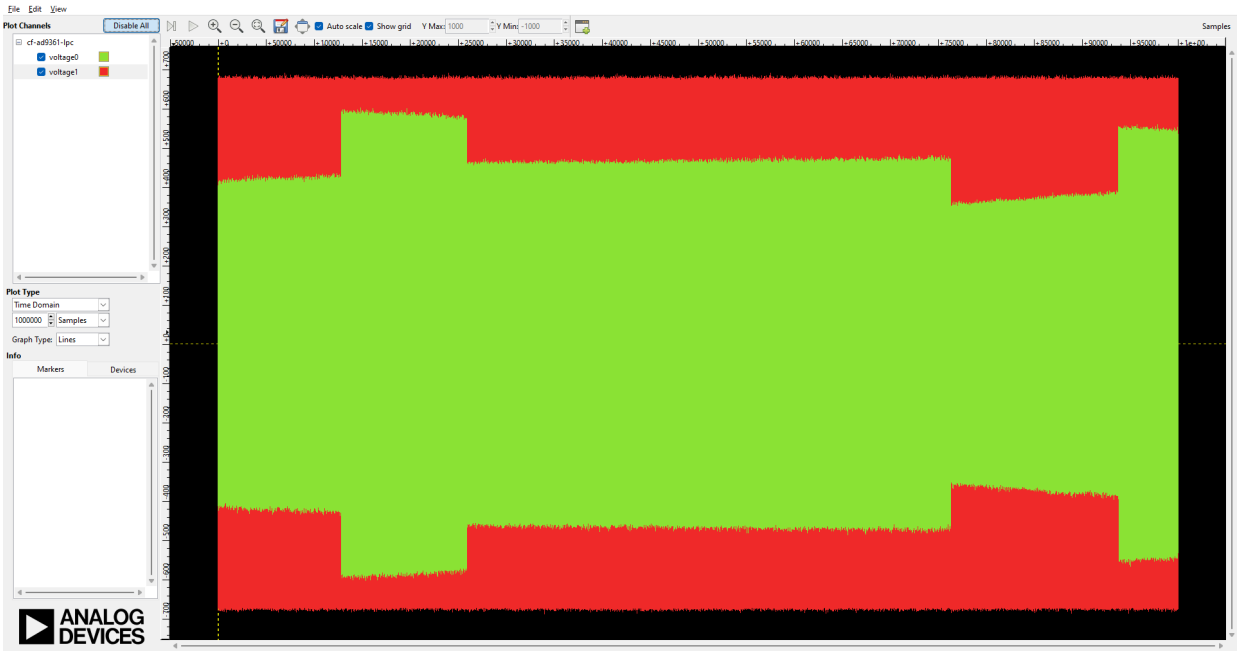



Figure 4.11. In-phase (green) and quadrature (red) when quadrature tracking is on.

Quadrature tracking does not possess knowledge about the specific signal type being transmitted or received. It may perceive any difference between the I and Q components as a fault and attempt to minimize it, leading to amplitude errors in the signal [60]. Disabling the quadrature tracking option can help alleviate these perturbations, as shown in Figure 4.12.

Furthermore, when the device is operated in an analog loopback configuration, where the RX port is connected to the TX port on the same device via an SMA cable or antennas, it is recommended to disable quadrature tracking. In this setup, random amplitude differences between the received I and Q components, referred to as phase randomization, may exist. Enabling quadrature tracking under such circumstances may mistakenly interpret these

amplitude differences as an imbalance or fault, introducing additional perturbations and compromising system performance.

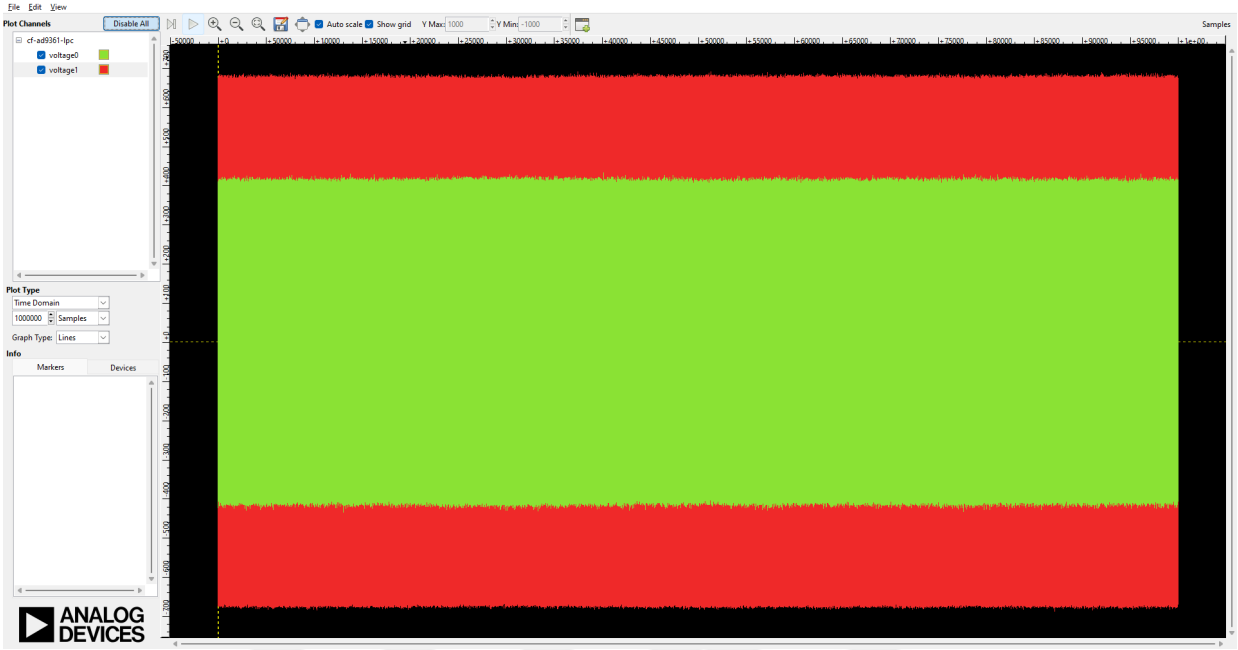



Figure 4.12. In-phase (green) and quadrature (red) when quadrature tracking is off.

### 4.2.2. Phase Randomization

Phase randomization in the analog loopback configuration arises from the inherent differences between the phase of the RX and TX PLLs (Phase Locked Loop). These differences lead to random variations in the amplitude between the I and Q components [60]. Notably, any adjustments made to the RX or TX PLL settings, such as modifying the LO frequency or sample rate, can alter this amplitude difference.

Additionally, when running the same flowgraph multiple times on GNU Radio, each simulation execution will introduce further randomization in the TX and RX PLLs. This is because Pluto blocks update every parameter, even if they have not been changed, during block initialization in the simulation. As a result, the phase randomization and corresponding amplitude differences between the I and Q components will vary across different runs of the flowgraph.

The details about the processes where LO is being used such as mixing and complex modulation are discussed in previous sections with the related mathematical expressions (2.1, 3.3). Nevertheless, in [61] each expression of modulation steps is shown, due to the simplification with trigonometric and Euler's identities the final equation of modulated signal can be written as

$$x(t) = \frac{(A+B)}{2} \times sin(\kappa + \omega) + \frac{(A-B)}{2} \times sin(\kappa - \omega), \tag{4.5}$$

where $\omega$ is the angular frequency of the signals that fed through I and Q channels, $\sin(\omega)$ and $\cos(\omega)$ respectively, and the $\kappa$ is equal to $(2\pi f_{LO} + \phi_{TX})$. When it comes to demodulating this signal, local oscillators of the RX path must be set as $\cos(\kappa)$ and $\sin(2\pi f_{LO} + \phi_{RX})$. Since the phase of RX and TX PLLs are not the same, there will be an amplitude mismatch between transmitted and received signals I and Q components. Figure 4.13 shows the transmitted and received signal pair by the Pluto and the effect of PLL randomization expresses itself as the direction and amplitude of received signals.

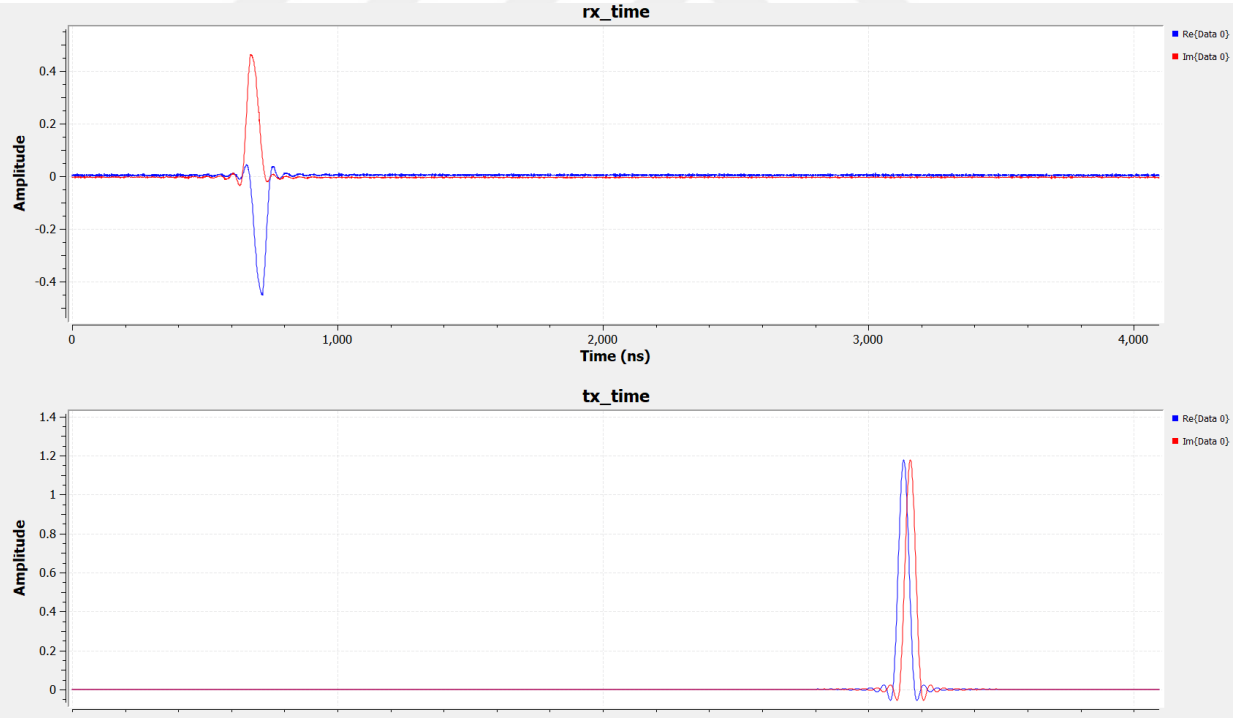



Figure 4.13. Effects of phase randomization on received signals, received I and Q pair (top), transmitted I and Q pair (bottom).

In order to point out the cause of the issue given in Figure 4.13, the generation of LOs is discussed elaborately in the following paragraphs. Although there are various applications

where PLL is used, in Adalm-Pluto SDR it is used to generate a stable LO frequency in accordance with the provided reference input frequency.

PLL is kind of a control circuit that uses several electronic components such as dividers, phase detectors, filters, and VCO (Voltage Controlled Oscillator) during the generation of output frequency [62]. AD9363 transceiver employs three fractional-n phase locked loops to generate TX - RX LO frequencies and the oscillator called baseband PLL that is used by data converters, digital filters, and I/O ports [46]. The used PLL topology for RX and TX in Pluto is given in Figure 4.14.

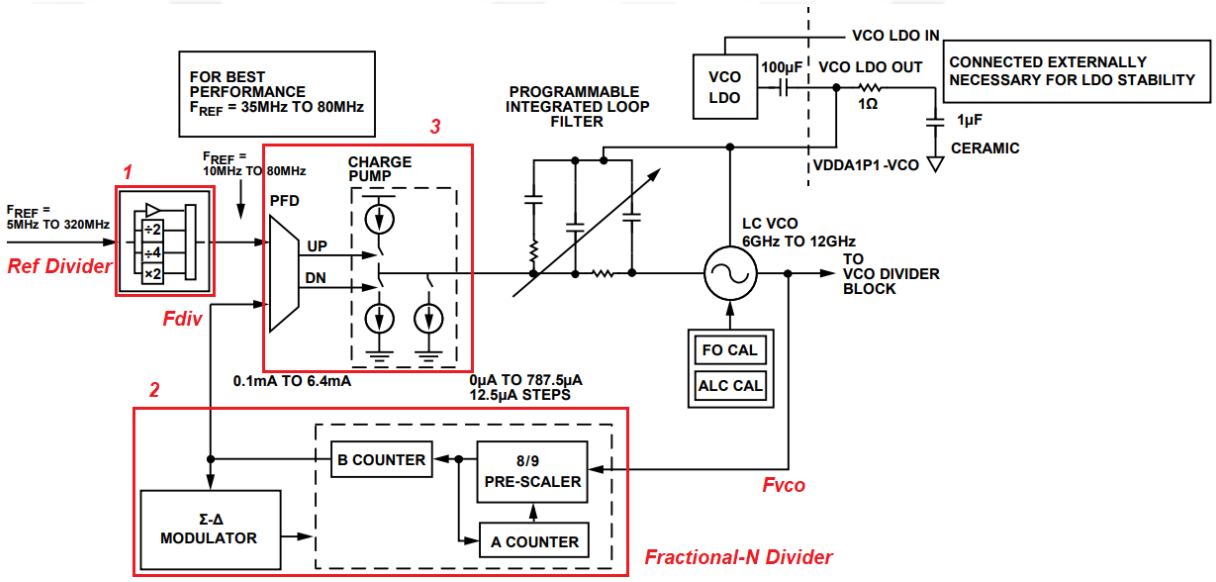



Figure 4.14. Pluto SDR TX and RX PLL synthesizer block diagram, (1) reference divider, (2) fractional-n divider, (3) frequency-phase detector and charge pump [46].

The VCO in the Pluto SDR utilizes an LC resonator to generate frequencies ranging from 6 GHz up to 12 GHz, with the output frequency being proportional to the input tuning voltage. However, the frequency produced by the VCO can experience drift due to variations in temperature, supply voltage, and other process-related factors. To address this, a feedback control loop is employed to lock the VCO frequency to the reference input frequency, ensuring stability and accuracy [62].

Pluto SDR uses an on-chip digitally controlled crystal oscillator (DCXO) and it delivers 20 MHz to 50 MHz frequency range to the ref divider (labeled as 1) as $F_{REF}$. The

first input of PFD (Phase Frequency Detector) fed by the "ref divider" output, reference frequency, $F_{REF}$, is either buffered, doubled, halved, or divided by 4 by the divider to scale DCXO frequency as close to 80 MHz since the manufacturer recommends 80 MHz reference clock to PFD for best RF PLL performance [46]. By default, DCXO produces 40 MHz and the ref divider doubles that frequency to the PFD input.

