

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SANTA CATARINA
CAMPUS SÃO JOSÉ

ARTHUR CADORE MATUELLA BARCELLA

**SIMULAÇÃO DE TRANSMISSOR E RECEPTOR PTT-A3 PARA
APLICAÇÕES NO SISTEMA DE SATÉLITES ARGOS-3**

SÃO JOSÉ

2025

Arthur Cadore Matuella Barcella

Simulação de Transmissor e Receptor PTT-A3 para Aplicações no
Sistema de Satélites ARGOS-3

Monografia apresentada ao Curso de Engenharia de Telecomunicações do Instituto Federal de Santa Catarina, para a obtenção do título de bacharel em Engenharia de Telecomunicações.

Área de concentração: Telecomunicações

Orientador: Prof. Roberto Wanderley da Nóbrega, Dr.

Coorientador: Prof. Richard Demo Souza,
Dr.

São José

2025

Arthur Cadore Matuella Barcella

Simulação de Transmissor e Receptor PTT-A3 para Aplicações no
Sistema de Satélites ARGOS-3

Monografia apresentada ao Curso de Engenharia de Telecomunicações do Instituto Federal de Santa Catarina, para a obtenção do título de bacharel em Engenharia de Telecomunicações.

São José, 16 de abril de 2025.

Prof. Roberto Wanderley da Nóbrega, Dr.
Instituto Federal de Santa Catarina

Professor Fulano, Dr.
Instituto Federal de Santa Catarina

Professora Fulana, Dra.
Instituto Federal de Santa Catarina

A minha família, por todo o apoio, amor e incentivo para que eu pudesse chegar até aqui, se não fosse por vocês, este trabalho não seria possível.

AGRADECIMENTOS

Agradeço à minha família, minha mãe Vanderleia Salete Cadore e também Gilmar Drehmer, por todo o apoio, pela paciência, pelos ensinamentos mesmo quando não queria aceitá-los, por acreditar em mim, por me incentivar a sempre buscar mais, e me inspirar sempre nas referências.

Sou grato aos meus professores, por todo o suporte, ensinamentos, e por me ajudarem a crescer como pessoa, academicamente e profissionalmente. Agradeço especialmente ao meu orientador Prof. Robertinho, por toda a paciência, ensinamentos, e por acreditar no meu potencial para que eu pudesse chegar até aqui.

Agradeço aos meus amigos, especialmente ao Gabriel Luiz Espindola Pedro, um irmão, que está ao meu lado desde o início dessa maravilhosa jornada pelo técnico de Telecomunicações e agora na graduação em Engenharia de Telecomunicações, por todo apoio nos momentos fáceis e difíceis, por todas as conversas, risadas, e por me ajudar a crescer como pessoa.

Por fim, agradeço a todas as demais pessoas com que tive algum contato durante esse período, que eu considero como amigos e que me consideram como amigo, sempre que precisarem de mim, estarei aqui para ajudar em tudo o que for possível.

“A imaginação é mais importante que o conhecimento. O conhecimento é limitado. A imaginação circunda o mundo.” (Einstein, 1931)

RESUMO

O presente trabalho tem como objetivo o desenvolvimento e simulação de um sistema de modulação e demodulação, utilizando os padrões da terceira geração do sistema de satélites ARGOS-3. Este padrão é utilizado em Plataformas de Coleta de Dados (PCDs) voltadas ao monitoramento ambiental, e emprega técnicas de comunicação digital, como modulação QPSK, codificação convolucional e embaralhamento de dados, com o objetivo de aumentar a confiabilidade do enlace de satélite. A metodologia adotada envolve o estudo de especificações do padrão de comunicação, a estrutura dos datagramas transmitidos pelas PCDs e a implementação, em ambiente simulado, dos blocos responsáveis pela transmissão e recepção digital. O conjunto de simulações vai desde a geração da portadora pura, passando pela palavra de sincronismo e codificação da mensagem do usuário, até a demodulação e recuperação dos dados transmitidos.

Palavras-chave: Comunicação por satélite; PTT-A3; ARGOS-3; Modulação digital;

ABSTRACT

The objective of this work is to develop and simulate a modulation and demodulation system based on the standards of the third generation of the ARGOS-3 satellite system. This standard is used in Data Collection Platforms (DCPs) designed for environmental monitoring and employs modern digital communication techniques such as QPSK modulation, convolutional coding, and data scrambling, aiming to increase the robustness of the satellite communication link. The adopted methodology involves a detailed study of the communication standard specifications, the structure of the datagrams transmitted by the DCPs, and the implementation, in a simulated environment, of the blocks responsible for digital transmission and reception. The proposed simulation set covers the entire chain, from the generation of the continuous wave (CW) carrier, through the synchronization word and user message encoding, to the demodulation and recovery of the transmitted data.

Keywords: Satellite Communication; PTT-A3; ARGOS-3; Digital modulation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Mapa das PCDs distribuídas pelo território nacional	20
Figura 2 – Satélites para coleta de dados ambientais	22
Figura 3 – Estrutura do quadro de transmissão ARGOS-3	24
Figura 4 – Comparação de portadora pura e sinal modulado	25
Figura 5 – Estrutura do datagrama ARGOS-3	26
Figura 6 – Agrupamento de sensores no payload do datagrama	28
Figura 7 – Diagrama de blocos do transmissor ARGOS-3	30
Figura 8 – Diagrama de blocos do codificador convolucional ARGOS-3	31
Figura 9 – Embaralhador de dados para o ARGOS-3	32
Figura 10 – Codificação de linha dos vetores I e Q	33
Figura 11 – Modulação de pulso dos canais I e Q	34
Figura 12 – Comparação entre modulação de pulso <i>Root Raised Cosine</i> (RRC) e <i>Rectangular Function</i> (Rect)	35
Figura 13 – Banda consumida nos canais I e Q	36
Figura 14 – Diagrama de blocos do modulador QPSK	36
Figura 15 – Modulação em banda passante dos canais I e Q	37
Figura 16 – Diagrama de blocos do receptor ARGOS-3	38
Figura 17 – Componentes I e Q após demodulação	39
Figura 18 – Polos e zeros do filtro Passa Baixa	41
Figura 19 – Componentes I Q após filtragem passa baixa	41
Figura 20 – Componentes I e Q após filtragem casada	42
Figura 21 – Amostragem das componentes I e Q	43
Figura 22 – Comparação dos vetores I e Q decididos com vetores transmitidos	44
Figura 23 – Diagrama de blocos do Desembaralhador ARGOS-3	45
Figura 24 – Comparação de BER vs E_b/N_0 utilizando codificação convolucional	46
Figura 25 – Streambits do datagrama ARGOS-3	47
Figura 26 – Codificação convolucional do datagrama ARGOS-3	48
Figura 27 – Multiplexação com preâmbulo	48
Figura 28 – Codificação de linha	49
Figura 29 – Diagrama de blocos do modulador de pulso para os canais I e Q	50
Figura 30 – Resposta ao impulso - Pulso RRC	50
Figura 31 – Resposta ao impulso - Pulso Manchester	51
Figura 32 – Modulação de pulso dos canais I e Q	52
Figura 33 – Modulação em banda passante canais I e Q	53
Figura 34 – Fase e Constelação do sinal modulado $s(t)$	53
Figura 35 – Comparação de portadora pura e sinal modulado	54

Figura 36 – Diagrama de blocos do canal	55
Figura 37 – Adição de múltiplas transmissões no canal	55
Figura 38 – Adição de ruído ao canal	56
Figura 39 – Diagrama de blocos da detecção de portadora	57
Figura 40 – Diagrama de waterfall do canal	57
Figura 41 – Detecção de componentes no espectro com base em P_t	58
Figura 42 – Diagrama de waterfall de detecção do canal	59
Figura 43 – Diagrama de waterfall de decisão do canal	60
Figura 44 – Demodulação banda base dos canais I e Q	62
Figura 45 – Filtragem passa baixa dos canais I e Q	63
Figura 46 – Filtragem casada dos canais I e Q	64
Figura 47 – Diagrama de blocos para montagem do vetor de sincronismo	65
Figura 48 – Vetor de amostras esperadas para sincronismo com I e Q	65
Figura 49 – Módulo de correlação entre I e Q e vetor de sincronismo	66
Figura 50 – Sincronização de I e Q com vetor de sincronismo	66
Figura 51 – Amostragem dos canais I e Q em T_b a partir de τ	67
Figura 52 – Constelação dos canais I e Q filtrados e amostrados	68
Figura 53 – Desembaralhamento dos canais I e Q	69
Figura 54 – Decodificação convolucional dos canais I e Q	70
Figura 55 – Comparação BER vs SNR - ARGOS3 e QPSK	75

LISTA DE QUADROS

Quadro 1 – Dados meteorológicos da PCD 31855 (10/10/2007 - 11/10/2007)	27
Quadro 2 – Comprimento em bits para cada tamanho de mensagem (T_m)	28
Quadro 3 – Comprimento da cauda (bits '0') para cada tamanho de mensagem	29

LISTA DE ABREVIATURAS E SIGLAS

AD9361 *Analog Devices 9361 - RF Transceptor.*

ARGOS *Advanced Research and Global Observation Satellite.*

ARGOS-2 *ARGOS v2 Communication Standard.*

ARGOS-3 *ARGOS v3 Communication Standard.*

AWGN *Additive White Gaussian Noise.*

CBERS *China-Brazil Earth Resources Satellites.*

CBERS-1 *China-Brazil Earth Resources Satellite 1.*

CBERS-4 *China-Brazil Earth Resources Satellite 4.*

CCSDS *Consultative Committee for Space Data Systems.*

CNES *Centre National d'Études Spatiales, França.*

CRC *Cyclic Redundancy Check.*

ETR *Estações Terrenas de Recepção.*

FFT *Fast Fourier Transform.*

FPGA *Field Programmable Gate Array.*

Hamming Distance *Hamming Distance.*

INPE *Instituto Nacional de Pesquisas Espaciais.*

IoT *Internet of Things.*

ISI *Inter-Symbol Interference.*

LPF *Low-Pass Filter.*

Manchester *Manchester Encoding.*

MF *Matched Filter.*

MLD *Maximum Likelihood Decoding.*

NASA *National Aeronautics and Space Administration.*

NOAA *National Oceanic and Atmospheric Administration.*

NRZ *Non-Return-to-Zero.*

PCD Plataformas de coleta de dado.

Plano IQ *In-phase and Quadrature plane.*

PM *Phase Modulation.*

PTT-A2 *ARGOS v2 Communication Standard PTT.*

PTT-A3 *ARGOS v3 Communication Standard PTT.*

PTT-ZE variação do PTT-A3 *Zero-Energy.*

QPSK *Quadrature Phase-Shift Keying.*

Rect *Rectangular Function.*

RRC *Root Raised Cosine.*

SBCDA Sistema Brasileiro de Coleta de Dados Ambientais.

SCD-1 Satélite de Coleta de Dados 1.

SCD-2 Satélite de Coleta de Dados 2.

SDR *Software Defined Radio.*

SINDA Sistema Integrado de Dados Ambientais.

SNR *Signal-to-Noise Ratio.*

Zynq-7000 família de FPGA arquitetura ARM.

LISTA DE SÍMBOLOS

A Amplitude do sinal modulado.

α Fator de roll-off do filtro RRC, $\alpha = 0.8$.

BER Bit Error Rate.

B_m Sequência de bits do comprimento de mensagem, $(0 \leq T_m < 8)_{10}$ (ARGOS-3).

bps Bits per second.

I In-phase component.

Q Quadrature component.

dB Decibel.

d_{free} Distância livre do código convolucional, $d_{free} = 10$ (ARGOS-3).

$d_I(t)$ Sinal modulado em banda base, canal *I*.

$d'_I(t)$ Sinal filtrado (filtragem passa baixa), canal *I*.

$d_Q(t)$ Sinal modulado em banda base, canal *Q*.

$d'_Q(t)$ Sinal filtrado (filtragem passa baixa), canal *Q*.

$d'_X(t)$ Sinal demodulado em banda base, canal *I*.

$d'_Y(t)$ Sinal demodulado em banda base, canal *Q*.

E_b/N_0 Energy per bit to noise power spectral density ratio.

E_m Cauda do datagrama, entre sete e nove bits '0' (ARGOS-3).

f_c Carrier frequency, ≈ 401.65 MHz (ARGOS-3).

f'_c Frequênciada portadora detectada.

f_n f_c randômica selecionada pela PCD, $401.620 \leq f_n \leq 401.650$ MHz (ARGOS-3).

f_s Frequênciade amostragem, $f_s = 128$ KHz.

G_0 Vetor gerador correspondente ao, canal *I*, $G_0 = [121]_{10}$ (ARGOS-3).

G_1 Vetor gerador correspondente ao, canal *Q*, $G_1 = [91]_{10}$ (ARGOS-3).

$g(t)$ Resposta ao impulso do filtro de formatação.

$g(-t)$ Resposta ao impulso do filtro de formatação invertido (filtro casado).

$H(f)$ Resposta em frequência do filtro passa baixa.

$H(s)$ Função de transferência do filtro passa baixa.

$h(t)$ Resposta ao impulso do filtro passa baixa.

$H(z)$ Função de transferência discreta do filtro passa baixa.

Hz *Hertz*.

$I[n]$ Sequência de símbolos, canal I .

$I'[n]$ Sequência de símbolos decididos, canal I .

I_{PCD} Número da PCD, $(0 \leq I_{PCD} < 1048575)_{10}$ (ARGOS-3).

$I'(t)$ Sinal filtrado (filtragem casada), canal I .

k Índice de frequência do espectro de frequência do segmento de sinal.

kHz *Kilohertz*.

k_{max} Índice de amostra de maior correlação com o preâmbulo.

L_{sync} Tamanho do preâmbulo em bits, $L_{sync} = 15$ (ARGOS-3).

m Índice de segmento de sinal.

MHz *Megahertz*.

n Índice de amostra/símbolo.

N_b Número de blocos, $0 < N_b < 9$ (ARGOS-3).

N_c Número de detecções minima para ser considerada como "confirmada", $N_c = 4$.

PCD_{ID} Identificador da PCD, número da PCD + bits de checksum.

P_d Duração do prefixo de símbolos para portadora pura, $P_d = f_s * \approx 0.082$ (ARGOS-3).

ϕ_0 Desvio de fase inicial do sinal modulado.

P_m Bit de paridade do campo B_m .

$P_n[k]$ Potência do espectro de frequência do segmento de sinal.

P_t Valor de potência de *threshold* para detecção de portadora.

$Q[n]$ Sequência de símbolos, canal Q .

$Q'[n]$ Sequência de símbolos decididos, canal Q .

$Q'(t)$ Sinal filtrado (filtragem casada), canal Q .

R_b Rate bit, 400 bps (ARGOS-3).

R_{PCD} Bits de checksum do Identificador da PCD.

$r(t)$ Ruído $AWGN$, em função de tempo t .

σ^2 Variância do ruído adicionado ao sinal modulado.

$S_I[n]$ Sequência de preâmbulo do, canal I .

$S[n]$ Sequência de preâmbulo, $S[n] = 2BEEEEBF_{16}$, (ARGOS-3).

$S_Q[n]$ Sequência de preâmbulo do, canal Q .

$s(t)$ Sinal modulado em banda passante, em função de tempo t .

$s'(t)$ Segmento de sinal modulado em banda passante, em função de tempo t .

t Vetor de Tempo.

τ Delay inicial para decisão de símbolos.

T_b Tempo de bit, $T_b = 1/R_b$.

T_m Tamanho da mensagem.

n Index do segmento de sinal.

u_t Vetor de entrada do codificador convolucional.

u'_t Vetor de saída do decodificador convolucional.

$v_t^{(0)}$ Saída do codificador convolucional correspondente a G_0 .

$v_t^{(0)'} \quad$ Sequência de bits desembaralhados, canal I .

$v_t^{(1)}$ Saída do codificador convolucional correspondente a G_1 .

$v_t^{(1)'} \quad$ Sequência de bits desembaralhados, canal Q .

W Largura de banda do sinal modulado, definido como $W = (1 + \alpha)/T_b$.

$x[k, m]$ Matriz de componentes detectadas, $x[k, m]$.

$x_i(t)$ Portadora cossenoidal utilizada na modulação do, canal I .

$X[n]$ Sequência de bits embaralhados, canal I .

$x_n[m]$ Segmento de sinal de $s(t)$ amostrado no tempo.

$X_n[k]$ Espectro de frequência do segmento de sinal.

$X'[n]$ Sequência de bits decididos após decodificação de linha, canal I .

$Y[n]$ Sequência de bits embaralhados, canal Q .

$Y'[n]$ Sequência de bits decididos após decodificação de linha, canal Q .

$y_q(t)$ Portadora senoidal utilizada na modulação do, canal Q .

SUMÁRIO

1	INTRODUÇÃO	19
1.1	OBJETIVOS	21
1.1.1	Objetivo geral	21
1.1.2	Objetivos específicos	21
1.2	ORGANIZAÇÃO DE TEXTO	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	COLETA DE DADOS SBCDA VIA SATÉLITE	22
2.1.1	Constelação Catarina	23
2.2	EVOLUÇÃO DO SISTEMA ARGOS	23
2.3	ESPECIFICAÇÕES DO PADRÃO PTT-A3	24
2.3.1	Portadora pura	24
2.3.2	Palavra de sincronismo (preâmbulo)	25
2.3.3	Datagrama	26
2.4	ESTRUTURA DE UM DATAGRAMA ARGOS-3	26
2.4.1	Dados de aplicação	27
2.4.1.1	<i>Sensores</i>	27
2.4.1.2	<i>Blocos de dados</i>	27
2.4.2	Tamanho de mensagem	28
2.4.3	Identificador da PCD	29
2.4.4	Bits de fim de mensagem	29
2.5	TRANSMISSOR PTT-A3	29
2.5.1	Codificador convolucional	30
2.5.2	Embaralhador	31
2.5.3	Codificação de Linha	32
2.5.4	Modulação de Pulso	33
2.5.5	Modulação em banda passante	36
2.6	RECEPTOR PTT-A3	37
2.6.1	Detecção de portadora	38
2.6.2	Demodulador banda base	39
2.6.3	Filtragem passa baixa	40
2.6.4	Filtragem casada	42
2.6.5	Decisão de símbolos	42
2.6.6	Decodificador de linha	44
2.6.7	Desembaralhador	44
2.6.8	Decodificador convolucional	45

3	DESENVOLVIMENTO E RESULTADOS	47
3.1	CADEIA DE TRANSMISSÃO	47
3.1.1	Sequência de transmissão	47
3.1.2	Modulação de pulso RRC e Manchester	49
3.1.2.1	<i>Pulso RRC e Manchester</i>	50
3.1.2.2	<i>Modulação de pulso dos canais I e Q</i>	51
3.1.3	Modulação em fase em quadratura (QPSK)	52
3.1.3.1	<i>Adição de portadora pura</i>	54
3.2	CANAL E ADIÇÃO DE RUÍDO	54
3.2.1	Modelo de canal	55
3.2.2	Geração de ruído AWGN	56
3.3	DETECÇÃO DE PORTADORA	56
3.3.1	Segmentação do sinal recebido	57
3.3.2	Detecção de componentes no espectro	58
3.3.3	Decisão de componentes detectadas	59
3.3.4	Segmentação do sinal recebido para recepção	60
3.4	CADEIA DE RECEPÇÃO	61
3.4.1	Demodulação banda base	61
3.4.2	Filtragem	62
3.4.3	Sincronização de símbolos	64
3.4.4	Decisão de símbolos	67
3.4.5	Recuperação do Datagrama	68
4	CONCLUSÃO	71
4.0.1	Trabalhos futuros	71
	Referências	73
	APÊNDICE A – BER VS SNR	75
	APÊNDICE B – SIMULADOR OPENSOURCE ARGOS-3	76

1 INTRODUÇÃO

O uso de tecnologias de comunicação via satélite tem se mostrado fundamental para aplicações de *Internet of Things* (IoT) em cenários remotos, especialmente no monitoramento ambiental, oceânico e meteorológico, através de plataformas equipadas com sensores, as chamadas Plataformas de coleta de dado (PCD), que são capazes de transmitir pequenas quantidades de dados periodicamente para satélites de órbita baixa (Centenaro *et al.*, 2021; Fraire; Céspedes; Accettura, 2019).

Dentre os sistemas internacionais voltados para esse tipo de aplicação, destaca-se o *Advanced Research and Global Observation Satellite* (ARGOS)¹, criado em 1978 por meio de uma cooperação entre o *Centre National d'Études Spatiales*, França (CNES), *National Oceanic and Atmospheric Administration* (NOAA) e *National Aeronautics and Space Administration* (NASA). Desde seu desenvolvimento, o sistema ARGOS tem passado por constantes evoluções, trazendo novos padrões de transmissão e modulação como os formatos *ARGOS v2 Communication Standard PTT* (PTT-A2), *ARGOS v3 Communication Standard PTT* (PTT-A3) e variação do PTT-A3 *Zero-Energy* (PTT-ZE), para aprimorar a eficiência espectral e a confiabilidade do enlace com o satélite.

No Brasil, o Sistema Brasileiro de Coleta de Dados Ambientais (SBCDA)², desenvolvido e operado pelo Instituto Nacional de Pesquisas Espaciais (INPE), emprega a tecnologia ARGOS. O SBCDA utiliza satélites em órbita baixa, como os Satélite de Coleta de Dados 1 (SCD-1), Satélite de Coleta de Dados 2 (SCD-2) e os satélites da série *China-Brazil Earth Resources Satellites* (CBERS), para coletar dados transmitidos por PCD instaladas em todo o território nacional (Rodrigues, 2018; Duarte *et al.*, 2021).

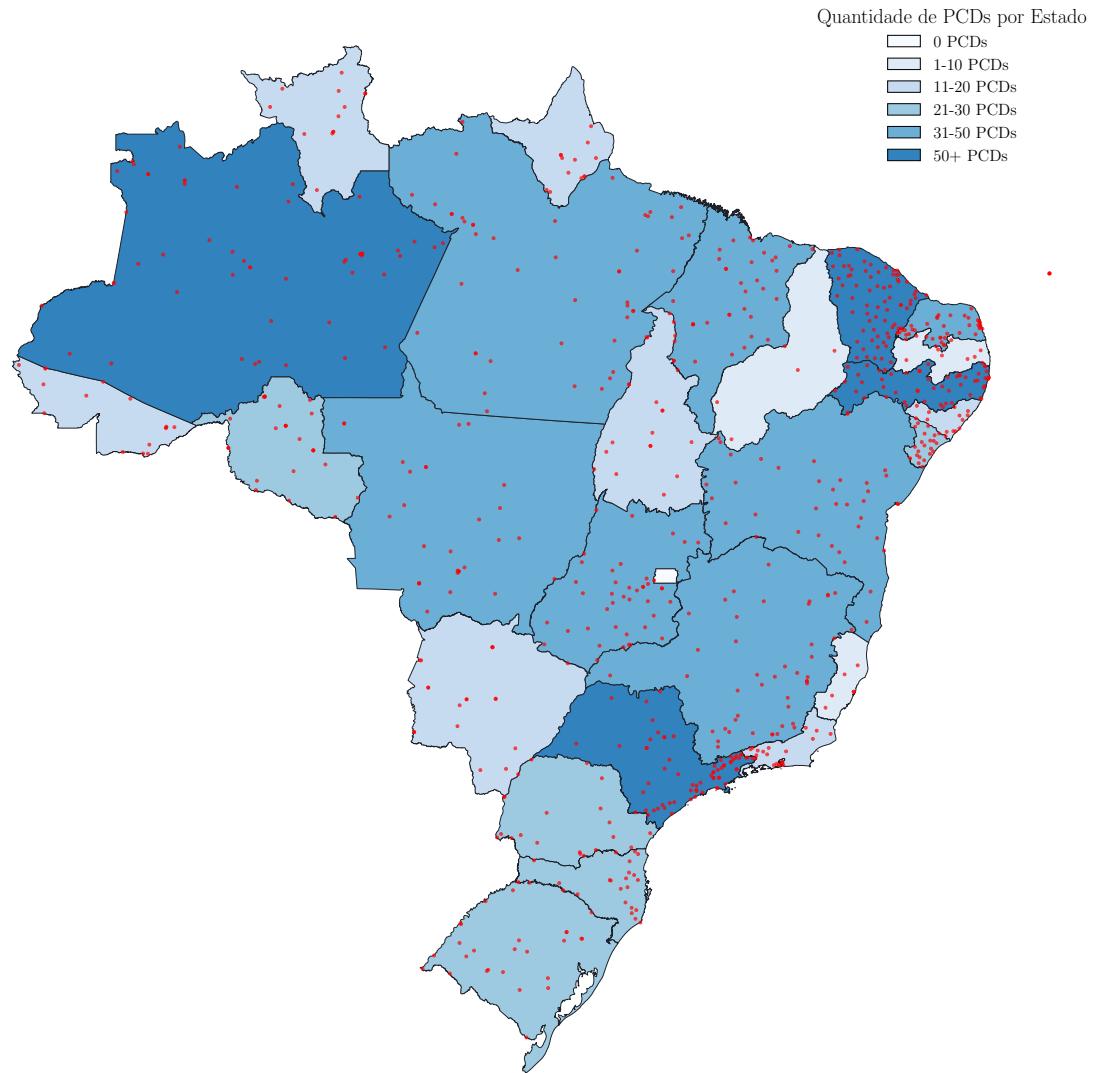
De acordo com dados do Sistema Integrado de Dados Ambientais (SINDA)³, atualmente é possível realizar a inspeção dos dados de 1030 pontos ativos de coleta distribuídos pelo país, cuja distribuição geográfica é apresentada na Figura 1 (Silva; Almeida, 2022).

