

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA AEROESPACIAL

José Eduardo Zanon

**ESTAÇÃO TERRENA RECEPTORA PARA NANOSATÉLITES DE BAIXA  
ÓRBITA - IMPLEMENTAÇÃO ATRAVÉS DE RÁDIO DEFINIDO POR  
SOFTWARE**

Santa Maria, RS  
2023

**José Eduardo Zanon**

**ESTAÇÃO TERRENA RECEPTORA PARA NANOSATÉLITES DE BAIXA ÓRBITA -  
IMPLEMENTAÇÃO ATRAVÉS DE RÁDIO DEFINIDO POR SOFTWARE**

Tese de Graduação apresentada ao Curso de Graduação em Engenharia Aeroespacial, Área de Concentração em Engenharia, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro em Engenharia Aeroespacial**. Defesa realizada por videoconferência.

ORIENTADOR: Prof. Fernando César Comparsi de Castro

Santa Maria, RS  
2023

---

©2023

Todos os direitos autorais reservados a José Eduardo Zanon. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: [jose.zanon@acad.ufsm.br](mailto:jose.zanon@acad.ufsm.br)

**José Eduardo Zanon**

**ESTAÇÃO TERRENA RECEPTORA PARA NANOSATÉLITES DE BAIXA ÓRBITA -  
IMPLEMENTAÇÃO ATRAVÉS DE RÁDIO DEFINIDO POR SOFTWARE**

Tese de Graduação apresentada ao Curso de Graduação em Engenharia Aeroespacial, Área de Concentração em Engenharia, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Engenheiro em Engenharia Aeroespacial**.

**Aprovado em 21 de julho de 2023:**

---

**Fernando César Comparsi de Castro, Dr. (UFSM)**  
(Orientador)

---

**Banca Um, Dr. (Natanael Rodrigues Gomes) (videoconferência)**

---

**Banca Dois, Dr. (Eduardo Escobar Bürger ) (videoconferência)**

Santa Maria, RS  
2023

## RESUMO

# ESTAÇÃO TERRENA RECEPTORA PARA NANOSATÉLITES DE BAIXA ÓRBITA - IMPLEMENTAÇÃO ATRAVÉS DE RÁDIO DEFINIDO POR SOFTWARE

AUTOR: José Eduardo Zanon

ORIENTADOR: Fernando César Comparsi de Castro

Com a popularização do rádio definido por software, SDR, cada vez mais aplicações estão sendo desenvolvidas de forma simplificada por meio de software ao invés de hardware. Diante do contexto exposto, existem inúmeras aplicações relacionadas ao segmento de TTeC para nanossatélites.

Tendo em vista este cenário, a presente pesquisa tem como objetivo propor o projeto de uma estação terrena receptora implementada através de rádio definido por software. Para cumprir com este objetivo uma metodologia foi definida, bem como inúmeros objetivos diversos que vão desde o orçamento de link ao teste dos demoduladores e decodificadores, utilizando softwares como MATLAB, 4NEC2, MATHCAD e GNURadio. Como resultado, um *link budget* foi definido para o enlace entre estação terrena e nanossatélite, uma antena yagi-uda foi projetada através de um dimensionamento e simulação, construída e testada preliminarmente em laboratório. Além disso, dois demoduladores foram propostos para trabalhar com os esquemáticos de modulação citados no trabalho, bem como houve a implementação do segmento de decodificação. Ainda, foi desenvolvida uma metodologia para trabalhar com *beacons* no GNURadio, resultando em dados telemétricos enviados pelo Tx de um *cubeSat*. Por último, houve o contato com um *cubesat* presente em órbita baixa, integrando os segmentos do SDR, antena e computador.

**Palavras-chave:** SDR. Estação terrena. Nanosatélite. (...)

## ABSTRACT

### GROUND STATION FOR LOW ORBIT NANOSATELLITES - IMPLEMENTATION BY SOFTWARE DEFINED RADIO

AUTHOR: José Eduardo Zanon

ADVISOR: Fernando César Comparsi de Castro

With the popularization of software-defined radio, more and more applications are being developed in a simplified way through software instead of hardware. Given the context, there are numerous applications related to the TTeC segment of nanosatellites. In this scenario, the present research aims to propose the project of a receiving ground station implemented through software defined radio. To fulfill this objective, a methodology was defined, as well as numerous tasks ranging from the link budget to the test of demodulators, using software res such as MATLAB, 4NEC2, MATHCAD and GNU Radio. As a result, a link budget was defined for the link between ground station and nanosatellite, a yagi-uda antenna was designed, built and preliminarily tested in the laboratory as well as the test and implementation of demodulators and decoders in the GNURadio software. Two demodulators were proposed to work with modulations schemes used in the dissertation as well as the implementation of the decoding segment. Also, a methodology was developed to work with *beacons* in GNURadio, resulting in telemetry data sent by an cubesat. Finally, there was contact with a cubesat present in low orbit, integrating the SDR, antenna and computer segments

**Keywords:** SDR. Ground Station. Nanosatellite. (...)

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 2.1 – Processo de modulação analógica. ....   | 17 |
| Figura 2.2 – <i>Mixer</i> .....  | 17 |
| Figura 2.3 – Diagrama Polar. ....  | 18 |
| Figura 2.4 – Constelação BPSK. ....  | 19 |
| Figura 2.5 – Sinal unipolar. ....  | 19 |
| Figura 2.6 – Modulação <i>Frequency-Shift Keying</i> .....   | 20 |
| Figura 2.7 – Constelação MSK. ....   | 21 |
| Figura 2.8 – Modulador IQ GMSK. ....   | 23 |
| Figura 2.9 – Demodulador de quadratura GMSK implementado em uma placa FPGA. ....                       | 23 |
| Figura 2.10 – Limiar de decisão da modulação 2 - PSK com $M = 2$ símbolos. ....                        | 25 |
| Figura 2.11 – Interação do sincronismo de símbolo no presente trabalho. ....                           | 26 |
| Figura 2.12 – Impacto do sincronismo de símbolo. ....  | 26 |
| Figura 2.13 – Diferença de <i>clock</i> entre Tx e Rx. ....  | 27 |
| Figura 2.14 – Loop de fase fechada. ....   | 28 |
| Figura 2.15 – <i>Timing error detector (TED)</i> .....   | 28 |
| Figura 2.16 – Resposta ao impulso do símbolo coma atraso $f(\tau)0$ .....                              | 29 |
| Figura 2.17 – Resposta ao impulso gaussiana. ....  | 32 |
| Figura 2.18 – Geometria de interação entre satélite e terra. ....                                      | 34 |
| Figura 2.19 – BER VS SNR para a modulação GMSK codificada. ....  | 37 |
| Figura 2.20 – Configuração 1U. ....  | 39 |
| Figura 2.21 – Subsistemas do FLORIPASAT-1 .....  | 40 |
| Figura 2.22 – Antena derivada do circuito de Thévenin. ....  | 41 |
| Figura 2.23 – Diagrama Linear do Padrão de Radiação. ....  | 43 |
| Figura 2.24 – Sistemas de coordenadas espaciais. ....  | 44 |
| Figura 2.25 – Perfis de radiação comuns. ....  | 44 |
| Figura 2.26 – Polarização do campo elétrico. ....  | 45 |
| Figura 2.27 – Yagi-Uda .....   | 47 |
| Figura 2.28 – Relação frente-costas. ....  | 48 |
| Figura 2.29 – Interação do protocolo no sistema comunicação digital de um transmissor. ....            | 50 |
| Figura 2.30 – <i>Frame</i> pertencente ao protocolo AX.25 .....  | 50 |
| Figura 2.31 – Camadas de comunicação desenvolvidas para a constelação de <i>cube-sats VCC</i> . ....   | 52 |
| Figura 2.32 – Interação entre os <i>frames</i> do AX.25, HDLC e KISS. ....                             | 53 |
| Figura 2.33 – Estrutura do UI <i>frame</i> .....   | 54 |
| Figura 2.34 – NGham <i>frame</i> .....   | 55 |
| Figura 2.35 – <i>Frames</i> do AX.25 transportados pelo KISS .....                                     | 56 |
| Figura 2.36 – <i>Frames</i> do protocolo KISS .....  | 56 |
| Figura 2.37 – Interação do codificador de canal no sistema comunicação digital de um transmissor. .... | 58 |
| Figura 2.38 – Codificador de linha. ....   | 59 |
| Figura 2.39 – Codificador de canal. ....   | 60 |
| Figura 2.40 – Estrutura da palavra código. ....  | 61 |
| Figura 2.41 – Estrutura da palavra código RS(255,223) .....  | 63 |
| Figura 3.1 – Metodologia Adotada. ....   | 68 |

|  |     |
|--|-----|
| Figura 4.1 – Diagrama de blocos representando a interação entre os sistemas. ....                            | 70  |
| Figura 4.2 – Arranjo com duas antenas yagi circularmente polarizadas. ....                                   | 74  |
| Figura 4.3 – <i>gamma match</i> . ....   | 75  |
| Figura 4.4 – Sistema de comunicação digital. ....  | 77  |
| Figura 4.5 – Fluxograma dos processos referentes a demodulação. ....   | 78  |
| Figura 4.6 – GMSK Mod. ....  | 80  |
| Figura 4.7 – GMSK Demod. ....  | 80  |
| Figura 4.8 – Demodulador de quadratura. ....   | 81  |
| Figura 4.9 – Binary Slicer. ....   | 82  |
| Figura 4.10 – AX-100 Deframer. ....  | 84  |
| Figura 4.11 – NGHam Deframer ....  | 84  |
| Figura 4.12 – Metodologia referente ao processamento de sinais. ....   | 85  |
| Figura 5.1 – Dipolo de meia onda. ....   | 88  |
| Figura 5.2 – Yagi UHF para 4 elementos modelada no MATLAB. ....  | 89  |
| Figura 5.3 – Diagrama de radiação da antena UHF de 15 elementos - Polarização horizontal ....                | 90  |
| Figura 5.4 – Diagrama de radiação da antena UHF de 4 elementos - Polarização Horizontal. ....                | 90  |
| Figura 5.5 – Coordenadas no 4NEC2 para o projeto da yagi cruzada. ....                                       | 91  |
| Figura 5.6 – Diagrama de radiação para as polarizações horizontal e vertical. ....                           | 92  |
| Figura 5.7 – Visualização da antena no 4NEC2 ....  | 93  |
| Figura 5.8 – Antena em construção. ....  | 95  |
| Figura 5.9 – Casador Construído. ....  | 96  |
| Figura 5.10 – Alimentação da Antena ....   | 96  |
| Figura 5.11 – Bancada de teste. ....   | 97  |
| Figura 5.12 – Parâmetro S11. ....  | 98  |
| Figura 5.13 – Fluxograma desenvolvido ....   | 99  |
| Figura 5.14 – Blocos <i>random source</i> e <i>pack k bits</i> . ....  | 99  |
| Figura 5.15 – Bits gerados pelo <i>random source</i> . ....  | 100 |
| Figura 5.16 – <i>Mapper</i> GMSK. ....   | 100 |
| Figura 5.17 – <i>Mapper com ruído AWGN</i> ....  | 101 |
| Figura 5.18 – Sinal de linha do demodulador de quadratura. ....  | 101 |
| Figura 5.19 – Sinal de linha do demodulador GMSK. ....   | 102 |
| Figura 5.20 – Dispersão reduzida pós sincronismo de símbolo ....   | 102 |
| Figura 5.21 – Demodulador GMSK implementado em uma FPGA vs demodulador proposto no GNURadio pelo autor. .... | 104 |
| Figura 5.22 – Resposta em frequência do filtro. ....   | 104 |
| Figura 5.23 – Desenho 3D do BeliefSat-1 ....   | 106 |
| Figura 5.24 – Estrutura dos CCSDS Reed Solomon <i>frames</i> . ....  | 107 |
| Figura 5.25 – Interface do demodulador+decodificador do BeliefSat-1 ....                                     | 108 |
| Figura 5.26 – Espectro do sinal originalmente recebido. ....   | 108 |
| Figura 5.27 – Espectro do sinal após uso do <i>frequency xlating fir filter</i> ....                         | 109 |
| Figura 5.28 – Espectro do sinal após-refiltragem. ....   | 110 |
| Figura 5.29 – Sinal RZ demodulado. ....  | 110 |
| Figura 5.30 – Visualização dos símbolos IQ sincronizados pelo bloco <i>symbol sync</i> . ..                  | 111 |
| Figura 5.31 – Símbolos IQ recuperados oriundos do demodulador FSK. ....                                      | 111 |
| Figura 5.32 – Decodificador OOT. ....  | 112 |
| Figura 5.33 – CCSDS Reed Solomon TM <i>frames</i> decodificados. ....  | 112 |



|   |     |
|---|-----|
| Figura 5.34 – Espectro do <i>beacon</i> recebido. ....  | 114 |
| Figura 5.35 – Espectro filtrado. ....   | 115 |
| Figura 5.36 – Processos realizados equivalentes a figura 5.37. ....                                 | 115 |
| Figura 5.37 – Fluxograma construído com o módulo OOT ....   | 116 |
| Figura 5.38 – Diagrama contendo o decodificador proposto. ....                                      | 116 |
| Figura 5.39 – Fluxograma para sincronismo de símbolo e decodificação desenvolvido. ....             | 116 |
| Figura 5.40 – Símbolos IQ recuperados pelo <i>symbol sync</i> . ....                                | 117 |
| Figura 5.41 – <i>Correlate access code tag</i> . ....   | 117 |
| Figura 5.42 – Bloco <i>sync and create PDU</i> . ....   | 118 |
| Figura 5.43 – Código CRC. ....  | 118 |
| Figura 5.44 – Decodificador Reed Solomon ....   | 119 |
| Figura 5.45 – <i>Frames do FLORIPASAT-1 decodificados em formato hexadecimal</i> . ....             | 119 |
| Figura 5.46 – <i>Telemetry Parser</i> . ....  | 119 |
| Figura 5.47 – Dados contidos no <i>payload</i> visualizados através do <i>telemetry parser</i> . .. | 120 |
| Figura 5.48 – Dados contidos no <i>beacon</i> . ....  | 120 |
| Figura 5.49 – Fluxograma construído para sincronismo de símbolo e decodificação ..                  | 121 |
| Figura 5.50 – Espectro originalmente recebido vs espectro filtrado. ....                            | 122 |
| Figura 5.51 – Visualização dos símbolos IQ sincronizados pelo bloco <i>Symbol Sync</i> . .          | 122 |
| Figura 5.52 – <i>ASM+Golay frames</i> decodificados. ....   | 123 |
| Figura 5.53 – <i>Telemetry Viewer</i> do AlfaCrux. ....   | 123 |
| Figura 5.54 – Adaptador <i>pigtail</i> e SDR. ....  | 124 |
| Figura 5.55 – Interface do N2YO. ....   | 125 |
| Figura 5.56 – Antena + SDR + Computador. ....   | 125 |
| Figura 5.57 – Apontamento manual em campo. ....   | 126 |
| Figura 5.58 – Configurações do receptor. ....   | 126 |
| Figura 5.59 – <i>Scheduler</i> do SDRUno. ....  | 127 |
| Figura 5.60 – Espectro recebido. ....   | 127 |
| Figura 5.61 – <i>beacon</i> recebido. ....  | 127 |

## LISTA DE TABELAS

|   |     |
|---|-----|
| Tabela 2.1 – Dimensões de configurações comuns.....   | 39  |
| Tabela 2.2 – SWR vs Potência refletida. ....  | 46  |
| Tabela 3.1 – Cronograma.....  | 69  |
| Tabela 4.1 – Tabela de projeto adaptada. ....   | 71  |
| Tabela 4.2 – Tabela confeccionada com base nas referências (Space Lab UFSC, 2022) e (LodeStar Aerospace, 2022)..... | 71  |
| Tabela 4.3 – Parâmetros dos elementos das antenas UHF.....  | 73  |
| Tabela 4.4 – Informações referentes à modulação e camada de enlace de dados ....                                    | 77  |
| Tabela 5.1 – Parâmetros de cálculo para o <i>link budget</i> .....  | 86  |
| Tabela 5.2 – <i>Link budget</i> .....   | 87  |
| Tabela 5.3 – Características das antenas UHF.....   | 91  |
| Tabela 5.4 – SWR, impedância e ganho da antena.....   | 92  |
| Tabela 5.5 – Parâmetros para definição do <i>gamma match</i> .....  | 93  |
| Tabela 5.6 – <i>Gamma Match</i> .....   | 94  |
| Tabela 5.7 – Resultados obtidos no analisador de rede vetorial.....   | 97  |
| Tabela 5.8 – Especificações do BeliefSat-1 .....  | 107 |

## LISTA DE ABREVIATURAS E SIGLAS

|             |                                      |
|-------------|--------------------------------------|
| <i>ADC</i>  | Analog to Digital Converter          |
| <i>AWGN</i> | Additive White Gaussian Noise        |
| <i>BER</i>  | Bit-Error Rate                       |
| <i>BPSK</i> | Binary Phase Shift Keying            |
| <i>BT</i>   | Bandwidth-Time Product               |
| <i>COTS</i> | Commercial off the Shelf             |
| <i>CSP</i>  | CubeSat Space Protocol               |
| <i>DSP</i>  | Digital Signal Processing            |
| <i>FIR</i>  | Finite Impulse Response              |
| <i>FNBW</i> | Largura de Feixe dos Primeiros Nulos |
| <i>FPGA</i> | Field Programed Gate Array           |
| <i>FSK</i>  | Frequency Shift Keying               |
| <i>GMSK</i> | Gaussian Minimum Shift Keying        |
| <i>GF</i>   | Galois Field                         |
| <i>HPBW</i> | Largura de Feixe de Meia Potência    |
| <i>IQ</i>   | In-Phase and In-Quadrature           |
| <i>ISI</i>  | Interferência Intersimbólica         |
| <i>KISS</i> | Keep it Simple Stupid                |
| <i>LEO</i>  | Low Earth Orbit                      |
| <i>LFS</i>  | Free Space Loss                      |
| <i>LFSR</i> | Linear Feedback Shift Register       |
| <i>LHCP</i> | Left-Hand Circular Polarization      |
| <i>LUT</i>  | Look Up Table                        |
| <i>MSK</i>  | Minimum Shift Keying                 |
| <i>NRZ</i>  | Non Return to Zero                   |
| <i>OOT</i>  | Out Of Tree                          |
| <i>PAM</i>  | Pulse Amplitude Modulation           |

|             |                                  |
|-------------|----------------------------------|
| <i>PDU</i>  | Protocol Data Unit               |
| <i>PSK</i>  | Phase Shift Keying               |
| <i>RHCP</i> | Right-Hand Circular Polarization |
| <i>Rx</i>   | Receptor                         |
| <i>RS</i>   | Reed Solomon                     |
| <i>SBC</i>  | Single Board Computer            |
| <i>SDR</i>  | Software Defined Radio           |
| <i>SNR</i>  | Signal Noise Ratio               |
| <i>SWR</i>  | Standing Wave Ratio              |
| <i>TED</i>  | Timing Error Detector            |
| <i>TTC</i>  | Telemetria Telecomando e Rastreo |
| <i>Tx</i>   | Transmissor                      |

## SUMÁRIO

|              |  |           |
|--------------|--|-----------|
| <b>1</b>     | <b>INTRODUÇÃO</b>                                    | <b>13</b> |
| 1.1          | OBJETIVOS  | 14        |
| <b>2</b>     | <b>REVISÃO BIBLIOGRÁFICA</b>                         | <b>15</b> |
| 2.1          | SDR - <i>SOFTWARE DEFINED RADIO</i>                  | 15        |
| 2.2          | GNURADIO   | 16        |
| 2.3          | MODULAÇÃO  | 16        |
| <b>2.3.1</b> | <b>Modulação GMSK</b>                                | <b>22</b> |
| <b>2.3.2</b> | <b>Demodulador de quadratura GMSK</b>                | <b>23</b> |
| 2.4          | SINCRONISMO DE SÍMBOLO - RECUPERAÇÃO DE <i>CLOCK</i> | 25        |
| <b>2.4.1</b> | <b>PLL</b>   | <b>28</b> |
| 2.4.1.1      | <i>Mueller-Muller TED</i>                            | 29        |
| 2.4.1.2      | <i>Gardner TED</i>                                   | 30        |
| 2.5          | FILTRO FIR   | 31        |
| 2.6          | <i>LINK BUDGET</i>                                   | 32        |
| <b>2.6.1</b> | <b>Perda de caminho livre - LFS</b>                  | <b>34</b> |
| <b>2.6.2</b> | <b>Atenuação devido a chuva</b>                      | <b>35</b> |
| <b>2.6.3</b> | <b>Figura de ruído do receptor</b>                   | <b>35</b> |
| <b>2.6.4</b> | <b>Cintilação Ionosférica</b>                        | <b>36</b> |
| <b>2.6.5</b> | <b>Apontamento</b>                                   | <b>36</b> |
| <b>2.6.6</b> | <b>Relação Sinal-Ruído</b>                           | <b>37</b> |
| <b>2.6.7</b> | <b>Polarização</b>                                   | <b>38</b> |
| <b>2.6.8</b> | <b>Efeito Doppler</b>                                | <b>38</b> |
| 2.7          | <i>CUBESAT</i>                                       | 38        |
| 2.8          | TEORIA DE ANTENAS                                    | 40        |
| <b>2.8.1</b> | <b>Diretividade, Ganho e Largura de Feixe</b>        | <b>42</b> |
| <b>2.8.2</b> | <b>Ressonância</b>                                   | <b>43</b> |
| <b>2.8.3</b> | <b>Perfil de radiação</b>                            | <b>43</b> |
| <b>2.8.4</b> | <b>Polarização</b>                                   | <b>45</b> |
| <b>2.8.5</b> | <b>Relação de Onda Estacionária</b>                  | <b>45</b> |
| <b>2.8.6</b> | <b>Casamento de impedância</b>                       | <b>46</b> |
| <b>2.8.7</b> | <b>Antena Yagi-Uda</b>                               | <b>47</b> |
| 2.8.7.1      | <i>Relação Frente-Costas</i>                         | 47        |
| 2.8.7.2      | <i>Arranjo Parasita</i>                              | 48        |
| 2.8.7.3      | <i>Gamma Match</i>                                   | 49        |
| 2.9          | PROTOCOLO DE COMUNICAÇÃO                             | 50        |
| 2.9.0.1      | <i>Camada física</i>                                 | 51        |
| 2.9.0.2      | <i>Camada de Enlace de Dados</i>                     | 51        |
| 2.9.0.3      | <i>Camada de Rede</i>                                | 51        |
| 2.9.0.4      | <i>Camada de Transporte</i>                          | 52        |
| <b>2.9.1</b> | <b>Protocolo AX.25</b>                               | <b>53</b> |
| <b>2.9.2</b> | <b>Protocolo NGHAM</b>                               | <b>54</b> |
| <b>2.9.3</b> | <b>Protocolo KISS</b>                                | <b>55</b> |
| <b>2.9.4</b> | <b>Protocolo CSP</b>                                 | <b>56</b> |

|               |  |            |
|---------------|--|------------|
| 2.10          | CODIFICAÇÃO DE CANAL .....   | 58         |
| <b>2.10.1</b> | <b>Código Corretor de Erro .....</b>                                       | <b>60</b>  |
| 2.10.1.1      | <i>Reed-Solomon (RS) .....</i>   | 63         |
| 2.10.1.2      | <i>Golay.....</i>  | 65         |
| <b>3</b>      | <b>METODOLOGIA .....</b>   | <b>67</b>  |
| 3.1           | CENÁRIO.....   | 67         |
| 3.2           | MÉTODO DE PESQUISA .....   | 67         |
| 3.3           | ETAPAS DA PESQUISA .....   | 67         |
| <b>4</b>      | <b>DESENVOLVIMENTO .....</b>   | <b>70</b>  |
| 4.1           | DIMENSIONAMENTO DA ESTAÇÃO.....  | 70         |
| <b>4.1.1</b>  | <b><i>Link Budget</i> .....</b>  | <b>71</b>  |
| 4.2           | ANTENA RECEPTORA - YAGI UDA.....   | 72         |
| <b>4.2.1</b>  | <b>Dimensionamento Preliminar da Antena .....</b>                          | <b>72</b>  |
| <b>4.2.2</b>  | <b>437.01 MHZ <i>Cross Yagi</i> .....</b>                                  | <b>73</b>  |
| <b>4.2.3</b>  | <b>Casamento de Impedância - <i>Gamma Match</i>.....</b>                   | <b>74</b>  |
| 4.3           | IMPLEMENTAÇÃO DO DEMODULADOR E DECODIFICADOR UTILIZANDO<br>GNURADIO .....  | 76         |
| <b>4.3.1</b>  | <b>GNU Radio OOT.....</b>  | <b>77</b>  |
| <b>4.3.2</b>  | <b>Demodulador - GNU Radio .....</b>                                       | <b>78</b>  |
| 4.3.2.1       | <i>GMSK Mod. ....</i>  | 80         |
| 4.3.2.2       | <i>GMSK Demod.....</i>   | 80         |
| 4.3.2.3       | <i>Demodulador de Quadratura .....</i>                                     | 81         |
| 4.3.2.4       | <i>Detector.....</i>   | 82         |
| <b>4.3.3</b>  | <b>Decodificador .....</b>   | <b>82</b>  |
| 4.3.3.1       | <i>AX-100 Deframer .....</i>   | 83         |
| 4.3.3.2       | <i>NGHam Deframer.....</i>   | 84         |
| <b>4.3.4</b>  | <b>Metodologia para processamento de sinais no GNU Radio .....</b>         | <b>84</b>  |
| <b>5</b>      | <b>TESTES E RESULTADOS EXPERIMENTAIS.....</b>                              | <b>86</b>  |
| 5.1           | <i>LINK BUDGET .....</i>   | 86         |
| 5.2           | DIMENSIONAMENTO PRELIMINAR E SIMULAÇÃO DA ANTENA .....                     | 88         |
| 5.3           | CONSTRUÇÃO DA ANTENA .....   | 94         |
| <b>5.3.1</b>  | <b>Resultados Preliminares da Antena.....</b>                              | <b>97</b>  |
| 5.4           | DEMODULADOR E DECODIFICADOR - GNURADIO .....                               | 99         |
| <b>5.4.1</b>  | <b>Demodulação - GNURadio.....</b>   | <b>99</b>  |
| 5.4.1.1       | <i>Fluxograma dos Demoduladores .....</i>                                  | 99         |
| <b>5.4.2</b>  | <b>Demodulador Proposto .....</b>  | <b>103</b> |
| <b>5.4.3</b>  | <b>Processamento de um <i>stream</i> de símbolos IQ no GNU Radio .....</b> | <b>105</b> |
| <b>5.4.4</b>  | <b>Decodificando o FLORIPASAT-1 .....</b>                                  | <b>113</b> |
| <b>5.4.5</b>  | <b>Decodificando o AlfaCrux .....</b>                                      | <b>121</b> |
| 5.5           | IMPLEMENTAÇÃO EM CAMPO .....   | 124        |
| <b>6</b>      | <b>CONCLUSÃO.....</b>  | <b>129</b> |
|               | <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>                                     | <b>131</b> |

## 1 INTRODUÇÃO

Atualmente, com a redução do custo de implementação de hardware e software para mecanismos espaciais, há um crescimento exponencial em tecnologias de baixo custo para pesquisas espaciais, desencadeando assim uma corrida entre as mais diversas empresas do mundo (ALOUINI, 2020). Em virtude disso, a utilização de satélites de pequeno porte, *cubesats*, *cansats*, *pocketqubes* e *smallsats* lançados na órbita baixa terrestre constitui um dos principais meios de pesquisa e de comunicações de baixo custo, já que a maioria desses mecanismos podem ser construídos com componentes comerciais comuns específicos disponíveis e de relativo fácil acesso (NASA, 2017).

Nesse contexto, no meio acadêmico, um dos principais mecanismos de exploração espacial de baixo custo são os *cubesats* operados em órbita baixa terrestre, devido a menor distância com relação a terra, menor requisito de potência de transmissão e menor perda por propagação (KUMAR, 2015). Atualmente, grande parte das pesquisas implementadas visam testar novas tecnologias, observação terrestre e validar a comunicação entre satélite e terra de alguma forma já que grande parte dos projetos implementados por instituições acadêmicas contam com estações terrenas para rastreamento, telemetria e telecomando confeccionadas por meio de componentes comerciais ou *COTS*, diminuindo custo e aumentando acessibilidade de projeto (ELFVELIN, 2022). Isso se dá devido a simplificação do projeto por meio da utilização do rádio definido por software que executa a maioria dos processos de recepção e transmissão do sinal, além da propriedade de configuração do dispositivo por meio de software e integração com um computador (BRIDGES, 2014).

Por conseguinte, a estação terrena deve operar com uma certa flexibilidade no que se refere ao espectro de frequência das faixas UHF e VHF a uma baixa taxa de transmissão de dados, utilizando protocolos semelhantes aos da internet contendo dados telemétricos também chamados de *beacons*.

(ELFVELIN, 2022) expõe uma metodologia de projeto separada em três partes, um segmento referente a antena receptora, outro relacionado ao processamento de sinal e o último que integra os demais por meio de um computador. Dessa maneira, é importante discorrer sobre a antena utilizada para captação ou emissão, transceptor e computador para obtenção e análise de dados.

Partindo disso, o presente trabalho inicialmente dimensiona uma estação terrena passiva baseado na organização proposta por (Tsuruda, 2017), ou seja um sistema de recepção com seus parâmetros adequados para receber sinal de um ou mais *cubesats* já presentes em órbita utilizando uma organização da implementação do segmento terrestre similar a (ELFVELIN, 2022).

## 1.1 OBJETIVOS

O objetivo geral desse trabalho é apresentar o projeto de um sistema de recepção para satélites em órbita baixa terrestre. Para atingir esse objetivo em um primeiro momento foram definidos as seguintes metas:

- Extensa revisão bibliográfica referente a temática multidisciplinar do trabalho.
- Dimensionamento da estação baseado em *cubesats* já em órbita.
- Cálculo de *Link Budget*.
- Estudo, projeto, simulação, construção e teste em campo da antena proposta.
- Implementação e teste dos demoduladores e decodificadores - GNURadio.
- Operacionalização do segmento do receptor de forma indireta através de *beacons* no GNURadio.
- Implementação da estação terrena em campo.



## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 SDR - *SOFTWARE DEFINED RADIO*

O rádio definido por software é um sistema de comunicação cujos parâmetros de operação podem ser estabelecidos por software ao invés de hardware. Isto permite que parâmetros a respeito do transceptor possam ser modificados sem mudança de hardware e portanto há redução do custo de projeto de um sistema de radiofrequência (WYGLINSKI; PU, 2013). Algumas características do sistema de radiofrequência relacionadas à configuração e adaptação são de difícil mudança por meio de hardware, sendo de certa forma um limitante no projeto de um receptor de forma que se o mesmo deve operar em duas faixas de frequência não há adaptação sem algumas mudanças no hardware (WYGLINSKI; PU, 2013). Apesar disso ser um custoso limitante em um projeto, a utilização do SDR oferece benefícios em diferentes contextos de sistemas de recepção e transmissão haja vista que esses elementos são comercialmente disponíveis e configuráveis por meio de software.

Uma prova disso se dá na utilização desses componentes em satélites como o STS-107 e o *Mars Reconnaissance Orbiter (MRO)*, onde o SDR possui proteção para radiação solar, permitindo o aumento da performance do subsistema de comunicação desse através de menores restrições no processo de modulação baseado nas condições de transmissão, bandas de frequência e taxa de transmissão de dados, embora essa adaptação dinâmica e uso de múltiplas bandas deixe o sistema mais complexo e dependente de uma maior alimentação (BRIDGES, 2014).

Dessa forma, na prática, a utilização do componente acaba substituindo vários componentes físicos e permite flexibilidade de operação sendo implementado juntamente de um computador pessoal de forma que os módulos de software do mecanismo são executados em tempo real por microprocessadores que podem executar a demodulação e modulação de sinal (BARBIN, 2012).

Isto é, após o recebimento do sinal o SDR executa as etapas de amplificação, conversão de radiofrequência para a frequência intermediária ou mixagem, filtragem, conversão (análogo-digital) e conversão para banda-base. Ademais, o mesmo pode operar em uma faixa de frequências de 1 KHz à 2 GHz e pode ser implementado com o software SDRUno instalado em um computador ou integrado por meio do software GNURadio na versão 3.7.12. (SDRPLAY, 2020).

## 2.2 GNURADIO

O GNURadio Companion (GRC) é uma ferramenta acessível por vários sistemas operacionais que serve para criar fluxogramas de sistemas de comunicações funcionais. O software fornece blocos de processamento de sinal para implementação de softwares definidos por rádio e sistemas de processamento de sinais (GNU, 2020). Por isso, é muito utilizado em projetos visando a redução de custos, implementando processos que seriam executados em hardware por meio de software (NOGUEIRA, 2015). O programa possui uma prototipagem rápida que facilita o entendimento do fluxograma dos processos bem como uma análise simples de erros que aponta os erros de configuração do programa.

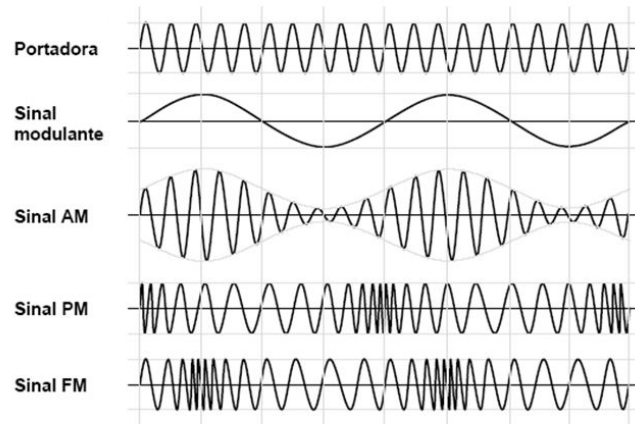
O GNU contém uma biblioteca com uma grande variedade de blocos que implementam várias funções distintas. Os blocos são separados em categorias distintas de acordo com a função, podendo implementar filtros, moduladores, operadores lógicos e matemáticos, demoduladores, decimadores, geradores de onda e várias outras funções relacionadas a comunicação digital e processamento de sinais (GNU, 2020).

## 2.3 MODULAÇÃO

A modulação é definida como o processo de alteração de um sinal, chamado de portadora, em função de outro sinal, chamado de sinal banda-base, que contém a informação para transmissão (HAYKIN, 1989). Esse processo é necessário devido às limitações de largura de banda dos canais de comunicação, estabelecidas por órgãos regulamentadores (AGILENT, 2002). Por exemplo, na transmissão de dados via satélite, a largura de banda do canal de comunicação para *cubesats* é regulamentada pela IARU e é de 20KHz (IARU, 2017). Esse requisito é devido ao comprimento de onda e frequência do sinal banda-base, que necessita de um sinal senoidal de alta frequência, normalmente chamado de portadora, para adequá-lo ao meio de comunicação (LATHI, 2010). Nesse contexto, a função da portadora é deslocar o espectro do sinal de banda-base para a banda-passante e essa possuirá uma frequência que, habitualmente, está no centro da largura de banda do canal.

Em geral, as características de frequência, amplitude ou fase da portadora são alteradas para transportar informação. No receptor, essa informação é recuperada através da detecção das alterações na portadora, por meio de um processo chamado demodulação (HAYKIN, 1989). Dentre as alterações das características da portadora, a mais comum é a frequência. Na modulação em frequência, a amplitude da portadora é mantida constante, enquanto a frequência varia em função do sinal modulador em banda-base.

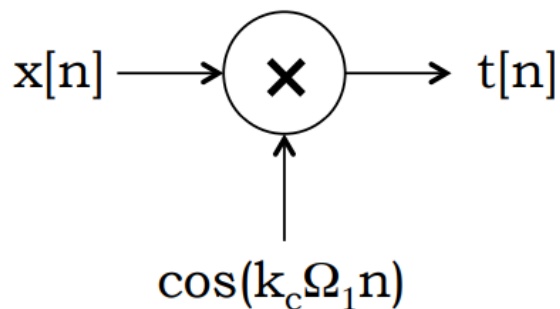
Figura 2.1 – Processo de modulação analógica.



Fonte: (TELECO, 2019)

Um dos conceitos principais envolvidos nesse processo é a heterodinação. Nela, a multiplicação de dois sinais senoidais pode ser escrita como a soma de ambos, mixando assim a moduladora com a portadora (MIT, 2012b). Conforme a figura 2.2, o sinal de entrada  $x[n]$  é multiplicado com a portadora  $\cos(k_c \Omega_1 n)$ , resultando no sinal modulado  $t[n]$ . Isso permite alternar a faixa de frequência para transmissão. Na prática, esse processo é realizado através de um *mixer* de frequência, que combina a portadora gerada em oscilador local controlado por tensão  $\cos(k_c \Omega_1, n)$ , onde  $\Omega_1$  é a frequência do sinal,  $k_c$  é o coeficiente não nulo da série de Fourier discretizada da portadora e  $n$  indica o sinal discretizado em amostras (MIT, 2012b). Esse sinal é combinado com o sinal em banda-base, por meio de um circuito elétrico composto de diodos, muito utilizado no esquemático de modulação FM (ELETRONICNOTES, 2011).

Figura 2.2 – Mixer



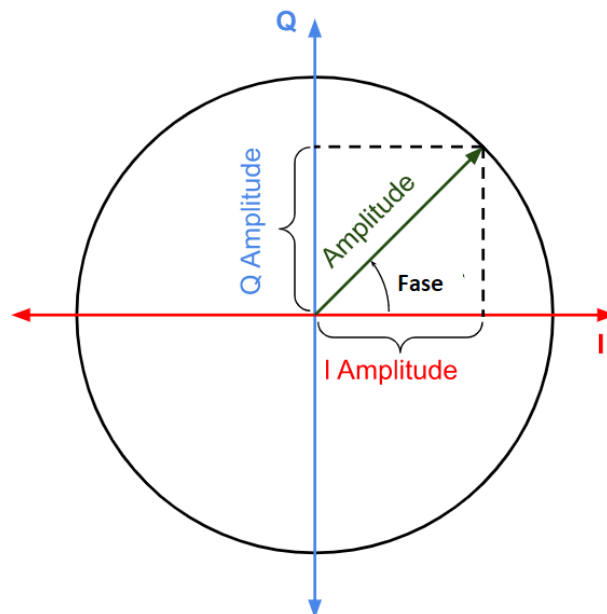
Fonte: (MIT, 2012b)

Associado ao tipo de informação que se deseja transmitir, se o sinal modulador for

contínuo no tempo, a modulação é analógica. Se o mesmo for um conjunto de símbolos ou bits, a modulação é digital (LATHI, 2010). Antes de adentrar nos tópicos relacionados à modulação digital, é importante abordar a respeito da representação dos sinais modulados em comunicações digitais, que são os sinais em fase (I) e em quadratura (Q), ou sinais complexos utilizados em diferentes aplicações além dos sistemas de comunicações digitais, tais como sistemas de radares (LYONS, 2008).

De acordo com a identidade de Euler, uma senoide pode ser representada como um fasor ou uma exponenciação complexa que possui as duas componentes, uma real e uma imaginária ortogonais. Isso permite definir um sinal IQ, que é um sinal que em um instante de tempo pode ser representado por um número complexo,  $i + j$ , com as componentes real e imaginária (LYONS, 2008).

Figura 2.3 – Diagrama Polar.



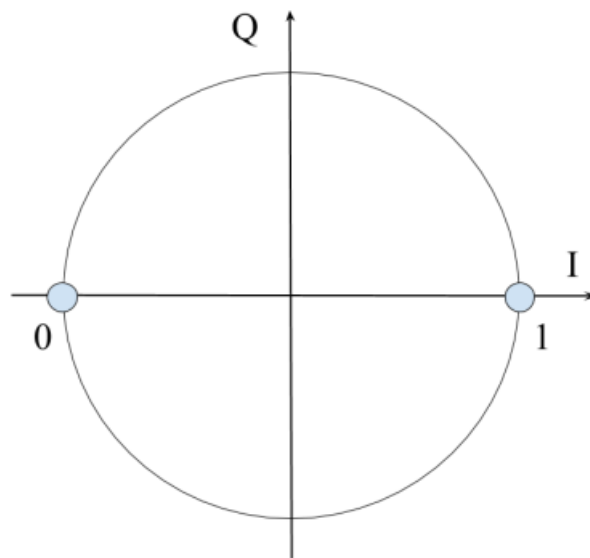
Adaptado de : (NORDICSEMICONDUCTOR, 2011)

A representação retangular no diagrama da figura 2.3 é comumente utilizada para representação de um ou mais sinais como vetores ou pontos, formando uma constelação de sinais, enfatizando assim a posição de um sinal com relação a outro. Nessa representação, a projeção do vetor do sinal no eixo I é sua componente Em-fase (I), e a projeção do mesmo vetor no eixo Q é sua componente Em-quadratura (Q). Além disso, a representação também inclui o ângulo de fase do sinal (SJODIN, 2019).

Nesse contexto, os pontos da constelação também são chamados de símbolos, que indicam a unidade de informação transmitida no momento da modulação. Por exemplo, na modulação digital BPSK (*Binary Phase-Shift Keying*), a fase da portadora é alterada em dois estados diferentes de fase,  $\theta = 0$  e  $\theta = 180$ , enquanto a amplitude é mantida cons-

tante (SJODIN, 2019). Esses dois estados consolidam que cada símbolo irá ser representado pelo seu respectivo bit, contendo então apenas dois pontos em sua constelação e transmitindo então 1 bit por símbolo. A relação entre o número de símbolos e bits é  $M = 2^k$ . Para BPSK,  $k$  só pode ser 0 ou 1 haja vista que é um esquemático de modulação binária e logo  $k = 1$  e  $M = 2$  símbolos (GUIMARAES, 2006). Na figura 2.4, o bit 1 é representado pelo símbolo +1 e o bit 0 é representado pelo símbolo -1 .