The other input of PFD ($F_{DIV}$) is connected to the fractional divider (labeled as 2) located on the loopback path consisting of a pre-scaler, counters, sigma-delta modulator and is called a fractional divider, which can generate fractional frequency values such as 600.2 MHz due to sigma-delta modulator. Since the PFD locks the frequency of given two inputs, the amount of division (N) can be found as

$$N = \frac{F_{VCO}}{F_{REF}} = \frac{F_{VCO}}{F_{DIV}}, \tag{4.6}$$

where $F_{VCO}$ can be any value between VCO output range 6 GHz to 12 GHz and $F_{REF}$ is the reference clock frequency after "ref divider" block. Usually, pre-scalers are used to deal with very high VCO frequencies just like the given VCO output range. The dashed rectangular in fractional-n divider represents pulse-swallow divider circuitry, the pre-scalar first divides input frequency by 9 (P+1) until A counter reaches 0 or it is full (depending on counter configuration) then the pre-scaler resumes to division by a factor of 8 (P) until an interrupt from B counter whether there is any item to count or not, note that both counters start counting simultaneously. Thus, the number of divisions can be calculated as follows from [63]

$$N = (P + 1) \times A + P \times (B - A) = (P \times B) + A, \tag{4.7}$$

which can be thought of more detailed version of (4.6).

PFD and charge pump are integrated together, labeled as 3 in Figure 4.14, PFD compares its inputs in terms of the phase and produces a correction voltage, which is converted to a correction current by the charge pump. If the compared frequency and the phases are the same, PLL is considered to be locked. The integrated loop filter collects the correction currents from the charge pump into a voltage that controls the VCO. Required hardware configurations for a given frequency such as divider values and filter configurations are handled automatically referring to the supplied lookup table by the vendor [46].

The VCO output signal from the previous stage is fed into the VCO divider block in the Adalm-Pluto SDR, as shown in Figure 4.15. This block serves the purpose of bringing the signal frequency within the desired range of 70 MHz to 6 GHz. The VCO itself has a much wider frequency range, and the VCO divider allows for the down-conversion of the signal to the desired frequency range.

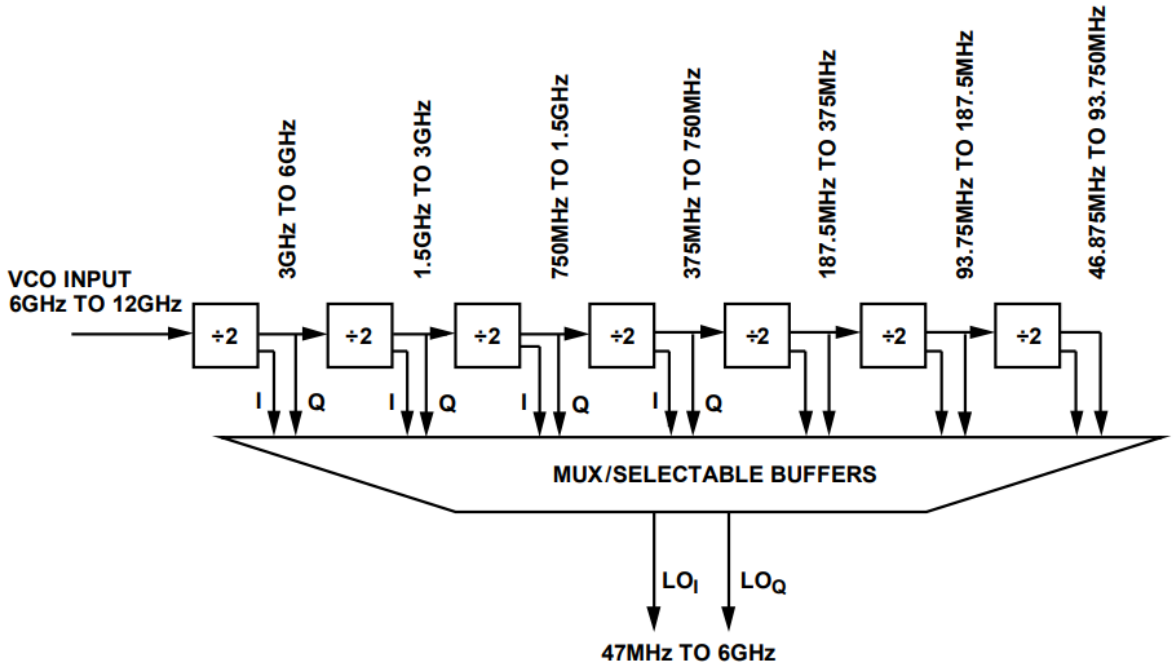



Figure 4.15. VCO divider [46].

However, this down-conversion step introduces a random phase difference between the LOs of the transmitter and receiver. The AD9363 transceiver used in the Adalm-Pluto SDR does not possess an RF PLL phase synchronization feature. In the divide-by-2 array within the VCO divider block, each division can occur on either the rising or falling edge of the clock. As a result, this introduces a phase ambiguity that is dependent on the division ratio [63]. In other words, the phase relationship between the transmitter and receiver LOs can vary due to this inherent characteristic of the divide-by-2 operation.

Assuming that 4 GHz LO frequencies are selected for the system, the VCO divider will divide VCO output by 2 according to Figure 4.15, the divided LO frequencies can only have a phase ambiguity of 0 or 180 as given in Figure 4.16. During the operation, the division can result in either a ”2” signal or a ”3” signal with respect to the input clock,

"1". Thus, it is said that there can be a phase ambiguity of either 0 degrees or 180 degrees between RX and TX LOs. Phase ambiguity can be interpreted by

$$\frac{360}{2^n}, \tag{4.8}$$

where n is the used divide by 2 modules, for example, dividing VCO output by 4 introduces a multiple of 90 degrees phase difference.

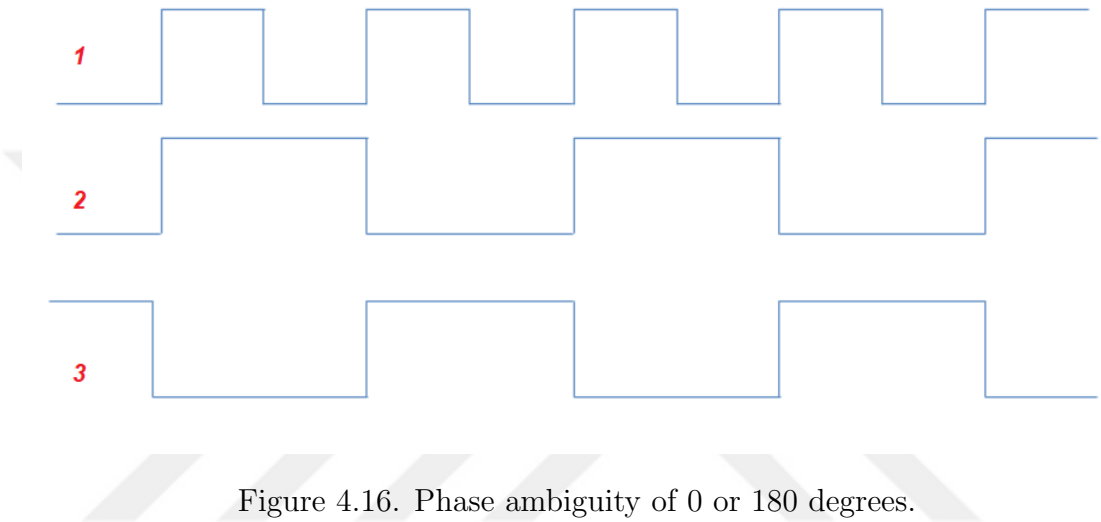



Figure 4.16. Phase ambiguity of 0 or 180 degrees.

## 4.3. Final Flowgraph

The use of Generic IIO (Industrial I/O) Device Source/Sink blocks in conjunction with Pluto SDR blocks provides a solution to address the random phase and amplitude issue. In this approach, the flowgraph is initially run with the Pluto SDR Sink and Source blocks to configure the device parameters such as LO frequency, sample rate, and gain. Once the device parameters are set, the Pluto SDR blocks are disabled, indicated by their greyed-out appearance in Figure 4.17.

After disabling the Pluto SDR blocks, the Generic IIO Device Source/Sink blocks are enabled. These blocks are used to directly interface with the underlying IIO subsystem, providing greater flexibility and control over the signal processing chain. By using the Generic blocks, the phase relation between the I and Q pairs is maintained, ensuring consistency in the signal processing.

It is important to note that although the phase relation remains constant throughout the flowgraph execution, it is still randomized due to the inherent phase ambiguity discussed earlier. However, by following this procedure and utilizing the Generic IIO Device blocks, the phase relation is preserved across multiple flowgraph runs, providing a more stable and predictable behavior compared to running the flowgraph with the Pluto SDR blocks alone.

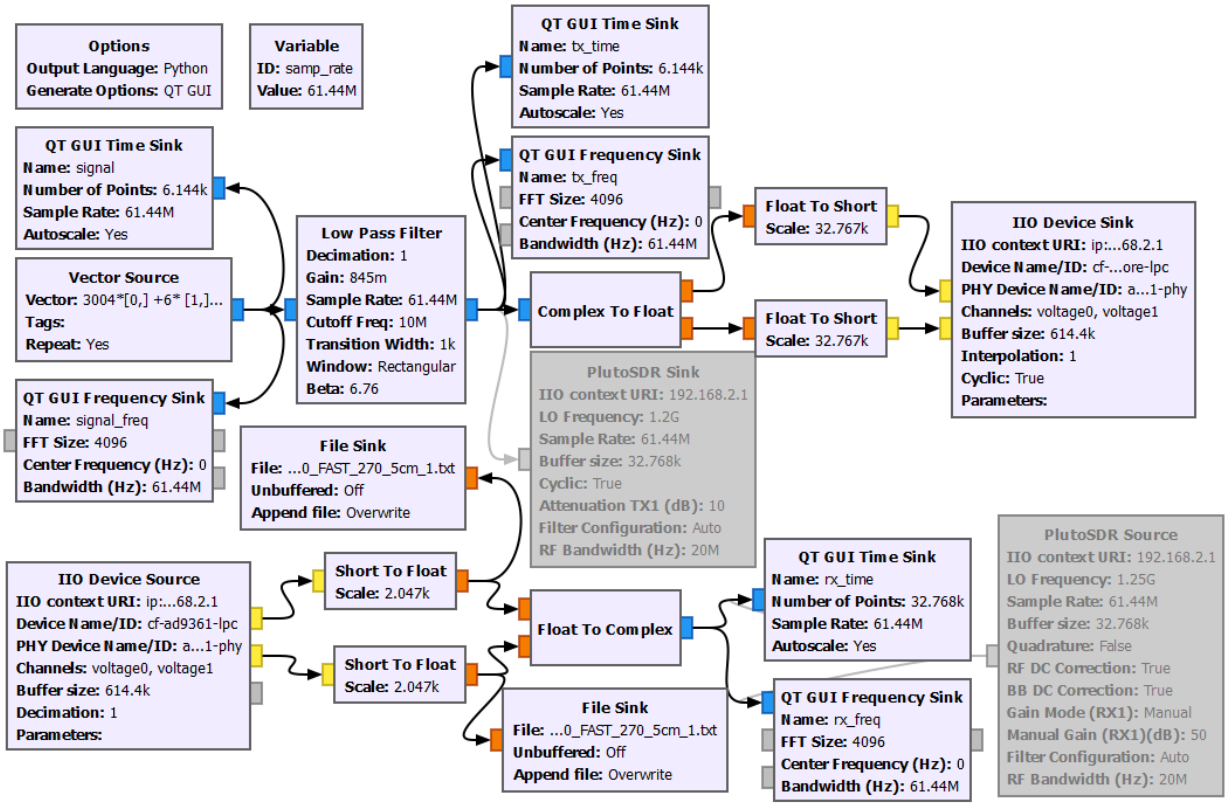



Figure 4.17. Flowgraph on GNU Radio with IIO blocks.

In addition to the Pluto SDR and Generic IIO Device blocks, the flowgraph shown in Figure 4.17 includes File Sink blocks, type converter blocks, and Complex to Float/Float to Complex blocks. The File Sink blocks are used to save the received signal as I and Q components to files on the host PC. These files can be further processed and analyzed using MATLAB or other software tools.

Since the Generic IIO Device blocks operate with short format data, type converter blocks are used to convert the data between different buffer types. In this case, Complex to Float and Float to Complex blocks are used. The Complex to Float block converts the complex data format (containing real and imaginary components) to a float format, where

each component is represented by a pair of 32-bit floating-point values. Similarly, the Float to Complex block performs the reverse operation, converting the float format back to a complex format. These conversions do not require a scale parameter since both formats are already represented using 32-bit floats. Conversions between Complex to Float and Float to Complex blocks do not require a scale parameter since both of them are 32 bits, float terminals have 2 ports for real and imaginary components of a complex signal where uses a pair of 32-bit floats are used to define the real and imaginary parts of a complex sample [65]. On the other hand, the float to short and short to float blocks in the flowgraph require a scaling parameter to ensure that the data does not overflow or underflow, as the 16-bit signed integer range is narrower than the 32-bit float range. In GNU Radio, the expected range for inputs and outputs is normalized to +1 and -1, and any values outside this range will be clipped to the range boundaries [66].

To achieve full-scale representation on the TX side for short data, the sinc signal, which has an amplitude of less than 1 unit, is multiplied by $(2^{15} - 1) = 32767$. This scaling factor ensures that the signal reaches the maximum amplitude within the range of the 16-bit signed integer format. For the IIO Device Source block, the short to float converter block is used to scale the DAC output into the GNU Radio normalized range of -1 to 1. Since the ADC and DAC in AD9363 use 12-bit samples, the least significant 4 bits are ignored in the 16-bit data representation. To normalize the 12-bit signed data, the I and Q outputs of the IIO Device Source block are divided by $(2^{11} - 1) = 2047$. These scaling operations are handled automatically when Pluto Sink and Source blocks are used.

The impact of PLL on received data is illustrated in Table 4.1. The measurements conducted at 1.4 GHz using an analog loopback configuration reveal a potential variation of up to 15% between samples taken under the same measurement setup and variables. Although attempts were made to mitigate this effect using IIO blocks, it was determined that they were not suitable for the intended use of the system.

By using IIO blocks, frequency translations are handled manually. Using this technique in the flowgraph makes it impractical to utilize the SDR device as a VNA with automated frequency sweeping capabilities. This means that the device cannot sweep through a range of frequencies automatically, limiting its functionality as a VNA.

