



MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
DE SANTA CATARINA-CÂMPUS ITAJAÍ

RENATO AUGUSTO SCHENKEL MENEGHIN MARCHIORI

**BANCADA DE ENSAIOS PARA SENSORES RESSONANTES DE  
MICRO-ONDAS BASEADA EM RÁDIO DEFINIDO POR  
SOFTWARE**

ITAJAÍ-SC  
2021



RENATO AUGUSTO SCHENKEL MENEGHIN MARCHIORI

**BANCADA DE ENSAIOS PARA SENSORES RESSONANTES DE  
MICRO-ONDAS BASEADA EM RÁDIO DEFINIDO POR  
SOFTWARE**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Dr. Roddy Alexander Romero Antayhua

ITAJAÍ-SC  
2021

Ficha de identificação da obra elaborada pelo autor.

Marchiori, Renato Augusto Schenkel Meneghin Marchiori BANCADA DE ENSAIOS PARA SENSORES RESSONANTES DE MICRO-ONDAS BASEADA EM RÁDIO DEFINIDO POR SOFTWARE /

Renato Augusto Schenkel Meneghin Marchiori Marchiori; orientação de Roddy Alexander Romero Antayhua. - Itajaí, SC, 2021.

122 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Itajaí. Bacharelado em Engenharia Elétrica. Departamento de Eletroeletrônica.

Inclui Referências.

1. Instrumentação Eletrônica. 2. Instrumentação Definida por Software. 3. Radio Definido por Software.

I. Roddy Alexander Romero Antayhua. II. Instituto Federal de Santa Catarina. III. Bancada de ensaios para sensores ressonantes de micro-ondas baseada em rádio definido por software.

# **BANCADA DE ENSAIOS PARA SENSORES RESSONANTES DE MICRO-ONDAS BASEADA EM RÁDIO DEFINIDO POR SOFTWARE**

**RENATO AUGUSTO SCHENKEL MENEGHIN MARCHIORI**

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletricista e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Itajaí, 15 de Outubro de 2021

BANCA EXAMINADORA:

---

**Dr. Wilson Valente Jr.**

Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina - Campus  
Itajaí

---

**Dr. Heron Eduardo de Lima Avila**

Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina - Campus  
Chapecó

---

**Dr. Roddy Alexander Romero Antayhua**

Orientador  
Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina - Campus  
Itajaí

ITAJAÍ-SC  
2021

## **AGRADECIMENTOS**

Agradeço à minha família, pelo apoio dado, ao professor orientador Roddy Alexander Romero Antayhua, pelo direcionamento e fornecimento, junto com o IFSC, dos equipamentos necessários para a execução do trabalho.

## RESUMO

O trabalho trata do desenvolvimento de uma bancada de ensaios baseada em uma placa de desenvolvimento de rádio definido por software (SDR), para a caracterização de sensores de transmissão ressonantes, que operam na faixa de micro-ondas, especificamente até 6 GHz. A bancada utiliza um computador que comanda uma placa de SDR, a qual transmite e recebe o sinal de radiofrequência, interpreta os sinais obtidos da placa e caracteriza o sensor. As rotinas de medição implementadas nessa plataforma de teste envolvem o registro de *offset* do sistema, utilizando conectores externos na configuração física da bancada, a determinação da resposta em frequência, a estimação da frequência de ressonância, o fator de qualidade e o rastreamento da ressonância, com propósito de realizar testes dinâmicos da variação da frequência de ressonância. Para verificar sua utilidade, foram fabricados dois sensores ressonantes planares para extrair suas características em três situações, sendo elas com o sensor descoberto, com metade da área coberta e totalmente coberto com água destilada. Os resultados obtidos foram contrastados com os medidos a partir de um analisador vetorial. Após caracterizados, os sensores foram submetidos a uma rotina de rastreamento da frequência de ressonância, determinando assim em tempo real se algum material foi inserido ou removido da superfície do sensor.

**Palavras-chave:** Instrumentação Eletrônica, Instrumentação Definida por Software, Rádio Definido por Software.

## ABSTRACT

This work presents the development of a Software Defined Radio (SDR) based test bench for characterization of transmission resonant sensors that operate in microwaves, specifically up to 6 GHz. The test bench uses a computer to command the SDR board, which transmits and receives the RF signal, interprets the signals received from the board and characterizes the sensor. The measurement routines implemented on this test platform were the offset registration, with external connectors used in the physical setting, the obtainment of the frequency response, the estimation of the resonance frequency, the quality factor and a frequency tracking, in order to realize dynamic tests with the variation of the resonance frequency. To verify its utility, two planar sensors have been produced to extract their characteristics under three circumstances, being them having the sensor uncovered, half covered and fully covered by distilled water. All results were compared with those measured with a vector network analyzer. After the characterization, the sensors were submitted to the frequency tracking routine, determining if any material was inserted or removed from the surface of the sensor in real time.

**Keywords:** Software Defined Instrumentation. Electronic Instrumentation. Software Defined Radio.

## LISTA DE FIGURAS

Figura 1 – Exemplos de estruturas de sensores de micro-ondas. As figuras (a) e (b) mostram a configuração básica de um sensor de transmissão e reflexão em uma tubulação, respectivamente. Em (c) é mostrada uma cavidade ressonante de micro-ondas cilíndrica com uma amostra do material a ser medido e em (d) mostra o exemplo de um acoplamento ressonador dielétrico cilíndrico a uma micro-fita . . . . .	7
Figura 2 – Linha de transmissão de micro-fita: (a) Geometria. (b) Linhas de campo elétrico e magnético . . . . .	8
Figura 3 – Vista superior do layout de um ressonador de micro-fita em anel com duas portas. . . . .	9
Figura 4 – Exemplo de resultado de medição da magnitude do coeficiente transmissão ( $ S_{21} $ ) em dois ressonadores em formato de anel, dispostos como na figura 3. . . . .	10
Figura 5 – Modelo de elemento concentrado de um pedaço infinitesimal de uma linha de transmissão. . . . .	10
Figura 6 – Exemplo de um equipamento da National Instruments que utiliza o padrão PXI . . . . .	13
Figura 7 – Diagrama do fluxo de um sistema de comunicação . . . . .	14
Figura 8 – Diagrama de uma estrutura ZIF de transmissão e recepção. . . . .	15
Figura 9 – Placa da plataforma SDR USRP B210 (a) e seu encapsulamento (b). . . . .	17
Figura 10 – Diagrama de blocos do CI transceptor AD9361 . . . . .	18
Figura 11 – Tela inicial do Ansys Electronics Desktop, com destaque aos tipos de software de simulação. . . . .	18
Figura 12 – Figura mostrando o processo matemática utilizado pelo HFSS. . . . .	19
Figura 13 – Janela do Ansys HFSS, com destaque a árvore de objetos e aos ícones de desenho. . . . .	19
Figura 14 – Figura mostrando o procedimento de uma simulação no HFSS. . . . .	20
Figura 15 – Interface do GRC versão 3.7. . . . .	21
Figura 16 – Biblioteca no GNU Radio 3.7, com as abas de geradores de forma de onda, UHD e operações matemáticas expandidas. . . . .	22
Figura 17 – Diagrama de exemplo dos tutoriais guiados, mostrando as conexões entre os blocos. Ao executá-lo, ele imprime a função cosseno no domínio do tempo. . . . .	22
Figura 18 – Blocos de comunicação com o USRP. Os blocos das figuras (a) e (b) envia e recebe o sinal para a placa USRP, respectivamente. . . . .	23

Figura 19 – Blocos utilizados para controlar variáveis e criar funções. (a) mostra o receptor de sinal, (b) transforma o sinal de recebido por (a) em uma variável e (c) pode ter a sua funcionalidade modificada ao escrever em um editor de python. . . . .	23
Figura 20 – Blocos de multiplicação. Em (a) se encontra o bloco de multiplicação por um parâmetro pré-determinado. (b) mostra o bloco de multiplicação conjugada entre dois sinais. . . . .	24
Figura 21 – Diagrama da bancada de ensaios com figuras dos componentes . . . . .	25
Figura 22 – Diagrama da bancada de ensaios. . . . .	26
Figura 23 – Diagrama para o processo de registro de <i>offset</i> . . . . .	27
Figura 24 – Algoritmo para a rotina de registro de <i>offset</i> . . . . .	28
Figura 25 – Algoritmo da varredura ao redor da ressonância. . . . .	29
Figura 26 – Algoritmo para a rotina de medição da frequência de ressonância pela fase. . . . .	30
Figura 27 – Algoritmo para a rotina de medição da largura de banda. . . . .	32
Figura 28 – Diagrama de fluxo para rastreamento da frequência de ressonância. . . . .	33
Figura 29 – Bloco de programação customizada com os códigos em Python referentes as rotinas implementadas. (a) mostra o bloco utilizado para a rotina de registro de diferença de conectores. (b) mostra o bloco utilizado para a detecção de ressonância, a medição da largura de banda e o rastreamento da ressonância. . . . .	33
Figura 30 – Diagrama de blocos do GNU Radio para simular o algoritmo de registro de <i>offset</i> do sistema. . . . .	34
Figura 31 – Diagrama de blocos do GNU Radio para simular os algoritmos de detecção de frequência de ressonância, caracterização e rastreamento de frequência de ressonância. . . . .	34
Figura 32 – Placas produzidas, sendo a menor para 1,299 GHz e a maior para 742 MHz, segundo cálculos teóricos. Fabricadas por Marcelo Jefferson Barbosa, técnico em eletroeletrônica pelo IFSC. . . . .	36
Figura 33 – Sensores modelados no ANSYS Electronic Desktop. . . . .	38
Figura 34 – Posições perpendicular (a) e paralela (b) à LT para inserir o recipiente na superfície do sensor. . . . .	38
Figura 35 – Exemplo do efeito da inclinação da fase sob um filtro passa faixa ideal proporcionado pela LT. . . . .	40
Figura 36 – Definindo a prioridade de execução do programa em tempo real, para o Windows. . . . .	41
Figura 37 – Diagrama de blocos para o processo de registro de diferença de conectores implementado no GNURadio. . . . .	42

Figura 38 – Diagrama de blocos para o processo de caracterização do sensor e de rastreamento da frequência de ressonância implementado no GNU Radio. . . . .	42
Figura 39 – Montagem para a Registro de <i>offset</i> do sistema do SDR . . . . .	43
Figura 40 – Configurações descoberta (a), parcial paralela a LT (b), parcial perpendicular a LT (c) e totalmente coberta (d) para a caracterização do sensor A, respectivamente. . . . .	43
Figura 41 – Configurações descoberta (a), parcial paralela a LT (b), parcial perpendicular a LT (c) e totalmente coberta (d) para a caracterização do sensor B, respectivamente. . . . .	44
Figura 42 – Configuração 1 (a), 2 (b), 3 (c) e 4 (d) para a obtenção das curvas do sensor A utilizando o VNA, respectivamente. . . . .	45
Figura 43 – Configuração 1 (a), 2 (b), 3 (c) e 4 (d) para a obtenção das curvas do sensor B utilizando o VNA, respectivamente. . . . .	46
Figura 44 – Resposta em frequência de módulo, fase (do parâmetro $S_{21}$ ) com e sem a compensação da inclinação da fase do sensor A no cenário descoberto obtidas a partir de simulação eletromagnética. . . . .	47
Figura 45 – Resposta em frequência de módulo e fase (do parâmetro $S_{21}$ ) do sensor B e no cenário descoberto obtidas a partir de simulação eletromagnética. . . . .	48
Figura 46 – Coeficiente de reflexão ( $S_{11}$ ) nas cartas de Smith dos sensores A (a) e B (b) obtidas a partir de simulação eletromagnética. . . . .	48
Figura 47 – Resposta em frequência de módulo e fase (do parâmetro $S_{21}$ ) do sensor de A na configuração 1, 2, 3 e 4 extraído a partir do VNA. . . . .	49
Figura 48 – Resposta em frequência de módulo e fase (do parâmetro $S_{21}$ ) do sensor de B na configuração 1, 2, 3 e 4. . . . .	50
Figura 49 – Cartas de Smith dos sensor A (a) e do sensor B (b) pelo VNA. . . . .	53
Figura 50 – Figura gerada pela rotina de registro de <i>offset</i> do GNU Radio. . . . .	53
Figura 51 – Resultado de uma medição com e sem o ajuste de <i>offset</i> e os algoritmos de compensação para o sensor A na configuração descoberta. . . . .	54
Figura 52 – Resultado de uma medição com e sem o ajuste de <i>offset</i> e os algoritmos de compensação para o sensor B na configuração descoberta. . . . .	55
Figura 53 – Captura de tela mostrando o início da rotina de caracterização do sensor. . . . .	56
Figura 54 – Captura de tela mostrando o início da rotina de caracterização do sensor, com o resultado na janela de texto situado no canto inferior esquerdo. . . . .	56
Figura 55 – Resposta em frequência de módulo e fase do sensor de A na configuração 1, 2, 3 e 4 extraído a partir do GNU Radio. . . . .	57
Figura 56 – Resposta em frequência de módulo e fase do sensor B na configuração 1, 2, 3 e 4 extraído a partir do GNU Radio. . . . .	58
Figura 57 – Sobreposição da resposta em frequência obtido pelo VNA e pelo SDR para o sensor A nos 4 cenários. . . . .	61

Figura 58 – Sobreposição da resposta em frequência obtido pelo VNA e pelo SDR para o sensor B nos 4 cenários. . . . .	62
Figura 59 – Finalização da busca pela nova ressonância a partir da rotina de rastreo da ressonância, tendo encontrado a resposta com o sensor coberto. . . . .	63
Figura 60 – Diagrama do misturador de transmissão . . . . .	73
Figura 61 – Diagrama do misturador de recepção. . . . .	78
Figura 62 – Início da rotina de registro de <i>offset</i> simulada. . . . .	101
Figura 63 – Final da rotina de registro de <i>offset</i> simulada. . . . .	102
Figura 64 – Arquivo gráfico obtido a partir da simulação da rotina de registro de <i>offset</i> . . . . .	102
Figura 65 – Início da rotina de encontrar a ressonância. . . . .	103
Figura 66 – Final da rotina de encontrar a ressonância e em espera da alteração da frequência de ressonância. . . . .	104
Figura 67 – Alteração da frequência de ressonância detectada pelo algoritmo em simulação. . . . .	104
Figura 68 – O algoritmo simulado finaliza a busca pela nova frequência de ressonância.	105
Figura 69 – Início da rotina de rastreo da ressonância. . . . .	106
Figura 70 – Fase intermediária da rotina de rastreo da ressonância, tendo encontrado a resposta com o sensor descoberto. . . . .	107
Figura 71 – Fase intermediária da rotina de rastreo da ressonância, detectado uma alteração na frequência de ressonância . . . . .	107

## LISTA DE TABELAS

Tabela 1 – Dados dos sensores fabricados. . . . .	35
Tabela 2 – Configurações de simulação no ANSYS para as micro-fitas e o material dielétrico da placa. . . . .	37
Tabela 3 – Configurações de solução no ANSYS. . . . .	37
Tabela 4 – Configurações da varredura no ANSYS. . . . .	37
Tabela 5 – Resultados da frequência de ressonância, fator de qualidade, largura de banda e frequências de meia potência para os sensores A e B na configuração descoberta obtidos através da simulação do ANSYS. . . . .	49
Tabela 6 – Resultados da frequência de ressonância, fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor A obtidos através do VNA e analisados pelo GNU Octave, pela amplitude. . . . .	50
Tabela 7 – Resultados da obtenção fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor A obtidos através do VNA e analisados pelo GNU Octave, pela fase. . . . .	51
Tabela 8 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor B obtidos através do VNA e analisados pelo GNU Octave. . . . .	51
Tabela 9 – Resultados da obtenção fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor B obtidos através do VNA e analisados pelo GNU Octave, pela fase. . . . .	51
Tabela 10 – Comparação entre os resultados do simulados pelo Ansys e medidos pelo VNA do sensor A no cenário descoberto. . . . .	52
Tabela 11 – Comparação entre os resultados do simulados pelo Ansys e medidos pelo VNA do sensor B no cenário descoberto. . . . .	52
Tabela 12 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor A através do GNU Radio e o SDR. . . . .	57
Tabela 13 – Resultados da largura de banda e fator de qualidade do sensor A obtidos através do GNU Radio e o SDR pela leitura de fase. . . . .	58
Tabela 14 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor B através do GNU Radio e o SDR. . . . .	59
Tabela 15 – Resultados da largura de banda e fator de qualidade do sensor B obtidos através do GNU Radio e o SDR pela leitura de fase. . . . .	59
Tabela 16 – Erros entre os resultados medidos pelo VNA e o SDR do sensor A. . . . .	59
Tabela 17 – Erros entre os resultados medidos pelo VNA e o SDR do sensor B. . . . .	60

## SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>1</b>
1.1 JUSTIFICATIVA	4
1.2 OBJETIVOS	5
1.3 OBJETIVOS ESPECÍFICOS	5
<b>2 – REVISÃO TEÓRICA</b>	<b>6</b>
2.1 SENSORES DE MICRO-ONDAS	6
2.2 SENSOR RESSONANTE EM ANEL DE MICRO-FITA	6
2.3 LINHAS DE TRANSMISSÃO	9
2.4 INSTRUMENTAÇÃO DEFINIDA POR SOFTWARE (SDI)	12
2.5 RÁDIO DEFINIDO POR SOFTWARE	13
<b>3 – MATERIAIS E MÉTODOS</b>	<b>16</b>
3.1 MATERIAIS UTILIZADOS	16
3.1.1 Placa de desenvolvimento de SDR USRP B210	16
3.1.2 ANSYS Electronic Desktop	17
3.1.3 GNU Radio	20
3.2 PROCEDIMENTO EXPERIMENTAL	25
3.2.1 Configuração da bancada de ensaios	25
3.2.2 Implementação das rotinas	26
3.2.2.1 Rotina de registro de <i>offset</i>	26
3.2.2.2 Obtenção da frequência de ressonância	28
3.2.2.3 Rotina de determinação da largura de banda e fator de qualidade	31
3.2.2.4 Rotina de rastreamento dinâmico	31
3.2.2.5 Implementação no GNURadio	31
3.2.3 Simulação dos algoritmos	32
3.2.4 Sensor de micro-ondas ressonante de micro-fita em anel	35
3.2.5 Simulação dos sensores	36
3.2.6 Cenários de teste do Sensor	37
3.2.7 Implementação em Bancada	38
3.2.7.1 Compensação da inclinação de fase	39
3.2.7.2 Configurações e parâmetros do GNU Radio em bancada	40
3.2.8 Validação dos resultados Obtidos pela bancada de ensaios	44
<b>4 – RESULTADOS E DISCUSSÕES</b>	<b>47</b>

4.1	SIMULAÇÕES DO ANSYS . . . . .	47
4.2	CARACTERIZAÇÃO DO SENSOR ATRAVÉS DO VNA . . . . .	49
4.2.1	Sensor A . . . . .	49
4.2.2	Sensor B . . . . .	50
4.2.3	Análise dos gráficos das características de amplitude e fase junto com a caracterização do sensor . . . . .	50
4.3	Caracterização do sensor através do GNU Radio . . . . .	52
4.3.1	Rotina de registro de <i>offset</i> . . . . .	53
4.3.2	Medição pelo GNU Radio e o SDR . . . . .	54
4.3.2.1	Sensor A . . . . .	54
4.3.2.2	Sensor B . . . . .	54
4.3.3	Análise dos resultados . . . . .	55
4.4	Comparação dos resultados entre o SDR e o VNA . . . . .	57
4.5	Rastreo de ressonância . . . . .	59
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>64</b>
5.1	Trabalhos Futuros . . . . .	64
	<b>Referências . . . . .</b>	<b>66</b>

## **Apêndices 71**

<b>APÊNDICE A</b>	<b>Cálculo teórico para a obtenção da frequência de ressonância através da diferença de fase . . . . .</b>	<b>72</b>
A.1	GERAÇÃO DO SINAL . . . . .	72
A.2	SINAIS NOS CANAIS . . . . .	74
A.2.1	Canal sem o sensor . . . . .	74
A.2.2	Canal com o adaptador SMA . . . . .	75
A.2.3	Canal com o sensor . . . . .	76
A.3	OBTENDO O VALOR DE FASE . . . . .	77
A.3.1	Desenvolvimento matemático para obter a expressão em fase (PDI) . . . . .	77
A.3.2	Desenvolvimento matemático para obter a expressão em quadratura (PDQ) . . . . .	79
A.3.3	Sinais de banda base recuperados no PC . . . . .	79
A.4	DESENVOLVIMENTO MATEMÁTICO PARA OBTER A EXPRESSÃO COMPLEXA E OBTER A DIFERENÇA DE FASE . . . . .	80
A.4.1	Recuperação de informação da resposta em frequência do sensor . . . . .	81
A.4.1.1	Recuperação de amplitude . . . . .	81

A.4.1.2	Recuperação de fase com canais de comprimentos iguais . . . . .	82
A.4.1.3	Recuperação de fase desvios de fase ou frequência dos osciladores . . . . .	83
A.4.1.4	Recuperação de fase com a rotina de registro de <i>offset</i> . . . . .	83
<b>APÊNDICE B</b>	<b>Códigos . . . . .</b>	<b>85</b>
B.1	Programação para a calibração . . . . .	85
B.2	Código da caracterização e varredura de frequência . . . . .	87
B.3	Rotina de rastreamento de ressonância . . . . .	93
<b>APÊNDICE C</b>	<b>Simulações das rotinas . . . . .</b>	<b>101</b>
C.1	Rotina de registro de <i>offset</i> . . . . .	101
C.2	Deteção e Rastreamento de Ressonância . . . . .	103
<b>APÊNDICE D</b>	<b>Rastreamento de ressonância do sensor A . . . . .</b>	<b>106</b>

## 1 INTRODUÇÃO

Durante a década de 1950, iniciou-se o desenvolvimento de sensores de micro-ondas com o objetivo de “desenvolver melhores métodos de se medir permissividade elétrica e estudar a sua relação com as propriedades físicas dos materiais” (NYFORS, 2000, Tradução do autor).<sup>1</sup> Nesse período, o elevado custo e dimensões dos componentes para radiofrequência, e a baixa capacidade de processamento do sinal, tornaram limitado o alcance da aplicação desses sensores. Posteriormente, nos anos 1970 e 1980, o crescimento dos dispositivos de estado sólido e a introdução dos microprocessadores digitais permitiram o barateamento e melhorias nos equipamentos de micro-ondas (NYFORS, 2000, p. 23).

Os sensores de micro-ondas podem ter formatos e métodos de medição diferentes, como ressonadores planares ou em cavidade, guias de ondas, reflexão, transmissão, radar, entre outros (NYFORS, 2000, p. 23). As principais vantagens dos sensores de micro-ondas são (NYFORS, 2000, p. 23):

- Não precisam de contatos mecânicos com o objeto a ser monitorado;
- As micro-ondas eletromagnéticas podem penetrar todos os materiais com exceção dos metais;
- Conseguem um grande contraste entre água e outras substâncias;
- Os sensores ressonantes de micro-ondas estáveis já que as suas características de ressonância estão relacionadas com as suas dimensões físicas.
- Sem influência da condutividade em corrente contínua ou da polarização, o que ocorre nos sensores de baixa frequência;
- São resilientes a condições ambientais, seguros, de rápida resposta (em comparação aos sensores radioativos [por radiação ionizante]) e;
- Não alteram o material sob teste de qualquer forma (não invasivos).

Como desvantagens dos sensores de micro-ondas, pode-se citar (NYFORS, 2000, p. 23):

- O uso da alta frequência, utilizada nesses, encarece os equipamentos eletrônicos necessários;
- Eles precisam ser calibrados para cada material diferente em que for utilizado para medição;
- Normalmente são adaptados para situações específicas, resultando em uma baixa universalidade e;
- são sensíveis a mais de uma variável, necessitando de compensação por outros sensores.

Apesar das características negativas, as vantagens tornaram os sensores de micro-

---

<sup>1</sup>Para fins de clareza, “micro-ondas” será utilizado para definir ondas eletromagnéticas de comprimento entre 1 mm a 1m, o que equivale a frequências de 300 MHz até 300 GHz no espaço livre.

ondas atrativos para a indústria e são utilizados em diversas aplicações, como aferição de distância, movimento, formato, tamanho de partículas e, principalmente, propriedades dos materiais (NYFORS, 2000, p. 23), como ZHANG (2019) utilizou um sensor de micro-ondas para medir a qualidade da água e obtenção de figuras bidimensionais de metais com formatos irregulares utilizando uma distribuição de sensores por ANKITA e BASU (2013). Para a medicina, o trabalho de MARIMUTHI (BIALKOWSKI; ABBOSH; 2016) traz a geração de imagens biomédicas utilizando um Radar Definido por Software.

A Instrumentação Definida por Software (*Software Defined Instrumentation*, SDI) (FRENZEL, 2007), tem seu início durante a década de 1970 (GROENENDAAL, 2018), sendo promovida como método de teste por empresas como a National Instruments e Keysight. Nos anos 1990, a ideia passou de não só utilizar como aquisição de dados, mas também para automação do processo de teste com o auxílio do padrão PXI (*Peripheral Component Interconnect [PCI] eXtensions for Instrumentation*)(INSTRUMENTS, 2021a). Hoje, se tem diversos equipamentos que utilizam o software em testes e medições, automatizados ou não, tais como nos analisadores de espectro e sinal da Rohde & Schwarz (2021) e o Laboratório Inteligente remoto da Tektronics (2021).

A utilização da SDI não era viável inicialmente, segundo o Groenendaal (2018), para sistemas de radiofrequência pelo tamanho necessário do equipamento para se realizar os testes. Porém, o padrão PXI, hoje, tem um grande impacto e está trazendo uma transformação fundamental em testes de micro-ondas automatizados. Alguns exemplos de aplicação do conceito de SDI são a implementação de uma malha de travamento de fase (*Phase Lock Loop*, PLL) (ZHANG; JIN, 2012), um sistema de posicionamento de um giroscópio com modelagem de erro estocástico (TONG e GAO, 2020), análise no domínio da frequência a partir da densidade espectral do sinal (NAYAK et al., 2017) e detecção de reflexões e mitigação de ruído por software (SCHMIDT et al., 2019). A aviação eletrônica também aplicou os padrões PXI para medições e simulações dos seus componentes, como os radares secundários (SHAW; LOPATO, 2014).

Para a radiofrequência, sistemas definidos por software estão sendo utilizados para testar produtos dos padrões WiMAX, GSM, 5G, dentre outros (GROENENDAAL, 2018). A radiofrequência em testes de produção trouxe grandes transformações para medições automatizadas e os equipamentos digitais passaram a substituir os analógicos para a faixa de micro-ondas. O Rádio Definido por Software (*Software Defined Radio*, SDR) é um sistema de comunicação de rádio onde componentes de tratamento de sinal tradicionalmente implementados em hardware são feitos em software e realiza diversas tarefas complexas simultâneas para permitir uma transmissão e recepção perfeita dos dados, definindo assim o SDR (KEIM, 2017) (COLLINS et al., 2018). Hoje, sistemas baseados em SDR são utilizados principalmente na telefonia sem fio, sendo o 3G e o 4G dois exemplos dessa aplicação (DEVICES, 2017).

Ettus Research (2020), Analog Devices (2021) e National Instruments (2021b)

são exemplos de empresas que desenvolvem placas de desenvolvimento SDR e pode-se utilizar algum software para interface entre o computador e a placa, como o LabVIEW (INSTRUMENTS, 2021b) ou o GNU Radio (RADIO, 2021a), sendo o segundo o mais utilizado por ser gratuito. Pela sua capacidade de trabalhar em altas frequências e gerar sinais compatíveis com micro-ondas, a plataforma de SDR pode também ser utilizada como uma ferramenta de SDI, para realizar testes de sensores de micro-ondas.

Recentemente, o uso de placas de desenvolvimento de SDR tem aumentando em aplicações de SDI. Alguns exemplos são na área da acústica, onde uma plataforma Periférica Universal de Software de Rádio (USRP) B200 com um microcomputador para um sistema interrogatório de sensor de onda acústica de superfície (HUMPHRIES et al., 2015). Na medicina, os trabalhos de detecção de padrão de respiração com auxílio de um algoritmo de árvore de decisão (REHMAN et al., 2021) e um USRP 2922 (INSTRUMENTS, 2021a). Além desses trabalhos de instrumentação, um protótipo de analisador vetorial de rede de 4 canais de imagem de micro-ondas foi implementado com uma placa USRP B210 (MEANEY et al., 2019), medição de campos eletromagnéticos de radiofrequência (KORNPROBST; MAUERMAYER; EIBERT, 2020) e o cancelamento da auto interferência em leitores RFID para melhoria da leitura de *tags* ou sensores sem chip (*chipless*) (ANTAYHUA; RAMBO; DE SOUZA, 2018).

Assim, esse trabalho se propõe a desenvolver uma bancada de ensaios, com o auxílio de uma plataforma de Rádio Definido por Software, para a caracterização de sensores de micro-ondas.

## 1.1 JUSTIFICATIVA

É necessário que sensores sejam conhecidos e testados, caracterizando assim a sua frequência de ressonância, seu fator de qualidade e sua largura de banda entre outros parâmetros. Esse procedimento é importante não somente para a comercialização, mas para verificar se os parâmetros dados pelo projeto estão de acordo com o esperado. Assim, a bancada de ensaios procura tornar o processo de caracterização robusto, confiável e com um investimento relativamente baixo e replicável em outros ambientes de ensino ou pesquisa similares. Além disso, a flexibilidade da placa de SDR permite que sejam realizados testes com diversos tipos de sensores, com características e frequências de operação diferentes. A sua implementação servirá também para dar suporte à caracterização de sensores desenvolvidos no campus no contexto de projetos de pesquisa.

A utilização do Rádio Definido por Software permite a aplicação dos circuitos moduladores e demoduladores, para gerar sinais compatíveis com o teste de sensores operando na faixa de micro-ondas. Esse trabalho utilizará a plataforma de Rádio Definido por Software B210, com duas portas que atuam tanto com receptores como transmissores, e duas portas exclusivas para recepção do sinal de micro-ondas, podendo trabalhar em uma faixa de frequência entre 70 MHz e 6 GHz. Essa plataforma possui conexão USB 3.0, em que os dados, convertidos de forma digital, são enviados para um outro dispositivo, neste caso um computador, para o processamento dos dados, com uma largura de banda de até 56 MHz. No computador, será utilizado o software GNURadio, por ter licença gratuita, permitir programação das tarefas em forma de blocos e também criação de blocos por meio de código de alto nível.

## 1.2 OBJETIVOS

Desenvolver uma mesa de ensaio para a caracterização de sensores ressonantes na faixa de micro-ondas utilizando Radio Definido por Software.

## 1.3 OBJETIVOS ESPECÍFICOS

- Desenvolver a expressão matemática baseada nas possibilidades fornecidas pela arquitetura do Radio Definido por Software, no caso o USRP B210 da Ettus Research, que permita a obtenção da resposta em frequência do sensor em amplitude e fase.
- Montar uma bancada de ensaios para caracterizar sensores ressonantes com o Radio Definido por Software, cabos e conectores de micro-ondas;
- Desenvolver algoritmos para a caracterização do sensor, por meio de uma rotina de registro de *offset*, uma rotina de identificação da frequência de ressonância e do fator de qualidade;
- Comparar os efeitos da rotina de registro de *offset* nos resultados do banco de ensaio;
- Desenvolver uma rotina de para seguir a frequência de ressonância do sensor ressonante quando essa for alterada;
- Estudar uma estrutura ressonante planar, selecionar e implementar essa estrutura para os testes;
- Utilizar a bancada de ensaios para caracterizar a estrutura ressonante implementada;

## 2 REVISÃO TEÓRICA

### 2.1 SENSORES DE MICRO-ONDAS

O princípio de funcionamento dos sensores de micro-ondas se baseiam na interação entre as ondas eletromagnéticas na faixa das altas frequências e o material a ser analisado (NYFORS, 2000). Dependendo do arranjo de medição e do fenômeno com o qual o sensor opera, os sensores de micro-ondas são divididos em grupos. As principais interações podem ser dadas sob as formas de reflexão, refração, espalhamento, emissão, absorção ou mudanças na velocidade e na fase da onda. As aferições por meio dos campos eletromagnéticos são passivas e não destrutivas.

Uma vez que os materiais em análise são não magnéticos na sua maioria, a característica elétrica que determina os parâmetros da onda no meio se dá pela permissividade elétrica relativa do material.

$$\varepsilon_r = \varepsilon'_r - j\varepsilon''_r \quad (1)$$

Onde:

$\varepsilon_r$  é a permissividade relativa do material;

$\varepsilon'_r$  é a parte real permissividade complexa e;

$\varepsilon''_r$  é a parte imaginária da permissividade complexa.

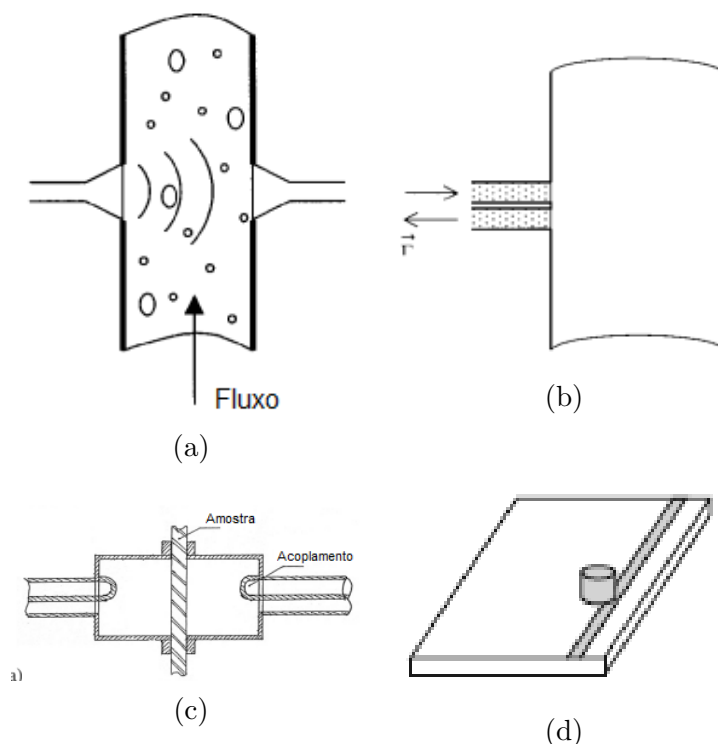
Como são possíveis diversos arranjos de medição, os sensores de micro-ondas possuem características e aplicações diferentes. Os exemplos de disposição são: transmissão por espaço livre, transmissão por guia de onda, reflexão, reflectometria no domínio do tempo, radar, tomografia, ressonadores e radiometria (NYFORS, 2000). Alguns sensores de micro-ondas estão dispostos na figura 1.

Os sensores de micro-ondas possuem as seguintes vantagens: não necessitam contato mecânico com o mensurando, as micro-ondas eletromagnéticas penetram todos os materiais com exceção dos metais, apresentam um excelente contraste entre água e outras substâncias, são estáveis, insensíveis a condições ambientes, sem interferências da operação de corrente contínua, são seguros e rápidos (NYFORS, 2000). Das desvantagens, os equipamentos necessários para a alta frequência são mais caros, precisam ser calibrados separadamente para cada objeto a ser medido e não universalidade (NYFORS, 2000).

### 2.2 SENSOR RESSONANTE EM ANEL DE MICRO-FITA

Um dos métodos aplicados para se medir a permissividade elétrica do meio é a partir de sensores ressonantes de micro-ondas. Entende-se aqui, como ressonância do sensor, a frequência em que os características reativas do sistema, no caso a indutância e a

Figura 1 – Exemplos de estruturas de sensores de micro-ondas. As figuras (a) e (b) mostram a configuração básica de um sensor de transmissão e reflexão em uma tubulação, respectivamente. Em (c) é mostrada uma cavidade ressonante de micro-ondas cilíndrica com uma amostra do material a ser medido e em (d) mostra o exemplo de um acoplamento ressonador dielétrico cilíndrico a uma micro-fita



Fonte: (a) e (b) (NYFORS, 2000); (c) (KRASZEWSKI; NELSON, 2004) e; (d) (POZAR, 2011).

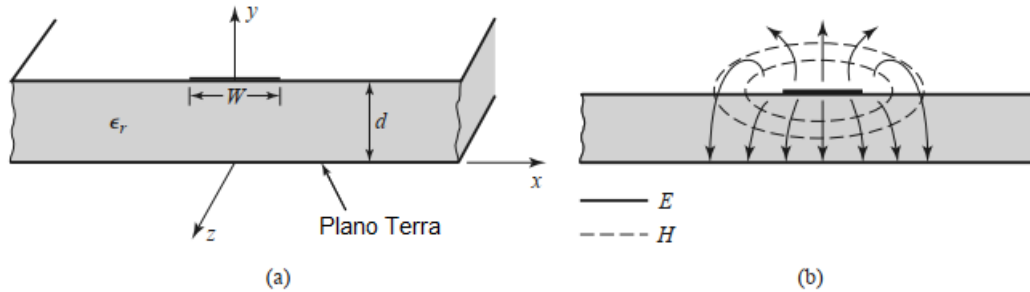
capacitância, se compensam, resultando em uma característica puramente resistiva (ALEXANDER; SADIKU, 2013). Geralmente, o tipo de linha de transmissão (LT) empregado nomeia esses sensores, como cavidades, coaxial, micro-fita, entre outros (NYFORS, 2000). Em circuitos planares, estruturas de micro-fitas em formato de anel, quadrado, linhas com meio comprimento de onda e outros formatos são utilizados como circuitos ressonantes para mensurar a atenuação e a permissividade do meio ao seu redor (BRAY; ROY, 2003).

“A geometria simples dos ressonadores planares, combinada com sua análise relativamente simples, contribuíram para a sua popularidade, principalmente para avaliação comparativa” (BRAY; ROD, 2003; tradução do autor). Além disso, inserir o ressonador em uma placa de circuito impresso permite a utilização de dispositivos de micro-ondas passivos e ativos, incluindo circuitos integrados de radiofrequência (BRAY; ROY, 2003).

Pelo uso da configuração planar, a alimentação e fabricação dos sensores ressonadores geralmente envolve a utilização de uma linha de transmissão, que está conectada e casada aos circuito de aquisição que compõe o sensor, e ela é acoplada eletromagneticamente ao ressonador, sendo a mais comum entre as linhas de transmissão a de micro-fita, mostrada na figura 2. Como nem todo o espaço é coberto pelo material dielétrico, a análise

teórica da linha é relativamente complexa (POZAR, 2011).

Figura 2 – Linha de transmissão de micro-fita: (a) Geometria. (b) Linhas de campo elétrico e magnético



Fonte: Engenharia de micro-ondas (POZAR, 2011)

A ressonância do sensor pode lembrar tanto um filtro passa-faixa ou um filtro rejeita-faixa, sendo que a variação desse valor nos traz a informação da parte real da permissividade complexa (NYFORS, 2000), como mostra a equação (2). Também segundo a resposta ideal de um filtro passa-faixa, as frequências de meia potência possuem uma diferença de fase de  $45^\circ$  com relação a fase na ressonância. Utilizando a equação (3) (ALEXANDER; SADIKU, 2013), tem-se que para obter o fator de qualidade, deve-se obter as frequências de meia potência laterais  $f_1$  e  $f_2$ , além de trazer a relação entre a parte real e imaginária da permissividade complexa (NYFORS, 2000).

$$f_r = \frac{f_{r0}}{\sqrt{\epsilon_r'}} \quad (2)$$

$$Q = \frac{f_c}{BW} = \frac{f_c}{f_2 - f_1} = \frac{\epsilon_r'}{\epsilon_r''} \quad (3)$$

Onde:

$f_r$  é a frequência de ressonância medida do sensor;

$f_{r0}$  é a frequência de ressonância do sensor no espaço livre (sem material);

$\epsilon_r'$  é a parte real permissividade complexa e;

$\epsilon_r''$  é a parte imaginária da permissividade complexa.