Figura 2.4 – Constelação BPSK.



Fonte : (SJODIN, 2019)

A representação é importante pois é relacionável com o esquemático de alguns moduladores e demoduladores digitais de quadratura que representam duas componentes independentes do sinal de informação (MIT, 2012b). Na modulação digital, a transmissão do sinal em banda-base consiste em um sinal de linha oriundo do codificador de linha do sistema de comunicação digital (XIONG, 2006). Os bits são enviados conforme a necessidade no formato de um sinal unipolar que retorna ao zero (RZ) ou não retorna ao zero (NRZ), para que sejam combinados com uma portadora (WONG, 2017).

Figura 2.5 – Sinal unipolar.

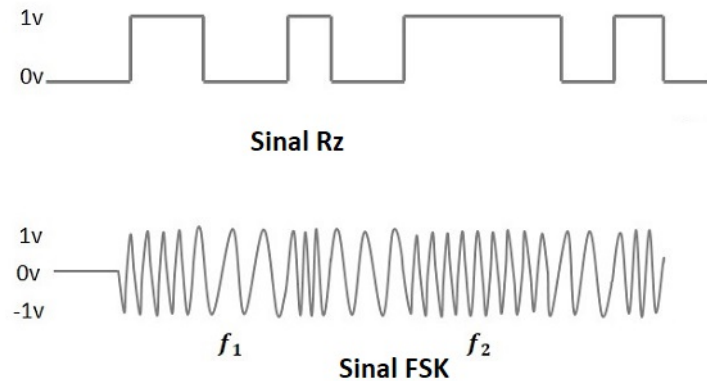


Fonte: (TUTORIALSPPOINT, 2019a)

Quando o sinal em banda-base é combinado com a portadora, a mesma modifica

seus parâmetros de frequência, fase e amplitude de acordo com o sinal em linha conforme a figura 2.6.

Figura 2.6 – Modulação *Frequency-Shift Keying*



Fonte: (TUTORIALSPPOINT, 2019b)

As principais formas de modulação digital passa-banda são divididas em coerentes e não coerentes. Na técnica de modulação coerente, o receptor possui o conhecimento da fase da portadora no processo de demodulação, enquanto que na técnica de modulação não coerente, o receptor não utiliza esse conhecimento (ASVIAL, 2019).

Ademais, é preciso definir conceitos fundamentais como a eficiência espectral, o *baudrate* e índice de modulação. A eficiência espectral deriva da representação gráfica da distribuição de energia do sinal em função da frequência. Isso define uma razão entre fluxo de dados e largura de banda, mostrando, assim, a quantidade de dados que podem ser transmitidas em uma largura de banda específica e, por isso, a eficiência espectral é frequentemente utilizada para comparar diferentes modulações para uma mesma largura de banda (SIMONS, 2010).

Já o *baudrate* ou *symbolrate*, expressa a taxa que a informação é transmitida no canal em bits por segundo ou baud. Paralelamente a isso está definido o conceito de tempo de símbolo, que é o inverso do *baudrate* (XIONG, 2006).

O índice de modulação,  $m$ , para FSK expressa uma relação entre os binários transmitidos e a portadora (XIONG, 2006). Para essa modulação binária é definida como:

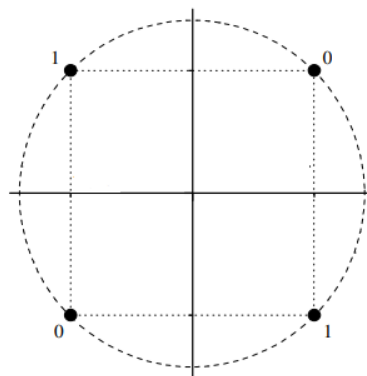
$$\frac{f_2 - f_1}{R} \quad (2.1)$$

Onde  $f_2$  é a frequência que representa o bit 1,  $f_1$  representa o bit 0 e  $R$  é a quantidade máxima de dados transmitidos no canal de comunicação.

Atualmente, as principais formas de modulação digital passa-banda são a FSK

(*Frequency-Shift Keying*), onde a frequência da portadora é alterada, a PSK (*Phase Shift Keying*) e a QAM (*Quadrature Amplitude Modulation*), onde a fase e a amplitude da portadora são alteradas simultaneamente. O presente trabalho se limita ao escopo da modulação binária que transmite 1 bit por símbolo e suas variações, como a MSK (*Minimum Shift Keying*) que é semelhante à FSK, porém com um índice de modulação  $m = 0.5$  e engloba a classe de modulações de fase contínua, amplamente utilizada por sistemas de satélite de comunicação da NASA e no sistema global para comunicações móveis, GSM (SIMONS, 2010).

Figura 2.7 – Constelação MSK.



Fonte: Adaptado de (GUIMARAES, 2006)

A transição entre símbolos é suave, pois, não há transições abruptas de fase e o sinal varia continuamente de 0 à 180° (GUIMARAES, 2006). Conforme a figura 2.7, o bit 0 é representado pelos símbolos  $1+1j$  e  $-1-1j$  no passo que o bit 1 é representado por  $1-1j$  e  $-1+1j$ . Como a amplitude é constante e a fase varia relacionada a frequência o mapa IQ resulta no círculo da figura 2.10 sobre o qual situam-se os valores IQ recebidos, posicionados de acordo com a fase instantânea do sinal (GUIMARAES, 2006).

Para o esquemático de modulação de fase contínua, a seleção das frequências  $f_2$  e  $f_1$  garante a transição de fase suave quando ocorre a transição entre as frequências de acordo com o sinal de linha, isso elimina possíveis erros na implementação do demodulador.

Segundo (LATHI, 2010) um sinal que engloba esse escopo é definido como:

$$s(t) = A \cos(2\pi f_c t + \phi(t, a)) \quad (2.2)$$

Onde  $A$  é a amplitude,  $f_c$  é a frequência da portadora e  $\phi(t, a)$  é a variação de fase contínua no tempo em função da sequência de símbolos  $a$ . A mudança de fase é definida de acordo com índice de modulação e a sequência de símbolos.

A demodulação é utilizada no receptor para adquirir a informação contida no sinal em banda-base, extraindo esse da portadora. Como a sequência de símbolos está associ-

ada ao argumento de fase pela equação 2.2, a demodulação pode ser realizada de acordo com a informação de fase para esse caso (LATHI, 2010). Usualmente, para modulações que não necessitam de equalizador para a redução da interferência intersimbólica (ISI), causada por cenários de ruído branco aditivo gaussiano e multipercorso, o processo é resumido nas etapas de *down-conversion*, filtragem, *down-sampling* e detecção através de um circuito de decisão (XIONG, 2006). As etapas descritas acima serão posteriormente descritas em outra seção.

### 2.3.1 Modulação GMSK

O Comitê de consulta de sistemas de dados espaciais (CCSDS), após uma extensa pesquisa ordenou o uso exclusivo da modulação binária GMSK (Gaussian Minimum Shift Keying), para todos os tipos de missões espaciais com alta taxa de *bitrate* devido a requisitos necessários de baixa potência e diminuição de largura de banda durante a transmissão (SIMONS, 2010). Esse esquemático de modulação é considerado uma versão alternativa da modulação MSK, com a adição de um filtro gaussiano que aumenta a eficiência espectral da modulação (GOVINDAIAH, 2012).

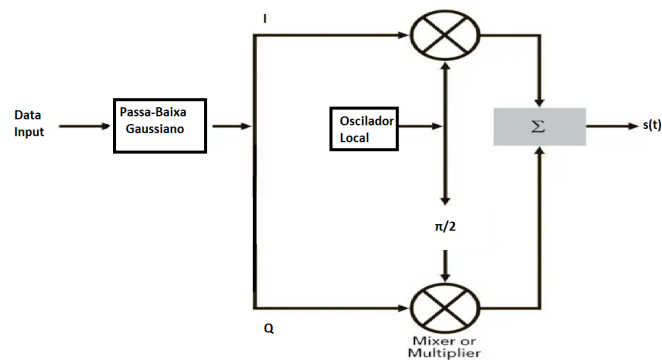
Para essa modulação de fase contínua, a eficiência espectral é obtida através da filtragem dos pulsos retangulares característicos da MSK por meio de um filtro FIR que contém uma resposta ao impulso gaussiana. A utilização do filtro define o produto BT, produto entre a largura de banda do filtro em banda-base e o tempo de símbolo, que define a eficiência espectral da modulação, valores menores de BT aumentam a eficiência espectral da modulação (MUROTA, 1981). Ademais, esse tipo de modulação provê uma ótima solução referente ao uso eficiente da largura de banda, sendo utilizado no sistema global de comunicações (GSM), e em comunicações espaciais desde órbita baixa à espaço-profundo (SIMONS, 2010).

O GMSK é um esquema de modulação binária que provê a amostragem de um sinal sem descontinuidades de fase no domínio do tempo. Segundo (VEA, 2012), na prática uma das maneiras de performar a modulação consiste nas seguintes etapas expostas conforme a figura 2.8:

- 1 Um sinal de linha NRZ, igual ao da figura 2.5, representando a sequência binária transmitida em banda-base é inserido na entrada de um filtro FIR gaussiano, antes da entrada do modulador.
- 2 Utilização do modulador em quadratura para adequar o sinal em banda-base para banda passante utilizando a portadora.



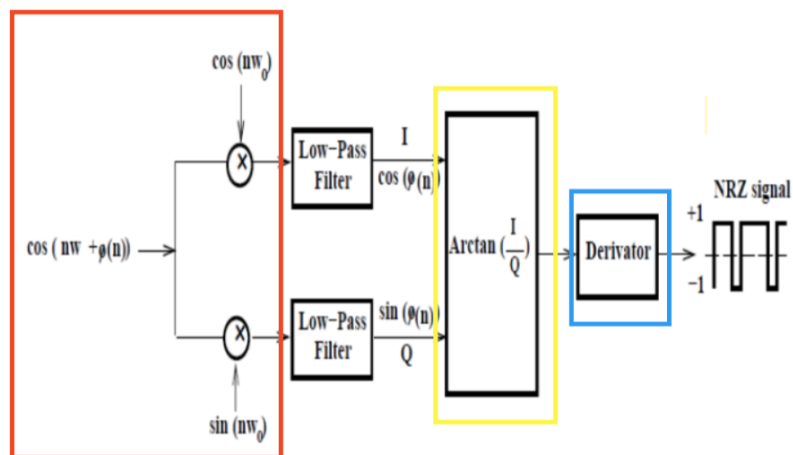
Figura 2.8 – Modulador IQ GMSK.



Adaptado de: (VEA, 2012)

### 2.3.2 Demodulador de quadratura GMSK

Figura 2.9 – Demodulador de quadratura GMSK implementado em uma placa FPGA.



Fonte : (GOVINDAIAH, 2012)

Conforme a figura 2.9 o sinal modulado em passa-banda é submetido a uma série de processos até a reconstrução dos bits. De início o *down-converter* indicado em vermelho na figura 2.9 efetua o processo de mixagem alterando a frequência do sinal para banda-base, para isso, o *mixer* de quadratura combina o sinal modulado com dois sinais

locais através da multiplicação, produzindo dois sinais de saída nos ramos I e Q deslocados 90 graus de fase entre si. Os sinais locais normalmente são senoides produzidas por um oscilador local (MIT, 2012b). O produto entre os sinais resulta em uma faixa de frequência adicional que é filtrada por dois filtros FIR passa-baixa gaussianos, para obter uma alta SNR o filtro do receptor deve ser casado com o do transmissor (SMITH, 2011).

Após a etapa de filtragem é feito o cálculo do arco-tangente das componentes I e Q, onde I representa a magnitude do sinal e Q a fase com relação a portadora indicado em amarelo na figura 2.9. Ainda que comumente esse tipo de metodologia baseada na fase é utilizada em sistemas de comunicação que utilizam modulação de fase, existem aplicações que utilizam a informação de fase para recuperar a informação original transmitida, já que mesmo para uma modulação de fase contínua como o GMSK a reconstrução da fase pode estimar a frequência do sinal conforme em (SHRESTHA, 2016).

Assim sendo, aplicações e algoritmos são desenvolvidos na arquitetura do demodulador para conceber a reconstrução de fase. O cálculo de fase é normalmente realizado através de um algoritmo, como o CORDIC, implementado em hardware digital (GOVINDAIAH, 2012). Isso é visto em (SHIMA, 1995), onde se utiliza um discriminador polar para medir a diferença de fase entre duas amostras consecutivas do sinal e posteriormente essa diferença torna-se a frequência instantânea do sinal. O ângulo de fase do n-ésimo símbolo em radianos é dado por:

$$\phi[n] = \text{atan}(Q[n]/I[n]) \quad (2.3)$$

O discriminador retorna a diferença de fase entre as duas amostras e a frequência angular instantânea é calculada através de :

$$\omega[n] = \frac{(\phi[n] - \phi[n - 1])}{\Delta t} \quad (2.4)$$

Onde  $\Delta t$  é o inverso da frequência de amostragem do conversor A/D. Através da relação acima a frequência instantânea em Hz pode ser calculada através da relação:

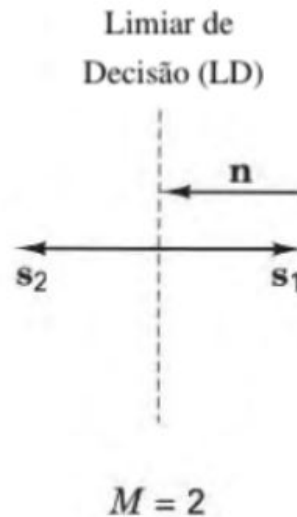
$$f[n] = \omega[n]/2\pi \quad (2.5)$$

Por último, um diferenciador ou derivador indicado em azul na figura 2.9 é utilizado para recuperação do sinal. No contexto de um demodulador de quadratura, o diferenciador produz um pulso quadrado representando as amostras dos símbolos 1 e 0, respectivamente (GOVINDAIAH, 2012). Visto que, para GMSK, um bit é representado pelo seu respectivo símbolo. O sinal de saída do diferenciador é utilizado pelo sincronizador de bits para a estimação do *stream* de bits contendo "0" e "1" (SHRESTHA, 2016).

No contexto acima, um algoritmo de decisão é utilizado no *de-mapper*, referente ao bloco indicado em azul na figura 2.9, esse opera na região de decisão do respectivo diagrama IQ, visto na figura 2.10. O vetor n representa a distância entre uma região a

outra, separados pelo símbolos  $s_1$  e  $s_2$  (SKLAR, 2001). Caso haja mais símbolos como a modulação 16-QAM, a probabilidade de realização de uma decisão incorreta aumenta, pois há 16 limiares pequenos de decisão para cada símbolo recebido enquanto que para  $M = 2$ , há apenas dois limiares.

Figura 2.10 – Limiar de decisão da modulação 2 - PSK com  $M = 2$  símbolos.



Fonte: (SKLAR, 2001)

Ao utilizar um algoritmo de decisão abrupto, do inglês *hard decision*, esse considera apenas as distâncias euclidianas entre o sinal e os possíveis sinais do diagrama IQ. Caso seja recebido um símbolo  $s$ , neste método ocorre o cálculo da distância euclidiana entre  $s$  e todos os símbolos da constelação da referência, sendo então a decisão baseada no símbolo que possui a menor distância euclidiana comparada ao sinal recebido (PROAKIS, 2008). Isso provê incerteza quando comparada a decisão suave.

No caso de decisão suave, a decisão é tomada baseada em uma sequência pré-estabelecida de símbolos de acordo com o codificador, executando a inferência de quais símbolos foram transmitidos com base na probabilidade conjunta dos símbolos dessa sequência *log-likelihood ratio* (LLR)(SKLAR, 2001).

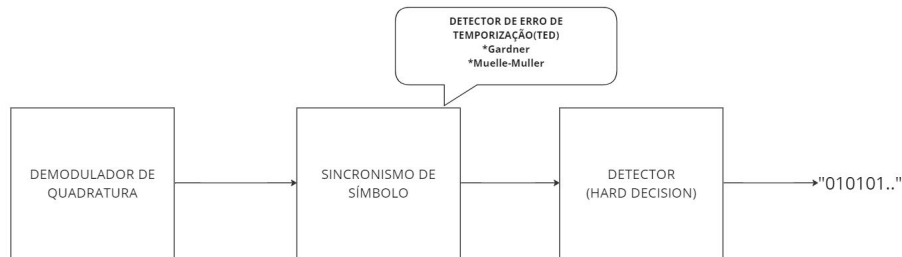
Posteriormente, um decodificador pode ser adotado para recuperar a informação codificada pelo transmissor.

#### 2.4 SINCRONISMO DE SÍMBOLO - RECUPERAÇÃO DE *CLOCK*

No presente trabalho, o sincronismo de símbolo é empregado nos demoduladores implementados no GNU Radio, após o cálculo da frequência instantânea performada pelo

demodulador de quadratura.

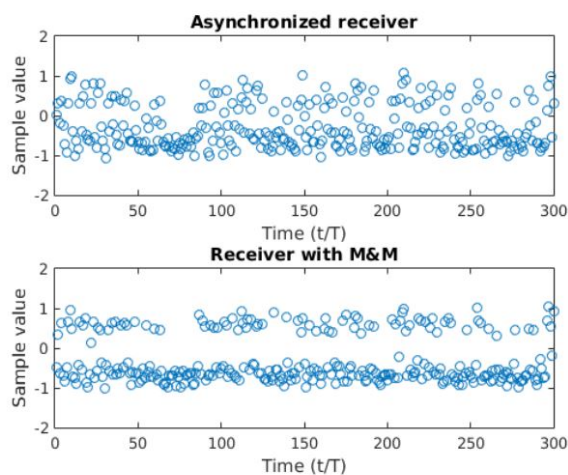
Figura 2.11 – Interação do sincronismo de símbolo no presente trabalho.



Fonte: O Autor

A recuperação de *clock* é necessária para identificar e separar corretamente os símbolos individuais, minimizando a dispersão e facilitando a estimação dos símbolos (SKLAR, 2001). Sobre isso, O impacto do detector de erro de temporização pode ser visualizado na figura 2.12, que implementa um algoritmo de Mueller-Muller no processamento de sinal referente ao nanossatélite LilacSat-2 (CHATZIGEORGIOU, 2020). O receptor que não implementa o sincronismo falha ao amostrar o sinal no momento exato, no passo que ao performar o Mueller-Muller os símbolos enquadram-se nas regiões de interesse, correspondente aos picos do sinal binário "+1 e -1"(CHATZIGEORGIOU, 2020).

Figura 2.12 – Impacto do sincronismo de símbolo.

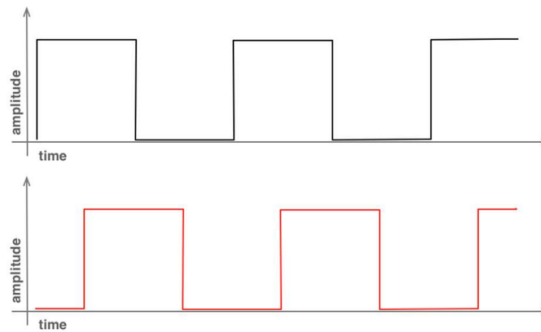


Fonte: (CHATZIGEORGIOU, 2020)

Em um enlace real via satélite, os *clocks* dos hardwares, como o transmissor (Tx) do satélite e o receptor (Rx) da estação terrena, nunca estarão em sincronismo nem operando na mesma frequência. Isso ocorre devido à variação da distância entre os mecanismos e às propriedades dos materiais utilizados nos hardwares (MALTE, 2016). Por isso, há um

desalinhamento no tempo entre os domínios de tempo amostrados entre o receptor e o transmissor devido a diferença de *clock*, já que não há sincronização entre as bordas de subida e descida de *clock* dos componentes.

Figura 2.13 – Diferença de *clock* entre Tx e Rx.



Fonte:(ALILI, 2021)

A recuperação de símbolo pode ser expressa matematicamente através da determinação do *offset* de tempo que maximiza a energia do sinal recebido, ou a seleção do instante correto de amostrar o sinal recebido, em uma linguagem de processamento de sinais (JOHNSON, 2011). Sendo que esse *offset* determina quando iniciar a amostragem.

Para isso, os algoritmos de recuperação de símbolo como o de Gardner e Mueller-Muller são empregados e implementados em software. Estes determinam o instante exato de amostragem do sinal recebido. Esta amostragem visa alinhar de certa forma o *clock* do Tx com o Rx, ajustando de maneira correta o tempo de início e fim de cada símbolo transmitido (PROAKIS, 2008).

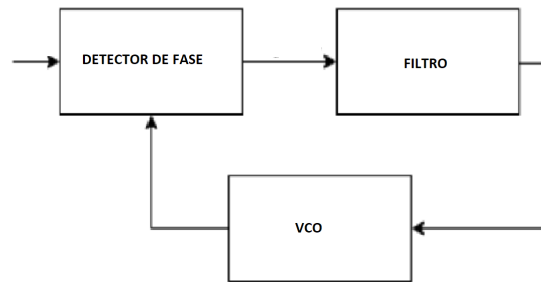
Isso é necessário pois um receptor não sincronizado aumenta a probabilidade de erro, diminuindo o desempenho do mesmo, e em virtude disso os símbolos não serão decodificados corretamente (JOHNSON, 2011). Isso pode ser visualizado através de um diagrama de olho, cuja construção é baseada na amostragem e sincronismo correto do sinal (PROAKIS, 2008).

Em meados do início do processamento de sinais digitais a sincronização de símbolo era realizada por meio da geração de um sinal cujo *clock* era alinhado com o *clock* do Tx, implementado por meio de um circuito contendo um VCC, *clock* controlado por voltagem, implementado em um *loop* de fase fechada (PLL) (Qasim Chaudhari, 2020). Atualmente, prefere-se a utilização de um PLL totalmente digital para recuperação, como o *Early-late*, *Gardner* e *Mueller and Muller* (MALTE, 2016).

### 2.4.1 PLL

Um *loop* de fase fechada é um dispositivo que constantemente ajusta a fase de um oscilador conforme a fase de um sinal de entrada, por meio de realimentação (MALTE, 2016).

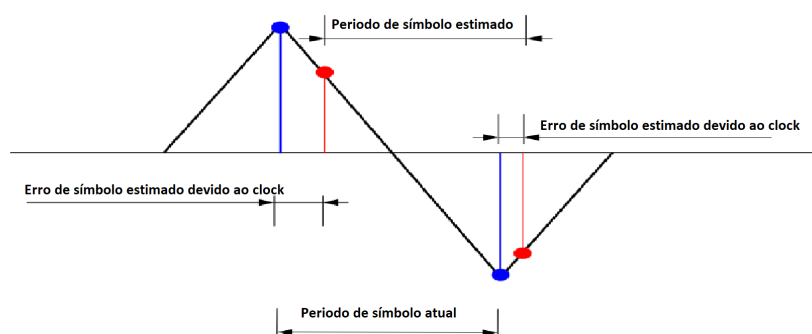
Figura 2.14 – Loop de fase fechada.



Adaptado de : (Qasim Chaudhari, 2020)

O detector de fase calcula a função de *offset* de fase entre o sinal de entrada e o sinal produzido pelo VCO, do inglês *voltage controlled oscillator*. A saída do detector de fase é filtrada pelo filtro, que estabelece uma performance dinâmica, suprimindo as componentes de alta frequência e ruído (CHATZIGEORGIOU, 2020). Para que o sincronismo possa ser obtido e a diferença de *clock* possa ser suprimida, é necessário que no PLL tenha um algoritmo de sincronização e um filtro interpolador, do inglês *timing error detector*, ou TED (WALLS, 2020).

Figura 2.15 – *Timing error detector (TED)*



Fonte: Adaptado de (WALLS, 2020)

O TED gera um sinal ou função de erro, proporcional a diferença no tempo entre o tempo ótimo de amostragem do símbolo e o tempo atual de amostragem de símbolo, simbolizado em vermelho na figura 2.15 (WALLS, 2020). Esse sinal é posteriormente filtrado até a saída do filtro ser constante. Paralelamente a isso o interpolador realiza uma re-amostragem, quando a saída é constante o erro de sincronismo é zero e há sincronismo entre Rx e Tx (SKLAR, 2001).

Alguns algoritmos comuns utilizados no TED incluem o algoritmo de Gardner, o algoritmo de Mueller-Muller, Early-Late e o D'Andrea GMSK (Qasim Chaudhari, 2020). Esses algoritmos podem variar em complexidade e desempenho, dependendo das características do canal de comunicação e do tipo de modulação utilizado.

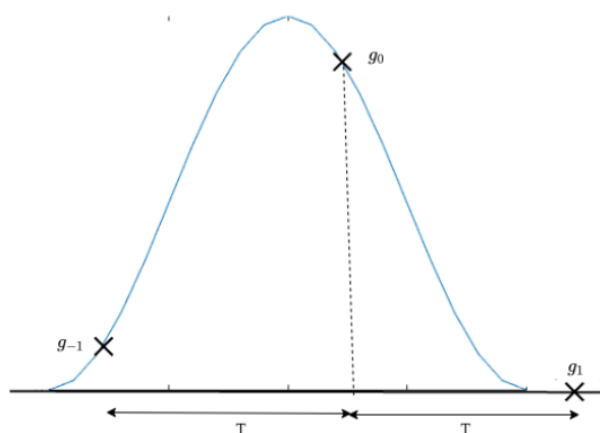
O presente trabalho se limita aos algoritmos de Mueller-Muller e Gardner.

#### 2.4.1.1 Mueller-Muller TED

Originalmente desenvolvido em 1976, o algoritmo de Mueller-Muller consegue operar com 1 amostra por símbolo, ao contrário dos demais incluindo o de Gardner (PORTELA, 2012). Esse algoritmo estima a defasagem de *clock* do receptor a partir de amostragens ótimas, utilizando as amostras de saída do filtro do *Loop* (Qasim Chaudhari, 2020).

Os instantes ótimos são aproximadamente localizados nos picos do sinal de entrada. Para isso, a função de erro  $f(\tau)$  computa a simetria da resposta ao impulso do símbolo, essa é zero somente quando a simetria coincide com o máximo da resposta ao impulso (CHATZIGEORGIOU, 2020).

Figura 2.16 – Resposta ao impulso do símbolo coma atraso  $f(\tau)0$



Fonte: (CHATZIGEORGIOU, 2020)

O interpolador produz uma amostra para cada símbolo recebido, uma no instante

anterior  $g - 1$ , e outra no instante exato de recebimento do símbolo (CHATZIGEORGIOU, 2020). Caso  $g - 1$  não coincida em simetria com  $g1$  não há sincronismo e portanto o tempo de amostragem é incorreto, o que pode ser visto na figura 2.16.

No software GNU Radio, o algoritmo de Mueller-Muller é implementado no bloco que realiza o sincronismo de símbolos, *Symbol Sync* (GNU, 2020).

O bloco inicialmente estima o número de amostras por símbolo de entrada, dado o número de referência definido no bloco e verifica um desvio entre estas, este parâmetro é necessário pois indica com quantas amostras do sinal um símbolo é representado (GNU, 2020). Por exemplo, para um *bitrate* de 1200 bits por segundo, e uma frequência de amostragem padrão de 48000KHz o número resulta em 40 amostras por símbolo, de acordo com:

$$Samples/Symbol = F_s * \frac{1}{bitrate} \quad (2.6)$$

Após isso este performa a sincronização para que o sinal seja amostrado no momento exato do tempo, quando  $f(\tau) = 0$  para o Mueller-Muller, por último decima as amostras para que a saída seja de 1 amostra por símbolo (GNU, 2020).

#### 2.4.1.2 Gardner TED

O algoritmo foi proposto por Floyd Gardner e é amplamente utilizado para sincronismos de símbolo, em sistemas de comunicação digital por modulação de desvio de fase e desvio de frequência (SANTOS, 2021).

O objetivo principal do algoritmo de Gardner, assim como o Mueller-Muller, é estimar o momento ótimo para realizar a amostragem do sinal recebido, garantindo a correta demodulação dos símbolos transmitidos (CANTU, 2019). O algoritmo opera com duas amostras por símbolo, sendo amplamente utilizado em vários receptores digitais devido à sua capacidade de sincronizar os símbolos mesmo em condições de interferência e desvanecimento de canal (CANTU, 2019).

O interpolador do PLL produz duas amostras para cada símbolo recebido, uma no instante exato de recebimento  $T$  e outra  $T/2$  segundos antes. Ainda, uma terceira amostra é utilizada, produzida no instante anterior,  $T - 1$  (PORTELA, 2012). A função de erro de Gardner é descrita então através de:

$$e = (y(T - 1) - y(T)) * y(T/2) \quad (2.7)$$

Portanto, esse método de detecção de erro de temporização consiste basicamente em calcular a diferença entre a amostra atual com a anterior, multiplicando pelo ponto médio das duas amostras. Em uma FPGA, esse cálculo é feito tanto para os símbolos em fase quanto para os em quadratura que são somados no fim do algoritmo (PORTELA, 2012).



## 2.5 FILTRO FIR

Diferentemente dos filtros analógicos, que processam o sinal na forma original usando componentes físicos e, por isso, são suscetíveis a distorções e interferências, os filtros digitais processam o sinal já discretizado em amostras, oferecendo maior flexibilidade e a possibilidade de programação por meio de algoritmos para atender aos requisitos do projeto (BIGLIERI, 2016).

O principal uso do filtro digital é restaurar o sinal original, já que há distorção devido ao canal de comunicação. A sua principal forma de implementação é através da convolução entre a resposta ao impulso digital do filtro e o sinal de entrada. Esse tipo de implementação difere dos filtros digitais recursivos e é característico do filtro FIR (Finite Impulse Response) (SMITH, 2011). As características operacionais do filtro são especificadas a partir das respostas ao impulso, ao degrau e em frequência. Para uma análise adequada das respostas do filtro, é necessário compreender como a informação é representada no domínio do tempo e no domínio da frequência. A resposta ao degrau está diretamente ligada à operação do filtro no domínio do tempo, enquanto que a resposta em frequência é usada para verificar a operação do filtro com relação à filtragem. Portanto, especifica-se na banda passante do filtro a faixa de passagem e a banda de rejeição, que estabelecem um limite para a faixa de passagem.

Em seguida, verifica-se a resposta em frequência do filtro, analisando as faixas de rejeição e passagem de frequências e especificando o tipo de resposta em frequência do filtro como passa-baixa, passa-banda, passa-alta ou rejeita-banda (SMITH, 2011). O filtro de média móvel gaussiano é utilizado principalmente no contexto de filtragem passa-baixa, atenuando assim ruído de alta frequência. Nesse sentido, a resposta ao impulso do filtro é suave e se aproxima de uma curva gaussiana (HALE, 2019). Isso deve-se ao tipo de cálculo de média móvel do filtro, que utiliza uma janela móvel gaussiana para calcular a média dos valores do sinal no tempo, sem utilizar pesos uniformes para cálculo da média. O filtro do tipo FIR é implementado através de:

$$y[n] = \sum_{m=n-M}^{n+M} h[n-m]x[n] \quad (2.8)$$

Onde  $h[n-m]$  é a resposta ao impulso gaussiana amostrada em  $n-m$  amostras,  $x[n]$  o sinal de entrada e  $y[n]$  o resultado da convolução, sendo  $m$  e  $n$  o número de amostras (HALE, 2019). A resposta ao impulso discretizada pode ser expressa no domínio do tempo através de:

$$h(t) = \frac{\sqrt{\pi}}{a} e^{-\frac{\pi^2 t^2}{a^2}} \quad (2.9)$$

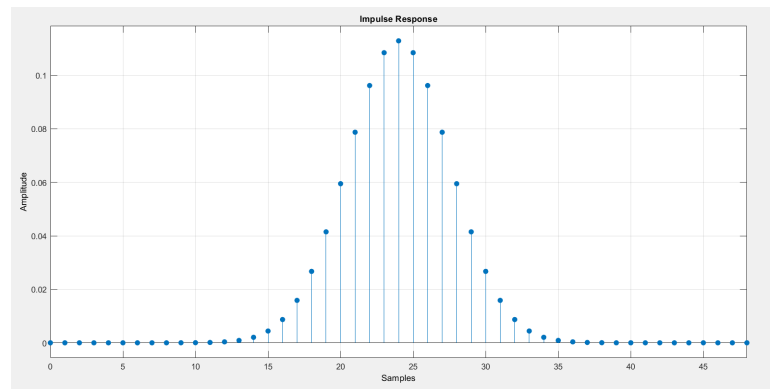
Onde o parâmetro  $a$  é muito importante para definição do parâmetro BT, *bandwidth-time*

*product*, constituído pela banda B e o tempo de símbolo adotados (MATHWORKS, 2022).

$$a = \frac{1}{BT} \sqrt{\frac{\log 2}{2}} \quad (2.10)$$

- B = Frequência de projeto.
- T = Tempo de símbolo.

Figura 2.17 – Resposta ao impulso gaussiana.



Fonte: O autor

Na presença de ruído, com o intuito de maximização da SNR, o filtro é adotado como filtro de casamento ou *matched filter*. Para isso, o filtro deve ajustar dinamicamente sua resposta de acordo com as características do sinal de entrada, para que a resposta do filtro do receptor seja a mais próxima possível da resposta do filtro do transmissor. Isso é realizado por meio dos ajustes dos coeficientes do filtro (BIGLIERI, 2016). Assim, a resposta do sinal desejado é maximizada e a relação de sinal-ruído, SNR, é aumentada.

## 2.6 LINK BUDGET

O *link budget* é utilizado para verificar o nível de sinal obtido pela antena receptora da estação terrena. Esse processo é feito no sistema de radio frequência por meio da potência total do transmissor e a soma de perdas e ganhos durante o processo de transmissão na finalidade de determinação do nível de sinal que chega no receptor, que deve ser maior que um nível mínimo operacional (CAMPBELL, 2016).

Esse processo é descrito por meio da equação:

$$P_{rx} = P_{tx} - L_{tx} + G_{tx} - L_{path} + G_{rx} - L_{rx} \quad (2.11)$$

Onde:

- $P_{tx}$  = Potência do sinal transmitido em  $dBm$ .
- $L_{tx}$  = Perda total do sistema no transmissor em  $dB$ .
- $G_{tx}$  = Ganho da antena em  $dB$  no transmissor.
- $L_{path}$  = Perdas de propagação do sinal entre as antenas transmissora e receptora.
- $G_{rx}$  = Ganho da antena receptora em  $dB$
- $L_{rx}$  = Perda total do sistema no receptor em  $dB$ .
- $P_{rx}$  = Potência do sinal recebido em  $dBm$

Em um primeiro momento, para estabelecer o orçamento de link é necessário calcular a potência do sinal transmitido ou potência de saída, dada pela relação:

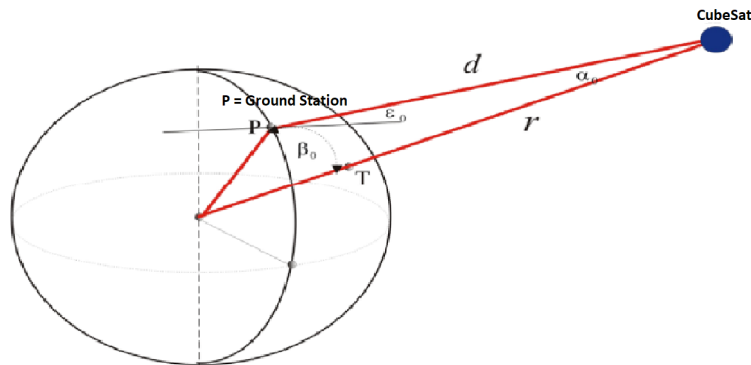
$$P_{dBm} = 10\log(P_{watts}) + 30 = P_{tx} \quad (2.12)$$

As perdas do sistema de transmissão,  $L_{tx}$  e de recepção  $L_{rx}$  são determinadas a partir da contabilidade de perdas da linha de transmissão somadas a alguma perda de impedância da antena.

Além disso, é verificado o ganho as antenas de transmissão e recepção,  $G_{rx}$ ,  $G_{tx}$ , além das perdas devido ao caminho de transmissão  $L_{path}$ , expressa por meio da soma das perdas de espaço livre e da interação da onda eletromagnética com o ambiente. Por último, a sensibilidade do receptor pode ser determinada para verificar se esse pode ser utilizado, ou seja é feito a verificação de margem de desvanecimento do projeto ou fator de redução do sinal em  $dBm$  tal que o sinal ainda é aproveitável no receptor (KYMETA, 2019).

Um primeiro passo para determinação do *link budget* é contemplar a largura de banda,  $B$ , e a relação sinal ruído, SNR que é diretamente relacionado com o esquemático de modulação (J.MILLER, 2012). Ainda, a largura de banda disponível é de 20KHz (IARU, 2017). Para estabelecer as perdas devido ao ambiente de propagação do sinal e de trajetória é necessário estabelecer uma geometria de interação entre transmissor e receptor. O transmissor em questão deverá estar inserido em órbita baixa terrestre, LEO, cuja altitude a partir do nível do mar é da ordem de 2000 quilômetros (SHURDI, 2011).

Figura 2.18 – Geometria de interação entre satélite e terra.



Adaptado de (SHURDI, 2011)

Nesse sentido, pode-se contemplar a distância de *slant range*, distância entre satélite e estação terrena, bem como a determinação de ângulos que orientam o satélite e a estação terrena os quais também são de extrema importância para a determinação do link (SHURDI, 2011). Por meio da aplicação da lei dos cossenos para o triângulo formado pelos ângulos de nadir,  $\alpha_0$ , central,  $\beta$ , e elevação  $\epsilon$ , essa é determinada :

$$d = Re \sqrt{\left[\frac{H + Re}{Re}\right]^2 - \cos^2 \epsilon_0 - \sin \epsilon_0} \quad (2.13)$$

A partir dos dados referentes ao raio terrestre,  $Re$ , a altitude do *CubeSat* com relação a estação  $H$ , correspondente a LEO, e o ângulo de elevação da estação,  $\epsilon_0$  pode-se calcular a distância entre satélite e estação. Ademais, se o ângulo de inclinação for incrementado até  $90^\circ$  a distância entre o satélite e a estação irá diminuir.

Somado ao apontamento é importante discorrer sobre a atenuação do sinal durante a transmissão, causada através da propagação e fatores locais. Por conseguinte, a perda de espaço livre, perdas de polarização, apontamento, efeitos ionosféricos e troposféricos constituem as perdas por propagação (CAPELA, 2012). Já por outro lado o ruído no receptor, cabeamento são definidos como perdas locais (CAPELA, 2012).

### 2.6.1 Perda de caminho livre - LFS

A principal atenuação em comunicações via satélite é a perda de caminho livre, consistindo da perda de sinal em dB durante a propagação do mesmo em direção ao receptor formulada por dois principais fatores, a diminuição na densidade de potência do sinal,  $S$ ,

e a eficiência da antena receptora medidas em termos da potência recebida (KYMETA, 2019). Combinando as equações referentes aos efeitos acima:

$$S(W/m^2) = \frac{P_t}{4\pi d^2} \quad (2.14)$$

$$P_r(W/m^2) = \frac{S\lambda^2}{4\pi} \quad (2.15)$$

Onde:

- S = Densidade de potência do sinal em  $W/m^2$ .
- $P_t$  = Potência transmitida em W.
- d = Distância entre Tx e Rx.
- $P_r$  = Potência recebida em  $W/m^2$ .
- $\lambda$  = Comprimento de onda do sinal em m.

Expressando o resultado em dB:

$$LFS = \frac{P_t}{P_r} = 20\log(d) + 20\log(f) + 92.45 \quad (2.16)$$

### 2.6.2 Atenuação devido a chuva

A atenuação devido a chuva é dependente da frequência do sinal onde o efeito referente a isso pode ser desconsiderado para frequências inferiores a 1GHz (BABIC, 2016). Para estimar a atenuação devido a chuva é necessário calcular a atenuação específica baseado no volume de chuva local no ano em mm/h.

### 2.6.3 Figura de ruído do receptor

A figura de ruído determina o desempenho de ruído do sistema de radiofrequência de modo que quanto mais baixa maior o desempenho do sistema (MENDEZ, 2022). O ruído térmico é geralmente modelado como AWGN e é dado através de:

$$N = kT_e B \quad (2.17)$$

Onde:

- $k$  = Constante de Boltzman.
- $T_e$  = Temperatura de ruído do receptor em Kelvin.
- $B$  = Largura de banda em KHz.

Normalmente a figura é fornecida ou encontrada em *datasheets* do mecanismo utilizado como receptor. Ainda, a figura de ruído pode ser definida como a relação de sinal-ruído de entrada e saída de um sistema de radiofrequência já que uma menor figura resulta em uma relação de SNR alta de saída.

$$NF = \frac{(SNR)_{in}}{(SNR)_{out}} \quad (2.18)$$

Visto que a figura de ruído tem relação direta com o ruído térmico do sistema de radiofrequência.

$$T_e = 290K(NF - 1) \quad (2.19)$$

#### 2.6.4 Cintilação Ionosférica

A modificação das ondas de rádio causadas por partículas atmosféricas também degrada o sinal. Para uma frequência inferior a 3GHz há a ocorrência de degradação de fase, amplitude, polarização principalmente em regiões polares e equatoriais (BABIC, 2016). Isto é considerado um evento não determinístico que ocorre na ionosfera devido a partículas carregadas, ou seja, depende do conteúdo total de elétrons que varia durante a altitude da camada no passo que para uma maior altitude ocorre um aumento não linear devido a atividade solar.

Devido a isso, os principais efeitos resultantes são a rotação de polarização, perdas por cintilação, absorção, variação na direção de chegada da onda, atraso de propagação, dispersão e alteração de frequência. Por sua vez, as perdas de cintilação devido ao índice de refração atmosférico inferem uma mudança grande durante a propagação por meio de variações de amplitude, fase e polarização da onda, resultando em perdas superiores à 5dB (MENDEZ, 2022).