Table 4.1. Magnitudes of the received signal for all possible I and Q cases.

| | **I** | **Q** | **Mag** |
|---|---|---|---|
| **1** | + | + | 0.864 |
| **2** | + | - | 0.818 |
| **3** | + | 0 | 0.905 |
| **4** | - | - | 0.815 |
| **5** | - | + | 0.815 |
| **6** | - | 0 | 0.903 |
| **7** | 0 | + | 0.959 |
| **8** | 0 | - | 0.958 |

Furthermore, the manual process of measuring predetermined frequency values consumes a significant amount of time. Each frequency value needs to be measured individually, resulting in inefficiency and increased workload. To address these challenges, a software module called "Spectrum Analyzer" given in Figure 4.18 has been developed using the Python programming language. GNU Radio provides the flexibility to create custom blocks, allowing users to extend the capabilities of the existing modules for specific applications that are not supported out of the box [67]. In the custom block, the code initially identifies the peak values of the received I and Q signals. These peak values are then compared with the desired I and Q cases. If a match is found, the next frequency value is loaded onto the Pluto Software Defined Radio hardware, and a significant amount of received data is saved for the current frequency. It is important to note that non-zero terms may have varying power levels. In the code, the in-phase signal is assumed to be positive if the data converter code is above zero. However, these different power levels can introduce more complexities, as selecting both signals as positive creates more combinations than selecting one of them as zero.

In cases where the desired combination is not found, the custom block adjusts the frequency by adding 500k to update the PLLs, resulting in new combinations of I and Q signals. Subsequently, 500k is subtracted to return to the actual frequency of interest. This search process is repeated 300 times to attempt to find the desired combination. In the event that the desired combination remains elusive after a series of 300 attempts, the

implemented algorithm engages in a dynamic recalibration of the threshold values. Then, it proceeds with the pursuit of the next set of frequency values in a manner consistent with previous iterations. This adaptive mechanism is employed to imbue the search process with enhanced flexibility, with the overarching objective of augmenting the likelihood of successfully discerning the sought-after combination. In addition to the previously used blocks, there are several new blocks introduced in the updated flowgraph given in Figure 4.18. These blocks serve specific purposes and enhance the functionality of the flowgraph.

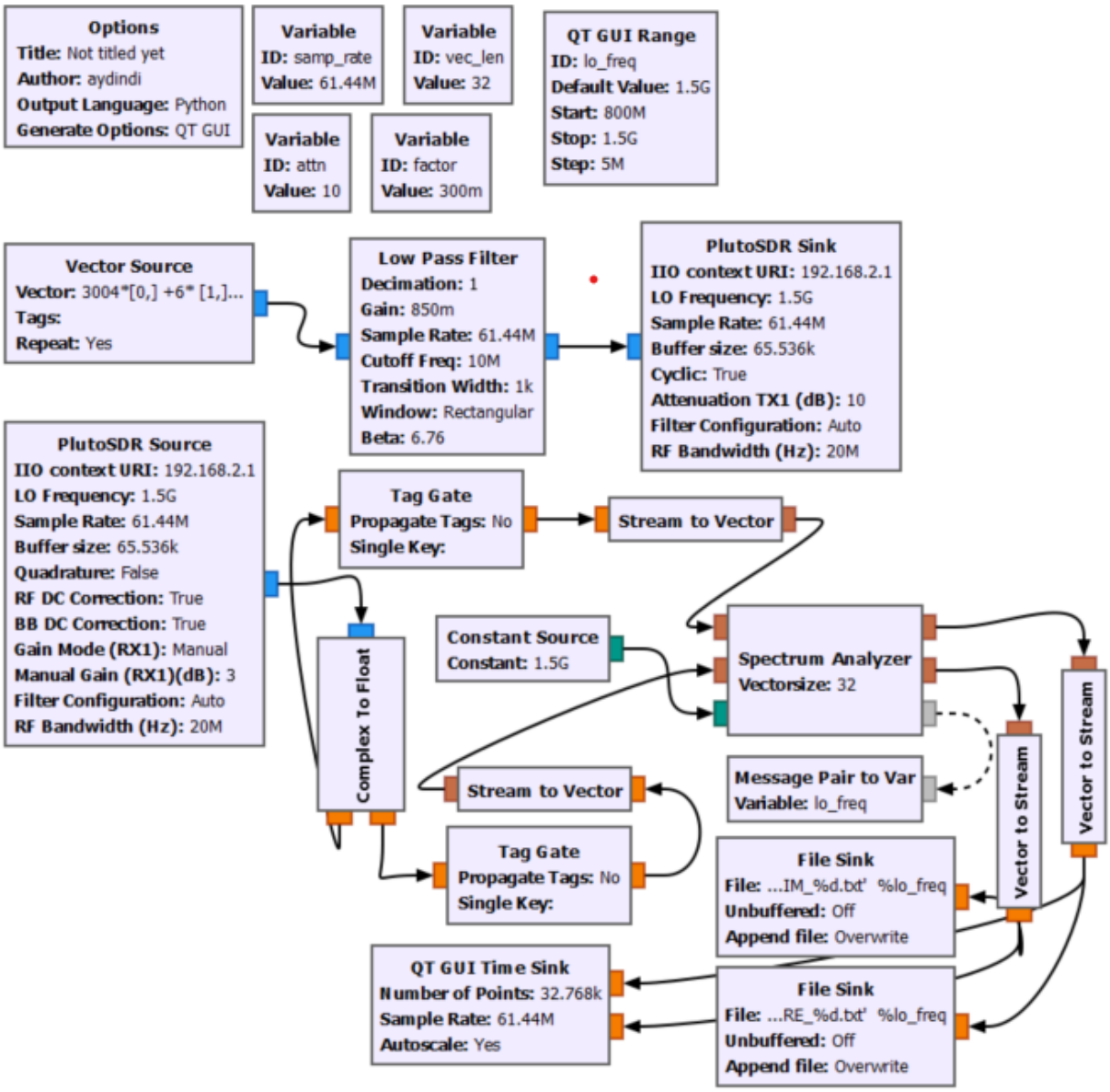



Figure 4.18. Final flowgraph with proposed custom Python block.

The following bullet list is specifically designed to offer a meticulous and in-depth elucidation of every individual block that has been recently introduced and seamlessly integrated into the overarching flowgraph. This comprehensive explanation is aimed at providing a thorough understanding of the purpose, functionality, and significance of each block within the larger context of the system:

- QT GUI Range: This block provides a graphical user interface (GUI) range slider that allows users to adjust a specific parameter within a given range. It is often used to control and set values dynamically during runtime. In this specific case, the LO frequency is controlled by the GUI slider, which allows the user to adjust the frequency within a specified range.

  The user has the option to select the operation type, which can be fully automated or manual. In the fully automated mode, the flowgraph sweeps the frequency automatically within the specified range, without any user intervention. This mode is suitable for applications where the frequency needs to be swept continuously or in a predetermined sequence without manual adjustments, which is mainly used in automated applications.

  Alternatively, the user can choose to manually change the frequency within the given range using the GUI slider. In this mode, the user can increment or decrement the frequency with a specific step size, allowing for more fine-grained control over the frequency selection. This mode is useful when the user wants to manually explore different frequency settings or target specific frequencies of interest.
- Constant Source: This block generates a constant value that can be used as a source or reference within the flowgraph. In this flowgraph, it provides a frequency value to Spectrum Analyzer Block. Despite its name, the frequency value provided by the Constant Source block can be changed dynamically through two methods: via the QT GUI interface or by broadcasting the frequency value from the custom block.
- Message Pair to Var: This block converts a pair of messages into a single variable or value. Messages in GNU Radio are used to carry metadata or control information along with the data stream. The Message Pair to Var block extracts relevant information from the messages and outputs it as a variable. In the proposed system, the frequency values broadcasted by the custom block have the capability to alter the LO frequency values of the Adalm-Pluto device thanks to this block.

- Stream to Vector and Vector to Stream Converters: These blocks are used to convert data between stream and vector formats. Stream format represents a continuous stream of samples, while vector format represents discrete blocks of samples. Since the created custom block cannot handle a continuous stream of samples, this data is divided into smaller vectors for proper operation. After the processing outputted data is returned into a stream to print the received data continuously.
- Tag Gate: The Tag Gate block acts as a gate or filter for data based on tags attached to the data stream. In this example, it blocks data based on the specified tags, and ensures better visualization of the received signal.

### 4.4. Measurements and Setup

The utilization of a specially designed data capture module is intended for tracking biodegradable sensors made from magnesium. To facilitate monitoring various phases of degradation, a Split Ring Resonator (SRR) structure is preferred, as described in reference [18], [68]. The layout and dimensions of the SRR can be found in Figure 4.19.

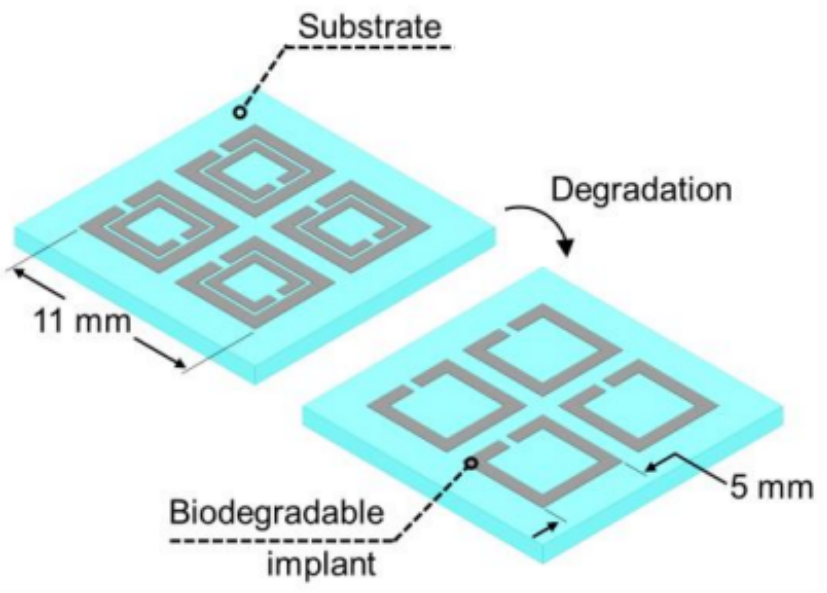



Figure 4.19. Different statues of SRR and dimensions of biodegradable implant [68].

The utilization of the Pluto SDR involves the continuous transmission of a generated sinc signal onto the Split Ring Resonator, maintaining a constant amplitude of 0 dBm. The

system operates by capturing the reflected signal emanating from the resonator, thereby extracting crucial information regarding its status, particularly pertaining to the concentration of the molecule of interest. This synergistic integration of software-defined radio and the split ring resonator enables the precise monitoring and quantification of the targeted molecule's quantity, thereby furnishing valuable insights into its dynamic behavior and the various stages of its degradation.

Figure 4.20 visually demonstrates the unique reflection behavior exhibited by each state of the split ring resonator. To investigate these behaviors, measurements were performed using a Rohde-Schwarz ZNLE6 vector network analyzer with a -10 dBm sine wave input. These measurements unequivocally reveal variations in the reflection behavior that are directly correlated with the ongoing degradation process.

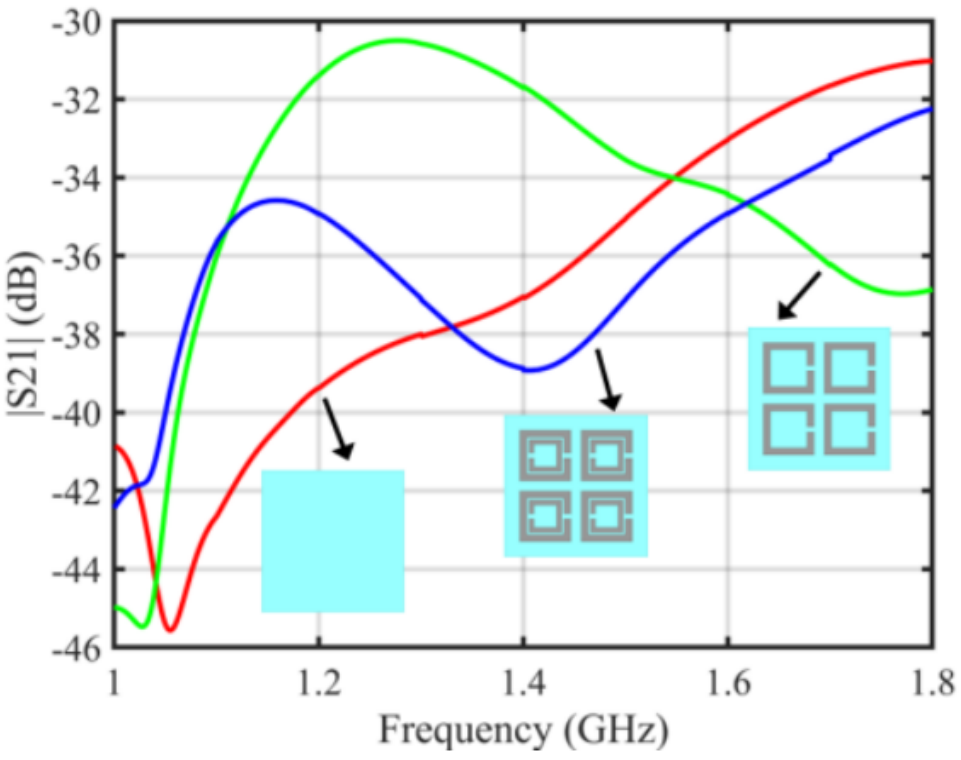



Figure 4.20. VNA output of S21 scattering parameter in dB for different SRR status and related SRR status for each curve; degraded (red), partially degraded (green), non-degraded (blue) [68].

These measurements yield valuable data, facilitating a comprehensive understanding of the distinct degradation phases exhibited by the biodegradable sensor. Furthermore, they provide insights into the evolving concentration or quantity of the molecule of interest as it undergoes the degradation process.

Figure 4.21 displays the simulation model meticulously crafted by members of the Bountenna Research Group to support the experimental setup. The experimental arrangement employed in this investigation comprises several components. First, a plexiglass container with dimensions measuring 20 cm × 20 cm × 12 cm serves as the primary housing. Within this container, there exists a 10 cm × 10 cm hole located at the bottom surface. This hole serves a crucial role in the setup. Contained within the plexiglass container is a specialized muscle phantom material, engineered to closely replicate the key properties of human tissue. This choice of material is essential for achieving accurate simulations in the study. Additionally, a smaller plexiglass cup measuring 2 cm × 2 cm × 11 cm is integrated into the setup. This cup is loaded with bacteria-MOPS, as detailed in reference [68], contributing to the comprehensive functionality of the experimental configuration.