$Q$  é o fator de qualidade;

$f_c$  é a frequência central ou de ressonância;

$BW$  é a largura de banda;

$f_2$  é a frequência de meia potência superior e;

$f_1$  é a frequência de meia potência inferior.

Utilizando uma micro-fita em formato de anel para o ressonador, a frequência de ressonância fundamental é dada pela equação (4) (MAZZARO, 2012), onde  $c$ ,  $r$  e  $e_{eff}$  são a velocidade da luz no espaço livre, o raio médio do anel e a constante dielétrica efetiva da estrutura da linha de transmissão (micro-fita). Quando o sensor almejado é do tipo

transmissão, o acoplamento do ressonador em anel à linha de micro-fita pode ser realizado por meio de um vão localizado entre cada extremos da linha de transmissão dividida ao meio. O vão representa um acoplamento capacitivo e pode ser diretamente finalizada com o *gap*, projetada ou não um contorno circular ao redor do anel (MAZZARO et al., 2012) conforme mostrados nas figuras 3(a) e 3(b). Próximo da frequência de ressonância fundamental do anel, o ressonador disposto como na figura 3 possui uma resposta de amplitude de transmissão máxima, seguido de um vale e um outro pico no múltiplo da sua fundamental, e assim por diante, como se mostra na figura 4.

$$f_0 = \frac{c}{2\pi r \sqrt{|\epsilon_{eff}|}} \quad (4)$$

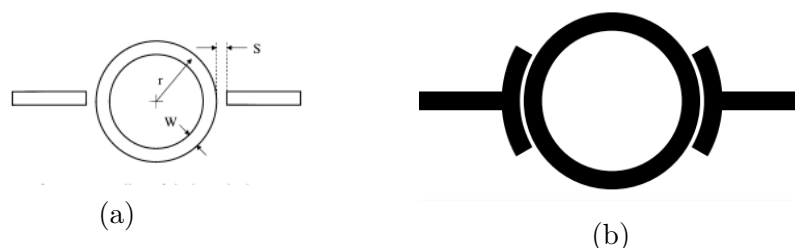
Onde:

$c$  é a velocidade da luz;

$r$  é o raio médio do anel e;

$\epsilon_{eff}$  é a constante dielétrica efetiva da estrutura da linha de transmissão (micro-fita).

Figura 3 – Vista superior do layout de um ressonador de micro-fita em anel com duas portas.



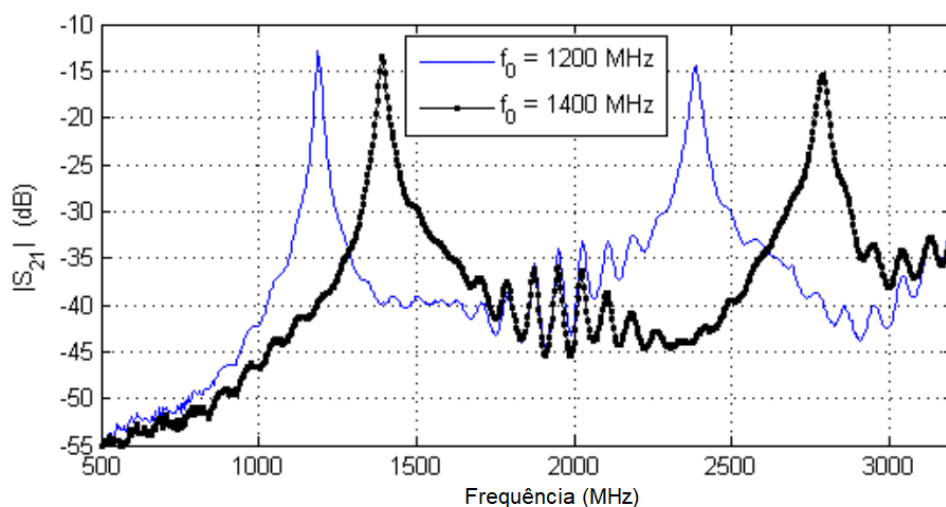
Fonte: Caracterização de micro-ondas de uma linha micro-fita usando um ressonador em anel com um modelo de elementos concentrados melhorado. (BRAY; ROY, 2003)

## 2.3 LINHAS DE TRANSMISSÃO

Para diversos circuitos, o comprimento elétrico dos seus componentes é desprezível em relação ao comprimento de onda e pode-se utilizar os parâmetros de elementos concentrados para uma descrição adequada. Se o comprimento de onda do sinal possui dimensões comparáveis às partes do circuito, “o efeito de atraso no tempo se torna apreciável, o que leva a diferenças de fase introduzidas por atraso” (HAYT; BUCK, 2013). Assim, lida-se com o fenômeno de ondas em linhas de transmissão (LT) de forma similar a propagação de energia de ponto a ponto no espaço livre ou dielétricos (HAYT; BUCK, 2013).

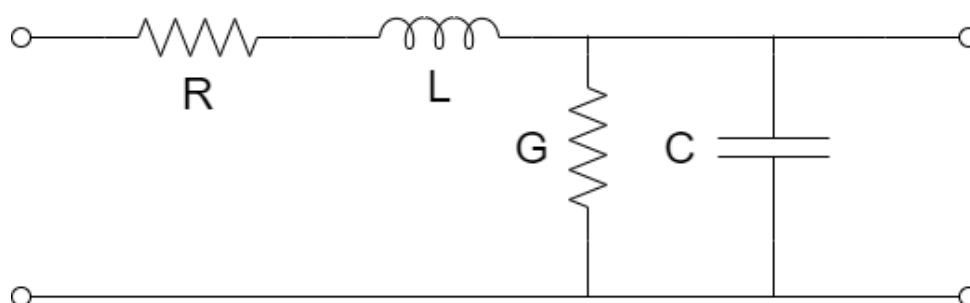
A principal diferença entre a teoria de circuitos e a de linha de transmissão é o comprimento elétrico. Assim, a LT é uma rede de parâmetros distribuídos, onde a magnitude e fase variam sob o seu comprimento. Um pedaço infinitesimal da LT pode ser

Figura 4 – Exemplo de resultado de medição da magnitude do coeficiente transmissão ( $|S_{21}|$ ) em dois ressonadores em formato de anel, dispostos como na figura 3.



Fonte: Medições de permissividade *in-situ* utilizando ressonadores em anel (MAZZARO, 2012)

Figura 5 – Modelo de elemento concentrado de um pedaço infinitesimal de uma linha de transmissão.



Fonte: Eletromagnetismo (HAYT; BUCK, 2013)

modelado como um circuito de elementos concentrados, conforme o da figura 5, com os parâmetros de resistência em série ( $R$ ), indutância em série ( $L$ ), condutância em paralelo ( $G$ ) e capacitância em paralelo ( $C$ ) por unidade de comprimento (HAYT; BUCK, 2013).

Os parâmetros por unidade de comprimento auxiliam no cálculo da constante de propagação complexa e na impedância característica da LT, conforme as equações (5) e (6) (HAYT; BUCK, 2013). Tanto a constante de propagação quanto a impedância característica são dependentes da frequência e são dados quanto a constante de atenuação ( $\alpha$ ) e a constante de fase ( $\beta$ ).

$$\gamma = \alpha + j\beta = \sqrt{(R + j\omega L)(G + j\omega C)} \quad (5)$$

Onde:

- $\gamma$  é a constante de propagação da onda;
- $\alpha$  é a constante de atenuação;

$\beta$  é a constante de fase;  
 $\omega$  é a frequência angular;  
 $R$  é a resistência em série por unidade de comprimento;  
 $L$  é a indutância em série por unidade de comprimento;  
 $G$  é a condutância em paralelo por unidade de comprimento e;  
 $C$  é a capacitância em paralelo por unidade de comprimento.

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \quad (6)$$

Onde:

$Z_0$  é a impedância característica da LT.

A velocidade de fase é definida como a relação entre a frequência angular e a constante de fase, conforme (7) (HAYT; BUCK, 2013).

$$v_p = \frac{\omega}{\beta} \quad (7)$$

Onde:

$v_p$  é a velocidade de fase da LT.

A constante de atenuação e de fase determinam o quanto será a perda de magnitude e o deslocamento de fase do sinal conforme o comprimento da LT, respectivamente. A impedância característica de uma linha é a relação entre a onda de tensão e corrente na LT. Em uma interconexão de LTs, caso a impedância característica das LTs seja diferente, ocorrerá reflexões, gerando uma onda viajando em sentido oposto a incidente e atenuando o sinal.

Em uma LT de micro-fita encontrada em uma placa de circuito impresso, uma parte da região é coberta com o material da placa enquanto outra é pelo ar. Devido a essas características, o cálculo da permissividade eficaz e da impedância característica são obtidos através das equações (8) e (9), respectivamente (POZAR, 2011).

$$\varepsilon_{eff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2} \frac{1}{\sqrt{1 + 12d/W}} \quad (8)$$

$$Z_0 = \begin{cases} \frac{60}{\sqrt{\varepsilon_{eff}}} \ln\left(\frac{8d}{W} + \frac{W}{4d}\right), & W/d \leq 1 \\ \frac{120\pi}{\sqrt{\varepsilon_{eff}}[W/d + 1,393 + 0,667 \ln(W/d + 1,444)]}, & W/d \geq 1 \end{cases} \quad (9)$$

Onde:

$\varepsilon_{eff}$ : permissividade eficaz da micro-fita;

$\varepsilon_r$ : permissividade relativa da placa;

$W$ : a largura da micro-fita;

$d$ : espessura da placa e;

$Z_0$ : impedância característica da linha de transmissão em micro-fita.

Considerando a propagação da onda em uma LT no eixo  $z$ , as equações (10) e (11) mostram a expressão da tensão ao longo da LT no domínio do tempo e da frequência, respectivamente. O coeficiente de reflexão de tensão em uma interconexão de linhas de transmissão é determinado com a equação (12) (HAYT; BUCK, 2013).

$$v(z,t) = |V_o^+| \cos(\omega t - \beta z + \phi^+) e^{-\alpha z} + |V_o^-| \cos(\omega t + \beta z + \phi^-) e^{\alpha z} \quad (10)$$

Onde:

- $v$  é a tensão no domínio do tempo;
- $t$  é o tempo;
- $z$  o comprimento com relação a origem (início da LT);
- $V_o^+$  é a onda incidente;
- $V_o^-$  é a onda refletida;
- $\phi^+$  é o ângulo da onda incidente e;
- $\phi^-$  é o ângulo da onda refletida.

$$V(z) = V_o^+ e^{-\gamma z} + V_o^- e^{\gamma z} \quad (11)$$

Onde:

- $V$  é a tensão no domínio da frequência;

$$\Gamma = \frac{V_o^-}{V_o^+} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (12)$$

Onde:

- $\Gamma$  é o coeficiente de reflexão;
- $Z_1$  é a impedância característica da primeira LT e;
- $Z_2$  é a impedância característica da segunda LT e;

## 2.4 INSTRUMENTAÇÃO DEFINIDA POR SOFTWARE (SDI)

A Instrumentação Definida por Software (*Software Defined Instrumentation*, SDI) é a utilização de dispositivos de hardware independentes, configurados por software, para realizar tarefas de medições customizadas (COOMBS, 2000). Essa abordagem procura uma forma mais genérica de se produzir os sistemas de medição, possuindo portas de entrada e saída e o processamento de dados realizados de forma digital, ao invés do processamento analógico, que possui um circuito específico para um tipo de medição.

Inicialmente, os sistemas utilizando a noção de SDI foram definidos pela primeira vez como instrumentos virtuais por Coombs (2000), sendo destacado pela habilidade de realizar funções de instrumentação personalizadas e de ser reconfigurável para realizar outro tipo de medição. Durante a década de 1970 (GROENENDAAL, 2018), o conceito

Figura 6 – Exemplo de um equipamento da National Instruments que utiliza o padrão PXI



Fonte: *PXI Plataforma Power and Cooling* (INSTRUMENTS, 2021)

foi promovido principalmente por empresas, como a National Instruments e a Keysight, e os principais produtos eram voltados para tarefas genéricas de aquisição de dados.

Nos anos 1990, o padrão PXI emergiu e levantou o foco também para os processos de medição automatizados para a SDI (GROENENDAAL, 2018). Além disso, a SDI foi além da simples aquisição de dados e expandiu para diversos equipamentos eletrônicos que utilizam o software em testes e medições, sejam automatizados ou não, como os multímetros digitais, analisadores de espectro, digitalizadores, osciloscópios, dentre outros. Exemplos de SDI em equipamentos incluem analisadores de espectro e sinal da Rohde & Schwarz (2021) e o Laboratório Inteligente remoto da Tektronics (2021).

Devido a possibilidade de utilização de funções personalizadas do usuário na instrumentação, com o uso de diversos módulos (INSTRUMENTS, 2021a), faz com que engenheiros possam fazer medições e configurar os aparelhos para a aplicação específica (FRENZEL, 2007). A utilização do padrão PXI (ELECTRONIC-NOTES, 2021) para hardware e software traz uma alta confiabilidade para padrões industriais.

## 2.5 RÁDIO DEFINIDO POR SOFTWARE

O Rádio Definido por Software (SDR) é um sistema de comunicação por radio-frequência onde componentes de tratamento de sinal tradicionalmente implementados em hardware são feitos em software, realizando as requeridas tarefas de processamento de sinal em um dispositivo de processamento digital de sinal (*Digital Signal Processing*, DSP) programável (KEIM, 2017) (HARRIS; LOWDERMILK, 2010). A plataforma dispõe de conversores analógicos-digitais (AD) e digitais-analógicos (DA) para fazer a interface com os circuitos de micro-ondas (HARRIS; LOWDERMILK, 2010).

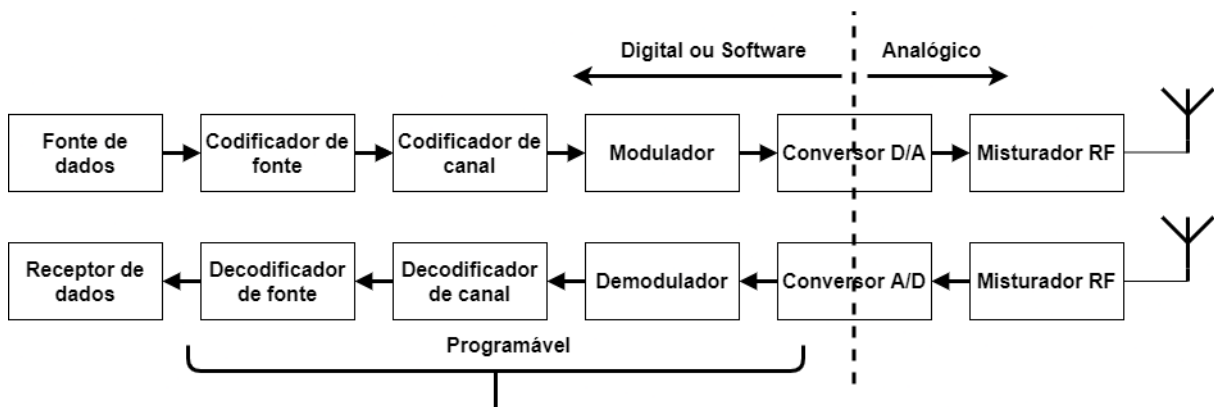
O DSP nos sistemas de rádio tem sido utilizado para executar as funções tradicionalmente executadas em hardware, tais como: circuitos misturadores, filtros, moduladores, entre outros. Tem-se como principal vantagem o baixo custo aliado ao bom desempenho

(HARRIS; LOWDERMILK, 2010). Circuitos digitais conseguem acompanhar melhor a lei de Moore do que os analógicos. Com isso, os rádios passaram a utilizar DSP para utilizar codificação de linha, trazendo maior imunidade ao ruído, menor potência do sinal e facilidade de detecção de erro (LATHI; Z., 2012).

Os sistemas de comunicação digitais são feitos pela combinação de sinais analógicos de RF com DSP (COLLINS et al., 2018). Entretanto, a utilização de DSP não indica que um rádio seja definido por software, já que as suas funções são executadas de forma específica. Um SDR é reconfigurável, trazendo outros parâmetros do rádio como frequência de transmissão e recepção, código de linha, dentre outros (HARRIS; LOWDERMILK, 2010).

Assim, um sistema de SDR é um dispositivo complexo que realiza diversas operações simultaneamente para transmitir e receber de dados (COLLINS et al., 2018), sendo possível a reconfiguração dos seus parâmetros. Na figura 7, observa-se o diagrama que mostra a estrutura básica de um transmissor e um receptor de um sistema de comunicação. Os blocos de codificação e decodificação geralmente são rotinas de PDS, já os moduladores e demoduladores são usualmente feitos em hardware cujos parâmetros são especificados em software. Uma plataforma de SDR tem a capacidade de modificar tanto as rotinas de PDS quanto os parâmetros dos moduladores e demoduladores por meio de programação.

Figura 7 – Diagrama do fluxo de um sistema de comunicação

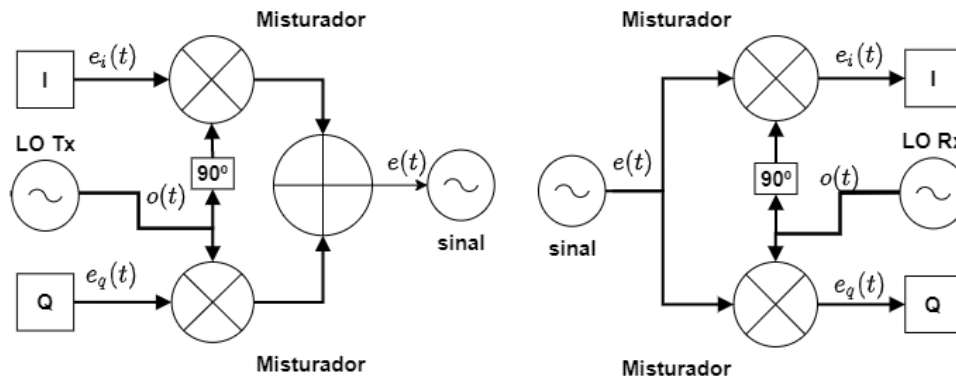


Fonte: O autor

Para a estrutura de transmissão e de recepção em um SDR, é comum encontrar a estrutura de frequência intermediária zero (*Zero Intermediate Frequency, ZIF*) (COLLINS et al., 2018). Um diagrama da estrutura ZIF de transmissão e de recepção é mostrada na figura 8. Os sinais de banda base I e Q possuem uma defasagem de  $90^\circ$ , possibilitando o tratamento dos dados em fase e quadratura.

O Processamento Digital de Sinais (*Digital Signal Processing, DSP*) foi fortemente influenciado pelo avanço da microeletrônica. A comunicação digital teve emprego amplo de microprocessadores nos sistemas transceptores, produzindo plataformas de radio mais versáteis e com maior processamento de sinal. Dentre os diferentes tipos de arquitetura

Figura 8 – Diagrama de uma estrutura ZIF de transmissão e recepção.



Fonte: (FRIZELLE; KEARNEY, 2017)

de microprocessadores utilizadas em SDRs, incluem-se: microprocessadores de uso geral; Arranjos de portas de campo programáveis (*Field-Programmable Gate Arrays*, FPGA); Unidades de Processamento Gráfico (*Graphic Processing Unit*, GPU); e *Acorn RISC Machine* (ARM).

Os ambientes de software que utilizam o SDR servem como base para programar as funções de modulação, demodulação, filtragem, entre outros, que serão executadas na plataforma e no computador. Dois exemplos de programas para configurar essas funções são o GNURadio (RADIO, 2021a) e o LabVIEW (INSTRUMENTS, 2021b).

### 3 MATERIAIS E MÉTODOS

#### 3.1 MATERIAIS UTILIZADOS

Os materiais utilizados para a realização do presente trabalho foram:

- 1 computador;
- 1 placa de desenvolvimento de SDR modelo USRP B210 da Ettus Research (RESEARCH, 2020);
- 1 analisador de rede vetorial (*vector network analyser*, VNA) PocketVNA;
- 3 cabos coaxiais RG-58 com conector SMA macho em ambos extremos, sendo 2 de 1,1 metro e 1 de 23 centímetros;
- 1 adaptador reto SMA fêmea para SMA fêmea;
- 2 ressonadores de micro-fita em anel;
- 2 cabos USB;
- 500 ml de água destilada e;
- 2 recipientes de plásticos polipropileno, sendo um com dimensões de 5,5x9x4 cm e outro de 8,5x13x5 cm.

Quanto aos requisitos do computador, o processador possui uma velocidade de 2,2 GHz, 4 GB de memória RAM, sistema operacional Windows 10 e aplicativo GNU Radio versão 3.7. Para o VNA, utilizou-se o próprio *software* do fabricante, PocketVNA versão 1.54 (POCKETVNA, 2021b). Para a simulação dos ressonadores, foi utilizado o ANSYS Eletronic Desktop. A geração das imagens de resposta em frequência e tratamento dos dados extraídos a partir do Pocket VNA e do SDR foram realizados através do GNU Octave.

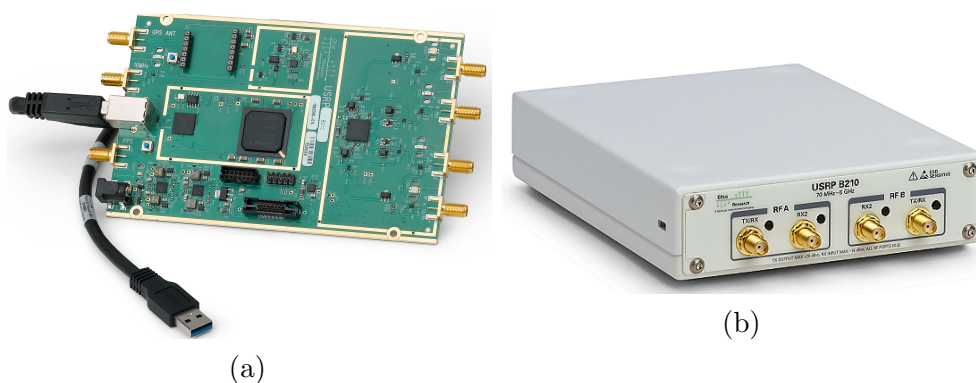
A água destilada foi fornecida por empresa produtora de medicamentos e substâncias para farmácia e hospitais.

##### 3.1.1 Placa de desenvolvimento de SDR USRP B210

A *Universal Software Radio Peripheral* (USRP) B210 é uma plataforma de SDR produzida pela Ettus Research (RESEARCH, 2020). É uma das placas disponíveis no IFSC - Campus Itajaí para realização de protótipos de baixo custo em radiofrequências. A placa e o encapsulamento da plataforma B210 são mostrados na figura 9.

A placa dispõe de um Circuito Integrado (*Integrated Circuit*, IC) transceptor RF AD9361 (DEVICES, 2017), cujo diagrama de blocos é mostrado na figura 10. Nela é possível observar o conjunto de blocos responsáveis pelos processos de modulação e demodulação na transmissão e recepção, respectivamente, além da interface dos dados digitais para o PC ou sistema embarcado à direita, na caixa verde. À esquerda, os blocos de recepção, na caixa vermelha, e os de transmissão, na caixa azul, com os seus respectivos

Figura 9 – Placa da plataforma SDR USRP B210 (a) e seu encapsulamento (b).



Fonte: Ettus Research (RESEARCH, 2020)

amplificadores, misturadores, osciladores e filtros.

O AD9361 garante à placa varrer a frequência de operação (portadora) de 70 MHz até 6 GHz, uma largura de banda base de 200 kHz até 56 MHz por canal, contando com 2 canais transceptores diferenciais (*Full-duplex, Multiple-Input Multiple-Output 2 Tx & 2 Rx*). Além disso, os conversores AD e DA de banda base são de 12 *bits* (RESEARCH, 2020). Como se pode observar na figura 10, cada cadeia transmissora e receptora conta com um oscilador local independente. A utilização de dois osciladores para a transmissão e recepção é uma característica comum para CIs telecomunicações em radiofrequência, já que o propósito deles geralmente é a comunicação digital, como 3G e 4G, em que sistemas duplex com divisão por frequência (*frequency division duplex, FDD*) não operam na mesma frequência de portadora na transmissão e na recepção. (DEVICES, 2017).

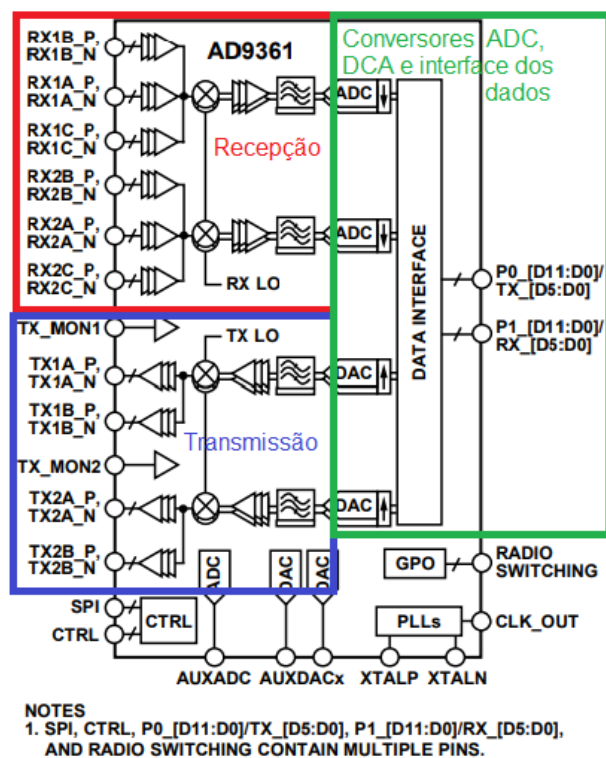
### 3.1.2 ANSYS Electronic Desktop

O Ansys Electronics Desktop (ANSYS, 2021b) é um programa desenvolvido pela empresa Ansys (ANSYS, 2021a) para simulações eletromagnéticas em geral. Nele, é possível modelar antenas, circuitos integrados, placas de circuito impresso, motores e geradores para simulações com 4 softwares inclusos, sendo eles o Ansys HFSS, o Ansys Icepack, o Ansys Maxwell e o Ansys Q3D. A tela inicial do Ansys Electronics Desktop é mostrada na figura 11.

Para este trabalho, foi utilizado o Ansys HFSS (ANSYS, 2021c), um software de simulação eletromagnética 3D para simulações de equipamentos de alta frequência. O método matemático aplicado pelo HFSS é o método dos elementos finitos (*Finite Element Method, FEM*). Este método consiste em subdividir a estrutura em estruturas menores, chamadas de elementos finitos (ANSOFT, 2009).

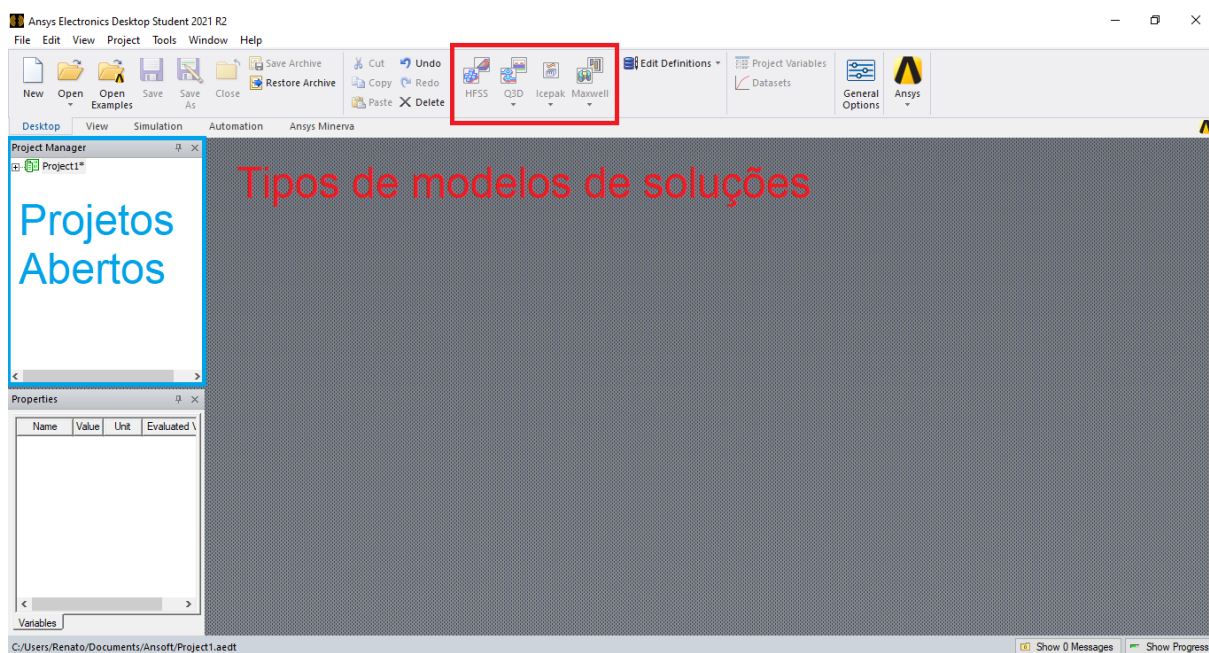
“Os elementos finitos utilizados pelo HFSS são tetraedros, e a composição total dos tetraedros é chamada de malha” (ANSOFT, 2009, Tradução do autor). A solução para os campos é encontrada dentro dos elementos finitos e de forma a respeitar as equações de Maxwell nas fronteiras entre cada elemento. A figura 12 detalha o procedimento de cálculo

Figura 10 – Diagrama de blocos do CI transceptor AD9361



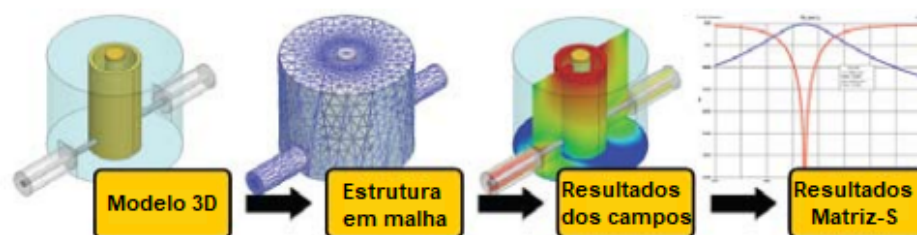
Fonte: Analog Devices (DEVICES, 2017)

Figura 11 – Tela inicial do Ansys Electronics Desktop, com destaque aos tipos de software de simulação.



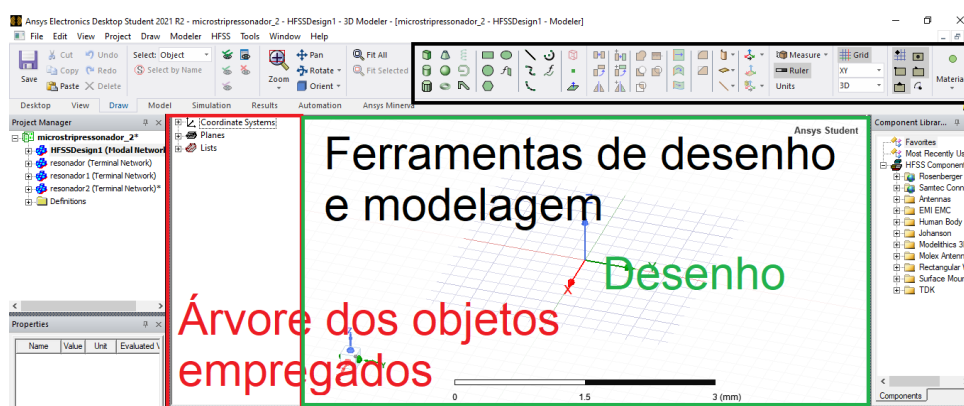
Fonte: O autor

Figura 12 – Figura mostrando o processo matemática utilizado pelo HFSS.



Fonte: Uma introdução ao HFSS, Princípios Fundamentais, Conceitos, e Utilização.  
(ANSOFT, 2009)

Figura 13 – Janela do Ansys HFSS, com destaque a árvore de objetos e aos ícones de desenho.



Fonte: O autor

utilizado pelo HFSS.

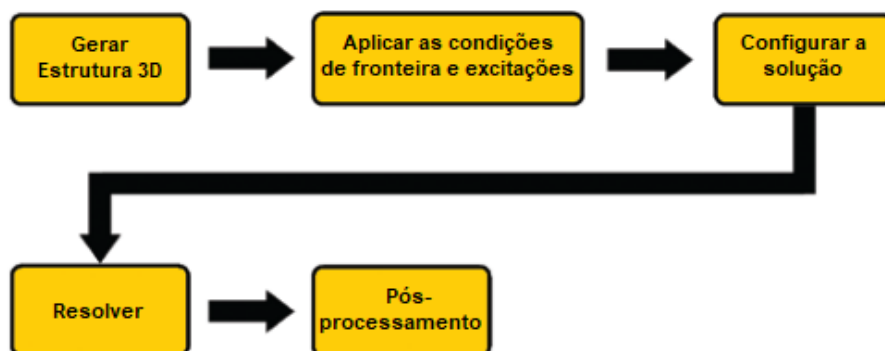
O Ansys HFSS possui ferramentas de desenho 2D e 3D para a geração da estrutura. Nos objetos em 3D são definidos os materiais, enquanto que nos 2D se define as fronteiras do sistema, como condutor perfeito, condutividade finita, excitações, etc (ANSOFT, 2009). As fronteiras são utilizadas para definir se o modelo é aberto ou fechado e para simplificar a complexidade geométrica/eletromagnética. A figura 13 mostra a janela do Ansys HFSS, dando destaque aos botões para criar a estrutura e a árvore de objetos.

De forma geral, uma simulação no HFSS envolve a geração da estrutura 3D, a aplicação das fronteiras e excitações, a configuração da solução, resolver e o pós-processamento (ANSOFT, 2009). O procedimento é ilustrado no fluxograma localizado na figura 14.

Quanto aos tipos de soluções, o HFSS dispõe de três opções: *Driven Modal*, *Driven Terminal* e *eigenmode*. A escolha depende se o usuário deseja realizar uma simulação direta do modelo, verificar a integridade do sinal ou verificar os autovalores ou ressonâncias da estrutura, respectivamente (ANSOFT, 2009).

Para a alimentação dos modelos, o Ansys HFSS dispõe de *wave ports* e *lumped ports*, principalmente. O primeiro é utilizado para excitação nos extremos da região de solução. O segundo é aplicável quando a alimentação é interna a região de solução e geralmente quando se deseja excitar um *gap* entre dois condutores.

Figura 14 – Figura mostrando o procedimento de uma simulação no HFSS.



Fonte: Uma introdução ao HFSS, Princípios Fundamentais, Conceitos, e Utilização. (ANSOFT, 2009)

A excitação por *wave port* representa uma região em que uma energia eletromagnética entra no modelo. Ela produz soluções e gráficos para os parâmetros de espalhamento (S), admitância (Y), impedância (Z), impedância de onda característica e *gamma* ( $\Gamma$ ), constante propagação e atenuação.

A excitação por *lumped port* é utilizada quando a fonte de energia precisa ser interna ao modelo. Ele também possui um detalhamento menor que o *wave port*, produzindo somente as matrizes S, Y e Z, além de precisar de uma linha de integração.

Das configurações de soluções, pode-se definir a solução principal do modelo para apenas uma, para várias ou para uma faixa de frequências. Essas opções são utilizadas de acordo com a frequência de operação principal do dispositivo simulado. De exemplo, deseja-se simular antenas para apenas certas frequências específicas, ou para uma banda larga.

Além da configuração da frequência principal da solução, o HFSS tem a opção de realizar uma varredura de frequência. A varredura pode ser ajustada para obter resultados melhores ao redor da frequência de operação (*fast sweep*), uma banda larga de frequências (*interpolation*) e com variações discretas de frequências (*discrete*).

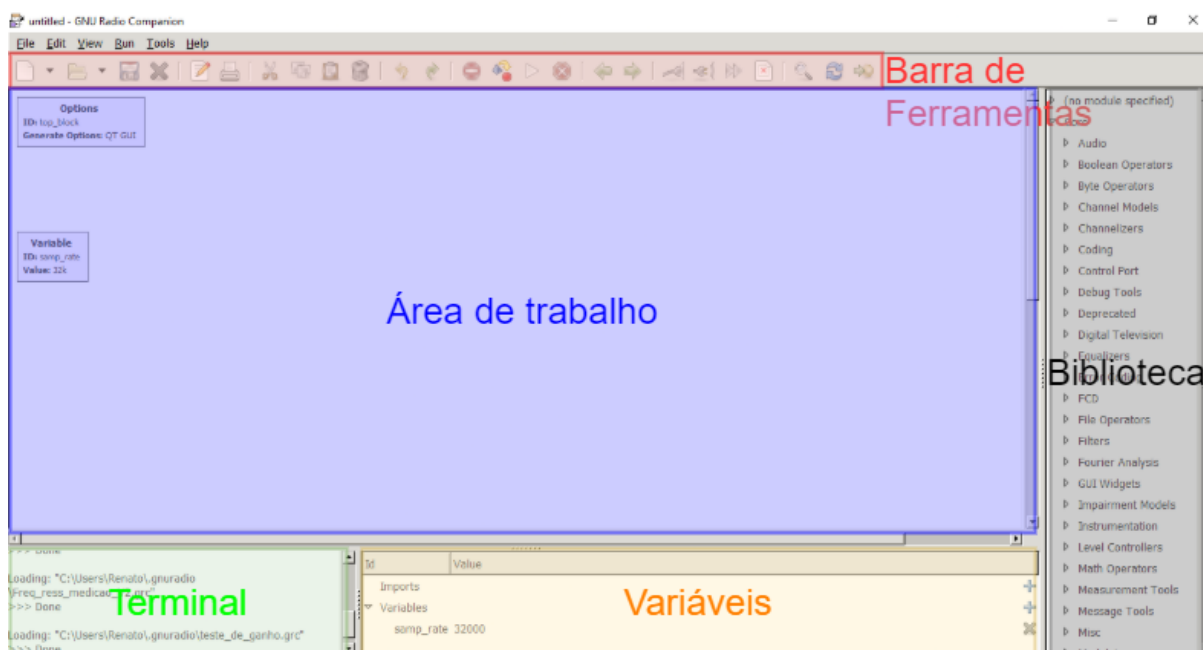
Com a configuração correta, a simulação por meio do HFSS pode trazer resultados próximos com os valores reais.

### 3.1.3 GNU Radio

O GNU Radio é um software que permite o projeto, simulação e implementação de sistemas de rádio reais (RADIO, 2021c). A programação das funções é por meio fluxogramas, com uma grande quantidade de blocos de processamento que permitem aplicações complexas de DSP.