#### 2.6.5 Apontamento

O apontamento está ligado com o sistema de atitude do veículo espacial, esse deve estar sempre apontado em direção a estação terrena para estabelecer a comunicação.

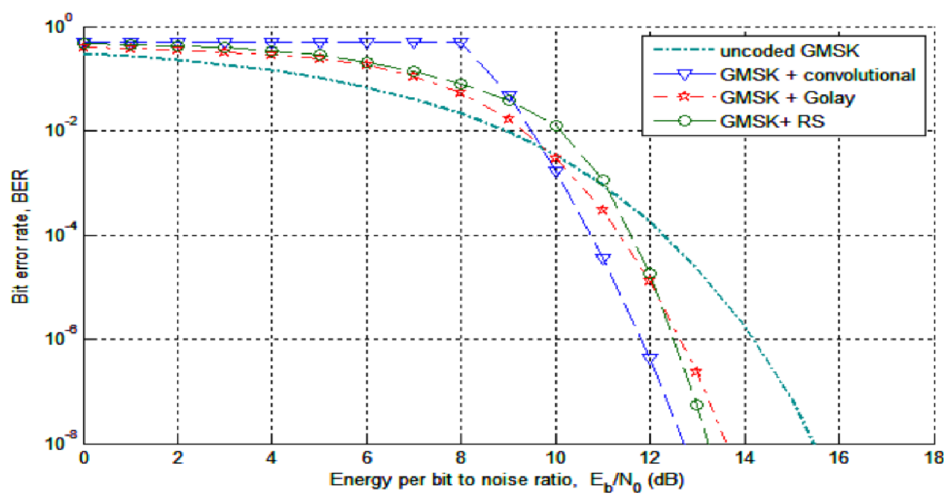
Na posição mais distante em relação a estação a perda devido ao apontamento é maior, chegando a 6dB para as bandas VHF e UHF (BABIC, 2016).

### 2.6.6 Relação Sinal-Ruído

A relação Sinal-Ruído, que expressa a relação de ganho do sinal sobre o ruído, é utilizada para verificar a taxa de variação de bits transmitidos, *bitrate*, através da relação  $E_b/N_0$  (BABIC, 2016). A medição do SNR para um sinal digital é efetuada em termos dessa relação que exprime a potência do sinal recebido, variando de acordo com o tipo de modulação digital utilizada, taxa de energia por bit e a densidade espectral de ruído. Normalmente nesse contexto há a possibilidade de verificar a potência do link para um *BER* onde frequentemente a taxa de desejada é de  $10^{-6}$ , expressando que para 1000000 bits transmitidos 1 bit está trocado ou com erro devido a degradação do ruído do canal de transmissão, por exemplo (PEARCE, 2000). Ainda mais, o *BER* pode indicar o quão frequente um pacote de bits deve ser retransmitido por causa de um erro.

Segundo (BOUALLEGUE, 2015), um código corretor de erro, que permite corrigir o erro derivado da degradação do canal, permite uma minimização da potência de sinal transmitido devido ao ganho do codificador. Nesse sentido para um canal com ruído aditivo branco gaussiano ocorre a plotagem na figura 2.19 para a modulação GMSK visando expor a performance do código corretor de erro em termos do BER vs SNR, para os códigos de *Golay*, *Reed-Solomon*, *Viterbi* e para uma transmissão não codificada.

Figura 2.19 – BER VS SNR para a modulação GMSK codificada.



Fonte : (BOUALLEGUE, 2015)

### 2.6.7 Polarização

O efeito de rotação de Faraday, onde um campo eletromagnético paralelo a propagação da onda exerce influência de forma que uma onda linearmente polarizada é rotacionada na ionosfera, variando a polarização da antena transmissora. No canal de comunicação há perdas no *link budget* tal que a despolarização do sinal é incluída, sendo esta equivalente a 3dB (BABIC, 2016). Ademais, o conteúdo total de elétrons presente no meio de propagação acaba exercendo efeito de maneira proporcional ao grau de rotação da onda. De acordo com esse contexto, uma maneira de mitigar o efeito é utilizar antenas de polarização circular na estação terrena (CAPELA, 2012).

### 2.6.8 Efeito Doppler

Diretamente ligado com o movimento relativo entre transmissor e receptor, durante o processo de recepção o efeito doppler pode acarretar um desvio de frequência de dezenas de KHz para a faixa UHF e devido a isso um ajuste na frequência de operação durante a passagem é necessário como forma de compensação (Hashiro, 2015). A alta velocidade relativa faz com que ocorra uma mudança rápida de frequência com o tempo. Ademais, são necessárias estratégias de compensação implementadas na estação terrena devido ao alto nível de desvio. A taxa de variação desse desvio de frequência pode também comprometer o processo de demodulação (CAO; ZHAI, 2021).

## 2.7 CUBESAT

Em 1999 um projeto feito pela universidade politécnica da califórnia, *Cal Poly*, em conjunto com a universidade de *Stanford*, propôs uma maneira acessível, de acesso ao espaço principalmente para instituições universitárias, por meio de satélites miniaturizados ou popularmente *CubeSats*. A partir disso várias universidades e instituições até mesmo governamentais aderiram ao conceito para formar seu próprio programa e testar novas tecnologias, haja vista que o projeto em geral é de baixo custo e é até mesmo barato para ser inserido como *payload* de veículos espaciais devido a sua proporção espacial (NASA, 2017).

Nesse contexto, para que o objeto seja considerado um *CubeSat*, o mesmo deve atender a critérios de dimensões, peso e formato normalmente referenciados como U ou *unit*, de forma que atualmente há configurações diversas de 1U à 12U dimensionadas especialmente para o *dispenser* que insere o mecanismo em órbita.



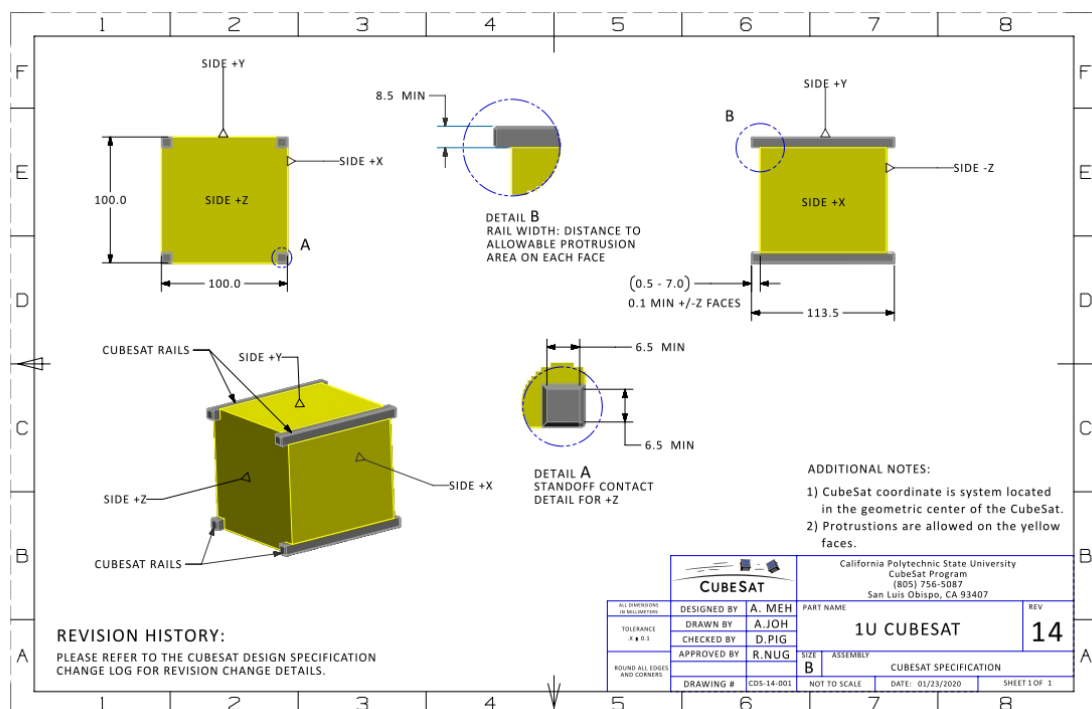
Tabela 2.1 – Dimensões de configurações comuns.

| Configuração | x(m) | y(m) | z(m)  | Massa |
|--------------|------|------|-------|-------|
| 1U           | 0.1  | 0.1  | 0.1   | 2 kg  |
| 1.5U         | 0.1  | 0.1  | 0.17  | 3 kg  |
| 2U           | 0.1  | 0.1  | 0.227 | 4 kg  |
| 3U           | 0.1  | 0.1  | 0.34  | 6 kg  |
| 6U           | 0.2  | 0.1  | 0.34  | 12 kg |

Fonte: (POLY, 2020)

Tipicamente sendo carregado como carga secundária o objeto é carregado na parte superior da baia de *payload*, sendo constantemente submetido a um alto ambiente vibracional de forma que se o mecanismo não cumpre o pré-requisito de integridade mecânica estipulado pelo veículo lançador o mesmo não pode ser inserido em órbita, embora se sabe-se que ainda há a ocorrência de danos e alterações, especialmente nos subsistemas mais sensíveis como o de comunicação (POLY, 2020).

Figura 2.20 – Configuração 1U.



Fonte : (POLY, 2020)

Ademais, os operadores que vão desde hobbistas, universidades e instituições demandam dados durante a operação, sejam relacionados com o cumprimento da missão ou apenas uma verificação do subsistema de atitude para verificar a trajetória, por exem-

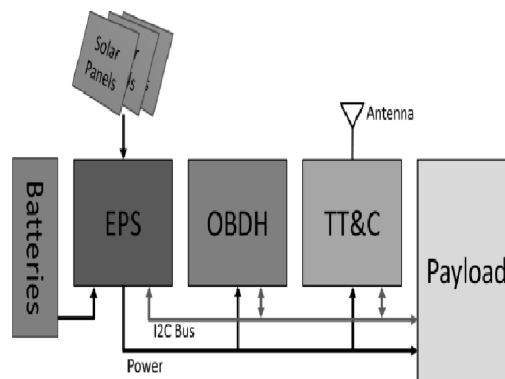
plo. Isso é estabelecido por meio do subsistema de telecomunicação do objeto através do *downlink* de dados. Outro fator muito importante de projeto é a possibilidade de utilização de itens comerciais de fácil acesso ou componentes *Commercial-Off-The-Shelf* como placas *SBG* com certificado de radiação para atuar no ambiente espacial.

Os principais subsistemas que compõem o satélite são os subsistemas de:

- Potência - EPS.
- Controle de dados - OBDH.
- Telecomunicações - TTC.
- Carga paga - *Payload*.

Estes são visualizados na figura 2.21.

Figura 2.21 – Subsistemas do FLORIPASAT-1



Fonte : (P.R.C., 2013)

## 2.8 TEORIA DE ANTENAS

Quando um condutor é exposto e excitado por meio de uma corrente elétrica variante no tempo ocorre a formação e propagação de uma onda eletromagnética que é irradiada no ambiente livre ou espaço guiado, sendo então considerado uma antena transmissora uma vez que a receptora realiza o processo oposto (MILIGAN, 2005).

Em virtude disso, quando um fio composto de um material condutor ideal é excitado ocorre a movimentação das cargas elétricas e criação de uma corrente elétrica,  $i$ , conforme a relação:

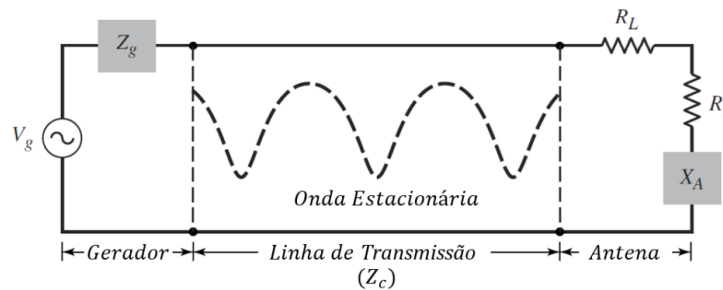
$$i = \frac{dq}{dt} \quad (2.20)$$

Onde  $q$  representa a carga em Coloumb e  $t$  o tempo em segundos.

Se o fio de comprimento  $l$  é idealmente fino e cilíndrico a corrente  $i$  é descrita pela equação 2.11. Quando o fluxo de cargas varia no tempo implica a geração de energia eletromagnética, através da variação da corrente (STUTZMAN; THIELE, 2013). Dessa forma, é possível classificar os tipos mais simples de antenas derivados da condução de corrente por um elemento linear ou seja o dipolo, comumente composto de dois fios condutores que ao ser excitado gera um campo eletromagnético emitido no plano que passa no centro do dipolo e ao qual o eixo de dipolo é perpendicular (MILIGAN, 2005).

Em um contexto de transmissão, a antena pode ser expressa pelo circuito de Thévenin, conforme a figura abaixo:

Figura 2.22 – Antena derivada do circuito de Thévenin.



Adaptado de (BALANIS, 1997)

O circuito é composto por uma fonte de tensão geradora  $V_g$ , a impedância do gerador  $Z_g$ , impedância da linha de transmissão  $Z_c$ , resistência relacionada com as perdas de condução devido a estrutura da antena  $R_l$ , resistência de radiação da antena  $R_r$  e a reatância da antena  $X_a$  caracterizando a parte imaginária da impedância associada com a radiação da antena. Em um contexto ideal a energia gerada por  $V_g$  é totalmente transmitida para  $R_r$ , ou seja a antena conectada a linha de transmissão irradiaria totalmente a energia gerada, todavia na prática ocorrem várias perdas de condução associadas à natureza do sistema (BALANIS, 1997). Nesse processo, a interação entre as ondas refletidas criadas pela interface da antena juntamente com as ondas incidentes da fonte em direção a antena geram as ondas estacionárias, que por concentrarem energia devem ser evitadas. Normalmente isso é feito por meio do casamento de impedância da antena (BALANIS, 1997).

Antes de discorrer sobre o projeto de uma antena é importante abordar os principais parâmetros a respeito do seu funcionamento que posteriormente são utilizados para verificar a viabilidade operacional. Para que os mesmos princípios possam ser aplicados para a transmissão ou recepção o meio de propagação da onda eletromagnética deverá ser idealmente linear passivo e isotrópico (MAZDA, 1998).

### 2.8.1 Diretividade, Ganho e Largura de Feixe

A diretividade,  $D$ , de uma antena é definida como a razão entre a intensidade de radiação máxima com relação à intensidade de radiação da energia irradiada por uma antena isotrópica (BALANIS, 1997).

$$D = \frac{U_{max}}{U_0} \quad (2.21)$$

A intensidade de radiação de uma fonte isotrópica é expressa como:

$$U_0 = \frac{P_{rad}}{4\pi} \quad (2.22)$$

E a diretividade  $D$  de uma antena em relação à antena isotrópica é dada por:

$$D = \frac{U}{U_0} = \frac{4\pi U}{P_{rad}} \quad (2.23)$$

Onde  $P_{rad}$  é a potência radiada em W,  $U$  a intensidade de radiação expressa em  $\frac{W}{sr}$ . Dessa forma, quanto maior a diretividade maior a capacidade de focalização de radiação em uma direção. Caso a diretividade seja unitária a mesma energia é irradiada em todas as direções, sendo então uma antena isotrópica.

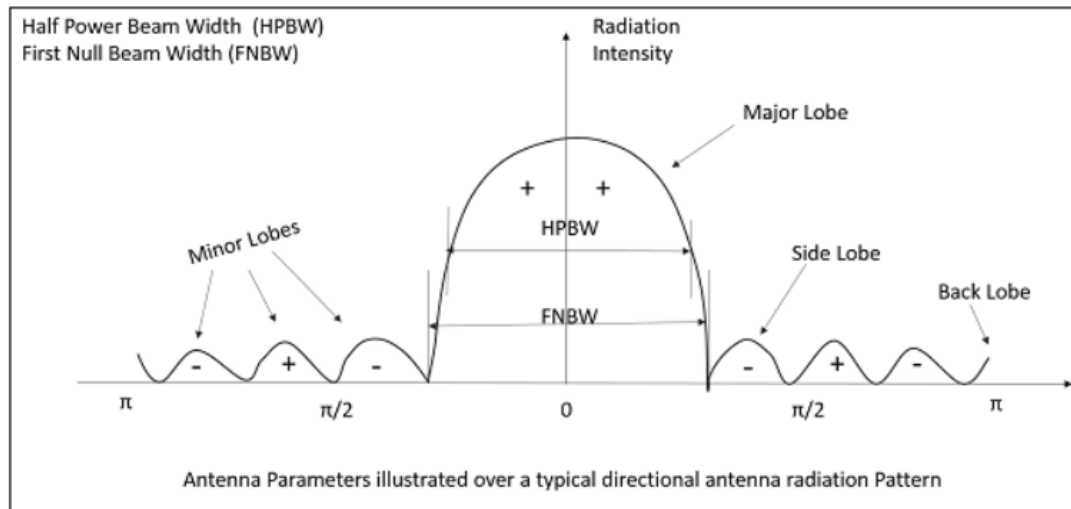
Comumente relacionado com a diretividade, o ganho de uma antena é classificado como a razão da intensidade em uma direção relacionada com a intensidade de radiação de uma antena isotrópica (IEEE, 2004). A razão entre a densidade de potência radiada por uma antena isotrópica,  $P_0$ , com a potência radiada  $P_r$  da antena é definida como a eficiência da antena  $\eta_e$ . O ganho é associado com a diretividade máxima da antena pela relação:

$$G = D\eta_e \quad (2.24)$$

Além disso o ganho é expresso em termos do ganho de potência de uma antena em relação à antena isotrópica, expresso em dBi.

Por último, a largura de feixe ou *beamwidth* é definida como um ângulo entre dois pontos do feixe de radiação, sendo então a separação angular entre dois pontos idênticos em lados opostos do padrão de radiação (BALANIS, 1997). Dessa forma, esse parâmetro é utilizado para demonstrar a capacidade de resolução da antena no apontamento para um alvo ou direção. A largura de feixe de meia potência, HPBW, é a forma mais comum de caracterização. Ela expressa o ângulo entre duas direções onde a intensidade de radiação é a metade do valor de máxima magnitude (IEEE, 2004). Analogamente a isso, a largura de feixe dos primeiros nulos, FNBW, também é utilizada durante a avaliação do padrão de radiação.

Figura 2.23 – Diagrama Linear do Padrão de Radiação.



Fonte : (TutorialsPoint, 2022)

Através da diretividade da antena, pode-se determinar a abertura angular da antena, assumindo que a abertura seja a mesma no plano de elevação e azimute de forma que  $HPBW_{\theta} = HPBW_{\phi}$  (ATLANTARF, 2013).

### 2.8.2 Ressonância

Quando um sistema oscila em maiores amplitudes em uma faixa de frequência, diz-se que esse sistema é ressonante. Uma antena ressonante possui sua impedância puramente resistiva o que facilita o casamento de impedância da antena (MILIGAN, 2005).

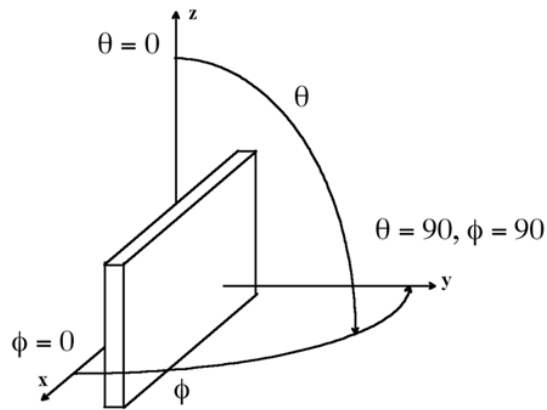
A reatância capacitiva da antena é cancelada com a reatância indutiva, e a antena é puramente resistiva de maneira que na prática isso advém das propriedades físicas ambientais e dimensões da antena (MILIGAN, 2005).

### 2.8.3 Perfil de radiação

O perfil de radiação ou *pattern* de uma antena é a representação gráfica das propriedades de radiação da antena no espaço geométrico, descrevendo assim o modo que a antena irradia ou recebe energia em relação a um ambiente (CISCO, 2007).

Conforme as propriedades matemáticas da radiação da antena, um padrão de radiação é desenhado de acordo com a variação do campo elétrico em função do ângulo de elevação  $\theta$  e azimute  $\phi$  que definem a orientação ou apontamento da antena no espaço. (BALANIS, 1997).

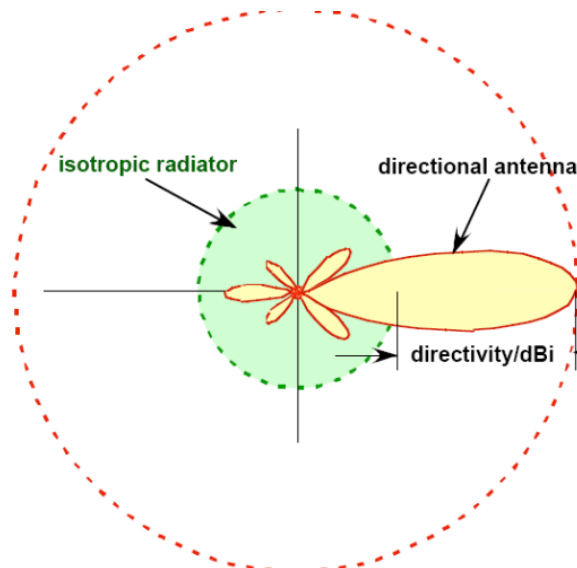
Figura 2.24 – Sistemas de coordenadas espaciais.



Fonte : (CISCO, 2007)

Nesse contexto, popularmente o sistema é caracterizado de acordo com o perfil de radiação, sendo omnidirecional ou seja a onda eletromagnética se propaga para todas direções perpendiculares a um eixo, ou direcional com a focalização da radiação em uma direção e pequena abertura angular (BALANIS, 1997).

Figura 2.25 – Perfis de radiação comuns.



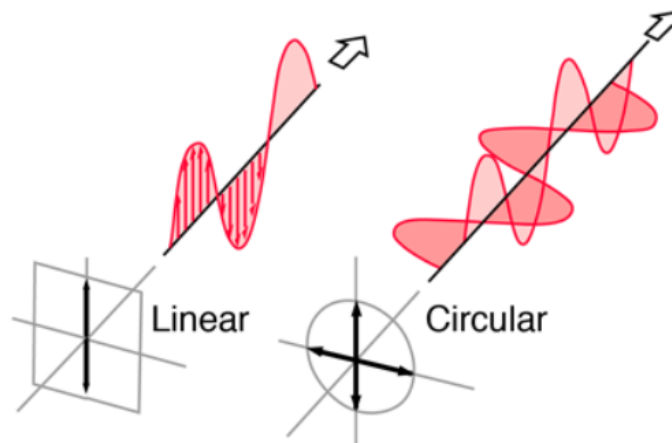
Fonte : (PYTI, 2021)

### 2.8.4 Polarização

A polarização da antena é definida como a polarização da onda transmitida ou irradiada pela antena, ou seja é uma propriedade de uma onda eletromagnética variando seu vetor campo elétrico no tempo observado ao longo da direção de propagação (BALANIS, 1997).

Normalmente a polarização relacionada com a orientação do campo elétrico é caracterizada como linear, circular ou elipsoidal, onde a polarização circular é um caso especial da polarização elipsoidal (MILIGAN, 2005).

Figura 2.26 – Polarização do campo elétrico.



Fonte : (TutorialsPoint, 2022)

Em termos práticos para atingir a polarização linear, seja essa horizontal ou vertical, o campo elétrico deve possuir apenas uma componente, já para a polarização circular o campo deve possuir duas componentes ortogonais de mesma magnitude e com uma diferença de fase de múltiplos de  $90^\circ$  (MILIGAN, 2005).

### 2.8.5 Relação de Onda Estacionária

A relação de onda estacionária ou *SWR* é um parâmetro utilizado durante o casamento de circuitos de constantes distribuídas, como cabos coaxiais e antenas, sendo usualmente utilizada para caracterizar a energia refletida na porta cabo-antena ou porta antena-cabo (WALRAVEN, 2006).

Se a *SWR* é maior que 1, um método de casamento de impedância é necessário haja vista que grande parte do sinal está sendo refletido (STUTZMAN; THIELE, 2013).

Tabela 2.2 – SWR vs Potência refletida.

| SWR | Potência Refletida (%) |
|-----|------------------------|
| 1.0 | 0                      |
| 1.1 | 0.2                    |
| 1.2 | 0.8                    |
| 1.3 | 1.7                    |
| 1.4 | 2.8                    |
| 1.5 | 4                      |
| 1.6 | 5.3                    |
| 1.7 | 6.7                    |
| 1.8 | 8.2                    |
| 1.9 | 9.6                    |
| 2.0 | 11                     |
| 3.0 | 25                     |
| 5.0 | 44.4                   |

Adaptado de (WALRAVEN, 2006).

A tabela adaptada de (WALRAVEN, 2006) fornece o retorno de potência de acordo com a relação de onda estacionária.

### 2.8.6 Casamento de impedância

A impedância da antena é um parâmetro dependente da frequência de operação, método de alimentação, geometria e ambiente de propagação. Durante o projeto deve ser avaliada visando a máxima transferência de potência (MAZDA, 1998). Nos terminais da antena é definida como:

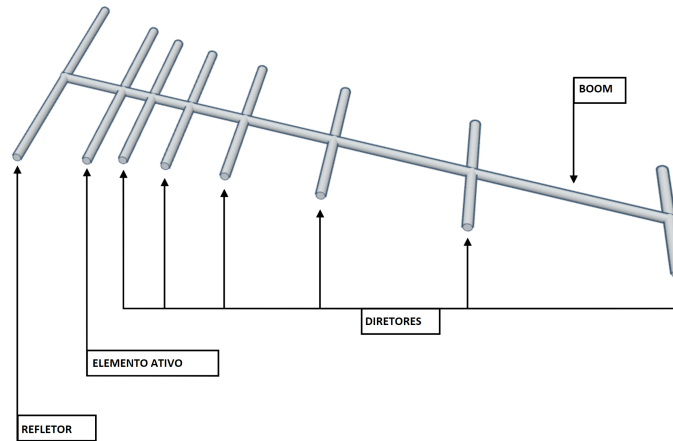
$$Z_a = R_A + jX_a \quad (2.25)$$

Onde  $R_a$  é a resistência da antena em  $\Omega$  composta pelas componentes de radiação  $R_r$  e perdas  $R_l$  no passo que  $X_a$  é a reatância em  $\Omega$ . Partindo do contexto acima, para que ocorra o casamento da antena de acordo com a impedância do gerador de impedância interna  $Z_g$  é necessário que a reatância da antena seja equivalente e de sinal contrário ao da reatância do gerador (BALANIS, 1997).



## 2.8.7 Antena Yagi-Uda

Figura 2.27 – Yagi-Uda



Adaptado de : (CISCO, 2019).

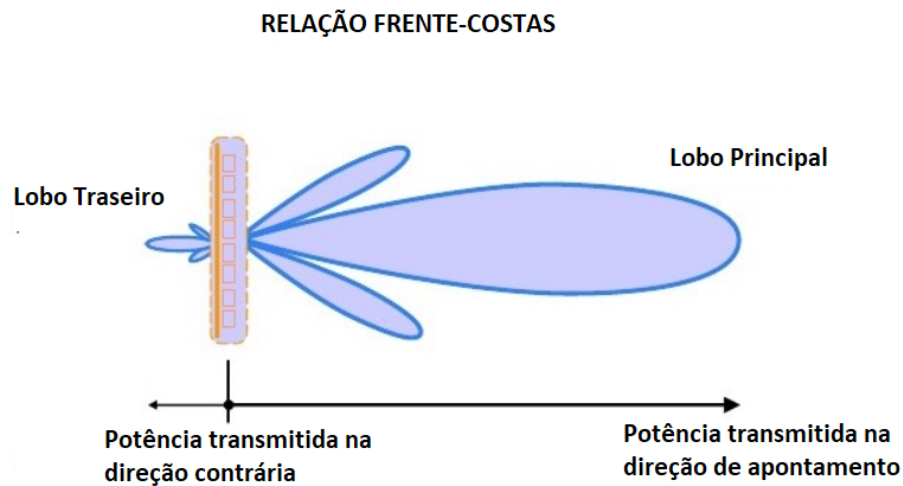
A antena Yagi-Uda é utilizada majoritariamente em situações de enlace de rádio nas faixa HF, VHF e UHF devido sua característica de concentrar a potência radiada em uma direção do espaço, apresentando uma alta diretividade ou ganho e alta relação frente costas, facilidade de construção, baixo peso e custo, sendo implementada nas configurações de polarização linear e cruzada (SWALLOW, 1995).

Ainda mais, a Yagi é formada por um arranjo entre os elementos passivos, diretores e refletores, e ativo que normalmente é um dipolo de  $\lambda/2$  para que durante a operação os diretores direcionem a onda irradiada pelo dipolo e o refletor a reflita (SWALLOW, 1995).

### 2.8.7.1 Relação Frente-Costas

Normalmente a relação Frente-Costas é definida como a razão numérica entre a máxima intensidade de radiação no lobo principal e o valor da intensidade de radiação na direção oposta ou hemisfério posterior da antena (IEEE, 2004). Isso pode ser verificado por meio do ajuste de espaçamento entre os elementos passivos de uma yagi.

Figura 2.28 – Relação frente-costas.



Fonte: Adaptado de (EveRF, 2022)

### 2.8.7.2 Arranjo Parasita

Um arranjo é denominado parasita na medida em que apenas um elemento é excitado os outros elementos adicionais formam um arranjo parasita, radiando o campo gerado pelo elemento ativo alimentado por uma linha de transmissão para que no fim exista a formação do campo distante gerado através da soma dos campos dos elementos passivos (STUTZMAN; THIELE, 2013). Em virtude disso, na prática o refletor ligeiramente maior que o elemento ativo é posicionado atrás desse e por sequência há a adição de elementos diretores de acordo com o tamanho do elemento alimentado para que ocorra um aumento de ganho com a adição de elementos diretores, caso a antena seja projetada corretamente. A incorporação de mais de um elemento parasita ao *boom*, elemento esse que sustenta toda a estrutura dos elementos ativos e parasitas, provê um acréscimo no ganho em relação ao dipolo de meia-onda, a partir de 5 elementos diretores o ganho aumenta em 1dB para cada elemento novo adicionado (BRITAIN, 2009).

De acordo com esse contexto, durante o projeto é necessário adequar o número de elementos e seu espaçamento para chegar em um ganho adequado e em uma abertura angular ideal para uma comunicação satelital. Essa adequação é realizada principalmente com o número de elementos da antena yagi haja vista que o principal fator que altera o ganho é número de elementos parasitas.

### 2.8.7.3 Gamma Match

Visando restringir a perda de potência entre a antena e a linha de transmissão e aumentar a relação SNR bem como o aumentar a potência transmitida e reduzir erros de amplitude e fase um método de casamento de impedância é necessário (MAZDA, 1998). Uma maneira de atingir tal objetivo muito utilizada para um elemento ativo é o casamento gamma ou gamma *match*. Isto é feito através da alimentação assimétrica do dipolo por meio de um arranjo normalmente formado por um capacitor cilíndrico e um tubo de alumínio, de maneira que para uma yagi pode-se ajustar a impedância sem comprometer a integridade do elemento alimentado (BALANIS, 2015).

Assim, determinada a impedância nos terminais e a frequência de operação do elemento ativo o tamanho do acoplador ou tubo gamma é calculado, bem como a capacitância do capacitor que por muitas vezes é pequena devido a natureza do acoplamento entre o tubo e o capacitor cilíndrico formado de dois tubos concêntricos, onde o interno é composto de um material dielétrico que por sua vez determina a capacitância do capacitor (WORTMAN, 2015). Isto é, definida as dimensões da antena e realizada a alimentação assimétrica ocorrerá um incremento na impedância de entrada:

$$Z_2 = \frac{Z_1}{\cos^2\theta} \quad (2.26)$$

Onde  $Z_1$  é a impedância no centro do elemento ativo,  $Z_2$  corresponde a impedância no ponto de toque do acoplamento e  $\theta$  é a distância angular entre centro e ponto de toque (HEALEY, 1969). Como o acoplamento é paralelo ao dipolo a impedância de radiação  $Z_2$  é expressa através da impedância  $Z_0$  do elemento ativo e do tubo gamma.

$$Z_2' = \frac{Z_2 + jZ_0 \tan\theta}{1 + \frac{jZ_2}{Z_0} \tan\theta} \quad (2.27)$$

Onde  $Z_0$  é estabelecido por meio das dimensões geométricas do tubo gamma e do elemento ativo.

$$Z_0 = 276 \log_{10} \frac{2S}{\sqrt{d_1 d_2}} \quad (2.28)$$

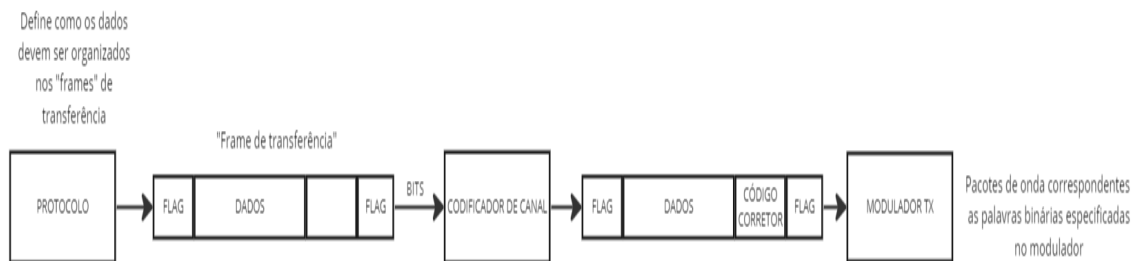
- S = Distância entre o acoplador e o elemento ativo em m.
- d1 = Diâmetro do capacitor cilíndrico em m.
- d2 = Diâmetro do elemento ativo em m.

Devido a orientação contrária das correntes fluindo no tubo gamma e na antena há o cancelamento da impedância  $Z_2$  por uma reatância indutiva  $X_p$ , a impedância do conjunto é reescrita através dessa reatância e da reatância do capacitor cilíndrico que por sua vez é responsável por fornecer apenas a componente resistiva como carga da linha de transmissão (HEALEY, 1969).

## 2.9 PROTOCOLO DE COMUNICAÇÃO

Em meados da guerra fria, a organização americana ARPA, do inglês *A Advanced Research Projects Agency Network*, iniciou o desenvolvimento da comunicação militar por meio de pacotes, sendo este conceito utilizado para fins civis posteriormente no ALOHA-net, um sistema pioneiro de rede de computadores (TOUCH, 2021). Um protocolo de comunicação é definido como um conjunto de regras que expressam como trocar dados em uma rede. Em outras palavras, essas regras determinam como as mensagens digitais ou bits devem ser formatados (KUROSE, 2021). Além disso, os protocolos podem oferecer propriedades como correção e detecção de erros, bem como técnicas de sincronização.

Figura 2.29 – Interação do protocolo no sistema comunicação digital de um transmissor.



Fonte: O Autor

No contexto de sistemas espaciais, a comunicação entre os segmentos é realizada por meio de um protocolo de comunicação espacial, como o AX.25, CSP, NGHAM, CCSDS, entre outros (CAMACHO-LARA, 2013).

O protocolo de comunicação é definido na camada de rede, conforme o modelo ISO (CCSDS, 2020). A integração entre a camada de rede e a camada de enlace de dados é indispensável para compreensão dos processos aqui tratados. Em relação a isso, na camada de enlace de dados, são enviados blocos de dados denominados *frames* (CLEGG, 2020). Cada *frame* é composto por pequenos grupos de dados, conforme ilustrado na figura 2.30. Na prática, a estação terrena recebe um fluxo de símbolos IQ que contém *frames* relacionados a um determinado protocolo (Daniel Esteves, 2020).

Figura 2.30 – *Frame* pertencente ao protocolo AX.25

| Flag     | Address      | Control   | PID    | Info     | FCS     | Flag     |
|----------|--------------|-----------|--------|----------|---------|----------|
| 01111110 | 112/224 Bits | 8/16 Bits | 8 Bits | N*8 Bits | 16 Bits | 01111110 |

Fonte: (TAYLOR, 1998).

O modelo de camadas de dados ISO é uma arquitetura de 7 camadas que descreve

o projeto de uma rede de comunicação. Ele define as funções de comunicação de cada camada e facilita o entendimento do processo de comunicação entre o usuário e a máquina em questão (BARBOZA, 2023). Existe uma relação mútua entre cada camada em termos de tarefas, com a troca de dados transmitidos ou recebidos (BARBOZA, 2023). Dessa forma, há uma compreensão maior de toda a rede de comunicação.

#### 2.9.0.1 Camada física

A camada física é responsável pelo meio físico de transmissão, ou seja, a parte relacionada à radiofrequência que utiliza frequências específicas para *downlink* e *uplink*, com modulação, potência e polarização da antena definidas. Nessa camada, as informações são representadas no formato de bits (CLEGG, 2020). Essa camada estabelece a conexão física entre o satélite e a estação terrena.

#### 2.9.0.2 Camada de Enlace de Dados

A camada de enlace de dados desempenha um papel fundamental na elaboração do presente trabalho. Ela é responsável pelo controle do fluxo da transmissão de dados por meio de *frames*. Nessa camada, os bits são organizados dentro dos *frames* (CLEGG, 2020).

A camada de enlace de dados possui duas subcamadas: a subcamada de protocolo de ligação de dados e a subcamada de codificação de canal e sincronização (CCSDS, 2020). Na prática, durante a recepção, a camada de enlace de dados verifica se uma conexão entre a camada física foi estabelecida ao receber um pacote. Isso é feito por meio da verificação do padrão de bits em um *frame* (CLEGG, 2020). Além disso, durante a transmissão, ela divide os dados em *frames*.

Ainda, essa camada normalmente implementa um código corretor de erro, como o código de Reed-Solomon e o código de checagem cíclica (CRC), para lidar com a possibilidade de *frames* transmitidos fora de sequência ou durante o processo de acesso ao satélite por meio do envio de telecomandos (CCSDS, 2020).

#### 2.9.0.3 Camada de Rede

Essa camada fornece os meios funcionais e procedimentais para transferir sequências de dados de comprimento variável (chamadas de pacotes) de um usuário de origem

para um usuário de destino. Isso inclui o endereçamento, o enfileiramento de pacotes e o encaminhamento de pacotes (CAMACHO-LARA, 2013)

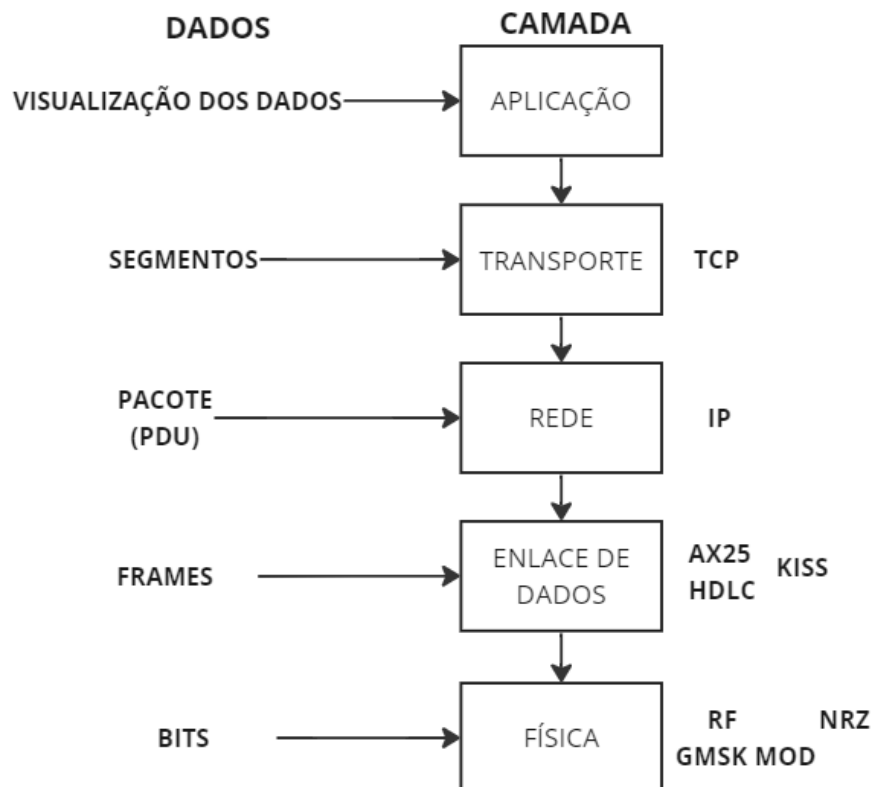
#### 2.9.0.4 Camada de Transporte

A camada de transporte é responsável pela transferência de dados em um formato de pacote (CAMACHO-LARA, 2013).

Durante a transmissão, a mesma recebe uma mensagem da camada de aplicação e realiza a segmentação, dividindo a mesma em segmentos, adicionando destino, segmento de cabeçalho, e transfere para a camada de rede. Já na recepção, essa recebe um pacote da camada de rede, e realiza o processo inverso segmentando-o (CLEGG, 2020).

Na figura 2.31 é possível visualizar de forma imagética as camadas, esquematizada para a constelação de *cubesats* da universidade de Virgínia - VCC.

Figura 2.31 – Camadas de comunicação desenvolvidas para a constelação de *cubesats* VCC.