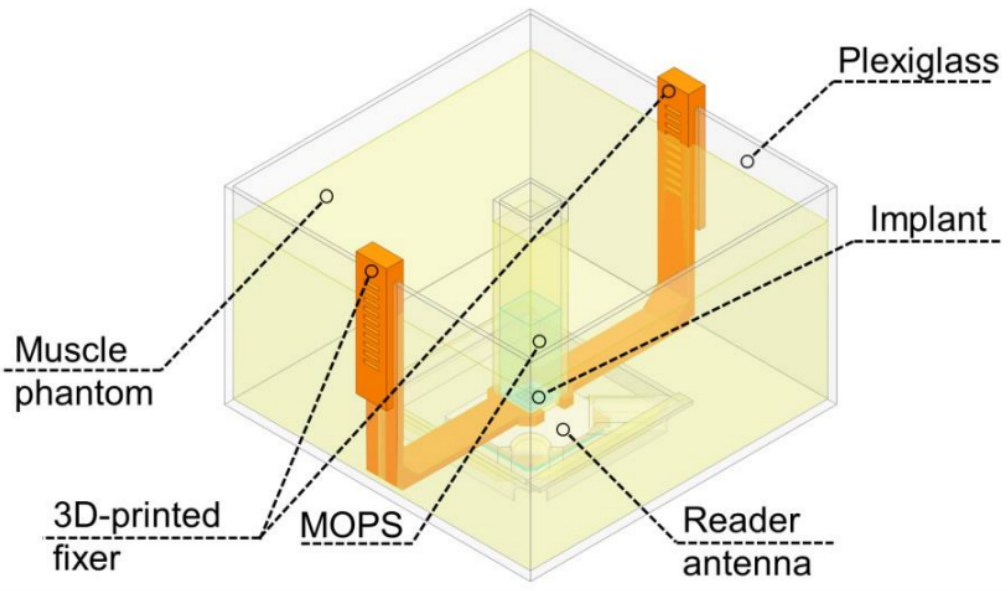



Figure 4.21. Simulation model of human tissue mimicking experimental setup [68].

The conversion of the simulation model into a tangible experimental apparatus was accomplished with precision by leveraging the Anycubic Photon M3 Max SLA printer,

prominently featured in Figure 4.22. To ensure the structural integrity and stability of the entire assembly, four custom-designed 3D-printed supports were strategically incorporated.

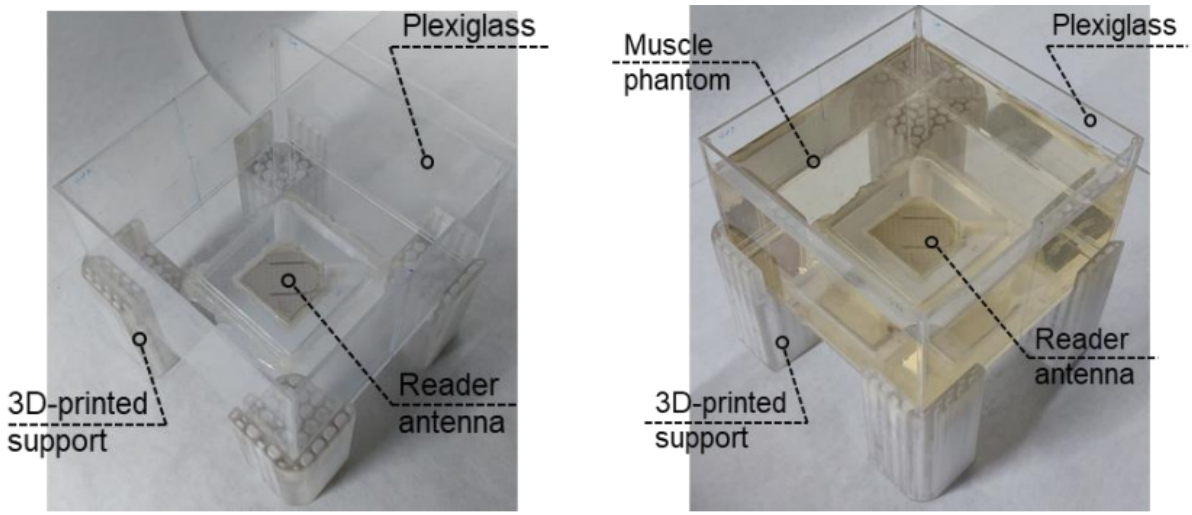



Figure 4.22. 3D printed versions of the experimental setup, without muscle phantom (left), filled with muscle phantom (right) [68].

The antenna, which is shown in Figure 4.23 with its dimensions in mm, was positioned at the bottom of a plexiglass container, and the plexiglass was filled with muscle phantom. The patch antenna proposed in reference [68] is utilized for measurements. This antenna is specifically designed to exhibit appropriate reflection and transmission coefficients within the desired frequency range of interest.

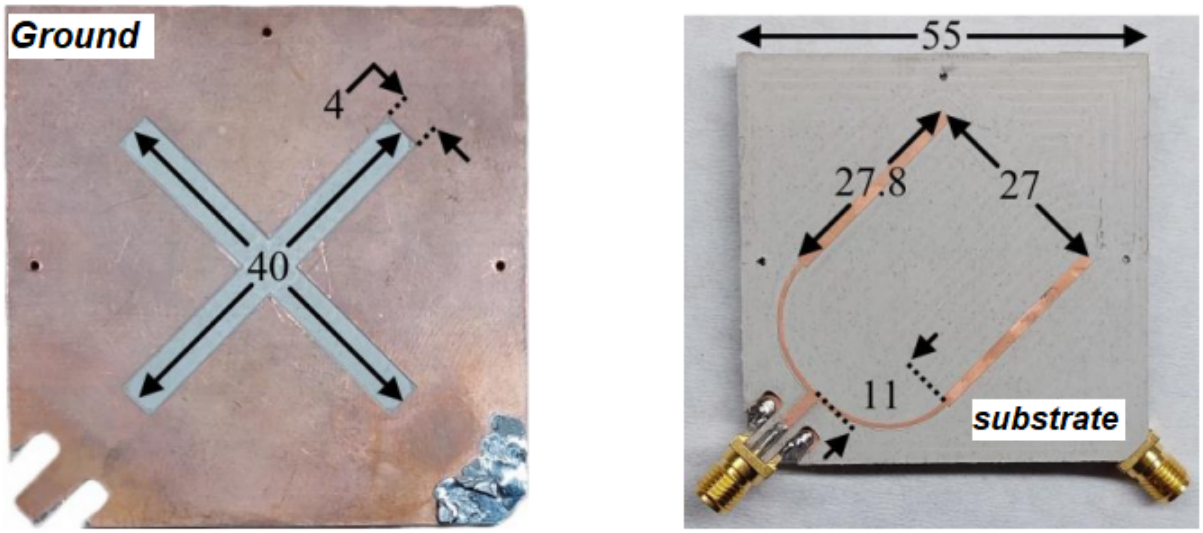



Figure 4.23. The fabricated antenna [68].

The simulation setup is finalized by incorporating 3D-printed fixers to securely position the biodegradable resonator inside the phantom, the final version is given in Figure 4.24. These 3D-printed fixers are thoughtfully designed to be movable, allowing for the flexible adjustment of the distance between the biodegradable sensor and the reader antenna. This adjustability enables the simulation of various scenarios where the Split Ring Resonator can be placed at different depths within a human body-like environment.

When the SRR is positioned near the surface of the phantom, the distance between the SRR and the reader antenna is maximized, simulating a scenario where the biodegradable sensor is placed deeper within the human body. By varying this distance, the direct coupling effect between the TX and RX antennas can be effectively reduced, resulting in more accurate and reliable measurements.

Placing the SRR deeper inside the phantom allows the reflected signal to be extracted without the need for extensive calibration. However, it is essential to consider that as the SRR is placed deeper, the attenuation inside the body also increases. This creates a trade-off between the depth of the SRR placement and the level of signal attenuation, requiring careful adjustment to strike an optimal balance between these factors. In the measurements, multiple depth values between the antenna and the SRR are employed to investigate the impact more comprehensively.

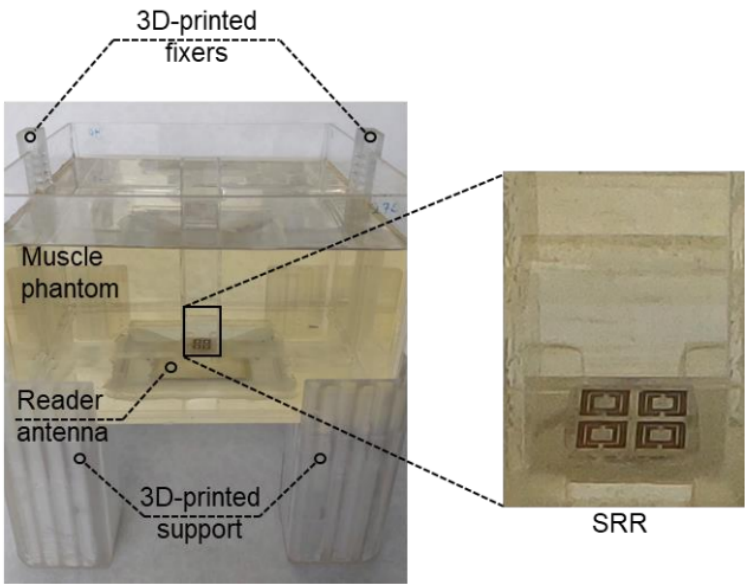



Figure 4.24. SRR included the final version of the experimental setup [68].

Figure 4.25 presents an overview of the overall system setup. In this configuration, data is extracted from the experimental setup using the Adalm Pluto Software-Defined Radio. The collected data is then saved to a PC for subsequent processing and analysis.

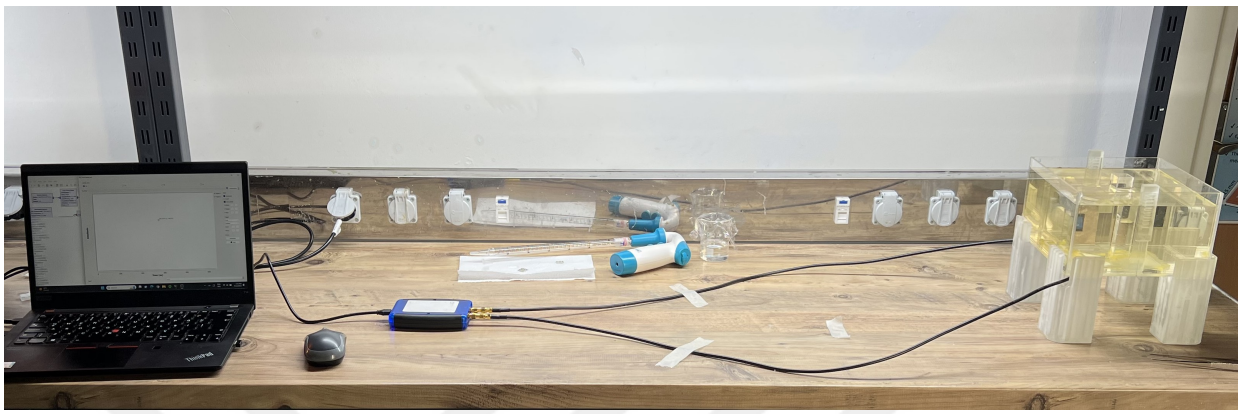


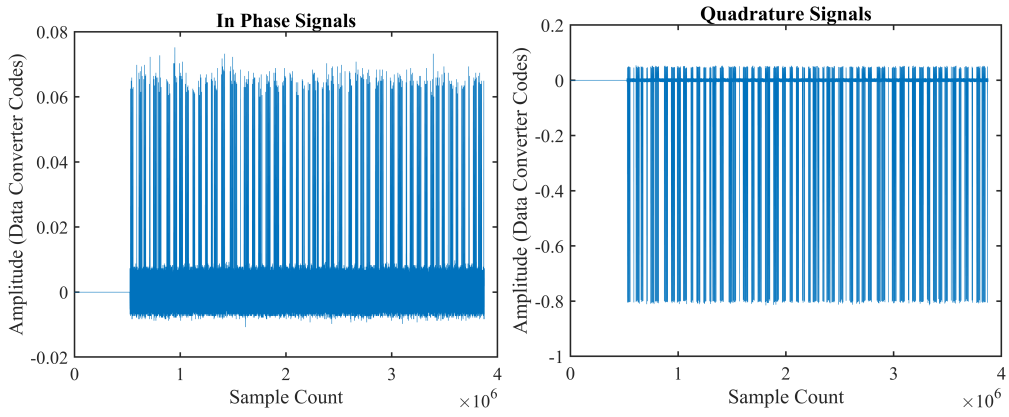
Figure 4.25. Overall system, including host PC, Pluto SDR, and experimental setup.

## 4.5. Post Processing with MATLAB

The Pluto SDR Sink operates in a cyclic mode, where the transmitted signal initially stored in the DMA (Direct Memory Access) buffers is continuously transmitted until the flowgraph is terminated. Therefore, It can be asserted that enlarging the buffer size or increasing the simulation duration will lead to an augmentation in the number of received signals.

The data received from the Adalm-Pluto SDR is stored in text files for subsequent processing in the MATLAB environment, utilizing functions available in the Signal Processing Toolbox. The raw data generated by GNU Radio Companion consists of multiple burst signals, which are represented as in-phase and quadrature components, as shown in Figure 4.26.

In this specific case, the received data includes 300 in-phase and quadrature components of the sinc signal captured by the device. The primary objective is to count the number of bursts in the given data file using MATLAB, separate them and calculate the average signal of these bursts.




(a) (b)

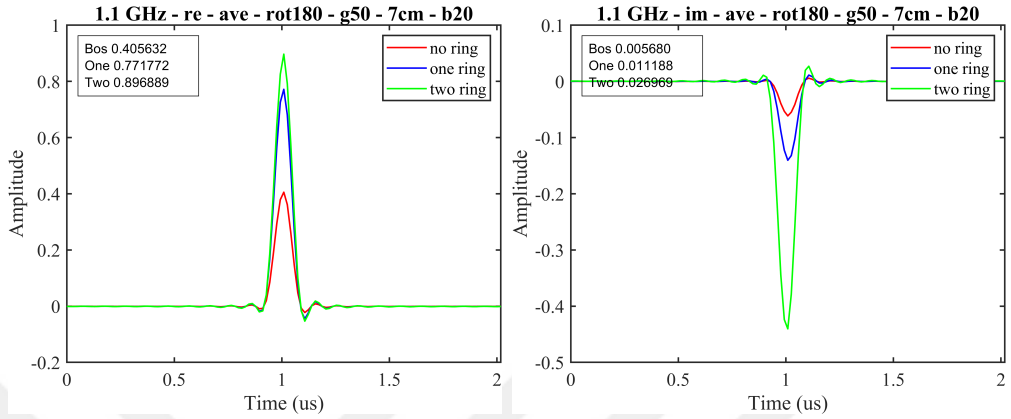
Figure 4.26. Received raw data with Pluto SDR, (a) in-phase, (b) quadrature.

In the initial step, the concatenated burst signals are separated by identifying the peak locations using the "findpeaks" function. However, this function only returns local maxima points that are larger than their two neighboring samples. To ensure that only peaks close to the maximum value of the data are considered, the return vector of the "findpeaks" function needs to be filtered. For this filtering process, a threshold value is employed, which is set to 80% of the maximum point of the data. Peaks that do not exceed this threshold are removed from the analysis and the other peak locations are saved in a vector.

To distinguish individual bursts within the raw data, a distinctive approach is employed. Specifically, a window of 125 consecutive samples is chosen, centered around the peak of each burst. By selecting the middle point of this window, which aligns with the peak, it becomes possible to isolate and identify each individual burst in the data.