¹ <https://www.argos-system.org/>

² <https://www.gov.br/aeb/pt-br/acoes-e-programas/aplicacoes-espaciais/dados-ambientais>

³ <http://sinda.crn.inpe.br/PCD/>

Figura 1 – Mapa das PCDs distribuídas pelo território nacional



Grande parte dessas PCD é compatível com o padrão *ARGOS v2 Communication Standard* (ARGOS-2), implantado a partir de 1993. Esses dispositivos, entretanto, operam com tecnologias de hardware e software limitadas à época de sua instalação, apresentando restrições quanto à eficiência espectral, robustez contra interferências e capacidade de transmissão de dados. Diante da evolução das demandas de comunicação, da necessidade de aprimorar o desempenho do sistema e da dificuldade na aquisição dos componentes legados para montagem do hardware compatível com ARGOS-2, o *ARGOS v3 Communication Standard* (ARGOS-3) foi desenvolvido.

O sistema ARGOS-3 introduziu o uso de transmissores do tipo PTT-A3 que empregam técnicas digitais mais avançadas, como modulação *Quadrature Phase-Shift Keying* (QPSK), codificação convolucional e embaralhamento de dados, visando maior robustez frente a desvanecimentos e erros de rajada. Esses avanços também motivaram o desenvolvimento de novos transmissores e receptores compatíveis com o padrão PTT-A3 (CNES,

2006).

Neste contexto, o presente trabalho tem como objetivo a simulação de um modulador/demodulador compatível com o padrão PTT-A3, conforme as especificações do sistema ARGOS-3, contribuindo para a formação de massa crítica com domínio técnico sobre os elementos do sistema, a fim de criar melhores condições para que futuros esforços de modernização das PCD do SBCDA possam ser conduzidos.

1.1 OBJETIVOS

Este trabalho tem como objetivo principal a simulação e análise de um sistema de modulação e demodulação compatível com o padrão PTT-A3, utilizado no sistema de satélites ARGOS-3.

1.1.1 Objetivo geral

Simular um sistema de modulação e demodulação compatível com o padrão PTT-A3 do sistema de satélites ARGOS-3.

1.1.2 Objetivos específicos

Para atingir o objetivo geral proposto, este trabalho foi estruturado em uma série de etapas com metas técnicas bem definidas. Os objetivos específicos foram organizados de modo a contemplar a compreensão teórico/prática do padrão de comunicação ARGOS-3. São eles:

- Estudar o padrão de comunicação ARGOS;
- Simular a cadeia de transmissão do padrão PTT-A3;
- Simular o efeito da adição de ruído;
- Simular a detecção de múltiplas portadoras;
- Simular a recepção e demodulação do sinal;
- Simular a sincronização de quadros.
- Simular a montagem e interpretação do datagrama.

1.2 ORGANIZAÇÃO DE TEXTO

O conteúdo deste trabalho está organizado da seguinte forma: O Capítulo 2 apresenta a fundamentação teórica necessária para a compreensão do projeto. Já o Capítulo 3 descreve o desenvolvimento e as etapas na execução do projeto.

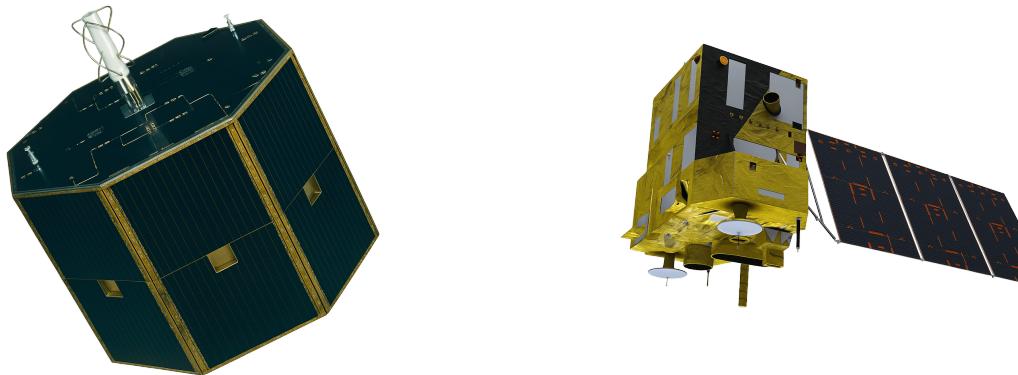
2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta conceitos e fundamentos teóricos para o desenvolvimento deste trabalho. São apresentados tópicos relacionados a comunicação por satélites, evolução do sistema ARGOS, e técnicas de modulação, codificação e sincronização envolvidas no padrão PTT-A3. O objetivo é fornecer uma base de conhecimento para a compreensão dos requisitos técnicos e operacionais do sistema de comunicação proposto.

2.1 COLETA DE DADOS SBCDA VIA SATÉLITE

A comunicação por satélites desempenha um papel fundamental na coleta e disseminação de dados ambientais em escala regional e global. No contexto brasileiro, essa função é desempenhada pelo SBCDA, operado pelo INPE. O SBCDA é composto pelos satélites SCD-1, SCD-2 e *China-Brazil Earth Resources Satellite 1* (CBERS-1) até *China-Brazil Earth Resources Satellite 4* (CBERS-4) apresentados na Figura 2 abaixo, que orbitam a aproximadamente 750 km de altitude, recebendo informações transmitidas por PCD espalhadas pelo território nacional (Lima, 2021).

Figura 2 – Satélites para coleta de dados ambientais



Satélite SCD-1

Satélite CBERS-4

Parâmetro	Valor	Parâmetro	Valor
Massa	115 kg	Massa	1 980 kg
Potência Elétrica	110 W	Potência Elétrica	2 300 W
Vida útil	4 anos	Vida útil	3 anos
Altitude média	≈ 750 km	Altitude média	≈ 778 km
Inclinação orbital	25°	Inclinação orbital	98,54°
Período orbital	99,7 min	Período orbital	100,32 min

Esses satélites recebem sinais transmitidos pelas PCD na faixa de frequência UHF (401,62 a 401,65 *MHz*) e os retransmitem para as Estações Terrenas de Recepção (ETR)

localizadas em solo, nas faixas de Banda-S (2267,5 MHz). Como operam em órbitas baixas, esses satélites realizam aproximadamente 14 revoluções por dia sobre o território nacional, o que permite ampla cobertura espacial.

Apesar da ampla cobertura, a comunicação com satélites de órbita baixa apresenta um grande desafio técnico, sendo a necessidade de visada simultânea entre a PCD transmissora e o satélite, o que limita a janela de transmissão e impõe restrições na coleta contínua de dados. Além disso, o movimento relativo entre a PCD e o satélite provoca o chamado efeito Doppler, responsável por deslocamentos na frequência do sinal recebido, podendo atingir até $\pm 79,4\ kHz$. Esse desvio precisa ser compensado para garantir a correta demodulação do sinal (Rae, 2005; Rodrigues, 2018).

A confiabilidade do enlace também é impactada por fatores como atenuação no espaço livre, ruídos térmicos, e variações atmosféricas. Para compensar esses fatores, são necessárias técnicas específicas de modulação, sincronização, codificação de dados e planejamento de enlace, de modo a garantir a confiabilidade das mensagens transmitidas.

2.1.1 Constelação Catarina

A Constelação Catarina é um projeto nacional baseado no uso de nanossatélites em órbita baixa, para atuar como um novo braço operacional do SBCDA. A Constelação Catarina é composta por pequenos satélites integrados com *Software Defined Radio* (SDR), capazes de receber sinais transmitidos pelas PCD no padrão ARGOS-2, com planos futuros de migração para o padrão ARGOS-3 (Gomes, 2024).

Diferentemente dos satélites tradicionais, os nanosatélites da Constelação Catarina são projetados para realizar a decodificação e o armazenamento dos dados a bordo, o que permite superar a limitação de visada simultânea entre satélite e ETR, ampliando a cobertura do sistema (Rodrigues, 2018).

A arquitetura dos satélites que compõem a constelação é baseada na integração do transceptor *Analog Devices 9361 - RF Transceptor* (AD9361) com uma *Field Programmable Gate Array* (FPGA) da família de FPGA arquitetura ARM (Zynq-7000), formando uma plataforma de SDR altamente flexível¹. Essa configuração permite a reconfiguração remota do hardware, o que é especialmente importante para futuras atualizações de protocolo ou migração para novos padrões de comunicação, como o PTT-A3.

2.2 EVOLUÇÃO DO SISTEMA ARGOS

O ARGOS-2, base do SBCDA desde 1993, utiliza transmissores do tipo PTT-A2, baseados em modulação analógica *Phase Modulation* (PM) com codificação Manchester. Essa versão se mostrou eficiente por muitos anos, mas suas limitações logo se tornaram

¹ <https://www.argos-system.org/wp-content/uploads/2023/01/ARTIC-Chipset-AnSem-Info-sheet.pdf>

evidentes, especialmente no que diz respeito à robustez frente a ruído, à largura de banda ocupada e à necessidade de visada simultânea entre PCD e satélite para a ETR (CNES, 2006).

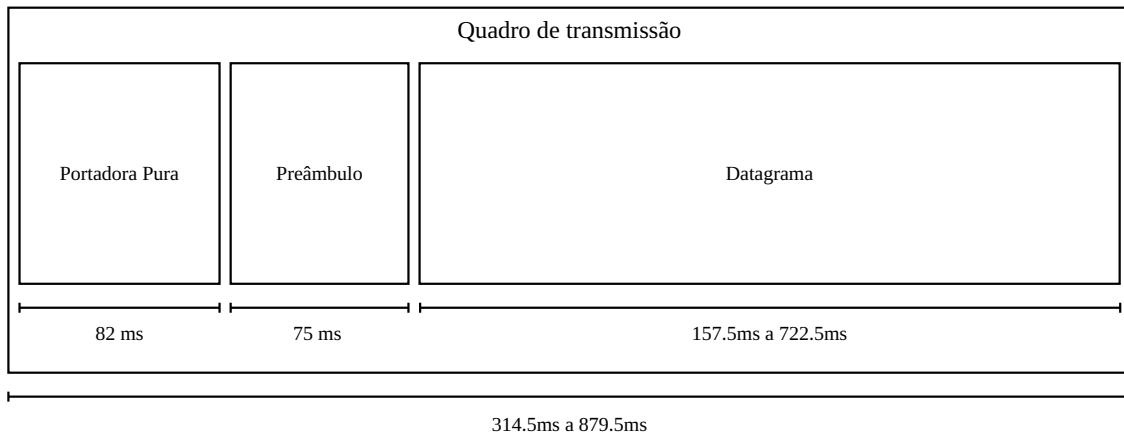
A evolução desse sistema levou ao desenvolvimento do ARGOS-3, que introduziu novas técnicas digitais de comunicação. Essa nova geração incorporou transmissores do tipo PTT-A3 e PTT-ZE, os quais se destacam pela adoção de modulação QPSK, codificação convolucional e embaralhamento de dados, resultando em maior confiabilidade na transmissão e maior eficiência espectral. Além disso, o ARGOS-3 permite o armazenamento e retransmissão de mensagens a bordo do satélite para a ETR (Lima, 2021; Rodrigues, 2018).

2.3 ESPECIFICAÇÕES DO PADRÃO PTT-A3

O transmissor do tipo PTT-A3 é um dos formatos definidos na terceira geração do sistema ARGOS, projetado para oferecer maior robustez na transmissão e maior eficiência na utilização do espectro de frequência.

A estrutura de um quadro PTT-A3 é composta por três campos principais, sendo eles: portadora pura, palavra de sincronismo (préâmbulo) e datagrama. Na Figura 3, a estrutura é apresentada de forma detalhada, considerando que a taxa de transmissão R_b é de 400 bps CNES (2006).

Figura 3 – Estrutura do quadro de transmissão ARGOS-3

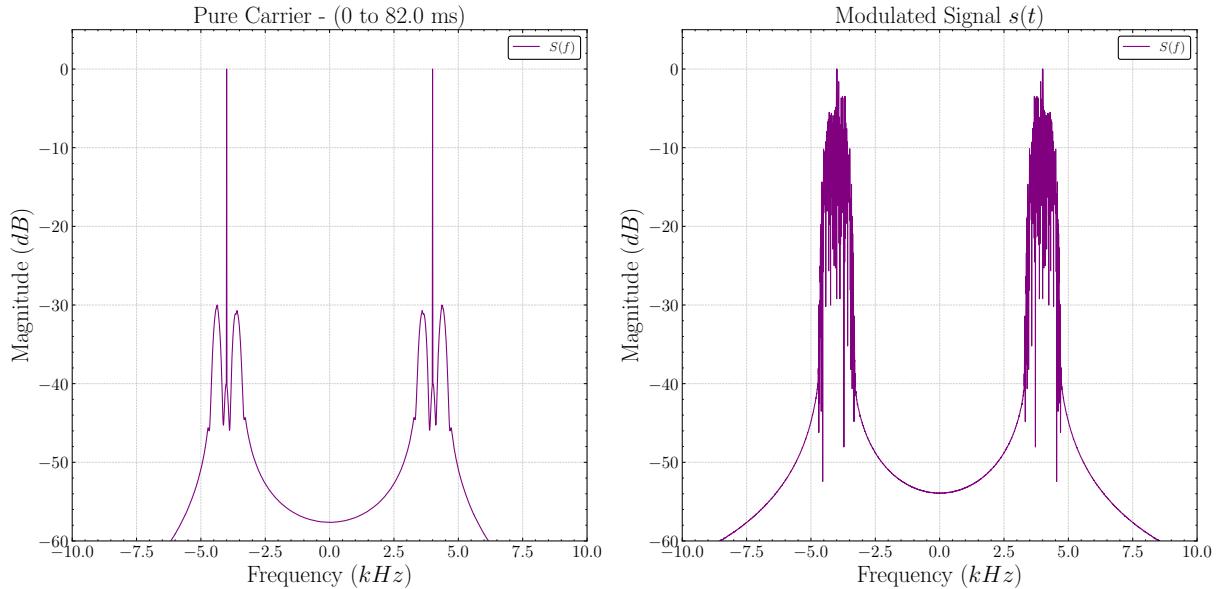


2.3.1 Portadora pura

A sequência de transmissão do quadro inicia-se com a portadora contínua ou pura, com duração de 82 ± 2 ms. Durante essa etapa a portadora não transmite dados modulados e é utilizada pelo receptor apenas para realizar a detecção do sinal, bem como para facilitar o processo de sincronização de frequência e fase da portadora.

A Figura 4 apresenta o sinal da portadora pura no espectro em comparação com o sinal modulado. Nota-se que quando apenas a portadora pura é transmitida, o espectro do sinal é concentrado em uma única frequência, sem componentes laterais. Já o sinal modulado apresenta componentes laterais que se estendem ao redor da frequência da portadora f_c , formando uma banda de uso do espectro mais ampla (CNES, 2006).

Figura 4 – Comparação de portadora pura e sinal modulado



O processo de detecção do sinal realizado pelo receptor monitora a presença de sinal que ultrapassa um determinado limiar (P_t), dessa forma é fundamental que no receptor o sinal esteja o mais concentrado e com a maior *Signal-to-Noise Ratio* (SNR) possível no momento da detecção, para que a frequência da portadora, f_c , seja identificada corretamente (CNES, 2006).

2.3.2 Palavra de sincronismo (preâmbulo)

Logo após a portadora pura, é transmitida uma palavra de sincronismo de 30 bits (correspondente a 15 símbolos QPSK), sendo fundamental para identificação do início da mensagem codificada, possibilitando a sincronização para decisão. Essa sequência é conhecida e fixa entre transmissor e receptor, no caso do PTT-A3 sendo $S = 2\text{BEEEEBF}_{16}$, o que permite alinhar corretamente a decisão e identificar o início do bloco de dados úteis. (CNES, 2006)

A sequência $S[n]$ é separada em dois vetores distintos, $S_I[n]$ e $S_Q[n]$, por meio de uma intercalação simples de seus bits. O processo de intercalação consiste em distribuir os bits de forma alternada entre os canais $S_I[n]$ e $S_Q[n]$, resultando em duas sequências de 15 bits cada, que serão transmitida como preâmbulo (CNES, 2006), usando a sequência padrão do ARGOS-3 podemos representar como

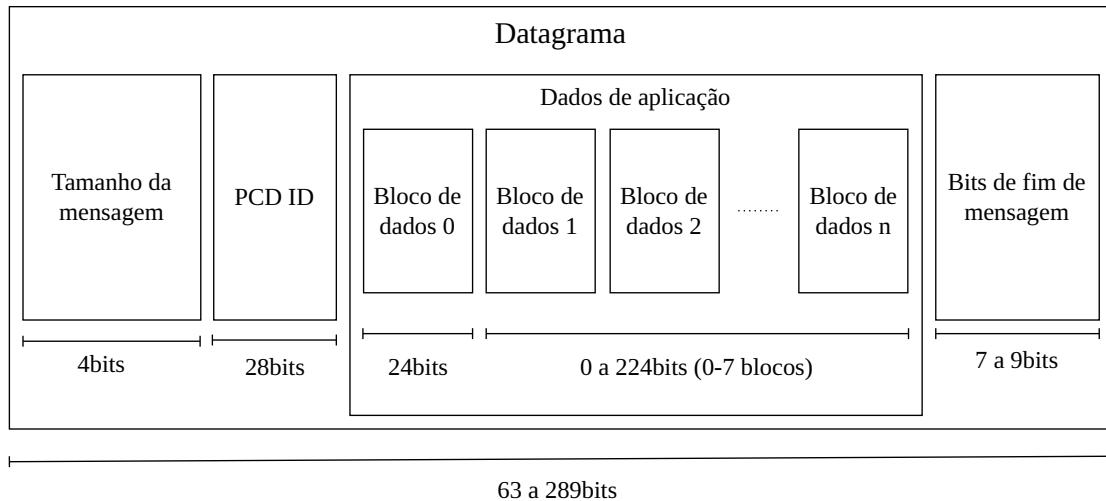
$$\begin{aligned} S_I[n] = [S_0, S_2, S_4, \dots, S_{28}] &\mapsto S_I[n] = [1111, 1111, 1111, 111] \\ S_Q[n] = [S_1, S_3, S_5, \dots, S_{29}] &\mapsto S_Q[n] = [0011, 0101, 0100, 111] \end{aligned} \quad (2.1)$$

Importante destacar que esta palavra não é codificada convolucionalmente ou embaralhada, sendo adicionada ao início do vetor de bits de cada canal após esses blocos. (CNES, 2006).

2.3.3 Datagrama

Após o envio da palavra de sincronismo, inicia-se a transmissão dos dados modulados. Esses dados são organizados segundo a estrutura definida pelo datagrama do padrão ARGOS-3, que contém os campos responsáveis pela identificação da plataforma, carga útil de dados e controle de finalização, conforme apresentado na Figura 5 (CNES, 2006).

Figura 5 – Estrutura do datagrama ARGOS-3



2.4 ESTRUTURA DE UM DATAGRAMA ARGOS-3

O datagrama transmitido no padrão ARGOS-3 possui uma estrutura bem definida, composta por campos de dados do usuário, que carregam as informações provenientes dos sensores da PCD, e por campos de cabeçalho, responsáveis por identificar a estação transmissora e informar o comprimento total da mensagem. Esses campos incluem o identificador da PCD, o número de blocos de dados e um bit de paridade, que auxilia na verificação de integridade da informação. Essa organização permite que o sistema receptor interprete corretamente o conteúdo transmitido e associe os dados recebidos à plataforma correspondente.

2.4.1 Dados de aplicação

A primeira etapa na montagem do datagrama consiste na coleta dos dados de aplicação, isto é, os dados que efetivamente contêm informação dos sensores a serem transmitidos da PCD para o satélite.

2.4.1.1 Sensores

Cada sensor presente nas PCD gera um valor de oito bits correspondente à variável monitorada, possibilitando assim 256 (2^8) níveis de monitoramento para cada sensor. A PCD pode ser equipada com diferentes sensores, de acordo com o cenário de instalação e os parâmetros ambientais de interesse.

Entre os sensores comumente utilizados, destacam-se os de direção do vento ($^\circ$), precipitação (mm), pressão atmosférica (mB), radiação solar acumulada (MJ/m²), temperatura do ar ($^\circ\text{C}$), umidade relativa (%), e velocidade do vento (m/s). Por exemplo, os dados ² coletados da PCD 31855 são mostrados no Quadro 1.

Quadro 1 – Dados meteorológicos da PCD 31855 (10/10/2007 - 11/10/2007)

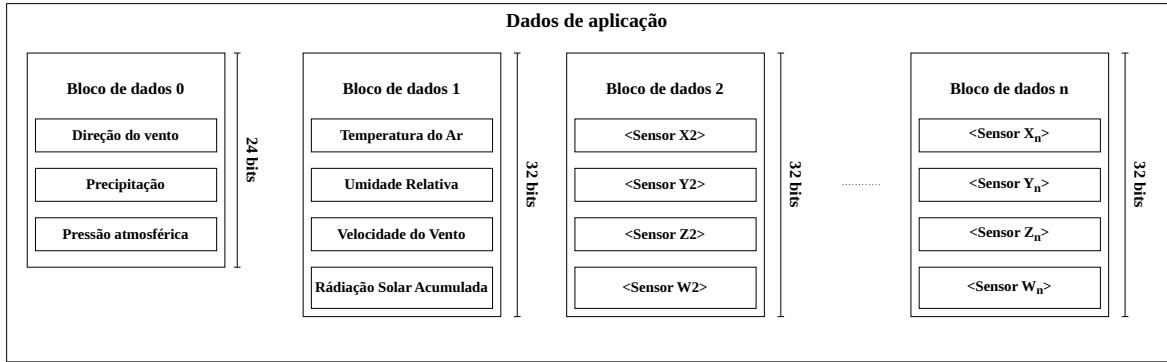
DataHora	DirVento	Precip.	PressãoAtm	RadSolAcum	TempAr	UmidRel	VelVento
2007-11-10 21h	0	0	945.5	2.3	30.8	25.6	0
2007-11-10 18h	0	0	943.8	8.75	36.5	20.8	0
2007-11-10 15h	0	0	947.3	9.72	33.6	28.8	0
2007-11-10 12h	0	0	950.1	4.98	28.0	51.2	0
2007-11-10 09h	0	0	949.3	0.17	20.9	60.8	0
2007-11-10 06h	0	0	947.8	0.00	21.8	52.8	0
2007-11-10 03h	0	0	948.0	0.00	23.5	40.0	0
2007-11-10 00h	0	0	948.4	0.00	21.9	35.2	0
2007-11-09 21h	0	0	946.1	0.48	30.8	20.8	0
2007-11-09 18h	0	0	945.1	9.13	36.3	16.0	0
2007-11-09 15h	0	0	948.4	10.13	34.5	22.4	0
2007-11-09 12h	0	0	950.9	3.44	28.5	46.4	0

2.4.1.2 Blocos de dados

Os dados dos sensores são agrupados em conjuntos denominados ‘blocos de dados’ contendo quatro sensores por bloco, conforme ilustrado na Figura 6. Assim, cada bloco de dados possui 32 bits ($4 \cdot 8$ bits), sendo o valor mínimo de um bloco de dados para montar um datagrama PTT-A3 (exceto o primeiro bloco que possui apenas 24 bits de comprimento). Para o caso específico em que a PCD irá transmitir apenas um bloco, ela poderá abrigar apenas três sensores. Caso haja mais de um bloco, o comprimento é dado por $24 + (N_b - 1) \cdot 32$ bits CNES (2006).