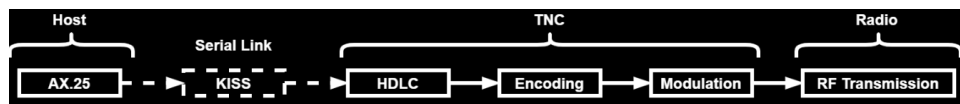
Fonte: Adaptada de (CLEGG, 2020)

Conforme a figura 2.31, o segmento físico, constituído pelos sistemas de radio-

frequência, é responsável pelo envio de telecomandos e recebimento de *beacons* da constelação de CubeSats, constituída por um arranjo de antenas yagi, operando em UHF, com modulação GMSK, codificação de linha NRZ e *bit rate* de aproximadamente 9600 baud (CLEGG, 2020). Já a camada de enlace de dados, é responsável pelos *frames* do protocolo AX.25. A camada pode ser dividida em subcamadas, sendo a camada superior a AX.25, que forma os *frames* AX.25 e sua estrutura e a camada inferior, HDLC, *High Level Data Link control*, que utiliza um código corretor de erro e insere os bits de *flag* (CLEGG, 2020).

O protocolo KISS é utilizado para movimentar dados entre as duas subcamadas, através de *frames* de transferência, por meio de um link RS-232, sendo esse um protocolo padrão para trocar de dados binários. (SHARMA, 2021).

Figura 2.32 – Interação entre os *frames* do AX.25, HDLC e KISS.



Fonte: (NOTBLACKMAGIC, 2023)

O protocolo KISS é usualmente utilizado com o AX.25 devido a grande disponibilidade de software para operação, incluindo decodificadores, como por exemplo o GNU Radio, que possui aplicações que suportam ambos protocolos (KARN, 1986). Os protocolos TCP e IP são utilizados para transportar os dados para as camadas acima, até a de aplicação, no caso TC, referente ao telecomando no sentido de transmissão, e TM, referente a telemetria enviada (CLEGG, 2020).

### 2.9.1 Protocolo AX.25

o protocolo AX.25 ou *AmateurX.25* é um protocolo de camada de rede projetado e desenvolvido especificamente por operadores de rádio amadores que ao longo do tempo passou a ser utilizado para comunicações via satélite (ENGBERG, 2017). Quando a comunicação é estabelecida, o satélite em questão envia os *frames* referentes ao protocolo AX.25. Em virtude disso, conforme (TAYLOR, 1998) existem 3 tipos de *frames* suportados pelo protocolo, o *frame* de informação(I), de supervisão(S) e não-numerado(U). O presente trabalho se limita ao escopo do UI *frame*, correspondente a informação enviada no canal, amplamente utilizado no âmbito de comunicações espaciais, formado pela combinação dos *frames* U e I (ENGBERG, 2017). O de supervisão é referente ao controle de enlace e o não-numerado visa conexão com as demais camadas do modelo de rede (TAYLOR, 1998).

Figura 2.33 – Estrutura do UI *frame*

| Flag     | Address  | Control | PID    | Info     | FCS     | Flag     |
|----------|----------|---------|--------|----------|---------|----------|
| 01111110 | 112 bits | 8 bits  | 8 bits | N*8 bits | 16 bits | 01111110 |

Fonte: (ENGBERG, 2017).

A estrutura do *Unnumbered Information frame* em questão é realizada da seguinte forma, o campo referente ao *Flag* é imposto no início e no fim do *frame*, sendo esse utilizado para marcar o início e o fim do mesmo, neste caso são 1 byte ou 8 bits. Dois *frames* podem compartilhar a mesma *flag*, sendo o fim de um e o início de outro (TAYLOR, 1998). O endereço ou *Address* indica o endereço e origem do *frame* ou o *call-sign*, os bits do campo de controle ou *Control* indica o tipo de *frame* enviado, para o UI o padrão é 00000011 (TAYLOR, 1998). Além do mais, o PID ou *Protocol Identifier* identifica o tipo de protocolo utilizado, ou seja o tipo da camada 3 do modelo ISO (TAYLOR, 1998).

Já o campo de informação ou *Information Field* contém os dados telemétricos em um tamanho máximo de 256 bytes. Por último há o FCS, *Frame Check Sequence* que implementa um código de checagem de erro, usualmente o CRC, *Cyclic Redundancy Check*, que verifica se o *frame* não foi totalmente corrompido pelo meio de transmissão e a *flag*, marcando o fim do *frame* (TAYLOR, 1998).

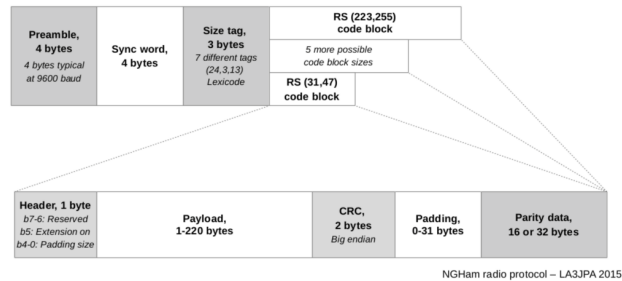
## 2.9.2 Protocolo NGHAM

O protocolo NGHAM, *Next Generation Ham Radio* foi desenvolvido por Jon Petter Skagmo como uma versão alternativa do protocolo AX.25, possuindo uma robustez maior durante a decodificação dos *frames* quando comparado ao AX.25 (SKAGMO, 2023). Uma das suas principais vantagens é a redundância, o protocolo possui 7 tamanhos diferentes de *frames*, permitindo reduzir o número de bytes redundantes inúteis ou *dummy bytes* no fim do *frame* utilizados para simplificação da decodificação no receptor (VEA, 2012). Paralelamente à isso, a codificação pode ser ajustável, usualmente o código RS pode ser ajustado, utilizando o menor número de *dummy bits* possíveis a serem inseridos (VEA, 2012). Ainda, há a implementação do CRC de 16 bytes no *frame*.



O *cubesat* brasileiro, FLORIPASAT-1, utiliza os *frames* do protocolo NGHAM (Space Lab UFSC, 2022). Que possuem a seguinte estrutura:

Figura 2.34 – NGham *frame*



Fonte: (SKAGMO, 2023)

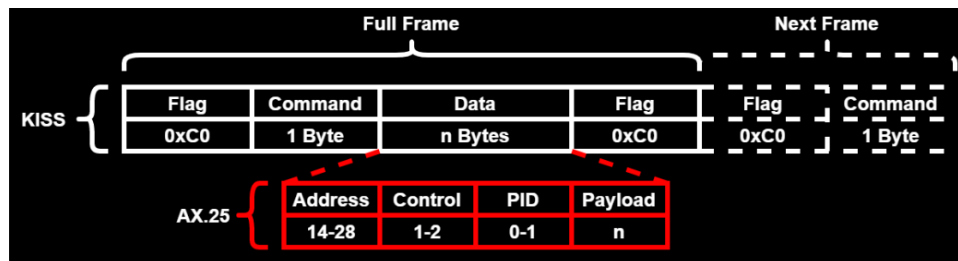
A robustez é conferida devido ao preâmbulo, palavra de sincronização e tamanho de *tag*, o preâmbulo é uma série de bytes padronizada de identificação do início do *frame*. A *tag* indica o tamanho do *frame* e a palavra de sincronização é utilizada para sincronização de *frame* (VEA, 2012).

### 2.9.3 Protocolo KISS

O *Keep it Simple, Stupid* é um protocolo usualmente utilizado em conjunto com outro protocolo, originalmente desenvolvido durante o programa KA9Q NOS conjuntamente dos protocolos TCP e IP (KARN, 1986). Atualmente é quase exclusivamente utilizado para transportar os *frames* do protocolo AX.25 (ALILI, 2021).

Sua simplicidade deriva deste ser um protocolo assíncrono e serial, portanto sem necessidade de sincronização entre dispositivos onde os bits são transmitidos em série, não possuindo suporte para correção de erros já que a conexão serial usualmente é por meio de cabos à curta distância, onde o mesmo é frequentemente utilizado para movimentar dados em subcamadas (ALILI, 2021). Dessa forma, os *frames* pertencentes ao protocolo interagem com os *frames* de outro protocolo, através de um agrupamento, como é o caso do AX.25.

Figura 2.35 – Frames do AX.25 transportados pelo KISS



Fonte: (NOTBLACKMAGIC, 2023)

Nesse contexto, alguns aplicativos desenvolvidos especificamente para visualização de dados telemétricos possuem como entrada um ou mais *frames* KISS, devido sua aplicabilidade com o protocolo de internet, IP (KARN, 1986). Ainda, segundo (KARN, 1986), o formato assíncrono serve para delimitar os *frames* por meio do FEND ou fim do *frame*, cuja função é muito similar à de uma *flag*. O FEND é inserido no início e no fim do *frame*, conforme a figura 2.36, sendo seguido pelo comando, o campo referentes aos dados e outro FEND.

Figura 2.36 – Frames do protocolo KISS

| Begin | Command  | Data 0 ... Data N | End  |
|-------|--|-------------------|------|
| FEND  | High nibble - Port index<br>Low nibble - Command | Data              | FEND |

Fonte: (VEA, 2012)

O *telemetry viewer* do *cubesat* AlfaCruX projetado pela Universidade de Brasília utiliza os *frames* KISS, para visualização dos dados telemétricos recebidos (LodeStar Aerospace, 2022).

#### 2.9.4 Protocolo CSP

O CSP, do inglês *CubeSat Space Protocol*, foi originalmente desenvolvido em 2008 por um grupo de estudantes da universidade de Aalborg e implementado anos depois na GOMspace, uma empresa dinamarquesa do setor aeroespacial (GOMSPACE, 2011).

Tratando-se de um protocolo de camada de rede projetado especificamente para CubeSats, o CSP permite uma integração dinâmica com as demais camadas do modelo de rede, facilitando assim a integração do subsistema de telecomunicações com os demais

subsistemas. O protocolo foi concebido para ser implementado em sistemas embarcados, como os microprocessadores AVR e ARM da Atmel (GOMSPACE, 2011). Também Possui um cabeçalho de 32 bits, normalmente codificado por um código de Golay (LodeStar Aerospace, 2022)

No presente trabalho, o CSP implementado na camada de rede é utilizado em conjunto com os *frames* de transferência (ASM+Golay) do transceptor NanoCom AX100, implementado na camada de enlace da dados (LodeStar Aerospace, 2022). Ainda, O formato do *frame* de *downlink* referente ao CSP não é divulgado pela AlfaCruX e GOMSpace.

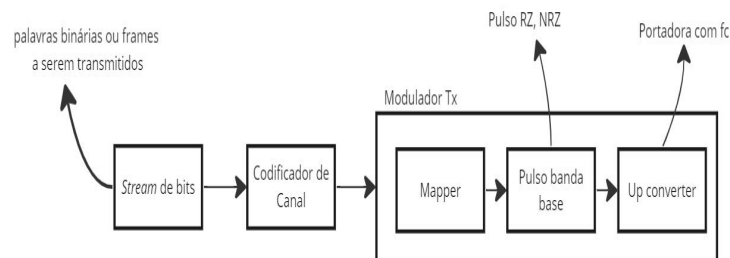
## 2.10 CODIFICAÇÃO DE CANAL

A codificação de canal é definida como uma técnica de processamento de sinais onde dados transmitidos, apesar de distorcidos pelo canal de comunicação, ainda sim são identificáveis no receptor (JOHNSON, 2011). Em uma transmissão de órbita baixa essas alterações são usualmente devidas a perda em espaço livre, e o ruído térmico do receptor (BRITAIN, 2009).

O codificador é projetado visando adequar uma taxa de erro definida pelo usuário e reconstrução dos dados binários originais no receptor, através do projeto do decodificador, que realiza o processo inverso (JOHNSON, 2011). Ainda, existem outros benefícios propostos como a obtenção de um *trade-off* entre potência do transmissor e largura de banda, quando a potência do transmissor é limitada é possível aumentar a largura de banda, se esta está disponível, por meio da adição de bits de redundância utilizado para detecção de erros no receptor (CAMACHO-LARA, 2013). Segundo a teoria da informação de Shannon, a redundância adicionada possibilita um uso eficiente do canal, com uma BER baixa (ALVIM, 2021). Dessa forma, com uma largura de banda pré-definida, para um *baud rate* definido, uma maneira de aumentar a performance do sistema é através da codificação performada por um código corretor de erro. Esse faz uso da redundância para detectar e corrigir erros, seja na transmissão ou recepção, isso permite um ganho, chamado ganho de codificação, reduzindo a potência de transmissão do transmissor (BOUALLEGUE, 2015).

De início, antes de abordar sobre o funcionamento de modo genérico de um decodificador de canal, é necessário visualizar através de um diagrama de blocos o papel do codificador em um sistema de comunicação digital.

Figura 2.37 – Interação do codificador de canal no sistema comunicação digital de um transmissor.



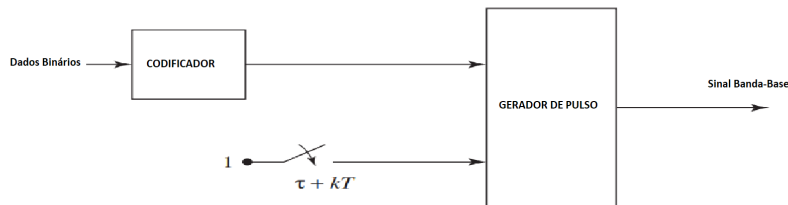
Fonte: O Autor

Se a informação original for constituída de um conjunto de bits não há necessidade de um conversor ADC, para digitalização da informação analógica. A mensagem binária B é inserida no codificador de canal digital, o codificador de canal adiciona redundância

a mensagem, através da adição de bits de paridade, e correlaciona a mensagem para o alfabeto de símbolos IQ,  $(-1, 1, +3, -3)$ , no *mapper* do modulador digital (JOHNSON, 2011). Por exemplo, a mensagem  $a$  é representada através dos binários como  $(01\ 10\ 00\ 01)$ , correspondentes aos símbolos IQ mapeados  $(-1\ 1\ -3\ -1)$ . Nesse caso, os símbolos estão agrupados em grupos de 4, constituindo um *frame* (JOHNSON, 2011).

O sinal ou pulso em banda-base tem seu formato modelado de acordo com a saída do decodificador, tal que o bit "1" é transmitido quando  $0 \leq t \leq T$ , onde  $T$  é o intervalo de tempo entre símbolos transmitidos, logo, o próximo pulso é no intervalo  $\tau + kT$ , onde  $k$  é o número de pulsos. Caso contrário o bit "0" é transmitido (JOHNSON, 2011).

Figura 2.38 – Codificador de linha.



Adaptado de : (JOHNSON, 2011)

Por exemplo, para transmitir o bit "0", a sequência "000" é enviada, sendo dois bits adicionados pelo codificador. O decodificador deve possuir conhecimento das sequências transmitidas, identificando e corrigindo os erros no receptor (JOHNSON, 2011). Caso a sequência recebida seja "100", o decodificador deve assumir que o erro foi no primeiro bit.

Ademais, a utilização da codificação adequada a probabilidade de erro definida em um projeto. Seja a banda alocada no canal de 3KHz, para um modulador operando com um *baud* ou bit *rate* de 3600 bits/s e um SNR definido de 13dB. Do teorema de capacidade do canal de Shannon, a capacidade do canal é definida em  $\frac{\text{bits}}{s}$  como:

$$C = B \log_2(1 + S/N) \quad (2.29)$$

Seja  $B = 3\text{KHz}$  e  $S/N = 13\text{dB}$ , o canal permite um bit rate de até 13000 bits/s, logo o bit rate do modulador é adequado ao canal. Como esta taxa ainda é menor que a capacidade do canal, a probabilidade de erro pode ser reduzida até um nível desejado, através da implementação do codificador de canal. Para adequar a probabilidade de erro do sistema, um código corretor de erro é adicionado, reduzindo a probabilidade de erro a uma taxa atendível pelo projeto do modulador, vale ressaltar que a taxa de erro deve ser mantida em

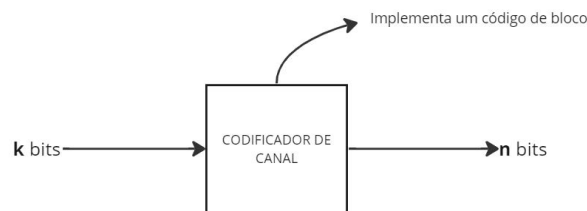
um limite aceitável definido pelo usuário (BRITAIN, 2009). Esse código implementado no codificador de canal oferece vantagens, em termos de BER e SNR, para a transmissão.

### 2.10.1 Código Corretor de Erro

As duas maiores classes de códigos corretores de erro são os códigos de bloco e os códigos convolucionais, o presente trabalho se limita aos códigos de bloco, onde os códigos de Reed-Solomon e Golay se enquadram. Essa classe de código usualmente pode ser decodificada por meio de uma síndrome e matriz de paridade (ALOUINI, 2020). Esse subgrupo de códigos são populares devido a viabilidade de implementação e performance, que é similar aos demais códigos, como os convolucionais (PROAKIS, 2008).

O codificador de canal é responsável por gerar  $n$  bits codificados a partir de  $k$  bits de informação ou bits mensagem, tornando-se assim um código de bloco  $(n, k)$  (MORELOS-ZARAGOZA, 2006).

Figura 2.39 – Codificador de canal.



Fonte: O Autor.

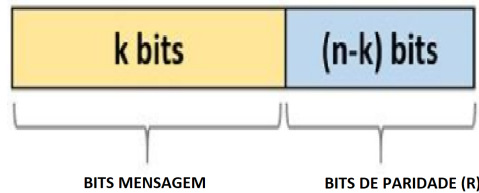
No caso de um esquema de modulação binário, o código é composto pelos símbolos 0 e 1, sendo, portanto, um código binário (PROAKIS, 2008). Na etapa de recepção, os bits redundantes são removidos na tentativa de recuperar a mensagem original. Isso pode ser feito através de dois métodos de decisão conforme o *de-mapper* do demodulador: utilizando decodificadores que operam com valores ou símbolos IQ suaves (*soft decision decoders*), ou utilizando decodificadores que operam com decisões abruptas (*hard decision decoders*) (MORELOS-ZARAGOZA, 2006).

Além disso, no contexto mencionado, é possível definir um código de bloco utilizando um operador  $\theta$ , que realiza o mapeamento entre os bits mensagem e os bits codificados. Em outras palavras, o processo de codificação é realizado por um código binário  $\theta(n, k)$ , onde  $k$  representa os bits da mensagem a serem codificados em palavras código de comprimento  $n = k + r$  bits. Nesse caso,  $r$  representa os bits redundantes adicionados (MORELOS-ZARAGOZA, 2006) ou bits de paridade. Adicionalmente, a razão de codifi-

cação,  $R_c$ , pode ser definida como a relação entre a duração de uma palavra código e a duração de uma mensagem.

$$R_c = k/n \quad (2.30)$$

Figura 2.40 – Estrutura da palavra código.



Fonte: Adaptado de (ROSHNI, 2020)

Na prática, por exemplo, no código  $\theta(4, 3)$  cada palavra código é formada por 4 bits, isto é feito adicionando-se 1 bit de paridade, utilizado para verificar erros durante a transmissão da mensagem de 3 bits (JOHNSON, 2011). Por isso,  $\theta(4, 3)$  é um código sistemático, pois cada palavra código de  $n$  bits é formada por  $k$  bits da mensagem, somados de  $r$  bits de paridade. Ainda, nesse contexto, o código binário  $\theta(n, k)$  também pode ser representado através de uma matriz geradora  $G$ , que permite realizar operações referentes ao processo de codificação e decodificação de forma matricial (MIT, 2012a). Caso a matriz geradora seja escrita através da matriz  $P$ , que define os  $n - k$  bits de paridade na palavra código, e a matriz identidade, na seguinte forma sistemática:

$$G = [I_k P] \quad (2.31)$$

- $I_k$  = Matriz identidade de ordem  $k$ .
- $P$  = Matriz de paridade de ordem  $k \times (n-k)$ .

O Equacionamento 2.26 ao ser manipulado algebricamente através de uma transposição matricial permite a obtenção da matriz de paridade,  $H$ , utilizada para verificação de erros da palavra código recebida (MORELOS-ZARAGOZA, 2006). Caso o resultado da multiplicação entre a palavra código recebida e a matriz de paridade transposta seja:

$$y_i H^t = 0 \quad (2.32)$$

A palavra código recebida é igual a palavra código transmitida, ou seja a mensagem foi recebida sem erros. Esse conceito também é utilizado na decodificação por arranjo padrão, utilizada para decodificar códigos lineares, como os códigos de bloco (MIT, 2012a).

Ademais, os conceitos de distância de Hamming e síndrome são necessários para abordagem da decodificação por arranjo padrão. Diante disso, a distância de Hamming, é definida como o número de componentes diferentes, ou seja o número de bits em posições correspondentes que são diferentes entre duas palavras código (PROAKIS, 2008). Seja  $C_1 = [0101]$  e  $C_2 = [1000]$ , a distância de Hamming é 3. Isso influencia na capacidade do código corretor de erro, haja vista que quanto maior a distância, maior a capacidade de correção. Um exemplo de código que faz o uso disso é o código de verificação cíclica de segurança, o CRC, que possui capacidade de corrigir 3 à 4 bits por frame, ele também utiliza dos conceitos de códigos polinomiais, sendo calculado por um polinômio gerador fixo, que é utilizado para a verificação da integridade dos dados (CHAKRAVARTY, 2004).

Ainda, nesse contexto, o padrão de erro entre a palavra código transmitida e a palavra código recebida tem seu peso dado através da distância de Hamming. O peso do padrão de erro  $e_i$  é associado com a síndrome do padrão de erro,  $S_i$  (MIT, 2012a).

$$S_i = e_i H^t \quad (2.33)$$

A síndrome é necessária para identificação do padrão de erro associada a síndrome calculada. O processo de decodificação por arranjo padrão é composto das seguintes etapas:

- 1 Cálculo da síndrome através da multiplicação da palavra-código recebida  $y_i$  e a matriz de paridade transposta
- 2 Identificação do padrão de erro  $e_i$
- 3 Cálculo da palavra-código decodificada
- 4 Recuperação da mensagem transmitida originalmente

O cálculo da palavra-código decodificada é performado através de:

$$C_{decod} = y_i + e_i \quad (2.34)$$

A recuperação da mensagem transmitida, para um código sistemático, consiste da eliminação dos bits de paridade  $r$  de  $C_{decod}$  (MIT, 2012a).

Adicionalmente, como o código de bloco também pode ser representado através de um polinômio gerador, o processo de codificação/decodificação neste caso envolve operações algébricas conforme um campo algébrico, usualmente o Campo de Galois (GF) (MORELOS-ZARAGOZA, 2006).

A utilização do Campo de Galois facilita a implementação da codificação, pois as operações de soma e multiplicação em GF(n), onde n representa o número de elementos finitos do campo, podem ser realizadas através de portas lógicas (TANOMARU, 2021).



### 2.10.1.1 Reed-Solomon (RS)

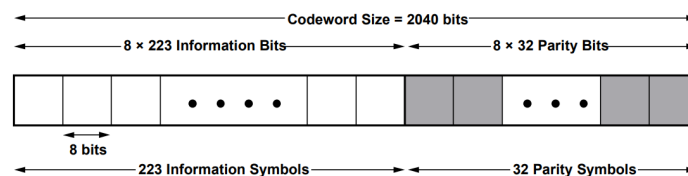
O código Reed-Solomon é englobado na classes dos códigos de blocos, sendo uma sub-classe dos códigos BCH (Bose-Chaudhuri-Hocquenghem), com alta eficiência de codificação e decodificação, desenvolvido inicialmente na década de 60 (RODOLFO, 2007). Atualmente são utilizados em grande escala nos sistemas de comunicações digitais, sua implementação varia desde dispositivos de armazenamento como códigos de barra, CDs à comunicação *wireless* e via satélite (RODOLFO, 2007). Isto confere uma alta complexidade ao código, sendo necessário conceitos relacionados a campos finitos, elementos primitivos, GF e LFSR (BERLEKAMP, 1968).

A propriedade cíclica facilita o cálculo das síndromes e erros, já que uma mudança da palavra código ou *bit flip* acabará neste caso gerando outra palavra código, no codificador isto é realizado por meio da implementação de um circuito eletrônico de registrador de deslocamento ou *shift register* (TANOMARU, 2021).

Paralelamente ao código binário  $\theta(n, k)$  que codifica mensagens de  $k$  bits em palavras código de  $n$  bits, o RS, representado por  $RS(n, k)$  codifica mensagens de  $k$  símbolos em palavras código de  $n$  símbolos, cada símbolo é um bloco de  $m$  bits (CCSDS, 2020). A capacidade de correção de bits é dada por:

$$t = \frac{n - k}{2} \quad (2.35)$$

Figura 2.41 – Estrutura da palavra código RS(255,223)



Fonte: (CCSDS, 2020)

Um código muito utilizado no contexto de nanossatélites é o código RS(255,233), utilizando por norma símbolos de  $m = 8$  bits, correspondente à 1 byte. Dessa forma, 233 símbolos são codificados em palavras código de 255 símbolos, com 22 símbolos de paridade para cada 233 bytes de dados. A capacidade de correção é  $t = 11$  símbolos, para a sequência de 255 símbolos recebidos.

Durante a codificação os símbolos são descritos de acordo com o formato polinomial dos códigos de bloco, sendo coeficientes denotados por  $m$  do polinômio conforme a equação 2.31.

$$m(x) = m_0 + m_1x + m_2x^2 \dots m_nx^n \quad (2.36)$$

A manipulação da mensagem no formato polinomial segue as operações aritméticas do campo de Galois (GF) , para que o codificador adicione os símbolos de paridade à mensagem (PRASAD, 2015). Isso é feito de forma sistemática através de operações no GF, relacionando o polinômios gerador, de paridade e o polinômio codificado de saída. Nesse contexto, os símbolos de paridade são adicionados a mensagem de  $k$  símbolos, formando a mensagem codificada  $c(x)$  através de  $k$  coeficientes do polinômio mensagem e  $2t$  símbolos de paridade que são coeficientes de um polinômio redundante (PRASAD, 2015).

$$c(x) = x^{2t}m(x) + p(x) \quad (2.37)$$

- $x^{2t}$  = Fator multiplicativo da notação polinomial.
- $m(x)$  = Polinômio mensagem.
- $p(x)$  = Polinômio redundante.

Ainda  $p(x)$  é definido através do polinômio gerador  $g(x)$  cujo grau é proporcional a quantidade de símbolos de paridade, onde  $g_0, g_1$  são coeficientes do campo de Galois.

$$g(x) = g_0 + g_1x + g_2x^2 + \dots \quad (2.38)$$

Em aspectos práticos , conforme (PRASAD, 2015) a equação 2.33 é descrita através de  $\alpha$ , um elemento primitivo do GF (BERLEKAMP, 1968). A codificação é realizada em um circuito eletrônico D Flip-Flop que implementa as tabelas de adição e multiplicação do GF em uma FPGA, sincronizadas através de um *clock* (TANOMARU, 2021). A mensagem  $m(x)$  descrita através do elemento  $\alpha$  é inserida no circuito, após os ciclos necessários do clock o registrador do circuito possui armazenado os símbolos de paridade, que são adicionados à  $m(x)$ .

Já o processo de decodificação, é realizado por meio das etapas de cálculo da síndrome, essa etapa é fundamental pois nela há o mapeamento das síndromes em seus erros correspondentes, pois, os conjuntos de síndromes são determinado pelo conjunto de padrões de erro que usualmente acometem o sistema (PRASAD, 2015). A verificação de erros na palavra código recebida é realizada por meio da substituição das raízes,  $\alpha$ , do polinômio gerador  $g(x)$  no polinômio recebido pelo sistema  $r(x)$ , ainda,  $r(x)$  é expressa pelo polinômio codificado e o polinômio de erro tal que:

$$r(x) = e(x) + c(x) \quad (2.39)$$

A substituição, que permite o cálculo da síndrome  $S_i$ , é realizada por meio de:

$$S_i = r(\alpha^i) \quad (2.40)$$

Ademais, uma palavra código para o RS pode ter até  $2t$  síndromes, dependendo somente de  $e(x)$ . Caso o resultado da equação 2.35 seja nulo,  $r(x) = c(x)$ . A próxima etapa se resume a encontrar a localização ou posição dos erros, usualmente feita através do algoritmo de Chien, *Chien Search*, em conjunto com algoritmo de Berlekamp-Massey (PRASAD, 2015). Por último o algoritmo de Fourny é utilizado para determinar o valor ou magnitude dos erros, para que estes possam ser corrigidos, recuperando a mensagem originalmente transmitida (PRASAD, 2015).

### 2.10.1.2 Golay

O código de Golay, criado por Marcel Golay e consolidado no ano de 1974, é usado em uma variedade de aplicações, incluindo telecomunicações, criptografia e armazenamento de dados. Uma das suas principais características é que ele possui uma estrutura muito regular, o que o torna fácil de implementar em hardware e software (MORELOS-ZARAGOZA, 2006). Diferentemente do Reed-Solomon, o código trabalha com bits e não bytes, sendo um código de bloco, codificando um bloco de bits de tamanho fixo pré determinado.

Ainda, devido ao seu formato,  $\theta = (23, 12)$ , o mesmo é chamado de código perfeito de Golay já que este codifica 12 bits de dados, em 23 bits codificados, corrigindo até 3 erros simultâneos e detectando até 6 erros simultâneos (SINGH, 2018). Ao adicionar um bit de paridade o código é estendido,  $\theta(24, 12)$ , muito utilizado no escopo de comunicações de baixa órbita, a distância de Hamming neste caso é 8. Devido suas dimensões, é possível operar o código através de uma tabela de busca ou LUT, *Look Up Table*, mapeando os dados de entrada nas respectivas palavras código correspondentes (MORELOS-ZARAGOZA, 2006).

A tabela de busca do codificador é criada gerando todas as possíveis combinações de vetores de entrada de 12 bits, nesse processo, a LUT deve ser contruída contendo a lista de bits codificados, referente à  $2^k = 2^{12} = 4096$  bits (MORELOS-ZARAGOZA, 2006). Ainda, ocorre o cálculo da síndrome de 11 bits anexada à parte menos significativa (LSB) da palavra de código correspondente de 23 bits.

Nesse escopo, a decodificação é performada gerando todos os possíveis padrões de erro com uma distância de hamming menor ou igual à 3, onde para cada padrão de erro uma síndrome contem informações sobre os erros que podem ter ocorrido durante a transmissão (SINGH, 2018). Esse vetor de síndrome é então usado como entrada na tabela de busca do decodificador Golay.

A tabela de busca do decodificador Golay contém informações pré-computadas sobre os padrões de síndrome e os vetores de erro correspondentes. Com base na síndrome recebida, a tabela de busca identifica o vetor de erro mais provável e o retorna como saída.

Esse vetor de erro estimado é usado para corrigir os bits errados no vetor recebido e recuperar os dados originais, aplicando o vetor de erro identificado à palavra recebida, portanto, realizando a correção dos bits errados (MORELOS-ZARAGOZA, 2006). O decodificador é implementado então através da decodificação aritmética do código de Golay, proposta por Vanstone e Oorschot (MORELOS-ZARAGOZA, 2006). A primeira etapa a ser realizada é o cálculo da síndrome  $s$ , utilizada para verificação de um padrão de erro na palavra código recebida:

$$s = rH^t \quad (2.41)$$

- $r$  = Vetor de  $n$  bits representante da palavra código recebida.
- $H^t$  = Matriz de paridade transposta.

Calculado a síndrome, caso a distância de hamming do padrão de erro associado a esta, for menor ou igual à 3, há 3 erros na palavra código recebida. Nesse caso, o peso do padrão de erro é a distância de hamming entre a palavra código recebida e a palavra código enviada, sendo este igual à 3. Assim, a decodificação é realizada por:

$$c = r + e \quad (2.42)$$

Onde:

- $r$  = palavra código recebida.
- $e$  = vetor de erro estimado.

Caso contrário, se há mais de 3 erros na palavra, o algoritmo proposto não é capaz de performar a decodificação (MORELOS-ZARAGOZA, 2006).

### 3 METODOLOGIA

#### 3.1 CENÁRIO

A maior parte das pesquisas espaciais de baixo custo realizadas por universidades utiliza *cubesats*. Para estabelecer contato com estes nanosatélites em órbita, são necessárias estações de TTeC (Estações de Telemetria, Rastreamento e Controle). Essas estações permitem a comunicação entre os responsáveis pelo projeto e possíveis operadores. Sendo assim, a implementação viável deste tipo de projeto foi possível graças à utilização de sistemas de rádio definido por software, que permitem realizar várias etapas de operação por meio de software ao invés de hardware, simplificando assim a estação terrena e aumentando a viabilidade de seu projeto.

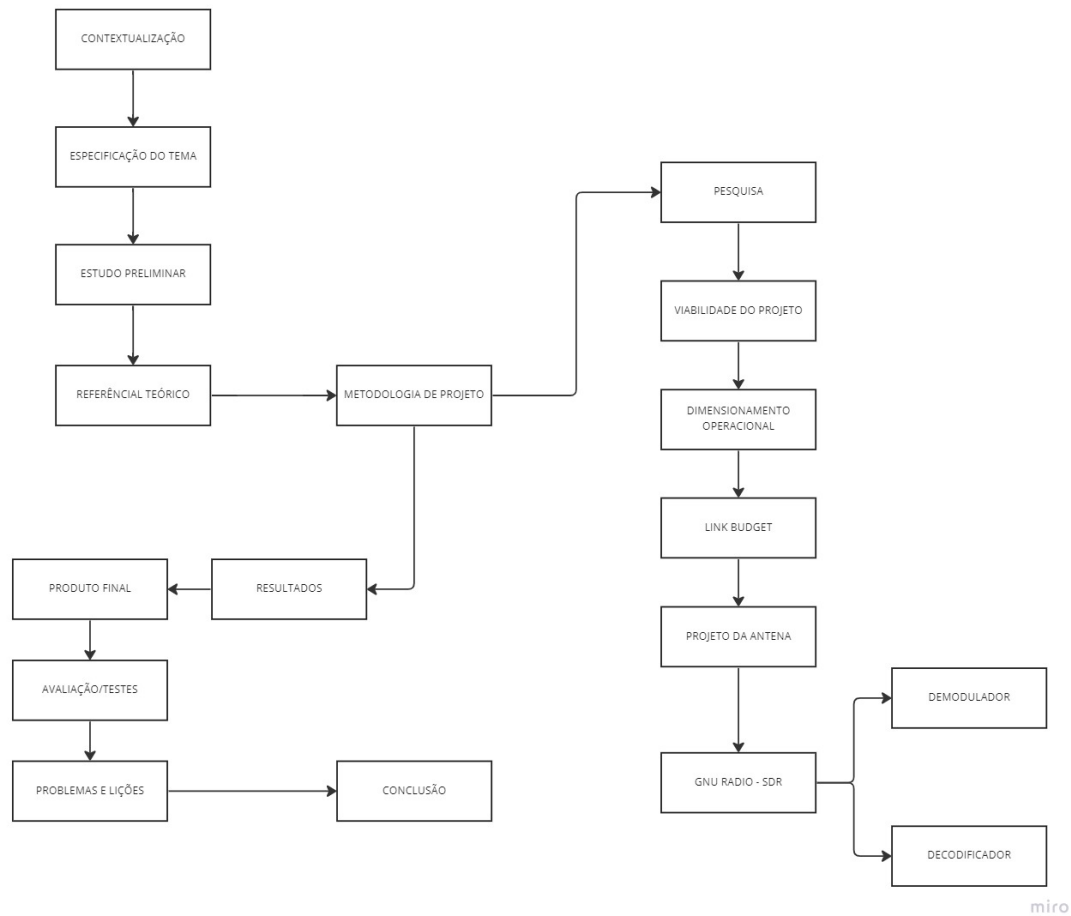
#### 3.2 MÉTODO DE PESQUISA

O presente trabalho de natureza é de aplicada, produzindo conhecimentos de aplicação prática através de um referencial teórico para elaboração da estação terrena, conforme os objetivos definidos na seção 1.1

#### 3.3 ETAPAS DA PESQUISA

As etapas de pesquisa foram definidas conforme a figura 3.1, que seguem a metodologia adotada.

Figura 3.1 – Metodologia Adotada.



Fonte: O Autor

Em um primeiro momento, após a especificação do tema, foi realizado um estudo preliminar sobre o tópico, visando esclarecer dúvidas fundamentais sobre a implementação. Para isso, foi necessário sanar tais dúvidas quanto às características de interação entre a estação terrena e *cubesat*, a fim de que o projeto pudesse ser classificado como viável. Em seguida, foi definido um ou mais *cubesats* para operação e, através de uma pesquisa, verificou-se a viabilidade. A partir disso, todo o trabalho foi desenvolvido.

A primeira etapa foi o dimensionamento da estação, o que permitiu a definição das tabelas 4.2 e 4.5, fundamentais para a elaboração inicial do trabalho. Com base nesses resultados, o *link budget* foi calculado, verificando a sensibilidade do receptor ao sinal recebido. Assim, a antena pôde ser projetada, com o ganho orientado pelos resultados do *link budget*. Para o projeto da antena, foi realizado um dimensionamento preliminar no software MATLAB e posteriormente simulou a antena no software 4NEC2. Ao mesmo tempo, a estimativa inicial do casador de impedância foi feita em software.

Com isso, a antena pôde ser confeccionada posteriormente, bem como o casador de impedância.

As etapas seguintes visam realizar o processo de demodulação e decodificação no software GNU Radio. Para atingir esse objetivo, é importante trabalhar a partir do compilado de informações a respeito dos *CubeSats*. As principais informações para o segmento de processamento de sinais no GNURadio referente ao demodulador e decodificador são a modulação, o tipo do *frame* especificado pelo protocolo e a codificação utilizada.

Nesse sentido, a partir dessas informações é possível implementar tais segmentos, por meio de uma metodologia de processamento de sinais IQ, difundida no trabalho. Por fim, o objetivo final desse trabalho consiste na obtenção dos *frames* decodificados contendo informações telemétricas, bem como a integração dos segmentos do SDR, antena e computador.

Tabela 3.1 – Cronograma

|   |          |
|---|----------|
| Levantamento de dados   | 08/2022  |
| Dimensionamento da estação  | 08/2022  |
| Confecção do <i>Link Budget</i>   | 10/2022  |
| Projeto Preliminar da Antena  | 11/2022  |
| Confecção da Antena   | 12/2022  |
| Levantamento de Dados - Modulação e Decodificação                           | 12 /2022 |
| Teste preliminar em Campo da Antena   | 1/2023   |
| Demodulador - GNU Radio   | 1/2023   |
| Decodificador - GNU Radio   | 5/2023   |
| Integração GNU Radio - Gr Satellites  | 4/2023   |
| Processamento de um <i>stream</i> de símbolos IQ em banda-base no GNU Radio | 5/2023   |
| Decodificação FLORIPASAT-1/AlfaCruz   | 6/2023   |
| Teste em operacional em campo   | 7/2023   |

Fonte: O Autor

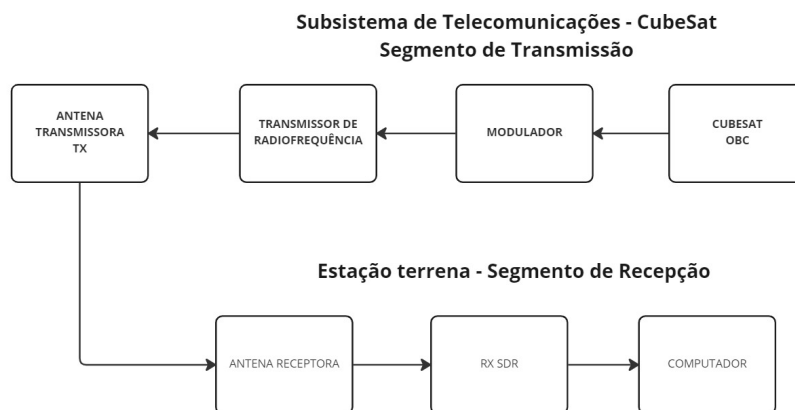
## 4 DESENVOLVIMENTO

### 4.1 DIMENSIONAMENTO DA ESTAÇÃO

Para a definição do sistema de recepção um dimensionamento visando avaliar as características do sistema deve ser realizado. Isto é feito por meio dos parâmetros relacionados ao subsistema de telecomunicação do nanossatélite visando estabelecer todos os critérios dos componentes da estação.

O subsistema de comunicações pode variar dependendo do tipo e quantidade de componentes, no entanto de uma maneira geral e resumida o processo de recepção do sinal pela estação terrena pode ser resumido por meio do seguinte esquemático:

Figura 4.1 – Diagrama de blocos representando a interação entre os sistemas.



miro

Adaptado de (Tsuruda, 2017)

Tendo em vista esse contexto, no presente trabalho, sabendo que o objetivo principal da missão é realizar a recepção e manipulação de dados de telemetria, todo o segmento em terra do subsistema de telecomunicações deve ser dimensionado para isso. Ou seja há um desenvolvimento em etapas para ser seguido de acordo com uma tabela de requisitos.

Baseado na tabela confeccionada por (Tsuruda, 2017), é possível expor os requisitos para elaboração da estação.



Tabela 4.1 – Tabela de projeto adaptada.

|                                       |   |
|---------------------------------------|---|
| Identificação de Requisitos           | Recepção de Dados Telemétricos          |
| Seleção de Frequência                 | Tipo : Frequência amadora VHF/UHF       |
| Seleção de Hardware                   | <i>Cross yagi</i> , SDRPlay, Computador |
| Seleção do tipo de protocolo de dados | NGHam, CSP                              |

Fonte: O Autor.

#### 4.1.1 *Link Budget*

Conforme visto na seção 2.6, o *link budget* é indispensável para a confecção de um sistema de radiofrequência. Nesse sentido o presente trabalho foca em um sistema de recepção que faz o uso das frequências de radio amadoras correspondentes as faixas de 145.8 MHz à 146.0 MHz na banda VHF e 435.0 à 438.0 MHz em UHF.