The isolated individual bursts are utilized to calculate the average burst for in-phase and quadrature components, as illustrated in Figure 4.27. By aggregating the individual bursts together, an average burst waveform is obtained, representing the overall burst characteristics present in the data. Using these average signals, the magnitude signal is determined to minimize the I and Q amplitude differences caused by PLLs, which are given in Figure 4.28. The average data presented here offers comparable information to the graph in Figure 4.20. The S21 parameter is derived by dividing these average signals by the TX signal, ensuring that the differences between the signals (as depicted in Figure 4.28) remain

intact. As a result, despite the data being incomplete and can't be purely comparable with S21 data, it still holds valuable insights when comparing the power levels alignment between VNA and SDR.




(a) (b)

Figure 4.27. Calculated averages from raw data for each resonator status, (a) in-phase, (b) quadrature.

Furthermore, in numerous measurement attempts, a notable disparity was observed between the number of in-phase and quadrature components, thus rendering the underlying assumption of alignment between these components invalid. The existing code relied solely on the detection of peaks in the in-phase signal, assuming that the peaks in the quadrature signal coincided with the corresponding samples. However, it was discovered that this assumption did not hold true.

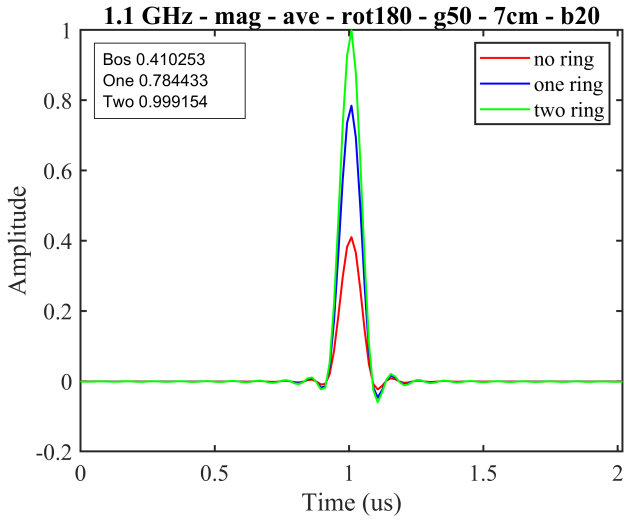



Figure 4.28. Magnitude of received signals for each SRR status in data converter codes.

To address this issue, a modified approach was implemented, considering the peaks in both the in-phase and quadrature signals. This alternative methodology was employed to accurately determine the burst boundaries and improve the reliability of the data. The outcomes of this revised approach are presented in Table 4.2, which showcases the pertinent data obtained from the quadrature signal. A comparative analysis of the standard deviations (SD) reveals that the alternative method offers more robust and dependable results.

Table 4.2. Comparison between proposed method and previous method.

| By Using I Peaks (Previous) | | | |
|---|---|---|---|
| Max Peak | 0.0816 | 0.00821 | 0.0879 |
| Min Peak | 0.0508 | 0.0532 | 0.0547 |
| Ave Peak | 0.0668 | 0.0701 | 0.0741 |
| SD | 0.0064 | 0.0059 | 0.0067 |
| By Using Q Peaks (Proposed) | | | |
| Max Peak | 0.0811 | 0.0796 | 0.0821 |
| Min Peak | 0.0711 | 0.0611 | 0.0616 |
| Ave Peak | 0.0713 | 0.0716 | 0.0732 |
| SD | 0.0028 | 0.0026 | 0.0028 |

To resolve the problem of inconsistent data counts between the in-phase and quadrature components, two potential reasons are identified. The first reason is related to the variations in IQ amplitudes. Due to the randomization introduced by the Phase Locked Loop, the amplitudes of the received I and Q components can be random. In cases where one of the amplitudes approaches zero, the signal count algorithm, which relies on detecting peaks in the received data, may not accurately interpret all the small peaks. Consequently, the count of the component with a zero amplitude will be significantly lower than the count of the other component.

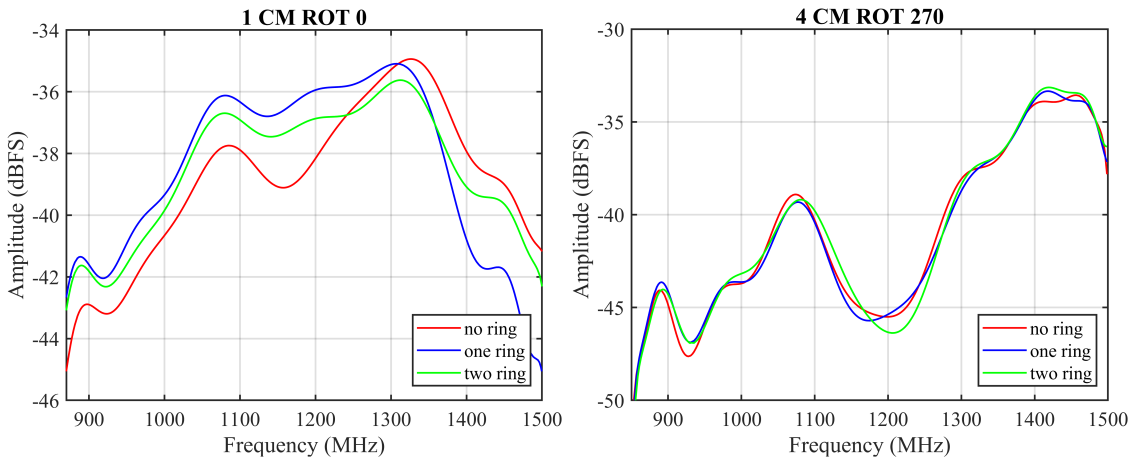
The second reason for the inconsistent data counts is the limitation of Universal Serial Bus (USB) 2.0. As USB 2.0 has a maximum data rate of 7 mega samples per second (MSPS), data drops can occur between each buffer during the transfer process. This bottleneck can

lead to missing or dropped data, resulting in a discrepancy between the counts of the I and Q components.

Once reliable data is extracted, the resulting magnitude signals can be visualized for three potential sensor scenarios in both the time and frequency domains. In the time domain, the magnitude signals provide insight into the peak amplitude variations over time and between different statuses of the sensor. By plotting 125 samples of the magnitude signals against the corresponding time instances and aligning them to the same initial point, the temporal characteristics and any observable patterns or trends can be analyzed.

In the frequency domain, the magnitude signals can be transformed using Fast Fourier Transform (FFT) function to obtain their frequency representations. This enables the identification of dominant frequencies or spectral components present in the signals.

Thanks to the integration of a Python module into the final flowgraph, the data provided to Matlab can now be utilized to create wide-frequency graphs similar to the ones produced by VNA outputs. Figure 4.29 below illustrates these graphs, demonstrating the efficient use of SDR as a cost-effective alternative to the expensive VNA tool.




(a) (b)

Figure 4.29. Proposed SDR results for different status and parameters in dBFS, (a) 1 cm - rotation 0 degrees, (b) 4 cm - rotation 270 degrees.

# 5. RESULTS AND COMPARISON

This chapter provides an in-depth analysis and comparison between the results obtained using Software-Defined Radio and Vector Network Analyzer methods. The measurement phase is divided into two phases.

In the first phase, specific critical frequency points are selected from the VNA results, and measurements are conducted only for these selected frequency values. This focused approach allows for a detailed examination of the behavior of the system at these critical frequencies.

In the second phase, SDR is utilized as a replacement for VNA. The measurements are carried out using SDR across a wide frequency range, enabling a comprehensive exploration of the system's performance throughout the frequency spectrum. This approach highlights the effectiveness and versatility of SDR as an alternative to the traditional and often expensive VNA tool.

## 5.1. Results from Selected Points

In the preliminary stage of the measurements, a meticulously selected set of 55 distinct frequency points is employed, encompassing the frequency range from 800 MHz to 1.5 GHz. These measurements are conducted while systematically varying the depth from 2 cm to 10 cm, representing either the placement of a biodegradable sensor inside the human body or the spatial separation between the sensor and the antenna.

Furthermore, within a given depth value, diverse positional arrangements for the sensor can be achieved by adjusting its rotation within the experimental setup. This configuration offers four distinct placement possibilities for the sensor, denoted by their respective rotation angles with respect to a designated reference point: rotation 0, 90, 180, and 270 degrees. This comprehensive phase of measurement furnishes us with the requisite data necessary to make a judicious determination about the suitability of employing Pluto SDR as a substitute for a vector network analyzer.

The 55 specific frequency points are chosen based on the graphical outputs provided by the Vector Network Analyzer, as depicted in Figure 5.1. This figure illustrates the received signal power ratio to transmitted power by the on-body reader antenna for different states of the biodegradable sensor. The frequency points that exhibit significant power level variations among the different sensor states are particularly selected. Consequently, Pluto SDR facilitates a simpler differentiation of various statuses. For instance, in Figure 5.1, frequencies of 1.1, 1.2, 1.3, and 1.4 GHz are picked as measurement points due to their distinct power characteristics.

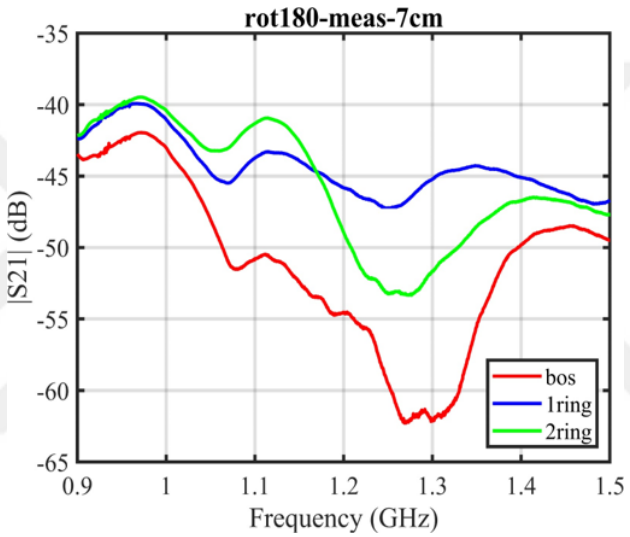
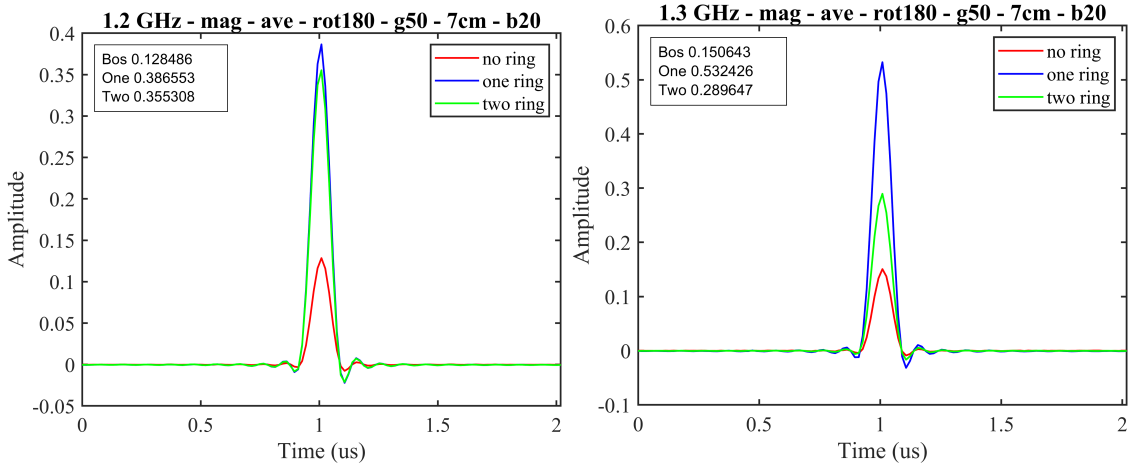



Figure 5.1. VNA output for different SRR statuses at 7 cm Depth for S21 scattering parameter.

Figure 5.2 displays magnitude curves for two different frequencies using SDR received data by the transmission of a sinc signal with a 20 MHz bandwidth. Additionally, Figure 4.27 includes the results for a 1.1 GHz frequency point obtained from the SRR with a sensor-to-antenna distance of 7 cm and a sensor rotation angle of 180 degrees.

Through a comparative analysis of amplitude level alignments among "no ring," "one ring," and "two ring" statuses, the correctness of each measurement is evaluated using the vector network analyzer and the software defined radio graphs. The figures, which present data from both VNA and SDR measurements, demonstrate a significant consistency between the two approaches. Remarkably, out of the total 55 measurements conducted using SDR, an impressive 90.1% of them align precisely with the corresponding VNA results.




(a) (b)

Figure 5.2. SDR measurement results in data converter codes for selected frequencies at 7 cm, (a) 1.2 GHz, (b) 1.3 GHz.

Upon initial examination, comparing VNA's S21 graphs with SDR's received signal graphs in the time domain might not be the most appropriate approach for a meaningful comparison of these quantities. This is likely attributed to the inclusion of supplementary calculations, encompassing the consideration of other scattering parameters when determining the S21 parameters, as well as accounting for the coupling effect between the two ports of the reader antenna. Primarily some calibration methods are employed to mitigate the coupling effect between TX and RX ports.

During calibration, it is assumed that the TX signal power remains constant for each measurement point in the given range. As a result, the alignments for different sensor statuses do not change since the used calibration signal is the same for each one. Therefore, calibration techniques designed to isolate the reflected signal are not employed in these particular cases, as they have no impact on the outcome. As a consequence, the given SDR results encompass a combination of both reflected and coupled signals, making the alignment of power levels a crucial factor in providing data that can be effectively compared with VNA measurements.

Furthermore, it is important to highlight the differences in measurement regimes employed by VNA and SDR. A VNA employs a series of calculations to determine S-parameters. For instance, the S21 transmission parameter is computed by dividing the

signal transmitted to port 2 by the incident signal applied to port 1. However, this capability is not available for Pluto SDR. There is no specified hardware to couple TX and RX signals inside the device and there is no such powerful processor in it. As a result, the SDR digitizes the signal that it captured from its RX terminal and amplifies it with stable gain throughout the entire measurements at a fixed value, such as 40 or 45 dB, while the VNA uses more sophisticated and advanced computations to find scattering parameters during the scanning of the given frequency range.

Due to this variation, comparing power levels between VNA and SDR measurements becomes meaningless. Nevertheless, different SRR statuses can be compared in measurements carried out by both devices. The key point is that the alignments among different statuses should be consistent in both VNA and SDR graphs. This is one of the reasons why received signals are left in the time domain. Additionally, upon examining Figures 5.2 and 4.27, it becomes evident that the received signal shapes are almost identical, and their central frequencies are equal. Consequently, when applying FFT to these signals, the resulting frequency domain images will be virtually indistinguishable, except for differences in their power levels with respect to full scale, which can also be observed in the time domain plots. During the investigation of instances where SDR fails to yield results consistent with the VNA, it was observed that certain frequency regions displayed abrupt changes in their power levels. This phenomenon became evident when analyzing plotted curves, exemplified in Figure 5.3, where a significant power difference, specifically in the red "no ring" curve at 1 GHz, was observed.