² <http://sinda.crn.inpe.br/PCD/SITE/novo/site/historico/action.php>

Figura 6 – Agrupamento de sensores no payload do datagrama



2.4.2 Tamanho de mensagem

O campo de tamanho de mensagem T_m é utilizado para informar ao receptor quantos blocos de dados estão sendo transmitidos no datagrama, este campo é formado pelo número de blocos em formato binário B_m e pelo bit de pariedade P_m para fechar uma seção de 4 bits. Como cada bloco possui 32 bits (exceto o primeiro), é possível determinar o comprimento total dos dados de aplicação de $B_m = (N_b - 1)_2$. O número de blocos N_b , pode variar de um a oito, resultando no comprimento esperado de bits no receptor para cada caso, conforme o Quadro 2.

Quadro 2 – Comprimento em bits para cada tamanho de mensagem (T_m)

Nº de Blocos	Nº de Bits	B_m	P_m	T_m
1	24	000	0	0000
2	56	001	1	0011
3	88	010	0	0100
4	120	011	1	0111
5	152	100	0	1000
6	184	101	1	1011
7	216	110	0	1100
8	248	111	1	1111

Conforme apresentado acima, para cada valor de B_m , um valor de P_m é calculado. O bit de paridade é calculado de forma a garantir que o número total de bits '1' na mensagem seja par, e é dado por

$$P_m = \begin{cases} 1, & \text{se } \left[\sum_{i=0}^{B_m} b_i = 0 \right] \bmod 2 \\ 0, & \text{se } \left[\sum_{i=0}^{B_m} b_i = 1 \right] \bmod 2 \end{cases}. \quad (2.2)$$

Ao final, o campo de tamanho de mensagem T_m é formado pela concatenação do valor de B_m e P_m , resultando em um campo de 4 bits CNES (2006).

2.4.3 Identificador da PCD

O identificador da PCD, PCD_{ID} , é um campo de 28 bits presente na estrutura da mensagem do usuário no formato PTT-A3. Ele é utilizado para identificar de forma única a PCD que está transmitindo a mensagem, sendo essencial para o correto encaminhamento e associação dos dados recebidos no centro de controle do sistema ARGOS, CNES (2006).

O PCD_{ID} é formado por um número de 20 bits, I_{PCD} , seguido por oito bits R_{PCD} de redundância calculados através da soma (checksum) dos bits do identificador, conforme

$$R_{PCD} = \left(\sum_{i=0}^{19} I_{PCD} \cdot 2^i \right) \bmod 256 , \quad (2.3)$$

$$PCD_{ID} = I_{PCD} \oplus R_{PCD} . \quad (2.4)$$

É importante destacar que a proteção contra erros nesse campo é assegurada de forma indireta pelo uso da codificação convolucional aplicada à mensagem como um todo, além da redundância oferecida pelo número de repetições da mensagem ao longo da passagem do satélite.

2.4.4 Bits de fim de mensagem

Ao final do datagrama, são inseridos entre sete e nove bits '0' com a finalidade de limpar o registrador do codificador convolucional, dando o encerramento da sequência codificada. A quantidade de bits de fim de mensagem adicionados depende do comprimento total da mensagem do usuário, conforme apresentado na Quadro 3. Apesar de não carregar dados úteis a nível de aplicação, esses bits são fundamentais para o correto funcionamento do processo de decodificação (CNES, 2006).

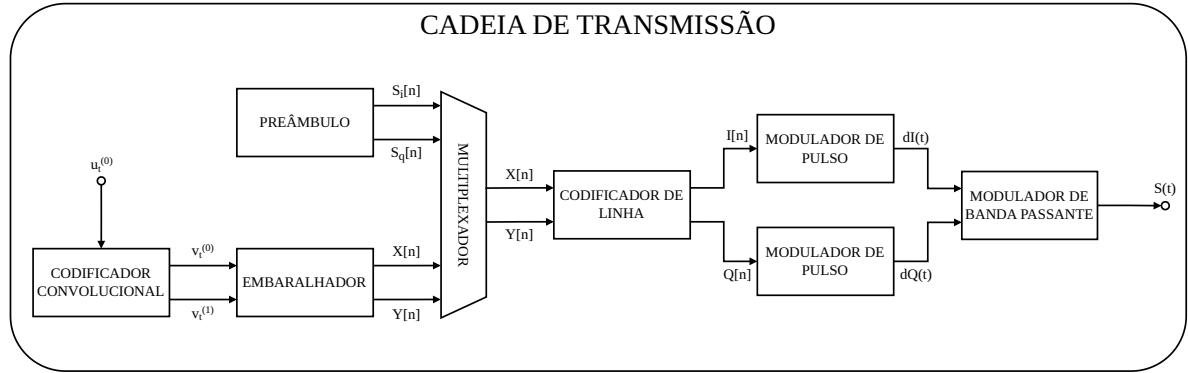
Quadro 3 – Comprimento da cauda (bits '0') para cada tamanho de mensagem

Nº de Blocos	Bits Aplicação	Bits Datagrama	Nº bits "0"
1	24	56	7
2	56	88	8
3	88	120	9
4	120	152	7
5	152	184	8
6	184	216	9
7	216	248	7
8	248	280	8

2.5 TRANSMISSOR PTT-A3

Na Figura 7 ilustra-se o diagrama de blocos do transmissor PTT-A3. Cada bloco é responsável por uma etapa da transmissão, desde a montagem do datagrama até a modulação em banda passante e a transmissão do sinal $s(t)$.

Figura 7 – Diagrama de blocos do transmissor ARGOS-3



2.5.1 Codificador convolucional

Antes da transmissão, os dados do datagrama precisam ser codificados, e esse processo é realizado através de um codificador convolucional. Essa técnica de codificação aplica uma operação lógica sobre uma janela deslizante de bits de entrada u_t , gerando uma sequência de saída com redundância controlada. Diferente da codificação por bloco, onde os dados são processados em blocos fixos, a codificação convolucional considera a sequência contínua de bits, combinando o bit atual com um número fixo de bits anteriores através de vetores geradores (Shu, 2011).

A taxa de codificação utilizada no padrão PTT-A3 é $R = 1/2$, o que significa que para cada bit de dados de entrada u_t , são gerados dois bits de saída, um no canal I e outro no canal Q , aumentando a redundância e melhorando a capacidade do sistema de detectar e corrigir erros. Para a codificação convolucional, são utilizados vetores geradores G_0 e G_1 , de acordo com o padrão CCSDS 131.1-G-2 (CNES, 2006). A representação binária dos vetores geradores é dada por

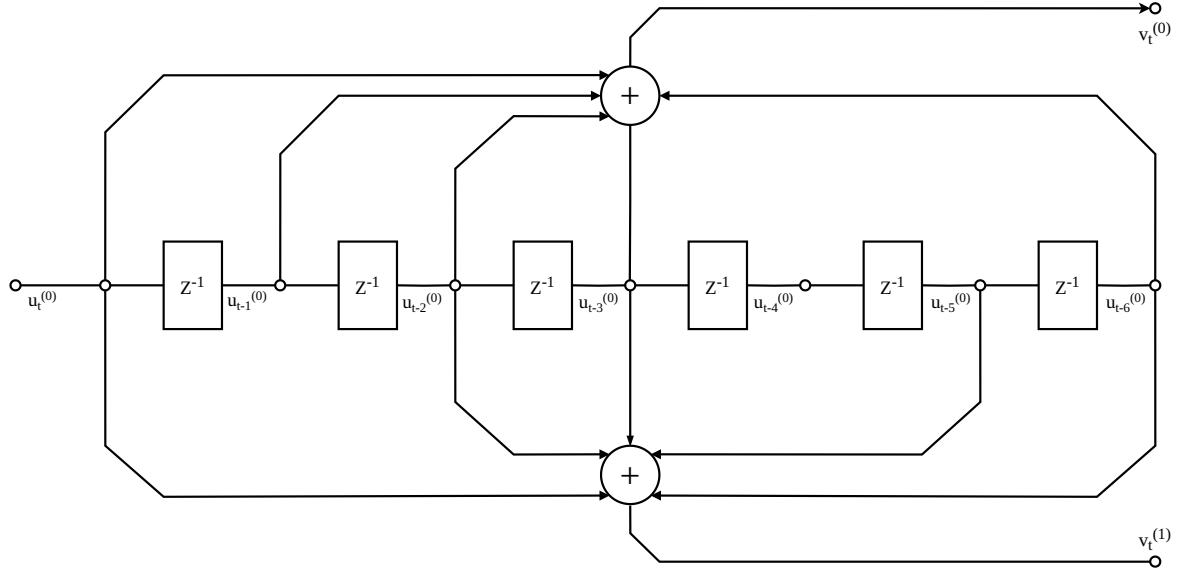
$$\begin{aligned} G_0 &= 121_{10} \quad \mapsto \quad G_0 = [1, 1, 1, 1, 0, 0, 1] \\ G_1 &= 091_{10} \quad \mapsto \quad G_1 = [1, 0, 1, 1, 0, 1, 1]. \end{aligned} \quad (2.5)$$

Os vetores geradores são utilizados para definir a estrutura do registrado do codificação convolucional aplicada à sequência de entrada u_t , resultando nas saídas $v_t^{(0)}$ e $v_t^{(1)}$, que correspondem, respectivamente, aos canais I e Q utilizados posteriormente na modulação QPSK. Essa operação pode ser representada por uma multiplicação vetorial entre uma janela deslizante de sete bits da entrada e a matriz formada pelos vetores geradores, conforme

$$\begin{bmatrix} v_t^{(0)} & v_t^{(1)} \end{bmatrix} = \begin{bmatrix} u_{(t)} & u_{(t-1)} & u_{(t-2)} & u_{(t-3)} & u_{(t-4)} & u_{(t-5)} & u_{(t-6)} \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}^T \quad (2.6)$$

que pode ser representada de forma equivalente pelo diagrama de blocos apresentado na Figura 8 (CNES, 2006).

Figura 8 – Diagrama de blocos do codificador convolucional ARGOS-3



2.5.2 Embaralhador

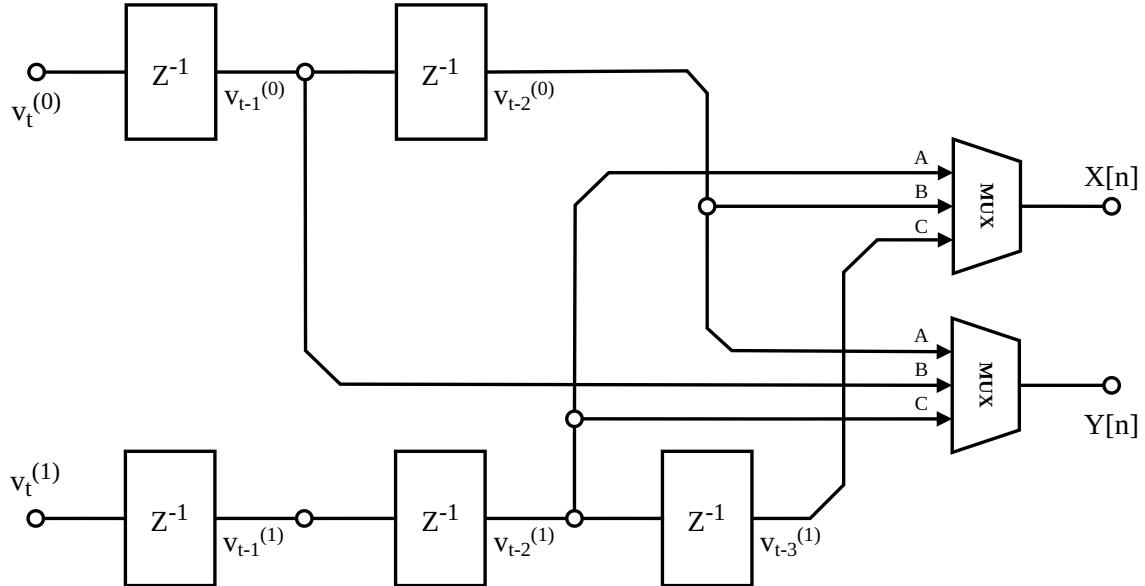
Após a codificação convolucional, os vetores de saída $v_t^{(0)}$ e $v_t^{(1)}$ são embaralhados para evitar padrões repetitivos, formando os vetores embaralhados $X[n]$ e $Y[n]$. O processo de embaralhamento é essencial para aumentar a robustez do sinal contra interferências e ruídos em rajada, pois os dados são espalhados ao longo da transmissão.

O embaralhador é implementado utilizando um registrador de deslocamento de 3 bits para cada canal, conforme apresentado na Figura 9. A cada novo bit de entrada, o registrador é deslocado, e o bit de saída calculado através de uma multiplexação entre as três saídas possíveis. Desta forma o index de embaralhamento de uma sequência de entrada é dado por

$$\begin{bmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & \dots \end{bmatrix} \mapsto \begin{bmatrix} Y_1 & X_2 & Y_2 & Y_4 & X_5 & Y_5 & \dots \end{bmatrix} \quad (2.7)$$

Onde $X[n]$ e $Y[n]$ são os vetores de bits embaralhados. O embaralhador é utilizado para garantir que os dados sejam melhor distribuídos diminuindo a correlação entre os bits de forma a aumentar a aleatoriedade antes da modulação QPSK, o que permite atingir uma melhor característica espectral. Esse processo pode ser representado pelo diagrama de blocos apresentado na Figura 9 (CNES, 2006; Rodrigues, 2018).

Figura 9 – Embaralhador de dados para o ARGOS-3



2.5.3 Codificação de Linha

Em seguida, os vetores de bits embaralhados $X[n]$ e $Y[n]$, já multiplexados com os vetores de preâmbulo $S_I[n]$ e $S_Q[n]$, precisam ser convertidos em vetores de simbolos, para isso é aplicada uma codificação de linha. O vetor $X[n]$ é codificado utilizando a técnica *Non-Return-to-Zero* (NRZ), enquanto o vetor $Y[n]$ é codificado utilizando a técnica *Manchester Encoding* (Manchester) (CNES, 2006).

A codificação NRZ é realizada removendo a componente DC da sequência de entrada, ou seja, no vetor $X[n]$ cada bit '1' é representado por '+1' e cada bit '0' é representado por '-1'. Esse processo pode ser descrito pela expressão

$$I[n] = \begin{cases} +1, +1, & \text{se } X[n] = 1 \\ -1, -1, & \text{se } X[n] = 0, \end{cases} \quad (2.8)$$

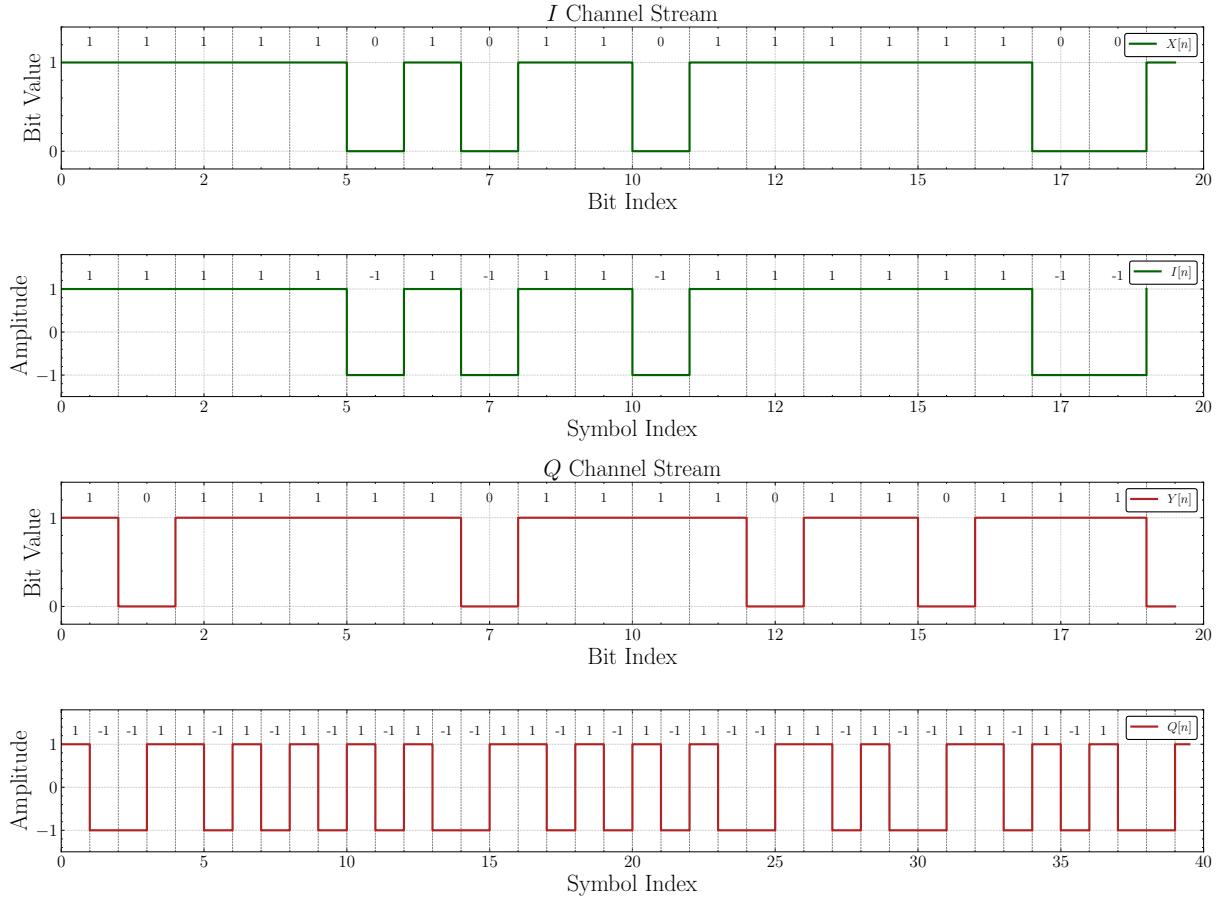
resultando em um vetor de saída $I[n]$, neste caso com o dobro do comprimento do vetor de entrada (para manter o mesmo comprimento do vetor $Q[n]$).

Para o vetor $Y[n]$, é aplicada a codificação Manchester, onde cada bit de entrada é representado por dois simbolos de saída, alternando entre '-1' e '+1'. A codificação Manchester é utilizada para garantir que haja transições de nível no sinal mesmo em sequências longas de bits iguais, essas transições refletem em mais trocas de símbolos na modulação QPSK. Esse processo pode ser descrito pela expressão

$$Y[n] = \begin{cases} +1, -1, & \text{se } Y[n] = 1 \\ -1, +1, & \text{se } Y[n] = 0 \end{cases}, \quad (2.9)$$

resultando assim em um vetor de saída $Q[n]$ codificado em Manchester.

Figura 10 – Codificação de linha dos vetores I e Q



2.5.4 Modulação de Pulso

Uma vez com os vetores de simbolo $I[n]$ e $Q[n]$, é aplicada a modulação de pulso, isto é, os vetores de simbolo são superamostrados, em função de f_s e filtrados para formar uma sequência contínua de símbolos ao longo de t , onde cada bit de informação é transmitido durante um período de tempo $T_b = 1/R_b$ (tempo de bit), definido com base na taxa de bit R_b (CNES, 2006).

Para aplicar a modulação de pulso e gerar os sinais analógicos $d_I(t)$ e $d_Q(t)$, os vetores de simbolo são multiplicados com um pulso com resposta ao impulso $g(t)$. A formatação é dada por

$$d'_I(t) = \sum_n I[n] \cdot g(t - nT_b) \quad \text{e} \quad d'_Q(t) = \sum_n Q[n] \cdot g(t - nT_b) , \quad (2.10)$$

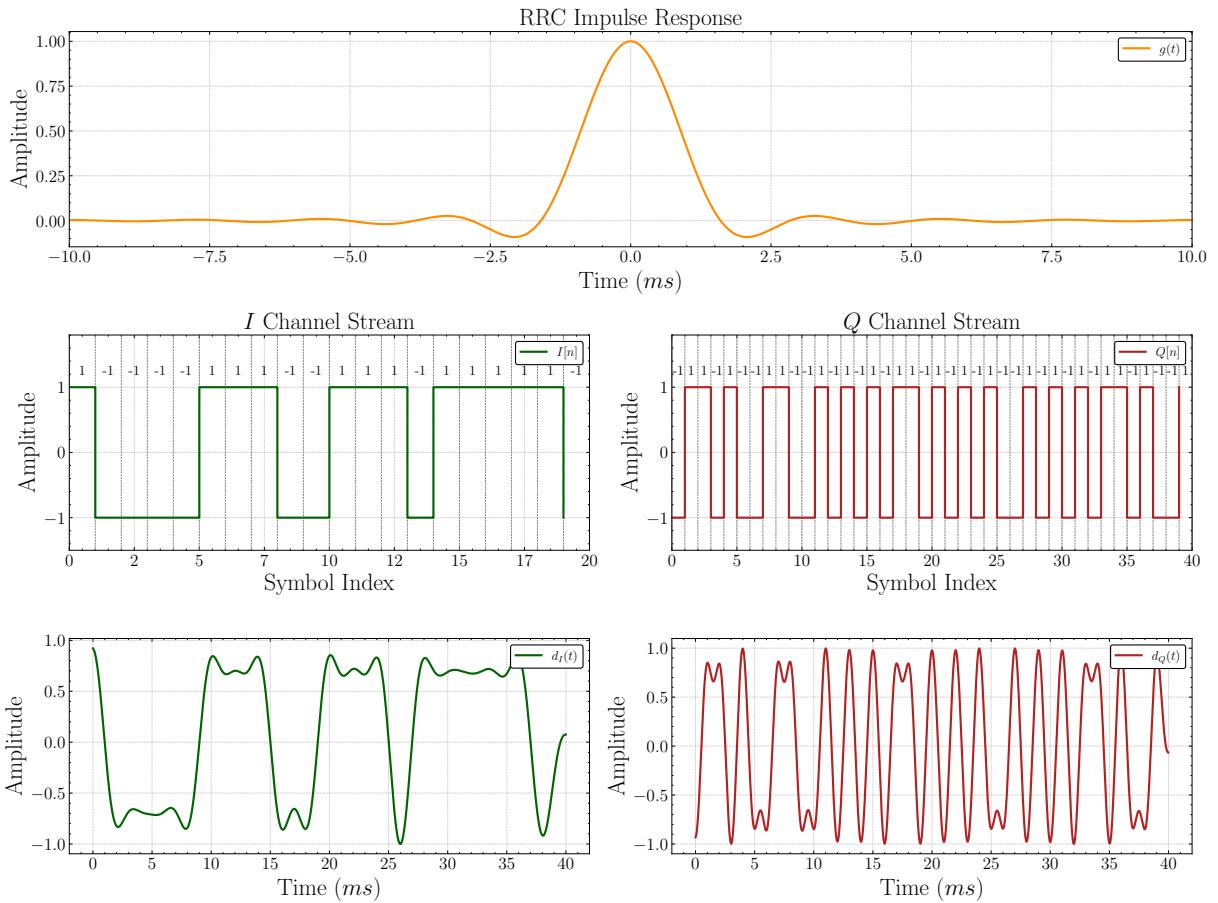
onde é utilizado um pulso $g(t)$ do tipo (RRC), definido por

$$g(t) = \frac{(1 - \alpha)\text{sinc}((1 - \alpha)t/T_b) + \alpha(4/\pi) \cos(\pi(1 + \alpha)t/T_b)}{1 - (4\alpha)^2} \quad (2.11)$$

onde α é o fator de roll-off do pulso, que controla a largura de banda W do sinal modulado.