Para confeccionar o *link budget* em um primeiro momento para dimensionamento da estação foram compiladas informações a respeito dos projetos AlfaCruz e FLORIPASAT-1, ambos brasileiros já validados e lançados em LEO à uma altitude de 1400 quilômetros em *slant range*.

Tabela 4.2 – Tabela confeccionada com base nas referências (Space Lab UFSC, 2022) e (LodeStar Aerospace, 2022).

| CubeSat                   | FLORIPASAT-1          | AlfaCruz    |
|---------------------------|-----------------------|-------------|
| Modulação                 | 2-GFSK                | GMSK        |
| Frequência de operação    | 145.9 MHz - 436.1 MHz | 437.1 MHz   |
| Potência de saída - Ptx   | 30 dBm (1w)           | 30dBm (1w)  |
| Taxa de transmissão (bps) | 1200-2400             | 4800        |
| Polarização da Antena     | Circular              | Circular    |
| Protocolo de dados        | AX.25 ou NGHAm        | CSP ou CCDS |
| Órbita                    | LEO                   | LEO         |

Fonte: O Autor

Por meio do levantamento de dados iniciais na tabela 4.2 o equacionamento para o *link* é realizado visando o objetivo de que o módulo SDR é alimentado por um sinal cuja a demodulação seja realizável. Isto é feito por meio da medição da voltagem na entrada dos terminais do receptor que posteriormente é comparada com a voltagem de sensibilidade presente no material do módulo, verificando assim a sensibilidade do receptor para que o nível necessário de sinal para operação seja atendido.

Por último, o cálculo referente ao *link budget* será apresentado na seção de resultados.

## 4.2 ANTENA RECEPTORA - YAGI UDA

Para definição da antena é necessário verificar suas dimensões ou estrutura, já que os elementos parasitas e o elemento ativo definem sua diretividade (MILIGAN, 2005). No caso da yagi é necessário adequar o número de elementos para chegar em um ganho adequado para que o *link budget* seja realizável. Essa adequação é realizada principalmente com o número de elementos do mecanismo haja vista que o principal fator que altera o ganho é o número de elementos parasitas (STUTZMAN; THIELE, 2013).

Em um primeiro momento o projeto da antena é feito a partir do tamanho do elemento ativo, isto é o dipolo de meia onda de ganho de 2.5dBi. Para VHF e UHF centrado nas respectivas frequências de 160MHz e 440MHz para  $\lambda$  por meio de:

$$c = \lambda \cdot f \quad (4.1)$$

Além disso, conforme o ganho atribuído para o receptor na confecção do *link budget*, o projeto da antena consistirá em obter um ganho maior ou igual a 9dBi, embora que uma configuração para obter um ganho muito superior pode ser realizável ao custo de complexidade e orçamento. Paralelamente ao requisito de ganho é necessário adequar uma abertura angular durante o apontamento em direção ao objeto em órbita. Se a antena for muito diretiva sua abertura angular será pequena e o apontamento deve ser mais preciso, ao diminuir o ganho a abertura angular aumenta facilitando o apontamento (KOCSIS, 2011).

Para a faixa de UHF, por meio das dimensões disponíveis em (SWALLOW, 1995) no que se referem ao espaçamento e comprimento entre o elemento refletor, elemento ativo e diretores, pode-se realizar um dimensionamento preliminar da antena com auxílio da *toolbox* de antenas do software *MATLAB*.

### 4.2.1 Dimensionamento Preliminar da Antena

A antena para a faixa UHF foi concebida preliminarmente com o espaçamento e o comprimento entre os elementos diretores obtidos das tabela 16.1 e 16.4 respectivamente do livro (SWALLOW, 1995) e auxílio do software *MATLAB*.

Em suma as dimensões utilizadas para a confecção estão na tabela abaixo.

Tabela 4.3 – Parâmetros dos elementos das antenas UHF

*R = Refletor, EA = Elemento Ativo, D = Diretores*

| Elementos | Espaçamento (m) | Elemento  | Comprimento (m) |
|-----------|-----------------|-----------|-----------------|
| R-EA      | 0.068           | R         | 0.36            |
| EA-D1     | 0.16            | EA        | $\lambda/2$     |
| D1-D2     | 0.16            | D1        | 0.295           |
| D2-D3     | 0.20            | D2        | 0.290           |
| D3-D4     | 0.22            | D3        | 0.285           |
| D4-D5     | 0.30            | D4        | 0.280           |
| D5-D6     | 0.32            | D5        | 0.275           |
| D6-D7     | 0.35            | D6        | 0.275           |
| D7-D8     | 0.37            | D7        | 0.270           |
| D8-D9     | 0.39            | D8        | 0.270           |
| D9-D10    | 0.40            | D9        | 0.265           |
| D10-D11   | 0.41            | D10       | 0.265           |
| D11-D12   | 0.40            | D11       | 0.265           |
| D12-D13   | 0.40            | D12       | 0.260           |
| D13-D15   | 0.42            | D12 - D15 | 0.260           |

Adaptado de (SWALLOW, 1995).

Já com relação a abertura angular da antena, conforme a equação que relaciona a diretividade e abertura angular disponível em (ATLANTARF, 2013):

$$D = 10 \log \frac{41253^2}{(HPBW\theta)(HPBW\phi)} \quad (4.2)$$

Esta é calculada especificando a largura de feixe de meia potência, *HPBW*, onde  $HPBW\theta = HPBW\phi$  e a partir da diretividade *D*, assumindo que a abertura seja aproximadamente a mesma no plano de elevação.

#### 4.2.2 437.01 MHz Cross Yagi

A utilização de dois elementos excitadores orientados perpendicularmente e uma alimentação com diferença de fase resultará em uma polarização circular, no passo que a utilização de apenas um elemento confere uma polarização horizontal ou vertical já que a polarização de uma onda eletromagnética é definida com relação à orientação do vetor campo elétrico (MAZDA, 1998).

Figura 4.2 – Arranjo com duas antenas yagi circularmente polarizadas.



Fonte: (HENK, 2010)

Nesse sentido, a definição do tipo de polarização da antena é um parâmetro importante dependendo do contexto operacional podendo haver atenuações no sinal devido a um desalinhamento. Uma maneira de evitar isso em um enlace entre satélite e estação terrena é utilizar a polarização circular que diminui a atenuação devido ao *spin* do transmissor e os efeitos de rotação de Faraday ocorridos na camada ionosférica. Isto é, ao utilizar a polarização linear o ganho da antena diminui na medida em que a polarização da onda recebida varia no passo que esse efeito é mitigado ao utilizar a polarização circular (KOCISIS, 2011).

No contexto do trabalho, a yagi de polarização circular foi modelada e simulada através do software 4NEC2, através do dimensionamento preliminar realizado no MATLAB com as dimensões da tabela 4.3.

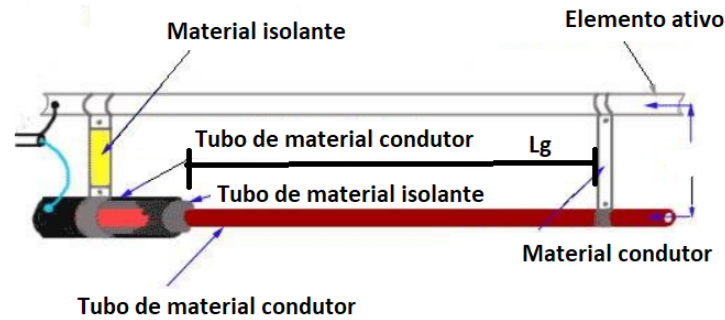
### 4.2.3 Casamento de Impedância - Gamma Match

Paralelamente a isso é importante discutir sobre o casamento de impedância visto que para operar na banda de frequência desejada com alta eficiência deve haver uma combinação de impedância entre a antena e seus componentes adjacentes, o cabo coaxial e o SDR. Nesse processo o *Gamma Match* é utilizado, devido a impedância da yagi cruzada,  $Z_a$  diferir drasticamente da impedância dos demais componentes,  $Z_0$ , haja vista que a impedância do SDR é  $50\Omega$  (WILSON, 2020) que é igual a do cabo coaxial (NEXANS, 2020), além da possibilidade de alimentar o elemento ativo sem alteração na geometria desse através do cabo.

Para que o casamento seja feito de forma correta é necessário dimensionar os

acopladores gamma que efetuam o casamento entre a impedância da yagi cruzada e o sistema de radiofrequência utilizado a partir do cálculo do tamanho do acoplador e da capacitância para os dois elementos ativos em questão da antena.

Figura 4.3 – gamma *match*.



Adaptado de : (CB, 2020)

Conforme a figura 4.7 o acoplamento consiste em determinar a dimensão do acoplador,  $Lg$  e a capacitância do capacitor cilíndrico formado pelo arranjo entre os tubos de material condutor e isolante, na finalidade de igualar a impedância da antena a do cabo coaxial, a partir da impedância de entrada da antena. Para que isso possa ser realizado, o desenvolvimento a respeito do acoplamento em (HEALEY, 1969) que em suma define um tamanho  $LG$  do acoplador por meio do método de tentativa e erro pode ser otimizado por meio do algoritmo LMS adaptativo (CASTRO, 2022).

Na prática, uma linha de alimentação é instalada diretamente no elemento ativo enquanto que a outra é conectada a um capacitor de capacitância ajustável. Isso faz com a corrente elétrica circulante no elemento ativo flua para a linha de transmissão acoplada aumentando a impedância de entrada e diminuindo a corrente do elemento ativo da yagi, analogamente a isso o fluxo das correntes do acoplador e do elemento na mesma direção é obtida através da diminuição da impedância do conjunto,  $Z_0$ , através de uma reatância, visto que as correntes em primeiro momento fluem em sentidos opostos (HEALEY, 1969). Para um ajuste fino e preciso da impedância requerida é necessário utilizar um capacitor com uma reatância necessária para terminar o ajuste.

A partir disso, a impedância do conjunto de entrada é calculada em função dos parâmetros tabelados acima, além da impedância de entrada da antena e a própria impedância da linha de transmissão de comprimento  $Lg$  em m. Seja  $\alpha$  um parâmetro dado em função da geometria do acoplamento, a partir de  $Z_a$  em  $\Omega$ :

$$Z_2 = (1 + \alpha)^2 \frac{Z_a}{2} \quad (4.3)$$

Determinada a impedância do elemento ativo no ponto de toque  $Z_2$ , além da impedância característica formada pela secção entre acoplador e elemento ativo,  $Z_o$ , é feita a transformação da impedância no ponto de toque para a impedância na entrada do acoplador, definido assim impedância  $z_2$ .

$$z_2 = \frac{Z_2}{Z_o} \quad (4.4)$$

Sendo assim, na finalidade de determinar a impedância do conjunto conforme (HEALEY, 1969), exposta por meio do circuito equivalente é necessário calcular o segmento referente a reatância capaz de compensar a reatância de entrada gamma, dado por  $z_g$ .

$$z_g = j \tan\left(\frac{2\pi}{Lg}\right) \quad (4.5)$$

Logo, a impedância de elemento ativo + acoplador é definida através de :

$$Z_{in} = Z_o \left( \frac{1}{\frac{1}{z_2} + \frac{1}{z_g}} \right) \quad (4.6)$$

Após a definição da impedância do acoplamento,  $Z_{in}$ , o algoritmo LMS é utilizado para um refino da impedância bem como o ajuste da capacitância do capacitor empregado.

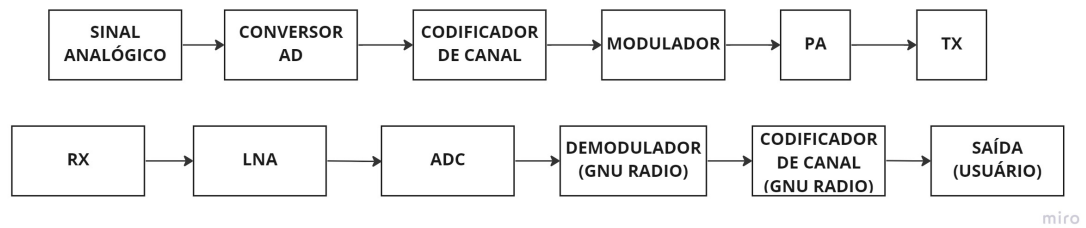
### 4.3 IMPLEMENTAÇÃO DO DEMODULADOR E DECODIFICADOR UTILIZANDO GNU-RADIO

Em um primeiro momento, é importante mencionar que o software GNU Radio é amplamente utilizado em conjunto com um SDR (Software Defined Radio) devido à sua capacidade de integração e processamento de sinais através de blocos funcionais. É possível implementar o SDR diretamente com o software para a versão 3.7.12, através do esquemático presente em (SDRplay, 2020).

Uma vez que o equipamento é integrado ao software, é possível implementar os demais blocos de processamento no programa GNU Radio ou, de forma alternativa, gravar em disco o *stream* de símbolos correspondente ao sinal recebido pelo receptor e trabalhar com esse arquivo no formato .WAV. Essa integração visa a obtenção de *frames* contendo dados telemétricos, após a correta demodulação e decodificação.

Visto esse contexto, para implementar o demodulador e o decodificador é necessário compreender anteriormente a arquitetura do sistema de comunicação digital, para que sejam compreensíveis os processos aqui tratados.

Figura 4.4 – Sistema de comunicação digital.



Adaptado de :(WILSON, 2006)

O transceptor do *cubesat* é responsável majoritariamente pelos processos de digitalização, codificação e modulação do sinal (GomSpace, 2020). Sendo assim, após o sinal ser recebido, o módulo SDR é utilizado para conversão de domínio do sinal, conversão para banda-base, e o sinal em banda-base é inserido no software para realizar a demodulação e decodificação.

Neste sentido, para implementar o demodulador e o decodificador no software GNU Radio, é necessário primeiramente levantar informações sobre a camada de enlace de dados dos *CubeSats* citados na tabela 4.4.

Assim, informações sobre o tipo de modulação, *frame* e código corretor de erros são fundamentais, pois cada palavra binária codificada é transmitida através do processo de modulação.

Tabela 4.4 – Informações referentes à modulação e camada de enlace de dados

| <i>CubeSat</i>          | AlfaCruX                | FLORIPASAT-1          |
|-------------------------|-------------------------|-----------------------|
| Modulação               | GMSK                    | 2-GFSK                |
| Bitrate                 | 4800                    | 1200                  |
| Formato do <i>frame</i> | ASM+Golay(AX-100)       | NGHam                 |
| Código Corretor de Erro | Golay(24,12) + CCSDS RS | RS(223,255)/RS(31,47) |

Fonte: O Autor

### 4.3.1 GNU Radio OOT

Para implementação dos blocos de processamento referentes ao demodulador e decodificador, utiliza-se módulos *Out Of Tree*, ou seja, estes não estão disponíveis no programa original e normalmente são programados utilizando Python, CMake ou C++ (GNU, 2020). O GNU radio disponibiliza o *script grmodtool*, que facilita o desenvolvimento de

novos módulos por meio da automação de alguns processos relacionados ao uso de *templates* e edição de arquivos. A complexidade desse processo deriva dos argumentos de *input* e *output* do bloco de processamento, que podem possuir N entradas e M saídas.

No presente trabalho, para demodulação e decodificação, utiliza-se o módulo programado em python cujo nome é *Gr Satellites* (Daniel Esteves, 2020). Esse módulo implementa uma série de decodificadores para satélites amadores suportando protocolos comuns como AX.25, AX 100, NanoCOM U482C e CCSDS (Daniel Esteves, 2020).

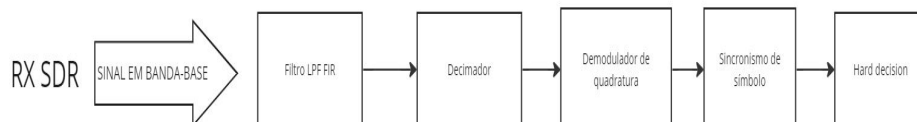
### 4.3.2 Demodulador - GNU Radio

Após o sinal ser codificado no codificador de canal presente no sistema de comunicação digital do satélite é realizada a modulação para transmissão contendo os pacotes telemétricos. Conseqüentemente, no segmento terrestre, ao obter o *stream* de símbolos IQ oriundo desse processo é necessário recuperar os bits transmitidos por meio do demodulador digital projetado, recuperando assim a informação original transmitida (XIONG, 2006).

Nesse sentido, a utilização dos sinais IQ em comunicações digitais permite a decomposição do sinal em duas componentes ortogonais que carregam a informação (LYONS, 2008). Isso permite a implementação de um demodulador em quadratura, já que o tratamento inicial no receptor do sinal, para conversão em banda-base em hardware pode ser feito através de um demodulador de quadratura.

De acordo com esse contexto, fatores de performance tais como capacidade de processamento e complexidade podem diferir de acordo com a metodologia implementado no demodulador (XIONG, 2006).

Figura 4.5 – Fluxograma dos processos referentes a demodulação.



Fonte: O Autor

O presente trabalho implementa um demodulador em quadratura e um demodulador FSK disponibilizados no software GNU Radio aliado de uma metodologia semelhante



a (VEA, 2012) e (LIPNITSKIY, 2010), haja vista que existem inúmeras aplicações do demodulador sendo estas desenvolvidas no próprio software para trabalhar com sinais FSK, GMSK, GFSK (GNU, 2020). Além disso, há implementações conforme em (SHRESTHA, 2016) via hardware utilizando o algoritmo CORDIC para recuperação de fase, é importante ressaltar que a implementação em software difere da implementação em hardware. É importante mencionar que no software ambos demoduladores são operados com o sinal já em banda-base.

A fase do sinal de saída do SDR em banda-base é calculada através do arco-tangente, a recuperação de *clock* pode ser adotada já que a sincronização temporal indica ao receptor quando cada bit de informação foi transmitido (Qasim Chaudhari, 2020). Isso permite ao receptor saber quando cada bit de informação começa e termina recuperando essa informação e sincronizando o receptor com o sinal modulado.

Paralelamente a isso um decimador pode ser adotado para reduzir a taxa de amostragem e facilitar o processo de demodulação (SMITH, 2011). Por último os símbolos são inseridos em um detector, que realiza a *hard decision*. O fator de decimação, utilizado pelo decimador, é a razão entre a taxa de amostragem de entrada e de saída (MALTE, 2016).

$$\frac{r_{Sin}}{dec_{fat}} = r_{Sout} \quad (4.7)$$

Esse fator é importante durante a conversão para banda-base, visando eliminar faixas de frequências irrelevantes. Paralelamente a isso, a questão do número de amostras por símbolo, *sps*, é indispensável para realização do sincronismo de símbolo, caso este seja utilizado.

$$sps = \frac{samprate}{symbolrate} \quad (4.8)$$

Para que o processo de demodulação seja feito corretamente é necessário adequar um filtro gaussiano no demodulador. Diferentemente do demodulador GMSK, o demodulador de quadratura implementado posteriormente, conforme a descrição exposta no manual em (GNU, 2020), não expõe se na arquitetura do demodulador há a implementação de dois filtros passa-baixa em seus respectivos ramos IQ. Devido a isso, uma alternativa é projetar um filtro FIR Gaussiano no software MATLAB e utiliza-lo no GNU Radio. Pode-se adotar também uma metodologia semelhante à (EMBEDDED, 2011) para implementação dos filtros FIR no fluxograma.

Na prática, após o sinal estar em banda-base os filtros de casamento devem maximizar a SNR e então facilitar o processo de detecção através da correlação entre o sinal desconhecido na presença do modelo de um sinal conhecido, ou seja, através da convolução entre o sinal desconhecido com uma versão conjugada (SMITH, 2011).

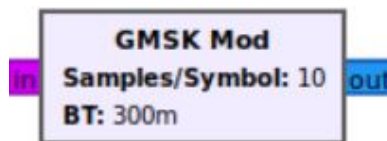
Para isso, o filtro foi projetado através do software MATLAB utilizando a função *gaussdesign*, visando um ganho de -3dB na frequência de corte do filtro que é a largura do sinal em banda-base. O filtro é projetado através das equações 2.9 e 2.10 e é exposto na

seção 5.4.2.

Os principais blocos utilizados no GNURadio, para análise da implementação do demodulador, de forma simulacional, são:

#### 4.3.2.1 GMSK Mod.

Figura 4.6 – GMSK Mod.



Fonte: O Autor

O modulador GMSK possui como entrada um *stream* de bits e saída um sinal GMSK em banda-base (GNU, 2020). A configuração do modulador é feita através do produto BT e amostras por símbolo. Há também parâmetros adicionais que geram informações no terminal do programa a respeito do funcionamento do bloco.

#### 4.3.2.2 GMSK Demod

Figura 4.7 – GMSK Demod.



Fonte: O Autor

O Demodulador GMSK fornecido pelo GNU consiste de um bloco de processamento que tem como entrada um sinal GMSK em banda-base e saída um *stream* de bits. O bloco consiste de um demodulador de quadratura implementado com sincronismo de *clock* e o detector *binary slicer* (GNU, 2020). A configuração do demodulador é feita através da definição do número de amostras por símbolo, largura de banda e tempo de símbolo e constantes que definem a sensibilidade do algoritmo de demodulação (GNU, 2020).

### 4.3.2.3 Demodulador de Quadratura

Figura 4.8 – Demodulador de quadratura.



Fonte: O Autor

O bloco demodulador de quadratura é utilizado para demodular sinais IQ, utilizando a técnica de recuperação de fase.

A entrada é um sinal em banda-base, a saída é um sinal PAM cuja amplitude varia de acordo com a frequência do sinal FSK, recuperada a partir do sinal modulado (GNU, 2020).

Isso é feito calculando o produto do sinal de entrada atrasado e conjugado em uma amostra com o sinal não atrasado e, em seguida, calculando o argumento (também conhecido como ângulo, em radianos) do número complexo resultante, visto na equação 4.18.

O Detector em quadratura calcula o argumento ou ângulo, em radianos do produto entre  $x^*$ , sinal de entrada atrasado e conjugado em uma amostra com o sinal não atrasado  $x[n]$

$$y[n] = \arg(x[n]x^*[n-1]) \quad (4.9)$$

O processo acima é equivalente a utilização de um discriminador polar, ou seja, uma forma de calcular a diferença de fase entre duas amostras complexas que geram uma terceira amostra cuja fase é a diferença de fase entre  $x^*$  e  $x$  (SHIMA, 1995). Logo, o ângulo de fase do sinal contém a informação necessária para performar a demodulação do sinal complexo em banda-base, determinando a frequência instantânea do sinal (SHIMA, 1995)

Seja a senóide complexa com as componentes em fase e quadratura do sinal descrita no tempo, onde  $A$  é amplitude,  $\phi$  é a fase em rad e  $\omega$  a frequência angular em rad/s:

$$x(t) = A\cos(\omega t + \phi) + jA\sin(\omega t + \phi) \quad (4.10)$$

Utilizando a identidade de euler:

$$x(t) = Ae^{j(\omega t + \phi)} \quad (4.11)$$

Discretizando o sinal para  $n$  amostras em uma taxa de amostragem de  $f/f_s$ , onde  $f$  é a frequência absoluta,  $f_s$  a frequência de amostragem e  $\phi$  a fase pertencente entre 0 e  $2\pi$  rad.

$$x[n] = Ae^{j2\pi(\frac{f}{f_s}n + \phi)} \quad (4.12)$$

Para o sinal atrasado e conjugado:

$$x[n - 1] = Ae^{-j2\pi(\frac{f}{f_s}(n-1)+\phi)} \quad (4.13)$$

Como A é constante o *arg* é invariável resultando em:

$$y[n] = A^2 e^{\frac{2jf\pi}{f_s}} \quad (4.14)$$

A equação 4.27 expressa um resultado muito importante, pois, para a modulação GMSK  $f$  é uma sequência de duas frequências que representam o bit 0 e o bit 1.

#### 4.3.2.4 Detector

Figura 4.9 – Binary Slicer.



Fonte: O Autor

O detector é o bloco *binary slicer* que produz o *stream de bits* e na verdade realiza a *hard decision*. O bloco compara a amplitude da amostra do sinal de entrada com zero e se o sinal for maior que zero, o mesmo produzirá um "1" binário, caso contrário, produzirá um "0". Portanto, se a entrada for um valor positivo, será gerado 1 e se for um valor negativo será gerado 0 (GNU, 2020).

- $A > 0 = 1$
- $A < 0 = 0$

#### 4.3.3 Decodificador

Conforme a seção 4.3.2, para que seja possível a implementação do decodificador para qualquer nanossatélite é necessário seguir de perto as recomendações da camada de enlace de dados, que define informações a respeito do protocolo e código corretor de erro. Analogamente a isso, a implementação do decodificador será realizada por meio de um módulo OOT do GNU Radio, chamado de Gr Satellites (Daniel Esteves, 2020). O módulo

provê suporte aos nanossatélites citados no trabalho, conforme a página disponibilizada em (Daniel Esteves, 2020).

Usualmente, para fins acadêmicos de pesquisa espacial a modulação é binária sendo somente dois tipos de símbolos transmitidos, facilitando assim a combinação entre esquemáticos de modulação digital e codificação, quando comparado com modulações do tipo 16QAM e 32QAM (JOHNSON, 2011). Nesse mesmo contexto, a codificação de canal é usualmente realizada por meio dos códigos Reed-Solomon, CRC.

Após a demodulação, o receptor deve recuperar a mensagem original transmitida por meio do decodificador, após recuperação e sincronização dos símbolos, seguido da decisão suave ou abrupta no *de-mapper* do demodulador (JOHNSON, 2011). Por isso, os decodificadores utilizados para os nanossatélites citados no dimensionamento da estação são o AX-100 *deframer* para o AlfaCruz e o NGHAM *deframer*, utilizado no FLORIPASAT-1.

É importante salientar que o decodificador em si não é composto somente com o segmento de codificação de canal, de acordo com o *frame* definido pelo protocolo, é necessário sincronismo de *frame* e checagem de preâmbulo.

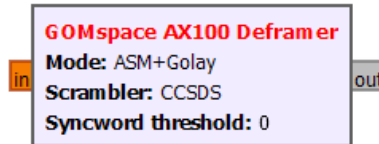
#### 4.3.3.1 AX-100 Deframer

O AX-100 Deframer utiliza os *frames* cujo formato é ASM+Golay e CCSDS Reed-Solomon, usualmente ligados ao protocolo CSP ou CCSDS, utilizados pelo transceptor da GOMspace, NanoCom AX100 (Daniel Esteves, 2020).

Nesse sentido, o ASM+Golay utiliza um cabeçalho codificado com um código de Golay(24,12), que indica o tamanho do *frame* transmitido. Já os dados transmitidos ao usuário referidos como carga ou *payload* são codificados por um CCSDS Reed-Solomon(255,223), que utiliza um embaralhador CCSDS. O decodificador realiza a correção de erro para o cabeçalho e *payload*, resultando como saída um ou mais PDU's (Daniel Esteves, 2020).

Além da correção de erro, o decodificador reorganiza os dados referente ao *frame*, resultando em várias unidades de dados menores chamadas de PDU's, que são posteriormente utilizadas em uma camada de aplicação

Figura 4.10 – AX-100 Deframer.



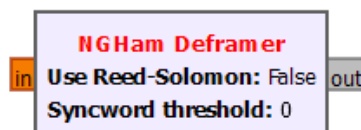
Fonte: O Autor.

A entrada do bloco consiste de um *stream* de símbolos IQ demodulados e suaves contendo *frames* oriundos do transceptor AX100. Por sua vez, a saída consiste PDU's decodificadas.

#### 4.3.3.2 NGHam Deframer

O NGHam Deframer utiliza os *frames* cujo formato é especificado conforme o protocolo NGHam (SKAGMO, 2023). Este realiza a correção de erro utilizando um Reed-Solomon (Daniel Esteves, 2020).

Figura 4.11 – NGHam Deframer



Fonte: O Autor

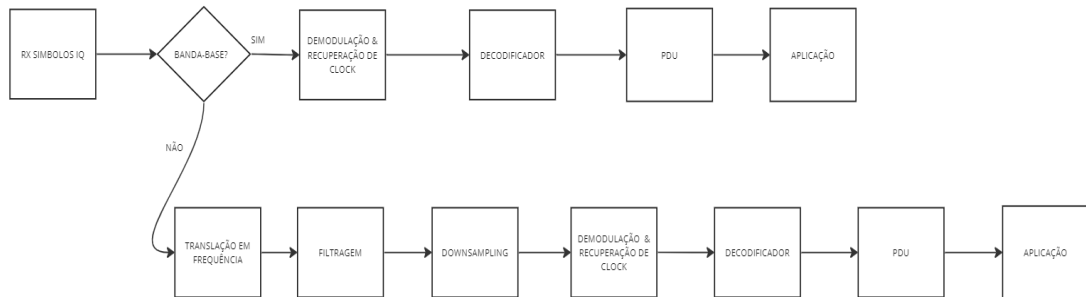
A entrada do bloco consiste de um *stream* de símbolos IQ demodulados e suaves contendo *frames* organizados conforme o protocolo NGHAM. A saída consiste PDU's decodificadas.

#### 4.3.4 Metodologia para processamento de sinais no GNU Radio

No escopo atual do trabalho, os processos gerais para o processamento de um *stream* de símbolos IQ são descritos na figura 4.12, cujo objetivo é realizar várias eta-

pas referente ao processamento de sinais digitais até obter um ou mais *frames* de dados decodificados especificados por um protocolo, do inglês *Protocol Data Unit*.

Figura 4.12 – Metodologia referente ao processamento de sinais.



Fonte: O Autor

De forma indireta, um *stream* de símbolos IQ é recebido, sendo este carregado em uma arquivo .WAV no GNU Radio. Caso o mesmo já se situe em banda-base, pode-se adotar processos referentes a filtragem e re-amostragem, eliminando faixas de frequências não necessárias e diminuindo o número de amostras entregues ao demodulador. Após isso, levando em consideração que o presente trabalho se limita a nanossatélites cujo escopo de modulação é binário, a demodulação é realizada de forma análoga a exposta na secção anterior. Paralelamente a este processo, o sincronismo de símbolo é implementado por meio dos algoritmos de Mueller-Muller ou Gardner.

A decodificação é então realizada, utilizando os decodificadores do módulo OOT, que funcionam com um *stream* de símbolos IQ suaves com dispersão reduzida e *stream* de bits, oriundos da *hard decision* performada pelo detector da figura 4.9. Usualmente, caso a decodificação seja realizada com êxito, os *frames* podem ser visualizados no formato hexadecimal ou PDU's.

Se o *stream* não situa-se em banda-base, é necessário realizar corretamente os processos de conversão para banda-base ou translação em frequência, filtragem e re-amostragem.

## 5 TESTES E RESULTADOS EXPERIMENTAIS

### 5.1 LINK BUDGET

Para o cálculo do *link budget* foram utilizadas as referências (LodeStar Aerospace, 2022) e (Space Lab UFSC, 2022), para que uma verificação da viabilidade do SDR possa ser realizada, a partir da consideração de todos os efeitos citados na secção 2.6

Tabela 5.1 – Parâmetros de cálculo para o *link budget*

|   |                   |
|---|-------------------|
| Potência transmitida                          | 1w = 30dBm        |
| Perca devido cintilação = $Scin_{loss}$       | 5dB               |
| Largura de banda - Bw                         | 20 kHz            |
| Perca devido polarização - $P_{loss}$         | 3dB               |
| Figura de ruído - Nf                          | 5dB               |
| Perca devido cabeamento - $Cable_{loss}$      | 1dB               |
| SNR para BER = 1E-5                           | 15dB              |
| Frequência do link                            | 0.16GHz; 0.437GHz |
| Perca devido impedância - $Rx_{imploss}$      | 50Ω               |
| Ganho antenna Tx do cubesat (pior caso) - GTx | 0 dBi             |
| Ganho da antenna receptora - Yagi             | 9 dBi             |
| Perca de caminho - LFS                        | -148.172 dB       |

Fonte: O Autor

Em um primeiro momento a sensibilidade que quantifica o menor nível de sinal onde se pode extrair informações é avaliada:

$$Sensitivity : -174 + 10\log\frac{Bw}{Hz} + Nf + SNR \quad (5.1)$$

De forma análoga a equação 2.11, é medida a potência do sinal nos terminais do receptor:

$$Prx = Ptx + Lfs + G_{yagi} + LFS + Gtx - (Pol_{loss} + Cable_{loss} + Scin_{loss}) \quad (5.2)$$

Para verificar a tensão nos terminais do SDR, é necessário converter o resultado de dBm para mW:

$$Prx_{mW} = 1mW \cdot 10^{\frac{Prx}{10}} \quad (5.3)$$



Por último, é medida a voltagem de entrada nos terminais do SDR:

$$R_{x_{input}} = \sqrt{P_{rx_{mW}} \dot{R}_{x_{inloss}}} \quad (5.4)$$

O que leva à confecção da tabela de resultados 5.2:

Tabela 5.2 – *Link budget*

|                             |                    |
|-----------------------------|--------------------|
| <i>Sensitivity</i>          | -110.99 dBm        |
| <i>Prx</i>                  | -118.172 dBm       |
| $R_{x_{input}}(\text{UHF})$ | 0.27 $\mu\text{V}$ |
| $R_{x_{input}}(\text{VHF})$ | 0.75 $\mu\text{V}$ |

Fonte: O Autor.

O resultado obtido pelo equacionamento acima deve ser comparado com o manual, (WILSON, 2020), tal que para a correta operação do sistema as condições devem ser conferidas:

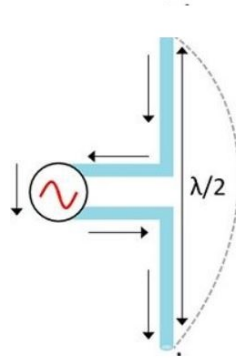
- VHF  $R_{x_{input}} > 0.70\mu\text{V}$
- UHF  $R_{x_{input}} > 0.16\mu\text{V}$

Através da implementação do equacionamento exposto em um *software* verifica-se que o SDR tem condições de operar o sinal emitido, para a faixa UHF a tensão nos terminais é de 0.27  $\mu\text{V}$  no passo que para VHF é 0.75 $\mu\text{V}$ .

## 5.2 DIMENSIONAMENTO PRELIMINAR E SIMULAÇÃO DA ANTENA

O primeiro passo para o dimensionamento preliminar da antena foi o cálculo da dimensão do dipolo de meia onda, correspondente a equação 4.1 da secção 4.2.

Figura 5.1 – Dipolo de meia onda.



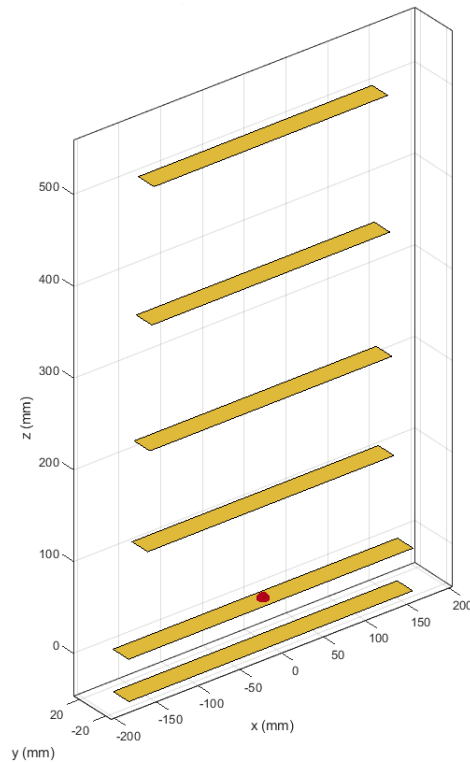
(CISCO, 2019)

O cálculo resulta em:

- $\frac{\lambda}{2}(\text{UHF}) = 0.34\text{m}$

A partir disso, um dimensionamento foi realizado baseado nas dimensões disponíveis em (SWALLOW, 1995), pertencentes a tabela 4.3. O objetivo principal a ser atendido de projeto da antena é o seu ganho, orientado pelo *Link Budget*. Para isso, a primeira etapa realizada na *toolbox* de antenas no software MATLAB, foi inserir as dimensões disponíveis da tabela 4.3 e verificar ganho, e através da equação 4.2 a abertura angular de acordo com a variação da geometria da antena na configuração de polarização horizontal, para constatar o número de elementos diretores necessários e validar a tabela de projeto presente em (SWALLOW, 1995).

Figura 5.2 – Yagi UHF para 4 elementos modelada no MATLAB.



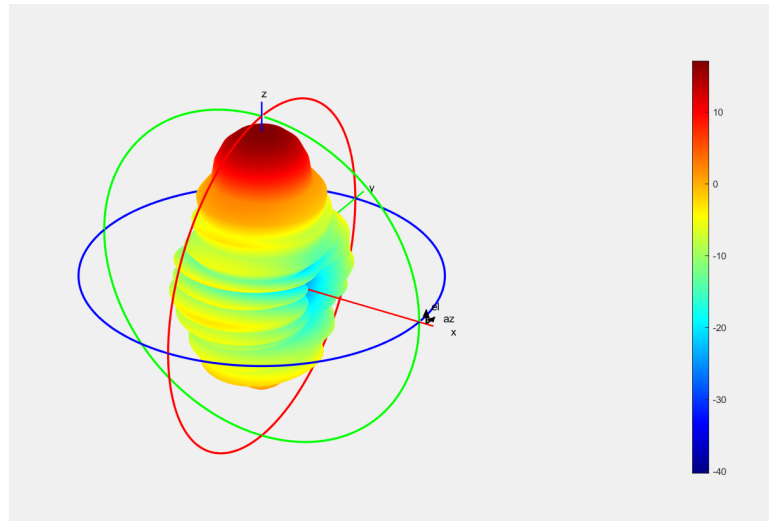
Yagi UHF para 4 elementos modelada no MATLAB.

Fonte: O autor

De início, o dimensionamento resultou em um ganho de 17.1 dBi para 15 elementos diretores que atende o *link budget*. Especificando a largura de feixe de meia potência, *HPBW*, onde  $HPBW_{\theta} = HPBW_{\phi}$  e a partir da diretividade  $D$ , assumindo que a abertura seja aproximadamente a mesma no plano de elevação e plano de azimute a abertura angular é de  $28.1^{\circ}$ .

Já o diagrama de radiação da antena orientada no eixo z com máxima elevação:

Figura 5.3 – Diagrama de radiação da antena UHF de 15 elementos - Polarização horizontal

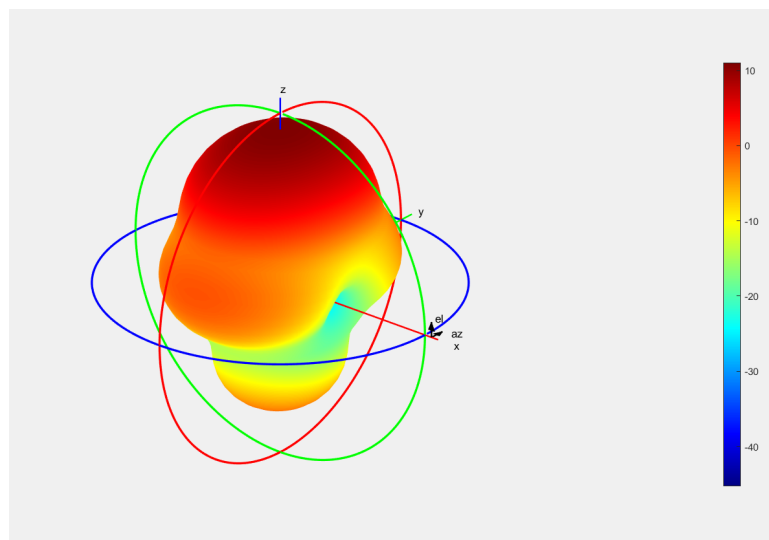


Fonte : autor

De maneira alternativa pode-se utilizar um número menores de elementos diretores, o que provê um menor ganho e em consequência conforme a equação 4.2 uma maior abertura angular, facilitando o apontamento. Utilizando 4 elementos diretores o ganho é de 10.9 dBi e a abertura é de  $57.9^\circ$  para uma antena dimensionada de maneira análoga a de 15 elementos.

Para o diagrama de radiação orientado no eixo Z com máxima elevação:

Figura 5.4 – Diagrama de radiação da antena UHF de 4 elementos - Polarização Horizontal.



Fonte : autor

Portanto, a relação entre número de elementos diretores, ganho e abertura angular é de suma importância no projeto de uma antena receptora, tal que a mudança de um parâmetro influencia de modo positivo ou negativo dependendo do objetivo específico de operação, seja este um apontamento com uma maior abertura ou maior ganho. Essas relações são visíveis nas figuras 5.4 e 5.3 e expostas na tabela 5.3.

Tabela 5.3 – Características das antenas UHF

| Número de diretores | Ganho   | Abertura angular | Dimensões (x,z) |
|---------------------|---------|------------------|-----------------|
| 15                  | 17.1dBi | 28.1 °           | (0.4,3.5) m     |
| 4                   | 10.9dBi | 57.9 °           | (0.4,0.5) m     |

Fonte : O autor

Desse modo, o contexto acima descreve uma etapa preliminar de dimensionamento da antena utilizando apenas um dipolo simples e uma única direção de polarização visando uma confecção preliminar nas frequências utilizadas, a configuração inicial exposta será utilizada como base na confecção de uma antena de polarização circular utilizada para operar na frequência do satélite escolhido para *downlink*.