Dentre as soluções já implementadas através do GNU Radio incluem processamento de áudio, telefonia, rastreamento de satélites, sistemas de radar, e muitos outros. O software não é uma solução em si para comunicações de rádio, porém ele pode ser utilizado para

Figura 15 – Interface do GRC versão 3.7.



Fonte: Tutoriais Guiados: GRC (RADIO, 2021b)

implementar qualquer tipo de padrão de comunicação em banda limitada (RADIO, 2021c).

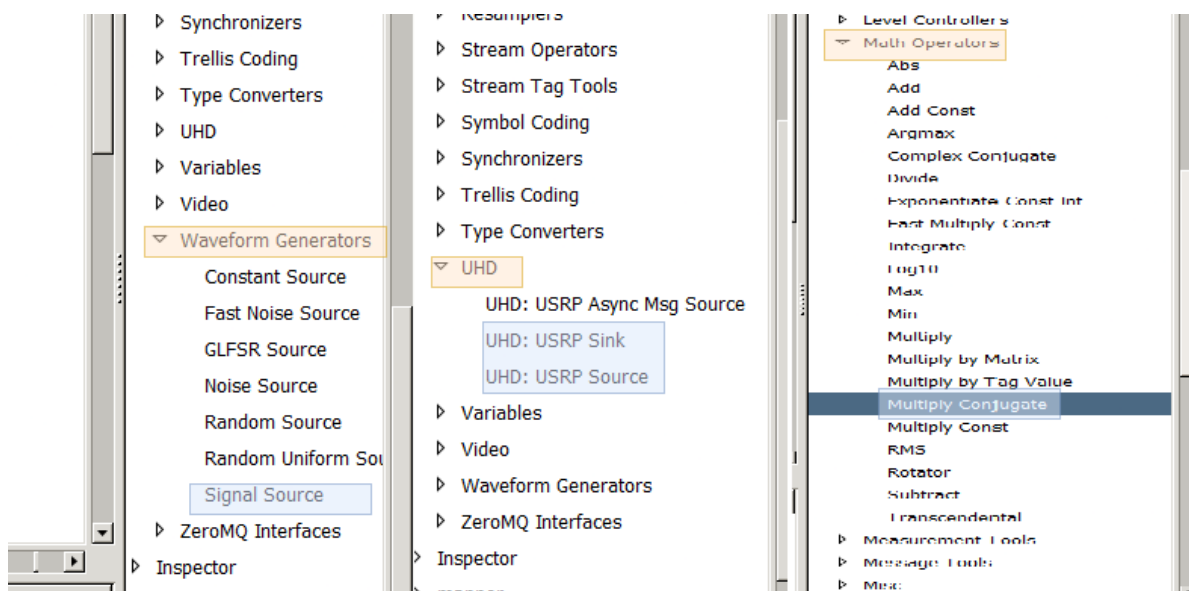
O programa GNU Radio Companion (GRC) é ferramenta gráfica do GNU Radio. Ela é utilizada de forma a criar uma maior abstração e uma programação via diagrama de blocos para o DSP, que será traduzida em linguagem python. A interface é dividida em cinco partes: a biblioteca, a barra de ferramentas, o terminal, a área de trabalho e as variáveis (RADIO, 2021b). Essa divisão é mostrada na figura 15.

Para inserir os blocos na área de trabalho, deve-se procurar ele na biblioteca. Encontrado o bloco, arrasta-se ele da biblioteca até área de trabalho. O GRC também possui uma ferramenta de pesquisa por nome para auxiliara buscá-lo (RADIO, 2021b). A figura 16 mostra as abas de geradores de forma de onda, UHD e operações matemáticas expandidas, destacando os blocos de fonte de sinal, de transmissão e recepção dos dados para a USRP e de multiplicação por conjugado complexo.

Escolhidos os blocos de DSP e inseridos na área de trabalho, o próximo passo é conectá-los. A ligação é realizada ao selecionar a saída do bloco e, em sequência, a entrada do próximo estágio desejado. A figura 17 mostra um diagrama exemplo do tutorial guiado (RADIO, 2021b). O bloco QT GUI *time sink* recebe e mostra um gráfico no domínio do tempo do sinal recebido.

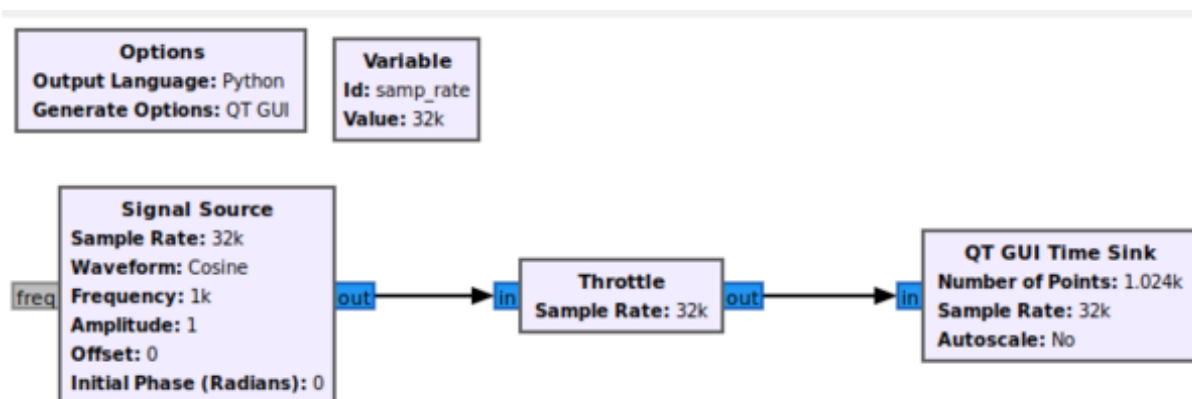
A saída e a entrada dos blocos possuem cores conforme o tipo de dado aplicado por ele. Para Este trabalho, a cor azul representa um dado complexo (Complexo de ponto flutuante de 32 bits, *Complex Float 32*) e a cor laranja representa uma variável real (Ponto flutuante de 32 bits, *Float 32*). A conexão entre os blocos só pode ser realizadas por

Figura 16 – Biblioteca no GNU Radio 3.7, com as abas de geradores de forma de onda, UHD e operações matemáticas expandidas.



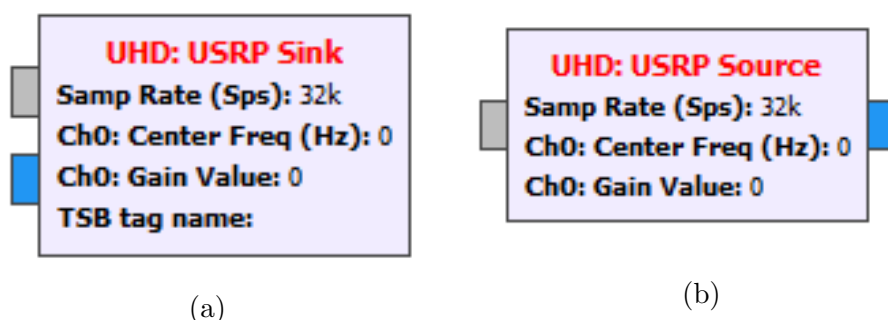
Fonte: O autor, adaptado de Tutoriais Guiados: GRC (RADIO, 2021b).

Figura 17 – Diagrama de exemplo dos tutoriais guiados, mostrando as conexões entre os blocos. Ao executá-lo, ele imprime a função cosseno no domínio do tempo.



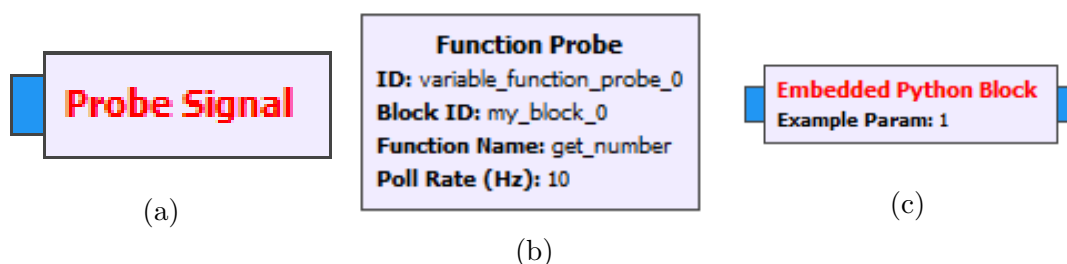
Fonte: Tutoriais Guiados: GRC (RADIO, 2021b).

Figura 18 – Blocos de comunicação com o USRP. Os blocos das figuras (a) e (b) envia e recebe o sinal para a placa USRP, respectivamente.



Fonte: O autor.

Figura 19 – Blocos utilizados para controlar variáveis e criar funções. (a) mostra o receptor de sinal, (b) transforma o sinal de recebido por (a) em uma variável e (c) pode ter a sua funcionalidade modificada ao escrever em um editor de python.



Fonte: O autor.

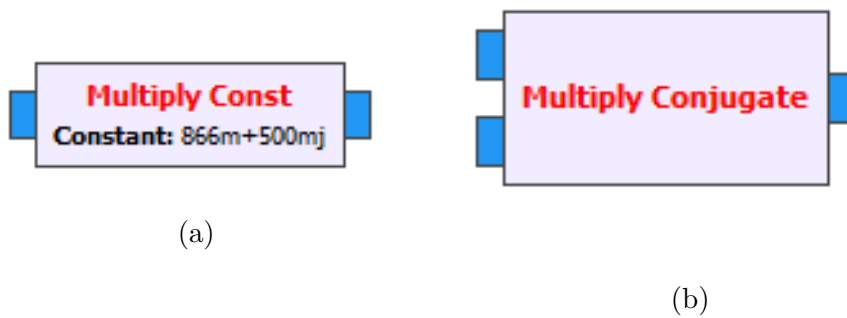
variáveis do mesmo tipo (RADIO, 2021b). O sinal cosseno complexo do GNU Radio possui na parte real e imaginária a própria função cosseno e seno, respectivamente.

A comunicação com uma plataforma USRP é realizada através dos blocos mostrados na figura 18. Os blocos recebem e enviam os dados em fase e quadratura do sinal. O bloco de fonte recebe o sinal proveniente do SDR, enquanto que o de sorvedouro envia para a placa.

Os blocos da figura 19 são importantes para o controle de um parâmetro. A sonda de sinal recebe o sinal e associa seu valor atual ao seu nome. A sonda de função utiliza o valor encontrado na sonda de sinal e o transforma em uma variável. O bloco de programação em Python (*embedded python block*) permite o usuário programar o seu funcionamento, a quantidade de portas de entrada e de saída e seus parâmetros na linguagem de programação Python.

Os últimos blocos que devem ser destacados para este trabalho são mostrados na figura 20. O bloco de multiplicação por uma constante multiplica o sinal recebido por um valor pré-determinado. Esse valor pode ser modificado caso seja atrelada uma variável do GRC ao seu parâmetro. O bloco de multiplicação conjugada realiza a multiplicação entre a primeira entrada e o conjugado complexo da segunda.

Figura 20 – Blocos de multiplicação. Em (a) se encontra o bloco de multiplicação por um parâmetro pré-determinado. (b) mostra o bloco de multiplicação conjugada entre dois sinais.



Fonte: O autor.

## 3.2 PROCEDIMENTO EXPERIMENTAL

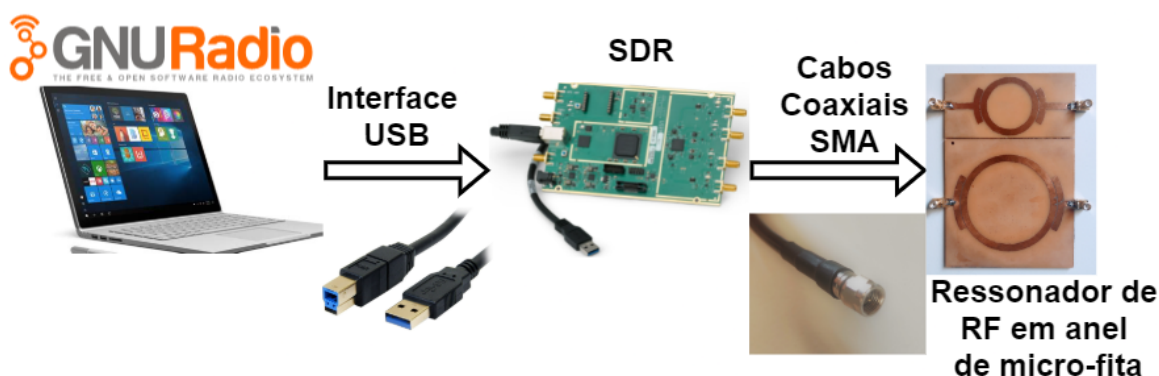
Nesta seção, serão detalhadas as rotinas utilizadas para determinar a frequência de ressonância do sensor, seu fator de qualidade e o rastreamento da ressonância. Também serão descritos os ressonadores utilizados, seus principais parâmetros e os métodos de comparação dos resultados.

### 3.2.1 Configuração da bancada de ensaios

Neste trabalho, foi montada uma bancada de ensaios composta por um computador, uma placa SDR modelo USRP B210 da Ettus Research (2020) e cabos coaxiais RG-58.

Para a bancada de ensaios proposta, foi utilizada a configuração conforme a figura 21. Nela está apresentado um computador com o *software* GNURadio, conectado com a plataforma SDR USRP B210 (RESEARCH, 2020) por meio de um cabo USB e os ressonadores, conectados com a placa de SDR por meio de cabos coaxiais RG-58.

Figura 21 – Diagrama da bancada de ensaios com figuras dos componentes

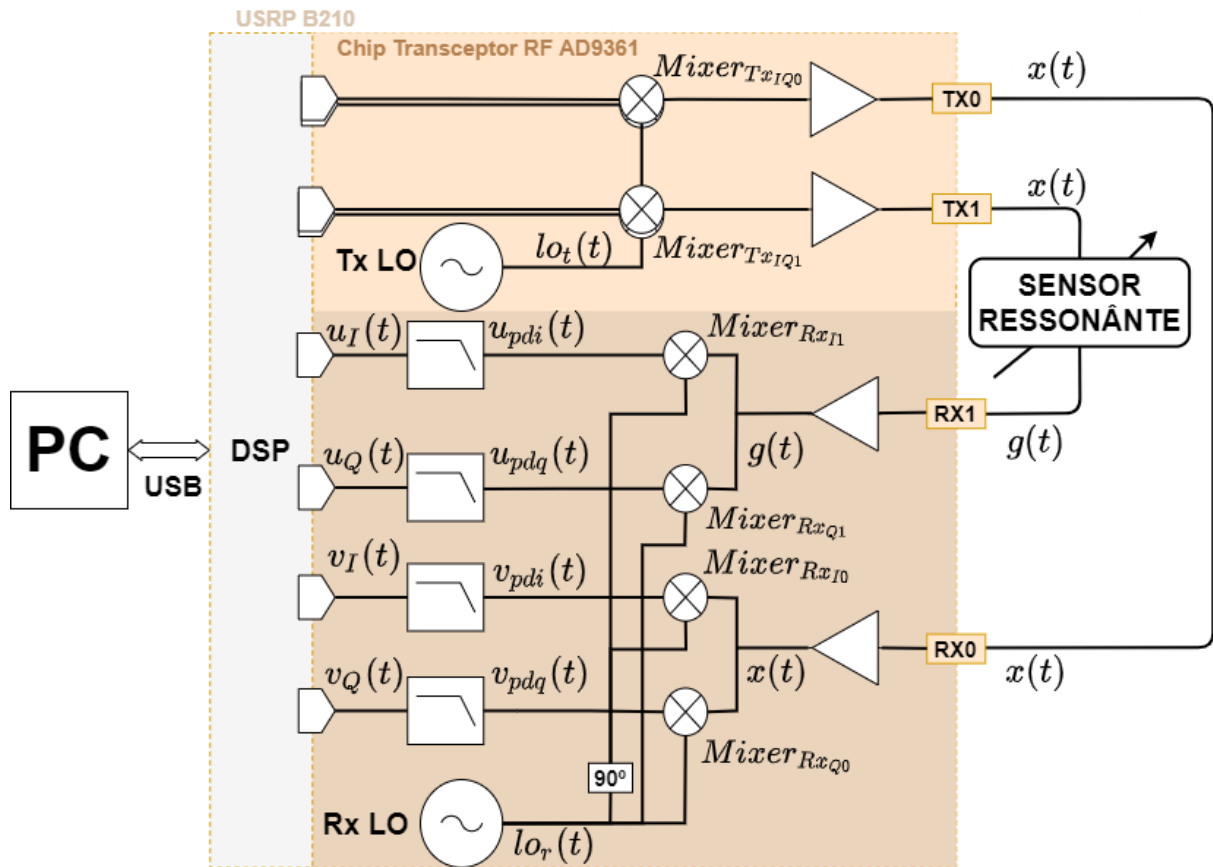


Fonte: O autor, utilizando algumas figuras de B210 (RESEARCH, 2020), "Qual computador escolher: requisitos para um bom PC"(POMES, 2021) e Cabo USB 3.0 A Macho X B Macho 1,8 metros CBUS0013 STORM (EIRELI, 2021).

A figura 22 mostra um diagrama com uma abstração maior da bancada de ensaios proposta. Nesse diagrama, é destacado o computador e sua conexão com a placa de SDR através de conexão USB, a arquitetura interna simplificada da plataforma USRP B210 (RESEARCH, 2020), com destaque ao diagrama de blocos do IC transceptor RF AD9391 adaptado da figura 10, onde os misturadores em quadratura foram evidenciados no canal de recepção (parte inferior). À direita da figura, na parte externa são observados os dois terminais de transmissão (TX0 e TX1) e recepção (RX0 e RX1). O sensor é conectado através de dois cabos entre as portas TX1 e RX1, enquanto que uma conexão direta é reservada entre TX0 e RX0.

Dentro do transceptor há um oscilador local (Tx LO) utilizado para ambos os canais de transmissão e um outro (Rx LO) para os canais de recepção. Por serem

Figura 22 – Diagrama da bancada de ensaios.



Fonte: O autor

osciladores distintos, assume-se um possível desvio de frequência e/ou fase mesmo que ambos osciladores sejam programados para terem o mesmo valor. Mesmo considerando o desvio de frequência nulo, sabe-se que cada vez que a placa é inicializada, ambos osciladores apresentam desvios de fase distintos (HERSHBERGER; PRATT; KOSSLER, 2017).

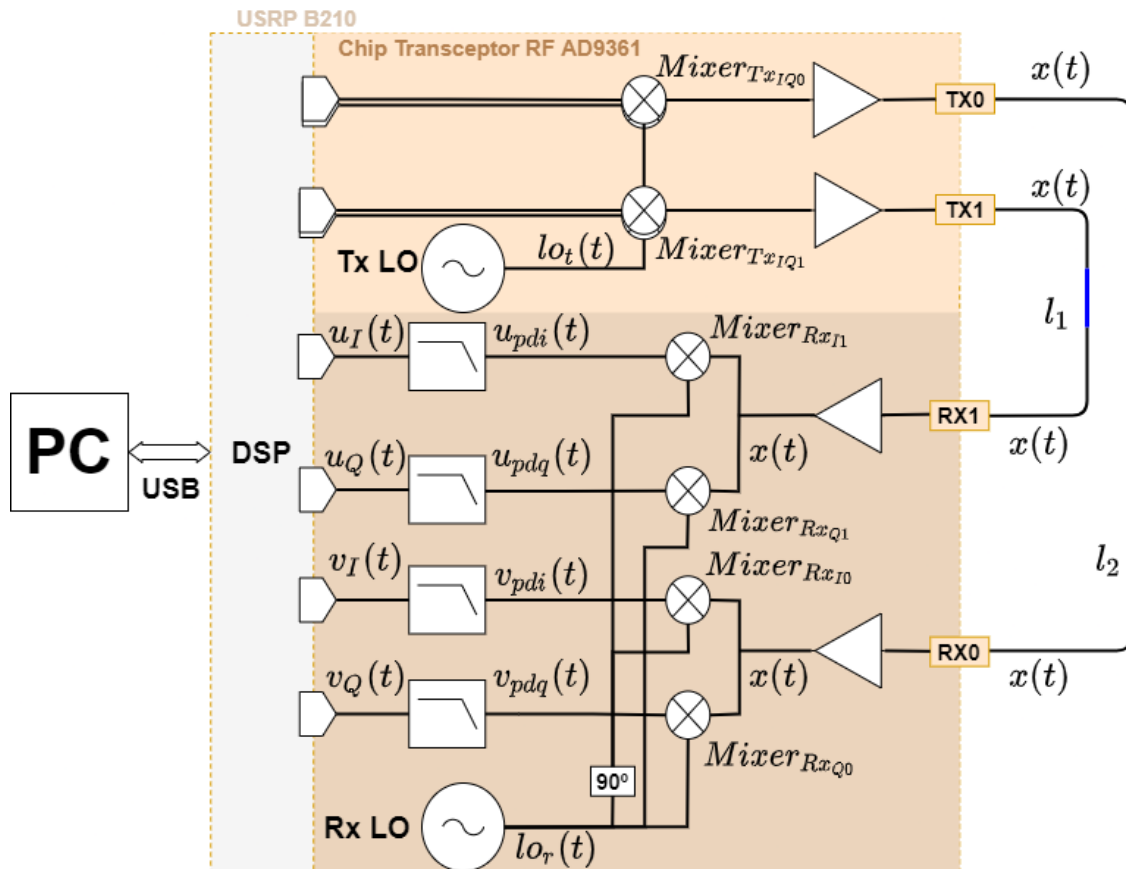
A bancada de ensaios foi programada para realizar algumas rotinas de medição. As rotinas implementadas incluem as de registro de *offset*, a de detecção da frequência de ressonância, a de caracterização do sensor e a de rastreamento de ressonância, as quais serão descritas na próxima subseção.

### 3.2.2 Implementação das rotinas

Nesta seção, serão mostrados os procedimentos para a implementação das rotinas de medição dos parâmetros do sensor, utilizando a bancada de ensaios. Será comentado como foram feitas as rotinas de registro de *offset* e de caracterização.

#### 3.2.2.1 Rotina de registro de *offset*

Como será explicado na seção da rotina de caracterização, a obtenção dos parâmetros de magnitude e fase do sensor assumem que a resposta em frequência dos pares de

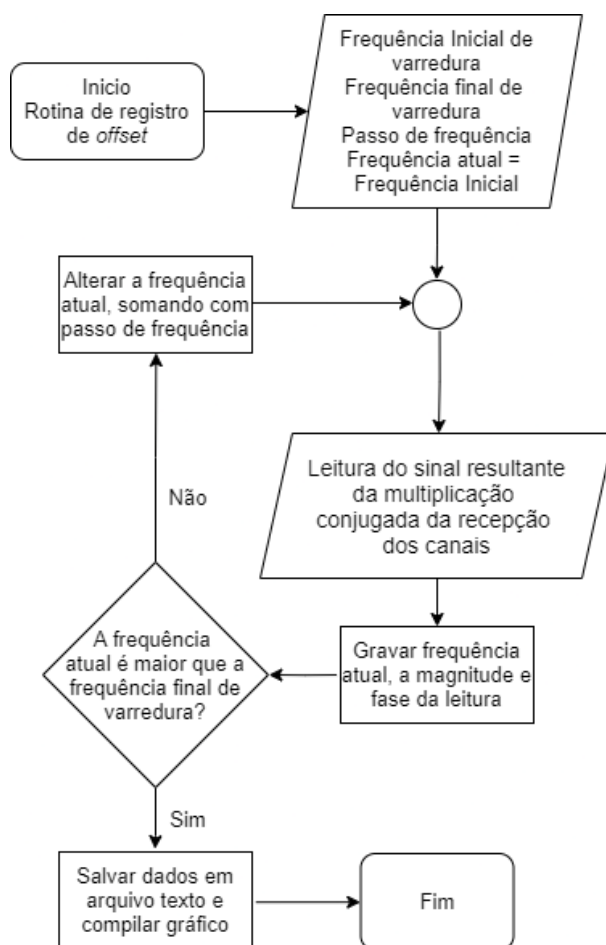
Figura 23 – Diagrama para o processo de registro de *offset*.

Fonte: O autor

interconexões internas entre os canais TX0-RX0 e TX1-RX1 são iguais, e principalmente as conexões externas (cabos e adaptadores de RF). Devido ao fato de que comprimento efetivo desses cabos ser diferente e haver certo grau de descasamento entre eles, a tal pressuposto não é necessariamente verdadeiro. Portanto, para reduzir o erro sistêmico correspondente, uma rotina de registro de *offset* de magnitude e fase entre os canais em função da frequência foi implementada na bancada de ensaios sem o sensor a ser caracterizado, como mostrado na figura 23.

Na implementação física da rotina de registro de *offset*, no canal TX0-RX0 será utilizado um adaptador reto SMA fêmea para SMA fêmea no lugar do sensor, fazendo com que o sinal enviado pelo SDR seja o mesmo que o recebido, como mostra a figura 23. Já nas rotinas de detecção da frequência de ressonância e de caracterização, o sensor foi inserido no posição indicada no diagrama mostrado na figura 22.

O fluxograma da figura 24 traz o procedimento da rotina de registro de *offset* quanto ao tratamento do sinal e da frequência, realizando a varredura e o armazenamento do sinal complexo. O sinal lido é resultado da aplicação da multiplicação complexa dos sinais de entrada no SDR, mostrado no apêndice A. Os valores armazenados são escritos tanto em um arquivo de texto quanto compilados em um gráfico de bode.

Figura 24 – Algoritmo para a rotina de registro de *offset*

Fonte: O autor

### 3.2.2.2 Obtenção da frequência de ressonância

Um dos parâmetros do sensor ressonante é o valor de sua frequência de ressonância. Essa medida indica se algum objeto foi inserido no ressonador e, no caso de se analisar a propriedade do material, retorna a parte real da permissividade complexa do sensor, pela equação (2). Assim, determiná-la é uma parte fundamental da caracterização do sensor.

Para encontrar a frequência de ressonância, foi utilizada uma rotina com variação de frequência e detecção da diferença de fase entre os sinais, é montado um detector de zero (RUGGIERO; LOPES, 1996), sendo o zero o resultado da diferença entre o ângulo atual e o da ressonância.

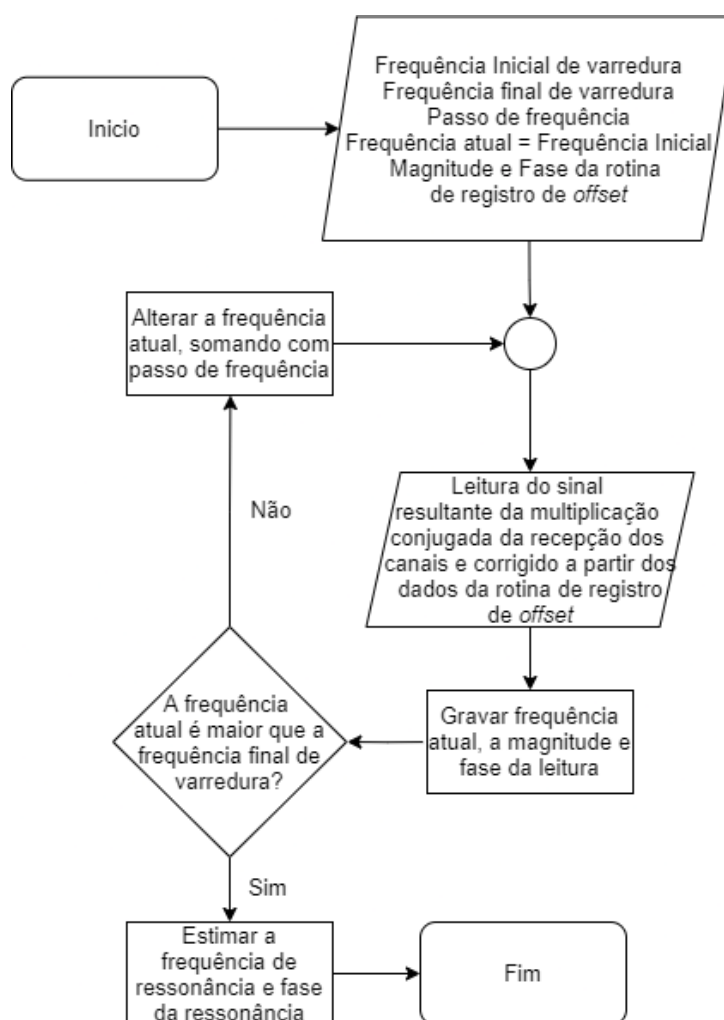
Em um primeiro momento, o algoritmo ajusta e realiza a leitura do vetor de frequências, magnitude e fase obtidos da seção 3.2.2.1, para realizar a correção da leitura. Após isso, o algoritmo realiza uma varredura nos arredores da frequência de ressonância com o intuito de determinar se o sensor em transmissão possui uma resposta próxima a de um filtro passa-faixa ou de um rejeita-faixa, de se obter seu ângulo de ressonância e estimar a frequência de ressonância.

Essa varredura possui uma faixa de frequência menor do que aquela descrita na

subseção 3.2.2.1, gravando os valores magnitude e fase do sensor no canal TX1-RX1 após a multiplicação complexa com a recepção do canal TX0-RX0. Com esses dados, a frequência de ressonância é estimada com a frequência onde se encontrou a maior amplitude, no caso do sensor que se aproxima ao comportamento de um filtro passa-faixa, enquanto que a fase de ressonância é tomada pela média das fases do vetor. O diagrama de blocos dessa varredura é mostrado na figura 25.

Os valores gravados recebem uma tratativa de média com relação à 7 valores medidos ao redor no vetor, sendo eles no próprio elemento do vetor, 3 acima e 3 abaixo. O motivo disso é que a fase do gráfico toma a forma esperada após a aplicação dessa média móvel.

Figura 25 – Algoritmo da varredura ao redor da ressonância.



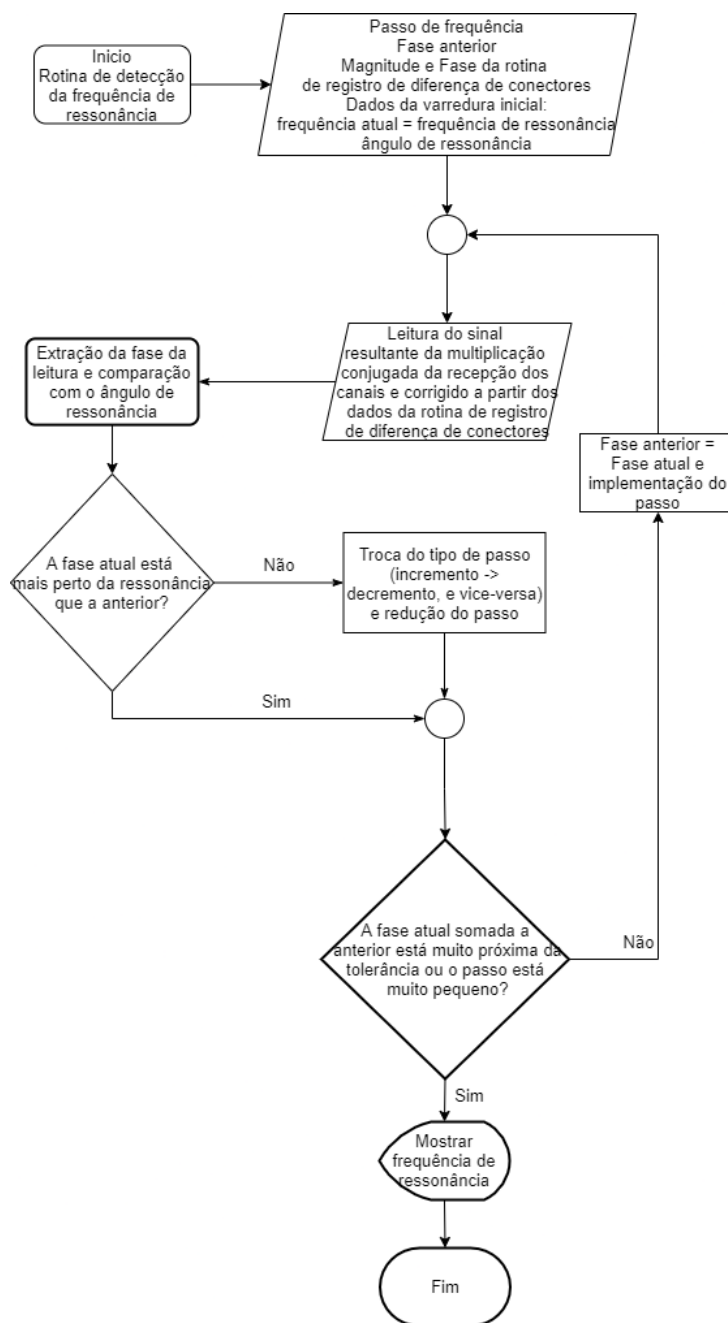
Fonte: O autor

Extraído que tipo de filtro representa a resposta do sensor e o ângulo da ressonância, o algoritmo passa a comparar o ângulo do sinal lido com relação ao de ressonância e altera a frequência de saída por meio de passos incrementais ou decrementais de frequência. O tipo de passo é determinado ao gravar a leitura da fase na frequência anterior e comparar com a atual. Caso a leitura de fase anterior esteja mais próxima do ângulo de ressonância,

o passo altera o seu sentido, de incremento para decremento, e vice-versa, e é reduzido por um valor determinado, neste caso pela metade.

Quando a soma dos módulos da fase lida e da anterior estiverem nos limites de uma tolerância, ou a variação de frequência estiver muito abaixo da inicial, a rotina é finalizada. A figura 26 mostra o fluxograma da rotina para encontrar a frequência de ressonância.

Figura 26 – Algoritmo para a rotina de medição da frequência de ressonância pela fase.



Fonte: O autor

### 3.2.2.3 Rotina de determinação da largura de banda e fator de qualidade

Finalizada a obtenção da frequência de ressonância utilizando o SDR na seção 3.2.2.2, o próximo passo para caracterizar o sensor é obter o seu fator de qualidade. Os parâmetros de largura de banda e fator de qualidade retornam tanto a relação de quanto o sensor é seletivo ou, para a propriedade do material, a relação de perdas do dielétrico.

Para a obtenção da largura de banda e o fator de qualidade, serão utilizadas as frequências de meia potência. A rotina utilizará a frequência de ressonância, obtida a partir da seção 3.2.2.2, como ponto de partida e irá incrementar a frequência até atingir a frequência de meia potência superior. Posteriormente, a rotina retornará a ressonância e irá reduzir a frequência até encontrar a frequência de meia potência inferior. Para um sensor com resposta similar a um filtro passa-faixa, a referência utilizada para as frequências de meia potência foi -3 dB da amplitude do sinal lido na ressonância. Para fins de comparação, a largura de banda também foi obtida a partir da fase do sensor e comparada com o valor obtido a partir da amplitude.

Obtida a largura de banda, aplica-se a equação (3) para se obter o fator de qualidade. Os dados que caracterizam o filtro serão escritos no terminal do GRC e em um arquivo texto para ser posteriormente acessados. O fluxograma para a rotina de obtenção da largura de banda se encontra na figura 27.

### 3.2.2.4 Rotina de rastreamento dinâmico

Com o procedimento testado, realizou-se uma rotina de rastreamento dinâmico para detectar a variação da frequência de ressonância do sensor. A rotina realiza a leitura constante do valor da fase do sinal resultante em banda base. Ao ser detectada a alteração do valor da fase ela verifica o tipo de sensor e passa a procurar a nova ressonância, seguindo um procedimento similar ao dado na seção 3.2.2.2, verificando a fase do sinal. O diagrama de fluxo da figura 28 mostra o funcionamento da rotina, sendo somente interrompida pelo término da execução do programa.

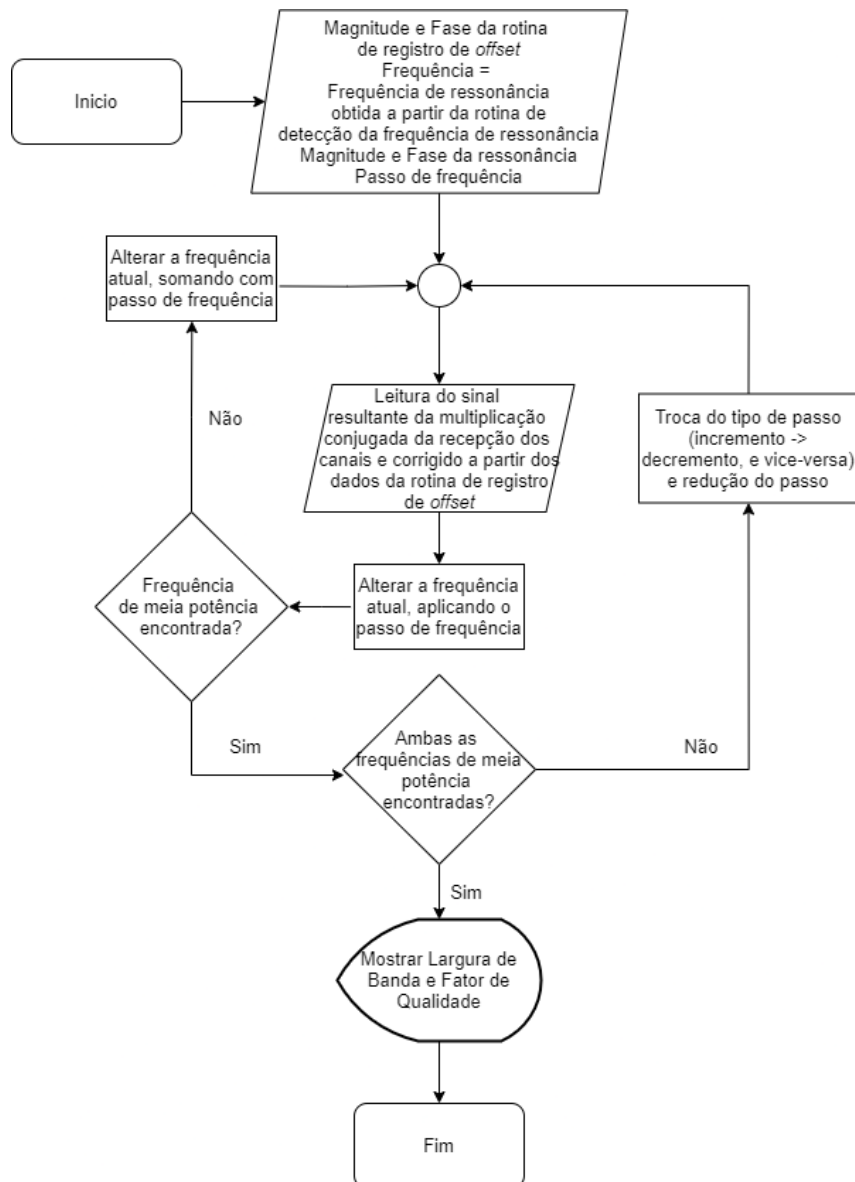
### 3.2.2.5 Implementação no GNURadio

Para aplicar as rotinas de medição no GNURadio, foi utilizado o bloco de programação em Python. Os blocos resultantes se encontram na figura 29.

Ambos os blocos utilizam duas entradas e uma saída. A entrada complexa recebe o sinal resultante da multiplicação conjugada da medição dos canais RX0 e RX1, enquanto que a recepção do valor real coleta um sinal de onda quadrada, utilizada para controlar o tempo de cada passo, garantindo a estabilidade do sistema antes de ler o sinal.