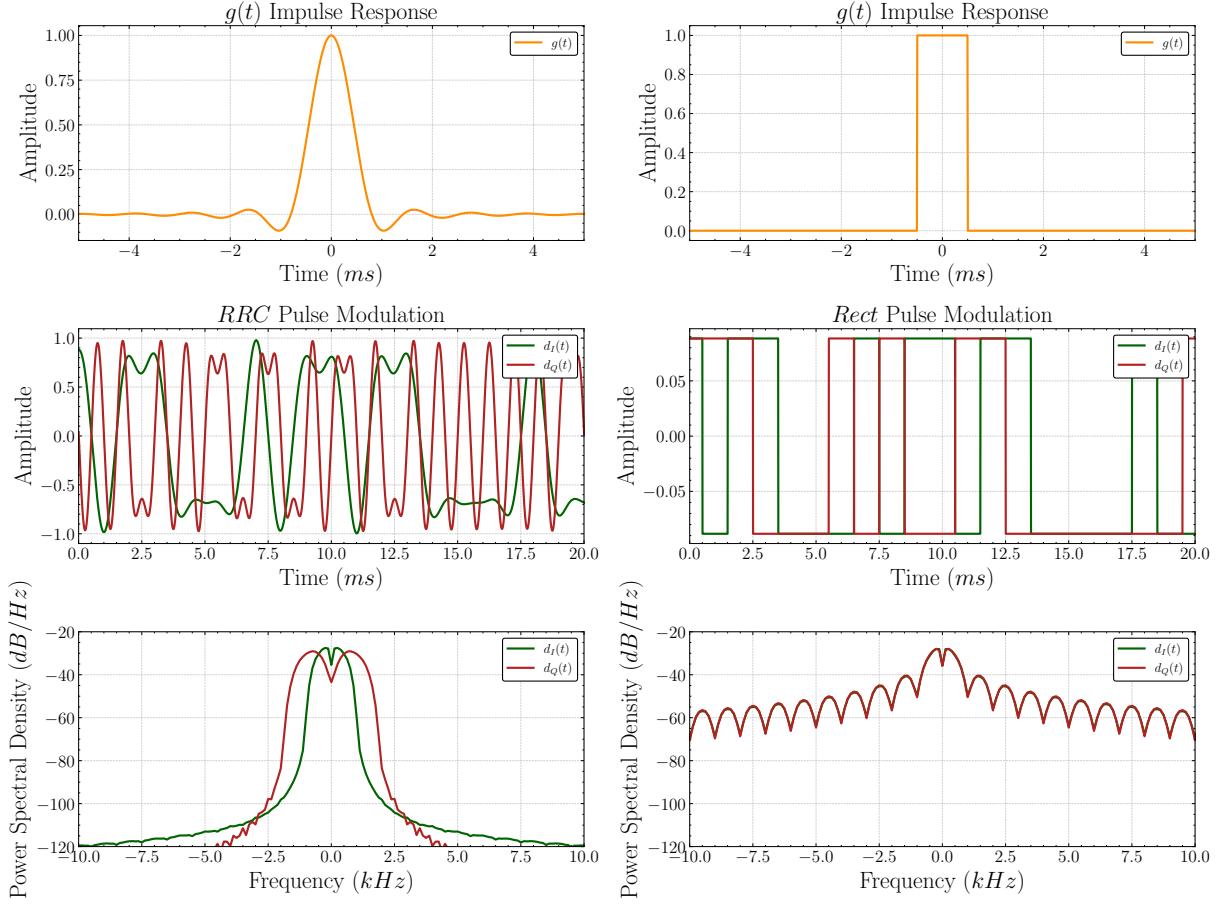
A formatação dos vetores $I[n]$ e $Q[n]$ é ilustrada na Figura 11, onde a resposta ao impulso $g(t)$ é apresentada em conjunto com os sinais $d_I(t)$ e $d_Q(t)$.

Figura 11 – Modulação de pulso dos canais I e Q



Quanto maior o α , mais suave é a transição entre os símbolos, mas também maior é a largura de banda ocupada (Sklar, 1988). A Figura 12 apresenta a comparação entre os sinais $d_I(t)$ e $d_Q(t)$ quando é aplicada a modulação de pulso com pulso RRC e Rect, ou seja, quando o vetor de simbolos é apenas superamostrado.

Figura 12 – Comparação entre modulação de pulso RRC e Rect



Nota-se que utilizando formatação Rect, o sinal apresenta transições abruptas entre os símbolos, o que pode gerar interferência entre símbolos adjacentes *Inter-Symbol Interference* (ISI) e aumentar a largura de banda do sinal transmitido, a resposta ao impulso do pulso Rect é dada por

$$g(t) = \text{rect}\left(\frac{t}{T_b}\right) \quad (2.12)$$

onde, $g(t)$ é a resposta ao impulso da função, t é o vetor de tempo e T_b é o período de bit. Já com a modulação de pulso, o sinal é suavizado no domínio do tempo, reduzindo a interferência entre símbolos e ajustando a largura de banda do sinal conforme os parâmetros do modulador de pulso (CNES, 2006; Sklar, 1988; Rodrigues, 2018).

Dessa forma, podemos calcular a banda passante unilateral do sinal modulado W para modulação de pulso RRC como

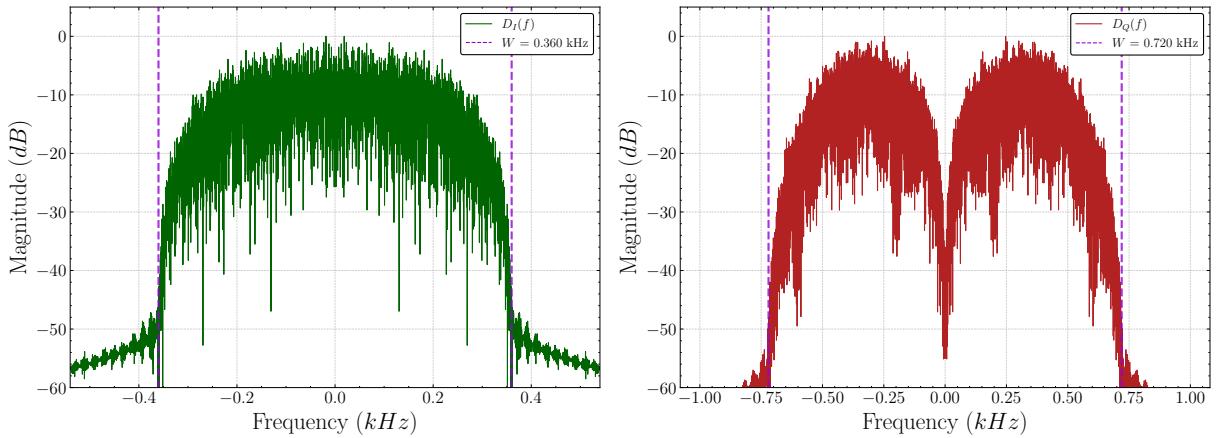
$$W_{RRC} = \frac{1 + \alpha}{2 \cdot T_b}, \quad (2.13)$$

onde α é o fator de roll-off do pulso e T_b é o período de bit. Já para modulação de pulso Rect, a banda passante do sinal modulado é dada por

$$W_{Rect} = \frac{1}{T_b} . \quad (2.14)$$

Assim, a largura de banda W do sinal modulado em RRC considerando um tempo de bit $T_b = 1/400$ e um fator de roll-off $\alpha = 0,8$ é de 360 Hz (banda unilateral), ou seja, $2W = 720 \text{ Hz}$ de banda bilateral, em contraste com os 800 Hz de banda (bilateral) do sinal modulado em Rect. Na Figura 13 é possível observar a diferença na banda consumida pelos sinais $d_I(t)$ e $d_Q(t)$ após a modulação de pulso sobre os canais.

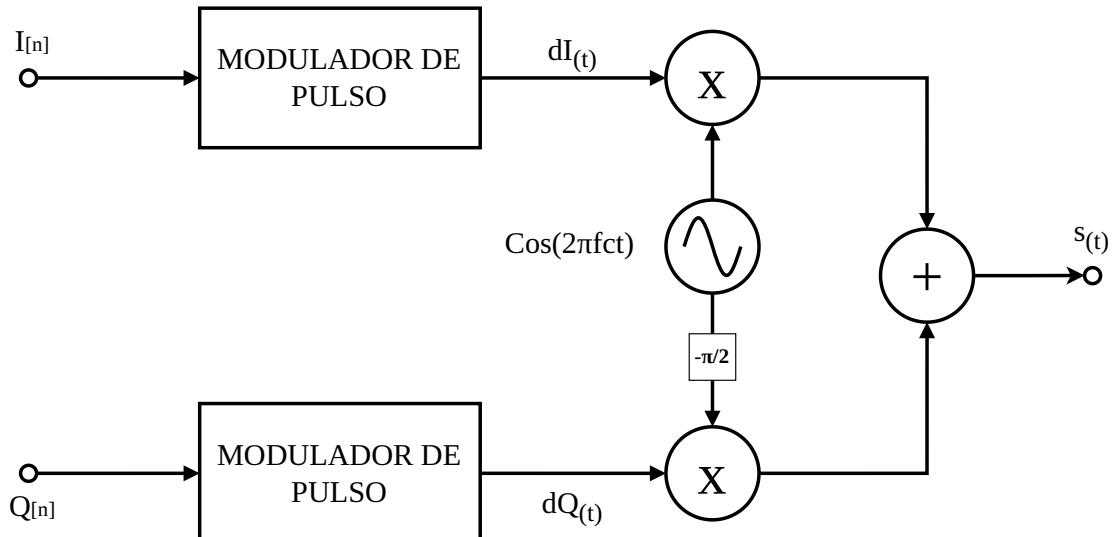
Figura 13 – Banda consumida nos canais I e Q



2.5.5 Modulação em banda passante

Uma vez com os pares de sinal $d_I(t)$ e $d_Q(t)$ correspondentes a cada canal, pode-se realizar a modulação em banda passante para transmissão do sinal $s(t)$. Na Figura 14 é ilustrado o diagrama de blocos do modulador QPSK (CNES, 2006).

Figura 14 – Diagrama de blocos do modulador QPSK

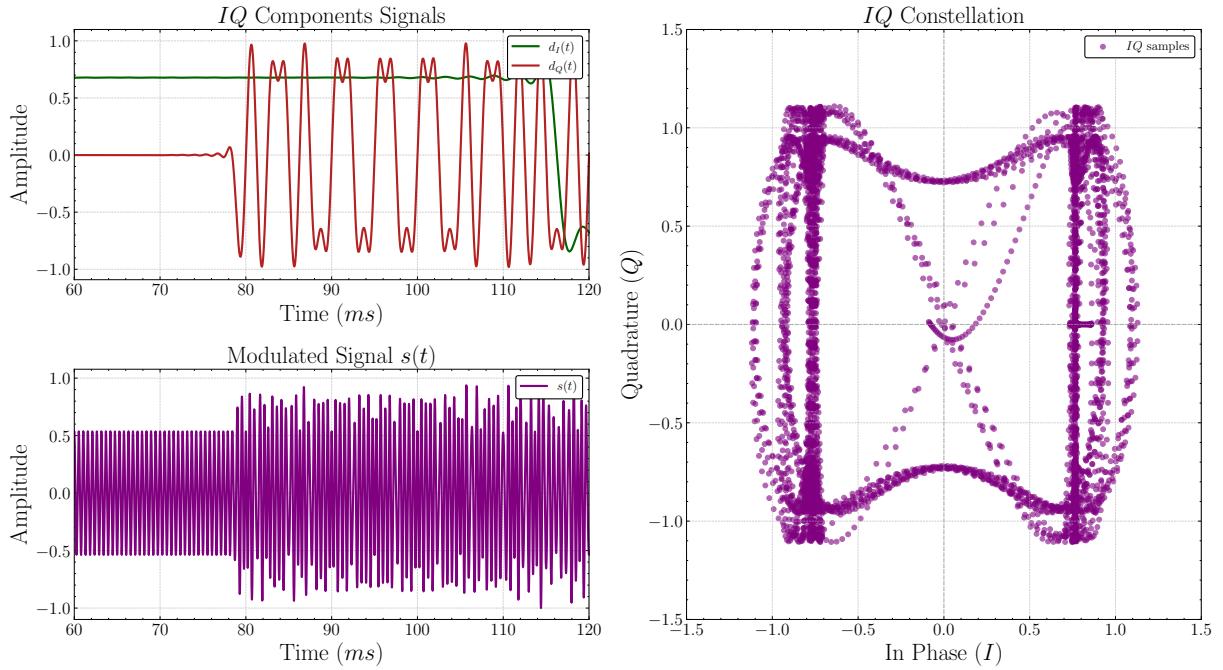


No processo de modulação QPSK, o sinal em fase $d_I(t)$ é multiplicado por uma componente cossenoide em frequência f_c e o sinal em quadratura $d_Q(t)$ é multiplicado por uma componente senoidal em f_c , seguindo a expressão

$$s(t) = Ad'_I(t) \cos(2\pi f_c t + \phi_0) - Ad'_Q(t) \sin(2\pi f_c t + \phi_0) , \quad (2.15)$$

onde A é a amplitude do sinal modulado, f_c é a frequência da portadora, em torno de $401,625 MHz$ e ϕ_0 é o desvio de fase inicial do sinal, que é considerado como zero ($\phi_0 = 0$). A Figura 15 ilustra o processo de modulação IQ, mostrando os sinais $d_I(t)$ e $d_Q(t)$ e a constelação resultante, onde cada ponto representa um símbolo modulado.

Figura 15 – Modulação em banda passante dos canais I e Q

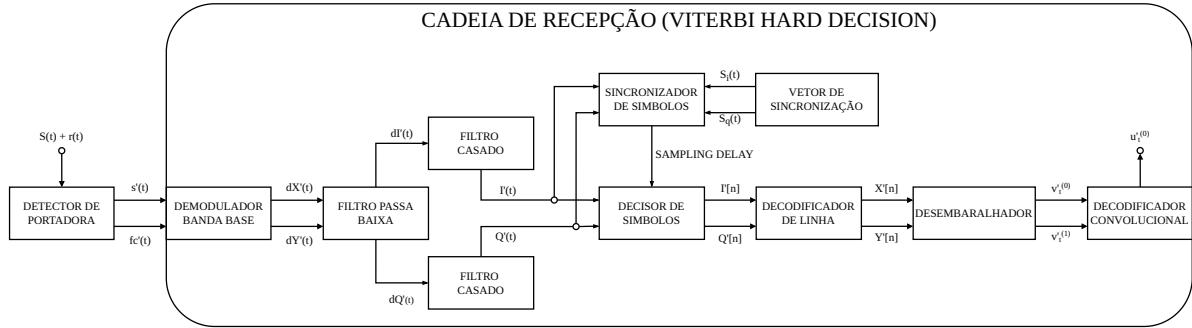


A constelação QPSK na Figura 15 mapea cada par de amostra dos vetores $d_I(t)$ e $d_Q(t)$ no plano complexo $(I + jQ)$. Após a transmissão, o sinal $s(t)$ é somado a um vetor de ruído $r(t)$ para simular as condições reais no momento da recepção.

2.6 RECEPTOR PTT-A3

Como a transmissão dos dados das PCD não segue uma estrutura de canais discretos, ou seja, cada PCD seleciona uma frequência f_n dentro da faixa de $401,62$ a $401,65MHz$ e realiza a transmissão. Dessa forma, o receptor no satélite necessita de um mecanismo de detecção de portadora para avaliar as candidatas de f_c , podendo assim demodular o sinal.

Figura 16 – Diagrama de blocos do receptor ARGOS-3



2.6.1 Detecção de portadora

Para realizar a detecção da portadora, o sinal recebido acrescido de ruído deve ser inicialmente amostrado e dividido em segmentos discretos $x_n[m]$ no tempo. O satélite ARGOS realiza a decisão do sinal a cada 10ms, esse processo é definido por

$$x_n[m] = s(mT_n) , \quad (2.16)$$

onde $x_n[m]$ representa o segmento de sinal, $s(t)$ é o sinal recebido, m é o índice de amostra do segmento de tempo e n é o tempo de amostragem, definido como $T_n = f_s * 10\text{ms}$, onde f_s é a frequência de amostragem do sistema.

Em seguida, aplica-se a *Fast Fourier Transform* (FFT) no vetor $x_n[m]$, obtendo-se o espectro de frequência do sinal amostrado $X_n[k]$, conforme a expressão

$$X_n[k] = \sum_{m=0}^{N-1} x_n[m] e^{-j2\pi km/N} . \quad (2.17)$$

A partir da amostras de $X_n[k]$, se calcula a potência em cada componente de frequência, para obter os valores $P_n[k]$, conforme

$$P_n[k] = |X_n[k]|^2 . \quad (2.18)$$

Em seguida, para cada índice k do espectro calculado, é feita uma comparação com um limiar pré-definido P_t , conforme foi apresentado anteriormente na Figura 4. Caso a potência $P_n[k]$ seja maior que P_t , a frequência é registrada pois existe a possibilidade de que uma portadora f_n esteja presente naquela frequência.

Por fim, no proximo segmento, $X_{n+1}[k]$, o sistema verifica se a frequência registrada no segmento anterior persiste, ou seja, se a potência de k também é maior que o limiar P_t , caso positivo, o sistema considera que a frequência f_n é a portadora f_c do sinal recebido e instância a cadeia de recepção passando f_c como parâmetro.

2.6.2 Demodulador banda base

Para realizar a demodulação do sinal $s(t)$ recebido retornando as componentes $d'_X(t)$ e $d'_Y(t)$ em banda base, o sinal $s(t)$ é multiplicado por duas portadoras ortogonais, $x_i(t)$ para demodular o canal I e $y_q(t)$ para demodular o canal Q . Assumindo sincronismo perfeito, o processo de demodulação para o canal I pode ser expresso como

$$d'_X(t) = s(t) \cdot x_I(t) = [A \cdot d'_I(t) \cos(2\pi f_c t) - A \cdot d'_Q(t) \sin(2\pi f_c t)] \cdot 2 \cos(2\pi f_c t) \quad (2.19)$$

$$d'_X(t) = \underbrace{A \cdot d'_I(t)}_{\text{Banda base}} + \underbrace{[A \cdot d'_I(t) \cos(4\pi f_c t) - A \cdot d'_Q(t) \sin(4\pi f_c t)]}_{\text{Dobro da frequência } f_c}, \quad (2.20)$$

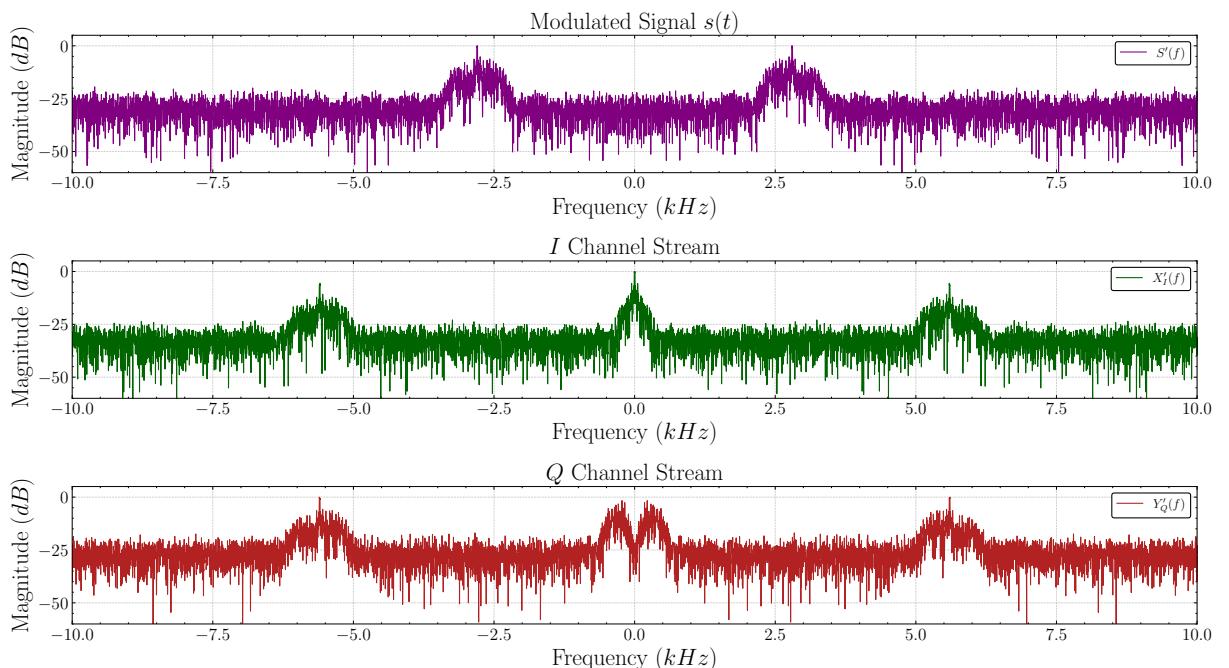
onde $d_I(t)$ é o sinal banda base. O mesmo processo é realizado para o canal Q , que isola o sinal em quadratura $d_Q(t)$ da seguinte forma

$$d'_Y(t) = s(t) \cdot y_Q(t) = [A \cdot d'_I(t) \cos(2\pi f_c t) - A \cdot d'_Q(t) \sin(2\pi f_c t)] \cdot 2 \sin(2\pi f_c t) \quad (2.21)$$

$$d'_Y(t) = \underbrace{A \cdot d'_Q(t)}_{\text{Banda base}} + \underbrace{[A \cdot d'_Q(t) \cos(4\pi f_c t) + A \cdot d'_I(t) \sin(4\pi f_c t)]}_{\text{Dobro da frequência } f_c}. \quad (2.22)$$

A Figura 17 ilustra o processo de demodulação dos canais I e Q , onde os sinais multiplicados por $x_i(t)$ e $y_q(t)$ são apresentados no espectro, mostrando a presença dos sinais em banda base e as componentes de alta frequência resultantes da multiplicação com as portadoras (considerando um $f_c = 2kHz$). (CNES, 2006)

Figura 17 – Componentes I e Q após demodulação



2.6.3 Filtragem passa baixa

Para isolar os sinais em banda base $d'_I(t)$ e $d'_Q(t)$, é necessário aplicar um *Low-Pass Filter* (LPF), para isso utiliza-se um filtro Butterworth (6^a Ordem) com frequência de corte de 720 Hz, com resposta ao impulso $h(t)$, que remove as componentes de alta frequência em $2 \cdot f_c$ resultantes da multiplicação de demodulação realizada anteriormente (Rodrigues, 2018).

O filtro Butterworth é escolhido por sua resposta em frequência plana e sem ondulações (ganhos ou atenuações) na banda passante, o que é ideal para preservar a integridade do sinal que está passando pelo filtro, neste caso como é um filtro passa baixa, o sinal que está em banda base. A aplicação do filtro LPF aos sinais $d'_X(t)$ e $d'_Y(t)$ pode ser expressa como

$$d'_I(t) = d'_X(t) * h(t) \quad \text{e} \quad d'_Q(t) = d'_Y(t) * h(t) , \quad (2.23)$$

onde $*$ representa a operação de convolução entre os sinais $d'_X(t)$ e $d'_Y(t)$ com a resposta ao impulso do filtro $h(t)$. A resposta ao impulso do filtro $h(t)$ é dada por

$$h(t) = \mathcal{F}^{-1}\{H(f)\} , \quad (2.24)$$

onde \mathcal{F}^{-1} é a transformada inversa de Fourier e $H(f)$ é a resposta em frequência do filtro Butterworth, definida por

$$H(f) = \frac{1}{\sqrt{1 + \left(\frac{f}{f_c}\right)^{2n}}} , \quad (2.25)$$

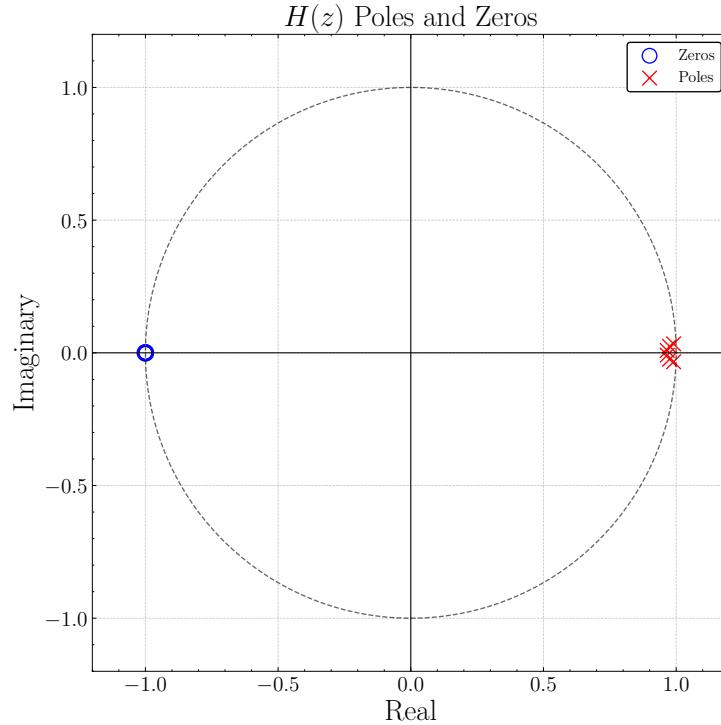
onde n é a ordem do filtro (neste caso, $n = 6$) e f_c é a frequência de corte do filtro, definida como 720 Hz.