Em um primeiro momento, a validação simulacional da antena por meio do software 4NEC2 é fundamental para verificar o requisito de 9dBi atendidos pelo orçamento de *link*, bem como definir o projeto do *gamma match*. Nesse contexto, o dimensionamento preliminar realizado com base em (SWALLOW, 1995) é fundamental, de forma que os valores observados nesse foram inseridos no software 4NEC2 em formato de coordenadas, além da própria dimensão dos elementos ativos e diretores conforme a figura abaixo:

Figura 5.5 – Coordenadas no 4NEC2 para o projeto da yagi cruzada.

| Geometry (Scaling=Meters) |      |     |      |      |          |          |      |         |         |        |  |  |              |
|---------------------------|------|-----|------|------|----------|----------|------|---------|---------|--------|--|--|--------------|
| Nr                        | Type | Tag | Segs | X1   | Y1       | Z1       | X2   | Y2      | Z2      | Radius |  |  | comment      |
| 1                         | Wire | 1   | 9    | 0    | -0.36/2  | 0        | 0    | 0.36/2  | 0       | 0.01   |  |  | Refletor.    |
| 2                         | Wire | 2   | 9    | 0.18 | -0.34/2  | 0        | 0.18 | 0.34/2  | 0       | 0.01   |  |  | EA(lambda/2) |
| 3                         | Wire | 3   | 9    | 0.2  | 0        | -0.34/2  | 0.2  | 0       | 0.34/2  | 0.01   |  |  | EA(lambda/2) |
| 4                         | Wire | 4   | 9    | 0.25 | -0.295/2 | 0        | 0.25 | 0.295/2 | 0       | 0.01   |  |  | D1           |
| 5                         | Wire | 5   | 9    | 0.26 | 0        | -0.295/2 | 0.26 | 0       | 0.295/2 | 0.01   |  |  |              |
| 6                         | Wire | 6   | 9    | 0.34 | -0.290/2 | 0        | 0.34 | 0.290/2 | 0       | 0.01   |  |  | D2           |
| 7                         | Wire | 7   | 9    | 0.35 | 0        | -0.290/2 | 0.35 | 0       | 0.290/2 | 0.01   |  |  |              |
| 8                         | Wire | 8   | 9    | 0.48 | -0.285/2 | 0        | 0.48 | 0.285/2 | 0       | 0.01   |  |  | D3           |
| 9                         | Wire | 9   | 9    | 0.49 | 0        | -0.285/2 | 0.49 | 0       | 0.285/2 | 0.01   |  |  |              |
| 10                        | Wire | 10  | 9    | 0.6  | -0.280/2 | 0        | 0.6  | 0.280/2 | 0       | 0.01   |  |  | D4           |
| 11                        | Wire | 11  | 9    | 0.61 | 0        | -0.280/2 | 0.61 | 0       | 0.280/2 | 0.01   |  |  |              |
| 12                        | Wire | 12  | 9    | 0.7  | -0.275/2 | 0        | 0.7  | 0.275/2 | 0       | 0.01   |  |  | D5           |
| 13                        | Wire | 13  | 9    | 0.71 | 0        | -0.275/2 | 0.71 | 0       | 0.275/2 | 0.01   |  |  |              |
| 14                        | Wire | 14  | 9    | 0.9  | -0.265/2 | 0        | 0.9  | 0.265/2 | 0       | 0.01   |  |  | D6           |
| 15                        | Wire | 15  | 9    | 0.91 | 0        | -0.265/2 | 0.91 | 0       | 0.265/2 | 0.01   |  |  |              |
| 16                        | Wire | 16  | 9    | 1.2  | -0.260/2 | 0        | 1.2  | 0.260/2 | 0       | 0.01   |  |  | D7           |
| 17                        | Wire | 17  | 9    | 1.21 | 0        | -0.260/2 | 1.21 | 0       | 0.260/2 | 0.01   |  |  |              |
| 18                        | Wire | 18  | 9    | 1.4  | -0.260/2 | 0        | 1.4  | 0.260/2 | 0       | 0.01   |  |  | D8           |
| 19                        | Wire | 19  | 9    | 1.41 | 0        | -0.260/2 | 1.41 | 0       | 0.260/2 | 0.01   |  |  |              |
| 20                        | Wire | 20  | 9    | 1.6  | -0.255/2 | 0        | 1.6  | 0.255/2 | 0       | 0.01   |  |  | D9           |
| 21                        | Wire | 21  | 9    | 1.61 | 0        | -0.255/2 | 1.61 | 0       | 0.255/2 | 0.01   |  |  |              |

Coordenadas no 4NEC2 para o projeto da yagi cruzada.

Fonte: O autor

Após modelar e incluir no 4NEC2 a alimentação do elemento que nesse caso é o elemento ativo de comprimento  $\lambda/2$ , para a faixa de frequência de 437.01 MHz é possível visualizar dados a respeito do SWR, impedância, ganho e diagrama de radiação.

Tabela 5.4 – SWR, impedância e ganho da antena

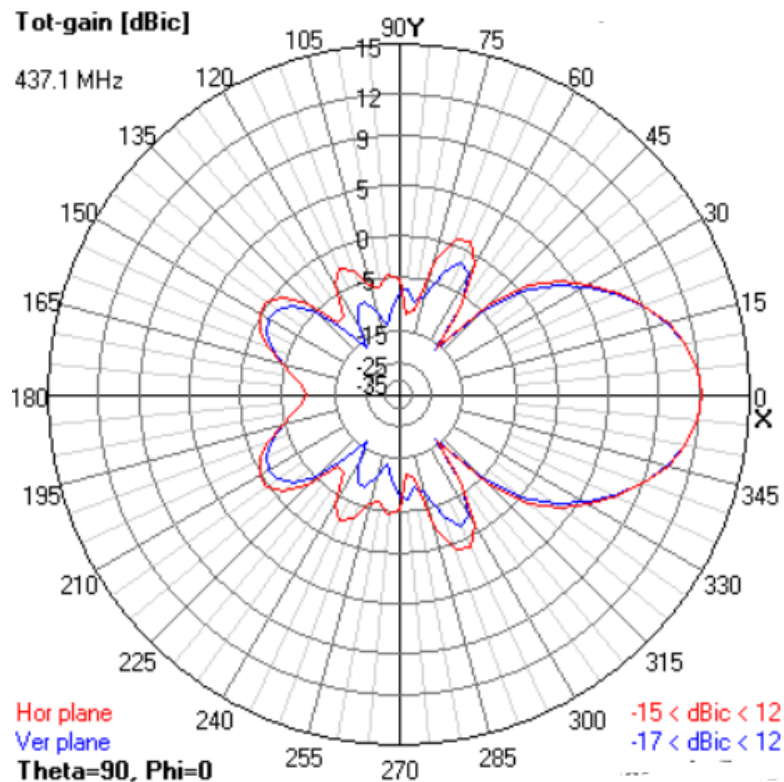
|              |              |
|--------------|--------------|
| SWR50        | 1.9          |
| Impedância R | 18.4 + 45.4j |
| Ganho Total  | 12 dBi       |

Fonte : O autor

A partir da razão de onda estacionária verifica-se que a antena necessita de casamento de impedância, já que  $SWR > 1$  para a faixa de frequência, refletindo assim o sinal e diferindo assim da impedância do cabo coaxial utilizado, Ademais, o ganho atingido é viável com relação ao orçamento de link.

Já o diagrama de radiação nos planos horizontal e vertical, onde o plano horizontal é caracterizado em vermelho, sendo este orientado por meio do ângulo  $\phi$ , e azul o plano vertical, sendo caracterizado pelo ângulo de elevação  $\theta$ , situa-se da seguinte maneira:

Figura 5.6 – Diagrama de radiação para as polarizações horizontal e vertical.

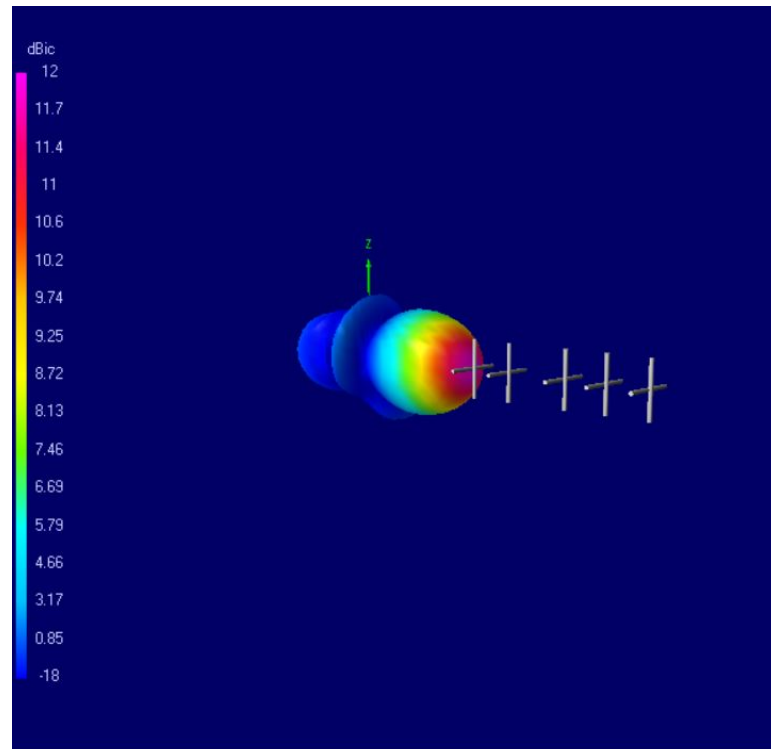


Fonte : O autor

Ainda, é possível visualizar a modelagem da antena no espaço, bem como seu

diagrama de radiação:

Figura 5.7 – Visualização da antenna no 4NEC2



Fonte: O autor.

Haja vista que a impedância medida na simulação difere da impedância ideal de  $50 \Omega$ , dos demais componentes utilizados, o *Gamma Match* será utilizado para casamento de impedância entre antenna, cabo e SDR. O *Gamma Match* é um casador de impedância que permite ajustar a impedância da antenna para se adequar ao sistema de recepção. A partir dos dados de entrada referentes a antenna, para que o acoplamento seja feito nos dois elementos ativos mediu-se a impedância das antenas separadamente no 4NEC2, haja vista que a polarização circular é obtida na configuração de cruzeta.

Tabela 5.5 – Parâmetros para definição do *gamma match*

|   |              |
|---|--------------|
| $\lambda$                                 | 0.686m       |
| Diâmetro linha de transmissão (a)         | 1 cm         |
| Diâmetro do acoplador (b)                 | 1 cm         |
| Impedância de entrada da yagi (Hor) $Z_a$ | $18.4+45.4j$ |
| Impedância de entrada da yagi (Ver) $Z_a$ | $19.4+31.8j$ |

Fonte: O Autor

Nesse sentido, a impedância do conjunto de entrada é calculada em função dos

parâmetros da tabela 5.5 pela equação 4.6, utilizando o algoritmo LMS adaptativo em (CASTRO, 2022).

Conseqüentemente, são obtidos o tamanho do acoplador e o valor da capacitância correspondente para alcançar a impedância final desejada, que deve ser próxima de  $50 \Omega$ , para ambos dipolos:

Tabela 5.6 – *Gamma Match*

| Gamma Match - Elemento ativo vertical | Gamma Match - Elemento ativo horizontal |
|---------------------------------------|---|
| $L_g = 0.09m$                         | $L_g = 0.08m$                           |
| $C = 3.548 \times 10^{-12} F$         | $C = 5.34 \times 10^{-12} F$            |
| $z_0 = 49.968 \Omega$                 | $z_0 = 49.968 \Omega$                   |

Fonte: O Autor.

Logo, a partir dos resultados acima é possível construir o *Gamma Match* em campo.

Os resultados obtidos por meio do dimensionamento no MATLAB e simulação no 4NEC2 mencionados acima foram de extrema importância para o desenvolvimento e construção da antena. Esses resultados forneceram informações cruciais sobre as dimensões ideais da antena.

Além disso, os dados de impedância dos elementos ativos foram utilizados para a elaboração do casador de impedância, que desempenha um papel fundamental na adaptação da impedância da antena ao sistema de recepção. Com base nessas informações e resultados, foi possível construir uma antena com características referentes a aplicação desejada. Portanto, os resultados obtidos por meio do dimensionamento e simulação foram indispensáveis no trabalho, pois forneceram as diretrizes necessárias para a construção de uma antena eficiente e funcional.

### 5.3 CONSTRUÇÃO DA ANTENA

Após as etapas de dimensionamento preliminar e simulação da antena em software iniciou-se a construção física da antena. Para confecção, os materiais utilizados foram:

- Cano de PVC para o *boom*.
- Tubos de alumínio para os elementos diretores, elementos ativos, refletor e casador de impedância.
- Cabo Coaxial de  $75 \Omega$ .
- Cabo Coaxial de  $50 \Omega$ .



- Durepoxi para fixação dos elementos.
- Abraçadeiras, arruelas e parafusos para construção do *gamma-match*.
- Estanhador

De início foram marcados no *boom* as posições dos dipolos e elementos parasitas para que posteriormente a fixação dos elementos seja feita corretamente com durepoxi. Após isso, foram feitos cortes nos tubos de alumínio nas respectivas dimensões de projeto conforme a figura 5.5.

Figura 5.8 – Antena em construção.



Fonte: O autor.

Os dipolos foram fixados com durepoxi no *boom* por último. Para construção do casador de impedância utilizou-se os resultados exibidos na tabela 5.6. Foram utilizados abraçadeiras, arruelas, tubos de alumínio de diferentes diâmetros e materiais dielétricos como plástico e borracha.

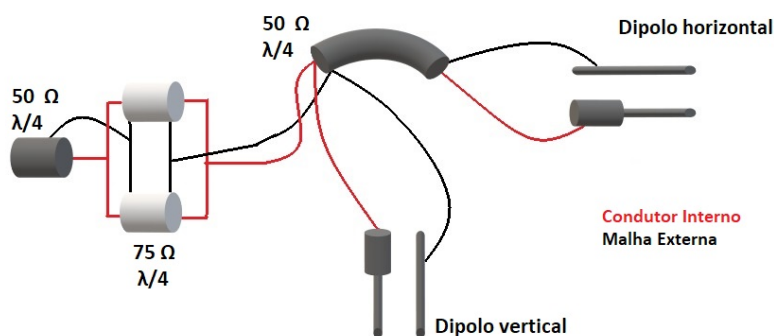
Figura 5.9 – Casador Construído.



Fonte: O autor.

Por último a alimentação da antena para obter a polarização circular RHCP foi realizada conforme (HENK, 2010), utilizando os cabos coaxiais disponíveis e um estanhador.

Figura 5.10 – Alimentação da Antena



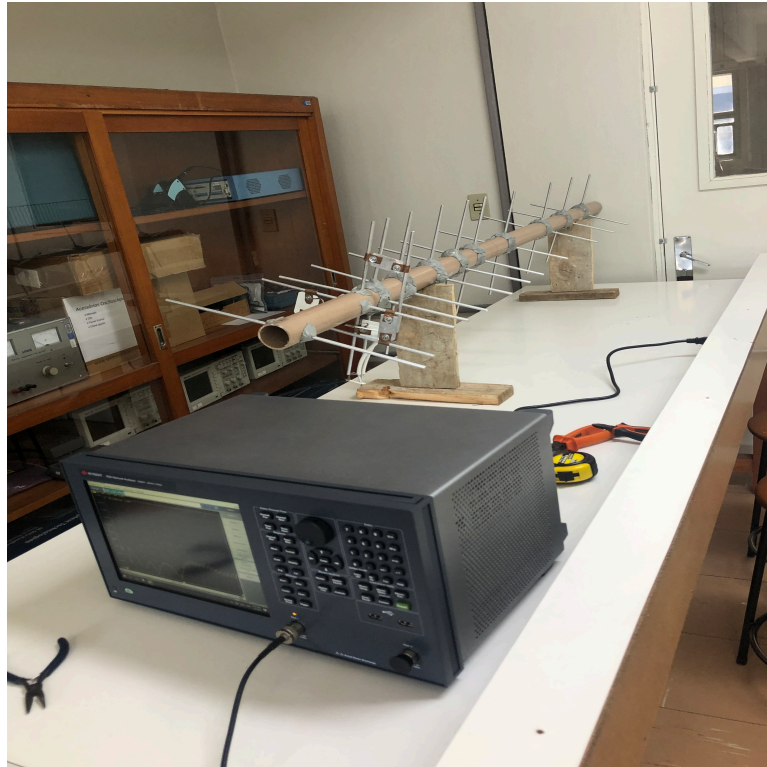
Adaptado de: (HENK, 2010)

No esquemático da figura 5.10 o dipolo horizontal é conectado ao dipolo vertical por uma conexão de  $\lambda/4$  pelo cabo de  $50\Omega$ . O fio condutor do cabo é conectado ao capacitor cilíndrico enquanto a malha externa é conectada aos dipolos. Dois cabos de  $75\Omega$  são

conectados em paralelo e ligados ao casador de impedância, por último um cabo de  $50\Omega$  é conectado ao arranjo paralelo.

### 5.3.1 Resultados Preliminares da Antena

Figura 5.11 – Bancada de teste.



Fonte: O autor

Após a confecção da antena verificou-se suas características através do analisador de rede vetorial Keysight E5061B. O cabo coaxial foi conectado ao aparelho e mediu-se SWR, Impedância e coeficiente de reflexão  $s_{11}$ .

Chegou-se nos seguintes valores de SWR e impedância:

Tabela 5.7 – Resultados obtidos no analisador de rede vetorial.

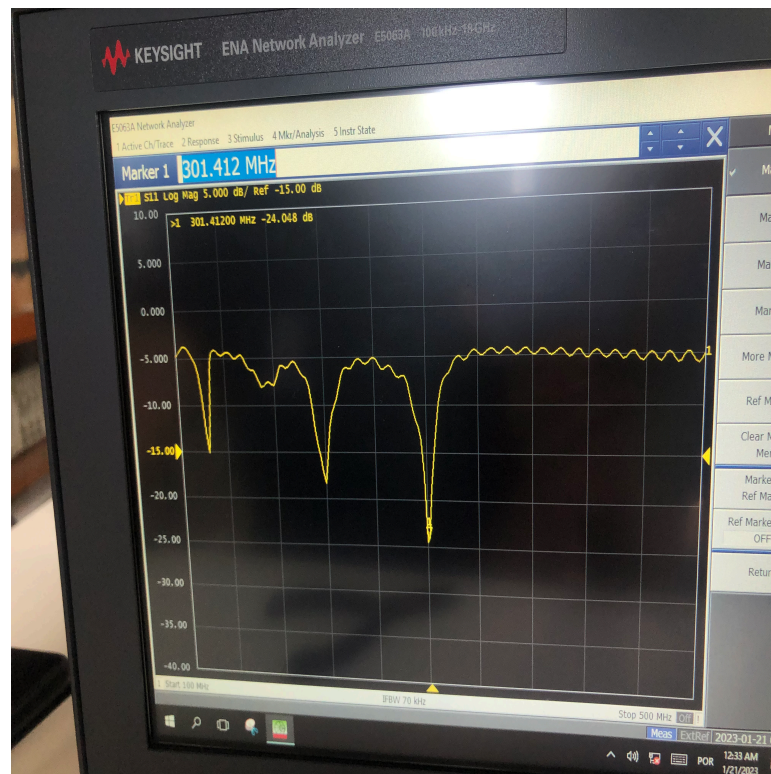
| SWR  | Impedância                | Frequência |
|------|---------------------------|------------|
| 1.36 | $49.760 + j11.078 \Omega$ | 300 MHz    |
| 3.5  | $39.575 - j55 \Omega$     | 437 MHz    |

Fonte : O autor.

Conforme a tabela 5.7 a antena acabou apresentando resultados não favoráveis quando as medições foram feitas na frequência de projeto. Um SWR de 3.5 indica que

mais de 20% da potência está sendo refletida conforme a tabela 2.2. Ao verificar a impedância em 437MHz verifica-se que o sistema não está em ressonância, haja vista que sua impedância não é puramente resistiva. Além disso, comparando os resultados da frequência de projeto com a frequência de 300MHz a antena apresentou resultados mais favoráveis no que diz respeito a impedância e SWR.

Figura 5.12 – Parâmetro S11.



Fonte: O autor

O parâmetro S11, que expressa a quantidade de potência que está sendo refletida pela antena, é similar ao SWR, pois quanto menor o coeficiente de reflexão, menor o SWR. Os resultados obtidos para o coeficiente de reflexão não são satisfatórios para a frequência de projeto, pois, assim como o SWR, o mínimo do S11 está na faixa de 300 MHz.

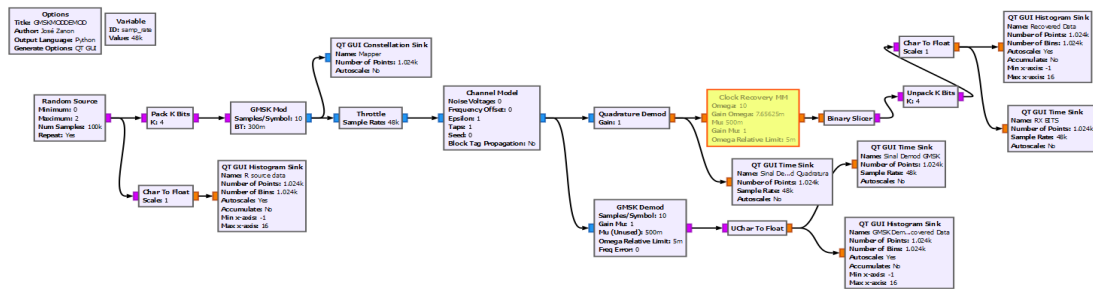
Portanto, os resultados obtidos para a frequência de projeto não são apropriados, embora não haja medição da diretividade e elaboração do diagrama de radiação da antena. Isso deve-se, em parte, ao segmento de alimentação que foi confeccionado de forma preliminar. O principal aspecto que deteriorou as medições foi a fixação dos elementos ativos e passivos com massa epóxi, que possui uma permissividade elétrica de aproximadamente 3.5 e, conseqüentemente, contribuiu para a mudança da frequência de ressonância da antena (THOMAS, 2008). Relacionado também à utilização dessa resina está a espessura do revestimento utilizada, já que a permissividade também aumenta com a espessura, diminuindo assim a capacidade eletromagnética da resina (BALDAN, 2022).

## 5.4 DEMODULADOR E DECODIFICADOR - GNURADIO

### 5.4.1 Demodulação - GNURadio

#### 5.4.1.1 Fluxograma dos Demoduladores

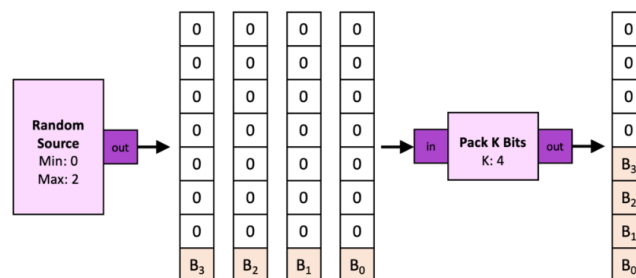
Figura 5.13 – Fluxograma desenvolvido



Fonte : O autor

Em um primeiro momento pode-se testar os demoduladores incluídos no software GNU Radio, através da inserção dos blocos de processamento. Assim sendo, a primeira etapa é a utilização do bloco *random source* para criação de um número de amostras de bits de maneira semelhante a um codificador de fonte que posteriormente são utilizados para a modulação, o bloco gera bits com um valor mínimo e um valor máximo-1. Nesse caso geramos os binários 0 e 1 que posteriormente são inseridos no bloco *pack k bits* que insere os bits em um pacote para representação de valores binários maiores ou palavras binárias. Para  $k=4$ , 4 bits fornecem de  $2^0 = 1$  à  $2^4 - 1 = 15$  palavras binárias. Além disso, o bloco é frequentemente utilizado para testar moduladores no escopo do programa.

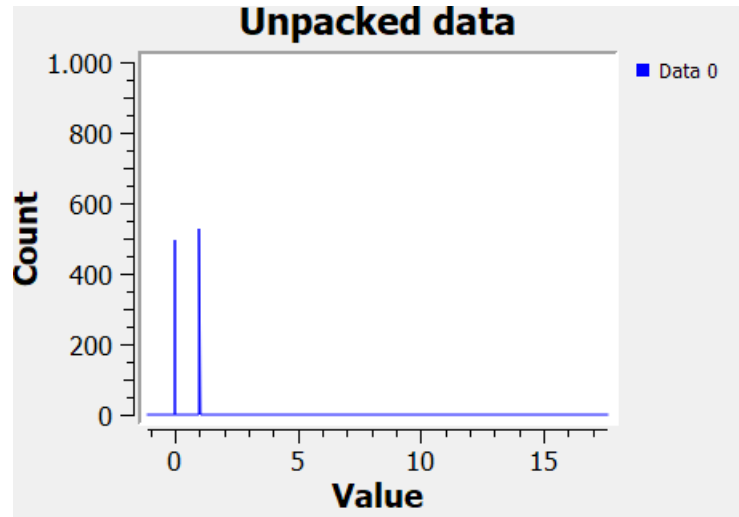
Figura 5.14 – Blocos *random source* e *pack k bits*.



Fonte : (NOGUEIRA, 2015)

Para  $k=4$  uma palavra binária com 4 bits será inserida no pacote, o *random source* fornece os bits B0 à B3 inseridos e a saída do bloco é o bit [0000b3b2b1b0]. Caso  $b_0=0, b_1=1, b_2=0$  e  $b_3=1$  o bit gerado é 00001010 e seu valor decimal é 10.

Figura 5.15 – Bits gerados pelo *random source*.

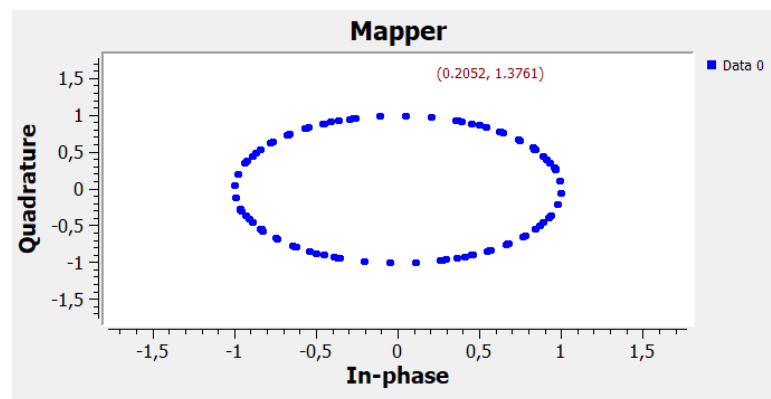


Fonte : O autor

O *stream* de bits é então fornecido ao modulador GMSK, o modulador é configurado com  $BT = 0.5$  e 10 amostras por símbolo.

O impacto do número de amostras utilizadas para representar cada símbolo IQ pode ser verificado no mapa IQ ou *mapper*. O Mapa referente ao sinal modulado é obtido através do display *Constellation Sink* utilizado para análise de constelação de sinais, e demonstra como as palavras binárias estão mapeadas no plano IQ.

Figura 5.16 – *Mapper* GMSK.



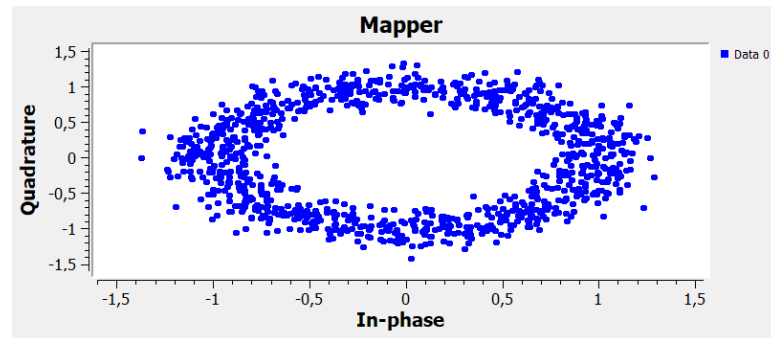
Mapper GMSK

Fonte: O autor

No GNU Radio é possível modelar um canal de comunicação, através da inserção do bloco *channel model* que insere ruído, desvio de frequência, diferença de clock entre

Rx e Tx e cenário *multipath* na simulação.

Figura 5.17 – Mapper com ruído AWGN



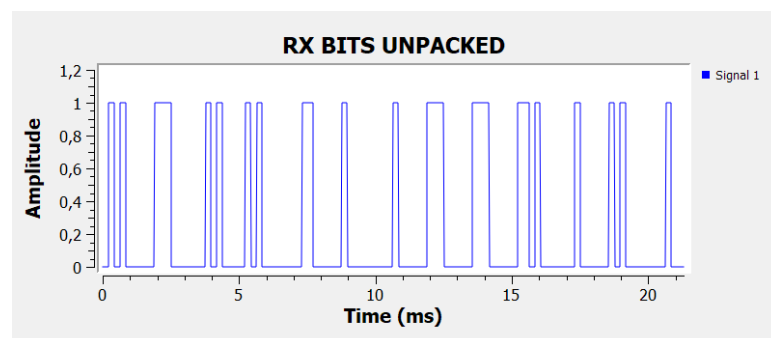
Fonte: O autor

Conforme a descrição dos moduladores, o sinal de entrada é em banda-base, logo a etapa de conversão para banda-base é eliminada da simulação. Caso seja necessária fazer a conversão para banda-base ou *down-conversion* pode-se usar o filtro *frequency xlating FIR filter*. O filtro pode também reduzir a taxa de amostragem do sinal através de um decimador (GNU, 2020).

O sinal de saída do modulador é fornecido aos demoduladores de quadratura e GMSK. O bloco *unpack k bytes* é utilizado para remover os bits do pacote, sendo este análogo ao bloco *pack k bytes*.

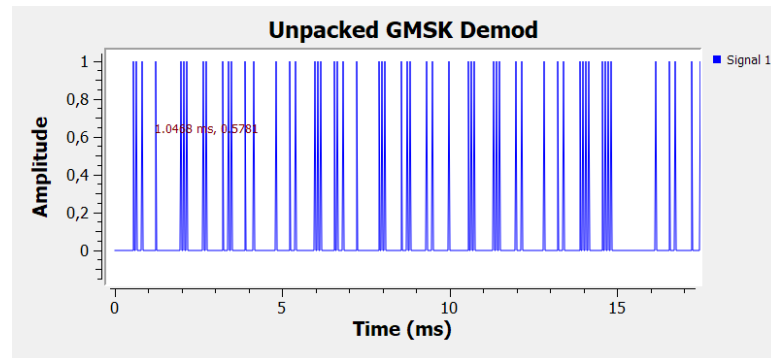
Para o demodulador de quadratura, o detector que neste caso é o *binary slicer* produz o *stream de bits*, sendo esta a *hard decision* (JOHNSON, 2011). A saída do demodulador GMSK difere graficamente do demodulador de quadratura proposto. Segundo (VEA, 2012), o demodulador GMSK inclui um algoritmo de sincronização Mueller e Muller, dessa forma, o sincronismo é empregado para determinar o instante inicial e final de cada símbolo transmitido, ainda que no contexto da simulação há apenas um *clock*, o do computador.

Figura 5.18 – Sinal de linha do demodulador de quadratura.



Fonte : O autor

Figura 5.19 – Sinal de linha do demodulador GMSK.



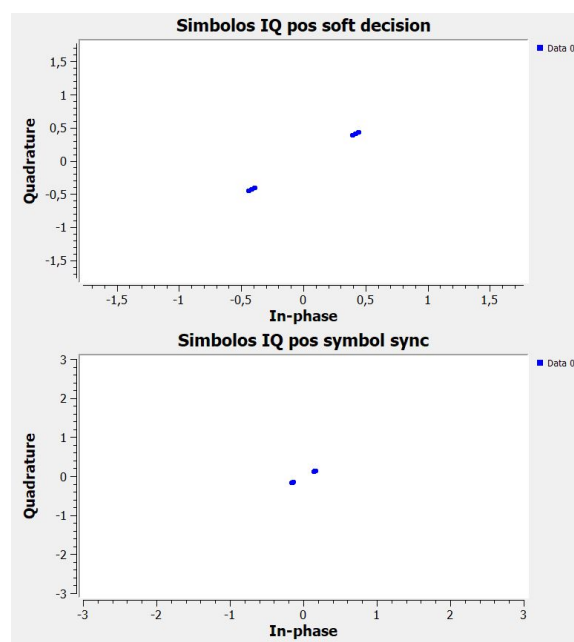
Fonte: O autor

É possível utilizar o bloco *Constellation Decoder*, que na verdade é um de-mapper utilizado para modulações de mais alta ordem, como a 16QAM. O bloco realiza a *Soft Decision*, enquadrando os símbolos conforme o de-mapper explicitado no bloco *Constellation Object* (GNU, 2020).

- $0 = -1-1j$
- $1 = 1+1j$

Neste contexto, a dispersão dos símbolos recebidos pode ser observada através do *Constellation Sink* (GNU, 2020). O contexto ideal é reduzir a dispersão dos símbolos recebidos em torno dos símbolos de referência, para que o respectivo símbolo possa ser corretamente demodulado.

Figura 5.20 – Dispersão reduzida pós sincronismo de símbolo



Fonte: O autor



Nota-se na figura 5.20 que os símbolos estão com uma menor dispersão, portanto tendendo as regiões especificadas de enquadramento.

O fluxograma dos demoduladores é uma forma prática de testar inicialmente todos os demoduladores e blocos relacionados ao processo de demodulação presentes no software, visualizando de forma gráfica os dados de entrada e de saída dos respectivos blocos utilizados, além de proporcionar uma compreensão maior sobre o esquemático de modulação GMSK.

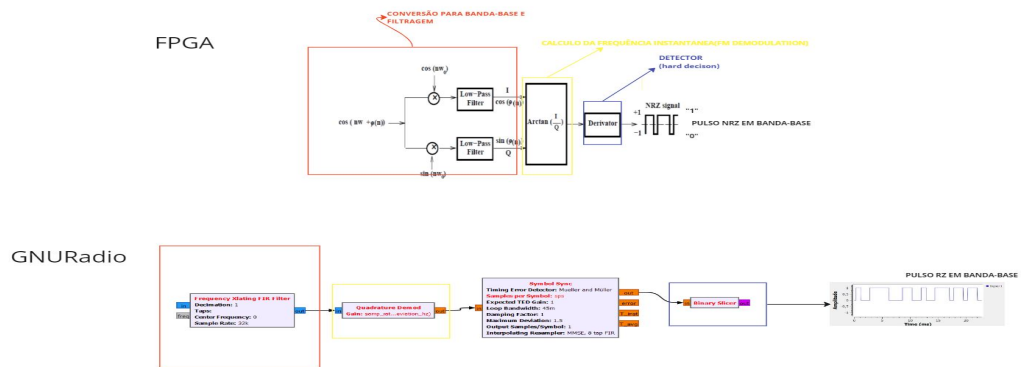
O *mapper* na figura 5.16, fornece informações fundamentais a respeito dos símbolos IQ modulados, não só nesse esquemático de modulação, mas na FSK, GMSK e GFSK a amplitude do sinal é constante e como a informação é inserida na frequência, esta varia de acordo com os bits 0 e 1. Por isso, uma vez que a amplitude é constante, esta corresponde ao raio do círculo formado pelos símbolos. A variação da fase no tempo ou derivada da fase, é a frequência do sinal, isso faz com que os símbolos sejam enquadrados nas suas respectivas regiões, de forma suave, pois as transições de fase são suaves. A dispersão acentuada dos símbolos, que usualmente acomete vários sistemas de comunicação, também é visualizada conforme o modelo do canal AWGN.

Além disso, outro aspecto muito importante observado é o funcionamento do demodulador de quadratura e o detector, *binary slicer*, que são futuramente empregados no demodulador proposto. Observou-se o funcionamento de cada bloco citado no fluxograma, na finalidade de expor a funcionalidade do software, e também auxiliar na elaboração de um demodulador que trabalhe com um *stream* de símbolos IQ oriundos do Tx de um *CubeSat*.

#### 5.4.2 Demodulador Proposto

O demodulador proposto se limita ao escopo de esquemáticos de modulação binários, como FSK, GMSK, GFSK, amplamente utilizado por *CubeSats*. Nesse contexto, a implementação do demodulador é realizada através dos blocos de processamento de sinais presentes no GNURadio.

Figura 5.21 – Demodulador GMSK implementado em uma FPGA vs demodulador proposto no GNURadio pelo autor.



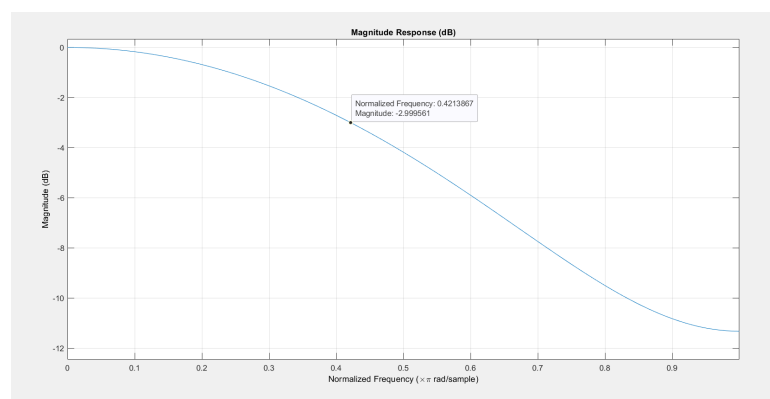
Fonte: O autor

No contexto do demodulador proposto, o filtro polifásico *frequency xlating FIR filter* realiza os processos de conversão para banda-base, equivalente aos *mixers* de quadratura da figura 2.9, e também performa a filtragem e re-amostragem.

O filtro FIR citado na secção 4.3.2 é projetado com auxílio da tabela 5.1, sabendo que a largura do sinal em banda-base corresponde a metade da largura do sinal que trafega no canal:

- $B = 10\text{KHz}$
- $T = 1/S_r = 1/\text{Bitrate}$

Figura 5.22 – Resposta em frequência do filtro.



Fonte: O autor.

Conforme a figura 5.22, o ponto de -3dB corresponde a uma frequência normalizada de 0.42, que, quando convertida para o domínio analógico, corresponde a uma frequência de 10 KHz, adotando uma frequência de amostragem padrão de 48 KHz. Em seguida, os coeficientes do filtro projetados são inseridos no filtro polifásico *frequency xlating FIR*

*filter*, ainda podendo ser implementados em um filtro padrão do GNU (GNU, 2020). É necessário frisar o contexto discutido anteriormente a respeito da implementação do filtro pelos demoduladores, pois, para que o processo seja feito de maneira correta é necessário que a resposta em frequência do filtro seja similar a figura 5.22.

O fator de decimação, é calculado pela equação 4.7 e inserido no filtro polifásico, ou pode-se também utilizar o bloco referente ao filtro FIR citado acima.

Em seguida o *Quadrature Demod* realiza a demodulação FM, a saída do bloco é um pulso PAM. Diferentemente do demodulador implementado em hardware, o bloco *Symbol Sync* é utilizado para recuperação de símbolo. Os símbolos recuperados são submetidos ao detector, *binary slicer*, que realiza o mapeamento dos símbolos recebidos através da *hard decision*. A saída é sinal de linha em banda-base RZ, cuja variação da amplitude corresponde ao *stream* de bits recebido. Este demodulador é posto em prática na seção 5.4.3

### 5.4.3 Processamento de um *stream* de símbolos IQ no GNU Radio

Visando validar a metodologia difundida na seção 4.3.4, e também aplicar grande parte dos conceitos referentes aos processos realizados por um receptor digital abordados no trabalho, que podem ser implementados por software, bem como expor a funcionalidade do GNU Radio, pode-se processar um *beacon* obtido através de (KJSIEIT, 2023) convertido para frequência intermediária (IF) e gravado em .WAV.

Dessa forma, o presente teste visa expor como ocorre o processamento de sinal no software GNU Radio, bem como demonstra a funcionalidade dos blocos referentes ao módulo OOT, Gr Satellites (Daniel Esteves, 2020). Constituindo uma grande significância ao trabalho, já que um dos principais objetivos é utilizar o GNU Radio para processamento de sinais oriundos de um SDR, contendo *frames* de um nanossatélite, utilizando a metodologia da figura 4.12.

Ao realizar o processamento de forma indireta através do arquivo .WAV, não há a necessidade de desenvolvimento de uma API que interliga o software referente ao SDR e o GNU Radio, em virtude disso o processamento não é realizado em tempo real. O processo permite a extração do conteúdo útil no *frame*, a partir da análise das especificações do sistema de comunicação tais como modulação e protocolo. Isso é normalmente feito através de um processo de engenharia reversa, caso não há divulgação dos desenvolvedores. Caso o módulo OOT possua suporte para o nanossatélite operado, os *frames* decodificados podem ser salvos ou visualizados no prompt de comando, a visualização é feita através de um *parser* onde pode-se observar informações a respeito do subsistema do satélite.

Há uma variedade de protocolos suportados, como AX.25, CSP, AO-40FEC utili-

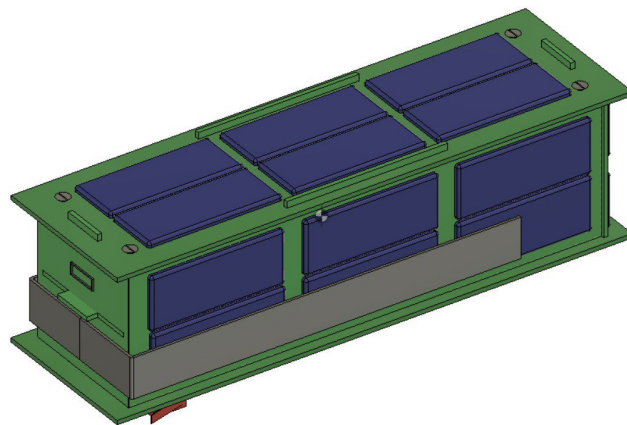
zado na família de *cubesats* FunCube dentre outros (Daniel Esteves, 2020). Para demonstração operacional do módulo é possível importar através do GitHub ou realizar o download de uma variedade de *beacons* disponibilizados para teste de demoduladores e decodificadores, estes usualmente são amostras obtidas através de um SDR ou radio convencional. Portanto, o módulo provê suporte para um *stream* de símbolos IQ oriundo de um SDR.