Como saída, o bloco envia o valor da frequência atual, que é coletado pelo bloco de sonda e transformada em variável para controlar a frequência dos osciladores locais. Os parâmetros do bloco indicam o comprimento da varredura de frequência, o passo de

Figura 27 – Algoritmo para a rotina de medição da largura de banda.



Fonte: O autor

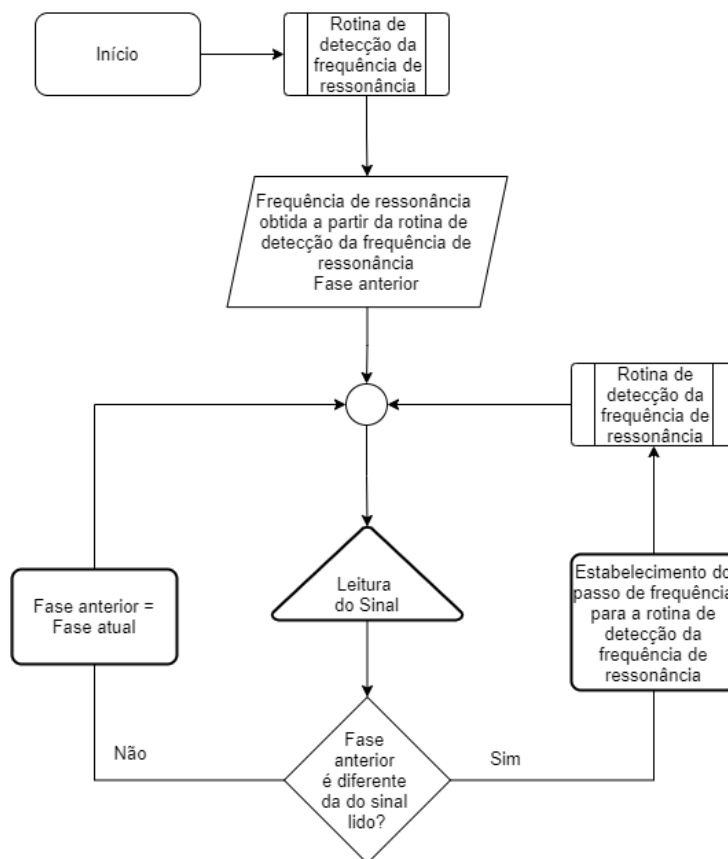
frequência e a localização dos arquivos onde os dados de frequência, magnitude e fase são gravados ou lidos para serem utilizados na geração dos gráficos ou correção das leituras.

As rotinas de detecção de largura de banda e de rastreamento de ressonância foram implementadas em conjunto com a rotina de detecção de ressonância. O motivo dessa implementação conjunta é para o programa utilizar o resultado obtido da frequência de ressonância para medição da largura de banda ou para rastrear a frequência de ressonância.

### 3.2.3 Simulação dos algoritmos

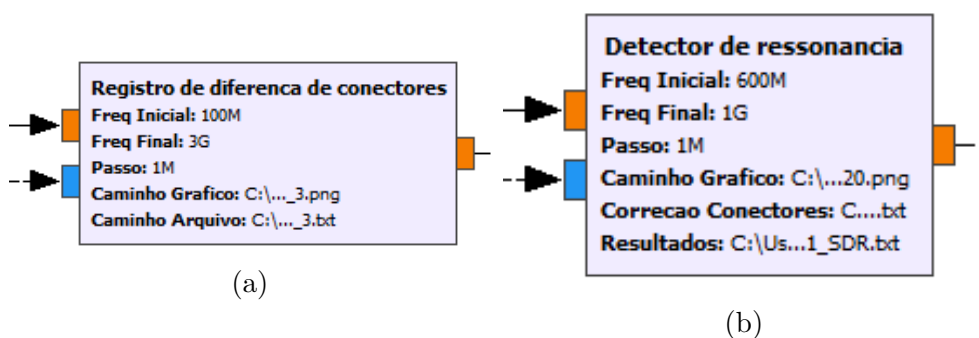
Antes de executar os algoritmos listados na seção 3.2.2 na placa de SDR, os algoritmos de registro de *offset*, de obtenção e de rastreamento da frequência de ressonância foram simulados para verificar a sua funcionalidade. O bloco de multiplicação constante do

Figura 28 – Diagrama de fluxo para rastreamento da frequência de ressonância.



Fonte: O autor

Figura 29 – Bloco de programação customizada com os códigos em Python referentes as rotinas implementadas. (a) mostra o bloco utilizado para a rotina de registro de diferença de conectores. (b) mostra o bloco utilizado para a detecção de ressonância, a medição da largura de banda e o rastreamento da ressonância.

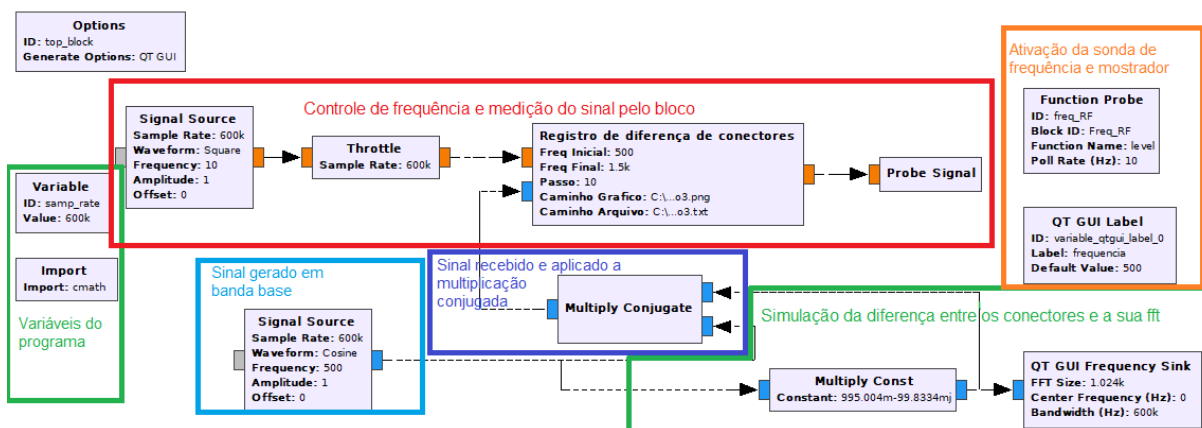


Fonte: O autor

GNU Radio permite simular o comportamento de fase e amplitude do sensor ressonante com o auxílio da variável frequência. O programa em diagrama de blocos feito no GRC para as rotinas de registro de *offset* e de rastreamento da frequência de ressonância são mostrados nas figuras 30 e 31, respectivamente.

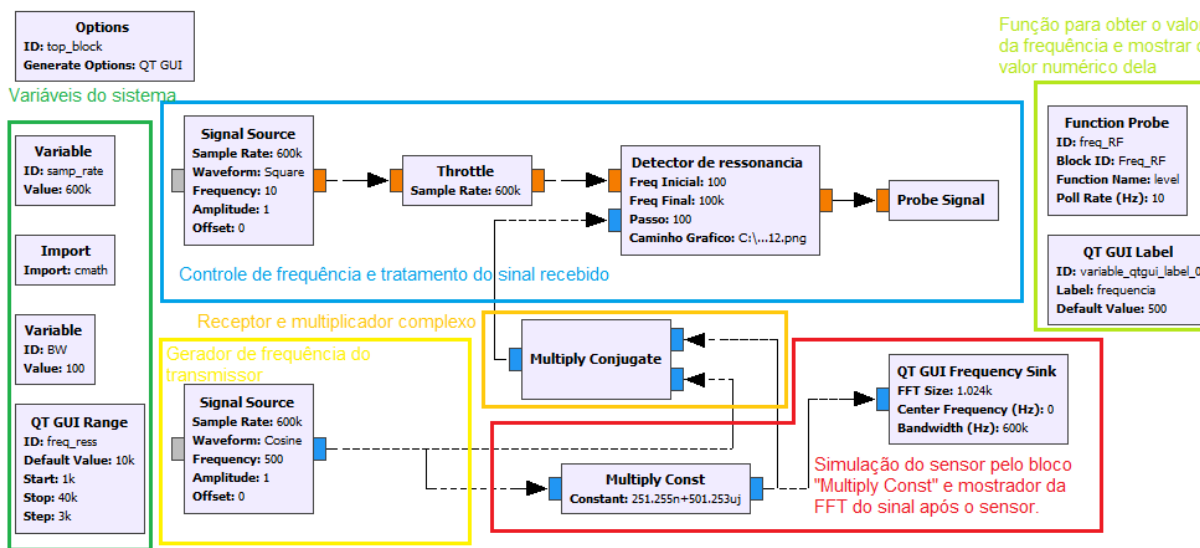
A simulação foi feita com uma frequência absoluta reduzida para não utilizar todo o processamento do computador e considerando a rotina de registro de *offset*. A varredura

Figura 30 – Diagrama de blocos do GNU Radio para simular o algoritmo de registro de *offset* do sistema.



Fonte: O autor

Figura 31 – Diagrama de blocos do GNU Radio para simular os algoritmos de detecção de frequência de ressonância, caracterização e rastreamento de frequência de ressonância.



Fonte: O autor

de frequência foi gerada através do conjunto de blocos salientado no quadro azul da figura 31, que implementa o algoritmo descrito na secção 3.2.2, tendo como saída a frequência da portadora do oscilador que é recebida pelo bloco de sonda (*Probe Signal*) de sinal e interpretada como variável pelo bloco de função da sonda (*Function Probe*) encontrado no quadro amarelo claro da figura 31. O bloco de multiplicação constante utiliza a frequência extraída pelo *Function Probe* para gerar a diferença de magnitude e fase em função da frequência.

### 3.2.4 Sensor de micro-ondas ressonante de micro-fita em anel

Para a validação do algoritmo e do banco de ensaio, é preciso medir um sensor ressonante e comparar os resultados com a referência de outro dispositivo de medição. No caso, o dispositivo que será utilizado como referência para a comparação será um analisador vetorial de rede (VNA) (POCKETVNA, 2021a) para caracterizar o sensor.

Dos sensores de micro-ondas mostrados na seção 2.1, foram pesquisadas os tipos e as características de frequência de ressonância e largura de banda dos sensores planares. Dentre as estruturas ressonadoras, destaca-se: a linha de meio comprimento de onda, o toco paralelo de um quarto de comprimento de onda, o anel, e o dielétrico (POZAR, 2011). Foi escolhido o sensor de micro-fita em formato de anel para a realização dos testes. Dentre os motivos da escolha desta estrutura, cita-se: o formato simples; frequências de ressonâncias menores com relação às medidas da placa, principalmente quando comparado aos ressonadores de toco ou em linha de um quarto ou de meio comprimento de onda, respectivamente; e a possibilidade de inserir um material dielétrico com uma área de efeito maior que os outros ressonadores mencionados.

Os sensores foram fabricados utilizando como base uma placa FR-4 possuindo com uma espessura de 1,6 mm, com um valor de permissividade relativa estimada em 4,4 ( $\epsilon_r = 4,4$ ), pois o fabricante não o fornece. O contorno utilizado para melhorar o acoplamento possui um ângulo de abertura de  $60^\circ$ . Utilizando as equações (8) e (9) (POZAR, 2011), os parâmetros assumidos e calculados dos sensores são mostrados na tabela 1.

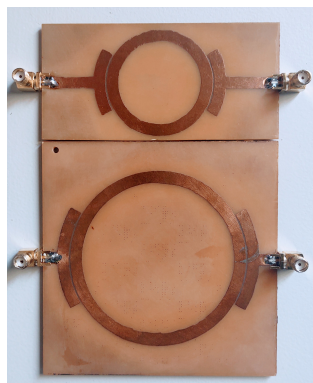
Tabela 1 – Dados dos sensores fabricados.

Sensor	$h_{placa}$ (mm)	$r_m$ (mm)	$w_{trilha}$ (mm)	$f_0$ (MHz)	$g$ (mm)	Material	$Z_0$ ( $\Omega$ )
A	1,6	34,995	4,53	742	0,5	FR-4 ( $\epsilon_r = 4,4$ assumido)	38,55
B	1,6	19,995	4,53	1299	0,5	FR-4 ( $\epsilon_r = 4,4$ assumido)	38,55

Fonte: O autor

A figura 32 mostra os sensores planares prontos com os conectores para os cabos coaxiais. O anel menor foi dimensionado para uma frequência de ressonância fundamental de 1,299 GHz e o anel maior de 742 MHz, segundo os cálculos teóricos. Como observado pela tabela 1, a impedância característica da placa é diferente do resto do sistema. Esse descasamento das impedâncias causará reflexões no sistema, como mencionado na seção 2.3, com o módulo do coeficiente de reflexão calculado em 0,1293 ( $|\Gamma| = 0,1293$ ).

Figura 32 – Placas produzidas, sendo a menor para 1,299 GHz e a maior para 742 MHz, segundo cálculos teóricos. Fabricadas por Marcelo Jefferson Barbosa, técnico em eletroeletrônica pelo IFSC.



Fonte: O autor

### 3.2.5 Simulação dos sensores

Para a verificação da resposta dos sensores, eles foram modelados e simulados no *software* ANSYS *Electronic Desktop*. A utilização do HFSS Ansys traz a possibilidade de obter os parâmetros de espalhamento do sensor em uma faixa de frequência ampla.

A placa de circuito impresso foi modelada com uma caixa com as dimensões dos sensores fabricados, sendo 100x100x1,6 mm para o sensor A e 100x50x1,6 mm para o sensor B, respeitando as medidas físicas da placa. O material utilizado no HFSS foi o FR-4\_epoxy, em que a permissividade elétrica relativa e a tangente de perdas estão localizadas na tabela 2.

O plano terra da placa foi modelado aplicando uma condição de fronteira de condutor perfeito na face inferior da caixa. Já as micro-fitas, localizadas na face superior, foram realizadas através dos objetos 2D: círculo, retângulo e curva por função. Esses formatos foram utilizados para implementar o anel, as LTs de alimentação e o contorno de acoplamento. Todos os condutores da face superior foram definidos como fronteira de condutância finita, utilizando como parâmetro o cobre, como mostra a tabela 2. As medidas físicas dos sensores A e B foram replicadas para no HFSS, utilizando a escala de mm.

Determinados os parâmetros geométricos, deve-se definir as excitações do modelo HFSS. Com esse fim, foram desenhados dois retângulos, do plano terra às duas LTs de alimentação do anel, com a espessura da micro-fita. Foi aplicado um *lumped port* nas portas do sensor, com uma curva de integração da face inferior a face superior.

O volume de simulação foi definido utilizando a função de criar região (*create region*), com uma condição de fronteira de radiação, e aplicando um preenchimento de 50 mm a mais em todas as direções. Essa configuração foi escolhida com o fim de impedir que linhas de campo ultrapassem o volume de simulação. O tipo de solução utilizado foi o padrão do HFSS, com a opção de análise de rede (*Network Analysis*), como mostra a

tabela 3.

A solução principal foi configurada conforme mostram os dados da tabela 3. Além disso, foi aplicada uma varredura de frequência para obter os resultados gráficos dos parâmetros de espalhamento obtidos através da simulação, tendo os seus parâmetros configurados conforme mostra a tabela 4.

Tabela 2 – Configurações de simulação no ANSYS para as micro-fitas e o material dielétrico da placa.

Sensor	Material da trilha	$\sigma_{trilha}$ (S/m)	Material plano terra	$\sigma_{GND}$ (S/m)	Material dielétrico	$\epsilon_r$ (F/m)	$\delta_{perdas}$
A e B	Cobre	$5,8 \times 10^7$	PEC	$\infty$	FR4_epoxy	4,4	0,02

Fonte: O autor

Tabela 3 – Configurações de solução no ANSYS.

Sensor	Frequência de solução	Frequência inferior	Frequência superior	nº de passos	tipo de solução	Opções de solução
A	Banda Larga	1 MHz	2 GHz	99	HFSS	Análise de rede, <i>modal</i>
B	Banda Larga	1 GHz	3 GHz	99	HFSS	Análise de rede, <i>modal</i>

Fonte: O autor

Tabela 4 – Configurações da varredura no ANSYS.

Sensor	Tipo de varredura	Distribuição	Frequência inferior	Frequência superior	nº de passos	$\Delta f$ do passo
A	Interpolação	Passo linear	1 MHz	2 GHz	19991	100 kHz
B	Interpolação	Passo linear	600 MHz	3 GHz	24001	100 kHz

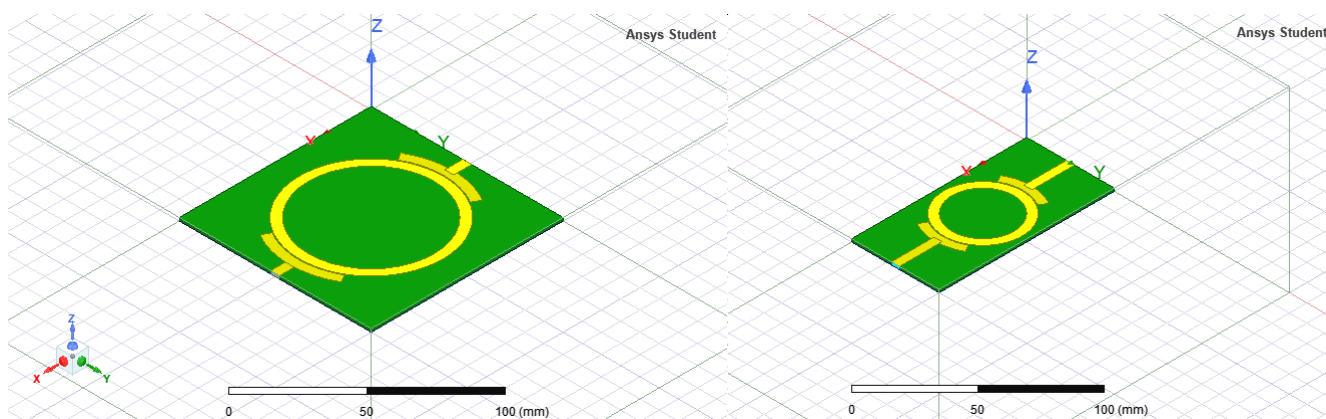
Fonte: O autor

Foram extraídas as respostas do sensor com os parâmetros de espalhamento, pelo coeficiente de transmissão ( $S_{21}$ ) para a resposta em frequência e pelo coeficiente de reflexão ( $S_{11}$ ) para as cartas de Smith de ambos os sensores para a observação da sua resposta em ressonância sem nenhum material inserido sobre ele, que representa o cenário descoberto. Os resultados foram exportados e analisados por um software de análise matemática.

### 3.2.6 Cenários de teste do Sensor

Com a rotina de rastreamento da ressonância e o sensor caracterizado, será aplicado água destilada em um recipiente de plástico na superfície do sensor para alterar a sua frequência de ressonância. As características do sensor serão medidas quando ele estiver descoberto, quando houver 50% de sua área coberta e quando estiver totalmente coberto

Figura 33 – Sensores modelados no ANSYS Electronic Desktop.



Fonte: O autor

pelo material. As medições utilizando o SDR serão confrontadas com as do VNA sob as mesmas condições.

Para a inserção dos recipientes na superfície da placa, foi considerada totalmente coberta quando todo o anel de ressonância da placa fosse coberto com o recipiente contendo água, tendo a água pelo menos 2 centímetros de altura, de forma a garantir que as linhas de campo acima da placa estejam inseridas no material dielétrico. Já para as condições de 50% preenchida, utilizou-se as metades que eram preenchidas conforme mostram as figuras 34a e 34b, a primeira sendo perpendicular em relação a alimentação e a segundo em paralelo.

Figura 34 – Posições perpendicular (a) e paralela (b) à LT para inserir o recipiente na superfície do sensor.



(a)

(b)

Fonte: O autor

### 3.2.7 Implementação em Bancada

Nesta seção será abordado alguns aspectos da implementação em bancada. Com a principal transmissão de tensão e de corrente sendo por LTs, a utilização da teoria de LT é de grande auxílio para os ajustes matemáticos além da registro de diferença de conectores.

### 3.2.7.1 Compensação da inclinação de fase

Como o circuito de micro-ondas possui um comprimento elétrico menor que as suas dimensões físicas, a fase da onda de tensão recebida possui um atraso com relação a enviada. Caso seja fixado um determinado comprimento para os cabos e sejam realizadas medições em uma varredura de frequência, a fase apresenta uma inclinação linear, em função da frequência.

A inclusão do ressonador proporciona uma resposta de fase similar a soma desse efeito de inclinação com a resposta do sensor em transmissão. Para que a resposta do sensor em transmissão apresente uma resposta de fase mais próxima do filtro ideal que o represente, essa inclinação foi medida e compensada.

A medição da inclinação da fase foi realizada com o sensor conectado, durante a varredura menor. Com os valores de fase e de frequência gravados, é analisada uma região que esteja fora da ressonância, para que o efeito calculada a média da inclinação da fase ao longo da varredura.

Tomando-se como exemplo a figura 35, os valores de fase analisados se encontram entre as frequências de 100 MHz e 400 MHz. A curva azul representa a resposta de fase do sensor sem a compensação. Com isso, se calcula a variação de fase com relação a variação de frequência para se obter a inclinação da fase sem a interferência da ressonância.

Uma função foi implementada para calcular o coeficiente de inclinação da curva de fase na região com os dados analisados. Um exemplo desse cálculo é mostrado na figura 35, onde a inclinação obtida entre as frequências de 100 MHz e 400 MHz foi de  $-1,08 \times 10^{-9}$ . Este valor aplicado ao sinal como mostra a equação (14), em cada ponto de medição.

$$H_{comp}(f) = H(f) \cdot 1\angle -k \cdot f = |H(f)|\angle\phi_H(f) \cdot 1\angle -k \cdot f \quad (13)$$

$$H_{comp}(f) = |H(f)|\angle\phi_H(f) - k \cdot f \quad (14)$$

Onde:

$H_{comp}$  é o sinal com a fase compensada;

$H$  é o sinal original;

$\phi_H$  é a fase do sinal original

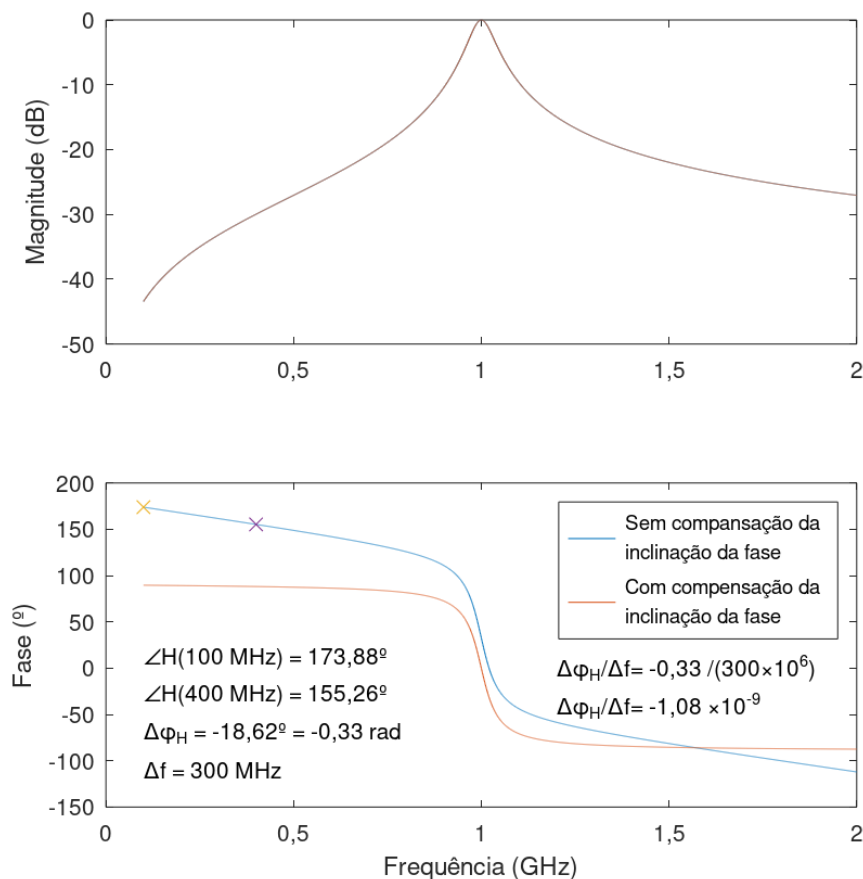
$f$  é a frequência;

$k$  é o coeficiente de inclinação da fase calculado;

O resultado da inclinação de fase é aplicado ao sinal recebido dos canais após a multiplicação conjugada de forma a compensar essa inclinação. Medições pre-liminares realizadas com o sensor no cenário descoberto indicaram que o coeficiente da inclinação de fase é de  $-2,518 \times 10^{-9}$  ou  $-6,35 \times 10^{-8}$ , dependendo se os dados de *offset* fossem utilizados ou não, respectivamente. Os dados foram extraídos de forma a utilizar um valor padrão caso a função não obtivesse o cálculo da inclinação de fase corretamente, seja por

variações na fase ou por inserção do material dielétrico. A figura 35 mostra a resposta esperada do sensor na curva azul, e a resposta com a compensação da fase.

Figura 35 – Exemplo do efeito da inclinação da fase sob um filtro passa faixa ideal proporcionado pela LT.



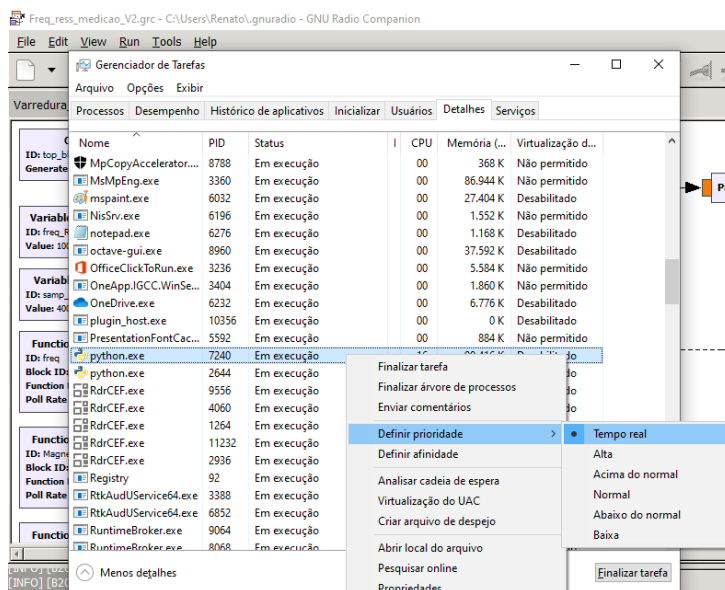
Fonte: O autor

### 3.2.7.2 Configurações e parâmetros do GNU Radio em bancada

Aqui serão listados os parâmetros e o diagrama de blocos utilizado para gerar o programa e extrair os dados dos sensores. Ao utilizar uma plataforma de SDR, o programa GNU Radio utiliza a máxima velocidade de transmissão de dados possíveis para o computador. Caso o último não tenha capacidade de processar os dados corretamente, o programa gera mensagens de erro no terminal, podendo até travar e deve-se reajustar os parâmetros para minimizar esses erros.

Para o funcionamento do programa, utilizando a plataforma SDR, com o mínimo de erros provenientes de *overflow* e *underflow*, definiu-se a taxa de amostras enviadas e obtidas da plataforma para inferior a 1 milhão de amostras por segundo, junto com a fixação da prioridade de execução do programa como tempo real, através do gerenciador de

Figura 36 – Definindo a prioridade de execução do programa em tempo real, para o Windows.



Fonte: O autor

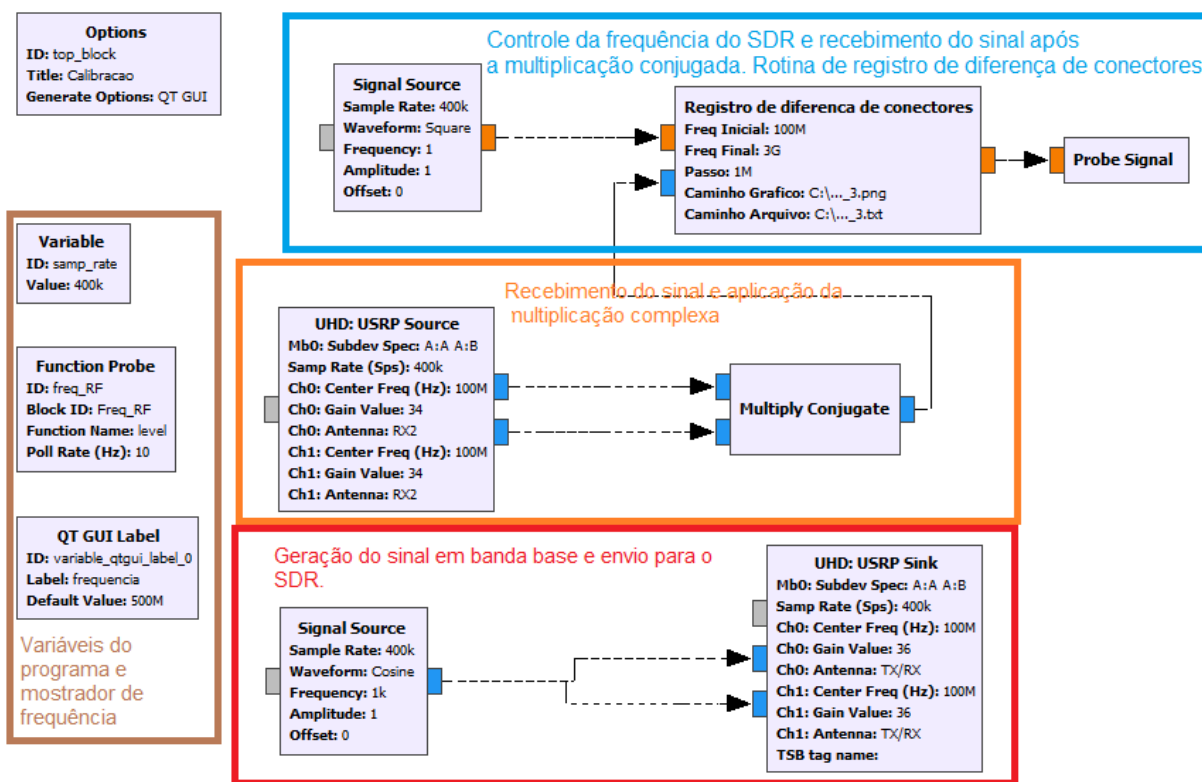
tarefas, conforme a figura 36. As mensagens de erro ainda aparecerem durante a execução do programa com essa taxa de amostragem, porém sem interferência nos resultados e sem travar o programa. Também foi fixada uma taxa de leitura do sinal pelo bloco inferior a 2 por segundo, para que o sistema alcance uma certa estabilidade.

O RFIC AD9361, da plataforma USRP B210, possui filtros passa-faixa para a recepção e o envio da banda base. Isso se deve para satisfazer o critério de Nyquist para a amostragem e eliminar possíveis polarizações CC (MEANEY et al., 2019). Devido a essas características, fez-se necessário gerar um sinal de banda base de pelo menos 1 kHz.

Para os ganhos dos misturadores de transmissão e recepção do sinal, teve-se o cuidado para não enviar um sinal com potência superior ao limite da plataforma, para não danificá-la e também de forma a não saturar o sinal. Segundo as curvas de potência fornecidas pela Ettus Research (2021), um ganho de transmissor abaixo de 40 dB ainda é um valor seguro para a utilização. Testes com um programa utilizando variáveis ajustáveis durante a execução determinaram que um ganho de 36 dB e 34 dB para os canais de transmissão e recepção, respectivamente, não saturam o canal sem o sensor na faixa de operação desejada, enquanto que canal com o ressonador permite ganhos de 40 dB de transmissão e 54 dB de recepção. Devido a variação da resposta do sistema ocasionada pela modificação do ganho, todas as rotinas utilizaram os ganhos de 34 dB e 36 dB par os canais de recepção e de transmissão, respectivamente.

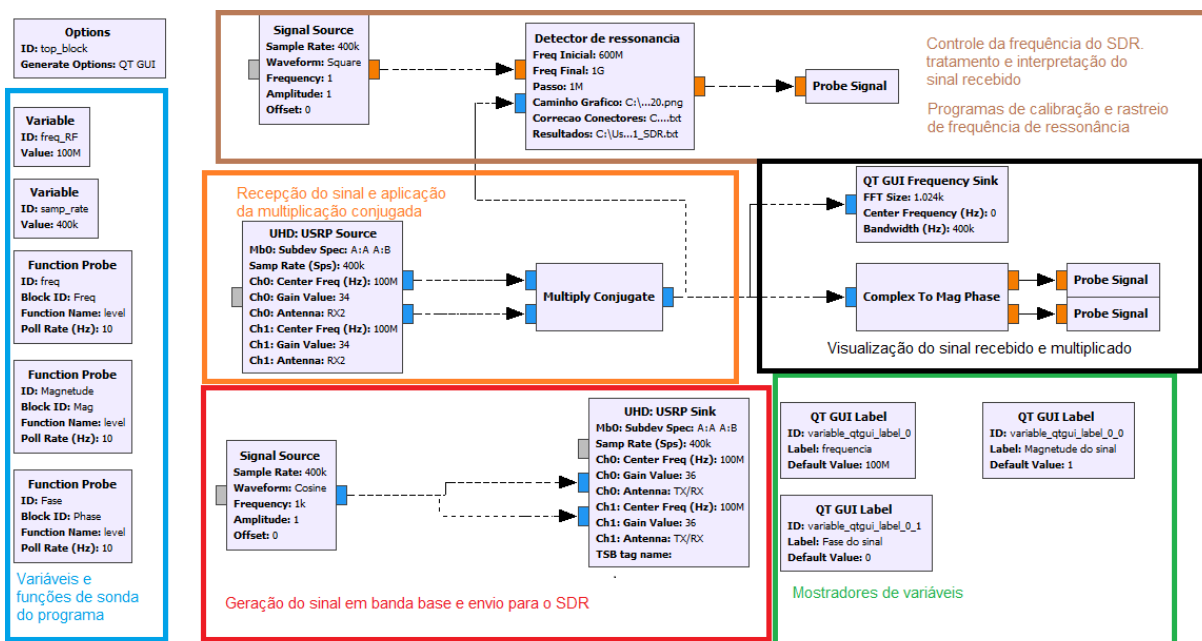
Fixados os parâmetros do programa, o diagrama de blocos montado para os processo de registro de *offset*, caracterização do sensor e rastreo de frequência de ressonância estão expostos nas figuras 37 e 38. Em todas as montagens, a principal mudança ocorre no bloco de programação customizada em Python.

Figura 37 – Diagrama de blocos para o processo de registro de diferença de conectores implementado no GNURadio.



Fonte: O autor

Figura 38 – Diagrama de blocos para o processo de caracterização do sensor e de rastreamento da frequência de ressonância implementado no GNURadio.



Fonte: O autor

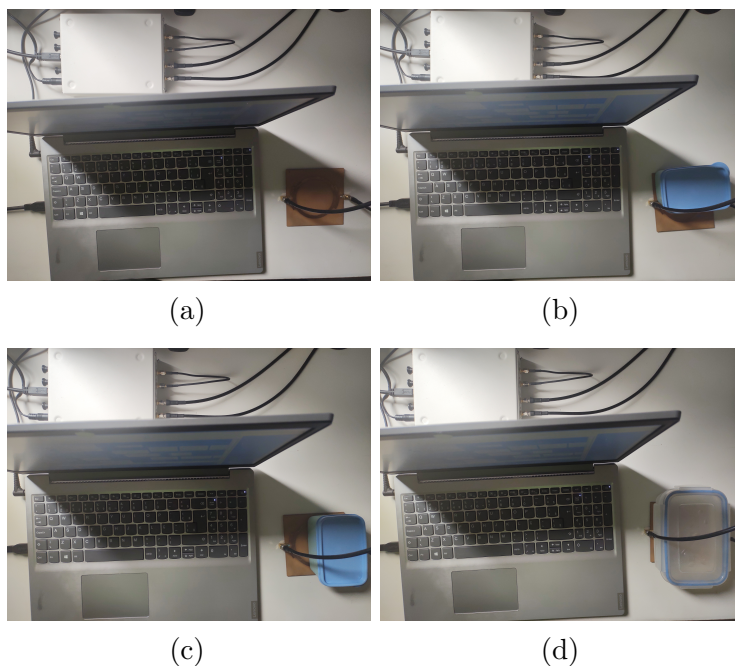
Figura 39 – Montagem para a Registro de *offset* do sistema do SDR

Fonte: O autor

Quanto a configuração física da bancada, ela foi montada em uma mesa de escritório com um computador portátil e a placa de SDR USRP B210. No cenário de registro de *offset*, os cabos foram acoplados por meio do adaptador SMA, como mostra a figura 39.

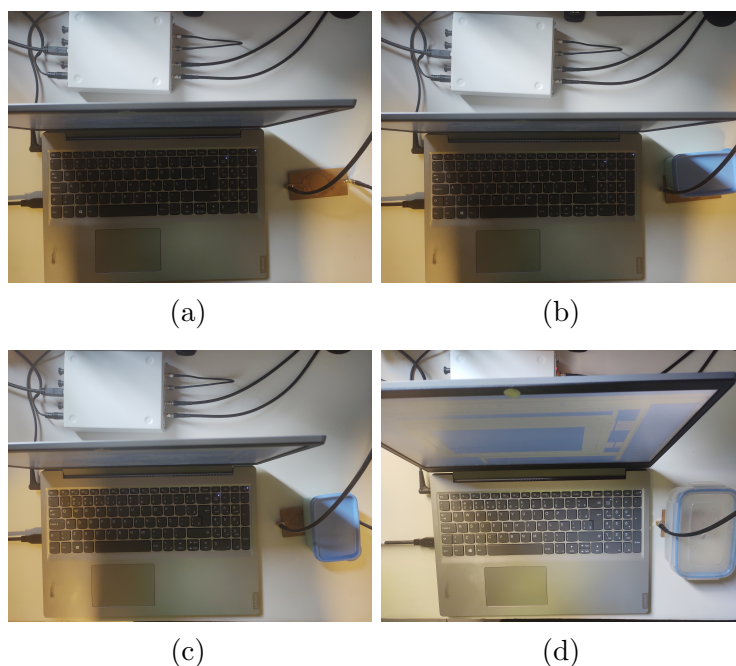
Os cenários do sensor descoberto, parcialmente paralelo a LT e perpendicular a LT e completamente coberto foram montados na bancada de forma a acomodar a disposição do sensor. As figuras 40 e 41 mostram configurações da bancada de ensaios.

Figura 40 – Configurações descoberta (a), parcial paralela a LT (b), parcial perpendicular a LT (c) e totalmente coberta (d) para a caracterização do sensor A, respectivamente.



Fonte: O autor

Figura 41 – Configurações descoberta (a), parcial paralela a LT (b), parcial perpendicular a LT (c) e totalmente coberta (d) para a caracterização do sensor B, respectivamente.