Os polos e zeros do filtro Butterworth também podem ser calculados com base na função de transferência discreta $H(z)$ do filtro. O cálculo dos coeficientes de $H(z)$ do filtro foram realizados utilizando a biblioteca *scipy.signal* do Python, que implementa o método de bilinearização para a conversão da função de transferência contínua $H(s)$ para o domínio discreto, a função de transferência discreta é dada por

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} , \quad (2.26)$$

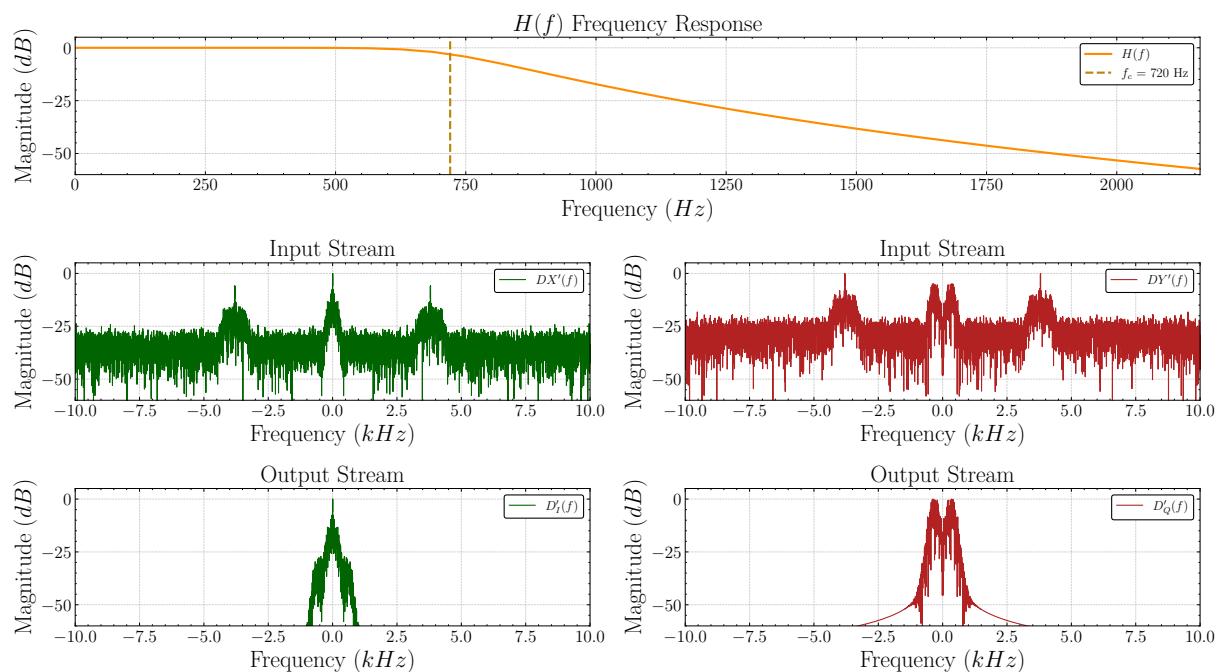
onde b_0, b_1, \dots, b_m são os coeficientes do numerador (zeros) e a_0, a_1, \dots, a_n são os coeficientes do denominador (polos) da função de transferência do filtro digital, assim podemos plotar os coeficientes de $H(z)$ no plano complexo, conforme apresentado na Figura 18 (Sklar, 1988).

Figura 18 – Polos e zeros do filtro Passa Baixa



A Figura 19 apresenta o espectro dos sinais $d'_X(t)$ e $d'_Y(t)$ antes e após a filtragem passa-baixa, bem como a resposta em frequência $H(f)$ do filtro, onde é possível observar a remoção das componentes de alta frequência, deixando apenas os sinais em banda base, além da frequência de corte do filtro (onde a atenuação alcança -3 dB).

Figura 19 – Componentes I Q após filtragem passa baixa



2.6.4 Filtragem casada

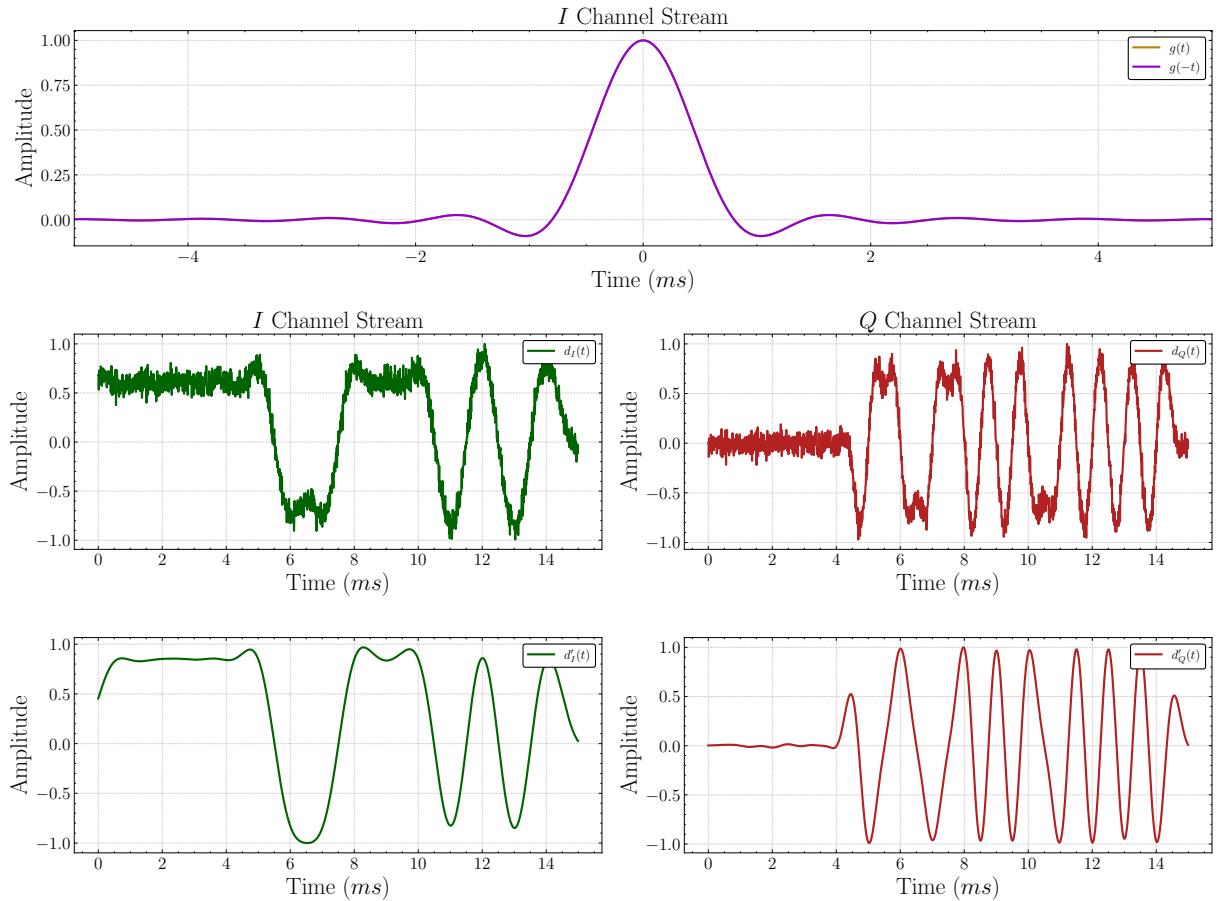
Os sinais $d'_I(t)$ e $d'_Q(t)$ em banda base, e já filtrados por um LPF, passam por um filtro casado *Matched Filter* (MF), que é utilizado para maximizar a relação sinal-ruído (SNR). O filtro casado é ajustado para coincidir com o inverso do pulso $g(-t)$ utilizado na formatação do sinal na transmissão (Sklar, 1988).

O filtro casado é aplicado aos sinais $d'_I(t)$ e $d'_Q(t)$ através da expressão

$$I'(t) = d_I(t) * g(-t) \quad \text{e} \quad Q'(t) = d_Q(t) * g(-t) . \quad (2.27)$$

A Figura 20 apresenta o espectro dos sinais $I'(t)$ e $Q'(t)$ após a filtragem casada, onde é possível observar a melhoria na relação sinal-ruído e a remoção de componentes indesejadas.

Figura 20 – Componentes I e Q após filtragem casada



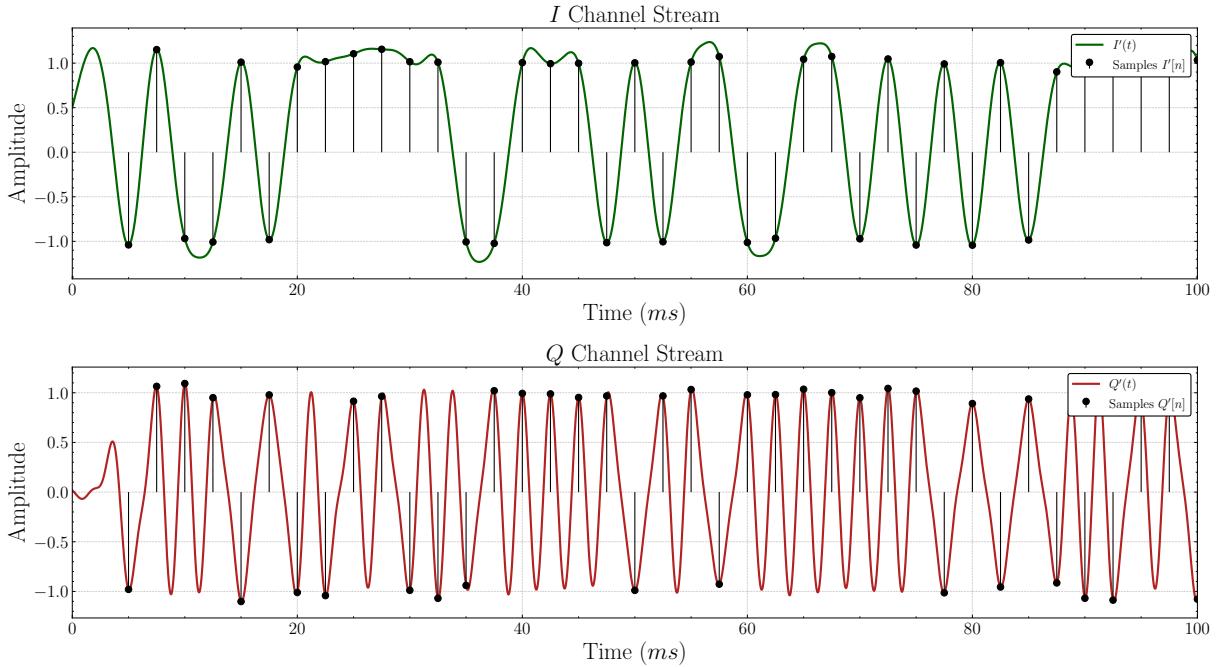
2.6.5 Decisão de símbolos

Uma vez com os sinais $d'_X(t)$ e $d'_Y(t)$ filtrados, considerando sincronismo perfeito, o próximo passo é a decisão dos sinais em instantes de T_b (instante ótimo de decisão) a partir de um delay inicial τ , para recuperar os símbolos transmitidos. O processo de decisão pode ser expresso como

$$I'[n] = d'_I(n \cdot T_b + \tau) \quad \text{e} \quad Q'[n] = d'_Q(n \cdot T_b + \tau) , \quad (2.28)$$

onde n é o índice de amostra, T_b é o tempo de bit $1/R_b$ e τ é o atraso inicial da decisão. A Figura 21 apresenta os sinais amostrados $I'[n]$ e $Q'[n]$.

Figura 21 – Amostragem das componentes I e Q

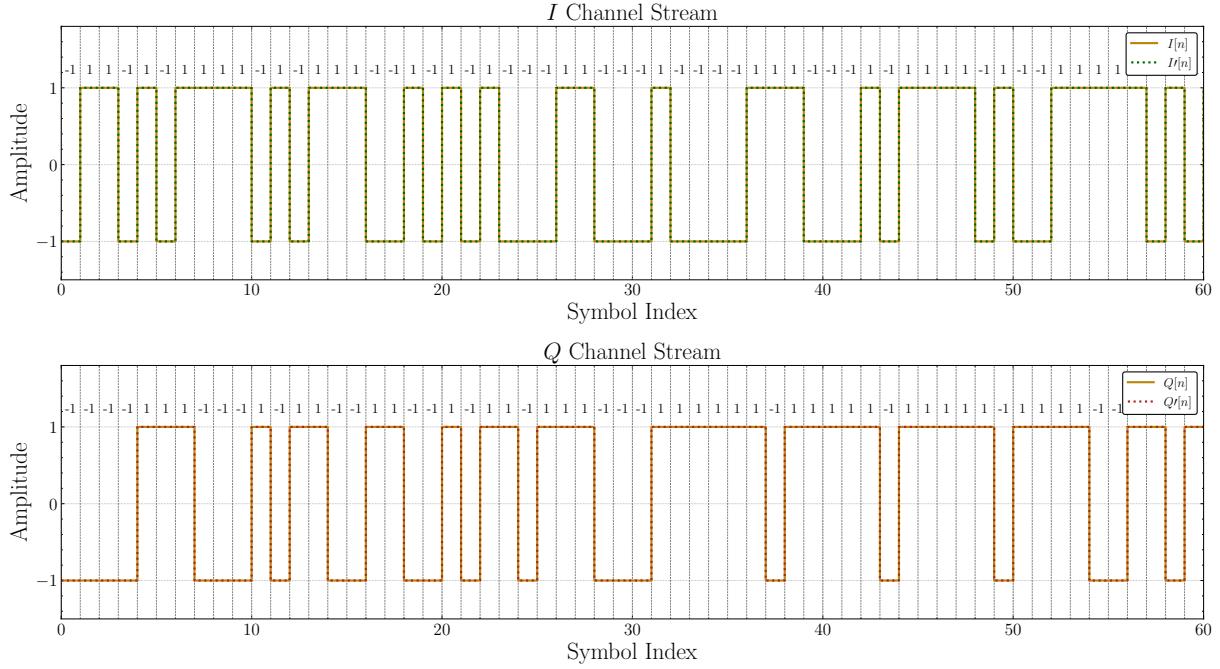


Os valores amostrados $I'[n]$ e $Q'[n]$ são então decididos, isto é, quantizados para valores discretos, correspondendo aos símbolos transmitidos. O mapeamento dos pares pode ser expressado como

$$I'[n] = \begin{cases} +1, & \text{se } I'[n] \geq 0 \\ -1, & \text{se } I'[n] < 0 \end{cases}, \quad Q'[n] = \begin{cases} +1, & \text{se } Q'[n] \geq 0 \\ -1, & \text{se } Q'[n] < 0 \end{cases}, \quad (2.29)$$

onde $I'[n]$ e $Q'[n]$ são os vetores simbolo. A Figura 22 apresenta os simbolos decididos nos vetores $I'[n]$ e $Q'[n]$ após o processo de quantização, em comparação com os vetores originais $I[n]$ e $Q[n]$ transmitidos.

Figura 22 – Comparaçāo dos vetores I e Q decididos com vetores transmitidos



2.6.6 Decodificador de linha

Após a decisão dos símbolos, os vetores $I'[n]$ e $Q'[n]$ precisam ser convertidos de volta para a representação original dos bits. Para isso, são aplicadas as técnicas de decodificação de linha inversas às utilizadas na transmissão: NRZ para o canal I e Manchester para o canal Q . A decodificação NRZ é realizada mapeando os valores '+1' para o bit '1' e '-1' para o bit '0', conforme

$$X'[n] = \begin{cases} 1, & \text{se } I'[n] = +1 \\ 0, & \text{se } I'[n] = -1 \end{cases}, \quad (2.30)$$

resultando no vetor de bits $X'[n]$ correspondente ao canal I . O mesmo processo é aplicado ao canal Q , onde a decodificação Manchester mapeia os pares de valores '+1,-1' para o bit '1' e '-1,+1' para o bit '0', conforme

$$Y'[n] = \begin{cases} 1, & \text{se } Q'[n] = +1, -1 \\ 0, & \text{se } Q'[n] = -1, +1 \end{cases}, \quad (2.31)$$

resultando no vetor de bits $Y'[n]$ correspondente ao canal Q .

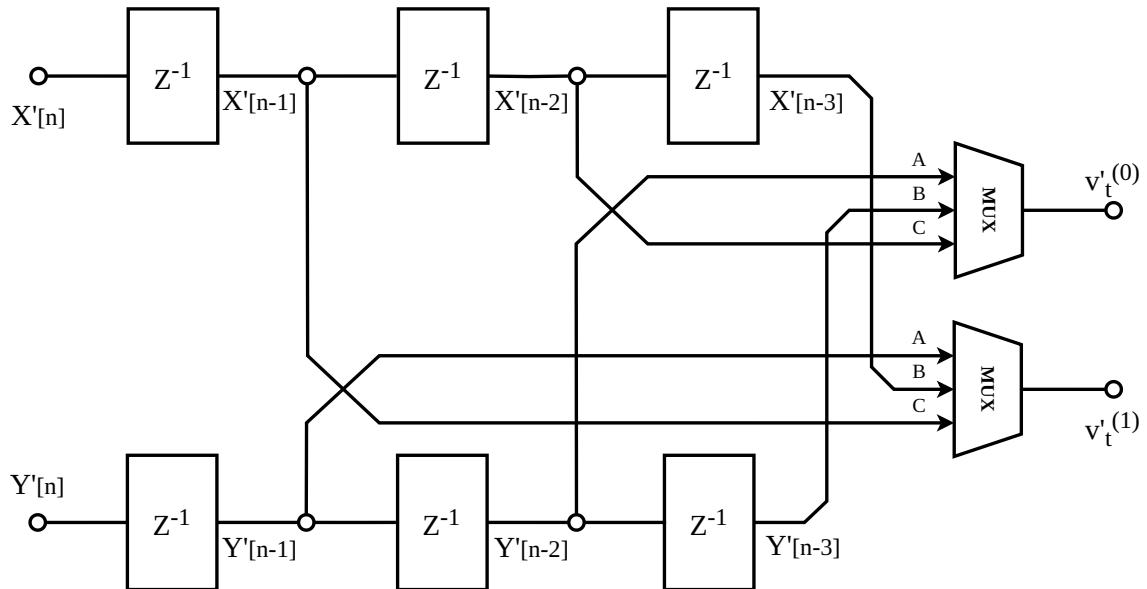
2.6.7 Desembaralhador

Após o processo de decodificação de linha, os dados vetores de bits $X'[n]$ e $Y'[n]$ ainda encontram-se embaralhados, então é necessário realizar o desembaralhamento para

restaurar a sequência original dos bits antes da decodificação convolucional (CNES, 2006).

O desembaralhador utiliza a mesma estrutura lógica do embaralhador, porém executando a operação inversa de reorganização dos bits, com base em regras posicionais dependentes do índice dos bits n . A Figura 23 apresenta o diagrama de blocos do desembaralhador utilizado no PTT-A3, que refaz a ordenação dos bits após a demodulação (Rodrigues, 2018).

Figura 23 – Diagrama de blocos do Desembaralhador ARGOS-3



O processo de desembaralhamento retorna os vetores de bit, $v_t^{(0)'} e v_t^{(1)'}$, que correspondem aos dados codificados nos canais I e Q , respectivamente.

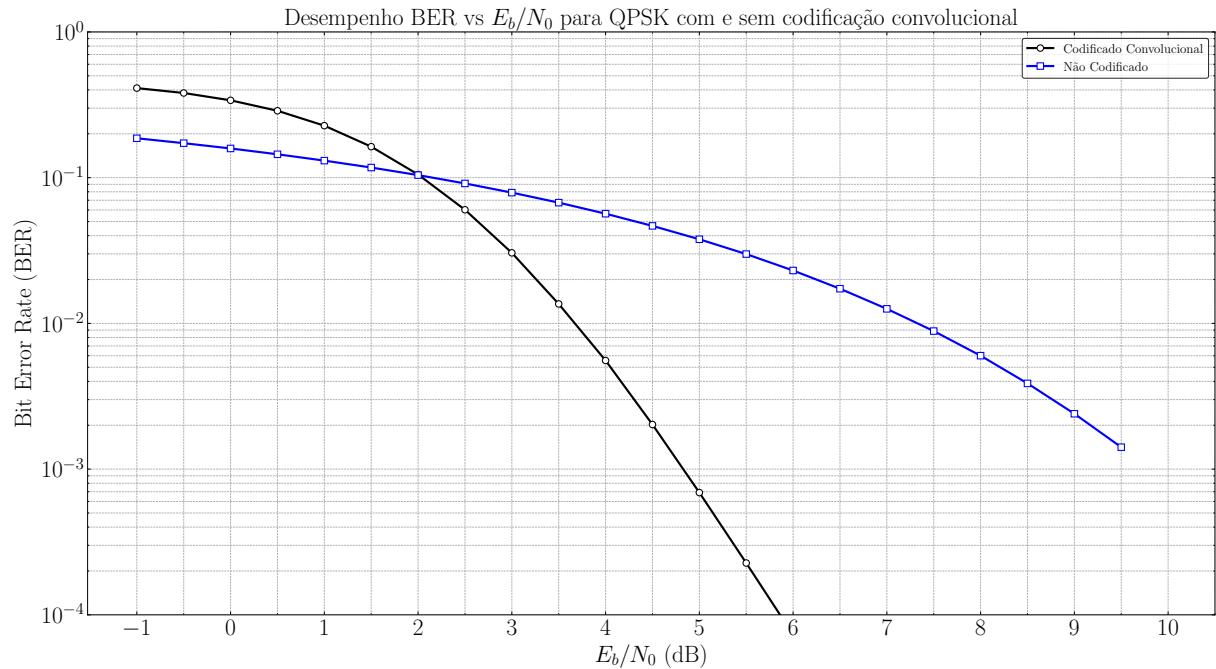
2.6.8 Decodificador convolucional

Após o desembaralhamento, os vetores $v_t^{(0)'} e v_t^{(1)'}$ estão prontos para a decodificação convolucional para obter novamente a sequência u'_t correspondente ao vetor de bits do datagrama. O algoritmo mais popularmente utilizado para essa etapa é o algoritmo de Viterbi, que implementa a decodificação *Maximum Likelihood Decoding* (MLD) para códigos convolucionais, calculando o caminho mais provável através de *Hamming Distance* (Hamming Distance) para a mensagem recebida em relação à mensagem possível (CNES, 2006; Rodrigues, 2018).

No padrão ARGOS-3, o código convolucional utilizado segue o padrão *Consultative Committee for Space Data Systems* (CCSDS) 131.1-G-2, que, por sua vez, possui uma distância livre conhecida de $d_{free} = 10$. Quanto maior for o valor de d_{free} , maior será a robustez do código e sua capacidade de detectar e corrigir erros.

Assim, conforme o comparativo ilustrado na Figura 24 entre a transmissão QPSK sem codificação e com codificação convolucional, o uso desta técnica na transmissão do sinal permite operar em valores de E_b/N_0 menores em relação a não utilização do codificador, para a BER visada pelo sistema, permitindo uso em ambientes com maior ruído para a mesma taxa de erro desejada.

Figura 24 – Comparação de BER vs E_b/N_0 utilizando codificação convolucional



3 DESENVOLVIMENTO E RESULTADOS

Este capítulo apresenta a implementação do sistema de comunicação digital proposto, detalhando cada etapa do processo, desde a geração dos bits até a recuperação do datagrama no receptor. A seguir, são descritas as principais componentes do sistema, incluindo a cadeia de transmissão, o canal de comunicação com ruído adicionado, a detecção de portadora e a cadeia de recepção.