The captured bandwidth of 20 MHz simultaneously encompasses this substantial power variation within the same frequency band. Consequently, such considerable differences, reaching up to 15 dB, have a pronounced impact on the components utilized in the received path even if differences are less than the data converters dynamic range. Notably, analog components, including the low-noise amplifier, can be driven into the saturation region due to the magnitude of these differences.

The saturation of components in the received path can introduce non-linear effects, leading to distorted signals and inaccuracies in the data collected by the SDR system. These variations must be accounted for and mitigated appropriately to ensure the reliability and

fidelity of the measurements obtained through SDR, particularly in scenarios where close comparisons are made with VNA results. Addressing these challenges is imperative to maintain the overall integrity and validity of the experimental outcomes.

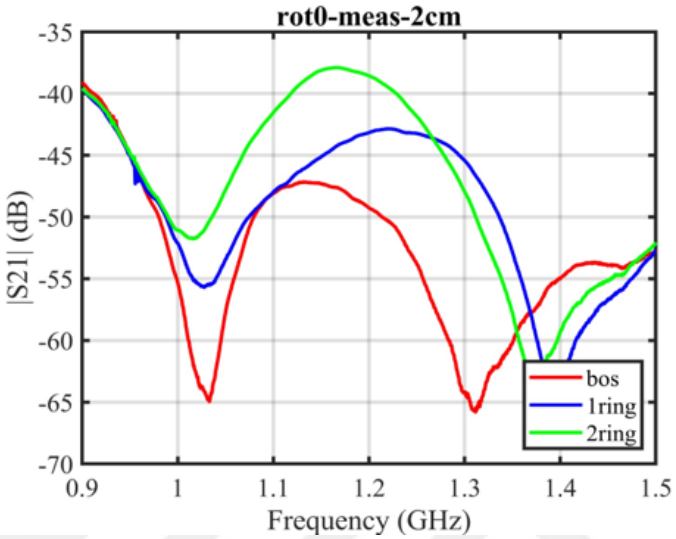



Figure 5.3. VNA output for different SRR status at 2 cm depth for S21 scattering parameter.

To mitigate this effect, the transmitted signal bandwidth is reduced to 2 MHz, and measurements are repeated at the problematic frequency points. Employing a narrower signal in transmission improves the overall efficiency of the results, achieving a success rate of 94.45%. However, there is a tradeoff associated with reducing the signal bandwidth, as it impacts the amount of information received per unit of time. Consequently, when employing a narrow-band signal for scanning a limited segment of the spectrum, the time required to cover the entire frequency range increases.

Furthermore, narrowing the signal's bandwidth extends its non-zero period in the time domain. This elongation can pose challenges in distinguishing the reflected signal from the coupling effect. Particularly, when the biodegradable sensor is placed deep within human tissue, the longer transmitted signal period causes the reflected signal to combine with the coupling signal. While placing the sensor deeper may enhance the likelihood of distinguishing the reflected signal, it also leads to higher attenuation of the reflected signals due to the extended time spent inside the human body. Hence, there exists a crucial balance

that must be maintained between signal bandwidth, time-domain characteristics, and the depth of sensor placement. Striking the right balance is essential to ensure accurate and reliable measurements, particularly when dealing with complex scenarios involving signal reflections and coupling effects within biological tissues.

In conclusion, the initial phase of measurements convincingly demonstrates that the SDR device can serve as a viable alternative to the costly and cumbersome VNA in the sensing system. The results obtained through SDR, especially after applying the necessary adjustments, showcase a high level of agreement with VNA measurements.

## 5.2. Using SDR as a VNA

Following the solution of all initial doubts concerning the application of SDR within the system, as confirmed by the attainment of highly satisfactory measurement results during the first phase, it has become evident that SDR can be suitably tailored to address the inherent challenges of handling VNA burdens in the proposed network. Initially, there were concerns that SDR might not be a suitable replacement due to the PLL randomization issue, despite the promising results obtained in the first phase.

A software module has been developed to monitor and control the in-phase and quadrature components' status within user-specified limits. The module is implemented using Python programming language and takes advantage of GNU Radio's capabilities to modify flow-graph variables. The main purpose of this system is to provide users with the ability to obtain only the desired I and Q signal pairs, allowing for precise customization of the signal processing chain and reducing PLL's non-deterministic phase and amplitude offset issue. By integrating Python with GNU Radio, the software module can intelligently track and regulate the behavior of I and Q components, ensuring that they remain within the user-defined variation set.

In addition to the described software module, there arises a crucial need for an automation system that can dynamically adjust the software defined radio's local oscillator frequency to predetermined locations. This automation system facilitates efficient scanning of the designated frequency range, eliminating the manual burden placed on users to set

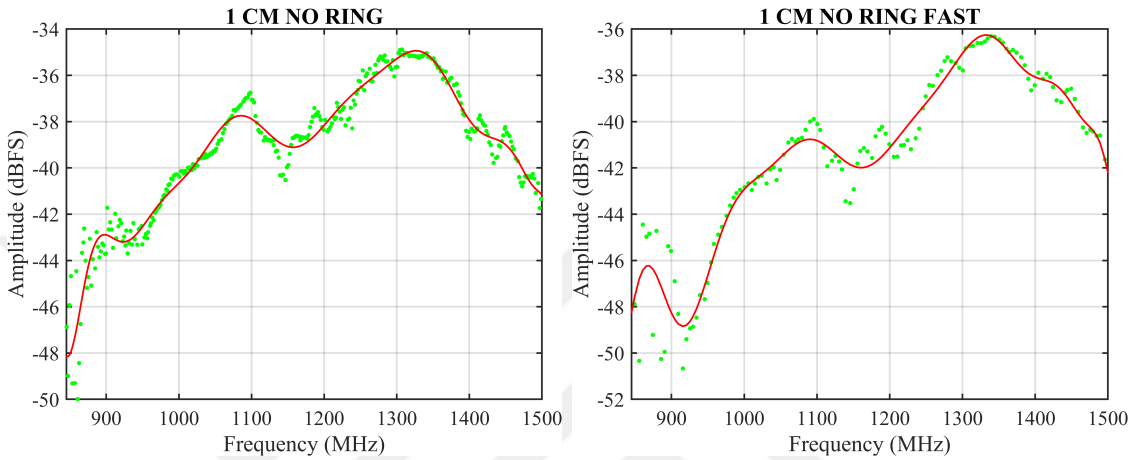
essential parameters, such as LO frequency and sampling rate, either during simulation or at the outset. Currently, there is a conspicuous absence of any existing automation mechanism or commented system that addresses this specific requirement. Users are typically expected to manually configure these parameters, which can be time-consuming, error-prone, and labor-intensive. To bridge this gap, the proposed automation system, which is included in the defined software module will offer the capability to seamlessly alter the SDR LO frequency in between the specified range and desired step sizes, thereby facilitating comprehensive scanning across the desired frequency range. By implementing an automated approach, users can significantly enhance the efficiency of their experiments.

### 5.2.1. Comparison Among Different Step Sizes

Two distinct step sizes are utilized for generating frequency curves for each SRR status: 2 MHz and 5 MHz. With each step size, adjustments are made to the bandwidth of the TX signal. This adaptation is necessary as larger step sizes necessitate broader signals in the frequency domain to effectively capture the entire frequency spectrum without any degradation. Correspondingly, transmitted signal bandwidths of 15 MHz and 20 MHz are employed.

Employing smaller step sizes in the frequency scanning process results in a greater number of frequency locations that need to be explored. As an illustration, when dealing with 2 MHz signals within the frequency range of 800 MHz to 1500 MHz, a total of 350 unique frequency points will be searched. In contrast, if a 5 MHz step size is utilized, the number of unique frequency points to be searched reduces to 140. Figure 5.4 displays the frequency points and fitted curves for both scenarios, points highlighted in green signify that the employed software module effectively captures signals according to the user-defined combinations. To enhance the clarity of the extracted data from the SDR, MATLAB's polynomial curve fitting techniques are employed for visualization. The graphs reveal a noticeable similarity in the overall shape of both curves. However, there are subtle distinctions present in the lower decibel values, where the SDR's sensing capability is relatively weaker compared to the regions with higher power levels. Moreover, these measurements are taken at different times, and the effect caused by different setup settings is more severe in lower-power regions.

Furthermore, although the software executed on the host computer adeptly manages these alterations in frequency, the hardware components of the SDR, including the Pluto SDR processor and FPGA, experience heightened demands owing to the escalated frequency transitions. As a result, a significant delay in the measurement duration is anticipated when contrasting different step sizes.




(a) (b)

Figure 5.4. Illustration of each frequency point and fitted curves for no ring case by using different step sizes, (a) 1 cm no ring 2 MHz step size, (b) 1 cm no ring 5 MHz step size.

Predicting precise timing specifications for the system is challenging due to the unpredictable behavior of the PLL. The software module's success in identifying the user-specified combination can occur on the initial attempt or even after multiple iterations. As a result of this variability, pinpointing the exact timing is difficult. Nevertheless, based on the measurements conducted thus far, scanning a designated frequency range using a 5 MHz step size completes in approximately 19 minutes, while employing the other step size necessitates around 42 minutes for the frequency band scan. Choosing between step sizes presents a trade-off between speed and accuracy. Utilizing more frequency points enhances the resolution of the final graph, providing finer details in the results. However, this comes at the cost of increased measurement duration due to the higher number of frequency locations being searched, impacting the overall speed of the scanning process. Conversely, larger step sizes lead to faster measurements as the number of frequency points to scan is reduced. Nevertheless, this comes at the expense of lower resolution, potentially sacrificing some accuracy in the final graph representation.

Table 5.1 provides an overview of the two step sizes, encompassing several factors that have been detailed in the preceding sections. However, one aspect not previously elaborated clearly upon is "accuracy." Accuracy serves as the metric by which the effectiveness of the designed software module in detecting user-specified combinations of I/Q signals.

Table 5.1. Comparison of simulation parameters, 2 MHz and 5 MHz step sizes.

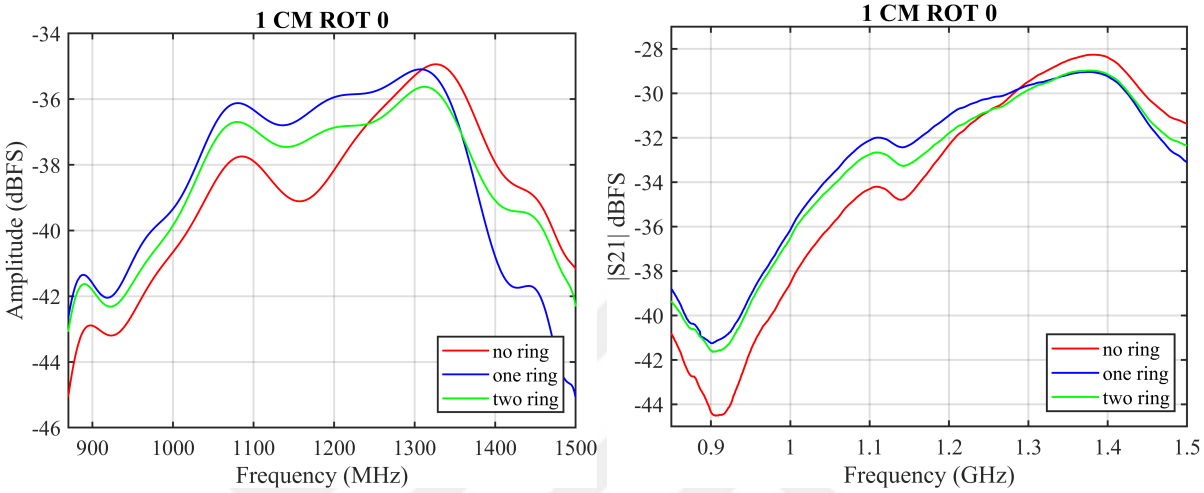
| Step Size | Resolution | Timing | Accuracy |
|---|---|---|---|
| 2 MHz | 0.4 MHz | 42 mins | 99.4% |
| 5 MHz | 1 MHz | 19 mins | 99.3% |

Ultimately, the selection of an appropriate step size depends on the specific requirements of the application. If higher accuracy and resolution are of utmost importance, a smaller step size may be preferred, even though it leads to longer measurement durations. On the other hand, when speed is a critical factor, a larger step size could be more suitable, at the cost of sacrificing some level of accuracy and resolution. Striking the right balance between speed and accuracy is essential to ensure optimal performance and meet the desired objectives of the frequency scanning process.

### 5.2.2. Comparison of SDR and VNA Results

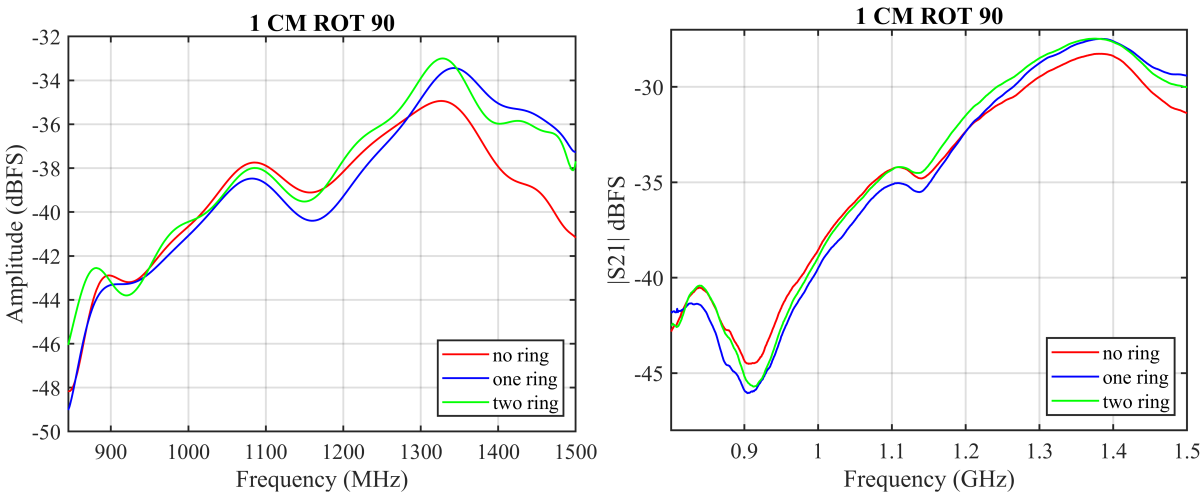
To establish frequency curves for different SRR statuses, a diverse range of depth and rotation combinations are employed. This section provides a comprehensive overview of outcomes for depths of both 1 cm and 4 cm. The SDR graphs, with a step size of 2 MHz, are given with the results from the vector network analyzer. Additionally, outcomes using a 5 MHz step size are also presented. The amplitude values for both SDR and VNA curves are expressed in the dBFS unit. GNU Radio's data normalization process confines Pluto SDR data within the range of -1 to 1. Consequently, performing FFT on these time domain data yields power level results in the frequency domain, represented in dBFS units. The VNA graphs are also annotated in dBFS units. Given that S21 represents a power ratio transferred from the TX to the RX within the utilized measurement setup, this ratio remains invariant regardless of the chosen TX power. As a result, both quantities are graphed in reference to the 0 dB power level. Within the context of a 1 cm depth and a

rotational angle ranging from 0 to 90 degrees, we present the results obtained through both devices respectively, Software-Defined Radio and Vector Network Analyzer. Figure 5.5 and Figure 5.6 specifically encapsulate these outcomes, offering a detailed visual representation of the data collected during this particular experimental configuration.