Fonte: O autor

### 3.2.8 Validação dos resultados Obtidos pela bancada de ensaios

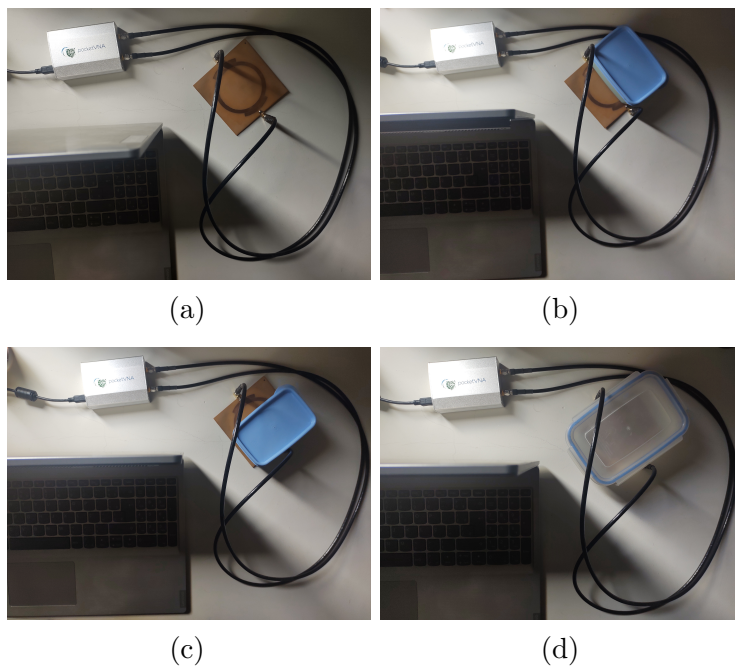
Os resultados da bancada de ensaios precisam ser validados utilizando um outro dispositivo que possa caracterizar o sensor e que já tenha sido certificado. Nesse caso, será um Analisador de Rede Vetorial (*Vector Network Analyser*, VNA) para aferir os parâmetros de espalhamento do sensor ( $S_{11}$ ,  $S_{12}$ ,  $S_{21}$  e  $S_{22}$ ) e formar um gráfico com relação a frequência. O VNA a ser utilizado é o Pocket VNA (POCKETVNA, 2021a), um equipamento com interface para o computador via USB, 2 portas, bidirecional e com faixa de frequência de 500 kHz até 4 GHz.

O VNA, com auxílio do programa fornecido pelo fabricante (POCKETVNA, 2021b), foi ajustado com a opção *2-port full SOLT* com velocidade  $2\times$  *slow* e nas medições foi utilizado a velocidade normal de medição, sendo coletados todos parâmetros de espalhamento, mas impressos em imagem somente o coeficiente de transmissão ( $S_{21}$ ) e as cartas de Smith dos sensores, que utiliza o coeficiente de reflexão ( $S_{11}$ ). O ajuste foi realizado com um adaptador SMA e ambos os cabos utilizados para conectar o sensor, e foi considerada uma extensão de conexão de 10 cm, o mesmo comprimento do sensor.

Da mesma forma que caracterizou-se os sensores com 4 configurações de inserção da água, montou-se as mesmas situações para a obtenção das curvas. As figuras 42 e 43 mostram as 4 configurações de medição para os sensores de A e B, respectivamente.

Com as curvas de espalhamento do sensor, estimou-se a sua resposta em frequência, contendo a ressonância e a largura de banda do sensor, com auxílio do software GNU

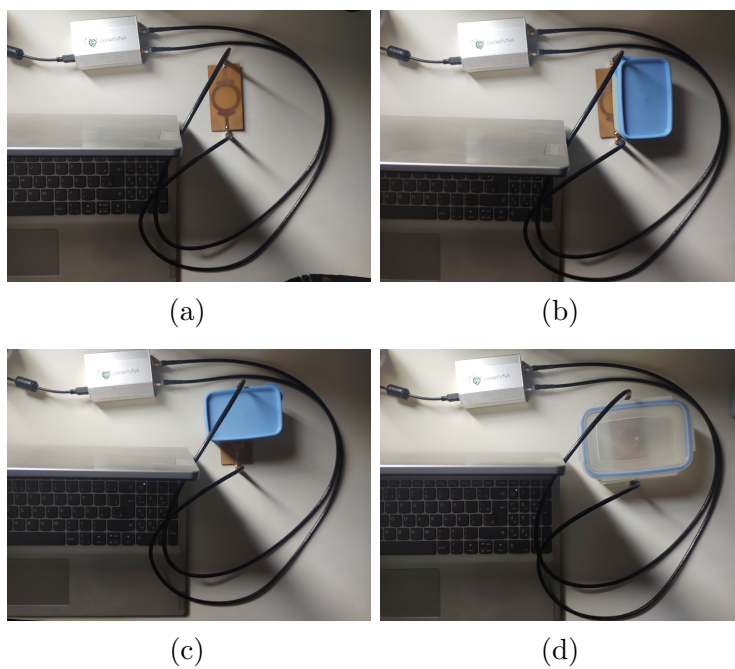
Figura 42 – Configuração 1 (a), 2 (b), 3 (c) e 4 (d) para a obtenção das curvas do sensor A utilizando o VNA, respectivamente.



Fonte: O autor

Octave. Sucedido esse procedimento, compara-se os resultados obtidos pelo VNA com os da bancada de ensaios.

Figura 43 – Configuração 1 (a), 2 (b), 3 (c) e 4 (d) para a obtenção das curvas do sensor B utilizando o VNA, respectivamente.



Fonte: O autor

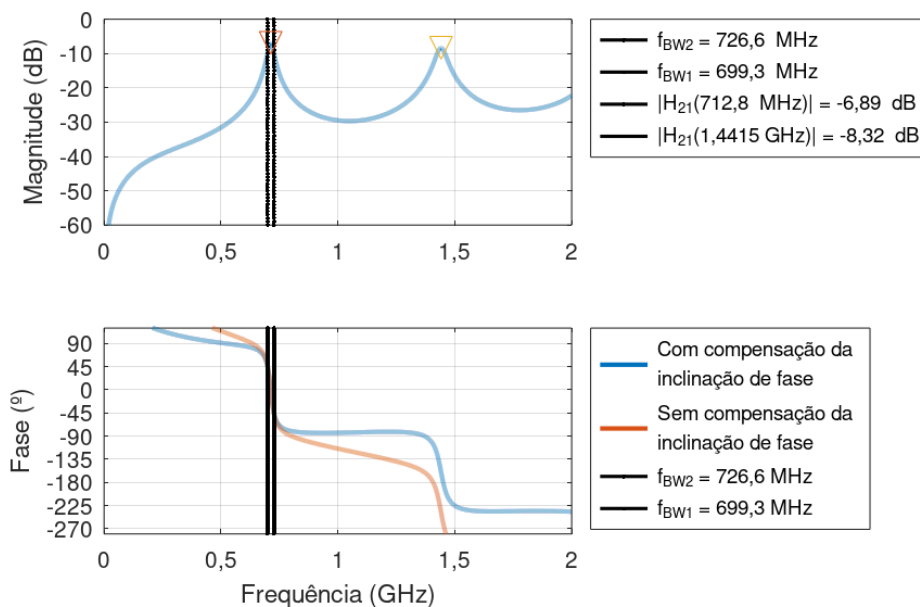
## 4 RESULTADOS E DISCUSSÕES

Nesta seção serão discutidos os resultados gráficos e textuais obtidos em cada rotina executada no GNU Radio 3.7, em conjunto com a plataforma SDR, e pelo VNA Pocket. Sempre que mencionado nos resultados, será considerada a ordem das configurações 1, 2, 3 e 4 definidas como sensor descoberto, sensor parcialmente coberto paralelo à LT 34a, perpendicularmente 34b e completamente coberto, respectivamente, na ordem indicada nas figuras 42 e 43 no capítulo 3.

### 4.1 SIMULAÇÕES DO ANSYS

Das simulações eletromagnéticas realizadas no Ansys HFSS, obtemos os gráficos de ganho de transmissão ( $S_{21}$ ) das figuras 44 e 45 para os sensores A e B na configuração descoberta, respectivamente. Já a figura 46 mostra a carta de Smith obtida através do Ansys com a variação de frequência. As configurações 2, 3 e 4 não foram simuladas.

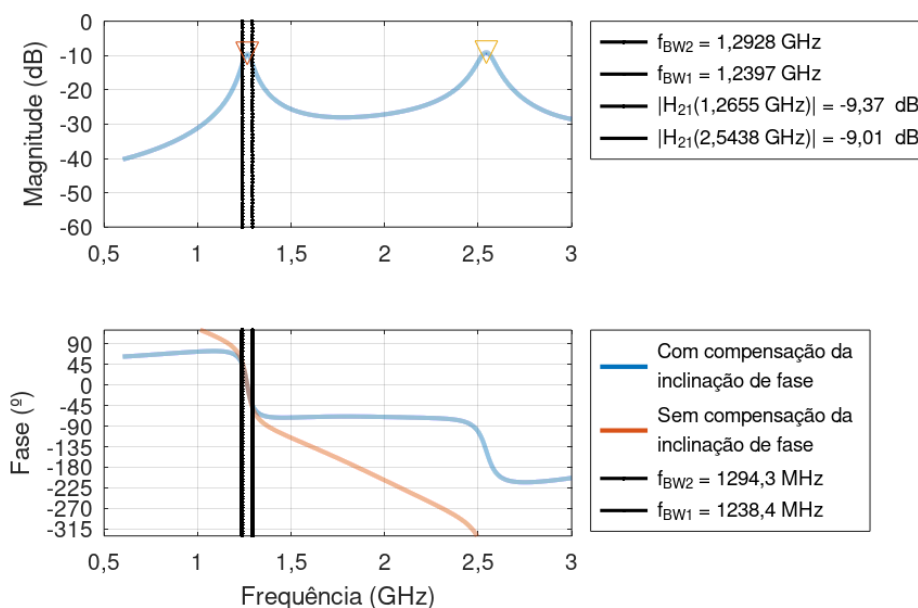
Figura 44 – Resposta em frequência de módulo, fase (do parâmetro  $S_{21}$ ) com e sem a compensação da inclinação da fase do sensor A no cenário descoberto obtidas a partir de simulação eletromagnética.



Fonte: O autor

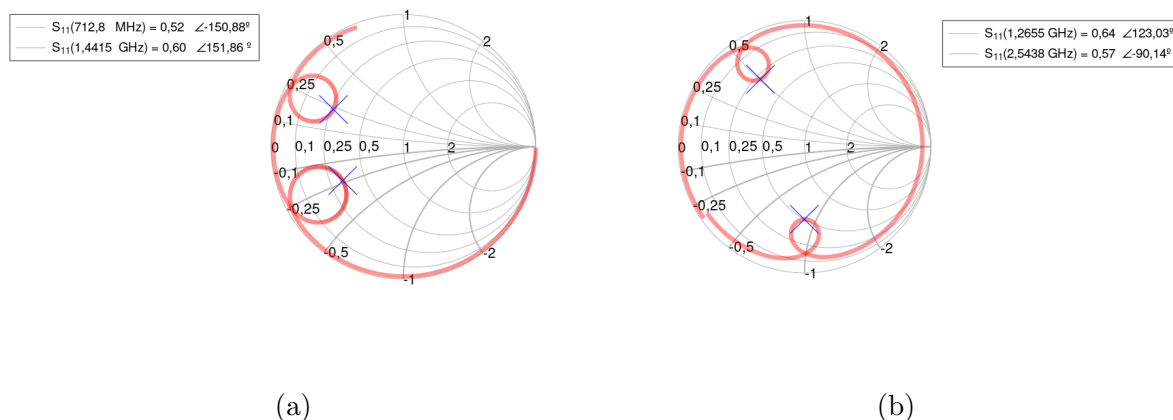
A curva vermelha escura da figura 44 mostra o efeito da LT sobre a fase do sinal, já a curva azul mostra a curva com a compensação deste efeito. A compensação foi realizada com auxílio de marcadores para a determinação da inclinação por frequência na região entre a fundamental e a segunda frequência de ressonância. As cartas de Smith mostram uma característica circular das curvas  $S_{11}$  e  $S_{21}$ , sendo que  $S_{22}$  e  $S_{12}$  não foram imprimidas

Figura 45 – Resposta em frequência de módulo e fase (do parâmetro  $S_{21}$ ) do sensor B e no cenário descoberto obtidas a partir de simulação eletromagnética.



Fonte: O autor

Figura 46 – Coeficiente de reflexão ( $S_{11}$ ) nas cartas de Smith dos sensores A (a) e B (b) obtidas a partir de simulação eletromagnética.



Fonte: O autor

devido a ao sensor ser simétrico. A curva externa mostra o coeficiente de reflexão ( $S_{11}$ ) enquanto a interna mostra o de transmissão ( $S_{21}$ ).

Os resultados dos gráficos estão resumidos na tabela 5. Os marcadores encontrados nas figuras 44 e 45 foram utilizados para extrair os dados. A frequência de ressonância esteve em um valor abaixo do valor teórico em ambos os casos.

Tabela 5 – Resultados da frequência de ressonância, fator de qualidade, largura de banda e frequências de meia potência para os sensores A e B na configuração descoberta obtidos através da simulação do ANSYS.

Sensor	$f_{ress}$ (MHz)	$Q$	$BW$ (MHz)	$f_2$ (MHz)	$f_1$ (MHz)
A	718,8	26,11	27,3	726,6	699,3
B	1265,5	23,83	53,1	1294,3	1238,4

Fonte: O autor

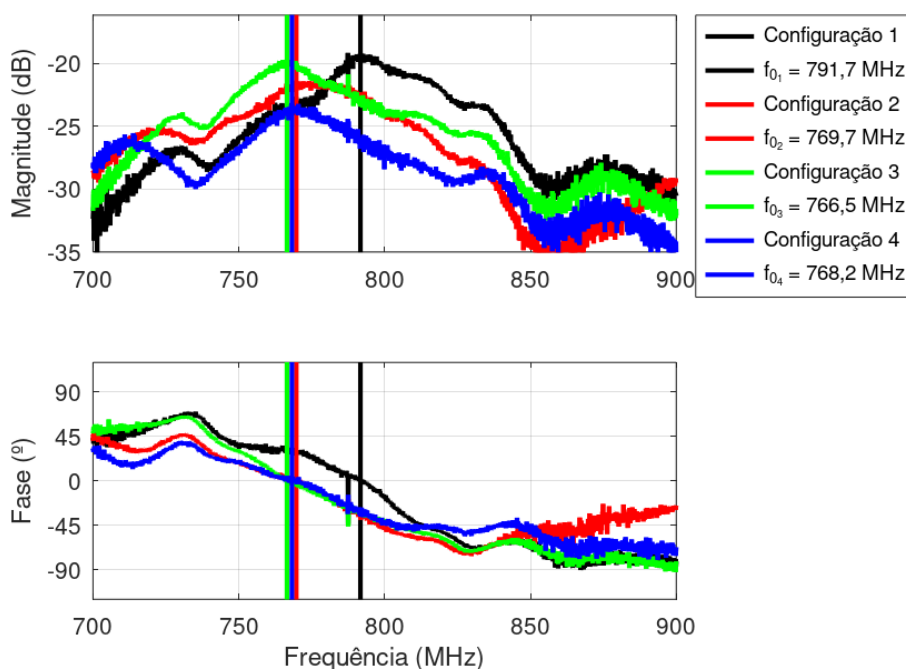
## 4.2 CARACTERIZAÇÃO DO SENSOR ATRAVÉS DO VNA

Utilizando o procedimento descrito na seção 3.2.8, os resultados gráficos obtidos através do VNA são apresentados nessa seção. Em todos os gráficos, o valor da fase foi corrigido para ser 0 na ressonância.

### 4.2.1 Sensor A

As imagens gráficas extraídas através do VNA e plotadas pelo GNU Octave para as configurações 1, 2, 3 e 4 do sensor de A estão mostradas na figura 47. Foram destacados os valores da frequência de ressonância, onde o máximo do ganho de transmissão ocorre para cada caso.

Figura 47 – Resposta em frequência de módulo e fase (do parâmetro  $S_{21}$ ) do sensor de A na configuração 1, 2, 3 e 4 extraído a partir do VNA.

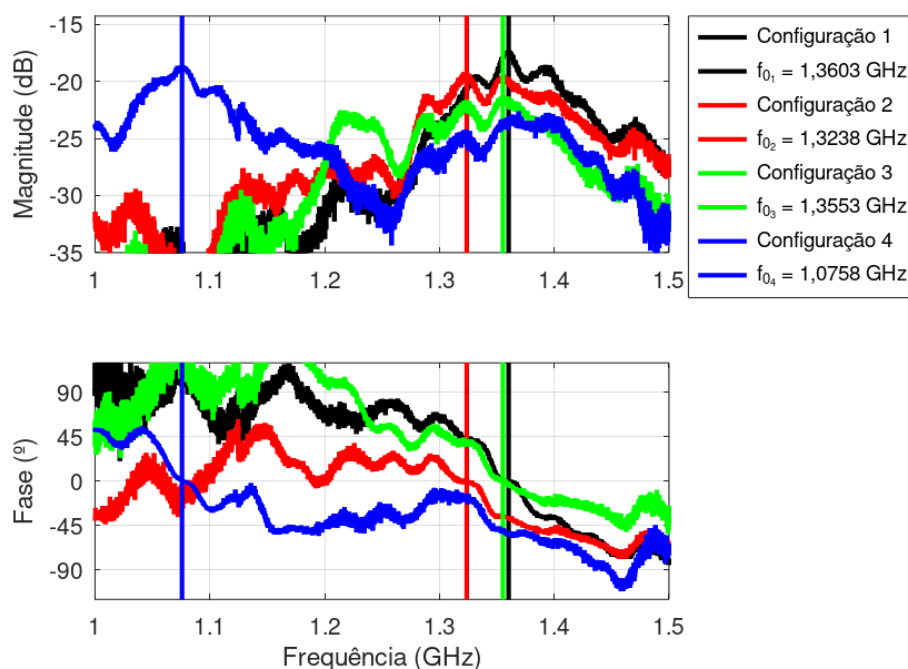


Fonte: O autor

### 4.2.2 Sensor B

As imagens gráficas extraídas através do VNA e plotadas pelo GNU Octave para as configurações 1, 2, 3 e 4 do sensor de B estão expostas na figura 48.

Figura 48 – Resposta em frequência de módulo e fase (do parâmetro  $S_{21}$ ) do sensor de B na configuração 1, 2, 3 e 4.



Fonte: O autor

### 4.2.3 Análise dos gráficos das características de amplitude e fase junto com a caracterização do sensor

Os resultados obtidos a partir do VNA nas seções 4.2.1 e 4.2.2, trazem algumas características próprias do sensor planar em formato de anel. Os dados de ambos os sensores foram analisados através do GNU Octave e compilados nas tabelas 6, 7, 8 e 9.

Tabela 6 – Resultados da frequência de ressonância, fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor A obtidos através do VNA e analisados pelo GNU Octave, pela amplitude.

Configuração	$f_{ress}$ (MHz)	$Q_{mag}$	$BW_{mag}$ (MHz)	$f_{2mag}$ (MHz)	$f_{1mag}$ (MHz)	$ S_{21} _{max}$ (dB)
Descoberto (ar)	791,7	19,79	40,0	819,9	779,9	-19,19
Paralelo a LT	769,7	15,68	49,1	800,8	751,7	-21,42
Perpendicular a LT	765,5	19,02	40,3	789,3	749,0	-19,72
Coberto (água destilada)	768,2	22,73	33,8	787,5	753,7	-23,21

Fonte: O autor

Tabela 7 – Resultados da obtenção fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor A obtidos através do VNA e analisados pelo GNU Octave, pela fase.

Configuração	$Q_{fase}$	$BW_{fase}$ (MHz)	$f_{2fase}$ (MHz)	$f_{1fase}$ (MHz)	$ S_{11} _{ress}$
Descoberto (ar)	11,59	68,3	812,4	744,1	0,74
Paralelo a LT	12,06	63,8	797,8	734,0	0,88
Perpendicular a LT	16,31	47,0	787,5	740,5	0,87
Coberto (água destilada)	7,31	105,1	799,2	694,1	0,87

Fonte: O autor

Tabela 8 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor B obtidos através do VNA e analisados pelo GNU Octave.

Configuração	$f_{ress}$ (GHz)	$Q_{mag}$	$BW_{mag}$ (MHz)	$f_{2mag}$ (GHz)	$f_{1mag}$ (GHz)	$ S_{21} _{max}$ (dB)
Descoberto (ar)	1,3603	24,42	55,7	1,403	1,347	-17,29
Paralelo a LT	1,3238	14,22	93,1	1,399	1,306	-19,18
Perpendicular a LT	1,3553	31,01	43,7	1,384	1,341	-21,24
Coberto (água destilada)	1,0758	15,28	70,4	1,112	1,041	-18,61

Fonte: O autor

Tabela 9 – Resultados da obtenção fator de qualidade, largura de banda e frequências de meia potência para cada configuração usando o sensor B obtidos através do VNA e analisados pelo GNU Octave, pela fase.

Configuração	$Q_{fase}$	$BW_{fase}$ (MHz)	$f_{2fase}$ (GHz)	$f_{1fase}$ (GHz)	$ S_{11} _{ress}$
Descoberto (ar)	16,94	80,3	1,407	1,327	1,05
Paralelo a LT	7,94	166,7	1,324	1,157	0,97
Perpendicular a LT	31,01	43,7	1,355	1,312	0,96
Coberto (água destilada)	42,52	25,3	1,076	1,051	0,93

Fonte: O autor

Uma comparação entre os resultados do Ansys e do VNA, para os valores do coeficiente de transmissão, de reflexão, da frequência de ressonância e do fator de qualidade obtido pela magnitude e pela fase, se encontram nas tabelas 10 e 11, para o sensor A e B, respectivamente.

Comparando os resultados entre o Ansys e o VNA, a medição obteve uma resposta da frequência de ressonância simulada. Algumas possíveis explicações podem ser consideradas, tais como o a permissividade real da placa ser menor que a de projeto (assumida em 4,4) ou o fato da área do plano terra ser próxima a do anel, alterando o valor da permissividade efetiva da micro-fita.

A amplitude dos sensores com relação à frequência forma diversos picos e vales locais, sendo que os primeiros apresentam a maior amplitude nos múltiplos da frequência de ressonância fundamental do anel. O módulo do coeficiente transmissão ( $S_{21}$ ) medido na

Tabela 10 – Comparação entre os resultados do simulados pelo Ansys e medidos pelo VNA do sensor A no cenário descoberto.

Método	$ S_{21} _{ress}$	$ S_{11} _{ress}$	$f_{ress}$ (MHz)	$Q_{mag}$	$Q_{fase}$
Simulado					
Ansys	-6,89 dB	0,52	712,8	26,11	26,11
VNA	-19,19 dB	0,74	791,7	19,79	11,59
$ e(\%) $	75,73	42,3	11,07	24,21	55,61

Fonte: O autor

Tabela 11 – Comparação entre os resultados do simulados pelo Ansys e medidos pelo VNA do sensor B no cenário descoberto.

Método	$ S_{21} _{ress}$	$ S_{11} _{ress}$	$f_{ress}$ (GHz)	$Q_{mag}$	$Q_{fase}$
Simulado					
Ansys	-9,37 dB	0,54	1,2655	23,83	22,64
VNA	-17,29 dB	1,05	1,3603	24,42	16,94
$ e(\%) $	59,82	94,44	7,49	2,48	25,18

Fonte: O autor

ressonância possui um valor menor que a metade do simulado. Uma das explicações dessa queda é que não foram modeladas todas as perdas da placa real na simulação do ANSYS.

O coeficiente de reflexão ( $S_{11}$ ) medido foi superior ao simulado em ambos os sensores. Para o sensor B, o módulo desse parâmetro também apresentou um valor superior ao máximo possível dos parâmetros de espalhamento, sendo o limite do módulo em 1. Esse resultado indica que o VNA apresenta erros residuais de seus ajustes.

A fase do sensor possui um comportamento similar ao de um filtro passa faixa, com a fase realizando um deslocamento em sentido horário no plano complexo a cada múltiplo da frequência fundamental do anel. O comportamento ondulatório observado na resposta de magnitude e de ângulo medidos podem derivar das reflexões, causadas pela falta de casamento de impedâncias, além dos erros residuais do ajuste do VNA.

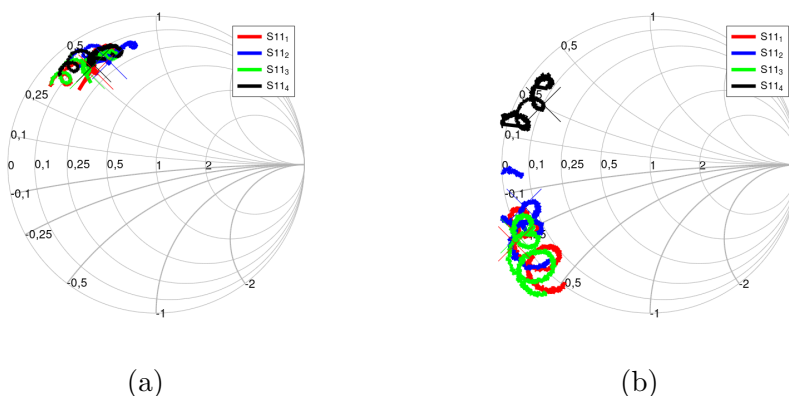
O fator de qualidade medido se manteve em uma ordem de grandeza similar ao simulado. Este resultado indica que a relação entre a parte real e a parte imaginária da permissividade efetiva complexa estão em um valor próximo ao da simulação.

As cartas de Smith para os sensores A e B obtidas pelo VNA estão na figura 49. Percebe-se o comportamento ondulatório da carta, similar ao das respostas de amplitude e fase, devido a reflexões e a imperfeições nas trilhas. Foi utilizado o parâmetro  $S_{11}$  para a geração das cartas de Smith.

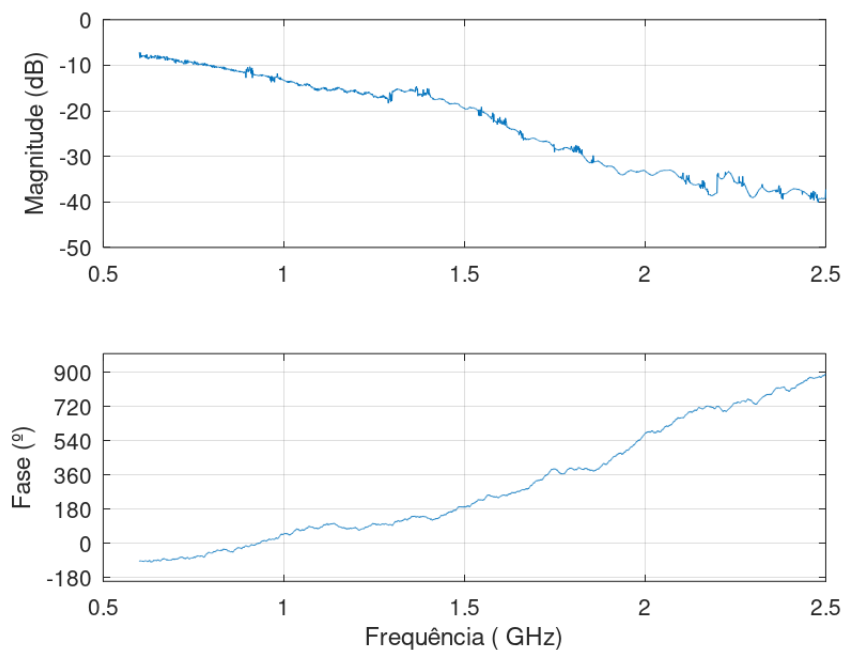
### 4.3 Caracterização do sensor através do GNU Radio

Nesta seção, serão apresentados os resultados das rotinas implementadas na bancada de ensaios. As rotinas foram simuladas conforme mostra o apêndice C.

Figura 49 – Cartas de Smith dos sensor A (a) e do sensor B (b) pelo VNA.



Fonte: O autor

Figura 50 – Figura gerada pela rotina de registro de *offset* do GNU Radio.

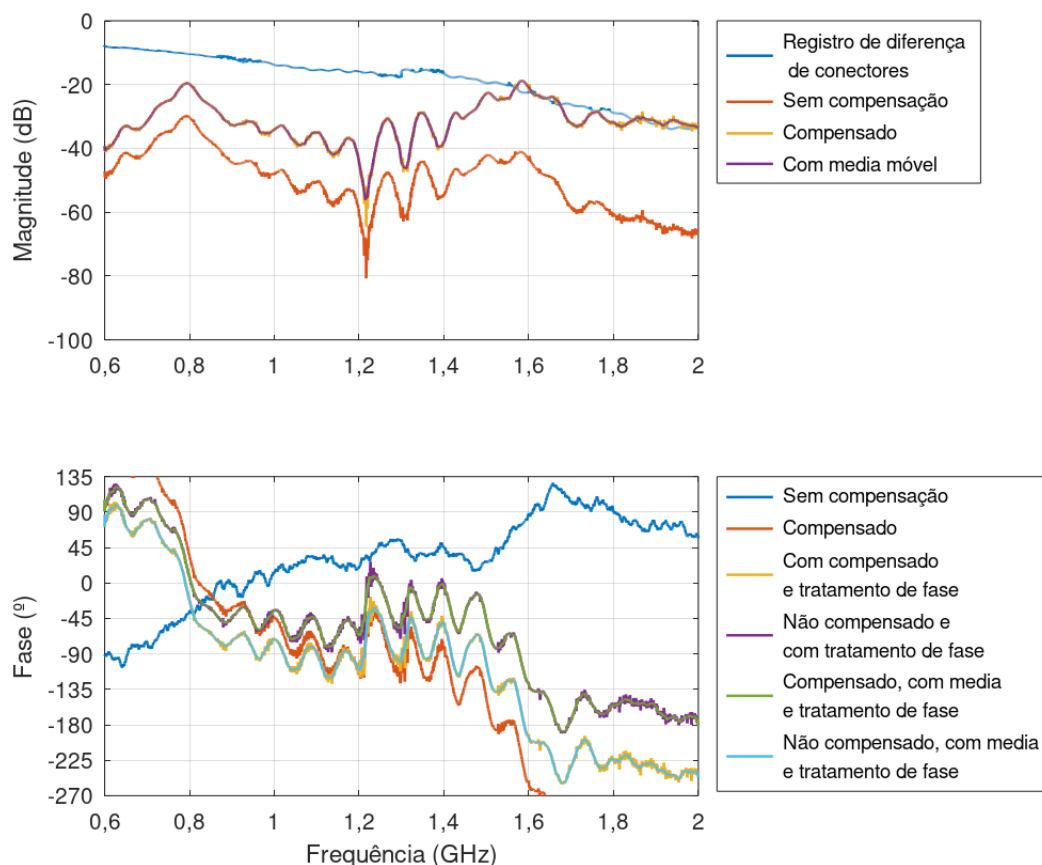
Fonte: O autor

#### 4.3.1 Rotina de registro de *offset*

Utilizando o diagrama da figura 37, rotina de registro de *offset* do GNU Radio obteve o resultado mostrado na figura 50. A curva foi extraída após a multiplicação conjugada dos sinais recebidos dos canais TX0-RX0 e TX1-RX1. A amplitude possui influência dos ganhos de ambos os canais e a fase mostrada é a diferença de fase entre os conectores.

Para ilustrar os efeitos dos dados da rotina de registro de *offset* nas medições, foram levantados os dados de medições do sensor A e B nas figuras 51 e 52. Uma delas mostra a medição sem o uso dos dados, enquanto a outra mostra o uso deles.

Figura 51 – Resultado de uma medição com e sem o ajuste de *offset* e os algoritmos de compensação para o sensor A na configuração descoberta.



Fonte: O autor

### 4.3.2 Medição pelo GNU Radio e o SDR

Para fins de visualização, obteve-se uma captura de tela no início e no fim da rotina de caracterização do sensor. As figuras 53 e 54, mostram os respectivos momentos. Os resultados foram copiados a partir do terminal.

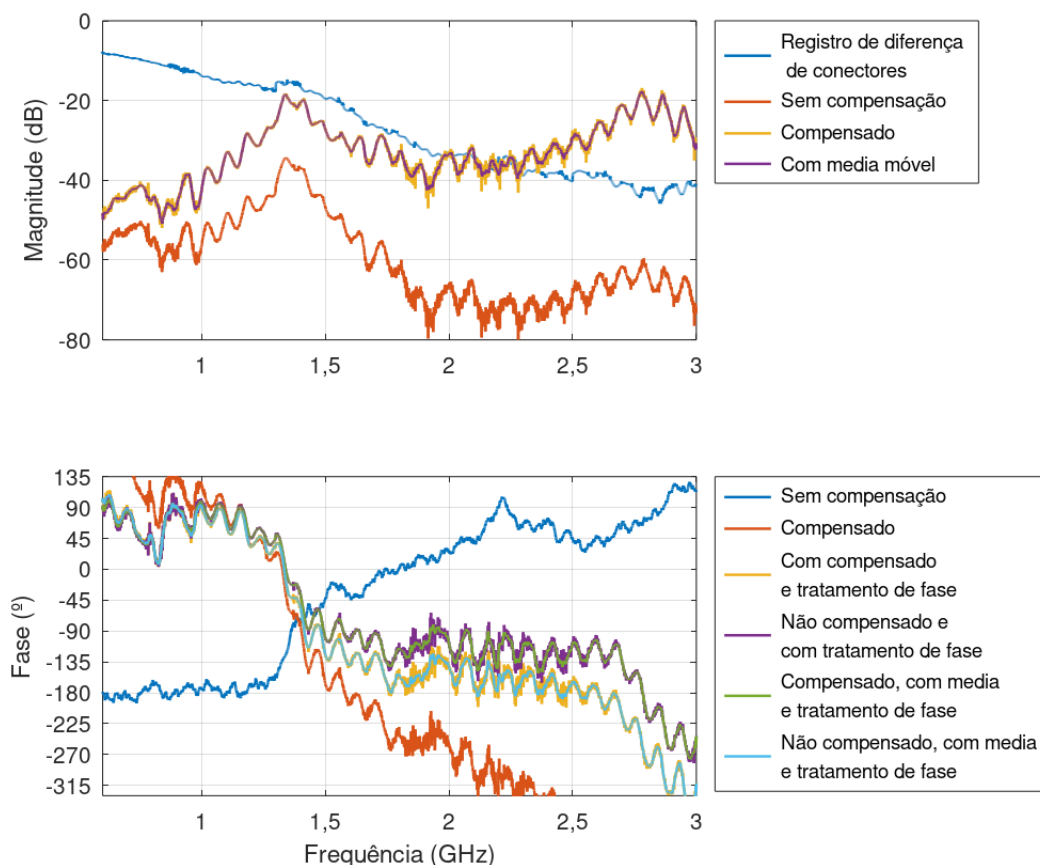
#### 4.3.2.1 Sensor A

Utilizando a plataforma GNU Radio, junto com o bloco de programação customizada em Python, foram obtidas as respostas em frequência do sensor A para as configurações 1, 2, 3 e 4 e mostradas na figura 55.

#### 4.3.2.2 Sensor B

Utilizando a plataforma GNU Radio, junto com o bloco de programação em Python, foram obtidas as respostas em frequência do sensor B para as configurações 1, 2, 3 e 4 e mostradas na figura 56. A resposta do sensor descoberto em uma largura de banda

Figura 52 – Resultado de uma medição com e sem o ajuste de *offset* e os algoritmos de compensação para o sensor B na configuração descoberta.



Fonte: O autor

até a sua segunda ressonância é mostrada na figura 52. Percebe-se um grande efeito nos resultados após a frequência de 1,5 GHz, quando a magnitude do registro de *offset* reduz significativamente.

O cenário 4 do sensor B apresentou a maior variação da frequência de ressonância. Enquanto os outros cenários apresentaram valores na região de 1,3 GHz, a configuração totalmente coberta chegou até 1,1 GHz.

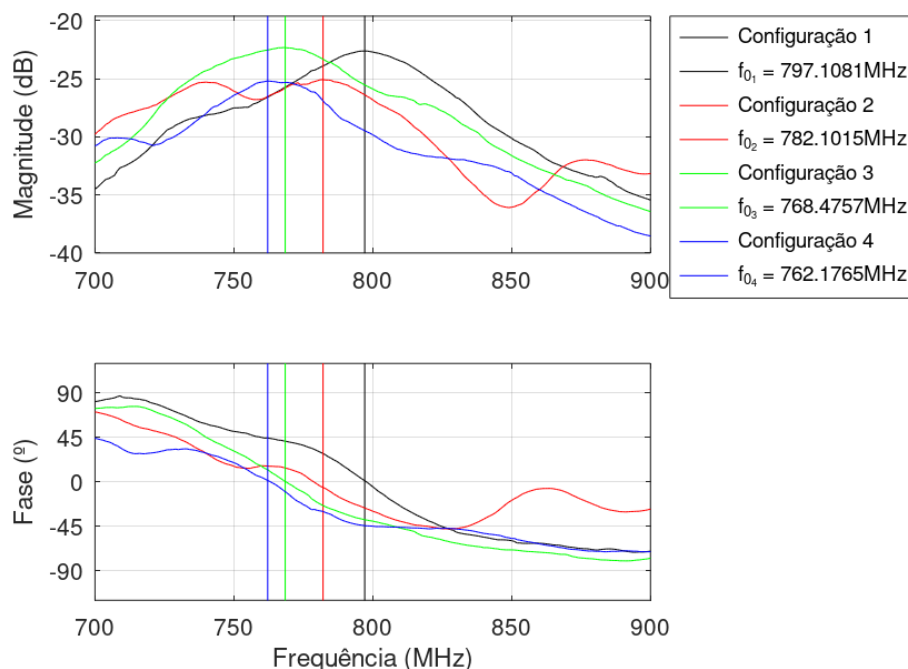
### 4.3.3 Análise dos resultados

As medidas obtidas da resposta em frequência dos sensores através do SDR e analisadas pelo GNU Radio estão compilados nas tabelas 12 e 13 para o sensor A e 14 e 15 para o sensor B. Todos os valores de ângulo foram ajustados para apresentarem o valor zero na ressonância e facilitar a visualização dos resultados. Dos dados do VNA, o ajuste ocorreu no Octave, enquanto que no SDR, a rotina realizou a correção.

A inserção da água torna o comportamento da fase irregular, principalmente quando comparado ao filtro ideal que o representa, tornando a função de cálculo e correção



Figura 55 – Resposta em frequência de módulo e fase do sensor de A na configuração 1, 2, 3 e 4 extraído a partir do GNU Radio.



Fonte: O autor

da inclinação de fase ineficiente. Para a devida correção nos cenários 2, 3 e 4, utilizou-se o valor pré-calculado da inclinação de fase do sistema de  $-2,518 \times 10^{-9}$  ou  $-6,35 \times 10^{-8}$ , caso o ajuste do *offset* fosse aplicado na fase ou não.

Tabela 12 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor A através do GNU Radio e o SDR.