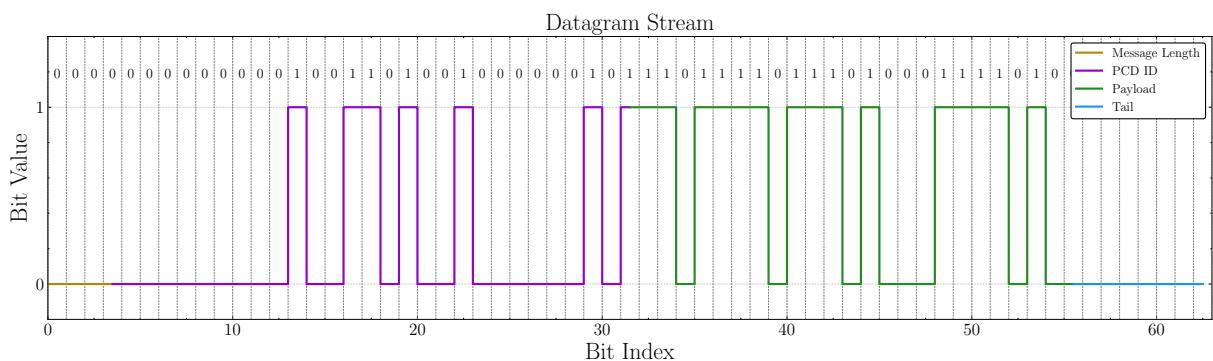
3.1 CADEIA DE TRANSMISSÃO

A primeira etapa do sistema é a geração da sequência de bits que compõem o datagrama ARGOS-3, conforme descrito na Seção 2.3.3. Essa sequência é multiplexada com um preâmbulo para permitir a sincronização de símbolos no receptor. Em seguida, os bits são codificados utilizando codificação de linha para melhorar as características espectrais do sinal modulado, como detalhado na Seção 2.5.3.

3.1.1 Sequência de transmissão

O primeiro passo na cadeia de transmissão é a geração da sequência de bits que compõem o datagrama ARGOS-3, conforme ilustrado na Figura 25, para a montagem do datagrama é necessário o número de identificação da PCD, I_{PCD} , e os dados que serão transmitidos (payload), conforme detalhado na seção 2.3.3, desta forma, o comprimento da mensagem T_m , o campo de identificação da PCD PCD_{ID} e a cauda do datagrama E_m podem ser calculados.

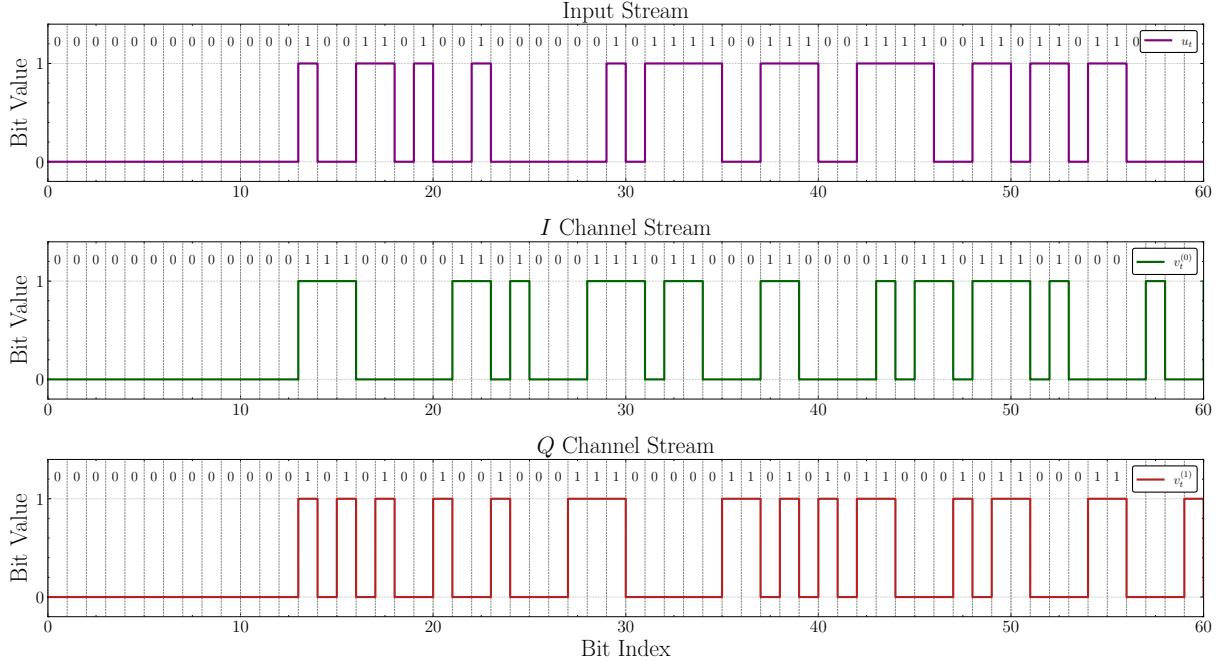
Figura 25 – Streambits do datagrama ARGOS-3



Após a montagem dos bits u_t do datagrama, os mesmos são codificados utilizando codificação convolucional, conforme detalhado na seção 2.5.1, resultando nos vetores de bits $v_t^{(0)}$ e $v_t^{(1)}$, que por sua vez são embaralhados resultando nas sequências $X[n]$ e $Y[n]$.

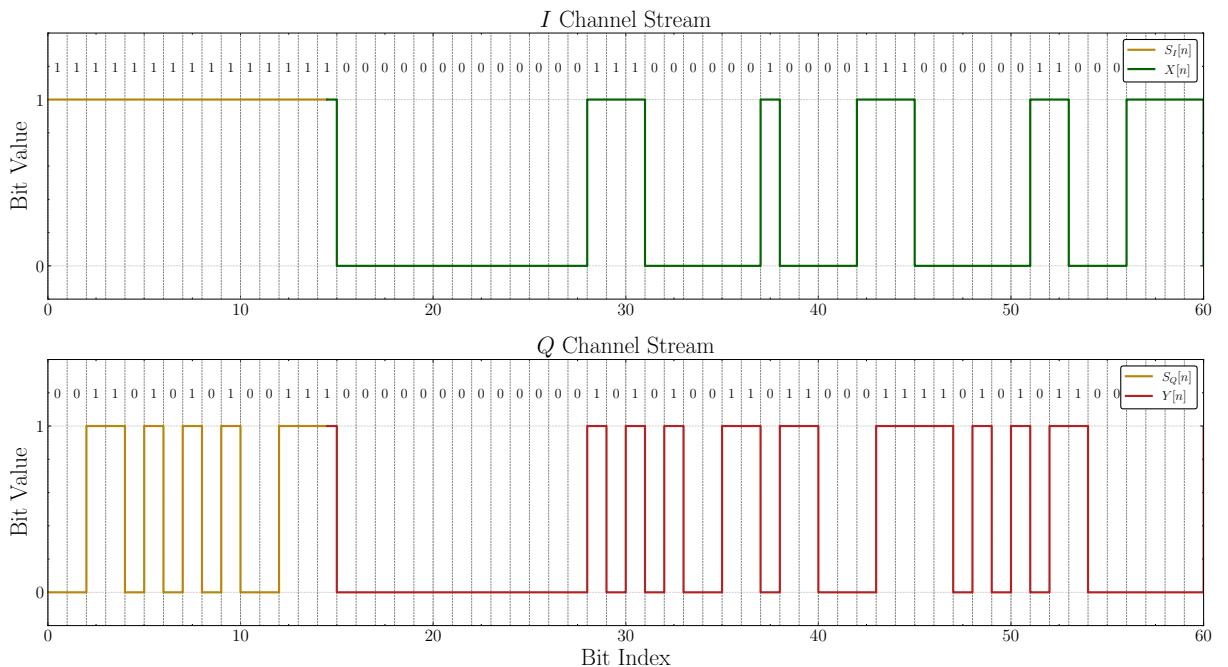
para os canais I e Q , respectivamente, o processo de codificação convolucional aplicado no vetor u_t é ilustrado na Figura 26.

Figura 26 – Codificação convolucional do datagrama ARGOS-3



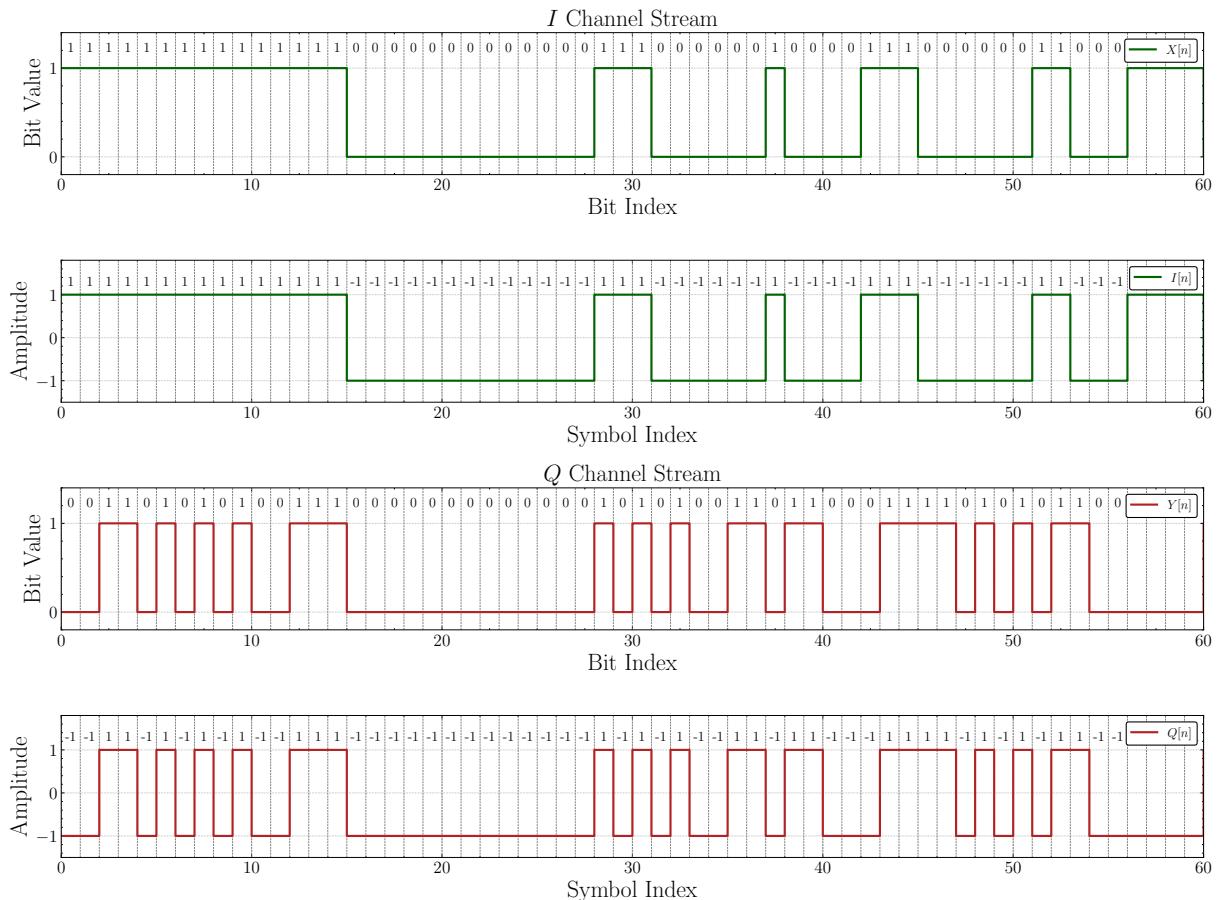
As sequências $X[n]$ e $Y[n]$ já embaralhadas, são então multiplexadas com o preâmbulo $S_I[n]$ e $S_Q[n]$ dos canais I e Q respectivamente, gerados conforme apresentado na seção 2.3.2. A adição das sequências é fundamental para permitir a sincronização de símbolos no receptor, esse processo é ilustrado na Figura 27.

Figura 27 – Multiplexação com preâmbulo



Em seguida, os bits são codificados utilizando codificação de linha para melhorar as características espectrais do sinal modulado, como detalhado na seção 2.5.3, neste processo é utilizado apenas codificação NRZ para os dois canais, pois a técnica Manchester será aplicada posteriormente na modulação de pulso do canal Q , o processo de codificação de linha de ambos os canais é ilustrado na Figura 28.

Figura 28 – Codificação de linha

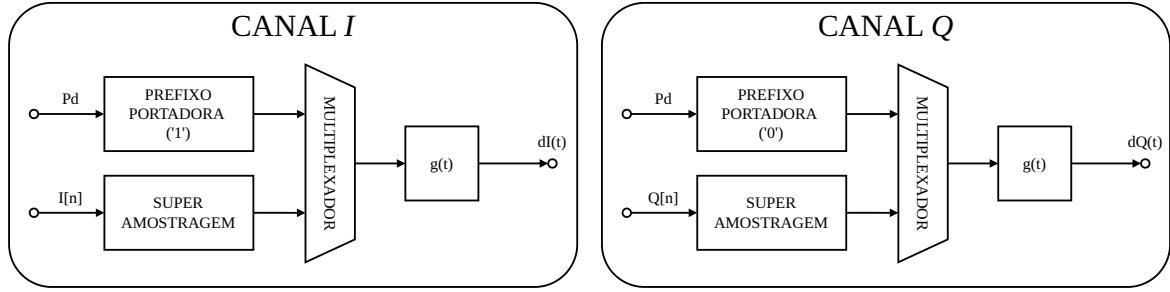


As sequências de símbolos $I[n]$ e $Q[n]$ resultantes da codificação de linha, estão prontas para serem moduladas utilizando modulação de pulso.

3.1.2 Modulação de pulso RRC e Manchester

Após a montagem das sequências de símbolos $I[n]$ e $Q[n]$, o próximo passo é a modulação de pulso, onde o canal I utiliza um filtro de pulso RRC e o canal Q utiliza um filtro de pulso Manchester, conforme detalhado na seção 2.5.3. A modulação de pulso é fundamental para reduzir a largura de banda W do sinal que está sendo transmitido e também reduzir a interferência entre símbolos ISI. O processo de modulação de pulso dos canais I e Q é ilustrado na Figura 29.

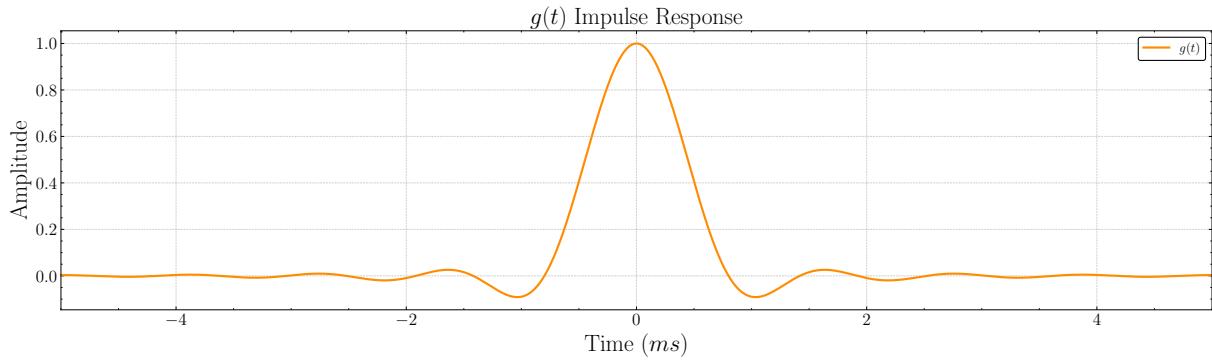
Figura 29 – Diagrama de blocos do modulador de pulso para os canais I e Q



3.1.2.1 Pulso RRC e Manchester

Antes de aplicar a modulação de pulso, é necessário gerar os filtros de pulso RRC e Manchester, que serão utilizados para modular os canais I e Q , respectivamente. A resposta ao impulso do filtro RRC é ilustrada na Figura 30, e é calculada conforme apresentado na seção 2.5.4.

Figura 30 – Resposta ao impulso - Pulso RRC



Já o pulso Manchester é gerado através de uma soma de dois pulsos RRC deslocados no tempo, um positivo deslocado em $T_b/2$ e outro negativo deslocado em $-T_b/2$, conforme ilustrado na Figura 31. O cálculo do pulso pode ser expresso como

$$g_{MAN}(t) = g_{RRC}(t + T_b/2) - g_{RRC}(t - T_b/2) \quad (3.1)$$

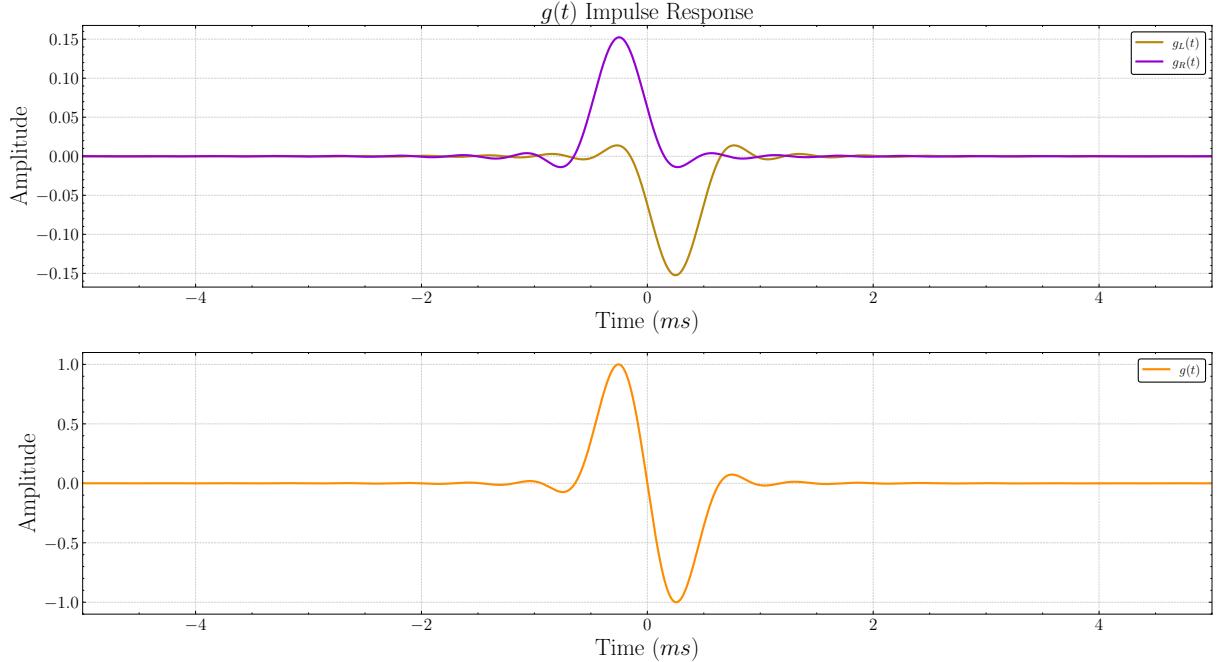
Onde $g_{MAN}(t)$ é o pulso Manchester, $g_{RRC}(t)$ é o pulso RRC e T_b é o tempo de bit. A banda do pulso Manchester é o dobro da banda do pulso RRC, seguindo a expressão

$$W_{MAN} = 2 \cdot \frac{1 + \alpha}{T_b} \quad (3.2)$$

Onde W_{MAN} é a largura de banda (unilateral) do pulso Manchester, α é o fator de roll-off do pulso RRC e T_b é o tempo de bit. Assim, a largura de banda W do sinal modulado em Manchester considerando um tempo de bits $T_b = 1/400$ e um fator de roll-off $\beta = 0.8$, é

de $720Hz$ (unilateral), ou seja, $2W = 1440Hz$ (bilateral). A resposta ao impulso do filtro Manchester é ilustrada na Figura 31.

Figura 31 – Resposta ao impulso - Pulso Manchester



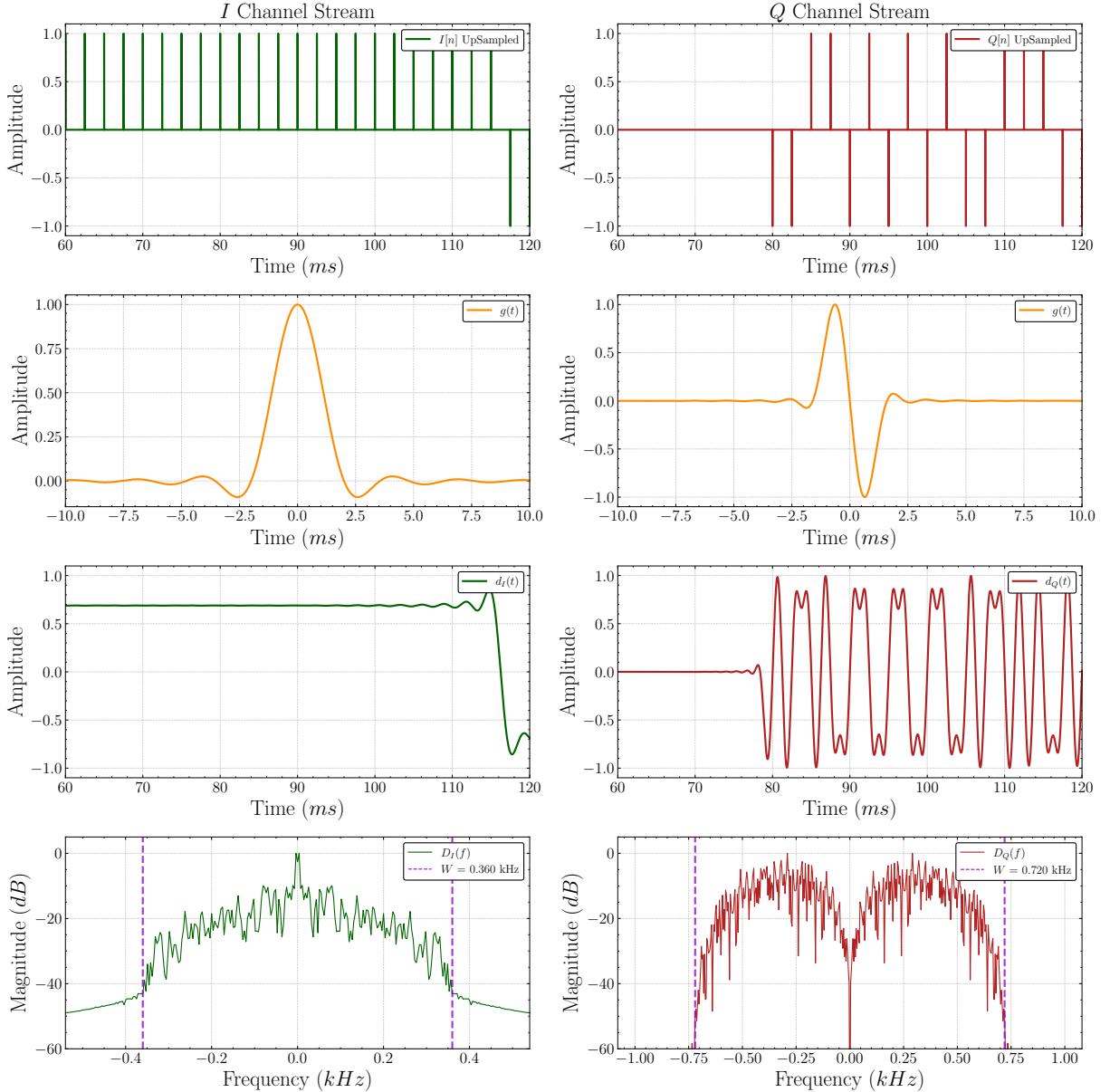
A utilização do pulso Manchester no canal Q ao invés da codificação de linha Manchester proposta originalmente tem como objetivo simplificar a implementação do sistema, reduzindo a complexidade computacional e o número de etapas necessárias na cadeia de transmissão.

3.1.2.2 Modulação de pulso dos canais I e Q

Uma vez com os filtros de pulso RRC e Manchester gerados, é possível aplicar a modulação de pulso nos canais I e Q , respectivamente gerando os sinais $d_I(t)$ e $d_Q(t)$ modulados em banda base. Esse processo envolve a superamostragem dos vetores de símbolos $I[n]$ e $Q[n]$, em função da frequência de amostragem f_s , seguida da filtragem com os respectivos filtros de pulso.

O resultado são as sequências $d_I(t)$ e $d_Q(t)$ contínuas ao longo do tempo t , onde cada bit de informação é transmitido durante um período de tempo $T_b = 1/R_b$ (tempo de bit), definido com base na taxa de bit R_b . A modulação de pulso dos canais I e Q é ilustrada na Figura 32.