(a) (b)

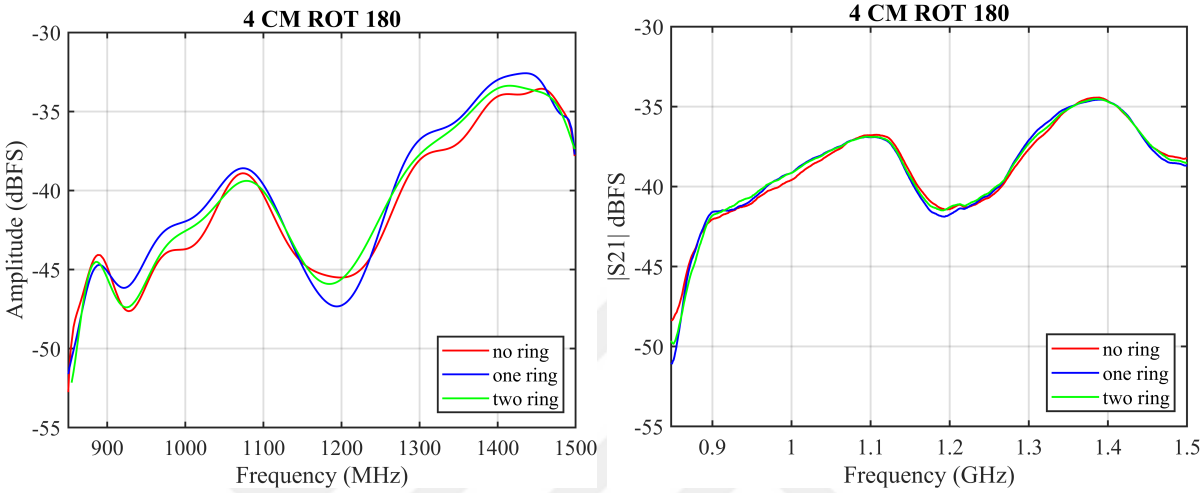
Figure 5.5. SDR and VNA measurement results for 0 degrees rotation at 1 cm in dBFS, (a) SDR, (b) VNA.




(a) (b)

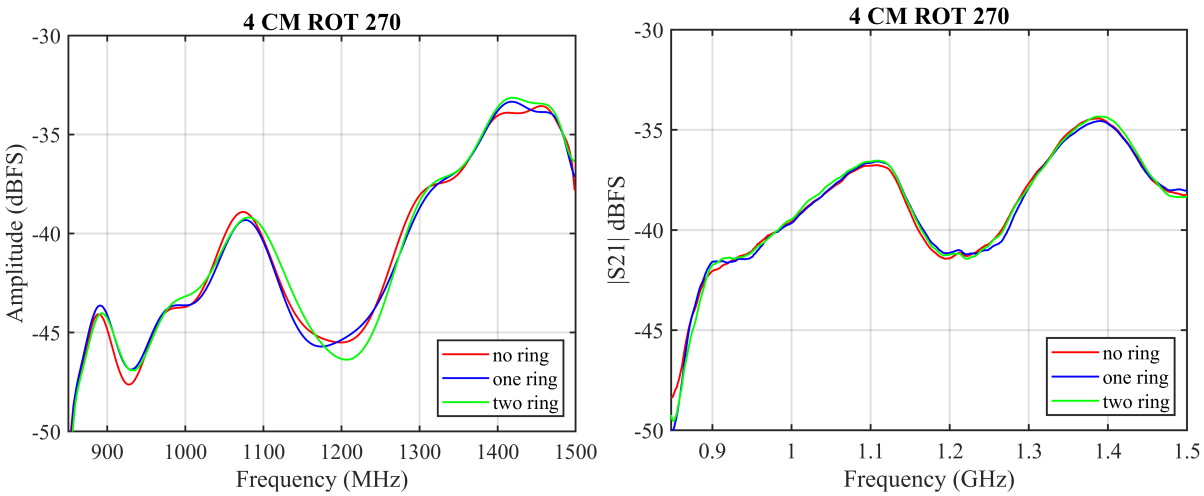
Figure 5.6. SDR and VNA measurement results for 90 degrees rotation at 1 cm in dBFS, (a) SDR, (b) VNA.

In Figures 5.7 and 5.8, you can observe a comprehensive depiction of the outcomes obtained through the combined use of Software-Defined Radio and Vector Network Analyzer technologies. These results were obtained while focusing on a 4 cm depth and encompassing a rotational angle spectrum spanning from 180 to 270 degrees.




(a) (b)

Figure 5.7. SDR and VNA measurement results for 180 degrees rotation at 4 cm in dBFS, (a) SDR, (b) VNA.



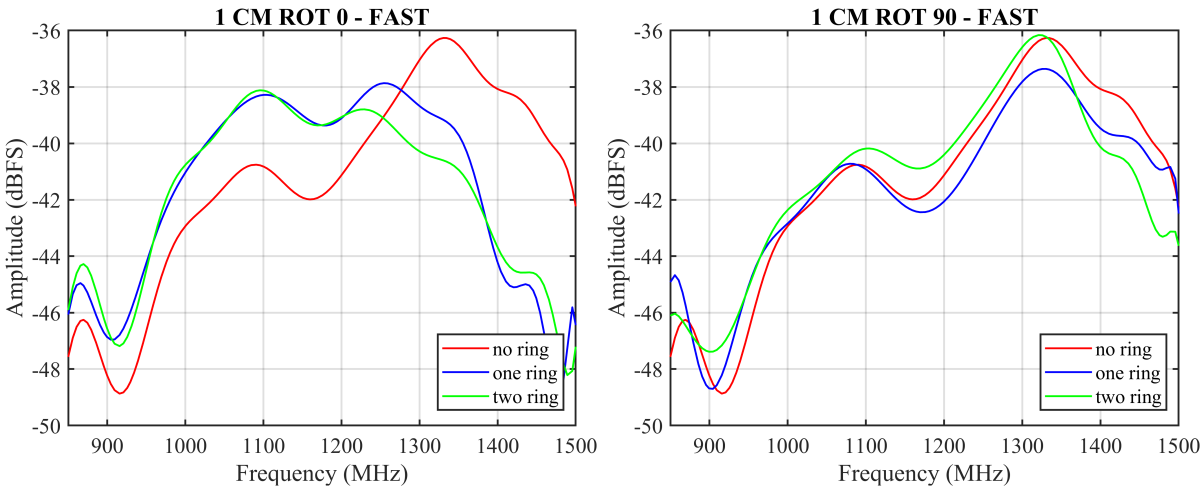

(a) (b)

Figure 5.8. SDR and VNA measurement results for 270 degrees rotation at 4 cm in dBFS, (a) SDR, (b) VNA.

Upon careful examination of the graphs, noticeable fluctuations become apparent in terms of power levels, signifying the presence of some differences between the two systems. It's important to expand the underlying differences in measurement methodologies between the Pluto SDR and the Vector Network Analyzer. In the context of Pluto SDR measurements, amplitude modulated signals are both transmitted and received, encompassing a sweep across frequencies from 800 MHz to 1500 MHz. Moreover, the Pluto SDR lacks the capability for unmodulated transmission, so it is not possible to take the RF mixer out of operation with software or hardware modifications. On the contrary, the VNA employs an unmodulated carrier signal as its transmitting signal.

In addition to previously explained SDR and VNA measurement differences, the complexities of curve fitting methods, inherent non-ideal behaviors of active components within the Pluto SDR, and differences in measurement setup may also affect the results obtained with Pluto SDR. These listed fundamental distinctions between the two devices inherently lead to minor discrepancies in their responses.

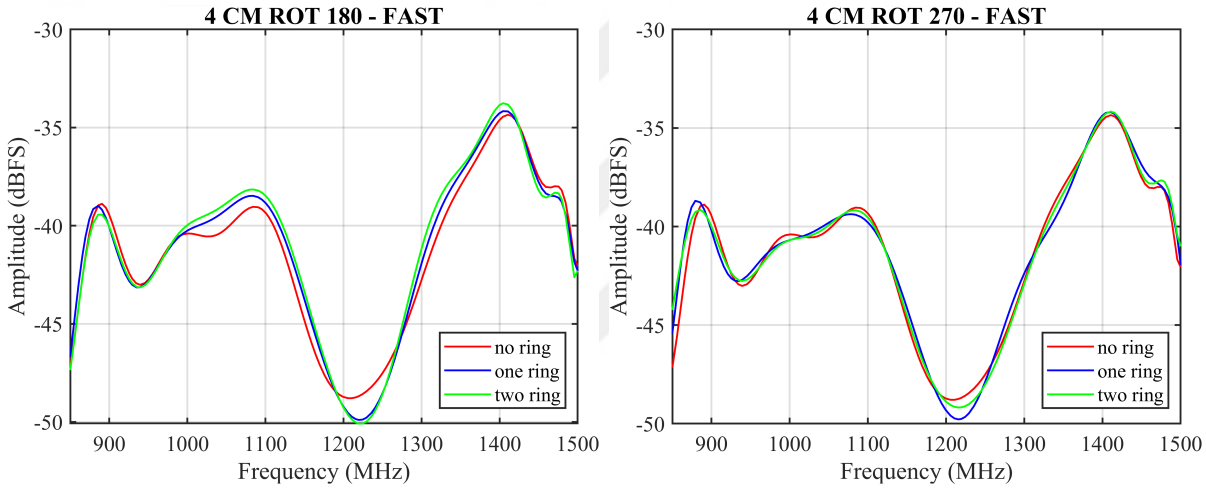
The figures in 5.9 and 5.10 present the outcomes achieved using solely Pluto Software Defined Radio during the shortened measurement time interval, with step sizes of 5 MHz.




(a) (b)

Figure 5.9. SDR measurement results for 0 and 90 degrees rotations at 1 cm with 5 MHz step size in dBFS, (a) rotation 0, (b) rotation 90.

Conversely, upon comparing the curve characteristics for each SRR status, the Pluto SDR exhibits commendable outcomes. The central objective of this system is to proficiently differentiate various statuses of a biodegradable sensor from external sources when placed within the body. In light of this objective, when the plotted graphs are evaluated, the data obtained from the Pluto SDR aligns well with the VNA plots across a substantial portion of the frequency range, with the exception of the measurement using a 5 MHz step size for a depth of 1 cm and a rotation of 90 degrees, which might be attributed to a lower resolution, the outcomes obtained through narrower step sizes exhibit strong alignment with the VNA results.




(a) (b)

Figure 5.10. SDR results for 180 and 270 degrees rotations at 4 cm with 5 MHz step size in dBFS, (a) rotation 180, (b) rotation 270.

For instance, when considering the 1 cm depth, the distinct statuses of the SRR are discernible from both sets of graphs. Furthermore, when analyzing the graphs for a depth of 4 cm, a close alignment between each curve corresponding to individual SRR statuses is anticipated. In this context, the Pluto SDR produces reasonable and congruous outcomes when compared to the VNA, validating its efficacy for these specific depth and rotation values.

While fluctuations in power levels arising from non-ideal components, gain, and modulation differences are evident, the Pluto Software Defined Radio's satisfactory performance in distinguishing split ring resonator statuses is notable. The alignment between the Pluto

SDR and VNA plots, particularly for several depth and rotation values, underscores the system's reliability in providing reasonable results. This compatibility with the vector network analyzer outcomes substantiates the Pluto SDR's viability as an effective tool for differentiating biodegradable sensor statuses, presenting promising prospects for its application in practical scenarios.

# 6. CONCLUSION AND FUTURE WORK

This thesis focuses on designing a Data Capture Module for Multi-scale Communication Systems and generating a Sine-Gaussian Signal for RF medical sensing applications. The chosen hardware for implementation is the Adalm-Pluto SDR from Analog Devices. The primary objective of this research is to capture reflected waves from a biodegradable sensor placed inside the human body, taking into account the signal attenuation in human tissue. The project is a continuation of the Bountenna Research Group's efforts toward developing a communication system based on engineered bacteria.

A notable contribution of this study is the innovative use of a low-cost SDR for in-body activity measurements from a biodegradable sensor. While some literature exists on using SDR for imaging tissues inside the body, the integration of biodegradable sensors into a communication system through SDR is a novel and unexplored area of research. By employing SDR as a replacement for traditional and expensive Vector Network Analyzers, the communication system becomes wearable and can be carried on the human body, allowing for remote and non-intrusive monitoring of medical conditions.

The research methodology involves conducting measurements for various frequency and depth values. In the initial phase, the Pluto SDR's capability is rigorously tested to determine its suitability as a VNA replacement. The measurement results are compared with those obtained using a VNA. The assessment shows that in 90.1% of the measurements, where a wideband transmission signal is used, the SDR device produces identical results to the VNA. With further bandwidth adjustment, the agreement increases to 94.45%, demonstrating the potential of SDR as a reliable and cost-effective alternative to VNA.

In the subsequent phase, the VNA's specifications for examining a wide frequency range are emulated in the SDR through software. This enables the generation of VNA-like plots using Matlab software, utilizing the data provided by the SDR. The results obtained from this phase are compared with those of traditional VNA measurements, validating the accuracy and effectiveness of the proposed SDR-based approach.

Although SDR is versatile and can operate in various applications ranging from amateur radio to satellite communications, its analog and digital components are chosen to accommodate a wide range of frequencies. However, this broad frequency range may lead to reduced performance in specific tasks, such as operating between 1 to 1.5 GHz. To address this limitation, future work includes selecting or designing specialized components tailored to the targeted frequency range, thereby improving the selectivity and sensitivity of the SDR system.

Additionally, the SDR's digital capabilities are limited to some extent due to its low-cost nature. As a result, advanced signal processing tasks are offloaded to a host computer. The FPGA, DSP, and CPU resources in the SDR are primarily utilized for essential communication tasks, such as decimation, interpolation, running Linux, DDS, DMA, etc. If the FPGA resources of Pluto SDR were sufficient, future work could explore the possibility of implementing DSP applications on the FPGA and DSP side, thereby reducing the dependency on the host PC. Furthermore, the design of a custom task-specific circuit could be considered to enhance system speed and mobility.

# REFERENCES


1. Fallahpour, M., *Synthetic Aperture Radar-Based Techniques and Reconfigurable Antenna Design for Microwave Imaging of Layered Structures*, Ph.D. Thesis, Missouri University of Science and Technology, 2013.

2. Ahmed, S. S., "Microwave Imaging in Security — Two Decades of Innovation", *IEEE Journal of Microwaves*, Vol. 1, No. 1, pp. 191–201, 2021.