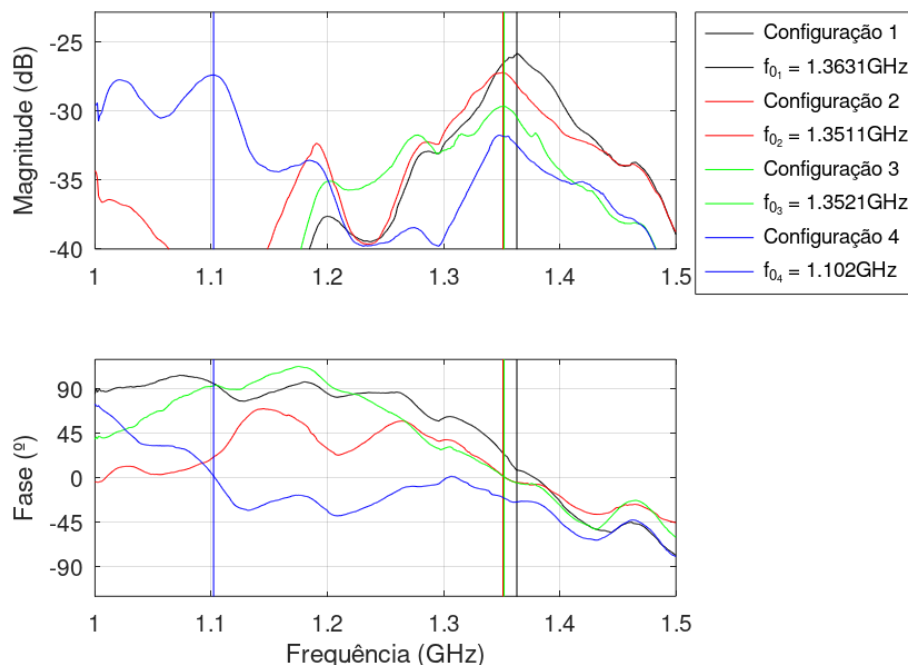
Configuração	$f_{ress}$ (MHz)	$Q_{mag}$	$BW_{mag}$ (MHz)	$f_{2mag}$ (MHz)	$f_{1mag}$ (MHz)	$ S_{21} _{max}$ (dB)
Descoberto (ar)	791,9	17,07	46,4	818,3	771,9	-20,70
Paralelo a LT	788,1	20,55	38,4	812,5	774,2	-23,92
Perpendicular a LT	775,3	12,94	59,9	799,4	739,5	-21,03
Coberto (água destilada)	768,1	16,70	46,0	792,1	746,2	-24,81

Fonte: O autor

#### 4.4 Comparação dos resultados entre o SDR e o VNA

Aqui serão discutidas as diferenças de resultados provenientes da caracterização com o SDR e o VNA. Os erros entre parâmetros de magnitude na ressonância, frequência de ressonância, fator de qualidade e largura de banda pela magnitude e fase para os métodos de medição, utilizando os dados do VNA como referência, estão compilados nas tabelas 16 e 17 para os sensores A e B, respectivamente.

Figura 56 – Resposta em frequência de módulo e fase do sensor B na configuração 1, 2, 3 e 4 extraído a partir do GNU Radio.



Fonte: O autor

Tabela 13 – Resultados da largura de banda e fator de qualidade do sensor A obtidos através do GNU Radio e o SDR pela leitura de fase.

Configuração	$Q_{fase}$	$BW_{fase}$ (MHz)	$f_{2fase}$ (MHz)	$f_{1fase}$ (MHz)
Descoberto (ar)	16,98	46,7	814,9	768,3
Paralelo a LT	9,01	87,5	821,4	733,9
Perpendicular a LT	14,02	55,3	798,4	743,1
Coberto (água destilada)	8,34	92,1	798,0	705,9

Fonte: O autor

Os resultados de frequência de ressonância e de fator de qualidade do SDR e do VNA obtiveram um erro menor que 5% em todos os cenários. Para o sensor A o VNA apresentou uma resposta que indicava a presença da água sob sensor ou não, visto que a frequência de ressonância para as configurações 2, 3 e 4 estarem próximas, enquanto que o SDR apresentou uma resposta de acordo com a quantidade da área propriamente coberta do sensor, representada em uma redução gradual da frequência de ressonância.

Para o sensor B, o VNA detectou alterações do fator de qualidade conforme a inserção de água das configurações, tendo alterado a ressonância significativamente somente na inserção completa. As medições através SDR obtiveram um fator de qualidade mais estável entre os cenários, para a medição de amplitude.

As figuras 57 e 58, mostram as sobreposição das respostas em frequência dos sensores A e B para cada uma das configurações. De forma geral, o gráfico obtido através

Tabela 14 – Resultados da frequência de ressonância e fator de qualidade para cada configuração usando o sensor B através do GNU Radio e o SDR.

Configuração	$f_{ress}$ (GHz)	$Q_{mag}$	$BW_{mag}$ (MHz)	$f_{2mag}$ (GHz)	$f_{1mag}$ (GHz)	$ S_{21} _{max}$ (dB)
Descoberto (ar)	1,337	26,22	51,0	1,371	1,320	-18,72
Paralelo a LT	1,336	40,24	33,2	1,354	1,320	-18,93
Perpendicular a LT	1,332	42,71	31,2	1,348	1,317	-20,16
Coberto (água destilada)	1,094	41,45	26,4	1,106	1,080	-17,98

Fonte: O autor

Tabela 15 – Resultados da largura de banda e fator de qualidade do sensor B obtidos através do GNU Radio e o SDR pela leitura de fase.

Configuração	$Q_{fase}$	$BW_{fase}$ (MHz)	$f_{2fase}$ (GHz)	$f_{1fase}$ (GHz)
Descoberto (ar)	17,78	75,2	1,395	1,320
Paralelo a LT	5,63	237,2	1,424	1,186
Perpendicular a LT	9,20	144,8	1,411	1,266
Coberto (água destilada)	6,98	156,6	1,186	1,030

Fonte: O autor

Tabela 16 – Erros entre os resultados medidos pelo VNA e o SDR do sensor A.

Cenário	$ S_{21} _{ress}$ (%)	$f_{ress}$ (%)	$Q_{mag}$ (%)	$BW_{mag}$ (%)	$Q_{fase}$ (%)	$BW_{fase}$ (%)
Descoberto	15,96	0,03	13,74	16,00	46,51	31,63
Paralelo a LT	25,01	2,39	31,06	21,79	25,29	37,15
Perpendicular a LT	14,00	1,28	31,97	48,64	14,04	17,66
Coberto (água destilada)	16,82	0,01	26,53	36,09	14,09	12,37

Fonte: O autor

do SDR possui curvas de formatos similares com as obtidas a partir do VNA. O SDR apresentou menores ondulações em sua resposta em geral. Isso se deve porque o SDR pode enviar um sinal com potência maior que o VNA e também devido a utilização das funções de redução de ruído no SDR.

Para o sensor A, o cenário 2 apresentou uma variação de 6 dB entre as medidas do VNA e o SDR. No sensor B, os cenários 2 e 3 apresentaram uma maior diferença entre as medições, tendo regiões com 10 dB de amplitude do SDR abaixo do VNA.

#### 4.5 Rastreamento de ressonância

A rotina de rastreamento de ressonância foi testada com o sensor A nas configurações descoberta e totalmente coberta. O banco rastreou a ressonância em até 10 segundos após a alteração. Apesar do funcionamento correto na simulação, a execução em bancada nem sempre detecta corretamente quando ocorre a alteração em frequência, dando falsas alterações, ou em qual direção se encontra a ressonância, tendo uma melhor eficiência

Tabela 17 – Erros entre os resultados medidos pelo VNA e o SDR do sensor B.

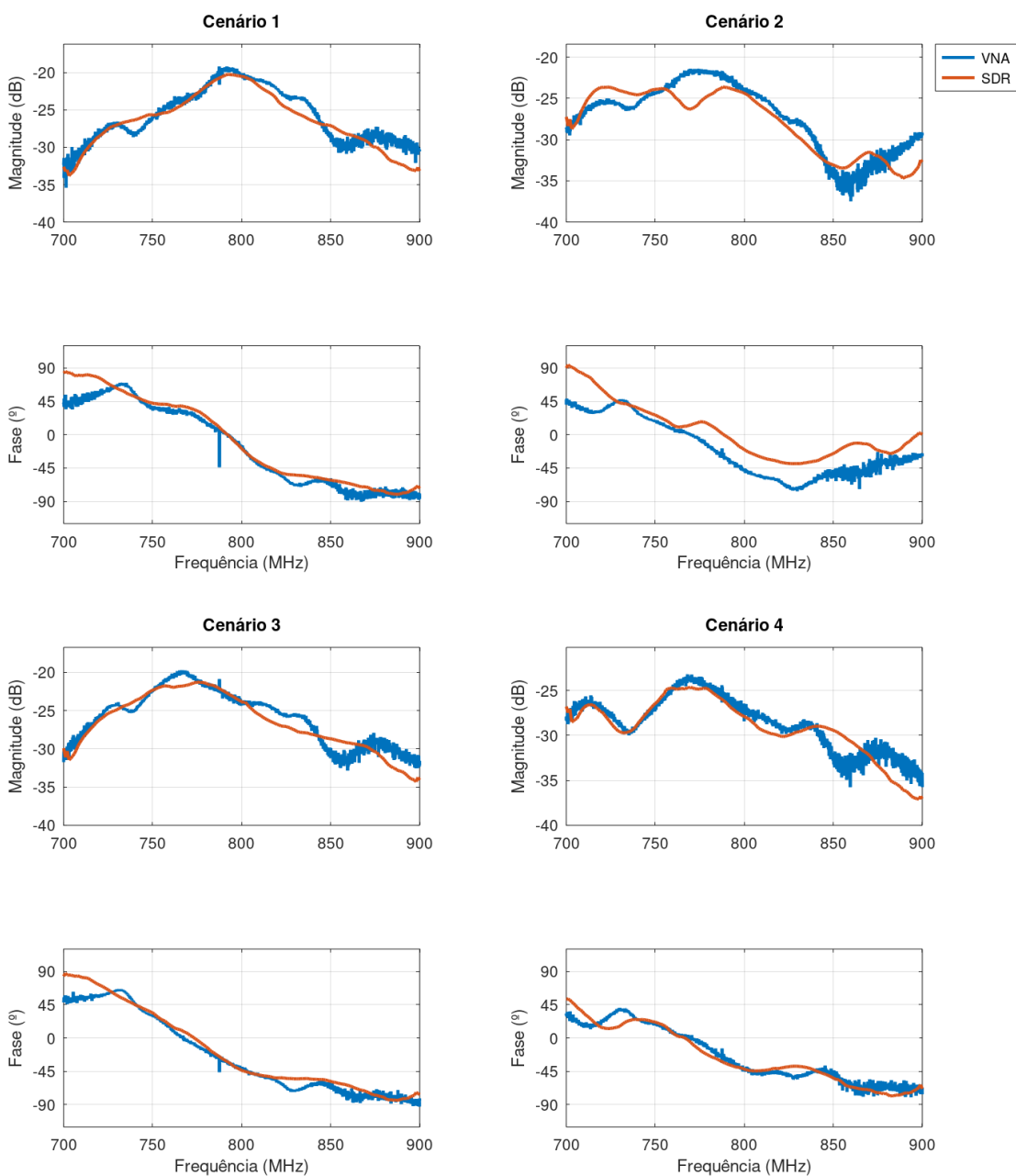
Cenário	$ S_{21} _{ress}$ (%)	$f_{ress}$ (%)	$Q_{mag}$ (%)	$BW_{mag}$ (%)	$Q_{fase}$ (%)	$BW_{fase}$ (%)
Descoberto	15,18	1,71	7,37	8,44	4,96	6,35
Paralelo a LT	2,92	0,92	182,98	64,34	29,09	42,29
Perpendicular a LT	13,24	1,72	37,73	28,60	70,33	231,35
Coberto (água destilada)	7,52	1,69	171,27	62,50	83,58	518,97

Fonte: O autor

quando a inserção do material é realizada o mais rápido possível. Uma explicação para isso seriam picos ou flutuações magnitude ou na fase que geram essas leituras.

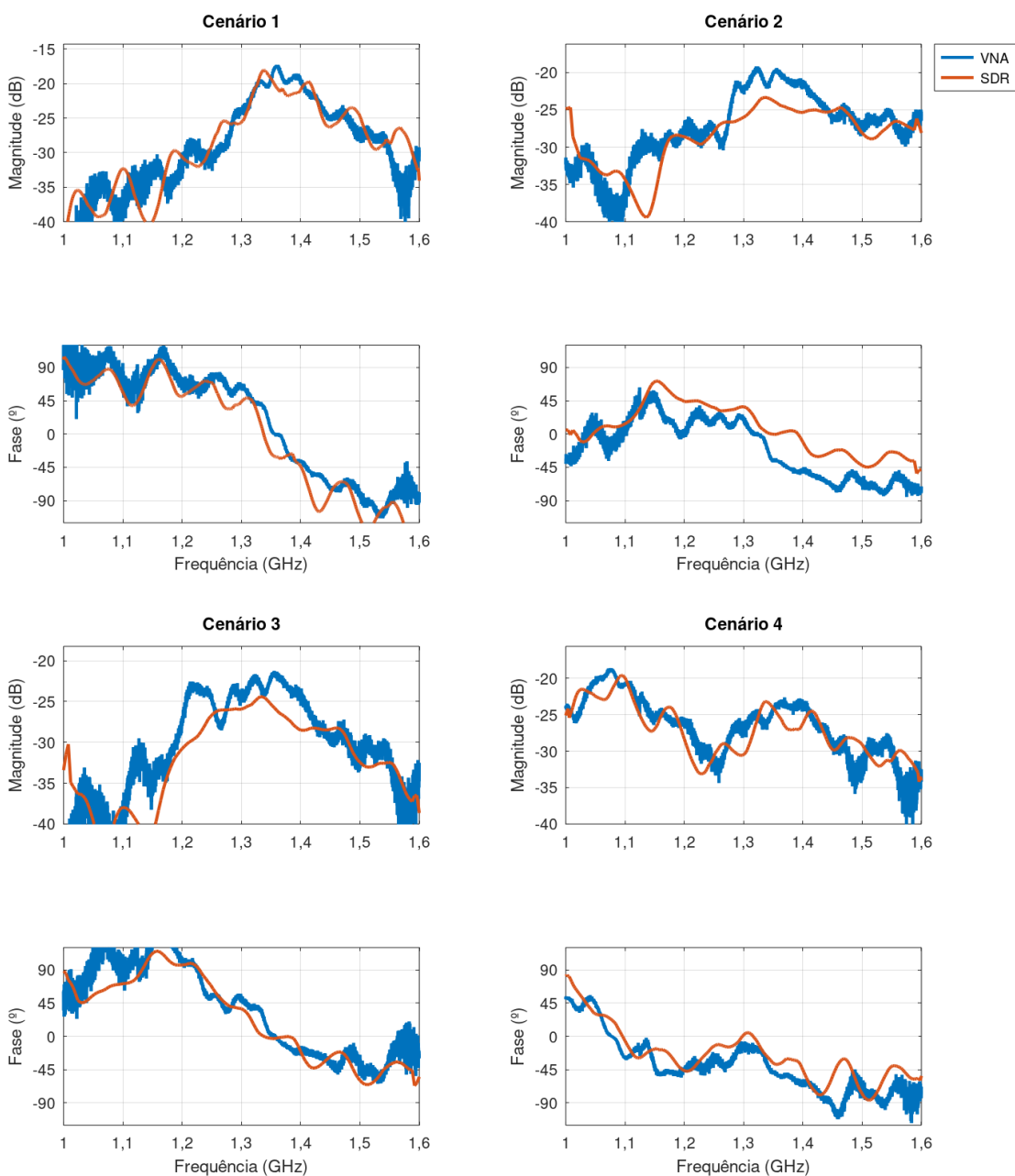
O detalhamento da execução rotina, mostrando os passos desde o início da execução e o encontro da frequência de ressonância antes da alteração, localiza-se no apêndice D. A figura 59 mostra o programa tendo concluído o processo de busca pela nova frequência de ressonância.

Figura 57 – Sobreposição da resposta em frequência obtido pelo VNA e pelo SDR para o sensor A nos 4 cenários.



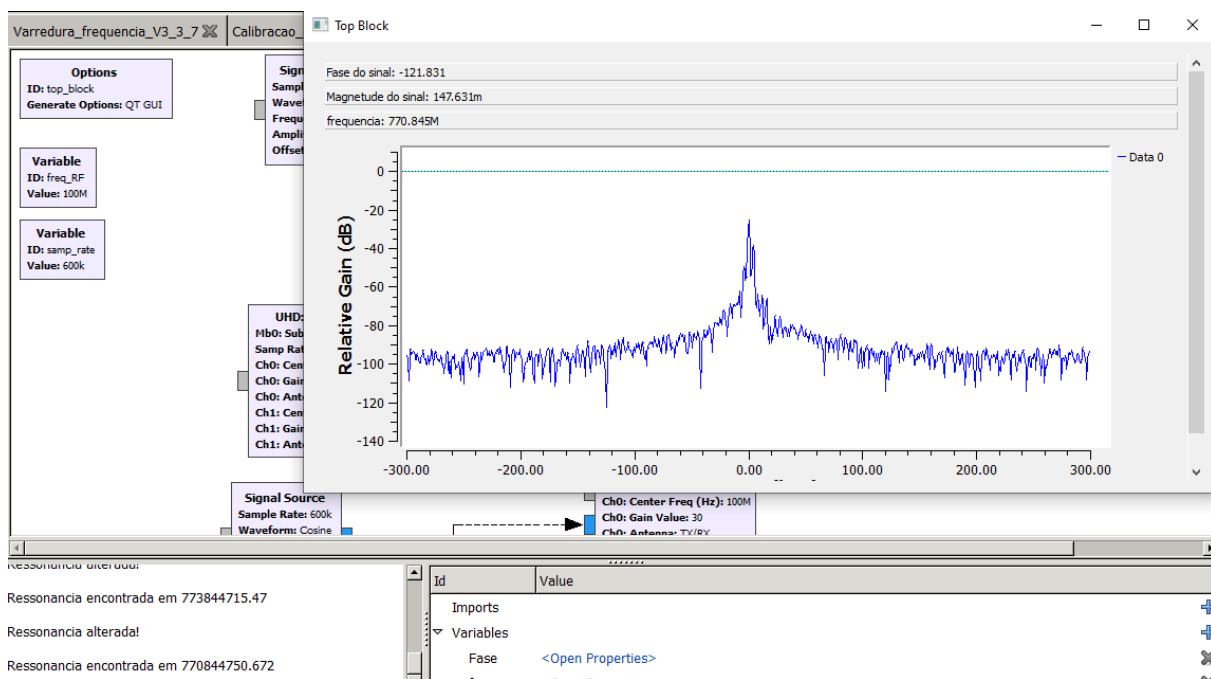
Fonte: O autor

Figura 58 – Sobreposição da resposta em frequência obtido pelo VNA e pelo SDR para o sensor B nos 4 cenários.



Fonte: O autor

Figura 59 – Finalização da busca pela nova ressonância a partir da rotina de rastreamento da ressonância, tendo encontrado a resposta com o sensor coberto.



Fonte: O autor

## 5 CONCLUSÃO

Neste trabalho, concluiu-se a bancada de ensaios baseada em SDR para sensores ressonantes de micro-ondas. Com a utilização da placa de SDR, foi possível implementar algoritmos para caracterização desses sensores e o rastreo da frequência de ressonância.

A implementação da bancada de ensaios com SDR traz a possibilidade de caracterizar sensores ressonantes de diversas naturezas com mínimas alterações de *hardware*, com relação aos conectores, tipo de cabos e distintos comprimentos e adaptadores de RF. Isso está conforme a diversas outras áreas em que foi aplicada a Instrumentação Definida por Software.

A caracterização dos sensores realizada através da bancada resultou em valores de frequência de ressonância próximos aos obtidos através do VNA, tendo erros inferiores a 5%. Apesar do maior erro entre os valores do fator de qualidade e da largura, destaca-se que eles obtiveram uma mesma ordem de grandeza.

A rotina de rastreo de ressonância ainda precisa de ajustes para funcionar de forma ótima, podendo detectar a alteração corretamente e o mais rápido possível, com a correta direção de alteração da frequência. Caso a rotina de rastreo de ressonância possa atingir esses parâmetros, poderá ser possível uma detecção em tempo real do valor da ressonância, possibilitando testes dinâmicos do sensor em análise.

Em seu estado atual, a bancada de ensaios baseada em SDR servirá como suporte para medição e caracterização de protótipos de sensores desenvolvidos pelos alunos de engenharia elétrica do IFSC - Campus Itajaí. Esse banco de ensaio também pode ser utilizado para outros trabalhos experimentais que envolvem a detecção de um material sobre o sensor.

### 5.1 Trabalhos Futuros

Além da utilização para auxiliar trabalhos que também aplicam a tecnologia RF, há outras possibilidades para aprimoramento e extensão do sistema. O algoritmo pode ser adaptado para receber sensores de anti-ressonância, cuja resposta alcança a magnitude mínima no valor de frequência projetado. Outra possibilidade seria estudar diferentes larguras de trilhas ou outros tipos de materiais para placas de circuito impresso.

Vários dos gráficos realizados resultaram em um valor máximo por volta de -20 dB. Se o pico for incrementado, o SDR poderá aproveitar melhor o alcance do conversor analógico-digital, melhorando as leituras. Faixas maiores de frequência podem ser utilizadas para uma maior descrição da resposta do sensor. Pode-se também realizar um ajuste para obter a variação da resposta do sistema com a modificação dos ganhos e implementar um controle de ganho automático (AGC), para aumentá-lo até próximo a saturação do sinal.

Os *scripts* podem ser aprimorados. Com isso, a leitura e a interpretação dos dados podem ser mais precisas, em conjunto com o tempo de execução mais rápido de cada programa. As rotinas complementares de média móvel, cálculo da inclinação e do ângulo de referência podem ser melhoradas, utilizando algoritmos mais elaborados de compensação da linha de transmissão, banco de dados para ter uma noção pre-liminar do comportamento do sensor e também com aprendizado de máquina.

A execução das rotinas pode ter um programa próprio e com interface mais amigável que a utilização do GNU Radio. Uma quantidade considerável dos ajustes das rotinas foi realizada puramente na programação direta em python. Uma possibilidade seria utilizar o Tkinter (FOUNDATION, 2021), que já possui inclusive a formação dos programas em python. Pode-se estabelecer pré-configurações para as rotinas e as sub-rotinas de ajuste com mais opções de parâmetros e assim evitar a utilização extensa de um editor de código. O programa precisaria ter um processamento mais eficiente ou já definir a prioridade de execução em tempo real, no caso do Windows.

O algoritmo de rastreamento poderia ser provido de melhorias no tempo de rastreamento, remoção da bandeira de direção e rejeição de detecções falsas. Outra possibilidade é a de utilizar uma arquitetura de PLL para o rastreamento dinâmico da frequência de ressonância.

## Referências

- ALEXANDER, C.; SADIKU, M. **Fundamentos de Circuitos Elétricos - 5ed.** AMGH Editora, 2013. ISBN 9788580551730. Disponível em: <<https://books.google.com.br/books?id=iSk6AgAAQBAJ>>. Citado 2 vezes nas páginas 7 e 8.
- ANKITA; BASU, A. Microwave imaging using distributed sensors. In: **IEEE MTT-S International Microwave and RF Conference**. [S.l.: s.n.], 2013. p. 1–4. Citado na página 2.
- ANSOFT, L. **An Introduction to HFSS. Fundamental Principles, Concepts, and Use.** 2009. Acesso em 17 de outubro de 2021. Disponível em: <[http://www.wiki.itajai.ifsc.edu.br/images/9/90/EEL\\_ITJ\\_Projeto\\_EDA\\_Guia\\_1.pdf](http://www.wiki.itajai.ifsc.edu.br/images/9/90/EEL_ITJ_Projeto_EDA_Guia_1.pdf)>. Citado 3 vezes nas páginas 17, 19 e 20.
- ANSYS. **ANSYS**. 2021. Acesso em 16 de outubro de 2021. Disponível em: <<https://www.ansys.com/en-in>>. Citado na página 17.
- ANSYS. **Ansys Electronics Desktop**. 2021. Acesso em 16 de outubro de 2021. Disponível em: <<https://www.ansys.com/en-in/academic/students/ansys-electronics-desktop-student>>. Citado na página 17.
- ANSYS. **Ansys HFSS**. 2021. Acesso em 16 de outubro de 2021. Disponível em: <<https://www.ansys.com/en-in/products/electronics/ansys-hfss>>. Citado na página 17.
- ANTAYHUA, R. R.; RAMBO, C. R.; DE SOUZA, F. R. Self-interference cancellation in chipless rfid readers for reading range enhancement. In: **2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)**. [S.l.: s.n.], 2018. p. 1–6. Citado na página 3.
- BRAY, J.; ROY, L. Microwave characterization of a microstrip line using a two-port ring resonator with an improved lumped-element model. **IEEE Transactions on Microwave Theory and Techniques**, v. 51, n. 5, p. 1540–1547, 2003. Citado 2 vezes nas páginas 7 e 9.
- COLLINS, T. F. et al. **Software Defined Radio for Engineers**. AMGH Editora, 2018. ISBN 978-1-63081-457-1. Disponível em: <<https://www.analog.com/media/en/training-seminars/design-handbooks/Software-Defined-Radio-for-Engineers-2018/SDR4Engineers.pdf>>. Citado 2 vezes nas páginas 2 e 14.
- COOMBS, C. F. J. **Electronic Instrument Handbook**. Terceira edição. [s.n.], 2000. Acesso em 11 de junho de 2021. Disponível em: <<https://www.accessengineeringlibrary.com/content/book/9780070126183/chapter/chapter45>>. Citado na página 12.
- DEVICES, A. **AD9361**. 2017. Acesso em: 23 de setembro de 2021. Disponível em: <<https://www.analog.com/en/products/ad9361.html#product-overview>>. Citado 5 vezes nas páginas 2, 16, 17, 18 e 83.
- DEVICES, A. **ADALM-PLUTO**. 2021. Acesso em 13 de setembro de 2021. Disponível em: <<https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview>>. Citado na página 2.

- EIRELI, J. E. **CABO USB 3.0 A MACHO X B MACHO 1,8 METROS CBUS0013 STORM**. 2021. Acesso em 29 de setembro de 2021. Disponível em: <https://www.japinhaeletro.com.br/cabos-e-adaptadores/cabo-usb-3-0-a-macho-x-b-macho-1-8-metros-cbus0013-storm>. Citado na página 25.
- ELECTRONIC-NOTES. **PXI: PCI eXtensions for Instrumentation Tutorial**. 2021. Acesso em 12 de jun. de 2021. Disponível em: <https://www.electronics-notes.com/articles/test-methods/pxi/pxi.php>. Citado na página 13.
- FOUNDATION, P. S. **tkinter - Python interface to Tcl,Tk**. 2021. Acesso em 22 de outubro de 2021. Disponível em: <https://docs.python.org/3/library/tkinter.html>. Citado na página 65.
- FRENZEL, L. E. **Instrumentation 2.0: How Software-Defined Instrumentation Is Changing T&M**. 2007. Acesso em 07 de jun. de 2021. Disponível em: <https://www.electronicdesign.com/technologies/boards/article/21750398/instrumentation-20-how-softwaredefined-instrumentation-is-changing-t>. Citado 2 vezes nas páginas 2 e 13.
- FRIZELLE, D.; KEARNEY, F. **Complex RF Mixers, Zero-IF Architecture and Advanced Algorithms: The Black Magic in Next Generation SDR Transceivers**. 2017. Acesso em 28 de junho de 2021. Disponível em: <https://www.analog.com/en/analog-dialogue/articles/complex-mixers-zif-architecture-advanced-algorithms-black-magic-next-generation-sdr-transceivers.html>. Citado na página 15.
- GROENENDAAL, H. v. d. **Software-defined testing is gaining traction**. 2018. Acesso em: 10 de junho de 2021. Disponível em: <https://www.ee.co.za/article/software-defined-testing-is-getting-traction.html>. Citado 3 vezes nas páginas 2, 12 e 13.
- HARRIS, F.; LOWDERMILK, W. Software defined radio: Part 22 in a series of tutorials on instrumentation and measurement. **IEEE Instrumentation Measurement Magazine**, v. 13, n. 1, p. 23–32, 2010. Citado 2 vezes nas páginas 13 e 14.
- HAYT, W.; BUCK, J. **Eletromagnetismo - 8ed**. Bookman Editora, 2013. ISBN 9788580551549. Disponível em: <https://books.google.com.br/books?id=emU3AgAAQBAJ>. Citado 4 vezes nas páginas 9, 10, 11 e 12.
- HERSHBERGER, J.; PRATT, T.; KOSSLER, R. Implementations of coherent software-defined dual-polarized radars. **IEEE Transactions on Microwave Theory and Techniques**, PP, p. 1–9, 03 2017. Citado na página 26.
- HUMPHRIES, J. et al. Standalone saw sensor interrogator using an embedded computer and software defined radio. In: **2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)**. [S.l.: s.n.], 2015. p. 1–5. Citado na página 3.
- INSTRUMENTS, N. **PXI Platform Power and Cooling**. 2021. Acesso em 28 de setembro de 2021. Disponível em: <https://www.ni.com/pt-br/support/documentation/supplemental/17/pxi-platform-power-and-cooling.html>. Citado na página 13.

- INSTRUMENTS, N. **PXI Systems**. 2021. Acessado em 12 de junho de 2021. Disponível em: <<https://www.ni.com/pt-br/shop/pxi.html>>. Citado 2 vezes nas páginas 2 e 13.
- INSTRUMENTS, N. **Software Defined Radio**. 2021. Acessado em 07 de junho de 2021. Disponível em: <<https://www.ni.com/pt-br/innovations/wireless/software-defined-radio.html>>. Citado na página 2.
- INSTRUMENTS, N. **USRP 2922**. 2021. Acesso em 27 de setembro de 2021. Disponível em: <<https://www.ni.com/pt-br/support/model.usrp-2922.html>>. Citado na página 3.
- INSTRUMENTS, N. **What is LabVIEW?** 2021. Acesso em 27 de setembro de 2021. Disponível em: <<https://www.ni.com/pt-br/shop/labview.html>>. Citado 2 vezes nas páginas 3 e 15.
- KEIM, R. **Introduction to Software-Defined Radio**. 2017. Acesso em 27 de setembro de 2021. Disponível em: <<https://www.allaboutcircuits.com/technical-articles/introduction-to-software-defined-radio/>>. Citado 2 vezes nas páginas 2 e 13.
- KORNPROBST, J.; MAUERMAYER, R.; EIBERT, T. Versatile low-cost and light-weight rf equipment for field measurements. In: . [S.l.: s.n.], 2020. p. 1–3. Citado na página 3.
- KRASZEWSKI, A. W.; NELSON, S. O. Microwave permittivity determination in agricultural products. **Journal of Microwave Power and Electromagnetic Energy**, Taylor & Francis, v. 39, n. 1, p. 41–52, 2004. Disponível em: <<https://doi.org/10.1080/08327823.2004.11688507>>. Citado na página 7.
- LATHI, B. P.; Z., D. **Sistemas de Comunicações Analógicas e Digitais Modernos - 4<sup>a</sup> ed.** [S.l.]: LTC, 2012. ISBN 9788521620273. Citado na página 14.
- MARIMUTHU, J.; BIALKOWSKI, K. S.; ABBOSH, A. M. Software-defined radar for medical imaging. **IEEE Transactions on Microwave Theory and Techniques**, v. 64, n. 2, p. 643–652, 2016. Citado na página 2.
- MAZZARO, G. et al. Portable ring-resonator permittivity measurement system: Design & operation. 04 2012. Citado na página 9.
- MAZZARO, G. J. In-situ permittivity measurements using ring resonators. In: RANNEY, K. I.; DOERRY, A. W. (Ed.). **Radar Sensor Technology XVI**. SPIE, 2012. v. 8361, p. 332 – 345. Disponível em: <<https://doi.org/10.1117/12.921791>>. Citado 2 vezes nas páginas 8 e 10.
- MEANEY, P. et al. A 4-channel, vector network analyzer microwave imaging prototype based on software defined radio technology. **The Review of scientific instruments**, v. 90, n. 4, p. 044708, April 2019. ISSN 0034-6748. Disponível em: <<https://europepmc.org/articles/PMC6483785>>. Citado 3 vezes nas páginas 3, 41 e 72.
- NAYAK, M. et al. A real time implementation of spectrum sensing system using software defined radio. In: **2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)**. [S.l.: s.n.], 2017. p. 603–607. Citado na página 2.
- NYFORS, E. Industrial microwave sensors—a review. **Subsurface Sensing Technologies and Applications**, v. 1, p. 23–43, 01 2000. Citado 5 vezes nas páginas 1, 2, 6, 7 e 8.

- POCKETVNA. **PocketVNA**. 2021. Acesso em: 11 de junho de 2021. Disponível em: <[https://pocketvna.com/?piwik\\_campaign=gad&piwik\\_kwd=low\EALw\\_wcB](https://pocketvna.com/?piwik_campaign=gad&piwik_kwd=low\EALw_wcB)>. Citado 2 vezes nas páginas 35 e 44.
- POCKETVNA. **PocketVNA Software**. 2021. Acesso em: 11 de junho de 2021. Disponível em: <<https://pocketvna.com/software>>. Citado 2 vezes nas páginas 16 e 44.
- POMES, C. **Quale computer scegliere: requisiti per un buon PC**. 2021. Acesso em 25 de setembro de 2021. Disponível em: <<http://www.navigaweb.net/2018/02/quale-computer-scegliere-requisiti-per.html>>. Citado na página 25.
- POZAR, D. **Microwave Engineering, 4th Edition**. Wiley, 2011. ISBN 9781118213636. Disponível em: <<https://books.google.com.br/books?id=JegbAAAQBAJ>>. Citado 4 vezes nas páginas 7, 8, 11 e 35.
- RADIO, G. **Página Inicial**. 2021. Acesso em: 01 de junho de 2021. Disponível em: <<https://www.gnuradio.org/>>. Citado 2 vezes nas páginas 3 e 15.
- RADIO, G. **Tutoriais Guiados GRC**. 2021. Acesso em: 18 de Outubro de 2021. Disponível em: <[https://wiki.gnuradio.org/index.php/Guided\\_Tutorial\\_GRC](https://wiki.gnuradio.org/index.php/Guided_Tutorial_GRC)>. Citado 3 vezes nas páginas 21, 22 e 23.
- RADIO, G. **Tutoriais Guiados Introdução**. 2021. Acesso em: 18 de Outubro de 2021. Disponível em: <[https://wiki.gnuradio.org/index.php/Guided\\_Tutorial\\_Introduction](https://wiki.gnuradio.org/index.php/Guided_Tutorial_Introduction)>. Citado 2 vezes nas páginas 20 e 21.
- REHMAN, M. et al. Rf sensing based breathing patterns detection leveraging usrp devices. **Sensors**, v. 21, 05 2021. Citado na página 3.
- RESEARCH, E. **B200/B210/B200mini/B205mini**. 2020. Acessado em 01 de junho de 2021. Disponível em: <<http://kb.ettus.com/B200/B210/B200mini/B205mini>>. Citado 4 vezes nas páginas 2, 16, 17 e 25.
- RESEARCH, E. **B200 RF Performance**. 2021. Acesso em 25 de agosto de 2021. Disponível em: <[http://kb.ettus.com/images/c/cb/B200\\_RF\\_Performance.pdf](http://kb.ettus.com/images/c/cb/B200_RF_Performance.pdf)>. Citado na página 41.
- RONDE&SCHWARZ. **Analisador de espectro e sinal**. 2021. Acesso em 11 de jun. de 2021. Disponível em: <[https://www.rohde-schwarz.com/br/produtos/teste-e-medicao/analísadores-de-espectro-de-sinal/visao-geral-pg\\_63665.html](https://www.rohde-schwarz.com/br/produtos/teste-e-medicao/analísadores-de-espectro-de-sinal/visao-geral-pg_63665.html)>. Citado 2 vezes nas páginas 2 e 13.
- RUGGIERO, M.; LOPES, V. da R. **Cálculo numérico: aspectos teóricos e computacionais**. [S.l.]: Pearson Makron Books, 1996. ISBN 9788534602044. Citado na página 28.
- SCHMIDT, E. et al. Software-defined radio gnss instrumentation for spoofing mitigation: A review and a case study. **IEEE Transactions on Instrumentation and Measurement**, v. 68, n. 8, p. 2768–2784, 2019. Citado na página 2.
- SHAW, G. R.; LOPATO, J. Software defined radio as a solution to testing rf avionics. In: **2014 IEEE AUTOTEST**. [S.l.: s.n.], 2014. p. 290–292. Citado na página 2.
- TEKTRONICS. **TekSmartLab**. 2021. Acesso em: 11 de jun de 2021. Disponível em: <[https://www.tek.com/datasheet/teksmartlab\(tm\)](https://www.tek.com/datasheet/teksmartlab(tm))>. Citado 2 vezes nas páginas 2 e 13.

TONG, K.; GAO, Y. The framework of an software-defined gyroscope and stochastic error modeling analysis. In: **2020 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)**. [S.l.: s.n.], 2020. p. 1–4. Citado na página 2.

ZHANG, C.; JIN, W. New three-phase phase-locked-loop method deployed on software defined instrumentation platform. In: **2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control**. [S.l.: s.n.], 2012. p. 1289–1292. Citado na página 2.

ZHANG, K. et al. Microwave sensing of water quality. **IEEE Access**, v. 7, p. 69481–69493, 2019. Citado na página 2.

## Apêndices

## APÊNDICE A – Cálculo teórico para a obtenção da frequência de ressonância através da diferença de fase

Os cálculos utilizados para esse projeto serão listados neste apêndice para explicar as decisões de usar cada bloco no diagrama no GNU Radio.

O objetivo final desses cálculos é conseguir separar, ou identificar, a fase e a amplitude de um sinal periódico senoidal em banda base. Serão utilizadas as variáveis mostradas no diagrama da figura 22 e considerados os efeitos de cada bloco para obter a expressão em banda base dos sinais recebidos. As identidades trigonométricas das equações (15), (16), (17) e (18) foram utilizadas para facilitar o cálculo:

$$\cos(\theta)\cos(\phi) = \frac{\cos(\theta - \phi) + \cos(\theta + \phi)}{2} \quad (15)$$

$$\sin(\theta)\sin(\phi) = \frac{\cos(\theta - \phi) - \cos(\theta + \phi)}{2} \quad (16)$$

$$\sin(\theta)\cos(\phi) = \frac{\sin(\theta + \phi) + \sin(\theta - \phi)}{2} \quad (17)$$

$$\cos(\theta)\sin(\phi) = \frac{\sin(\theta + \phi) - \sin(\theta - \phi)}{2} \quad (18)$$

### A.1 GERAÇÃO DO SINAL

Partindo da saída dos conversores digitais-analógicos, o sinal de banda-base possui uma frequência de 1 kHz, visto que o RFIC AD9361 possui um bloqueador CC, como comentado na seção 3.2.7.2 (MEANEY et al., 2019). Portanto, as partes em fase e em quadratura do sinal após o conversor são mostradas nas equações (19) e (20).

$$e_i(t) = k\cos(\omega_b t) \quad (19)$$

$$e_q(t) = k\sin(\omega_b t) \quad (20)$$

Onde:

$e_i$  é o sinal de banda-base em fase;

$e_q$  é o sinal de banda-base em quadratura;

$k$  é a amplitude do sinal após o conversor;

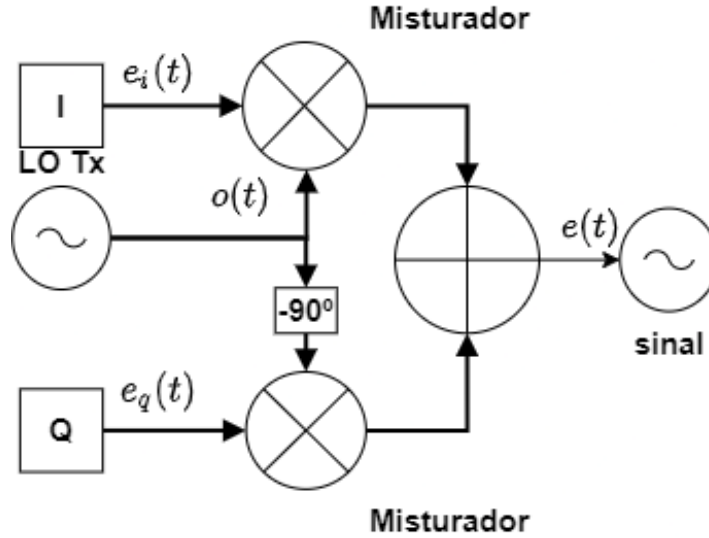
$\omega_b$  é a frequência de banda-base e;

$t$  é o tempo.