Figura 32 – Modulação de pulso dos canais I e Q

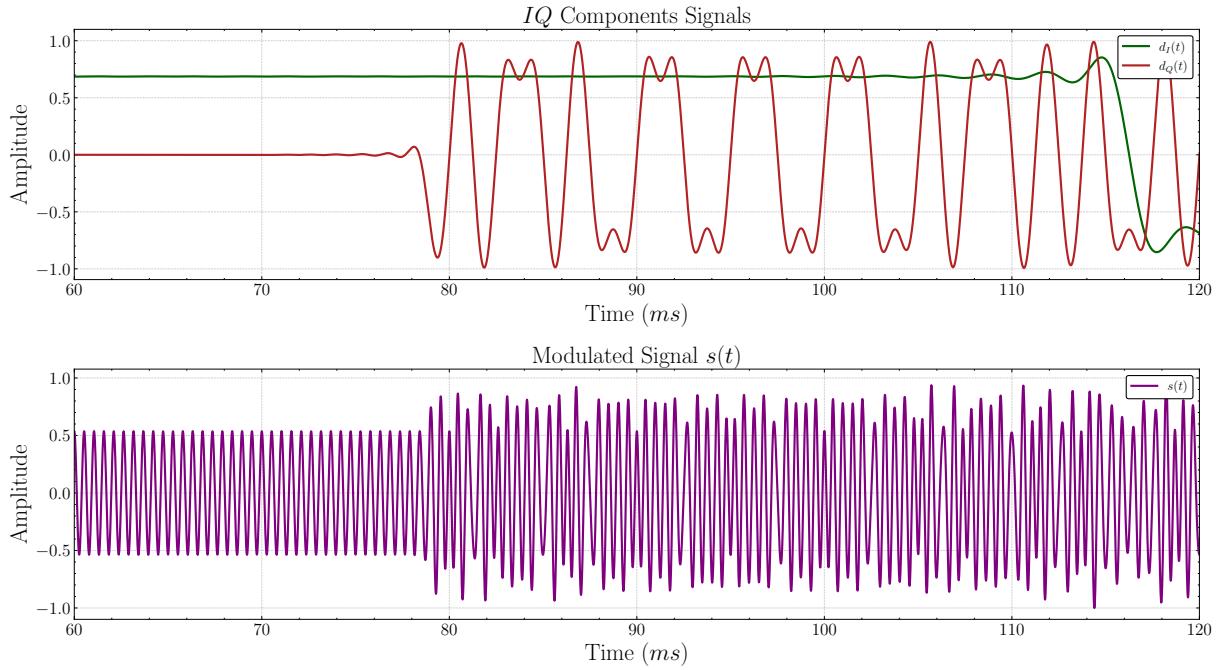


Além da mudança nas características do sinal no domínio do tempo, pode-se observar também a mudança no espectro do sinal, onde é possível observar a largura de banda W do sinal, conforme ilustrado detalhado na seção 2.5.4. As sequências $d_I(t)$ e $d_Q(t)$ estão agora prontas para serem moduladas em banda passante utilizando modulação em fase em quadratura (QPSK), conforme detalhado na seção 3.1.3.

3.1.3 Modulação em fase em quadratura (QPSK)

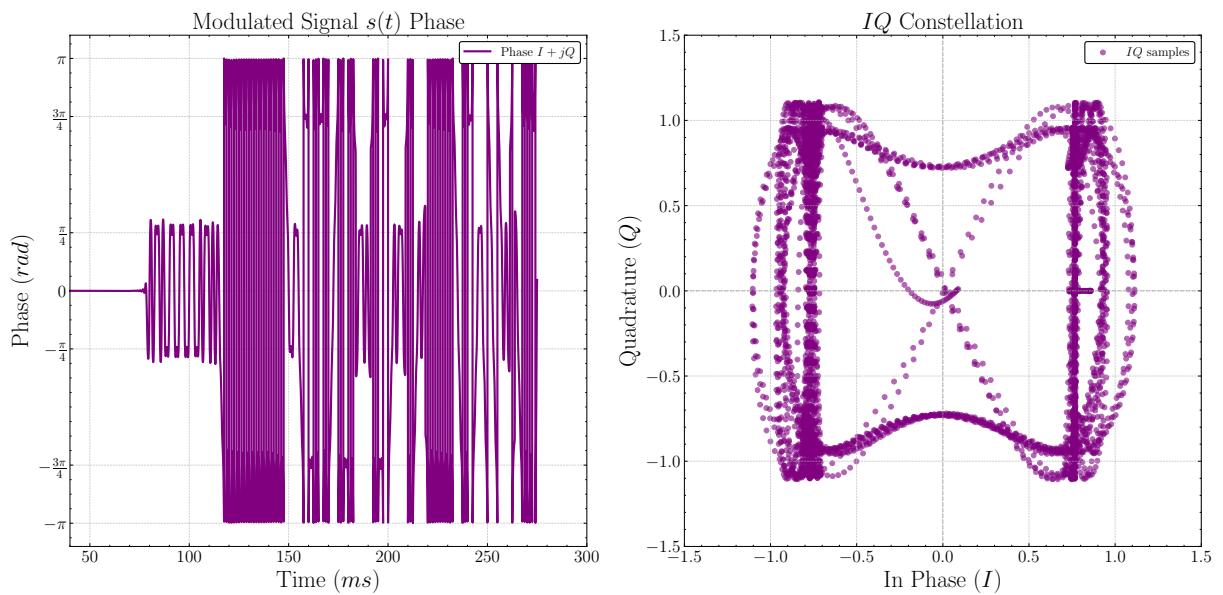
Na modulação QPSK, os sinais modulados em banda base $d_I(t)$ e $d_Q(t)$ são utilizados para modular uma componente senoidal e uma componente cossenoidal, respectivamente, com frequência f_c , o somatório dessas duas componentes moduladas resulta no sinal modulado em banda passante $s(t)$, conforme ilustrado na Figura 33.

Figura 33 – Modulação em banda passante canais I e Q



A partir do sinal modulado $s(t)$, é possível observar a constelação do sinal QPSK, isto é o comportamento do sinal $s(t)$ no *In-phase and Quadrature plane* (Plano IQ), que idealmente para o QPSK é composto por quatro pontos distintos, cada um representando uma combinação única dos bits transmitidos pelos canais I e Q . A constelação do sinal modulado $s(t)$ é ilustrada na Figura 34, onde é possível observar a fase do sinal modulado e a constelação resultante no Plano IQ.

Figura 34 – Fase e Constelação do sinal modulado $s(t)$



3.1.3.1 Adição de portadora pura

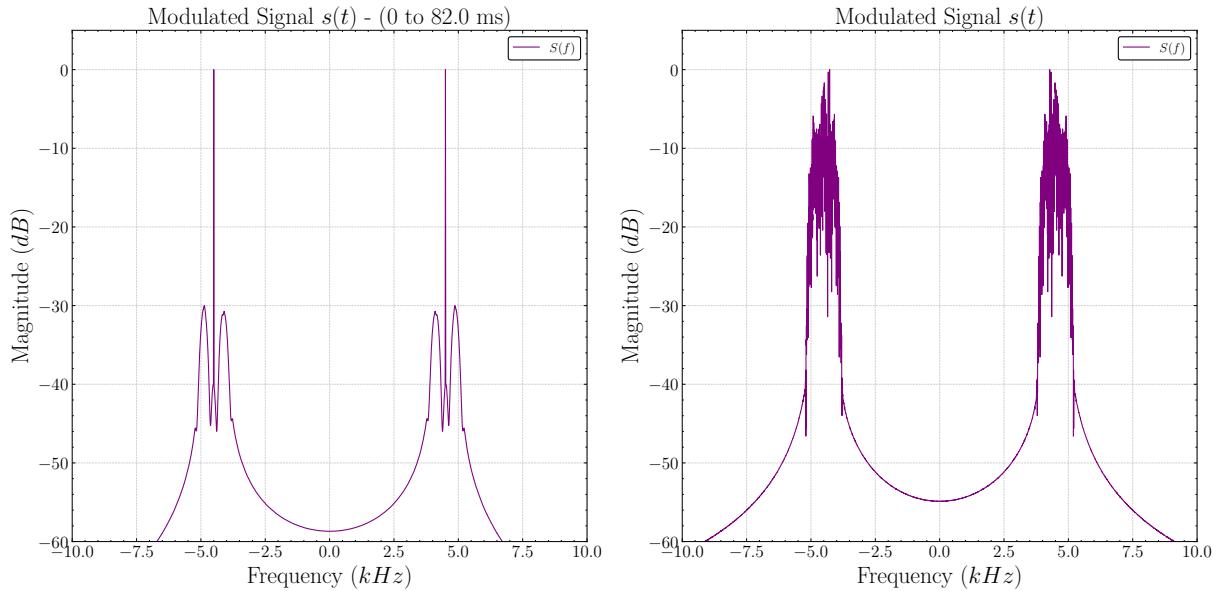
Ao se observar as figuras 33 e 32 pode-se notar que no inicio da transmissão, a componente $d_I(t)$ está com amplitude estável e proxima á ‘1‘, enquanto que a componente $d_Q(t)$ está estável com amplitude igual a ‘0‘, essa é configuração é proposital dentro do P_d para que a sequência resultante modulada em banda passante tenha um periodo de fase estável, ou seja, uma portadora pura, antes do inicio da transmissão dos dados. O equacionamento para o período de portadora pura pode ser expresso como

$$s(t) = 1(t) \cdot \cos(2\pi f_c t) - 0(t) \cdot \sin(2\pi f_c t) \mapsto s(t) = \cos(2\pi f_c t) \quad (3.3)$$

Onde $1(t)$ é a componente $d_I(t)$ com valor constante ‘1‘ e $0(t)$ é a componente $d_Q(t)$ com valor constante ‘0‘. O período de portadora pura é fundamental para o receptor identificar a frequênciada portadora f_c e realizar a detecção do sinal corretamente, conforme detalhado na seção 3.3. A duração do período de portadora pura é definida pelo parâmetro P_d , que no padrão ARGOS-3 é aproximadamente 0.082 segundos.

Com o sinal modulado em banda passante $s(t)$ conforme definido acima, podemos também verificar a presença da portadora pura nos primeiros instantes da transmissão, isto é ilustrado na figura 35.

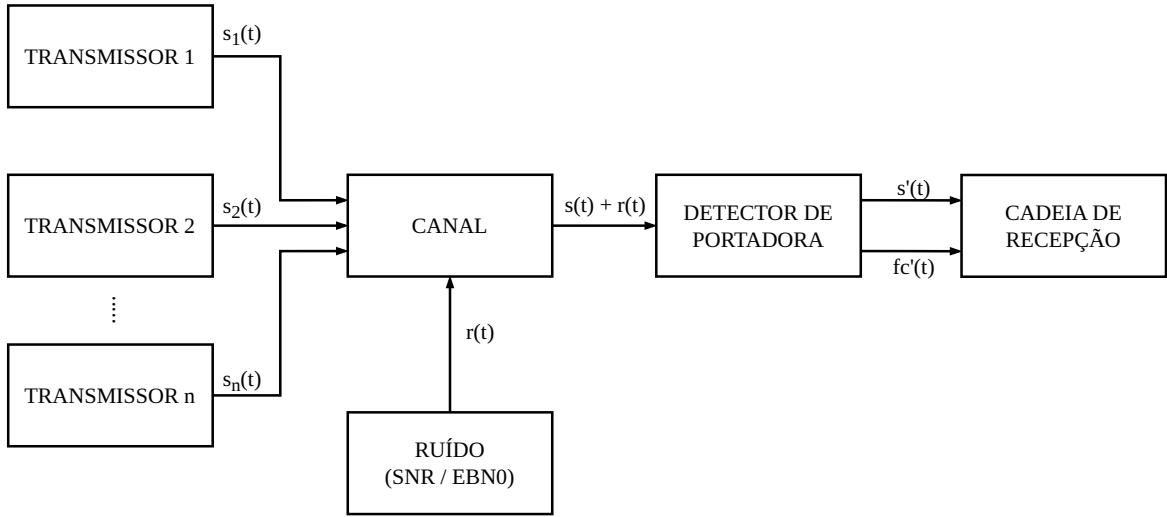
Figura 35 – Comparaçāo de portadora pura e sinal modulado



3.2 CANAL E ADIÇÃO DE RUÍDO

Após a modulação do sinal $s(t)$ em banda passante, o próximo passo é a transmissão através do canal de comunicação, que permite a adição de múltiplas transmissões simultâneas e a introdução de ruído *Additive White Gaussian Noise* (AWGN) ao sinal transmitido. O diagrama de blocos do canal é ilustrado na Figura 36.

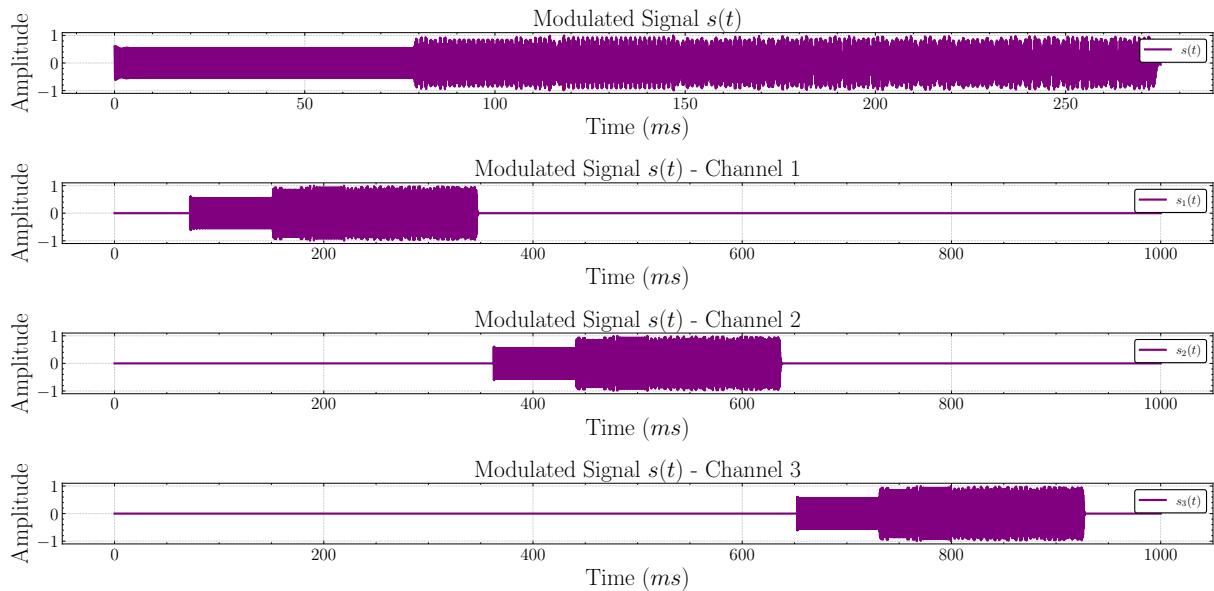
Figura 36 – Diagrama de blocos do canal



3.2.1 Modelo de canal

O modelo de canal é instânciado com base na f_s e na duração desejada para o canal, padrão 1 segundo. A partir disto, é possível calcular a quantidade de amostras necessárias para compor o canal, e então criar um vetor de zeros com o tamanho adequado, que servirá como base para a adição das transmissões. O modelo de canal é ilustrado na Figura 37.

Figura 37 – Adição de múltiplas transmissões no canal



Em seguida, cada transmissão é adicionada ao canal junto a fator de posição desejado, fator que pode variar entre 0 e 1, onde 0 representa o início do canal e 1 representa o final do canal. A posição de cada transmissão é calculada com base no fator de posição e na quantidade total de amostras do canal.

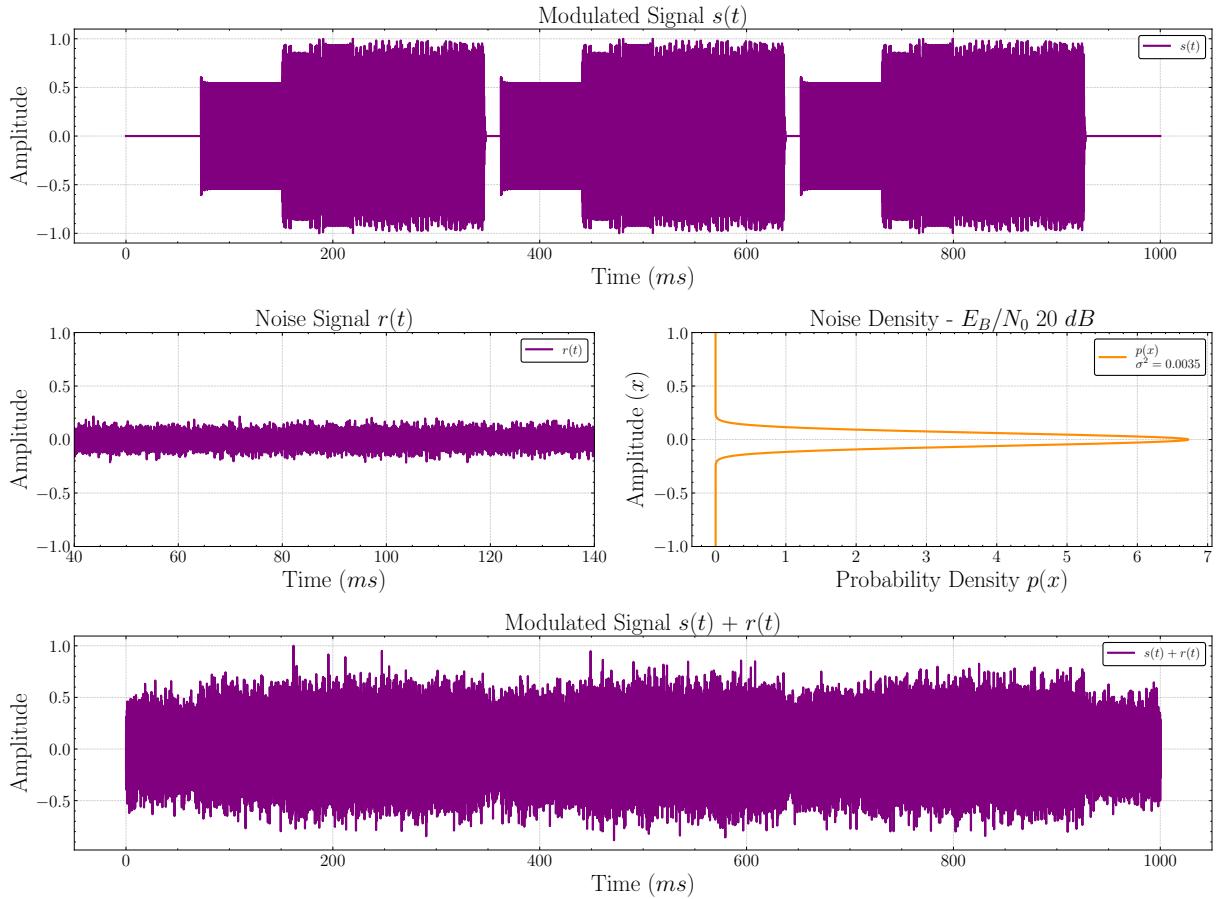
3.2.2 Geração de ruído AWGN

Uma vez com as transmissões adicionadas ao canal, o próximo passo é a geração do ruído AWGN com base na SNR ou E_b/N_0 desejada. O ruído é gerado como um vetor de números aleatórios com distribuição normal, com média zero e variância calculada com base no tipo de ruído utilizado e na potência do sinal transmitido. O cálculo da σ^2 utilizada na geração do ruído SNR pode ser expresso como

$$\sigma^2 = \frac{\mathbb{E}[|s(t)|^2]}{10^{\frac{\text{SNR}_{dB}}{10}}} \quad (3.4)$$

Onde $\mathbb{E}[|s(t)|^2]$ é a potência do sinal transmitido e SNR_{dB} é a relação sinal-ruído em decibéis. A geração e adição do ruído $r(t)$ ao sinal $s(t)$ é ilustrada na Figura 38.

Figura 38 – Adição de ruído ao canal

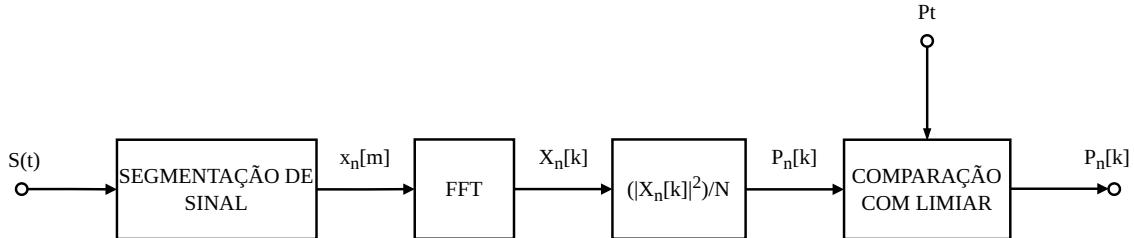


3.3 DETECÇÃO DE PORTADORA

Com as múltiplas transmissões e o ruído adicionados ao canal, o próximo passo é a detecção das portadoras presentes no canal para posterior recepção. A detecção de portadora é realizada através de um processo de segmentação do sinal recebido em segmentos $x_n[m]$, análise do espectro de cada segmento para identificação de componentes

com potência acima de um limiar definido P_t , e decisão sobre quais componentes serão consideradas como portadoras válidas. O diagrama de blocos do detector é ilustrado na Figura 39.

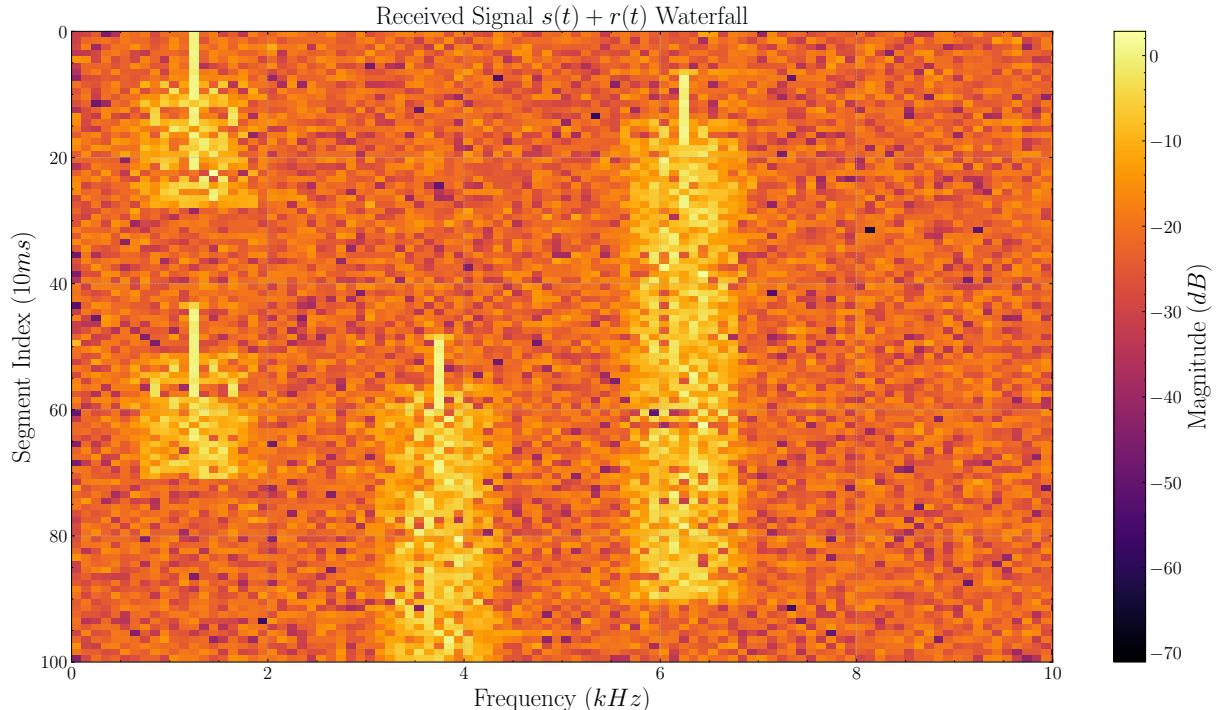
Figura 39 – Diagrama de blocos da detecção de portadora



3.3.1 Segmentação do sinal recebido

O primeiro passo na detecção de portadora é a segmentação do sinal recebido $r(t)$ em segmentos $x_n[m]$ de duração T_m , conforme detalhado na seção 3.3. Cada segmento é então analisado individualmente para identificar a potência $P_n[k]$ de cada componente k de cada segmento m , formando assim uma matriz $m \times k$, ou $x[k, m]$, podendo ser representado como um diagrama waterfall, conforme ilustrado na Figura 40.