3. Reid, S., J. Tam and M. Yang, "Technospheric Mining of Rare Earth Elements from Bauxite Residue (Red Mud): Process Optimization, Kinetic Investigation, and Microwave Pretreatment", *Scientific Reports*, Vol. 7, No. 1, p. 152, 2017.

4. Raza, A., A. Jabbar, D. A. Sehrai, H. Atiq and R. Ramzan, "SDR Based VNA for Characterization of RF Sensors and Circuits", *2021 1st International Conference on Microwave, Antennas & Circuits (ICMAC)*, pp. 1–4, Islamabad, Pakistan, 2021.

5. Osumi, N. and K. Ueno, "Microwave Holographic Imaging of Underground Objects", *IEEE Transactions on Antennas and Propagation*, Vol. 33, No. 2, pp. 152–159, 1985.

6. Lackey, R. I. and D. W. Upmal, "Speakeasy: The Military Software Radio", *IEEE Communications Magazine*, Vol. 33, No. 5, pp. 56–61, 1995.

7. Nikolova, N. K., "Microwave Imaging for Breast Cancer", *IEEE Microwave Magazine*, Vol. 12, No. 7, pp. 78–94, 2011.

8. Wang, Y., A. Abbosh, B. Henin and P. Nguyen, "Synthetic Bandwidth Radar for Ultra-Wideband Microwave Imaging Systems", *IEEE Transactions on Antennas and Propagation*, Vol. 62, No. 2, pp. 698–705, 2014.

9. Marimuthu, J., K. S. Bialkowski and A. M. Abbosh, "Software-Defined Radar for Medical Imaging", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 64, No. 2, pp. 643–652, 2016.

10. Ahmed, S. S., A. Schiessl, F. Gumbmann, M. Tiebout, S. Methfessel and L. P. Schmidt, "Advanced Microwave Imaging", *IEEE Microwave Magazine*, Vol. 13, No. 6, pp. 26–43, 2012.

11. Meaney, P. M., M. W. Fanning, D. Li, S. P. Poplack and K. D. Paulsen, "A Clinical Prototype for Active Microwave Imaging of the Breast", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 48, No. 11, pp. 1841–1853, 2000.

12. Fear, E., X. Li, S. Hagness and M. Stuchly, "Confocal Microwave Imaging for Breast Cancer Detection: Localization of Tumors in Three Dimensions", *IEEE Transactions on Biomedical Engineering*, Vol. 49, No. 8, pp. 812–822, 2002.

13. You, K. Y., Y. Song, N. Derek, C. Lee, Z. Abbas, C. Meng and L. S. Yeng, "Low-Cost Vector Network Analyzer for Communication Devices Testing - Brief Review", *International Journal of Advances in Microwave Technology*, Vol. 2, No. 1, pp. 94–98, 2017.

14. Bialkowski, K. S., J. Marimuthu and A. M. Abbosh, "Biomedical Imaging System Using Software Defined Radio", *2015 IEEE International Symposium on Antennas and Propagation, USNC/URSI National Radio Science Meeting*, pp. 542–543, Vancouver, BC, Canada, 2015.

15. Stancombe, A. E. and K. S. Bialkowski, "Portable Biomedical Microwave Imaging Using Software - Defined Radio", *2018 Asia-Pacific Microwave Conference (APMC)*, pp. 572–574, Kyoto, Japan, 2018.

16. Sezgen, O. F., O. Altan, A. Bilir, M. G. Durmaz, N. Haciosmanoglu, B. Camli, Z. C. C. Ozdil, A. E. Pusane, A. D. Yalcinkaya, U. O. S. Seker, T. Tugcu and S. Dumanli, "A Multiscale Communications System Based on Engineered Bacteria", *IEEE Communications Magazine*, Vol. 59, No. 5, pp. 62–67, May 2021.

17. Erden, O. K., A. Bilir, C. Karabulut, U. O. Seker and S. Dumanli, "Antennas Reconfigured by Living Cells: AntennAlive", *IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, pp. 882–883, Denver, Colorado, USA, 2022.

18. Bilir, A. and S. Dumanli, "Wide-band Dual Port Cross Slot Wearable Antenna for In-body Communications", *17th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, Florence, Italy, 2023.

19. Mitola, J., "Software Radios-Survey, Critical Evaluation and Future Directions", *NTC-92: National Telesystems Conference*, pp. 13–23, Washington, DC, USA, 1992.

20. Youngblood, G., *A Software-Defined Radio for the Masses*, QEX, Newington, CT, 2002.

21. "1900.1-2019 - IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management", IEEE, Piscataway, NJ, USA, 2019.

22. Carr, J. J., *The Technician's Radio Receiver Handbook: Wireless and Telecommunication Technology*, Newnes, Boston, 2001.

23. Collins, T., R. Getz, D. Pu and A. M. Wyglinski, *Software-Defined Radio for Engineers*, Artech House, Boston, 2018.

24. Mushtaq, S., C. Lohr and A. Gravey, "Network Management and Control Framework for Hybrid Converged Environment", *Cyber Journals: Multidisciplinary Journals in Science and Technology*, Vol. 1, No. 1, pp. 45–55, 2011.

25. Hall, B. and W. Taylor, "X- and Ku-Band Small Form Factor Radio Design", 2017, `https://www.analog.com/media/en/technical-documentation/tech-articles/X-and-Ku-Band-Small-Form-Factor-Radio-Design.pdf`, accessed on December 20, 2022.

26. Abidi, A., "Low-Power Radio-Frequency ICs for Portable Communications", *Proceedings of the IEEE*, Vol. 83, No. 4, pp. 544–569, 1995.

27. Giannini, V., J. Craninckx and A. Baschirotto, *Baseband Analog Circuits for Software Defined Radio*, Springer, Dordrecht, 2008.

28. Analog Devices Inc., "Mixers and Modulators", 2009,

https://www.analog.com/media/en/training-seminars/tutorials/mt-080.pdf, accessed on December 26, 2022.

29. Razavi, B., *RF Microelectronics*, Prentice Hall Press, USA, 2011.

30. Proakis, J. G. and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, NJ: Prentice Hall, Upper Saddle River, 1996.

31. Fernández, J., "Software Defined Radio: Basic Principles and Applications", *Revista Facultad de Ingenier'ia*, Vol. 24, No. 38, pp. 79–96, 2015.

32. Baldini, G., T. Sturman, A. R. Biswas, R. Leschhorn, G. Godor and M. Street, "Security Aspects in Software Defined Radio and Cognitive Radio Networks: A Survey and A Way Ahead", *IEEE Communications Surveys, Tutorials*, Vol. 14, No. 2, pp. 355–379, 2012.

33. Seo, J., Y.-H. Chen and D. S. D. Lorenzo, "A Real-Time Capable Software-Defined Receiver Using GPU for Adaptive Anti-Jam GPS Sensors", *Sensors Journal*, Vol. 11, No. 9, pp. 8966–8991, 2011.

34. Haziza, N. and M. Berbineau, *Multi-technology Vehicular Cooperative System Based on Software Defined Radio (SDR) in Communication Technologies for Vehicles*, Springer, Berlin, Heidelberg, 2013.

35. Yoo, H. S., B. Park and S. H. Kim, "Seamless Vertical Handover in Software Defined Radio Terminal", *International Journal of Control and Automation*, Vol. 2, No. 3, pp. 43–48, 2009.

36. Asadi, A., V. Mancuso and R. Gupta, "DORE: An Experimental Framework to Enable Outband D2D Relay in Cellular Networks", *IEEE/ACM Transactions on Networking*, Vol. 25, No. 5, pp. 2930–2943, 2017.

37. Islam, M., M. A. Hannan, S. A. Samad and A. Hussain, "Software Defined Radio for RFID Application", *Proceedings of the World Congress on Engineering and Computer Science 2009*, Vol. 1, pp. 6–8, San Francisco, USA, 2009.

38. Maguire, J., "Cognitive Radio: Making Software Radios More Personal", *IEEE Personal Communications*, Vol. 6, No. 4, pp. 13–18, 1999.

39. Minden, J. and L. Searl, "Cognitive Radios for Dynamic Spectrum Access: An Agile Radio for Wireless Innovation", *IEEE Communications Magazine*, Vol. 45, No. 5, pp. 113–121, 2007.

40. Mandke, K., S. H. Choi, G. Kim, R. Grant, R.Daniels, W. Kim, R. W. Heath and S. M. Nettles, "Early Results on Hydra: A Flexible MAC/PHY Multihop Testbed", *Proceedings of the 65th IEEE Vehicular Technology Conference*, pp. 1896–1900, Dublin, Ireland, 2007.

41. Matlis, E., T. Corket and S. Goginenis, "Plasma Anemometer for Hypersonic Mach Number Experiments", *ICIASF 2005 Record International Congress on Instrumentation in Aerospace Simulation Facilities*, pp. 245–256, Sendai, Japan, 2005.

42. Bradford, G., *A Framework for Implementation and Evaluation of Cooperative Diversity in Software-Defined Radio*, Master's Thesis, University of Notre Dame, 2008.

43. Gupta, N., V. Renugopalakrishnan, D. Liepmann, R. Paulmurugan and B. Malhotra, "Cell-Based Biosensors: Recent Trends, Challenges and Future Perspectives", *Biosensors and Bioelectronics*, Vol. 141, No. 1, p. 111435, 2019.

44. Analog Devices Inc., "RF Agile Transceiver", 2016, `https://www.analog.com/media/en/technical-documentation/data-sheets/AD9363.pdf`, accessed on January 07, 2023.

45. Analog Devices Inc., "ADALM-PLUTO SDR Active Learning Module", 2017, `https://www.analog.com/media/en/news-marketing-collateral/product-highlight/adalm-pluto-product-highlight.pdf`, accessed on January 07, 2023.

46. "AD9361 Reference Manual", Analog Devices, Norwood, Massachusetts, 2015.

47. Lyons, R., "A Quadrature Signals Tutorial: Complex, But Not Complicated", 2013, `https://www.dsprelated.com/showarticle/192.php`, accessed on June 5, 2023.

48. Dunne, B., "The What, How and Why of Complex Sampling for SDR Transceivers", *ASEE North Central Section Conference*, Michigan, 2019.

49. Analog Devices Inc., "Linux Industrial I/O Subsystem", 2020, `https://wiki.analog.com/software/linux/docs/iio/iio`, accessed on February 3, 2023.

50. Analog Devices Inc., "Hands-on Workshop: Introduction to ADALM-PLUTO and IIO", 2019, `https://wiki.analog.com/software/linux/docs/iio/iio`, accessed on February 3, 2023.

51. Analog Devices Inc., "About Libiio", 2021, `https://wiki.analog.com/resources/tools-software/linux-software/lib iio_internals`, accessed on February 3, 2023.

52. Analog Devices Inc., "Libiio", 2014, `https://wiki.analog.com/resources/eval/user-guides/ad-fmcdaq2 ebz/software/linux/applications/libiio`, accessed on February 3, 2023.

53. GNU Radio, "What is GNU Radio?", 2020, `https://wiki.gnuradio.org/index.php/What_is_GNU_Radio%3F`, accessed on February 5, 2023.

54. GNU Radio, "Vector Source", 2022, `https://wiki.gnuradio.org/index.php/Vector_Source`, accessed on February 10, 2023.

55. GNU Radio, "PlutoSDR Sink", 2021, `https://wiki.gnuradio.org/index.php?title=PlutoSDR_Sink`, accessed on February 15, 2023.

56. GNU Radio, "PlutoSDR Source", 2022, `https://wiki.gnuradio.org/index.php?title=PlutoSDR_Source`, accessed on February 15, 2023.

57. Mashhour, A., W. Domino and N. Beamish, "On the Direct Conversion Receiver — A Tutorial", *Microwave Journal*, Vol. 44, No. 6, p. 114–128, 2001.

58. Svitek, R. and S. Raman, "DC Offsets in Direct Conversion Receivers: Characterization and Implications", *IEEE Microwave Magazine*, Vol. 6, No. 3, pp. 76–86, 2005.

59. MathWorks, "Quadrature Tracking", 2020, https://www.mathworks.com/help/supportpkg/plutoradio/ug/quadrature-tracking.html, accessed on February 23, 2023.

60. Analog Devices Inc., "Dealing with Non-Quadrature Signals", 2019, https://wiki.analog.com/university/tools/pluto/users/non_quad#receiving_a_real_signal, accessed on February 23, 2023.

61. Analog Devices Inc., "FMComms Math", 2015, https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms1-ebz/math, accessed on February 25, 2023.

62. Banerjee, D., *PLL Performance, Simulation and Design*, Dog Ear Publishing, Indianapolis, 2006.

63. Krishna, M., M. Do, K. S. Yeo, C. Boon and W. Lim, "Design and Analysis of Ultra Low Power True Single Phase Clock CMOS 2/3 Prescaler", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 57, No. 12, pp. 72–82, 2010.

64. Taylor, W., "Developing Multiple-Input Multiple-Output (MIMO) Systems with the AD9361", 2019, https://www.analog.com/en/education/education-library/webcasts/developing-multiple-input-multiple-output.html, accessed on March 5, 2023.

65. GNU Radio, "Signal Data Types", 2022, https://wiki.gnuradio.org/index.php?title=Signal_Data_Types#: :text=The%20most%20common%20data%20types,char)%20data%20type%20in%20purple., accessed on March 5, 2023.

66. Analog Devices Inc., "AXI_AD936", 2022, https://wiki.analog.com/resources/fpga/docs/axi_ad9361#internal_interface_description, accessed on February 25, 2023.

67. GNU Radio, "Creating Your First Block", 2022, https://wiki.gnuradio.org/index.php?title=Creating_Your_First_Block, accessed on March 25, 2023.

68. Bilir, A., “Antennas Reconfigured by Living Cells”, B.Sc. Thesis, Bogazici University, 2023.

# APPENDIX A: PERMISSIONS FOR VISUAL MATERIALS

- Richard Lyons, the author of the article titled "A Quadrature Signals Tutorial: Complex, But Not Complicated", has granted permission for the inclusion of Figures 3.11, 3.12, 3.13, and 3.14 from the aforementioned article in the thesis. In his correspondence via mail, he stated, "*Yes, you have permission to use my figures.*"
- Analog Devices, the company that possesses the copyright for the datasheets and articles containing Figures 3.1, 3.2, 3.3, 3.4, 3.15, 3.16, 3.17, 4.14, and 4.15, has granted permission for the incorporation of these materials within this thesis. An engineer from the pertinent department has confirmed this authorization, stating, "*The figures and materials you are using are fine. This looks like fair use in my opinion. There is no need to modify anything.*"
- Sema Dumanli Oktar; an author of the scientific publications from which figures have been taken, the principal investigator of the research group that released these publications, and the supervisor of this thesis, has approved the inclusion of Figures 2.8, 2.9, 2.10, and 2.11 in this thesis.
- Ahmet Bilir, the author of the thesis titled "Antennas Reconfigured by Living Cells", has granted permission for the inclusion of Figures 4.19, 4.20, 4.21, 4.22, and 4.23 in this thesis.