Como o sinal possui uma frequência aceita pelo filtro passa-faixa, ele não sofre alterações. Em sequência, o sinal passa pelo misturador de transmissão, mostrado na figura

60, em que a frequência modulante, cuja expressão é mostrada na equação (21), é aplicada ao sinal em fase e em quadratura. A expressão do sinal de saída da placa B210 é mostrada na equação (24), em que a expressão de fase e quadratura são somados.

Figura 60 – Diagrama do misturador de transmissão



Fonte: O autor

$$l_{o_t}(t) = \cos(\omega_c t) \quad (21)$$

$$e_{ic}(t) = e_i(t)l_{o_t}(t) = k\cos(\omega_b t)\cos(\omega_c t) = k\frac{\cos((\omega_b - \omega_c)t) + \cos((\omega_b + \omega_c)t)}{2} \quad (22)$$

$$e_{qc}(t) = e_q(t)l_{o_t}(t - 90^\circ) = k\sin(\omega_b t)\sin(\omega_c t) = k\frac{\cos((\omega_b - \omega_c)t) - \cos((\omega_b + \omega_c)t)}{2} \quad (23)$$

$$\begin{aligned} e(t) &= e_{ic}(t) + e_{qc}(t) = k\frac{\cos((\omega_b - \omega_c)t) + \cos((\omega_b + \omega_c)t)}{2} + \\ &+ k\frac{\cos((\omega_b - \omega_c)t) - \cos((\omega_b + \omega_c)t)}{2} = \\ &= k\cos((\omega_b - \omega_c)t) = k\cos((\omega_c - \omega_b)t) \end{aligned} \quad (24)$$

Onde:

$l_{o_t}(t)$  é o sinal da portadora gerado pelo oscilador de transmissão;

$e_{ic}(t)$  é o sinal em fase após o misturador;

$e_{qc}(t)$  é o sinal em quadratura após o misturador;

$e(t)$  é o sinal de saída do misturador, após a soma dos sinais de fase e quadratura e;

$\omega_c$  é a frequência da portadora.

Destaca-se que o sinal resultante de saída do misturador complexo, localizado na equação (24) possui apenas uma banda lateral de modulação (*Single-Sideband Modulation*, SSB). Este resultado é comum para essa arquitetura de misturadores.

## A.2 SINAIS NOS CANAIS

Saindo da plataforma USRP B210, o sinal entra nas LTs. Dependendo do canal, o sinal passa pela linha de transmissão e retorna diretamente a plataforma, ou é aplicado sob o sensor. Esta seção discutirá, de forma simbólica, as perdas dos canais desde a saída do transmissor até a recepção. O formato simbólico será utilizado porque não se pretende calcular a amplitude e fase do sinal, e sim descrever a alteração do sinal ao longo do percurso.

O sinal descrito pela equação (25) será utilizado para representar o sinal após o ganho de saída da placa B210.

$$x_{Tx}(t) = Ce(t) = Ck\cos((\omega_c - \omega_b)t) \quad (25)$$

Onde:

$C$  é o ganho de saída e;

$x_{Tx}$  é sinal de saída da plataforma USRP B210 nos canais transmissores.

### A.2.1 Canal sem o sensor

Para o canal sem o sensor e sem o conector, ele é aplicado diretamente a linha de transmissão e retorna a plataforma USRP B210. Para a análise do efeito da LT sobre o sinal, é preferível a análise no domínio da frequência. O sinal de transmissão no domínio da frequência é mostrado na equação (26).

$$X_{Tx} = Ck\angle 0^\circ \quad (26)$$

Onde:

$X_{Tx}$  é sinal de saída da plataforma USRP B210 no domínio da frequência.

Por haver casamento de impedância entre a linha de transmissão e a porta de saída da plataforma USRP B210, o sinal após a LT pode ser considerado como a multiplicação de um número complexo em que o módulo e a fase são dados pelas atenuação e de fase da LT. A equação (27) traz a expressão na recepção da plataforma no domínio da frequência.

$$X_{Rx0} = X_{Tx} e^{-\alpha_0 l_2 - j\beta_0 l_2} = Ck\angle 0^\circ e^{-\alpha_0 l_2} \angle -\beta_0 l_2 = Ck e^{-\alpha_0 l_2} \angle -\beta_0 l_2 \quad (27)$$

Onde:

$X_{Rx0}$  é sinal recebido pela plataforma USRP B210 no domínio da frequência a partir do canal sem o conector;

$\alpha_0$  é a constante de atenuação da LT do canal Tx0-Rx0;

$\beta_0$  é a constante de fase da LT do canal Tx0-Rx0 e;

$l_2$  é comprimento da LT no canal onde não é inserido o sensor.

Colocando o sinal da equação (27) no domínio do tempo, obtém-se a expressão dada pela equação (28).

$$x_{Rx0}(t) = Cke^{-\alpha_0 l_2} \cos((\omega_c - \omega_b)t - \beta_0 l_2) \quad (28)$$

$x_{Rx0}$  é sinal recebido pela plataforma USRP B210 no domínio do tempo a partir do canal sem o conector.

Podemos agrupar os termos de amplitude e fase, descritos nas equações (29) e (30), respectivamente. A equação (31) descreve o sinal após a aplicação das equações (29) e (30), na equação (28).

$$B = G_0 Ck e^{-\alpha_0 l_2} \quad (29)$$

$$\phi_x = -\beta_0 l_2 \quad (30)$$

$$x(t) = B \cos((\omega_c - \omega_b)t + \phi_x) \quad (31)$$

Onde:

$x(t)$  é o sinal recebido pela plataforma USRP B210 do canal sem o sensor ou o conector após o ganho de recepção;

$G_0$  é o ganho do canal de recepção do canal Rx0;

$B$  é a amplitude após o ganho do canal de recepção e;

$\phi_x$  é a fase recebida no canal sem o sensor.

### A.2.2 Canal com o adaptador SMA

O canal com o adaptador SMA, possui um procedimento de cálculo similar ao do canal sem o sensor, porém com um comprimento de cabo diferente. Esta subseção descreve o sinal pelo canal utilizando o adaptador SMA para a rotina de ajuste de *offset*.

O adaptador SMA considerado não possui comprimento, de forma a considerar somente o comprimento dos cabos coaxiais do canal Tx1-Rx1. Ele é utilizado para ligar dois cabos coaxiais RG-58, que foram utilizados para conectar o sensor a plataforma USRP B210.

Assim, parte-se da equação (26) para a análise. A equação (32) traz a expressão na recepção da plataforma no domínio da frequência após o cabo coaxial entre os canais Tx1-Rx1.

$$X_{Rx1} = X_{Tx} e^{-\alpha_1 l_1 - j\beta_1 l_1} = Ck \angle 0^\circ e^{-\alpha_1 l_1} \angle -\beta_1 l_1 = Ck e^{-\alpha_1 l_1} \angle -\beta_1 l_1 \quad (32)$$

Onde:

$X_{Rx1}$  é sinal recebido pela plataforma USRP B210 a partir do canal com o adaptador SMA no domínio da frequência e;

$\alpha_1$  é a constante de atenuação da LT do canal Tx1-Rx1;

$\beta_1$  é a constante de fase da LT do canal Tx1-Rx1 e;

$l_1$  é comprimento da LT no canal onde é inserido o sensor.

Colocando o sinal da equação (32) no domínio do tempo, obtém-se a expressão dada pela equação (33).

$$x_{Rx1}(t) = Cke^{-\alpha_1 l_1} \cos((\omega_c - \omega_b)t - \beta_1 l_1) \quad (33)$$

$x_{Rx1}$  é sinal recebido pela plataforma USRP B210, a partir do canal Rx1 com o adaptador SMA, no domínio do tempo.

Podemos agrupar os termos de amplitude e fase, descritos nas equações (34) e (35), respectivamente. A equação (36) descreve o sinal após a aplicação das equações (34) e (35), na equação (33).

$$A_c = G_1 Cke^{-\alpha_1 l_1} \quad (34)$$

$$\phi_1 = -\beta_1 l_1 \quad (35)$$

$$x_c(t) = A_c \cos((\omega_c - \omega_b)t + \phi_1) \quad (36)$$

Onde:

$x_c(t)$  é o sinal recebido pela plataforma USRP B210 através do canal com o adaptador SMA após o ganho de recepção;

$G_1$  é o ganho do canal de recepção do canal Rx1;

$A_c$  é a amplitude após o ganho do canal de recepção e;

$\phi_1$  é a fase recebida no canal Tx1-Rx1 com o adaptador SMA.

### A.2.3 Canal com o sensor

As mesmas considerações dadas para o canal com o adaptador SMA. Assim, pode-se aproveitar o resultado do apêndice A.2.2 e aplicar a resposta do sensor para obter a resposta final.

$$\begin{aligned} X_{Rxs} &= \frac{X_c}{G_1} H e^{j\phi_{sensor}} = Ck \angle 0^\circ e^{-\alpha_1 l_1} \angle -\beta_1 l_1 H \angle \phi_{sensor} = \\ &= HCke^{-\alpha_1 l_1} \angle -\beta_1 l_1 + \phi_{sensor} = \frac{A_c}{G_1} H \angle \phi_1 + \phi_{sensor} \end{aligned} \quad (37)$$

Onde:

$X_c$  é o sinal recebido pela plataforma USRP B210 do canal com o conector SMA após o ganho de recepção, no domínio da frequência;

$X_{Rxs}$  é sinal recebido pela plataforma USRP B210 no domínio da frequência;

$H$  é amplitude da resposta do sensor e;

$\phi_{sensor}$  é a fase da resposta do sensor.

Colocando o sinal da equação (37) no domínio do tempo, obtém-se a expressão dada pela equação (38).

$$x_{Rxs}(t) = H \frac{A_c}{G_1} \cos((\omega_c - \omega_b)t + \phi_1 + \phi_{sensor}) \quad (38)$$

$x_{Rxs}$  é sinal recebido pela plataforma USRP B210 a partir do canal com o sensor no domínio do tempo.

Podemos agrupar os termos de amplitude e fase, descritos nas equações (39) e (40), respectivamente. A equação (41) descreve o sinal após a aplicação das equações (39) e (40) na equação (38).

$$A = A_c H = C k e^{-\alpha_1 l_1} G_1 H \quad (39)$$

$$\phi_g = \phi_1 + \phi_{sensor} \quad (40)$$

$$g(t) = A \cos((\omega_c - \omega_b)t + \phi_g) \quad (41)$$

Onde:

$g(t)$  é o sinal recebido pela plataforma USRP B210 do canal com o sensor após o ganho de recepção;

$A$  é a amplitude do sinal no canal Tx1-Rx1 após o ganho de recepção e;

$\phi_g$  é a fase recebida no canal Tx1-Rx1 com o sensor.

### A.3 OBTENDO O VALOR DE FASE

#### A.3.1 Desenvolvimento matemático para obter a expressão em fase (PDI)

Os cálculos aqui serão realizados para obter a expressão em fase do sinal na entrada do canal de recepção proveniente da resposta do sensor ressonante.

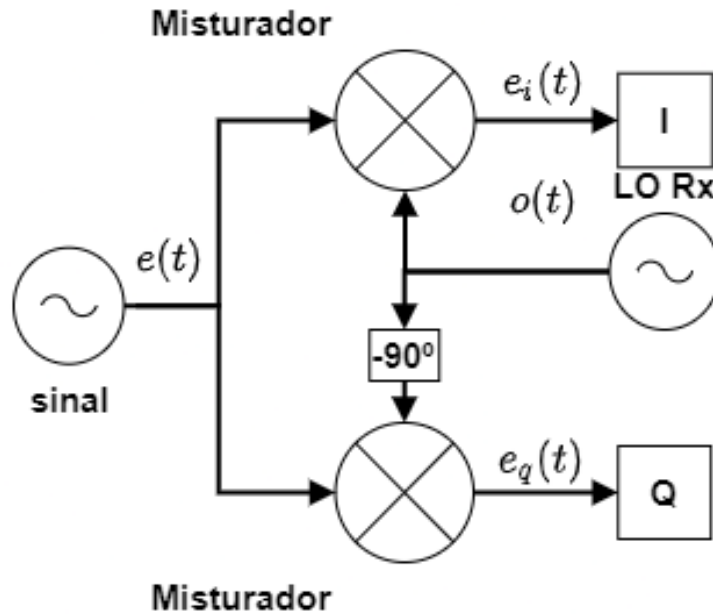
Da figura 22, os misturadores de recepção possuem uma arquitetura similar a da figura 61. Seja  $x(t) = B \cos((\omega_c - \omega_b)t + \phi_x)$  o sinal original,  $l_{or}(t) = \cos(\omega_c t)$  o sinal proveniente do oscilador local dos blocos receptores (LO Rx) e  $g(t) = \cos((\omega_c - \omega_b)t + \phi_g)$  o sinal após o sensor. Após o misturador complexo, a expressão em fase do sinal é:

$$u_{pdi}(t) = g(t)l_{or}(t) = A \cos((\omega_c - \omega_b)t + \phi_g) \cos(\omega_c t)$$

Usando a identidade (15), obtemos:

$$u_{pdi}(t) = 0,5A(\cos(\omega_b t + \phi_g) + \cos((2\omega_c - \omega_b)t + \phi_g))$$

Figura 61 – Diagrama do misturador de recepção.



Fonte: O autor

Após o misturador, o filtro passa-baixa é utilizado na plataforma SDR para eliminar o termo de alta frequência:

$$u_I = 0,5A\cos(\omega_b t + \phi_u) \quad (42)$$

Seguindo o mesmo procedimento para o canal sem o sensor, a expressão recebida pelo processador digital é:

$$v_I = 0,5B\cos(\omega_b t + \phi_v) \quad (43)$$

Onde:

$u_{pdi}$  é a parte em fase do sinal após o misturador de recepção do canal Rx1;

$lo_r$  é o sinal da portadora gerado pelo oscilador de recepção;

$u_I$  é a parte em fase do sinal após o filtro de recepção do canal Rx1, em que será aplicado o conversor analógico-digital;

$v_I$  é a parte em fase do sinal após o filtro de recepção do canal Rx0, em que será aplicado o conversor analógico-digital;

$\phi_u$  é a fase do sinal  $u_I$  e;

$\phi_v$  é a fase do sinal  $v_I$ .

Os sinais  $u_I$  e  $v_I$  e as fases  $\phi_u$  e  $\phi_v$  fazem referência aos sinais localizados na figura 22, localizado na seção 3.2.1.

### A.3.2 Desenvolvimento matemático para obter a expressão em quadratura (PDQ)

Seguindo o mesmo procedimento dado pelo apêndice A.3.1, mas utilizando gerado pelo oscilador defasado para  $l_{o_r}(t - 90^\circ) = \text{sen}(\omega_c t)$ , obtemos a seguinte expressão após o misturador:

$$u_{pdq}(t) = g(t)l_{o_r}(t - 90^\circ) = A\cos((\omega_c - \omega_b)t + \phi_g)\text{sen}(\omega_c t)$$

Usando a identidade (18), obtemos:

$$u_{pdq}(t) = 0,5A(\text{sen}((2\omega_c - \omega_b)t + \phi_g) - \text{sen}(-\omega_b t - \phi_g))$$

Com o filtro após o misturador, o sinal resulta em:

$$u_Q = \frac{A}{2}\text{sen}(\omega_b t + \phi_u) \quad (44)$$

Para o sinal recebido do canal sem o sensor, a sua expressão é:

$$v_Q = \frac{B}{2}\text{sen}(\omega_b t + \phi_v) \quad (45)$$

$u_{pdq}$  é a parte em quadratura do sinal após o misturador de recepção do canal Rx1;

$u_Q$  é a parte em quadratura do sinal após o filtro de recepção do canal Rx1, em que será aplicado o conversor analógico-digital;

$v_Q$  é a parte em quadratura do sinal após o filtro de recepção do canal Rx0, em que será aplicado o conversor analógico-digital;

$\phi_u$  é a fase do sinal  $u_Q$  e;

$\phi_v$  é a fase do sinal  $v_Q$ .

Os sinais  $u_Q$  e  $v_Q$  recebem o mesmo comentário sinais  $u_I$  e  $v_I$  no apêndice A.3.1.

### A.3.3 Sinais de banda base recuperados no PC

Os sinais recuperados em banda base passam por um conversor AD e são processados no computador como  $u$  e  $v$ , onde:

$$u = 0,5A(\cos(\omega_b t + \phi_u) + j\text{sen}(\omega_b t + \phi_u)) \quad (46)$$

$$v = 0,5B(\cos(\omega_b t + \phi_v) + j\text{sen}(\omega_b t + \phi_v)) \quad (47)$$

#### A.4 DESENVOLVIMENTO MATEMÁTICO PARA OBTER A EXPRESSÃO COMPLEXA E OBTER A DIFERENÇA DE FASE

Seja  $v = 0,5B(\cos(\omega_b t + \phi_v) + j\text{sen}(\omega_b t + \phi_v))$  o sinal original e  $u = 0,5A(\cos(\omega_b t + \phi_u) + j\text{sen}(\omega_b t + \phi_u))$  o sinal após o sensor, primeiramente foi aplicada a multiplicação a esses sinais:

$$uv = 0,5A(\cos(\omega_b t + \phi_u) + j\text{sen}(\omega_b t + \phi_u)).0,5B(\cos(\omega_b t + \phi_v) + j\text{sen}(\omega_b t + \phi_v)) \quad (48)$$

$$uv = 0,25AB[\cos(\omega_b t + \phi_u)\cos(\omega_b t + \phi_v) - \text{sen}(\omega_b t + \phi_u)\text{sen}(\omega_b t + \phi_v) + j[\cos(\omega_b t + \phi_u)\text{sen}(\omega_b t + \phi_v) + \text{sen}(\omega_b t + \phi_u)\cos(\omega_b t + \phi_v)]] \quad (49)$$

Utilizando as identidades (15), (16), (17) e (18), obtemos a seguinte expressão:

$$uv = 0,25AB \times \left[ \frac{\cos(2\omega_b t + \phi_u + \phi_v) + \cos(\phi_u - \phi_v)}{2} + \frac{\cos(2\omega_b t + \phi_u + \phi_v) - \cos(\phi_u - \phi_v)}{2} + j \left( \frac{\text{sen}(2\omega_b t + \phi_u + \phi_v) - \text{sen}(\phi_u - \phi_v)}{2} + \frac{\text{sen}(2\omega_b t + \phi_u + \phi_v) + \text{sen}(\phi_u - \phi_v)}{2} \right) \right]$$

Simplificando a equação, obtemos o resultado final dessa multiplicação.

$$uv = 0,25AB[\cos(2\omega_b t + \phi_u + \phi_v) + j(\text{sen}(2\omega_b t + \phi_u + \phi_v))] \quad (50)$$

Ao analisar essa equação, observamos que foi obtida a soma das fases ao invés da diferença, entretanto ela permite perceber que os termos que realizaram as anulações das fases poderia ser evitado se um dos sinais fosse um sinal conjugado. Ao aplicar o conjugado em  $v$ , obtemos a seguinte expressão no lugar de (49).

$$uv^* = 0,25AB[\cos(\omega_b t + \phi_u)\cos(\omega_b t + \phi_v) + \text{sen}(\omega_b t + \phi_u)\text{sen}(\omega_b t + \phi_v) + j[-\cos(\omega_b t + \phi_u)\text{sen}(\omega_b t + \phi_v) + \text{sen}(\omega_b t + \phi_u)\cos(\omega_b t + \phi_v)]]$$

Desenvolvendo a equação:

$$uv^* = 0,25AB \times \left[ \frac{\cos(2\omega_b t + \phi_u + \phi_v) + \cos(\phi_u - \phi_v)}{2} - \frac{\cos(2\omega_b t + \phi_u + \phi_v) - \cos(\phi_u - \phi_v)}{2} + j \left( -\frac{\text{sen}(2\omega_b t + \phi_u + \phi_v) - \text{sen}(\phi_u - \phi_v)}{2} + \frac{\text{sen}(2\omega_b t + \phi_u + \phi_v) + \text{sen}(\phi_u - \phi_v)}{2} \right) \right]$$

$$y = uv^* = 0,25AB[\cos(\phi_u - \phi_v) + j\text{sen}(\phi_u - \phi_v)] \quad (51)$$

Agora, a função é apenas dependente da diferença de fase entre o sinal original e a resposta do sensor. Caso os sinais transmitidos possuam a mesma frequência de banda base, a mesma será eliminada pela multiplicação conjugada.

#### A.4.1 Recuperação de informação da resposta em frequência do sensor

Nesta seção, será discutido como o resultado da equação 51 pode trazer a resposta em frequência do sensor.

As fases dos sinais  $u$  e  $v$  são relacionadas com os sinais das expressões (31) e (41) de acordo com as equações dadas em (52) e (53), respectivamente.

$$\phi_u = \phi_g = \phi_1 + \phi_{sensor} = \phi_{sensor} - \beta_1 l_1 \quad (52)$$

$$\phi_v = \phi_x = -\beta_0 l_2 \quad (53)$$

##### A.4.1.1 Recuperação de amplitude

Para a recuperação da amplitude do sensor, pode-se considerar duas situações: quando o comprimento e a constante de atenuação dos canais são iguais, e quando esta condição não é satisfeita.

Caso o comprimento e a constante de atenuação das LTs seja identico em ambos os canais, tem-se a substituições localizadas em (54) e (55) em  $A$  e  $B$ , resultando nas equações (56) e (57) para os sinais  $u$  e  $v$ , respectivamente.

$$\alpha_1 = \alpha_2 = \alpha \quad (54)$$

$$l_1 = l_2 = l \quad (55)$$

$$|u| = A = CkG_1 H e^{\alpha_1 l_1} = CkG_1 H e^{\alpha l} \quad (56)$$

$$|v| = B = CkG_1 e^{\alpha_0 l_2} = CkG_1 e^{\alpha l} \quad (57)$$

Dividindo (56) por (57), obtemos diretamente a amplitude do sensor, conforme mostra (58).

$$\frac{|u|}{|v|} = \frac{A}{B} = \frac{CkG_1 H e^{\alpha l}}{CkG_1 e^{\alpha l}} = H \quad (58)$$

Caso a constante de atenuação ou o comprimento das LTs não seja o mesmo, pode-se aplicar as equações (39) e (29) em (51), obtendo (60).

$$|uv^*| = 0,25AB = 0,25Cke^{\alpha_1 l_1} G_1 H C k e^{\alpha_0 l_2} G_1 \quad (59)$$

$$|uv^*| = (0,5CkG_1)^2 e^{\alpha_1 l_1} e^{\alpha_2 l_2} H \quad (60)$$

Para a recuperação completa da amplitude, a rotina de registro de *offset* realiza a leitura de (36) no lugar de (38). Isso modifica o valor de  $u$  para o da equação (61), e a expressão final da amplitude se encontra em (62).

$$u_1 = 0,5A_e(\cos(\omega_b t + \phi_1) + j\text{sen}(\omega_b t + \phi_1)) \quad (61)$$

Onde:

$u_1$  é o sinal recuperado pelo SDR do canal Tx1-Rx1 na rotina de registro de *offset*.

$$|u_1 v^*| = (0,5CkG_1)^2 e^{\alpha_1 l_1} e^{\alpha_2 l_2} \quad (62)$$

Portanto, a recuperação de amplitude é dada pela divisão de (60) por (62), resultando em (63).

$$\frac{|uv^*|}{|u_1 v^*|} = \frac{(0,5CkG_1)^2 e^{\alpha_1 l_1} e^{\alpha_2 l_2} H}{(0,5CkG_1)^2 e^{\alpha_1 l_1} e^{\alpha_2 l_2}} = H \quad (63)$$

#### A.4.1.2 Recuperação de fase com canais de comprimentos iguais

Primeiramente, considera-se que a constante de fase entre os canais é a mesma e o comprimento entre os canais é o mesmo, como mostram as equações (64) e (55), respectivamente.

$$\beta_1 = \beta_2 = \beta \quad (64)$$

Logo, a diferença entre as fases localizadas nas equações (52) e (53) é dada pela expressão (66):

$$\phi_u - \phi_v = \phi_g - \phi_x = \phi_{sensor} - \beta_1 l_1 - (-\beta_0 l_2) \quad (65)$$

$$\phi_u - \phi_v = \phi_{sensor} - \beta_1 l_1 + \beta_0 l_2 \quad (66)$$

Aplicando (64) e (55) em (66), obtém-se (67). Esta última, representa a fase do sensor totalmente completa pelo programa.

$$\phi_u - \phi_v = \phi_{sensor} - \beta_1 l_1 + \beta_0 l_2 = \phi_{sensor} - \beta l + \beta l = \phi_{sensor} \quad (67)$$

#### A.4.1.3 Recuperação de fase desvios de fase ou frequência dos osciladores

Como o RFIC utiliza dois osciladores distintos para as cadeias de transmissão e de recepção, eles podem ter desvios tanto de frequência, quanto de fase.

Para modelar esses desvios, será acrescentado dois termos para modelar o desvio ocasionado pelos osciladores de transmissão e de recepção. Aplicando eles nas equações (52) e (53) obtém-se (68) e (69), respectivamente.

$$\phi_u = \phi_g + \phi_{dT_x} + \phi_{dR_x} = \phi_1 + \phi_{sensor} + \phi_{dT_x} + \phi_{dR_x} = \phi_{sensor} - \beta_1 l_1 + \phi_{dT_x} + \phi_{dR_x} \quad (68)$$

$$\phi_v = \phi_x + \phi_{dT_x} + \phi_{dR_x} = -\beta_0 l_2 + \phi_{dT_x} + \phi_{dR_x} \quad (69)$$

A diferença de as fases de (68) e (69) e considerando os comprimentos das LTs dos canais e a constante de fase são iguais, obtém-se 71.

$$\begin{aligned} \phi_u - \phi_v &= (\phi_g + \phi_{dT_x} + \phi_{dR_x}) - (\phi_x + \phi_{dT_x} + \phi_{dR_x}) = \\ &= \phi_{sensor} + \phi_{dT_x} + \phi_{dR_x} - \beta_1 l_1 - (-\beta_0 l_2) - \phi_{dT_x} - \phi_{dR_x} \end{aligned} \quad (70)$$

$$\phi_u - \phi_v = \phi_{sensor} \quad (71)$$

Como ambos os osciladores são aplicados a ambos os canais, os desvios são inseridos de forma idênticas neles. A multiplicação complexa realiza a função de eliminar esses termos que foram adicionados. Deve-se notar que esta eliminação só foi possível devido a forma que os osciladores foram implementados no RFIC AD9361 (DEVICES, 2017) e pela utilização de ambos os canais.

#### A.4.1.4 Recuperação de fase com a rotina de registro de *offset*

Para o caso em que as linhas de transmissão possuem comprimentos ou constantes de fase diferentes, a recuperação da fase do sensor pode ser realizada com auxílio da rotina de registro de *offset*.

Para a fase do sinal com o sensor, será utilizada a equação (66). Utilizando o sinal descrito em (61), a multiplicação complexa com  $v$  resulta em (72).

$$u_1 v^* = 0,25 A_c B [\cos(\phi_1 - \phi_v) + j \text{sen}(\phi_1 - \phi_v)] \quad (72)$$

A fase de (72) é descrita por (73).

$$\phi_1 - \phi_v = -\beta_1 l_1 - (-\beta_0 l_2) = -\beta_1 l_1 + \beta_0 l_2 \quad (73)$$

A diferença entre (66) e (73) é dada por (74).

$$\begin{aligned}(\phi_u - \phi_v) - (\phi_1 - \phi_v) &= \phi_{sensor} - \beta_1 l_1 + \beta_0 l_2 - (-\beta_1 l_1 + \beta_0 l_2) = \\ &= \phi_{sensor} - \beta_1 l_1 + \beta_0 l_2 + \beta_1 l_1 - \beta_0 l_2 = \phi_{sensor}\end{aligned}\tag{74}$$

Observa-se que a fase do sensor pode ser completamente recuperada de forma completa com o auxílio da rotina de registro de *offset*.

## APÊNDICE B – Códigos

Nessa seção serão apresentados os códigos em Python que implementam as rotinas utilizadas neste trabalho.

### B.1 Programação para a calibração

Código em Python utilizado para a calibração do sistema.

```
import numpy as np
from gnuradio import gr
import matplotlib.pyplot as plt

\#constantes
media_movel = 7

def bodeplot(frequencias, sinal, local): \#imprimi a resposta em
                                         frequencia

    mag, Phase, freq = [], [], []
    caminho_arquivo = local
    for element in sinal:
        if abs(element) == 0:
            mag.append(float(-200.0)) \#erro de leitura
        else:
            value = log10(abs(element))
            mag.append(20*value)
        Phase.append(phase(element)*180.0/pi)
    for element in frequencias:
        freq.append(element)

    f1 = plt.figure()
    plt.subplot(2,1,1)
    plt.plot (freq, mag, color="blue", linewidth="1")
    plt.xlabel ("Frequency")
    plt.ylabel ("Magnitude")
    plt.subplot(2,1,2)
    plt.plot (freq, np.unwrap(Phase), color="red", linewidth="1.1")
    plt.xlabel ("Frequency")
    plt.ylabel ("Phase")
    plt.savefig (caminho_arquivo, dpi=300, format="png")

def write_data(frequencias,sinal,file): \#escreve os resultados da
                                         calibracao

    f = open(file, "w")
    f.write(str(frequencias) + '\n')
```

```
f.write(str(sinal))
f.close()

class blk(gr.sync_block):
    def __init__(self, freq_inicial = 5e8, freq_final = 1.5e9, passo =
                2e7, caminho_grafico = "C:\\
                Users\\User\\.gnuradio\\versao_3
                \\calibracao.png",
                caminho_arquivo = "C:\\Users\\
                User\\.gnuradio\\versao_3\\
                calibracao.txt"):

        gr.sync_block.__init__(
            self,
            name='Calibracao',
            in_sig=[np.float32,np.complex64],
            out_sig=[np.float32]

        )
        self.borda = False
        self.freq, self.freq_fim = freq_inicial , freq_final
        self.var_freq = passo
        self.freq_vec = []
        self.fim = False
        self.signal = []
        self.caminho = caminho_arquivo
        self.caminho2 = caminho_grafico

    def tratar_media(self): \#media movel
        sinal, media = self.signal, 0
        for i in range(-1,media_movel-2):
            media += sinal[i]/media_movel
        for i in range(int(media_movel/2),len(sinal)-int(media_movel/2)):
            media += (sinal[i + int(media_movel/2)] - sinal[i - int(
                media_movel/2) - 1])/media_movel
        self.signal[i] = media

    def salvar_dados(self, frequencia, sinal): \#armazenamento dos
                pontos lidos

        self.freq_vec.append(frequencia)
        self.signal.append(sinal)

    def passo(self): \# passo de frequencia
        self.freq += self.var_freq
        if self.freq < 0.0:
            self.freq *= -1.0
            self.var_freq *= -1.0
```

```

def work(self, input_items, output_items): \#funcao principal
    clock = input_items[0][0] \#leitura do sinal de clock
    if ((clock > 0.8) and (self.borda == False)):
        self.borda = True
        medidas = input_items[1][0] \#leitura do sinal
        if self.freq < self.freq_fim: \#condicao de continuacao de
            passo
            self.salvar_dados(self.freq, (medidas))
            self.passo()
        else:
            if not(self.fim): \#fim da calibracao
                self.fim = True
                self.tratar_media() write_data(self.freq_vec, self.
                    signal, self.
                    caminho)
                bodeplot(self.freq_vec, self.signal, self.caminho2)
                self.fim = True
    if clock < 0.2:
        self.borda = False
    output_items[0][:] = self.freq
    return len(output_items[0])

```

## B.2 Código da caracterização e varredura de frequência

Código em python utilizado para executar a caracterização do sensor. A rotina de varredura de frequência está implementada em conjunto para se obter uma estimativa da frequência de ressonância fundamental.

```

import numpy as np
from gnuradio import gr
from math import log10
from cmath import phase, pi, exp
import matplotlib.pyplot as plt

\#constantes
media_movel = 7

\#funcoes externas
def media(entrada):
    valor = 0.0
    for elemento in entrada:
        valor += elemento
    return valor/len(entrada)

def bodeplot(frequencias, sinal, local): \#imprimi a resposta em
    frequencia do sensor

```

```
mag, Phase, freq = [], [], []
caminho_arquivo = local
for element in sinal:
    if abs(element) == 0:
        mag.append(float(-100))
    else:
        value = log10(abs(element))
        mag.append(20*value)
    Phase.append(phase(element)*180.0/pi)
for element in frequencias:
    freq.append(element)

f1 = plt.figure()
plt.subplot(2,1,1)
plt.plot (freq, mag, color="blue", linewidth="1")
plt.xlabel ("Frequency")
plt.ylabel ("Magnititude")
plt.subplot(2,1,2)
plt.plot (freq, (Phase), color="red", linewidth="1.1")
plt.xlabel ("Frequency")
plt.ylabel ("Phase")
plt.savefig (caminho_arquivo, dpi=300, format="png")

def write_data(frequencias,sinal,file): \#dados em texto do sensor
    f = open(file, "w")
    f.write(str(frequencias) + "\n")
    f.write(str(sinal))
    f.close()

def read_calibracao(arquivo): \#leitura da calibracao
    f = open(arquivo,"r")
    frequencias = f.readline()
    sinal = f.readline()
    f.close()
    \#tratamento do texto lido do arquivo
    frequencias = frequencias.replace("[", "")
    frequencias = frequencias.replace("]\n", "")
    frequencias = frequencias.split(",")
    frequencia = []
    for x in frequencias:
        frequencia.append((x))
    sinal = sinal.replace("[", "")
    sinal = sinal.replace("]\n", "")
    sinal = sinal.split(", ")
    sinal = []
    for x in sinal:
        sinal.append((x))
```

```

    return frequencia, sinal

class blk(gr.sync_block):
    def __init__(self, freq_inicial = 5e8, freq_final = 1.5e9, passo =
                2e7, caminho_grafico = "C:\\
                Users\\Renato\\.gnuradio\\
                versao_3\\bodeplot.png",
                caminho_arquivo = "C:\\Users\\
                Renato\\.gnuradio\\versao_3\\
                calibracao.txt", resultados = "C
                :\\Users\\Renato\\.gnuradio\\
                versao_3\\filtro.txt"):

        gr.sync_block.__init__(
            self,
            name='Detector_de_ressonancia',
            in_sig=[np.float32,np.complex64],
            out_sig=[np.float32]
        )
        self.borda = False
        self.freq, self.freq_inicial, self.freq_fim = freq_inicial,
                freq_inicial , freq_final
        self.var_freq, self.passo_inic = passo, passo
        self.freq_vec = []
        self.signal = []
        self.ref_abs = 0
        self.ref_ang = 0
        self.ang = 0
        self.caminho = caminho_arquivo
        self.caminho2 = caminho_grafico
        self.caminho3 = resultados
        self.processo = 0
        self.freq_ress = 0
        self.freq_BW = []
        self.BW = 0
        self.inclinacao = 0
        self.freq_cal, self.sinal_cal = read_calibracao(self.caminho)

    def identificar_filtro(self): \\#identificacao de filtro ressonante
                ou anti-ressonante (nao
                concluida para o segundo)

        frequencia = self.freq_vec
        sinal = self.signal
        magnetude = []
        for elemento in sinal: magnetude.append(abs(elemento))
        if max(magnetude) > magnetude[0]*1.5 or max(magnetude) >
                magnetude[-1]*1.5:

            tipo = 1

```

```

    else:
        tipo = 2
    self.tipo_filtro = tipo
    return tipo

def identificar_picos(self): \#identifica a primeira estimativa da
                                frequencia de ressonancia na
                                fundamental

    magnetude = []
    for elemento in self.signal: magnetude.append(abs(elemento))
    freq_ress_index = magnetude.index(max(magnetude))
    self.freq_ress = self.freq_vec[freq_ress_index]

def tratar_calibracao(self): \#tratamento da calibracao
    i = 0
    while self.freq < float(self.freq_cal[i]):
        i += 1
    if abs(complex(self.sinal_cal[i])) < 1e-6:
        sinal_calibrar = complex(1)
    else:
        sinal_calibrar = complex(1)/complex(self.sinal_cal[i])
    return sinal_calibrar

def tratar_media(self): \#aplicacao da media movel nos dados
                                gravados

    sinal, media = self.signal, 0
    for i in range(-1,media_movel-2):
        media += sinal[i]/media_movel
    for i in range(int(media_movel/2),len(sinal)-int(media_movel/2)):
        media += (sinal[i + int(media_movel/2)] - sinal[i - int(
            media_movel/2) - 1])/media_movel

    self.signal[i] = media

def calcular_inclinacao(self): \#calculo da inclinacao (incompleta)
    inclinar = []
    phi = []
    for elemento in self.signal:
        phi.append(phase(elemento))
    phi = np.unwrap(phi)
    quantidade = int(0.3*len(self.signal))
    for i in range(quantidade):
        dphi1 = phi[(i+2)]-phi[(i+1)]
        df1 = self.freq_vec[(i+2)]-self.freq_vec[(i+1)]
        inclinar.append(dphi1/df1)
    \#self.inclinacao = media(inclinar)
    self.inclinacao = -6,35e-8 \#determinado para o sistema
                                utilizado

```

```

def eliminar_inclinacao(self): \#elimina a inclinacao nos dados
                                gravados

    iteracoes = len(self.signal)
    for i in range(iteracoes):
        self.signal[i] *= exp(-1j*self.inclinacao*self.freq_vec[i])

def acumulador_de_dados(self, frequencia, sinal): \#grava os dados
                                                da varredura de frequencia

    self.freq_vec.append(frequencia)
    self.signal.append(sinal)

def passo(self): \#passo de frequencia
    self.freq += self.var_freq
    if self.freq < 0.0:
        self.freq *= -1.0
        self.var_freq *= -1.0

def calcular_media(self): \#calcula da media angular para a
                            referencia da ressonancia

    iteracoes = len(self.signal)
    media_fase = 0
    for elementos in self.signal:
        media_fase += (phase(elementos))
    media_fase /= iteracoes
    self.ref_ang = media_fase
    for i in range(iteracoes):
        self.signal[i] *= exp(-1j*media_fase)

def work(self, input_items, output_items):
    clock = input_items[0][0]
    if ((clock > 0.8) and (self.borda == False)):
        self.borda = True
    else:
        correcao_calibrar = self.tratar_calibracao()
        Sinal = media(self.var_sinal)*correcao_calibrar
        Phase = phase(Sinal*exp(-1j*self.freq*self.inclinacao -
                                1j*self.ref_ang)) \#
                                                correcao da
                                                inclinacao e da
                                                media da fase

    if self.processo == 0: \#rotina de varredura
        if self.freq < self.freq_fim:
            self.acumulador_de_dados(self.freq,Sinal)
        else:
            self.tratar_media()
            self.calcular_inclinacao()