É possível operar o mesmo através do prompt de comando, porém por motivos operacionais a execução será por meio da sua integração com o software GNURadio. Demonstrando todas etapas referentes ao processamento de sinal.

Assim sendo, para verificar a operacionalidade do módulo e aplicar a metodologia apresentada, um teste será utilizado para demodulação e decodificação do *pocketqube* indiano BeliefSat-1. O *pocketqube* compreende a classe dos nanossatélites, sendo ligeiramente menor que um *cubesat*.

O BeliefSat-1 foi desenvolvido pelo instituto de tecnologia K.J Somaiya sendo parte do programa de construção da plataforma orbital PS4, e será utilizado também paralelamente ao SomaiayaPod, dispositivo responsável por inserir o dispositivo em órbita (KJSIEIT, 2023). Sua construção foi realizada majoritariamente utilizando componentes COTS nos subsistemas de comunicação, OBC e potência para demonstração tecnológica.

Figura 5.23 – Desenho 3D do BeliefSat-1



Fonte: (KJSIEIT, 2023)

As especificações do mesmo são encontradas em (KJSIEIT, 2023) e expostas na tabela abaixo:

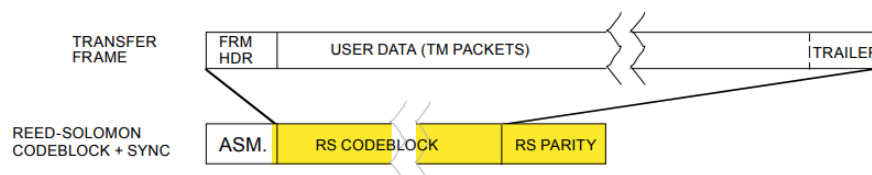
Tabela 5.8 – Especificações do BeliefSat-1

|                         |                                |
|-------------------------|--------------------------------|
| Modulação               | FSK                            |
| Frequência de operação  | UHF                            |
| Bitrate                 | 1200 bps                       |
| Formato do frame        | CCSDS Reed-Solomon             |
| Código Corretor de erro | Reed-Solomon(255,223)          |
| Protocolo               | CCSDS Reed-Solomon TM protocol |

Adaptado de: (KJSIEIT, 2023)

Durante a transmissão, os dados são formados e organizados de acordo com o protocolo especificado, implementando o código CCSDS Reed-Solomon (CCSDS, 2020). O formato padrão segundo o protocolo, para os *frames* codificados com o RS é exposto na figura 5.24. O código codifica todos os bytes referentes ao cabeçalho, data do usuário, e bytes adicionais para propósitos de correção no passo que o ASM, utilizado para sincronização não é codificado (CCSDS, 2020). É importante salientar que o usuário pode alterar o conteúdo presente no *frame*, de acordo com seu interesse, uma vez que o protocolo em alguns casos define somente a estrutura mas não conteúdo.

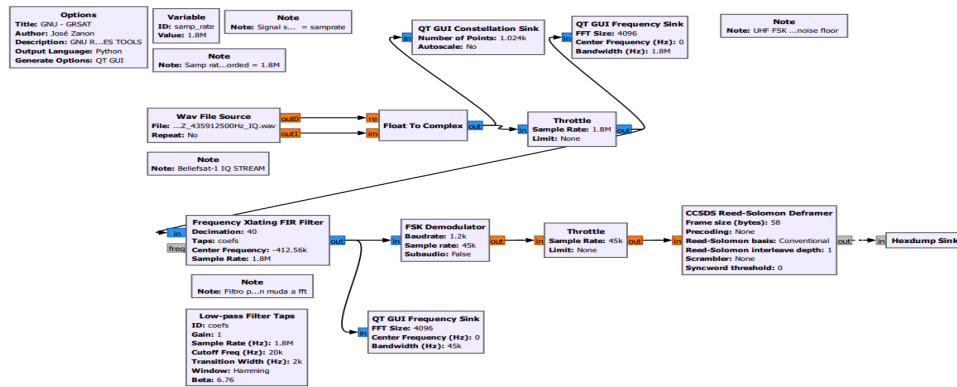
Isso pode ser visto na própria implementação do protocolo CCSDS no pocketcube, já que segundo (KJSIEIT, 2023) o *frame* possui *index*, utilizado para identificar os dados, como o tipo do pacote, o *call Sign* é o indicativo de chamada referente ao pocketcube, ainda há o número de *resets*, identificador, modo da missão, dados do sensor, e os códigos CRC e RS, utilizados na codificação.

Figura 5.24 – Estrutura dos CCSDS Reed Solomon *frames*.

Fonte: (CCSDS, 2020)

Para decodificação, os símbolos reconstruídos são inseridos no decodificador presente no módulo OOT, e baseado na tabela 5.8 o decodificador é implementado para adquirir um ou mais *frames*, no formato hexadecimal.

Figura 5.25 – Interface do demodulador+decodificador do BeliefSat-1

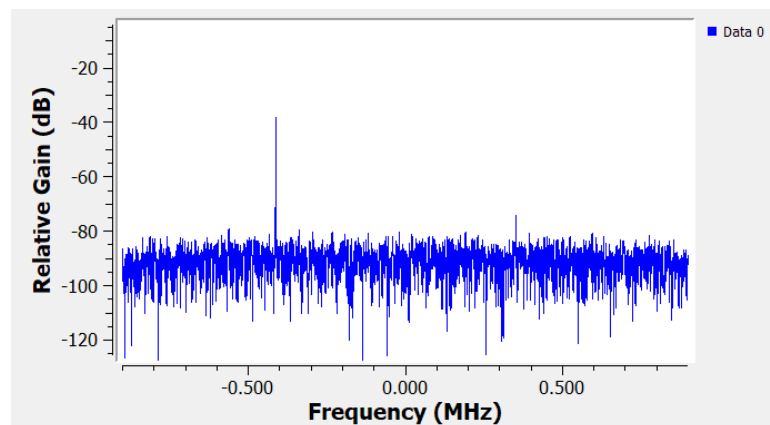


Fonte: O autor

Todas etapas referentes ao processamento do *stream* IQ seguem a figura 5.25. Usualmente o sinal recebido pelo satélite é submetido ao processo de down-conversão para a frequência intermediária (IF), visando registrar com fidelidade o sinal digitalizado, como por exemplo o *stream* de símbolos IQ do Beliefsat-1, com frequência centrada na faixa de 400KHz.

De início, o sinal deve ser convertido da banda-passante para banda-base e filtrado. Para realizar esses processos é necessário carregar o *stream* de símbolos IQ no bloco *Wav File Source*. A partir disso, utilizando o bloco *QT GUI Frequency Sink*, que consiste da aplicação da Transformada Rápida de Fourier (FFT), é possível visualizar a representação das componentes de frequência do sinal. A amplitude representa a intensidade do sinal em uma determinada frequência, isto é pode-se identificar a frequência predominante no sinal que pode estar relacionada com a informação transmitida, ainda que a amplitude pode ser afetada por fatores como ruído, atenuação e interferência.

Figura 5.26 – Espectro do sinal originalmente recebido.



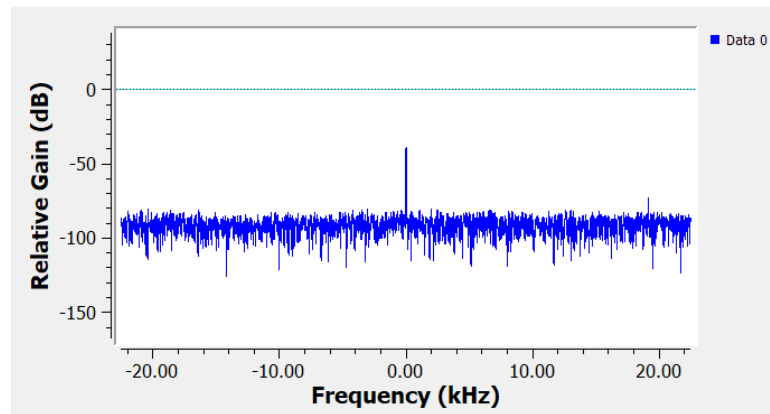
Fonte: O Autor.

A análise espectral da figura 5.26 revela uma frequência dominante, correspondente

à frequência de operação do satélite situada na faixa UHF. Isso pode revelar a transmissão de um pacote contendo dados telemétricos, ainda que há outros fatores ambientais tais como cenário multi-percurso e até mesmo componentes analógicos do receptor.

Para processamento do sinal é necessário que o sinal esteja em banda-base. Isso é realizado através de um filtro digital polifásico, o *frequency Xlating Fir Filter*. Os processos de filtragem e *downsampling* podem ser realizados também utilizando o mesmo bloco, nesse contexto, utiliza-se um filtro FIR passa-baixa para filtragem do sinal.

Figura 5.27 – Espectro do sinal após uso do *frequency xlating fir filter*

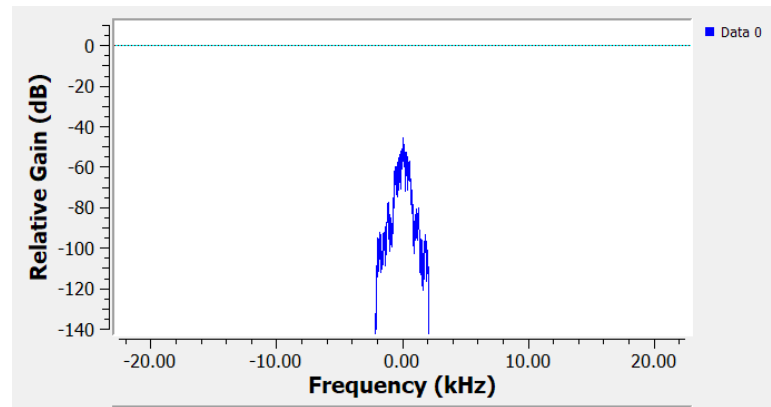


Fonte: O autor.

O filtro FIR pode ter seus coeficientes projetados através da ferramenta *low pass filter taps*, ou também pode-se projetar em outro software, como o MATLAB, e entrar com seus coeficientes. Nesse caso, os coeficientes do filtro projetado na seção 5.4.2 foram inseridos. É necessário também submeter o sinal ao processo de *downsampling*, para que haja re-amostragem do sinal e também a redução da taxa de amostras do sinal entregue ao demodulador, reduzindo assim o custo de processamento através de uma menor taxa de memória ocupada. Uma interpretação grosseira desse processo seria "jogar fora" algumas amostras do sinal (IOWEGIAN, 2020). No GNU Radio especifica-se um fator de decimação, para este caso a taxa de amostragem do sinal é de 1.8MHz, e para processamento nos blocos do GNU Radio, cujo definido por padrão é em torno de 48KHz. Como se deseja obter algo em torno de 48KHz, o fator situa-se na casa dos 40, calculado pela equação 4.7.

Ainda é possível filtrar o sinal novamente, removendo as demais componentes, deixando assim somente a componente de frequência referente ao pacote, por meio de um FIR passa-baixa projetado no próprio software. Especifica-se o ponto de -3dB em 2KHz, atenuando assim as demais componentes de frequência.

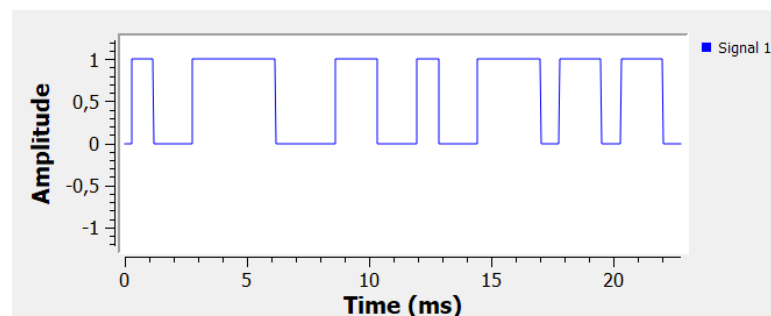
Figura 5.28 – Espectro do sinal após-refiltragem.



Fonte: O Autor.

Agora, a próxima etapa é a demodulação do sinal, de início testando o demodulador em quadratura proposto e posteriormente o demodulador FSK, disponibilizado pelo módulo OOT. Conforme exposto na secção 5.4.2, o demodulador de quadratura computa a frequência instantânea no sinal, no passo que um detector atua determinando o *stream* de bits referentes a *hard decision*, após o sincronismo de símbolo. Após isso, é possível visualizar através do display *QT GUI Time Sink* o sinal RZ demodulado.

Figura 5.29 – Sinal RZ demodulado.

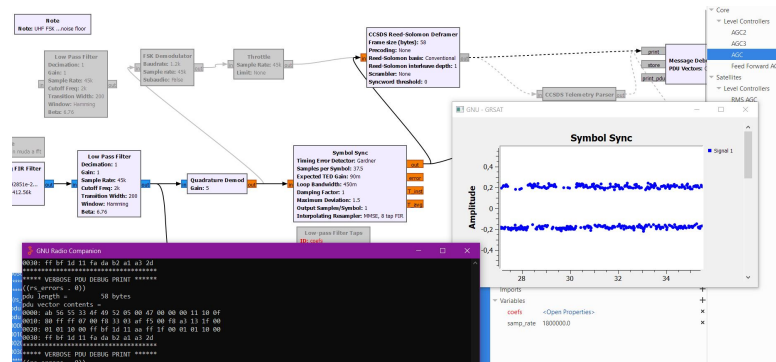


Fonte: O autor

O demodulador proposto em 5.4.2 utiliza o sincronismo de símbolo, necessário para correta demodulação do sinal, haja vista que o decodificador CCSDS Reed-Solomon opera somente quando o sincronismo é bem sucedido. Nesse sentido, caso se deseje realizar a demodulação utilizando o demodulador proposto na secção 5.4.2 a *hard decision*, realizada pelo bloco *binary slicer* não é performada. Para o demodulador proposto, o sincronismo resulta em:



Figura 5.30 – Visualização dos símbolos IQ sincronizados pelo bloco *symbol sync*.

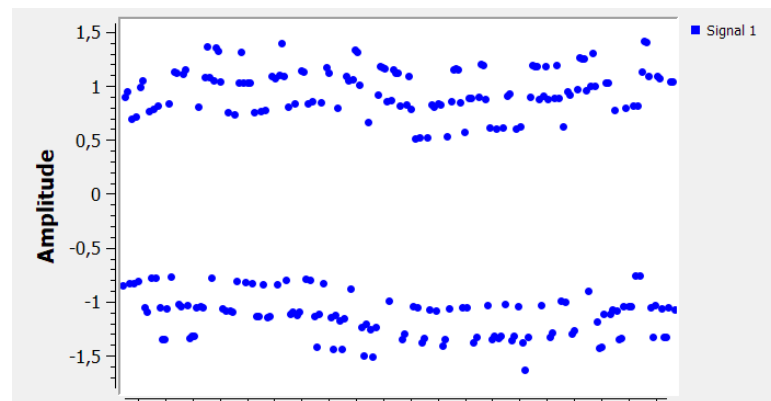


Fonte: O Autor

Ao utilizar o demodulador FSK do módulo OOT, também é possível visualizar também os símbolos IQ sincronizados, correspondentes aos picos do sinal PAM binário resultante do demodulador de quadratura. Este é o instante ótimo para amostragem performed pelo sincronismo conforme visto em (CHATZIGEORGIOU, 2020).

Ainda, o símbolo correspondente ao seu respectivo bit, ainda que distorcido, +1 representando o bit "1" e -1 representando o bit 0.

Figura 5.31 – Símbolos IQ recuperados oriundos do demodulador FSK.



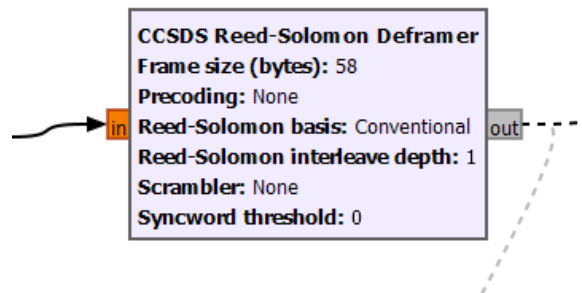
Fonte: O autor.

Conforme observado, ainda que a amplitude referente ao sinal PAM na figura 5.30 seja pequena quando comparada a figura 5.31, a *frame* decodificada é obtida da mesma forma, conforme observado no prompt situado no canto esquerdo da figura 5.30. Essa diferença é devido a utilização do DC block, AGC e filtro RRC no demodulador OOT.

Com estes símbolos IQ, o decodificador *CCSDS Reed-Solomon Deframer* pode levar em consideração não apenas se o bit está em 0 ou 1, mas também quão confiável é essa leitura, o que pode ajudar a melhorar a taxa de sucesso na decodificação do código.

Já no bloco referente ao decodificador:

Figura 5.32 – Decodificador OOT.



Fonte: O autor.

Especifica-se o tamanho do *frame* que segundo (KJSIEIT, 2023) é de 58 bytes, as outras opções são referente ao algoritmo do RS, como pré-codificação ou se o codificador utilizou um embaralhador, CCSDS Scrambler, visando randomizar os dados antes da transmissão. O BeliefSat não utiliza um *scrambler* e pré-codificação.

A saída do bloco referente ao decodificador, para todos os decodificadores do módulo OOT é em formato hexadecimal, chamado de PDU, do inglês *protocol data unit*, onde 1 byte é representado por dois dígitos. Ainda, os dados do pacote são organizados em vetores, sendo estes uma representação da informação transportada pelo pacote. Para visualização dos *frames* o bloco *Hexdump Sink* é utilizado.

Figura 5.33 – CCSDS Reed Solomon TM *frames* decodificados.

```
((rs_errors . 0))
pdu length = 58 bytes
pdu vector contents =
0000: ab 56 55 33 4f 49 52 05 00 48 00 00 11 10 0f
0010: 80 ff ff 07 00 f8 33 03 af f5 00 f8 a3 13 1f 00
0020: 01 01 10 00 ff bf 1d 11 aa ff 1f 00 01 01 10 00
0030: ff bf 1d 11 fa da b2 a1 66 09
*****
message_debug :info: The `print_pdu` port is deprecated and will be removed; forwarding to `print`.
**** VERBOSE PDU DEBUG PRINT ****
((rs_errors . 11))
pdu length = 58 bytes
pdu vector contents =
0000: ab 56 55 33 4f 49 52 05 00 46 00 00 11 10 0f
0010: 80 ff ff 07 00 f8 33 03 af f5 00 f8 a3 13 1f 00
0020: 01 01 10 00 ff bf 1d 11 aa ff 1f 00 01 01 10 00
0030: ff bf 1d 11 fa da b2 a1 1c 2b
*****
message_debug :info: The `print_pdu` port is deprecated and will be removed; forwarding to `print`.
**** VERBOSE PDU DEBUG PRINT ****
((rs_errors . 10))
pdu length = 58 bytes
pdu vector contents =
0000: ab 56 55 33 4f 49 52 05 00 47 00 00 11 10 0f
0010: 80 ff ff 07 00 f8 33 03 af f5 00 f8 a3 13 1f 00
0020: 01 01 10 00 ff bf 1d 11 aa ff 1f 00 01 01 10 00
0030: ff bf 1d 11 fa da b2 a1 a3 2d
*****
```

Fonte: O autor

Por último, o *frame* possui como destino as camadas subsequentes até a obtenção da informação útil. Não existe um *parser* implementado no GNU e não há um *telemetry viewer* disponibilizado por (KJSIEIT, 2023), para que se possa submeter os *frames* e visualizar os dados úteis.

Diante do exposto, os resultados referentes ao processamento do sinal do nanos-

satélite expressam grande parte das etapas de processamento de sinais em um receptor através de software, além de justificar parte da metodologia para trabalhar com uma gravação oriunda de um SDR que não está em banda-base. Os processos descritos e realizados no GNU elucidam de forma prática as etapas de conversão para banda-base, filtragem e *downsampling*, fazendo o uso de um filtro polifásico ao invés de *mixers*, filtros e *downsamplers* em quadratura. Ainda, aplica os conceitos referentes a demodulação e sincronismo de símbolo, implementando o demodulador proposto e o demodulador do módulo OOT.

O demodulador proposto neste caso apresentou resultados referentes a decisão abrupta, conforme visto em 5.29, onde há um sinal RZ em banda-base, cuja amplitude denota os binários. Ainda nesse mesmo contexto, o sincronismo teve êxito, sendo o demodulador proposto também operado em conjunto com decodificador CCSDS Reed Solomon. Já o demodulador adaptativo, foi também utilizado em conjunto com o decodificador, sendo este operado também com êxito. Além disso, um dos principais aspectos interessantes do módulo OOT é a implementação de vários decodificadores, suportados por uma variedade de nanossatélites. Esse aspecto foi exposto por meio do bloco referente ao decodificador CCSDS RS, o qual recebe os *frames* transmitidos pelo Tx do *cubeSat*, formatados de acordo com o protocolo CCSDS, realizando o processo de decodificação por meio de um RS(255,223).

Diante de todo o contexto, o trabalho fomenta a implementação de uma estação para nanossatélites de baixo custo. A prova disso se dá na implementação de grande parte de blocos de processamento de sinais digitais no software GNU Radio, neste caso foram realizados processos que em um receptor físico seriam implementados através de hardware, aumentando o custo do projeto. Conclui-se então que é possível realizar vários processos intrínsecos a um receptor por meio de software, já que no presente trabalho é de suma importância alcançar essa capacidade. Grande parte da implementação do projeto será inteiramente em software, o que confere uma magnitude significativa a essa abordagem.

#### 5.4.4 Decodificando o FLORIPASAT-1

Para que os processos de decodificação do sinal possam ser realizados corretamente, conforme a figura 4.12, a primeira etapa é levantar informações fundamentais do nanossatélite, expostas na secção de dimensionamento da estação.

Os *beacons* são transmitidos com a portadora centrada em 146MHz, a modulação é GFSK binária, que essencialmente é a modulação FSK com um filtro gaussiano no modulador para contenção espectral. O protocolo utilizado é o NGHam, que possui a estrutura dos *frames* de acordo com a figura 2.34, codificados por um código RS.

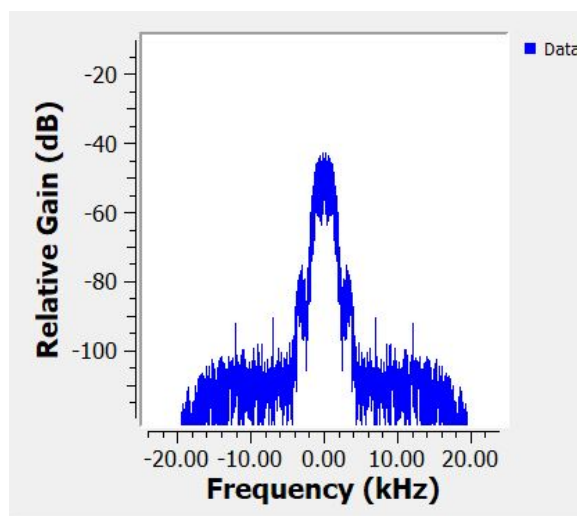
Em um primeiro momento, como o próprio *cubesat* não se encontra totalmente

operacional segundo o time de projeto, é preferível utilizar os sinais disponibilizados pela equipe do FLORIPASAT-1, gravados em um arquivo .WAV, referentes ao *beacon* situado em 146MHz e o *downlink* em 436MHz disponibilizados em (GRS, 2023).

Todo o processo de decodificação será realizado com o *beacon* em 146MHz, que já se encontra em banda-base e demodulado, isto é um *stream* de símbolos IQ demodulado e em banda-base. Nesse contexto, os processos de conversão para banda-base e demodulação não são necessários.

Após carregar no GNU Radio o arquivo .WAV, pode-se visualizar o espectro do sinal, através do bloco *QT GUI Frequency Sink*

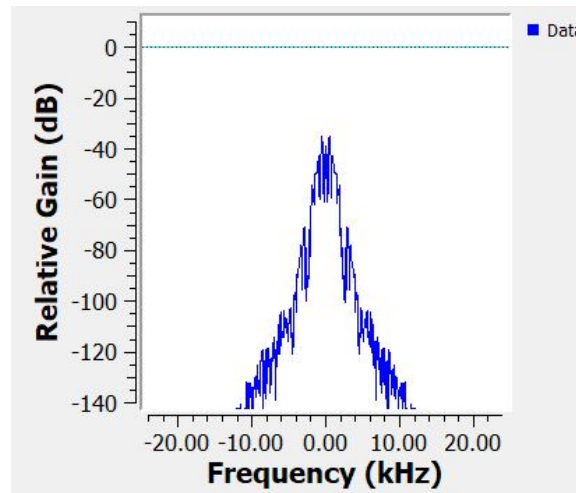
Figura 5.34 – Espectro do *beacon* recebido.



Fonte: O autor

A taxa de amostragem do sinal é de 48KHz, o que pela equação 4.8 resulta em 40 amostras por símbolo, nesse caso como queremos 8 amostras por símbolo decimamos por 5, inserindo esse valor no filtro FIR, conforme a equação 4.18. Ainda, pode-se filtrar novamente o sinal, com os coeficientes do filtro gaussiano projetado na secção 5.4.2, eliminando as componentes de frequência acima de 10KHz.

Figura 5.35 – Espectro filtrado.



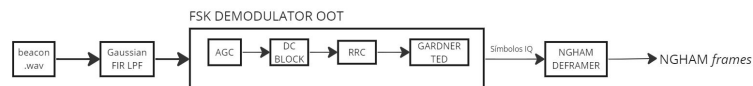
Fonte: O autor

A saída do filtro FIR é conectada ao bloco do demodulador do módulo OOT.

O NGHAm *deframer* opera somente com *soft decisions*, por isso, o demodulador do módulo OOT é adaptivo performando apenas o sincronismo de símbolo.

Logo, de acordo com a figura ??, O bloco referente ao AGC, ajusta o ganho do sinal no passo que o DC Block permite a passagem de frequências mais altas (incluindo a banda útil de frequência) enquanto atenua ou remove a frequência zero (DC), de forma semelhante a um filtro (BRITAIN, 2009). Ao remover a componente DC do sinal, o bloco ajuda a melhorar a qualidade do sinal demodulado, eliminando distorções ou deslocamentos indesejados (GNU, 2020). Após isso, conforme exposto no fonte referente ao demodulador FSK, o filtro RRC é utilizado como um filtro de formato de pulso, do inglês *pulse shaping filter*, e por último há o detector de erro de temporização de Gardner.

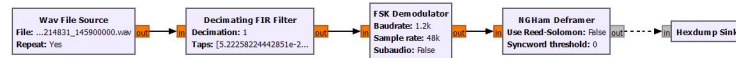
Figura 5.36 – Processos realizados equivalentes a figura 5.37.



Fonte: O autor

Dessa forma, com o decodificador do módulo OOT e o demodulador FSK é possível decodificar e obter os dados transmitidos, cuja implementação segue a figura 5.37.

Figura 5.37 – Fluxograma construído com o módulo OOT

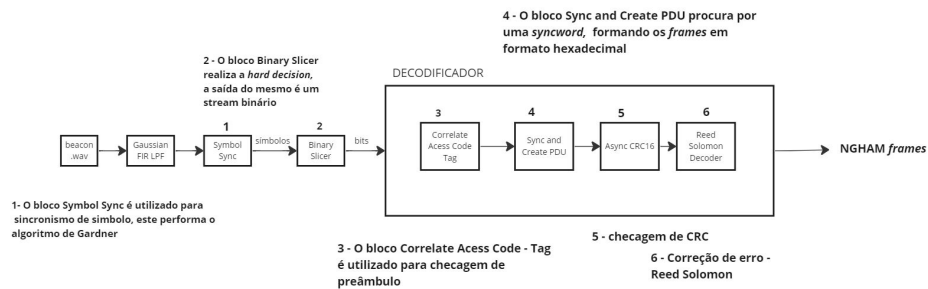


Fonte: O autor

O fluxograma acima realiza o sincronismo de símbolo e a decodificação, a saída decodificada é um arquivo em formato hexadecimal, que contém um pacote NGHAM. Este arquivo é popularmente chamado de PDU, *Protocol Data Unit*, que nada mais é o *frame* contendo as informações transmitidas.

Todavia, a metodologia adotada é diferente, exposta na figura 5.38, utilizando os blocos fornecidos pelo GNU Radio e módulo OOT, capazes de operar com um *stream* de bits.

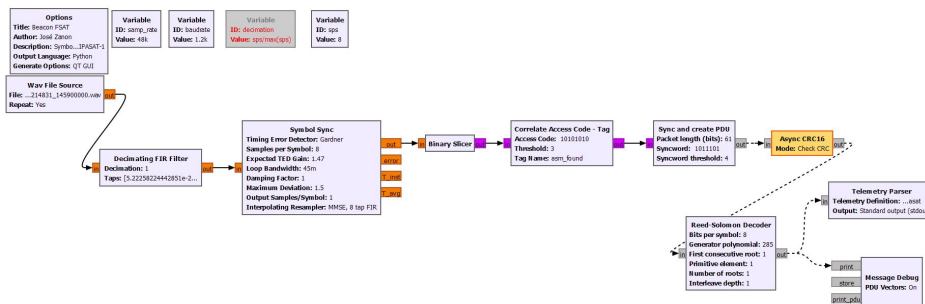
Figura 5.38 – Diagrama contendo o decodificador proposto.



Fonte: O autor.

No GNU Radio, o fluxograma ficou da seguinte forma:

Figura 5.39 – Fluxograma para sincronismo de símbolo e decodificação desenvolvido.

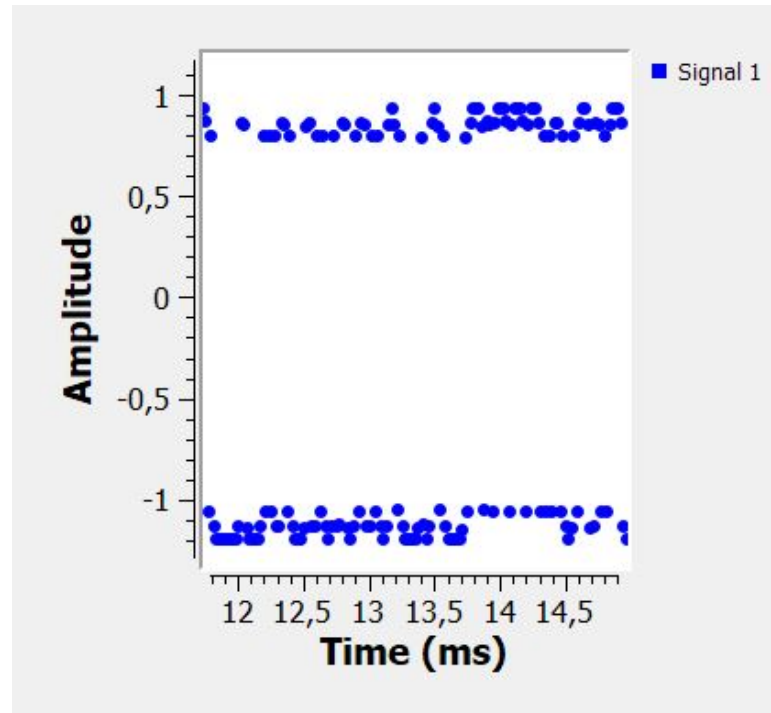


Fonte: O autor.

A primeira etapa consiste da utilização do filtro FIR passa-baixa gaussiano, idêntico ao citado acima.

O sincronismo de símbolo é implementado em seguida, por meio do bloco *symbol sync*, que utiliza a função de erro de Gardner. Os principais parâmetros a serem configurados são o número de amostras por símbolo, configurado em 8, e o número de amostras por símbolo na saída do bloco de sincronismo, cujo valor é 1.

Figura 5.40 – Símbolos IQ recuperados pelo *symbol sync*.



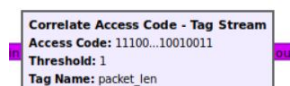
Fonte: O autor

O tempo ótimo para sincronismo segue os picos referentes ao sinal PAM, em +1 e -1, conforme fomentado na seção 5.4.3

A saída deste bloco é conectada ao *binary slicer*, que realiza a *hard decision*, através da amplitude da amostra do sinal recebido.

O *stream* de bits oriundo do bloco é então manipulado através do *correlate access code*, utilizado para checagem de *flag*, e funciona através da especificação de um preâmbulo (GNU, 2020).

Figura 5.41 – *Correlate access code tag*.

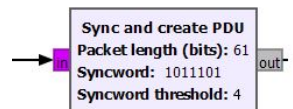


Fonte: O autor

A principal configuração do bloco é o código de acesso, equivalente ao preâmbulo *0XAA* fornecido por (Space Lab UFSC, 2022).

Após a detecção do preâmbulo, o bloco *sync and create PDU*, realiza duas tarefas, a primeira é a localização das palavras de sincronização, do inglês *syncword*, no *stream* e a segunda é a organização de um PDU, após a detecção da *syncword*, cujo o tamanho em bytes deve corresponder ao tamanho do pacote enviado. Portanto, deve-se fornecer o tamanho do *frame* e a *syncword*.

Figura 5.42 – Bloco *sync and create PDU*.



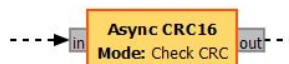
Fonte: O autor

Nessa etapa, a principal dificuldade foi adequar o tamanho do *frame* recebido, para que o bloco descrito acima possa operar de maneira correta. As informações divulgadas no protocolo afirmam que o mesmo pode ter tamanho variável, de acordo com a informação que se deseja transmitir, pois os projetistas do nanossatélite possuem total liberdade de modificar o mesmo.

Constatou-se através de testes que o tamanho do *payload* é de 58 bytes, ainda 2 bytes referente ao preâmbulo e a *syncword*, respectivamente, são adicionados, bem como 2 bytes correspondem ao CRC, e os demais são os bytes de paridade do RS, podendo ser o 16 ou 32 bytes, conforme especificado no protocolo NGHAM.

As próximas etapas são referentes a correção de erros, nesse sentido o Reed-Solomon e o código de verificação cíclica de redundância são implementados. O CRC, utilizado para verificar a integridade dos dados recebidos, é implementado pelo bloco *Async CRC16*, já que 16 bits ou 2 bytes são especificados para o CRC conforme o protocolo NGHAM, dessa forma o bloco verifica os bytes referentes ao CRC na PDU.

Figura 5.43 – Código CRC.

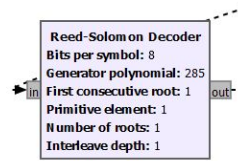


Fonte: O autor

Ademais, a configuração do RS foi mantida padrão, operando com 8 bits por símbolo ou 1 byte por símbolo e *interleaver* de profundidade 1.



Figura 5.44 – Decodificador Reed Solomon



Fonte: O autor

Com respeito a isso, existe uma discussão referente a má implementação do código RS no codificador de canal do nanossatélite (ESTEVES, 2019).

A saída do bloco RS é conectada ao bloco *message debug*, para visualização de um ou mais *frames* decodificados.

Figura 5.45 – Frames do FLORIPASAT-1 decodificados em formato hexadecimal.

```

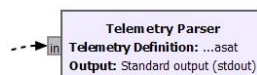
**** VERBOSE PDU DEBUG PRINT ****
((rs_errors . 1))
pdu length =      58 bytes
pdu vector contents =
0000: 00 00 00 01 00 01 01 01 01 00 00 01 00 00 01 00
0010: 00 01 00 00 00 01 00 00 01 01 00 00 01 01 00 01
0020: 00 00 00 01 00 00 00 00 01 00 00 01 01 00 00 00
0030: 00 01 00 00 00 01 01 01 01 00
*****
**** VERBOSE PDU DEBUG PRINT ****
((rs_errors . 1))
pdu length =      58 bytes
pdu vector contents =
0000: 00 00 01 01 01 00 01 01 01 00 01 01 00 00 00 01
0010: 01 01 01 00 01 00 00 01 00 01 01 00 01 00 01 01
0020: 01 01 01 01 01 01 00 00 00 01 01 01 01 00 00 01
0030: 00 01 00 01 00 00 00 00 01 01
*****
**** VERBOSE PDU DEBUG PRINT ****
((rs_errors . 1))
pdu length =      58 bytes
pdu vector contents =
0000: 00 01 01 01 00 00 01 00 00 00 01 01 00 01 01 01
0010: 00 01 01 00 00 00 01 01 01 01 00 00 00 01 01 01
0020: 00 01 01 00 01 00 00 00 01 00 00 01 00 00 01 01
0030: 00 00 01 00 01 01 01 00 00 00
*****
**** VERBOSE PDU DEBUG PRINT ****

```

Fonte: O autor

Por último, estes dados obtidos podem ser manipulados, como por exemplo através de um *parser* implementado no módulo OOT, para que as informações referentes a telemetria possam ser visualizadas.

Figura 5.46 – Telemetry Parser.



Fonte: O autor.

Figura 5.47 – Dados contidos no *payload* visualizados através do *telemetry parser*.

```
Container:
  ngram_padding = 0
  id = (enum) beacon_ngram_obdh_data 0
  callsign = b'\x00\x01\x00\x01\x01\x01\x01' (total 7)
  payload = Container:
    eps = Container:
      battery_voltage = ListContainer:
        0.0
        0.039064
      battery_temperature = ListContainer:
        1.0
        1.0
      battery_charge = 0.0
      solar_panel_current = ListContainer:
        0.03788803788803788
        0.00014800014800014796
        0.03788803788803788
        0.00014800014800014796
        0.03788803788803788
        0.03788803788803788
      solar_panel_voltage = ListContainer:
        0.0005683760683760684
        0.15685652235652235
        0.0005683760683760684
      energy_level = 0
    status = Container:
      antenna = False
      imu = False
      sd_card = False
      rush = False
      eps = False
      imu_accelerometer = ListContainer:
        0.125
        0.00048828125
        0.125
      imu_gyroscope = ListContainer:
        0.0
        1.953125
        0.0
      system_time = Container:
        seconds = 1
        minutes = 65793
      obdh_resets = 0
```

Fonte: O Autor

Neste caso, constata-se que o *beacon* foi enviado normalmente, correspondendo a um tamanho de *payload* de 58 bytes, contendo todas as informações conforme especificado em (Space Lab UFSC, 2022). Validando portanto o processo de decodificação.

Figura 5.48 – Dados contidos no *beacon*.

Em operação normal do satélite, os pacotes de beacon contém os seguintes dados:

| Campo   | Posição | Tamanho [bytes] | Conteúdo     |
|---|---------|-----------------|--------------|
| ID do satélite  | 0       | 10              | "FLORIPASAT" |
| Tensão nas baterias                                   | 10      | 4               | Variável     |
| Temperatura das baterias                              | 14      | 6               | Variável     |
| Carga total das baterias                              | 20      | 2               | Variável     |
| Corrente dos painéis solares                          | 22      | 12              | Variável     |
| Tensão dos painéis solares                            | 34      | 6               | Variável     |
| Status geral do satélite                              | 40      | 2               | Variável     |
| Medidas do acelerômetro e do giroscópio               | 42      | 12              | Variável     |
| Tempo desde a última inicialização                    | 54      | 4               | Variável     |
| Número de reinicializações do OBDH desde o lançamento | 58      | 2               | Variável     |

Fonte: (Space Lab UFSC, 2022)

Assim sendo, o processo de decodificação referente ao FLORIPASAT-1, anteriormente a operacionalização da estação terrena, valida a implementação do segmento do decodificador para este nanossatélite. Logo, operacionalizar este segmento através dos

blocos de processamento presentes no GNU Radio e no módulo OOT, elucidam de forma prática um dos principais objetivos propostos no trabalho, referente a implementação de um receptor quase que totalmente em software. Pois, em campo é possível utilizar esta exata metodologia implementada.

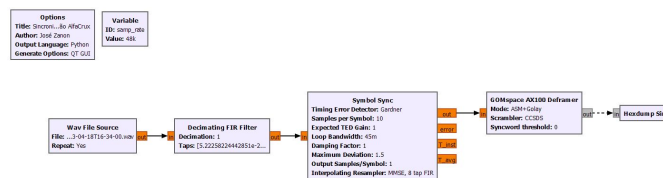
#### 5.4.5 Decodificando o AlfaCruz

Haja vista que a implementação em campo exposta na seção 5.5 não foi bem sucedida, é possível trabalhar com o *beacon* do *cubesat* em formato .WAV disponibilizado pelo *SatNOGS* (SATNOGS, 2023). O projeto SatNOGS é uma plataforma de código aberto que visa criar uma rede de estações terrestres via satélite (SATNOGS, 2023).

Dessa forma, com o *beacon* em .WAV gravado e disponibilizado por uma estação terrena e compartilhada no *SatNOGS* é possível realizar os processos de sincronismo de símbolo e decodificação no GNURadio para obter os *frames* transmitidos pelo *cubesat* cujo formato é definido pelo protocolo CSP.

Os sinais disponibilizados no *SatNogs* são convertidos para banda-base, demodulados e possuem uma frequência de amostragem de 48KHz.

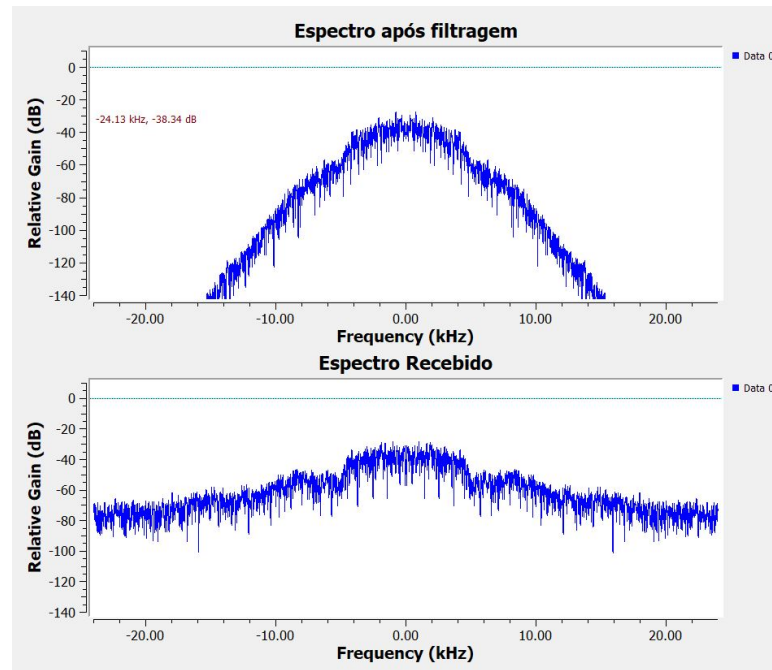
Figura 5.49 – Fluxograma construído para sincronismo de símbolo e decodificação



Fonte: O autor

Após carregar no GNU Radio o arquivo .WAV, pode-se visualizar o espectro do sinal, através do bloco QT GUI Frequency Sink. Ao mesmo tempo o sinal é filtrado utilizando o bloco *decimating FIR filter*, cuja resposta é exposta na secção 5.4.2.

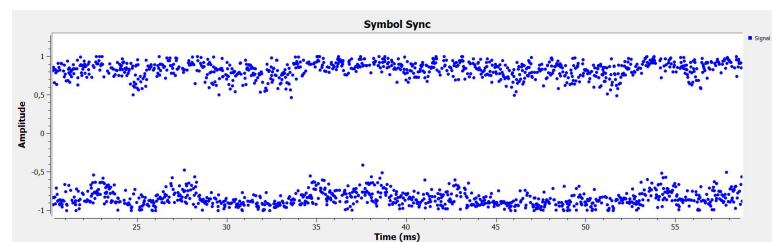
Figura 5.50 – Espectro originalmente recebido vs espectro filtrado.



Fonte: O autor.

Como a taxa de amostragem situa-se em 48KHz e a taxa de símbolos por segundo é igual a 4800 símbolos por segundo, o número de amostras por símbolo entregue ao bloco *symbol sync* e ao decodificador AX-100 *deframer* é igual a 10, segundo a equação 4.8. Inserindo este valor na configuração do bloco *symbol sync*, é possível performar corretamente o sincronismo de símbolo utilizando a função de erro de Gardner. De forma idêntica a secção 5.4.3, o sincronismo enquadra as amostras nos picos do sinal PAM, já que o sinal GMSK foi transformado em um pulso PAM pelo demodulador.

Figura 5.51 – Visualização dos símbolos IQ sincronizados pelo bloco *Symbol Sync*.



Fonte: O autor.

O decodificador AX-100 *deframer* apresentado na secção 4.3.3.1 utiliza os *frames* utilizados pelo transceptor NanoCom AX-100, utilizado pelo AlfaCrux. O decodificador é configurado no modo ASM+Golay, com um desembaralhador CCSDS, haja vista que a codificação dos *frames* do AlfaCrux é performada por um Golay(24,12) e um CCSDS

Reed-Solomon. A saída do bloco consiste nos *frames* ou PDU's decodificados enviados pelo Tx do *cubesat*, cujo formato é definido pelo protocolo CSP.

Figura 5.52 – *ASM+Golay frames* decodificados.

```
(  
pdu length =      208 bytes  
pdu vector contents =  
0000: 82 f3 9d 00 10 11 0e 00 0e 08 01 ff ff 00 ba 10  
0010: 03 19 92 8f 03 e8 4c 09 03 8f 6c ff 00 01 00 00  
0020: 07 d0 01 00 01 73 f4 00 00 00 00 00 00 00 43  
0030: a6 ec 4f 43 7a 00 00 c3 8b 3b 14 00 87 1a 88 00  
0040: 00 07 ff be a5 46 54 bf ef 1e f1 bf 3f a5 fa 40  
0050: 28 82 1c 00 00 00 00 03 e0 00 00 00 00 00 02  
0060: fb 60 00 01 01 00 00 00 00 01 01 01 00 00 0a  
0070: 00 0c 00 03 00 00 00 00 00 00 00 04 00 00  
0080: 00 04 00 15 64 3e c6 d1 00 1e 78 88 01 00 00 01  
0090: 87 95 38 a5 44 00 00 00 00 00 00 2a 01 00 00  
00a0: 76 31 2e 32 2e 30 2d 31 31 31 2d 32 30 32 32 30  
00b0: 32 31 36 2d 31 31 33 39 33 34 00 00 00 00 00  
00c0: 02 02 00 00 00 00 01 01 8e 30 00 00 0f b7 fb  
*****  
message_debug :info: The `print_pdu` port is deprecated and will be removed; forwarding to `print`.  
***** VERBOSE PDU DEBUG PRINT *****  
(  
pdu length =      175 bytes  
pdu vector contents =  
0000: 82 f3 9d 00 10 11 0f 00 0e 08 01 ff ff 00 99 10  
0010: 03 19 92 90 03 e8 4c 09 03 8f 6d 2e 00 02 00 00  
0020: 07 d0 01 00 00 01 87 95 38 a2 2b 00 00 00 00  
0030: 00 00 1b 00 00 00 00 02 86 ed 00 00 00 00 00  
0040: 00 00 0b 00 00 00 00 00 00 00 00 00 00 0d 0d  
0050: a4 0d c3 0d c2 20 73 00 52 02 06 00 b6 01 16 00  
0060: 67 00 44 00 01 00 03 00 4d 00 01 00 02 00 00  
0070: 00 00 00 00 00 00 00 00 03 00 04 00 03 00
```

Fonte: O autor.

Para a aplicação, e visualização dos dados é necessário inserir estes *frames* no *telemetry viewer* elaborado pelos desenvolvedores do AlfaCrux, esta aplicação possui como entrada os *frames* KISS, cuja estrutura é definida pelo protocolo KISS, normalmente utilizado para o transporte de *frames*.

Figura 5.53 – *Telemetry Viewer* do AlfaCrux.

Were you able to receive AlfaCrux packages? Share with us and become one of our distinguished collaborators. For that, just fill the form below and send your KISS frames. It will be checked by the AlfaCrux team, and later uploaded in our database associated to your call sign!

|   |   |
|---|---|
| Email address<br><input type="text" value="Enter email"/> | Kiss Frame File<br><input type="text" value="Escolher arquivo"/> Nenhum arquivo escolhido |
| Name<br><input type="text" value="Complete Name"/>        | Latitude<br><input type="text" value="Latitude"/>   |
| Call Sign<br><input type="text" value="Call Sign"/>       | Longitude<br><input type="text" value="Longitude"/>                                       |

If needed, click here to get the coordinates of your location

Fonte: O autor.

Ainda, como os desenvolvedores não expõem o conteúdo de *payload* presente nos *frames* do tipo *ASM+Golay*, o processo de visualização destes *frames* em formato hexadecimal torna-se inviável, já que cada valor hexadecimal exposto nos *frames* tem sua respectiva correspondência em termos telemétricos. O que é indispensável para o desenvolvimento de uma aplicação que torne estes valores em valores de fácil interpretação.

## 5.5 IMPLEMENTAÇÃO EM CAMPO

A implementação em campo consiste de uma tentativa de recepção do *beacon* do *cubesat* AlfaCruz, para tanto houve a integração da antena, o SDR RSP1A e o computador. Essa integração visava realizar o processamento do *beacon* de forma indireta, isto é um arquivo .WAV gerado pelo software SDRUno referente ao *beacon* deve ser inserido no GNURadio para demodulação e decodificação, extraindo os *frames* transmitidos.

Para isso, necessitou-se de um adaptador *pigtail* do tipo N fêmea + SMA macho para união do SDR e antena.

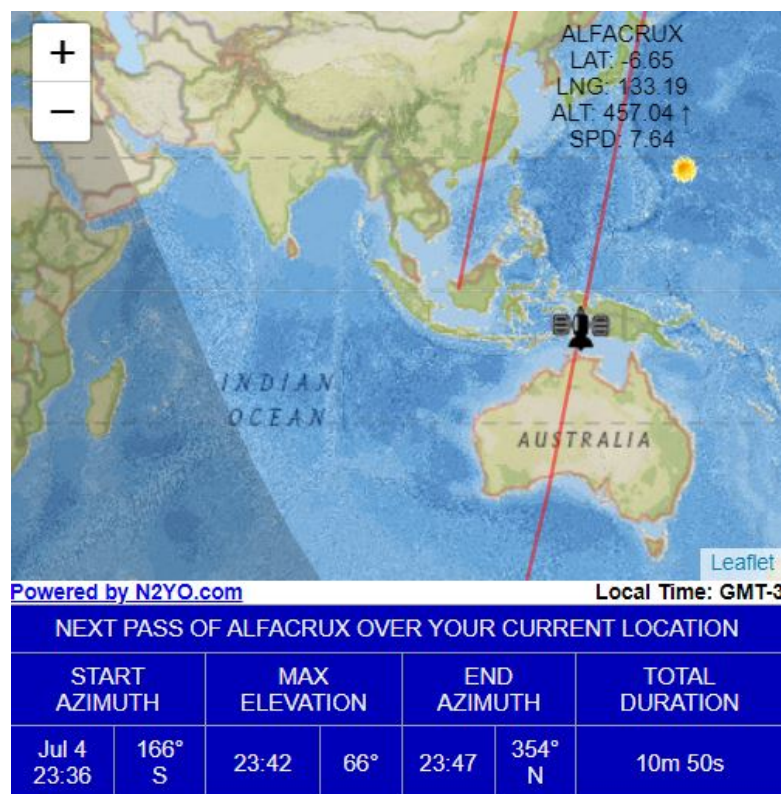
Figura 5.54 – Adaptador *pigtail* e SDR.



Fonte: O autor

Para realizar a recepção, é necessário realizar o rastreamento do *cubesat*, afim de orientar a elevação e azimute para que o apontamento seja correto. Para isso o website N2YO foi utilizado.

Figura 5.55 – Interface do N2YO.



Fonte: O autor

A orientação foi realizada conforme o azimute de inicio e fim, para ajuste do apontamento inicial. O apontamento da antena foi totalmente manual, conforme o nível de sinal recebido verificado no computador pelo SDRUno.

Figura 5.56 – Antena + SDR + Computador.



Fonte: O autor

Figura 5.57 – Apontamento manual em campo.



Fonte: O autor

No SDRuno, é necessário também configurar a interface para recepção:

Figura 5.58 – Configurações do receptor.



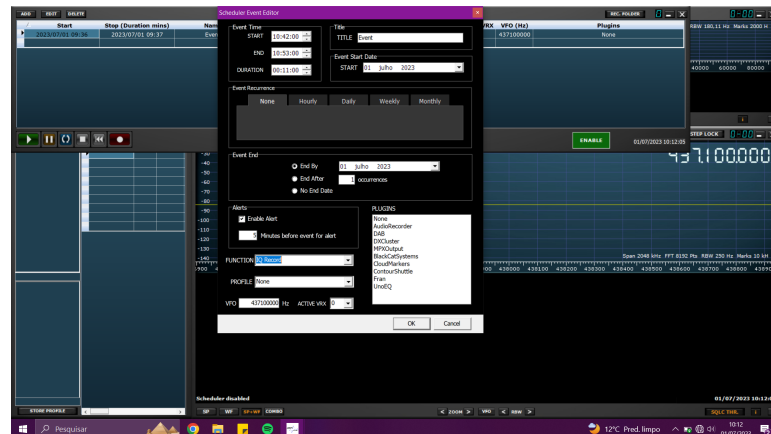
Fonte: O autor

O controle de ganho automático do RSP1A é configurado em *fast AGC*, para compensar a perda de sinal devido ao apontamento. O modo é *IQ out*, para obter o *stream de símbolos IQ*, o filtro foi configurado em 20KHz, e o *fm mode* foi configurado em *narrow fm*, já que o sinal ocupa apenas uma porção pequena do espectro.

É necessário configurar o *scheduler* para gravação do sinal.



Figura 5.59 – Scheduler do SDRUno.



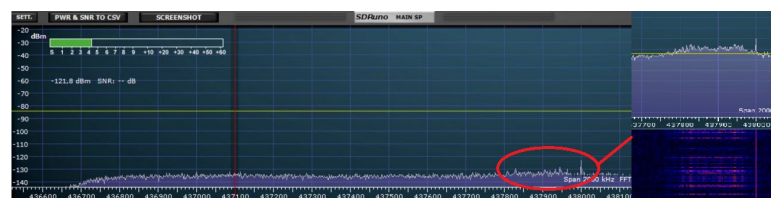
Fonte: O autor

Como resultado, foi possível obter o *beacon* transmitido, haja vista que não havia nada indicado na interface do SDRUno momentos antes de iniciar o contato com o *cube-sat*. O momento cujo sinal foi captado de maneira mais precisa foi correspondente aos primeiros minutos da passagem, no entanto essa recepção foi muito fraca, pois o *beacon* situou-se pouco acima do *noise floor*, sendo este o nível onde não é possível performar nenhum tipo de detecção.

Figura 5.60 – Espectro recebido.



Fonte: O autor

Figura 5.61 – *beacon* recebido.

Fonte: O autor

Na figura 5.61, o gráfico abaixado no analisador de espectro do SDRUno é a *waterfall*, que mostra o conteúdo na frequência do sinal, pode-se constatar que nas faixas de frequências adjacentes a destacada em vermelho não há variação no ganho. Ainda,

o receptor foi configurado em 437.1 MHz, todavia o RSP1A utilizado não foi calibrado anteriormente, este necessita de uma pré-calibração do oscilador local e por isto o *beacon* recebido sofreu este desvio de frequência , acrescentado do desvio doppler, que para um satélite em LEO é por volta de 10KHz. Ainda o arquivo WAV gravado foi aberto no GNURadio, mas haja vista que o nível do sinal situa-se pouco acima do *noise floor*, o processamento é inviável.

## 6 CONCLUSÃO

O presente trabalho objetivou propor o projeto de uma estação terrestre para *cubesats* via rádio definido por software. Para tanto, uma série de etapas foram elaboradas com o objetivo de expor o projeto de maneira simples, dada a multidisciplinaridade dos conceitos envolvidos. Para alcançar o objetivo principal, vários sub-objetivos foram elaborados e realizados, levando em conta a complexidade do trabalho. Esses sub-objetivos incluem desde um levantamento preliminar de dados sobre *cubesats* operantes em órbita até o projeto de uma antena.

Dessa forma, uma das principais etapas para a realização do trabalho foi o levantamento de dados preliminares, para que, a partir dele, o *link budget* possa ser confeccionado. A respeito disso, verificou-se que, para os dois *cubesats* propostos inicialmente, o receptor possui a capacidade de operar com o sinal recebido, dada as condições do link entre satélite e estação terrena. O *link budget* é uma etapa fundamental do trabalho, pois ele é definido a partir dos dados do próprio SDR e é o parâmetro inicial para o projeto da antena, definindo o ganho mínimo necessário para a operação. Existem uma variedade de antenas que podem ser utilizadas para estabelecer contato com o nanossatélite, todavia a yagi-uda de polarização circular é amplamente utilizada para esse propósito e foi escolhida.

A antena foi projetada preliminarmente em software, juntamente com o seu casador de impedância. Neste contexto, ela atendeu aos requisitos de projeto. No entanto, na prática, algumas dificuldades foram encontradas durante a construção, principalmente relacionadas ao segmento de alimentação. Além disso, a fixação realizada com durepoxi também contribuiu para a divergência da ressonância da antena em relação à frequência de projeto.

O software GNU Radio foi proposto para demodulação e decodificação. No GNU Radio, a viabilidade do demodulador foi estudada por meio de simulações, permitindo avaliar e implementar esse processo de forma conceitual. Além disso, realizou-se uma simulação para compreender o funcionamento dos demoduladores já presentes no GNU Radio. Em seguida, o demodulador proposto foi utilizado para processar um *stream* de símbolos IQ transmitidos por um *cubesat*. Após a integração do módulo OOT Gr Satellites, o autor também implementou o demodulador FSK adaptativo.

Apesar do software utilizado disponibilizar um demodulador específico para as modulações FSK e GMSK, o demodulador de quadratura proposto mostrou-se capaz de trabalhar com diferentes esquemas de modulações binárias, incluindo essas. Além disso, ele pode ser integrado com o filtro FIR presente no GNU Radio.

Com relação ao segmento de decodificação, encontraram-se muitas incertezas, principalmente devido à escassez de informações divulgadas sobre os *frames* transmi-

dos pelos *cubesats*. Por exemplo, no protocolo NGHAM, existem 7 tamanhos diferentes de *frames*. No entanto, a implementação dos segmentos de decodificação foi concebida com o auxílio do Gr Satellites, o que facilitou o trabalho do autor, pois pôde ser implementado em conjunto com os blocos do GNU Radio.

Esses segmentos referentes ao GNU Radio comprovam a funcionalidade contida no software, pois através dele é possível realizar a demodulação e decodificação. Adicionalmente, o autor pode expor uma metodologia de processamento de símbolos IQ, o que possui uma grande magnitude no trabalho.

Ainda, a parte mais importante do trabalho é exposta ao decodificar os *frames* do Beliefsat-1, AlfaCrux e FLORIPASAT-1, provando assim a metodologia adotada pelo autor e expondo o propósito final do trabalho por meio da operacionalização do segmento do receptor de forma indireta através de *beacons* no GNU Radio. No processo de decodificação do AlfaCrux, constatou-se a falta de informações referentes à estrutura dos *frames* ASM+Golay transmitidos pelo transceptor da NanoCom, o que impossibilita a interpretação dos *frames* decodificados em formatos hexadecimal. Além disso, erros na plataforma CMAKE impossibilitaram a implementação de um decodificador de Golay, tornando a decodificação totalmente dependente do decodificador AX-100 *deframer*.

Por último, uma integração do SDR, antena e computador foi realizada, na tentativa de receber o *beacon* do *cubesat* AlfaCrux, todavia essa recepção não foi totalmente bem sucedida, haja vista o baixo nível de sinal adquirido, o que impossibilita o processamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AGILENT. **Digital Modulation in Communications Systems An Introduction**. US: Artech House, 2002.
- ALILI, G. **Software Defined Radio Based Communication Subsystem for C3 Ground Control Station**. 2021. 73 f. Dissertação (Master of Science - Computer Engineering) — Politecnico di Torino, Italy, 2021.
- ALOUINI, N. S. . A. E. . H. A. . H. D. . T. Y. A.-N. . M.-S. Cubesat communications: Recent advances and future challenges. **IEEE**, v. 1, p. 24, 2020.
- ALVIM, M. S. **Information Theory: The source coding theorem**. UFMG, 2021. Acesso em 2 Mai. 2023. Disponível em: <<https://homepages.dcc.ufmg.com.br>>.
- ASVIAL, I. M. Digital communication inter symbol interference, coherent and non coherent detection. **Center for Information and Communication Engineering Research**, v. 1, p. 10, 2019.
- ATLANTARF. Antennas : An overview. **Atlanta RF.com**, v. 1, p. 55, 2013.
- BABIC, D. B. . J. V. . D. Link budget analysis for a proposed cubesat earth observation mission. **Ericsson Nikola Tesla d.d. Research and Development Centre**, v. 1, p. 6, 2016.
- BALANIS. Gamma match. In: WILEY (Ed.). **Antenna Theory, Analysis and Design**. 7. ed. USA: Wiley, 2015. cap. 9, p. 20.
- BALANIS, C. A. **Antenna Theory Analysis and Design**. ASU: Wiley, 1997. 959 p.
- BALDAN, N. A. S. G. H. K. L. C. F. L. M. I. M. S. T. G. A. L. L. M. R. Comportamento eletromagnético da resina epóxi sob a ação de frequências na banda x em um radome. **WETE**, v. 1, p. 10, 2022.
- BARBIN, A. S. . A. L. R. . K. G. L. L. M. . S. Introduction to the software-defined radio approach. **IEEE LATIN AMERICA TRANSACTIONS**, v. 10, p. 6, 2012.
- BARBOZA, Y. **O modelo OSI e suas camadas**. Alura, 2023. Acesso em 4 Mai. 2023. Disponível em: <<https://www.alura.com.br/artigos/conhecendo-o-modelo-osi>>.
- BERLEKAMP, E. R. **Algebraic Coding Theory**. US: McGraw Hill, 1968.
- BIGLIERI, S. K. W. W. **Academic Press Library in Mobile and Wireless Communications**. US: Elsevier, 2016.
- BOUALLEGUE, R. A. . K. R. . R. On the evaluation of gmsk scheme with ecc techniques in wireless sensor network. **International Journal of Wireless Mobile Networks (IJWMN)**, v. 7, p. 13, 2015.
- BRIDGES, M. R. M. . C. P. Software defined radios for small satellites. **2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)**, v. 1, p. 8, 2014.
- BRITAIN, R. S. of G. **The Radio Communication Handbook, 6th edition**. UK: RSGB, 2009. 864 p.

CAMACHO-LARA, J. N. P. S. M. S. **Handbook of Satellite Applications**. US: Springer, 2013.

CAMPBELL. The link budget and fade margin. **Campbell Scientific Application note**, v. 1, p. 18, 2016.

CANTU, G. C. **Implementação de Toolbox Octave de Técnicas de Sincronismo de Símbolo em Sistemas de Telecomunicações**. 2019. 68 f. Dissertação (Master of Science) — Instituto Federal de Santa Catarina, BR, 2019.

CAO, C.; ZHAI, S. The influence of leo satellite doppler effect on lora modulation and its solutions. **Journal of Physics: Conference Series**, v. 1, p. 7, 2021.

CAPELA, C. J. R. **Protocol of Communications for VORSAT Satellite**. 2012. 25 f. Dissertação (Master Degree in Electrical Engineering) — Universidade de Porto, Portugal, 2012.

CASTRO, F. D. C. . M. D. Acoplamento antena-cabo através de gamma match (acoplador gama) procedimento automatizado. **fccdecastro.com.br**, v. 1, p. 5, 2022.

CB. **Gamma Match Construction**: Gamma match diagram. CB, 2020. Acesso em 6 Nov. 2022. Disponível em: <<https://www.google.com/search?q=gamma+match&oq=gamma+match&aqs=chrome..69i57.2256j0j4&sourceid=chrome&ie=UTF-8#imgrc=TbXwQPQc8E64pM>>.

CCSDS. **TM SYNCHRONIZATION AND CHANNEL CODING SUMMARY OF CONCEPT AND RATIONALE**. ccsds.org, 2020. v. 1, 130 p. Acesso em 30 mai. 2022. Disponível em: <<https://public.ccsds.org/default.aspx>>.

CHAKRAVARTY, P. K. Cyclic redundancy code (crc) polynomial selection for embedded networks. **The International Conference on Dependable Systems and Networks(DSN)**, v. 1, p. 11, 2004.

CHATZIGEORGIOU, R. **Experimental study of satellite signal processing with software defined radios**. 2020. 53 f. Dissertação (Master of Science) — TECHNICAL UNIVERSITY OF CRETE, Greece, 2020.

CISCO. Antenna patterns and their meaning. **Cisco Systems**, v. 1, p. 17, 2007.

CISCO. **THE YAGI-UDA ANTENNA: AN ILLUSTRATED PRIMER**. DuoLabs, 2019. Acesso em 4 Nov. 2022. Disponível em: <<https://duo.com/labs/tech-notes/the-yagi-uda-antenna-an-illustrated-primer>>.

CLEGG, Z. L. J. B. . K. S. I. H. B. A prototype virginia ground station network. **ODU Digital Commons**, v. 1, p. 22, 2020.

Daniel Esteves. **Gr-Satellitess**. Daniel Esteves, 2020. Acesso em 6 Dez. 2022. Disponível em: <<https://gr-satellites.readthedocs.io/en/latest/>>.

ELETRONICNOTES. **RF Mixing / Multiplication: Frequency Mixers**. EletronicNotes, 2011. Acesso em 5 Jan. 2023. Disponível em: <<https://www.electronics-notes.com/articles/radio/rf-mixer/rf-mixing-basics.php>>.

ELFVELIN, M. **Design and Development of the Space Campus Ground Station for Small Satellites**. 2022. 60 f. Dissertação (Space Engineering master) — Luleå University of Technology, Sweden, 2022.

EMBEDDED. **DSP Tricks: Frequency demodulation algorithms**. Embedded, 2011. Acesso em 8 Dez. 2022. Disponível em: <<https://www.embedded.com/dsp-tricks-frequency-demodulationalgorithms/>>.

ENGBERG, M. C. **Satellite communication simulator for Cubsats**. 2017. 64 f. Dissertação (Master of Science) — School of Information and Communication Technology, KTH, 2017.

ESTEVEZ, D. **Decoding FloripaSat-1**. Destevez.net, 2019. Acesso em 6 Mai. 2023. Disponível em: <<https://destevez.net/2019/12/decoding-floripasat-1/>>.

EveRF. **Antenna Parameters**: Fbr. EveRf.com, 2022. Acesso em 4 Oct. 2022. Disponível em: <[EverythingRF.com](http://EverythingRF.com)>.

GNU. **GNU Radio Wiki**. GNU Radio, 2020. Acesso em 6 Dez. 2022. Disponível em: <[https://wiki.gnuradio.org/index.php/Main\\_Page](https://wiki.gnuradio.org/index.php/Main_Page)>.

GOMSPACE. **CubeSat Space Protocol (CSP)**. GOMspace.com, 2011. v. 1, 6 p. Acesso em 19 mai. 2023.

GomSpace. **NanoComAX100 Datasheet**. GomSpace Aps, 2020. Acesso em 6 Dez. 2022. Disponível em: <[www.gomspace.com](http://www.gomspace.com)>.

GOVINDAIAH, P. K. Design and development of gaussian minimum shift keying (gmsk) demodulator for satellite communication. **Bonfring International Journal of Research in Communication Engineering**, v. 2, p. 6, 2012.

GRS. **floripasat grs**. github.com, 2023. Acesso em 6 Jun. 2023. Disponível em: <<https://github.com/floripasat/grs/tree/master/samples>>.

GUIMARAES, D. A. Contributions to the understanding of the msk modulation. **INATEL**, v. 1, p. 10, 2006.

HALE, D. Recursive gaussian filters. **CWP**, v. 1, p. 10, 2019.

Hashiro. **Compensating for Doppler Shift When Contacting the ISS**: Space communications. Ron Hashiro, 2015. Acesso em 12 Oct. 2022. Disponível em: <<https://www.qsl.net.com>>.

HAYKIN, S. **An Introduction to Analog and Digital Communications**. US: Wiley, 1989.

HEALEY, D. An examination of the gamma match. **QST**, v. 1, p. 6, 1969.

HENK. Cross-yagis for terrestrial and satellite work. **qsl.net**, v. 1, p. 3, 2010.

IARU. **IARU VHF Handbook**. 1. ed. Connecticut USA: IARU, 2017. v. 1, p.

IEEE. Ieee standard definitions of terms for antennas. **IEEE Std 145-1993(R2004)**, v. 1, p. 40, 2004.

IOWEGIAN. **Decimation**: Dsp design tools. DSP GURU, 2020. Acesso em 5 Mai. 2023. Disponível em: <<https://dspguru.com>>.

J.MILLER. 9600 baud packet radio modem design. In: ARRL 7TH COMPUTER NETWORKING CONFERENCE, 7., 1988. **Baud Packet...** USA: ARRL, 2012.

JOHNSON, W. A. S. A. G. K. C. R. **Software Receiver Design**. US: Cambridge, 2011.

KARN, M. C. P. **The KISS TNC: A simple Host-to-TNC communications protocol**. github.com, 1986. v. 1, 6 p. Acesso em 13 mai. 2023. Disponível em: <<https://erikarn.github.io/ax25/CNC1987-KISS-TNC-K3MC-KA9Q.pdf>>.

KJSIEIT. **BeliefSat**. Github, 2023. Acesso em 17 Mai. 2023. Disponível em: <<https://github.com/NewLeapKjsieit/BeliefSat>>.

KOCSIS, J. Circularly polarized yagi antennas for satellite communications. **WA9PYH**, v. 1, p. 5, 2011.

KUMAR. **Wireless Communications Fundamental and Advanced Concepts: Design Planning and Applications**. EUA: Aalborg River Publishers, 2015. 790 p.

KUROSE, K. W. R. **What is a Protocol?: A origem das redes de computadores**. UFF, 2021. Acesso em 5 Mai. 2023. Disponível em: <<https://ic.uff.com.br>>.

KYMETA. Link budget calculations for a satellite link with an electronically steerable antenna terminal. **Kymeta Corporation**, v. 1, p. 25, 2019.

LATHI, B. **Modern Digital and Analog Communication Systems**. [S.l.]: Oxford, 2010. 927 p.

LIPNITSKIY, I. Y. **Implementing a Software-Defined Radio Ground Station for Satellite Communications**. 2010. 63 f. Dissertação (Master of Electrical Engineering) — The Pennsylvania State University, USA, 2010.

LodeStar Aerospace. **AlfaCrux**: Amateurradio. UNB, 2022. Acessado em 17 set 2022. Disponível em: <<https://lodestar.aerospace.unb.br/projects/alfacrux/radio>>.

LYONS, R. Quadrature signals: Complex, but not complicated. v. 1, p. 17, 2008.

MALTE, J. **Development of a Versatile Groundstation Utilizing Software Defined Radio**. 2016. 64 f. Dissertação (Master of Science) — NAVAL POSTGRADUATE SCHOOL, USA, 2016.

MATHWORKS. **FIR Gaussian Pulse-Shaping Filter Design**: Continuous-time gaussian filter. MATLAB, 2022. Acesso em 8 Jan. 2023. Disponível em: <<https://www.mathworks.com/help/signal/ug/fir-gaussian-pulse-shaping-filter-design.html>>.

MAZDA. **Telecommunications Engineers Reference Book**. Oxford, UK: Focal Press, 1998. 1216 p.

MENDEZ, B. L. E. **Link Budget for NTNU Satellite**. 2022. 30 f. Dissertação (Monograph) — Norwegian University of Science and Technology, Norway, 2022.

MILIGAN, T. A. **Modern Antenna Design**. Hoboken: Wiley, 2005. 633 p.

MIT. Linear block codes: Encoding and syndrome decoding. **OCW 6.02 DRAFT Lecture Notes**, v. 1, p. 17, 2012.

\_\_\_\_\_. Modulation and demodulation. **MIT 6.02 DRAFT Lecture Notes**, v. 1, p. 23, 2012.

MORELOS-ZARAGOZA, R. H. **The Art of Error Correcting Coding**. US: Wiley, 2006.



MUROTA, H. **GMSK Modulation for Digital Mobile Radio Telephone**. [S.l.]: IEEE, 1981. 238 p.

NASA. Cubesats. In: \_\_\_\_\_. **CubeSat 101 Basic Concepts and Processes for First-Time CubeSat Developers**. USA: NASA, 2017. cap. 1, p. 10–20.

NEXANS. **Coaxial cables**. iandi.sales@nexans.com, 2020. Coaxial Cable Datasheet, 2 p. Acesso em 10 oct. 2022. Disponível em: <file:///C:/Users/User/Desktop/76152.pdf>.

NOGUEIRA, W. S. S. R. S. C. F. M. A. M. V. F. M. V. M. S. Introdução a rádios definidos por software com aplicações em gnu radio. v. 1, p. 50, 2015.

NORDICSEMICONDUCTOR. **Phase samples, I and Q components**. NordicSemiconductor, 2011. Acesso em 5 Jan. 2023. Disponível em: <https://infocenter.nordicsemi.com>.

NOTBLACKMAGIC. **HDLC/AX.25**: Ax.25 frame generator. NBM, 2023. Acesso em 5 Mai. 2023. Disponível em: <https://notblackmagic.com/bitsnpieces/ax.25/>.

PEARCE, J. Ebn0 explained. **Fall 2000 issue of Spread Spectrum Scene Online**, v. 1, p. 4, 2000.

POLY, C. Cubesat design specification rev 14.1 (1u-12u). **SLO**, v. 1, p. 34, 2020.

PORTELA. **Técnicas de recuperação de relógio para sistemas DP-QPSK**. 2012. 123 f. Dissertação (Master of Science) — Universidade de Brasília, BR, 2012.

PRASAD, P. P. Fpga based design and implementation of reed-solomon encoder for error detection and correction. **2015 Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG)**, v. 1, p. 6, 2015.

P.R.C., V. A complete cubesat mission: the floripa-sat experience. **IAA-BR-14-02-07**, v. 1, p. 13, 2013.

PROAKIS, J. G. **Digital Communications**. US: McGrawHill, 2008.

PYTI. Radiation and fundamentals of antennas. v. 1, p. 18, 2021.

Qasim Chaudhari. **Mueller and Muller Timing Synchronization Algorithm**. Wireless Pi, 2020. Acesso em 6 Dez. 2022. Disponível em: <https://wirelesspi.com/mueller-and-muller-timing-synchronization-algorithm/>.

RODOLFO, A. H. L. da S. T. A. **Implementação de uma Arquitetura Reed-Solomon para uso em Redes OTN 10.7 Gbps**. 2007. 67 f. Dissertação (Trabalho de Conclusão de Curso) — Pontifícia Universidade Católica do Rio Grande do Sul, Brasil, 2007.

ROSHNI. **Linear Block Code**: Satellite communication. Electronics Desk, 2020. Acesso em 8 Abril. 2023. Disponível em: <https://electronicsdesk.com/linear-block-code>.

SANTOS Ítalo B. **Implementação em FPGA do módulo corretor fino de frequência e desenvolvimento do simulador de enlace em aritmética de ponto fixo aderentes ao protocolo DVB-S2X**. 2021. 84 f. Dissertação (Master of Science) — Universidade Federal de Brasília, BR, 2021.

SATNOGS. **SatNOGS Database**. db.satnogs.org, 2023. Acesso em 6 Jun. 2023. Disponível em: <https://db.satnogs.org/satellite/LVHV-7932-2909-1988-0093>.

SDRPLAY. **Introduction to the RSPduo Dual tuner 14-bit 1kHz 2GHz SDR**. <http://www.sdrplay.com/>, 2020. 7 p. Acesso em 21 oct. 2022.

SDRplay. **New easy SDRplay set-up for GNU Radio on Windows**: Sdrplay for gnu radio. SDRplay, 2020. Acesso em 6 Dez. 2022. Disponível em: <<https://www.sdrplay.com/new-easy-sdrplay-set-up-for-gnu-radio-on-windows/>>.

SHARMA, T. **RS232 Serial Communication Protocol**: Working and specifications. Circuit Digest, 2021. Acesso em 12 Mai. 2023. Disponível em: <<https://circuitdigest.com/article/RS232-serial-communication>>.

SHIMA, J. M. **Fm Demodulation Using A Digital Radio And Digital Signal Processing**. 1995. 91 f. Dissertação (Master of Science) — University of Florida, USA, 1995.

SHRESTHA, L. K. D. K. M. . R. Vlsi-design and fpga-implementation of gmsk-demodulator architecture using cordic engine for low-power application. **Defence Research and Development Organization (DRDO), Hyderabad, India**, v. 1, p. 6, 2016.

SHURDI, S. C. . B. K. . V. K. . O. The range and horizon plane simulation for ground stations of low earth orbiting (leo) satellites. **Int. J. Communications, Network and System Sciences**, v. 1, n. 1, p. 5, 2011.

SIMONS, M. K. **Bandwidth-Efficient Digital Modulation with Application to Deep-Space Communications**. [S.l.]: Joseph H Youen, 2010. 238 p.

SINGH, M. R. H. A review of design of binary golay code and extended binary golay code for error correction. **International Journal of Trend in Scientific Research and Development (IJTSRD)**, v. 2, p. 5, 2018.

SJODIN, C. J. Sdr implementation for satellite communication. v. 1, p. 10, 2019.

SKAGMO. **NGHam protocol**. Github, 2023. Acesso em 5 Mai. 2023. Disponível em: <<https://github.com/skagmo/ngham>>.

SKLAR. **Digital Communications**. US: Prentice Hall Upper Saddle River, 2001.

SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. The Scientist and Engineer's Guide to Digital Signal Processing, 2011. Acesso em 5 Jan. 2023. Disponível em: <<http://www.dspguide.com/pdfbook.htm>>.

Space Lab UFSC. **FLORIPASAT-1**: Info rádio amador. UFSC, 2022. Acessado em 17 set 2022. Disponível em: <<https://floripasat.ufsc.br/pt/info-radio-amador/>>.

STUTZMAN; THIELE. **Antenna Theory and Design**. Hoboken: Wiley, 2013.

SWALLOW, P. Practical vhf/uhf antennas. In: RSGB (Ed.). **The Radio Communication Handbook**. 6. ed. UK: RSGB, 1995. cap. 19, p. 38.

TANOMARU, D. C. J. **Correção de Erros: Campos Finitos**: Campos finitos ou campos de galois. Teleco, 2021. Acesso em 12 Mai. 2023. Disponível em: <<https://teleco.com.br>>.

TAYLOR, W. A. B. . D. E. N. . J. **AX.25 Link Access Protocol for Amateur Packet Radio**. tapr.org, 1998. v. 2, 143 p. Acesso em 13 mai. 2023. Disponível em: <<https://tapr.org/ax25.2.2.pdf>>.

TELECO. **TV Digital I: Modulação e Propagação**: Teoria da modulação e codificação. 2019. Acesso em 8 Jan. 2023. Disponível em: <<https://teleco.com.br>>.

THOMAS, S. S. M. J. Permittivity and tan delta characteristics of epoxy nanocomposites in the frequency range of 1 mhz1 ghz. **IEEE Transactions on Dielectrics and Electrical Insulation**, v. 1, p. 10, 2008.

TOUCH, I. **A Origem da Comunicação por Redes de Computadores**: A origem das redes de computadores. INFO TOUCH, 2021. Acesso em 5 Mai. 2023. Disponível em: <<https://infotouch.com.br>>.

Tsuruda. **INTRODUCTION TO CUBESAT COMMUNICATION SUBSYSTEM**: Home. Kibo-Cube, 2017. Acesso em 12 abr. 2022. Disponível em: <[https://www.unoosa.org/oosa/en/ourwork/access2space4all/KiboCUBE/KiboCUBE\\_Index.html](https://www.unoosa.org/oosa/en/ourwork/access2space4all/KiboCUBE/KiboCUBE_Index.html)>.

TUTORIALSPPOINT. **Digital Communication - Line Codes**: Digital communication tutorial. Tutorialspoint, 2019. Acesso em 8 Jan. 2023. Disponível em: <<https://tutorialspoint.com>>.

\_\_\_\_\_. **Frequency Shift Keying**: Digital communication tutorial. Tutorialspoint, 2019. Acesso em 8 Jan. 2023. Disponível em: <<https://tutorialspoint.com>>.

Tutorialspoint. **Antenna Theory**: Antenna basics. Tutorialspoint.com, 2022. Acesso em 4 Oct. 2022. Disponível em: <<https://www.tutorialspoint.com/index.htm>>.

VEA, K. D. **NUTS: Ground station with GNU Radio and USRP**. 2012. 41 f. Dissertação (Master of Science in Eletronics) — Norwegian University of Science and Technology, Norway, 2012.

WALLS, A. Samples to digital symbols: Symbol clock recovery and improved symbol synchronization blocks. **SilverBlock Systems**, v. 1, p. 22, 2020.

WALRAVEN, D. Understanding swr by example. **QST ARRL**, v. 1, p. 5, 2006.

WILSON, M. J. **SDRplay RSPdx Software-Defined Receiver**. va2pv@arll.net, 2020. NCAR TECHNICAL NOTE: NCAR/TN-475+STR, 4 p. Acesso em 21 set. 2022. Disponível em: <<https://www.sdrplay.com/docs/RSPdxARRLmeasurements.pdf>>.

WILSON, S. **Digital Modulation and Coding**. US: Artech House, 2006.

WONG, W. **Whats the Difference Between NRZ, NRZI, and Manchester Encoding**. ElectronicDesign, 2017. Acesso em 8 Jan. 2023. Disponível em: <<https://eletronicdesign.com>>.

WORTMAN, W. R. A revision to section 3.81. of low band dxing. **N6MW**, v. 1, p. 6, 2015.

WYGLINSKI, A. M.; PU, D. **Digital Communication Systems Engineering with Software-Defined Radio**. Boston: Artech House, 2013. 375 p.

XIONG, F. **Digital Modulation Techniques**. US: Artech House, 2006.

NUP: 23081.090334/2023-12

Prioridade: Normal

**Ato de emissão de documentos de banca de graduação**

125.322 - Bancas examinadoras de TCC: indicação e atuação

**COMPONENTE**

| Ordem | Descrição        | Nome do arquivo                    |
|-------|------------------|------------------------------------|
| 6     | Versão final TCC | TCC_ESTAÇÃO_TERRENA_JOSÉ_ZANON.pdf |

**Assinaturas**

22/08/2023 08:25:34

FERNANDO CÉSAR COMPARI DE CASTRO (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))  
07.38.00.00.0.0 - DEPARTAMENTO DE ELETRÔNICA E COMPUTAÇÃO - DELC

22/08/2023 10:09:55

NATANAEL RODRIGUES GOMES (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))  
07.09.14.00.0.0 - CURSO DE ENGENHARIA EM TELECOMUNICAÇÕES - CETEL

23/08/2023 13:53:43

EDUARDO ESCOBAR BÜRGER (PROFESSOR DO MAGISTÉRIO SUPERIOR (Ativo))  
07.35.00.00.0.0 - DEPARTAMENTO DE ENGENHARIA MECÂNICA - DEM



Código Verificador: 3186925

Código CRC: 7c1cb7bb

Consulte em: <https://portal.ufsm.br/documentos/publico/autenticacao/assinaturas.html>