```

```

        self.eliminar_inclinacao()
        self.calcular_media()
        self.processo = self.identificar_filtro()
        self.identificar_picos()
        self.freq = self.freq_ress
        bodeplot(self.freq_vec, self.signal, self.caminho2
                )
        write_data(self.freq_vec, self.signal, self.
                caminho3)

        self.var_freq /= 10

elif self.processo == 1: \#rotina de deteccao da
                        frequencia de
                        ressonancia
    if abs(Phase) + abs(self.ang) > 0.02*pi and abs(
        Phase) + abs(
        self.ang) < 1.98
        *pi and abs(self
        .var_freq) > 1e2
        :
        if abs(Phase - self.ang) < abs(self.ang):
            self.var_freq /= -2
    else:
        self.processo = 3
        self.var_freq = self.passo_inic/100
        self.freq_ress = self.freq
        self.ref_abs = abs(Sinal)
        self.ref_ang = Phase

elif self.processo == 3: \#Rotina de caracterizacao e
                        obtencao do fator de
                        qualidade
    if abs(Phase) > pi/4:
        self.freq_BW.append(self.freq)
        if len(self.freq_BW) < 2:
            self.var_freq *= -1
            self.freq = self.freq_ress
        else: \# impressao dos resultados
            self.processo = 4 \#fim
            self.freq = self.freq_ress
            print("frequencia de ressonancia: " + str(
                self.
                freq_ress
                ))
            print("Fator de qualidade: " + str(self.
                freq_ress
                /self.BW

```

```

                                                                 ))
        self.var_freq = 0

        self.passo()
        self.ang = Phase

    if clock < 0.2: self.borda = False

    output_items[0][:] = self.freq
    return len(output_items[0])

```

### B.3 Rotina de rastreamento de ressonância

Código em python utilizado para rastrear a frequência de ressonância. A rotina de varredura de frequência foi implementada em conjunto para se obter uma estimativa da frequência de ressonância fundamental.

```

import numpy as np
from gnuradio import gr
from math import log10
#import math
from cmath import phase, pi, exp
import matplotlib.pyplot as plt

#constantes
media_movel = 7

def media(entrada):
    valor = 0.0
    for elemento in entrada:
        valor += elemento
    return valor/len(entrada)

def bodeplot(frequencias, sinal, local):
    mag, Phase, freq = [], [], []
    caminho_arquivo = local
    for element in sinal:
        if abs(element) == 0:
            mag.append(float(-100))
        else:
            value = log10(abs(element))
            mag.append(20*value)
            Phase.append(phase(element)*180.0/pi)
    for element in frequencias:
        freq.append(element/1e6)
    f1 = plt.figure()

```

```
plt.subplot(2,1,1)
plt.plot (freq, mag, color="blue", linewidth="1.5")
plt.ylabel ("Magnitude (dB)")
plt.subplot(2,1,2)
plt.plot (freq, np.unwrap(Phase), color="red", linewidth="1.5")
plt.xlabel ("Frequencia (MHz)")
plt.ylabel ("Fase")
plt.savefig (caminho_arquivo, dpi=300, format="png")

def write_data(frequencias,sinal,file):
    f = open(file, "w")
    f.write("Frequencias , Amplitude (dB), Fase\n")
    mag, Phase= [], []
    for elemento in sinal:
        mag.append(20*log10(abs(elemento)))
        Phase.append(180/pi*phase(elemento))
    for i in range(len(frequencias)):
        f.write(str(frequencias[i]) + " , " + str(mag[i]) + " , " + str(
            Phase[i]) + "\n")

    f.close()

def var(entrada):
    valor = 0
    u = media(entrada)
    for elemento in entrada:
        valor += (elemento - u)**2

    return valor/(len(entrada)-1)

def read_data(arquivo):
    f = open(arquivo,"r")
    inclinacao = float(f.readline())
    media_fase = float(f.readline())
    f.close()
    return inclinacao, media_fase

def read_calibracao(arquivo):
    f = open(arquivo,"r")
    inclinacao = float(f.readline())
    media_fase = float(f.readline())
    frequencias = f.readline()
    signal = f.readline()
    f.close()

    frequencias = frequencias.replace("[", "")
    frequencias = frequencias.replace("]\n", "")
    frequencias = frequencias.split(",")
```

```

frequencia = []

for x in frequencias:
    frequencia.append((x))

#print(frequencias)

signal = signal.replace("[", "")
signal = signal.replace("]\n", "")
signal = signal.split(", ")

sinal = []

for x in signal:
    sinal.append((x))

return frequencia, sinal

class blk(gr.sync_block): # other base classes are basic_block,
                          # decim_block, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self, freq_inicial = 5e8, freq_final = 1.5e9, passo =
                  2e7, caminho_grafico = "C:\\
                  Users\\Renato\\.gnuradio\\
                  versao_3\\bodeplot.png",
                  correcao_conectores = "C:\\Users
                  \\Renato\\.gnuradio\\versao_3\\
                  registro_conectores.txt",
                  resultados = "C:\\Users\\Renato
                  \\gnuradio\\versao_3\\filtro.
                  txt"): # only default arguments
                          here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
            name='Detector de ressonancia',
            in_sig=[np.float32,np.complex64],
            out_sig=[np.float32]
        )
        # if an attribute with the same name as a parameter is found,
        # a callback is registered (properties work, too).
        self.borda = False
        self.freq, self.freq_inicial, self.freq_fim = freq_inicial,

```

```

                                freq_inicial , freq_final
self.var_freq, self.passo_inic = passo, passo
self.freq_vec = []
self.signal = []
self.picos = []
self.freq_anterior = 0
self.ref_abs = 0
self.ref_ang = 0
self.ang = 0
self.caminho = correcao_conectores
self.caminho2 = caminho_grafico
self.caminho3 = resultados
self.processo = 0
self.tipo_filtro = 0
self.var_sinal = []
self.freq_ress = 0
self.direcao = 1
self.freq_BW = []
self.BW = 0
self.inclinacao = 0
self.freq_cal, self.sinal_cal = read_calibracao(self.caminho)
self.copia_sinal = []
print("Inicio rastreador de ressonancia")

def identificar_filtro(self):
    frequencia = self.freq_vec
    sinal = self.signal
    magnetude = []

    for elemento in sinal: magnetude.append(abs(elemento))

    if max(magnetude) > magnetude[0]*1.5 or max(magnetude) >
                                                magnetude[-1]*1.5:
        tipo = 1
    else:
        tipo = 2
    self.tipo_filtro = tipo
    return tipo

def identificar_picos(self):
    magnetude = []

    for elemento in self.signal: magnetude.append(abs(elemento))
    freq_ress_index = magnetude.index(max(magnetude))
    self.freq_ress = self.freq_vec[freq_ress_index]

def tratar_calibracao(self):
```

```

i = 0
#print(self.freq)
while self.freq >= float(self.freq_cal[i]):
    i += 1
    #print(i)
if abs(complex(self.sinal_cal[i])) > 1e-9:
    sinal_calibrar = complex(1)/(complex(self.sinal_cal[i]))
else:
    sinal_calibrar = complex(1)

return sinal_calibrar

def tratar_media(self):
    sinal, media = self.signal, 0

    for i in range(-1,media_movel-2):
        media += sinal[i]/media_movel

    for i in range(int(media_movel/2),len(sinal)-int(media_movel/2)):
        media += (sinal[i + int(media_movel/2)] - sinal[i - int(
            media_movel/2) - 1])/media_movel

    self.signal[i] = media

def calcular_inclinacao(self):
    inclinar = []
    phi = []
    for elemento in self.signal:
        phi.append(phase(elemento))

    phi = np.unwrap(phi)
    quantidade = int(0.2*len(self.signal))
    for i in range(quantidade):
        dphi1 = phi[(i+2)]-phi[(i+1)]
        df1 = self.freq_vec[(i+2)]-self.freq_vec[(i+1)]
        inclinar.append(dphi1/df1)

    self.inclinacao = -2.518e-9
    #self.inclinacao = media(inclinar)

def eliminar_inclinacao(self):
    iteracoes = len(self.signal)
    for i in range(iteracoes):
        self.signal[i] *= exp(-1j*self.inclinacao*self.freq_vec[i])

def acumulador_de_dados(self, frequencia, sinal):
    self.freq_vec.append(frequencia)
    self.signal.append(sinal)

```

```

def passo(self):
    self.freq += self.var_freq
    if self.freq < 0.0:
        self.freq *= -1.0
        self.var_freq *= -1.0

def calcular_media(self):
    iteracoes = len(self.signal)
    media_fase = 0
    for elemento in self.signal:
        media_fase += elemento
    self.ref_ang = (media_fase)/iteracoes

def work(self, input_items, output_items):
    clock = input_items[0][0]

    if ((clock > 0.8) and (self.borda == False)):

        self.borda = True
        if False:
            pass
        else:
            correcao_calibrar = self.tratar_calibracao()
            Sinal = input_items[1][0]*correcao_calibrar
            Phase = phase(Sinal*exp(-1j*self.freq*self.inclinacao -
                                1j*self.ref_ang))

            self.var_sinal = []
            if self.processo == 0:
                if self.freq <= self.freq_fim:
                    self.acumulador_de_dados(self.freq,Sinal)
                else:
                    self.tratar_media()
                    self.calcular_inclinacao()
                    self.eliminar_inclinacao()
                    self.calcular_media()
                    self.processo = self.identificar_filtro()
                    self.identificar_picos()
                    self.freq = self.freq_ress
                    self.freq_anterior = self.freq
                    self.var_freq /= 10
                    self.mag_anterior = abs(Sinal)

            elif self.processo == 1:
                if abs(Phase) + abs(self.ang) > 0.02*pi and abs(
                    Phase) + abs(
                        self.ang) < 1.98

```

```

* $\pi$  and abs(self
.var_freq) > 1e2
:

if abs(Phase) > abs(self.ang):
    self.var_freq /= -2
self.mag_anterior = abs(Sinal)
self.freq_anterior = self.freq
else:
    self.processo = 4
    self.var_freq = 0
    self.freq_ress = self.freq
    self.ref_abs = abs(Sinal)
    self.ref_ang = phase(exp(1j*(self.ref_ang+Phase)
))

self.direcao *= -1
for i in range(len(self.signal)): self.signal[i]
    *= exp(-1j*
self.ref_ang
)

print(self.ref_ang)
print(Phase)
bodeplot(self.freq_vec, self.signal, self.caminho2
)

write_data(self.freq_vec, self.signal, self.
caminho3)

print("\nRessonancia encontrada em " + str(self.
freq_ress) +
"\
nMagnitude
na
ressonancia
(dB): " +
str(20*log10
(self.
ref_abs)))

elif self.processo == 4:
    if abs(Phase) > pi/4.8:
        self.var_freq = self.passo_inic*self.direcao
        self.processo = self.identificar_filtro()
        #self.freq_anterior = self.freq_vec[0]
        self.freq_BW = []
        print("\nRessonancia alterada!")

self.passo()
self.ang = Phase

```

```
if clock < 0.2: self.borda = False

output_items[0][:] = self.freq
return len(output_items[0])
```

## APÊNDICE C – Simulações das rotinas

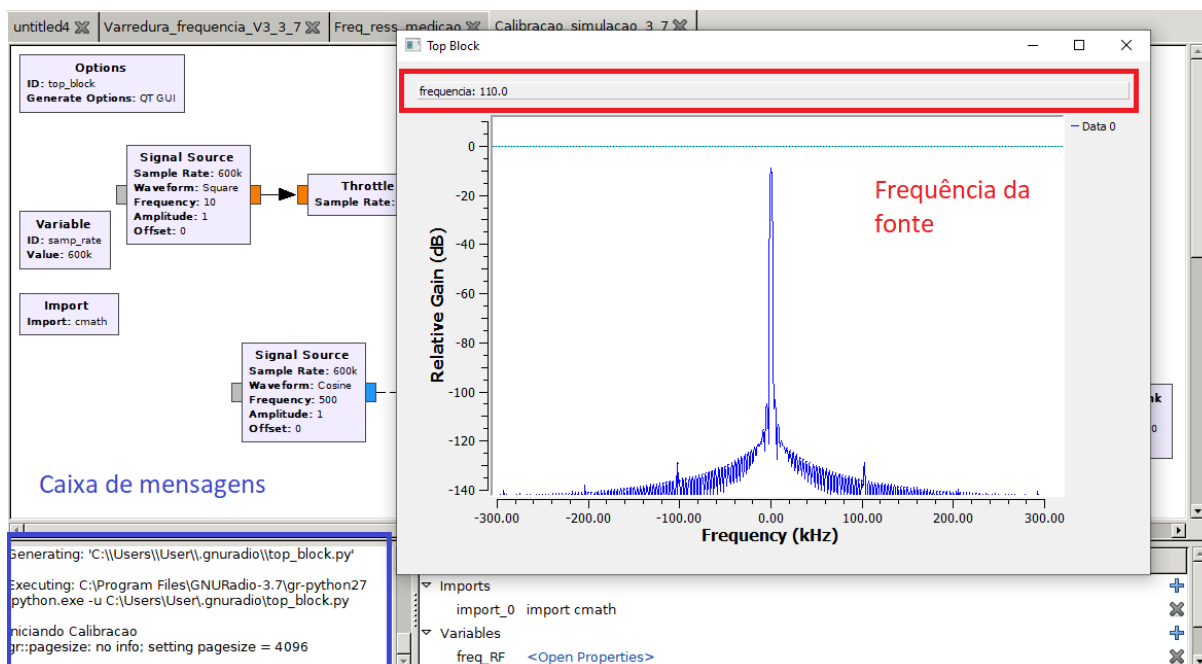
Neste apêndice, serão apresentados os resultados para as simulações dos algoritmos utilizando o GNU Radio 3.7. Os resultados dessas simulações servem para verificar e demonstrar o funcionamento de cada rotina ao longo da execução do programa.

### C.1 Rotina de registro de *offset*

Utilizando como base o diagrama de blocos encontrado na figura 30, essa rotina executa uma varredura de frequência, tendo o seu início e término nas figuras 62 e 63, respectivamente. Para o início da simulação, tem-se a frequência inicial da varredura, mostrada através da janela de saída (*Top Block*), com o valor numérico da frequência.

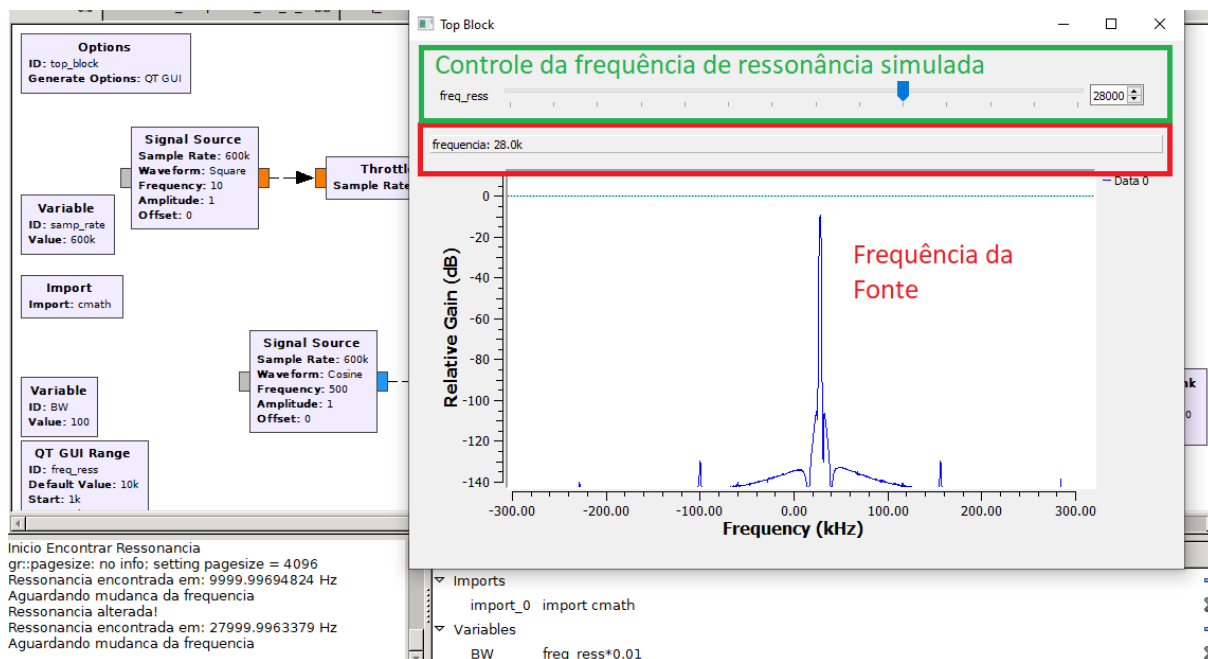
A figura 63 mostra a rotina chegando ao término da varredura. Percebe-se tanto o deslocamento da frequência central do espectro, quanto a caixa indicando o valor da variável de frequência. A simulação da diferença do comprimento das LTs dos canais foi realizada através de um bloco de multiplicação, que alterava o valor da fase conforme a frequência. O gráfico de saída dessa simulação é mostrado na figura 64, mostrando principalmente o deslocamento da fase. Esse resultado está de acordo com o esperado pela simulação, em que o bloco simulando o deslocamento de fase do canal, na direção negativa, foi o sinal em que a operação de conjugado foi aplicada.

Figura 62 – Início da rotina de registro de *offset* simulada.



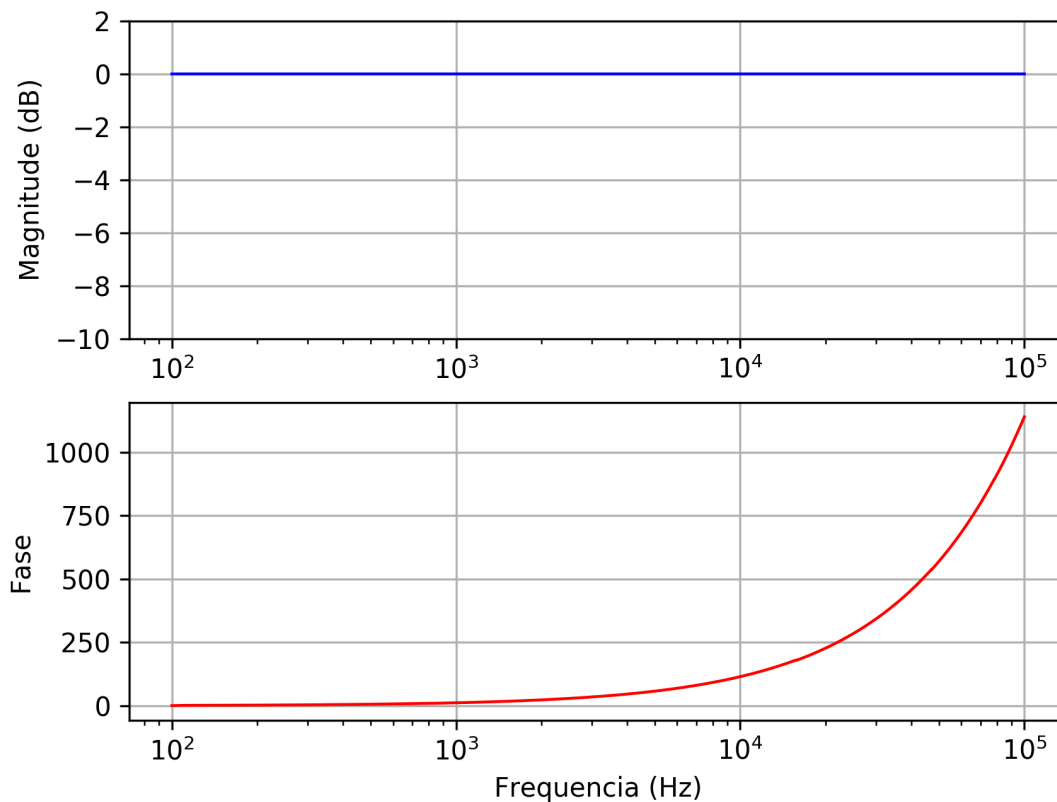
Fonte: O autor

Figura 63 – Final da rotina de registro de *offset* simulada.



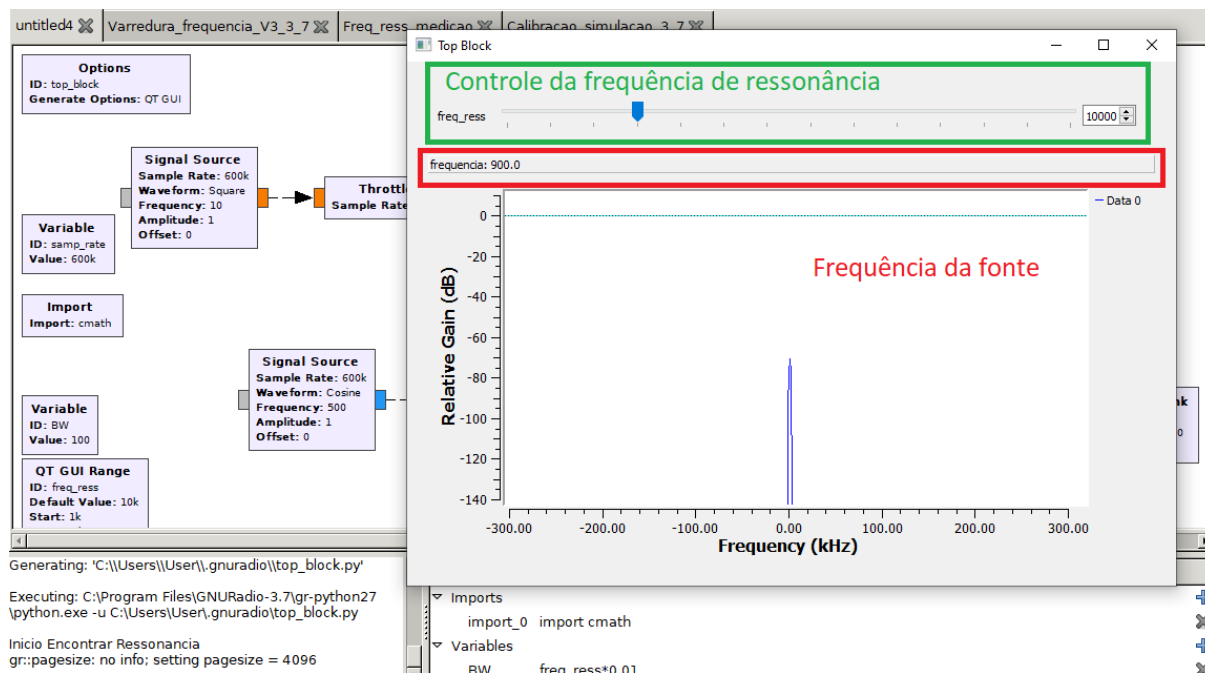
Fonte: O autor

Figura 64 – Arquivo gráfico obtido a partir da simulação da rotina de registro de *offset*.



Fonte: O autor

Figura 65 – Início da rotina de encontrar a ressonância.



Fonte: O autor

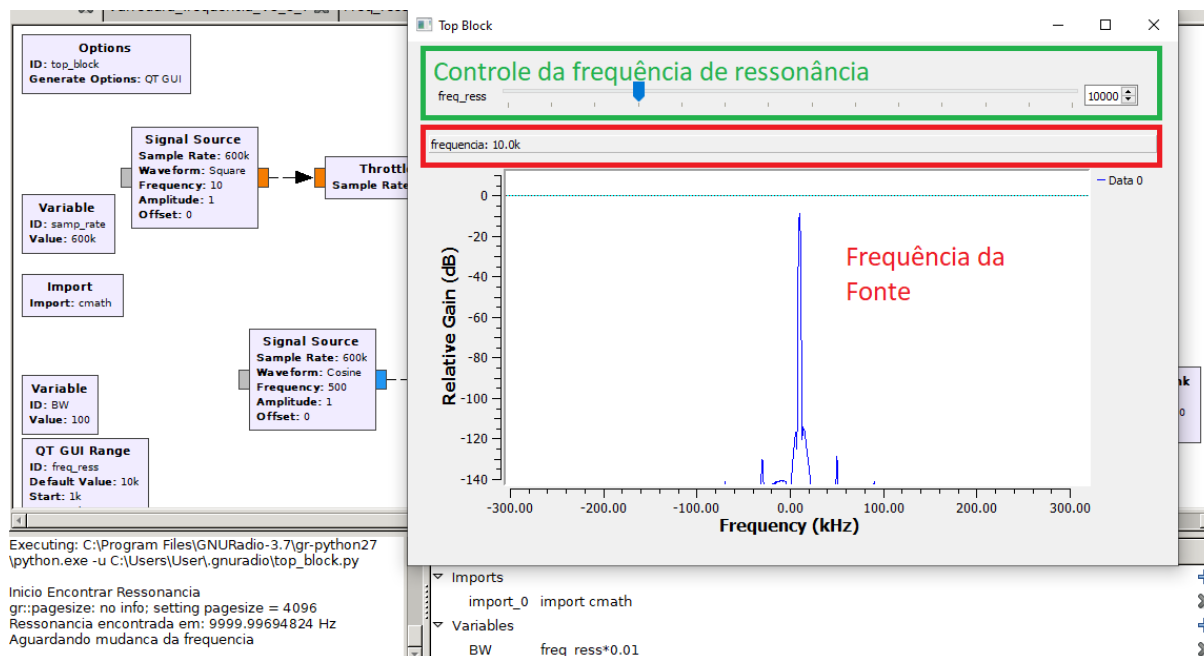
## C.2 Detecção e Rastreamento de Ressonância

Nesta simulação, foram implementadas as rotinas de detecção de ressonância e de rastreamento de ressonância no bloco de programação customizada. Como forma de verificá-las, foi implementado o diagrama de blocos do GNURadio localizado na figura 31. O sensor foi simulado pelo bloco de multiplicação, que simula o efeito de um filtro passa faixa, podendo alterar a sua frequência de ressonância através de um botão deslizante (*slider*), localizado no topo da janela de simulação gerada (*Top Block*). A janela também mostra o espectro do sinal após o filtro simulado.

A figura 65 mostra o início do programa, com o espectro próximo a zero e com a variável que determina a frequência central do filtro fixada em 10 kHz. A 66 mostra o programa término da detecção da ressonância, com a frequência do sinal igual a frequência central do filtro. Nesse estágio, o algoritmo aguarda até que a frequência de ressonância seja alterada para voltar a alterar a frequência. A detecção da frequência de ressonância ocorre por uma detecção da variação de fase de saída do filtro.

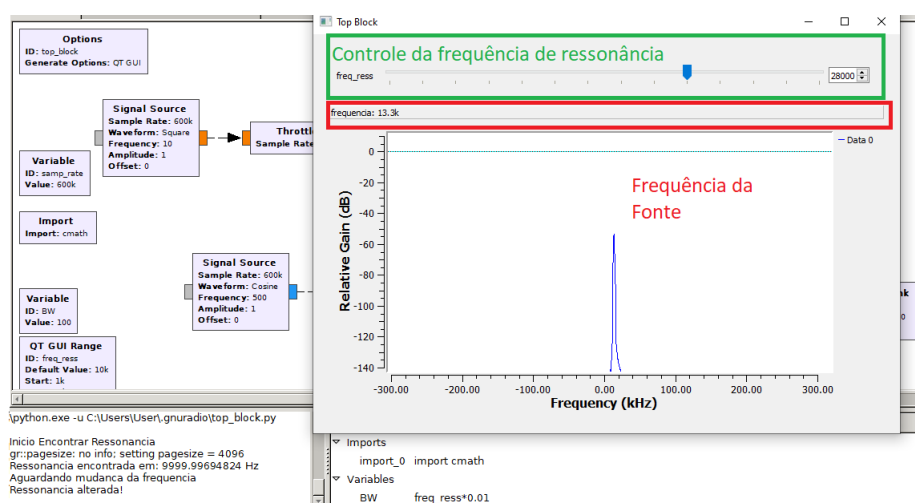
Modificando a frequência de ressonância através do botão, novo valor da frequência de ressonância se encontra em 40 kHz e o programa passa a rastrear a nova frequência de ressonância, exposto na figura 67. O algoritmo segue ajustando a frequência até encontra-lá novamente, conforme mostra a figura 68.

Figura 66 – Final da rotina de encontrar a ressonância e em espera da alteração da frequência de ressonância.



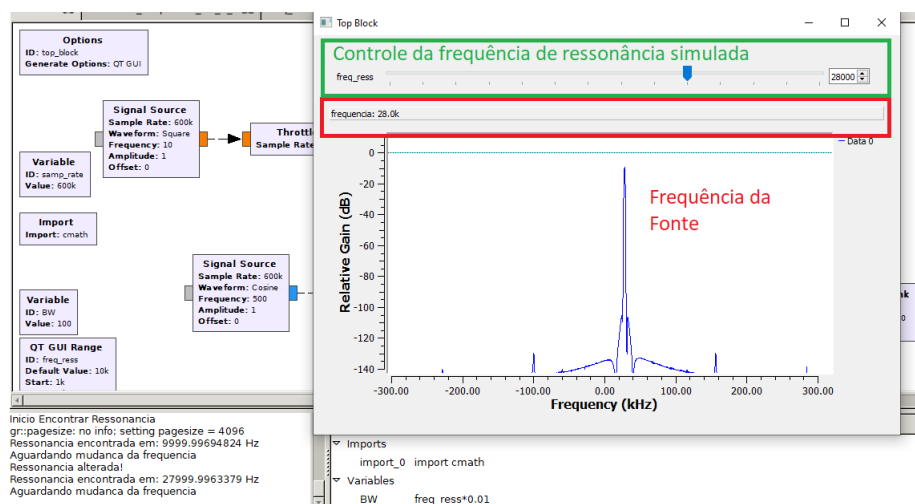
Fonte: O autor

Figura 67 – Alteração da frequência de ressonância detectada pelo algoritmo em simulação.



Fonte: O autor

Figura 68 – O algoritmo simulado finaliza a busca pela nova frequência de ressonância.



Fonte: O autor

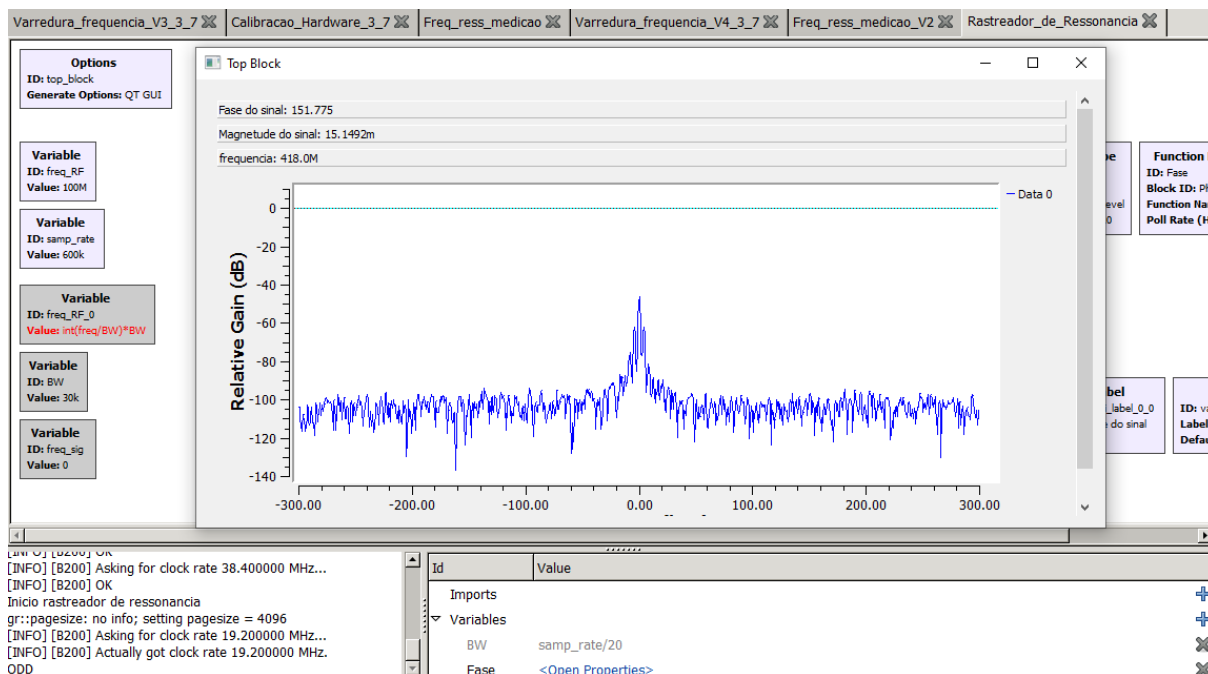
## APÊNDICE D – Rastreo de ressonância do sensor A

Neste apêndice, será mostrado como ocorreu a execução da rotina de rastreo de ressonância implementada com a bancada de ensaios. O programa utilizou a implementação das rotinas de detecção e rastreo de ressonância no bloco de programação em Python. A variação da frequência de ressonância foi realizada através da inserção ou remoção completa da água destilada, alternando assim entre os cenários 1 e 4. A janela de saída (*Top Block*) expõe a frequência dos osciladores, a magnitude e fase resultante da multiplicação conjugada dos sinais recebidos pelos canais TX0-RX0 e TX1-RX1 e uma visualização gráfica da amplitude.

A figura 69 mostra o início da rotina, iniciando a varredura menor. A figura 70 mostra o encontro da frequência de ressonância através da rotina de detecção de frequência e aguardando a inserção do material para alterar a frequência de ressonância.

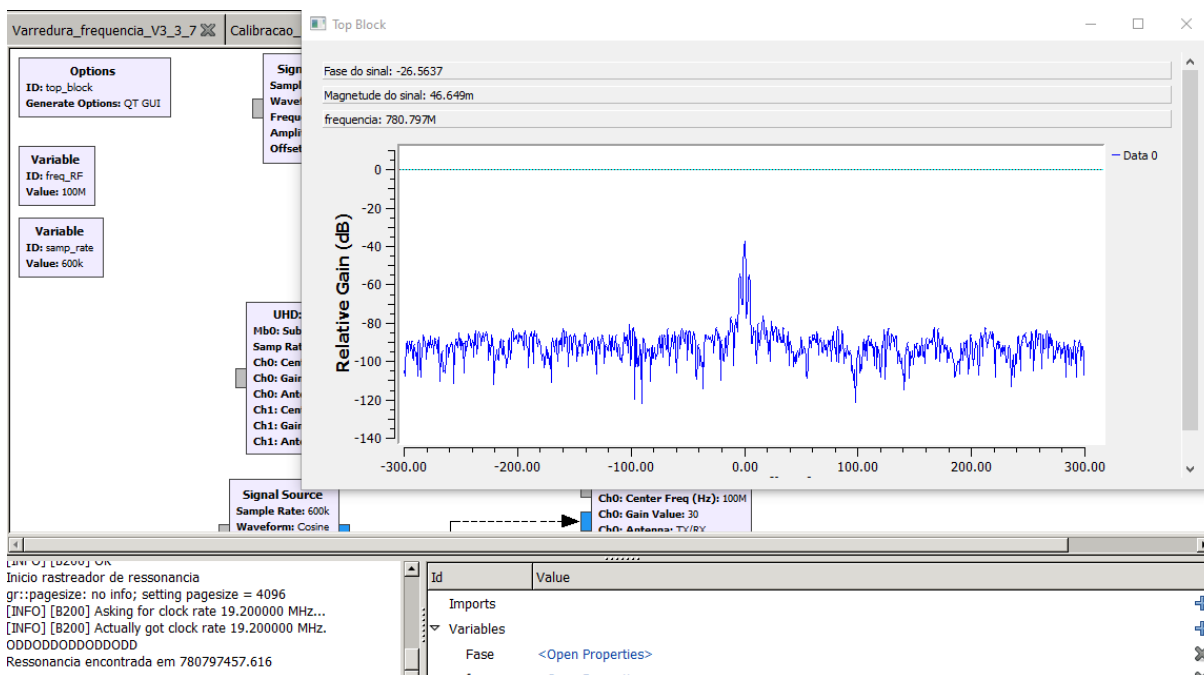
A figura 71 mostra o programa detectando a alteração da frequência de ressonância e iniciando a busca pelo seu novo valor. A rotina busca a frequência de ressonância até chegar a figura 59, localizada na seção 4.5.

Figura 69 – Início da rotina de rastreo da ressonância.



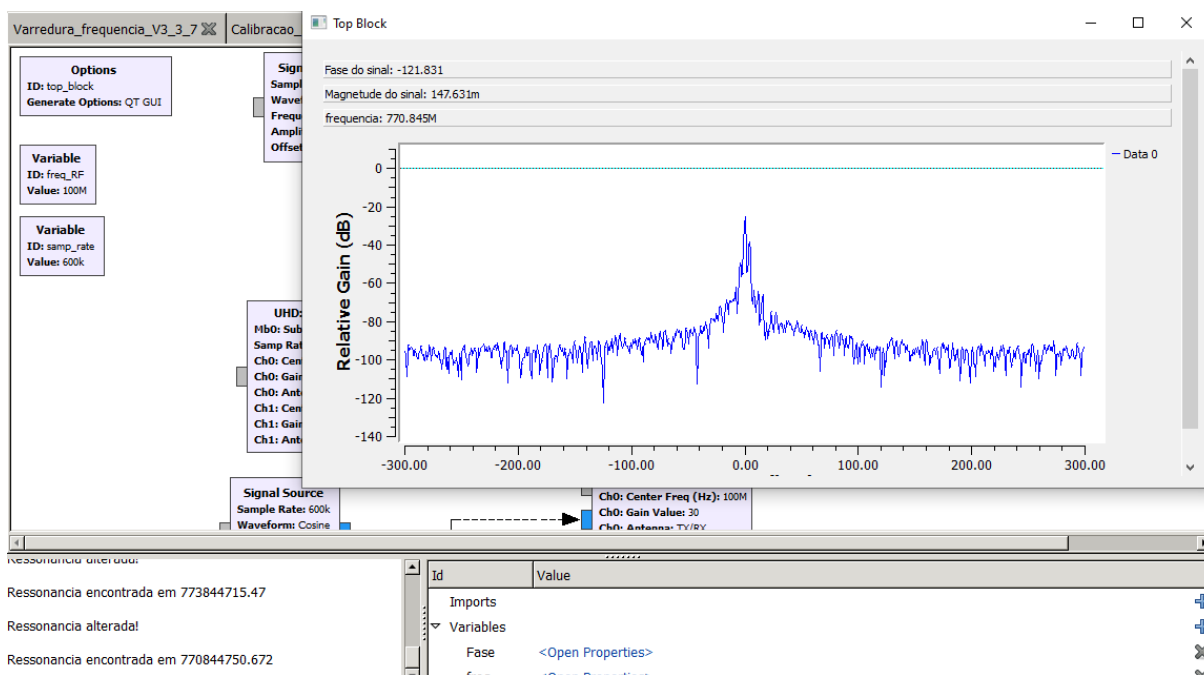
Fonte: O autor

Figura 70 – Fase intermediária da rotina de rastreo da ressonância, tendo encontrado a resposta com o sensor descoberto.



Fonte: O autor

Figura 71 – Fase intermediária da rotina de rastreo da ressonância, detectado uma alteração na frequência de ressonância



Fonte: O autor