Figura 40 – Diagrama de waterfall do canal

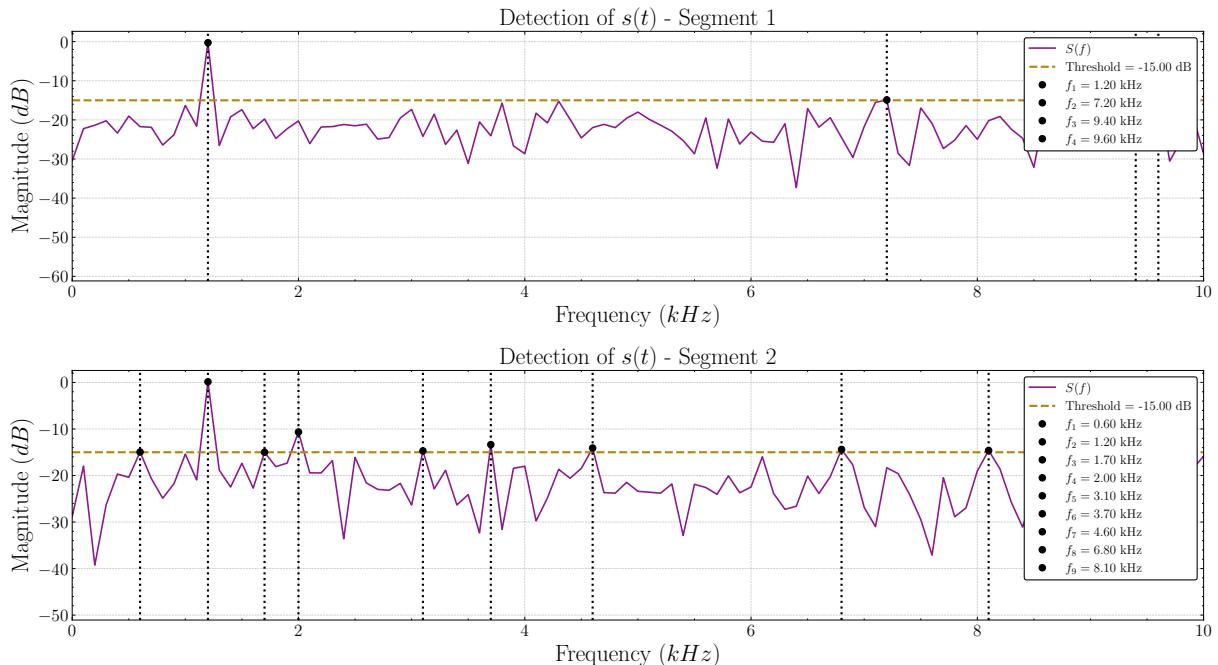


Na Figura 40 é possível observar claramente a presença das transmissões realizadas, além disso, nota-se que no inicio de cada transmissão uma sequência de altas componentes, formando uma linha, que corresponde ao período de portadora pura P_d , conforme detalhado na seção 3.1.3.

3.3.2 Detecção de componentes no espectro

Para cada segmento $x_n[m]$, as componentes $P_n[k]$ são comparadas com o limiar de potência P_t , e aquelas que apresentarem potência acima do limiar são consideradas como componentes detectadas. O resultado dessa comparação é uma matriz $m \times k$, onde cada elemento indica se a componente k do segmento m foi detectada ou não. A detecção de componentes no espectro com base em P_t é ilustrada na Figura 41.

Figura 41 – Detecção de componentes no espectro com base em P_t

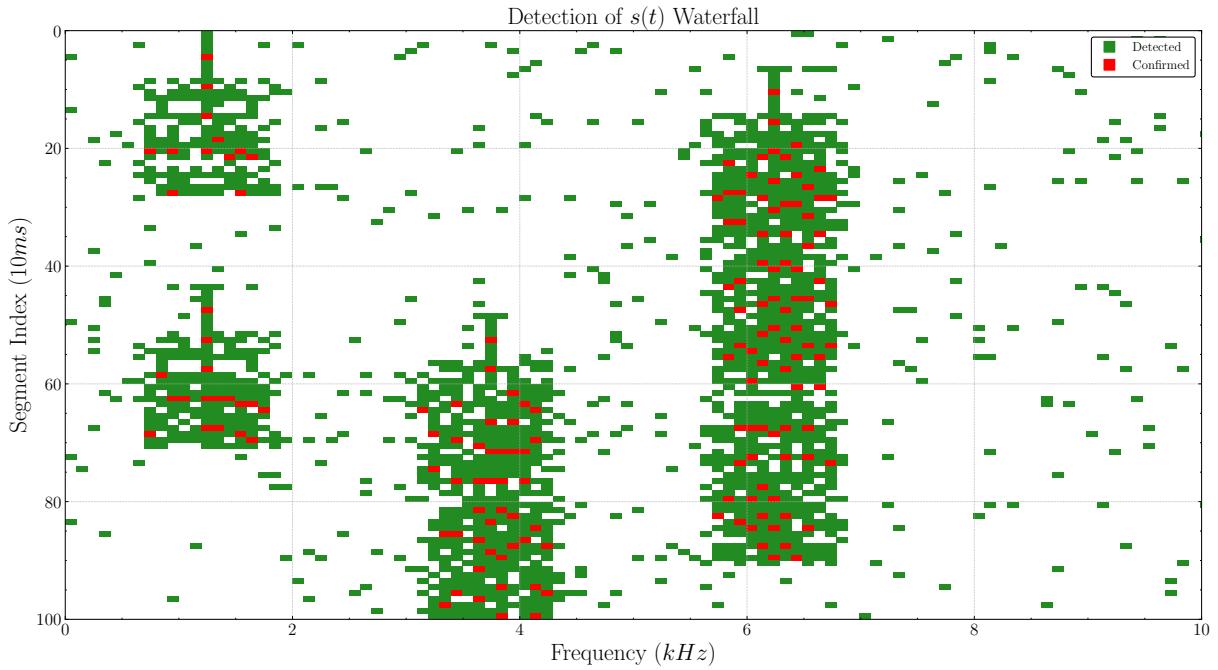


Além disso, quando uma componente é detectada múltiplas vezes consecutivas, essa componente é considerada como "confirmada", ou seja, é uma componente que provavelmente corresponde a uma portadora válida, a quantidade de vezes consecutivas que uma componente deve ser detectada para ser considerada como confirmada é definida pelo parâmetro N_c , esse processo pode ser expresso como

$$x[k, m](\text{confirmada}) = \begin{cases} 1, & \text{se } \sum_{i=-N_c+1}^0 x[k, m+i] = N_c \\ 0, & \text{caso contrário} \end{cases}, \quad (3.5)$$

Onde $x[k, m](\text{confirmada})$ é a componente que foi confirmada no índice k , $x[m, k]$ é a componente detectada no segmento m e índice k , e N_c é o número de confirmações consecutivas necessárias para considerar a componente como confirmada. A matriz de componentes detectadas e confirmadas pode ser representada como um diagrama waterfall, conforme ilustrado na Figura 42.

Figura 42 – Diagrama de waterfall de detecção do canal



No simulador, N_c é fixado em 4, por ser metade do comprimento da portadora pura, entretanto, quanto maior o valor de N_c , maior a certeza de que a componente detectada é realmente uma portadora válida portanto maior a tolerância da decisão a P_t mais baixos, porém, isso também pode aumentar falhas de detecção, caso o período de amostragem do detector não seja suficiente para conformar a transmissão.

3.3.3 Decisão de componentes detectadas

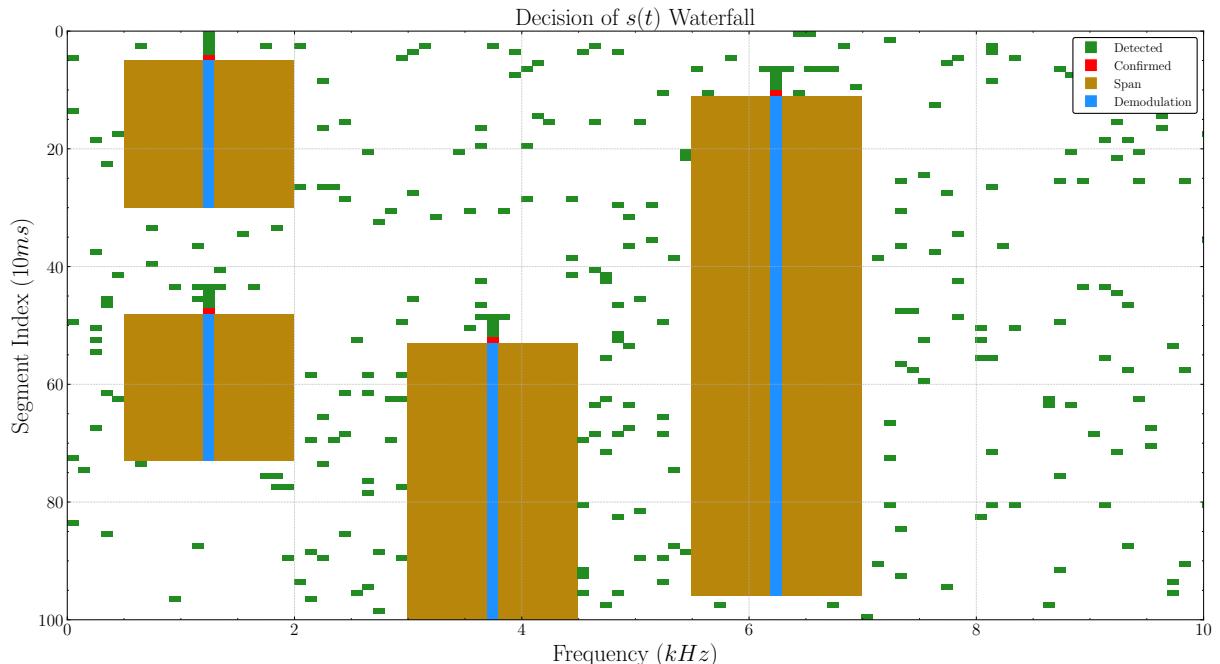
Com base na matriz de componentes confirmadas, o próximo passo é a decisão sobre quais componentes serão efetivamente consideradas como transmissões válidas para recepção. A decisão é baseada na continuidade da detecção de componentes ao longo dos segmentos, assim a primeira componente confirmada em $k = f'_c$ no segmento $m = S_{ini}$ é considerada como o inicio de uma transmissão.

A partir dela, é aplicado um "span", isto é, uma proteção para ignorar qualquer detecção, de largura W sobre amostras em $f'_c - W$ e $f'_c + W$, do segmento S_{ini} até o fim das detecções sobre a frequência f'_c com o objetivo de evitar falsas recepções do próprio sinal que já foi confirmado sobre S_{ini} . A aplicação do span sobre as amostras detectadas é dada por

$$x'[k, m] = \begin{cases} 0, & \text{se } (f'_c - W) < k < (f'_c + W) \text{ e } x[f'_c, m] = 1 \\ x[k, m], & \text{caso contrário.} \end{cases}, \quad (3.6)$$

Onde $x[k, m]$ é a componente no indice k e segmento m (valido para qualquer m a partir de S_{ini}), f'_c é a frequêcia da portadora detectada, e W é a largura de banda do sinal transmitido. A decisão de componentes detectadas pode ser representada como um diagrama waterfall, conforme ilustrado na Figura 43.

Figura 43 – Diagrama de waterfall de decisão do canal



3.3.4 Segmentação do sinal recebido para recepção

Para cada componente f'_c decidida como portadora válida, tendo seu segmento de inicio e seu segmento de fim, o próximo passo é a segmentação do sinal recebido $s(t)$ para recepção em segmentos de sinal $s'(t)$. Cada segmento $s'(t)$ é então processado individualmente na cadeia de recepção com sua respectiva f'_c , conforme detalhado na seção 3.4. O processo de segmentação do sinal recebido para recepção é apresentado no código de exemplo 3.1.

Código 3.1 – Recepção de sinal $s'(t)$

```

1 import argos3
2 import numpy as np
3
4 # Frequêcia da portadora (entre 1kHz e 8kHz)
5 fc = np.random.randint(10,80)*100
6
7 # Exibe a frequêcia da portadora
8 print(fc)
9 2400
10
11 # Cria o transmissor e transmite o datagrama gerando o sinal s(t)

```

```

12 transmitter = argos3.Transmitter(fc=fc)
13 t, s = transmitter.transmit(argos3.Datagram(pcdnum=1234, numblocks=1))
14
15 # Cria o canal com duração de 1 segundo
16 channel = argos3.Channel(duration=1, noise_mode="ebn0", noise_db=20)
17 # Adiciona o sinal no meio do canal
18 channel.add_signal(s, position_factor=0.5)
19 channel.add_noise()
20
21 # Cria o detector de portadora e detecta as transmissões contidas no canal
22 detector = argos3.CarrierDetector(seg_ms=10, threshold=-15)
23 detector.detect(channel.channel.copy())
24 detections = detector.return_channels()
25
26 # Exibe as detecções no formato (fc, start_seg, end_seg)
27 print(detections)
28 [(np.float64(2400.0), 41, 65)]
29
30 # Isola o sinal recebido  $s'(t)$  a partir das amostras de início e fim da detecção
31 first_sample = int((detections[0][1] - detector.history) * detector.fs * detector.
    seg_s)
32 last_sample = int(detections[0][2] * detector.fs * detector.seg_s)
33 st_prime = channel.channel[first_sample:last_sample]
34
35 # Cria o receptor, recebe o sinal  $s'(t)$  e tenta decodificar o datagrama
36 receiver = argos3.Receiver(fc=detections[0][0])
37 datagramRX, success = receiver.receive(st_prime)
38 print(success)
39 True

```

3.4 CADEIA DE RECEPÇÃO

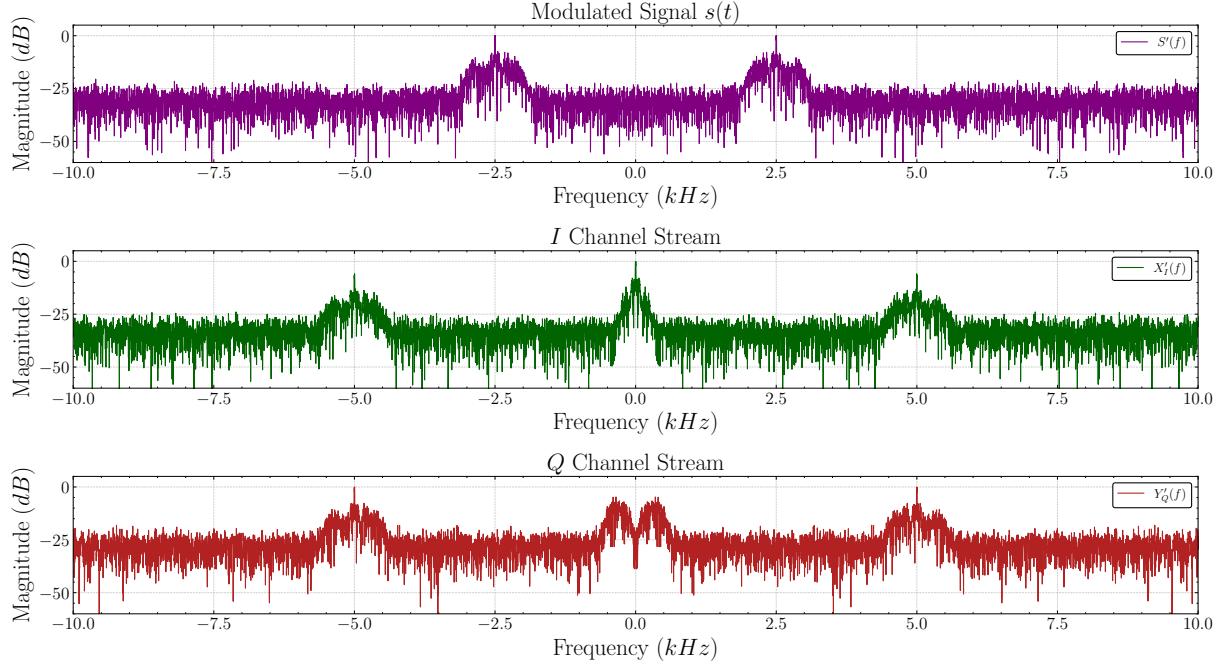
A cadeia de recepção é responsável por recuperar o datagrama transmitido a partir do sinal recebido $s'(t)$, que foi segmentado e identificado como uma portadora válida na etapa de detecção de portadora, conforme detalhado na seção 3.3. A cadeia de recepção é dividida em demodulação em banda base, filtragem, sincronização de símbolos, decisão de símbolos e recuperação do datagrama conforme apresentado na seção 2.6

3.4.1 Demodulação banda base

O primeiro passo na cadeia de recepção é a demodulação do sinal recebido $s'(t)$ para retornar o sinal para banda base. Para isso, é necessário injetar uma componente senoidal e cossenoidal no $s'(t)$ com a mesma frequência f_c' da portadora detectada, fazendo com que o sinal seja transladado, isto é, deslocado para banda base formando as componentes

$d'_X(t)$ e $d'_Y(t)$, o processo de demodulação também faz com que uma componente identica seja gerada na frequência $2 \cdot f'_c$, esse processo é ilustrado na Figura 44.

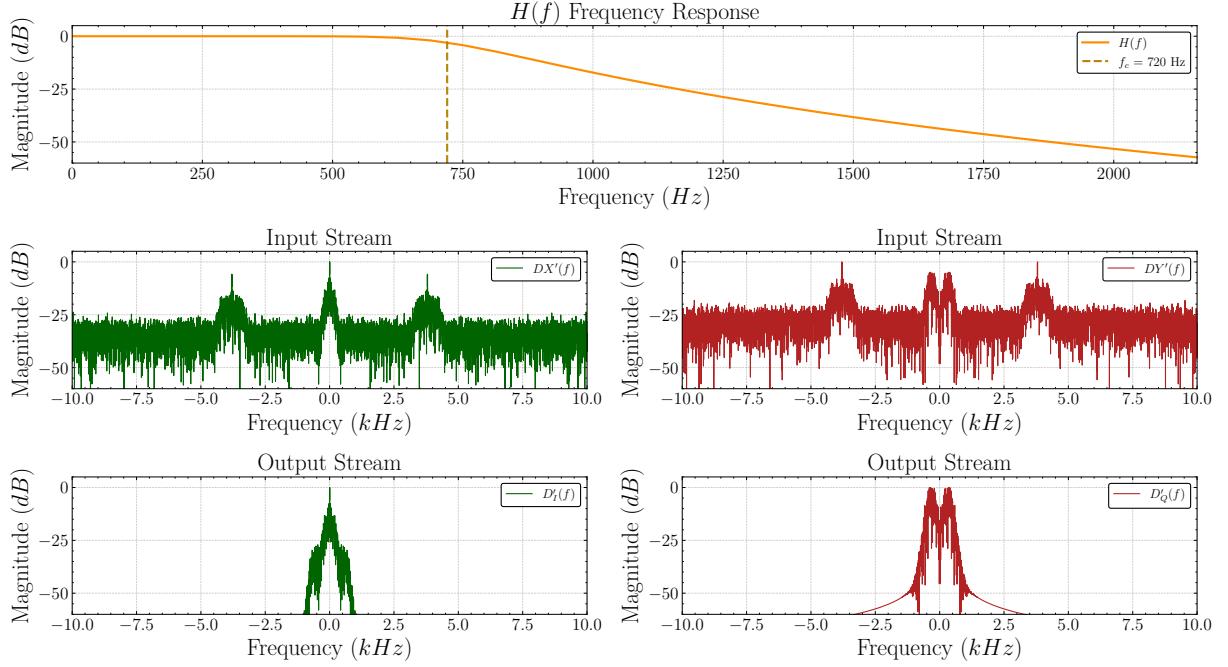
Figura 44 – Demodulação banda base dos canais I e Q



3.4.2 Filtragem

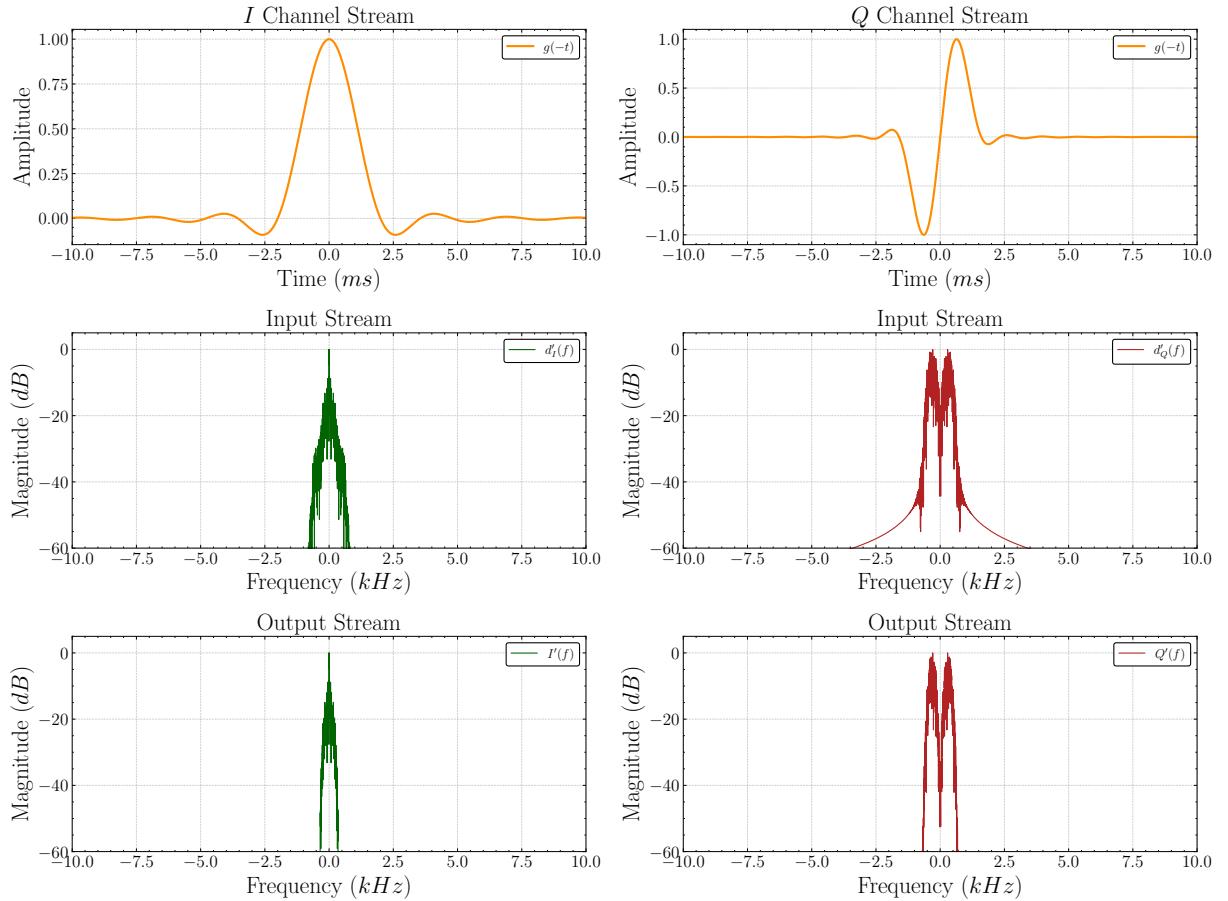
Após a demodulação, o próximo passo é a filtragem dos canais I e Q para remover as componentes de alta frequência geradas no processo de demodulação, bem como reduzir o máximo possível ruído presente em cada canal. A filtragem é realizada em duas etapas, primeiramente uma filtragem passa baixa para remover as componentes de alta frequência, conforme detalhado na seção 2.6.3, resultando nos sinais $d'_I(t)$ e $d'_Q(t)$, conforme ilustrado na Figura 45.

Figura 45 – Filtragem passa baixa dos canais I e Q



Em seguida, os sinais $d'_I(t)$ e $d'_Q(t)$ já filtrados são submetidos outra filtragem, com o objetivo de maximizar a SNR do sinal, para isso é aplicada uma filtragem casada utilizando os mesmos filtros de pulso utilizados na modulação de pulso porém invertidos no tempo, ou seja, com resposta ao impulso $g(-t)$, conforme detalhado na seção 2.6.3, resultando nos sinais $I'(t)$ e $Q'(t)$. O processo de filtragem casada é ilustrado na Figura 46.

Figura 46 – Filtragem casada dos canais I e Q

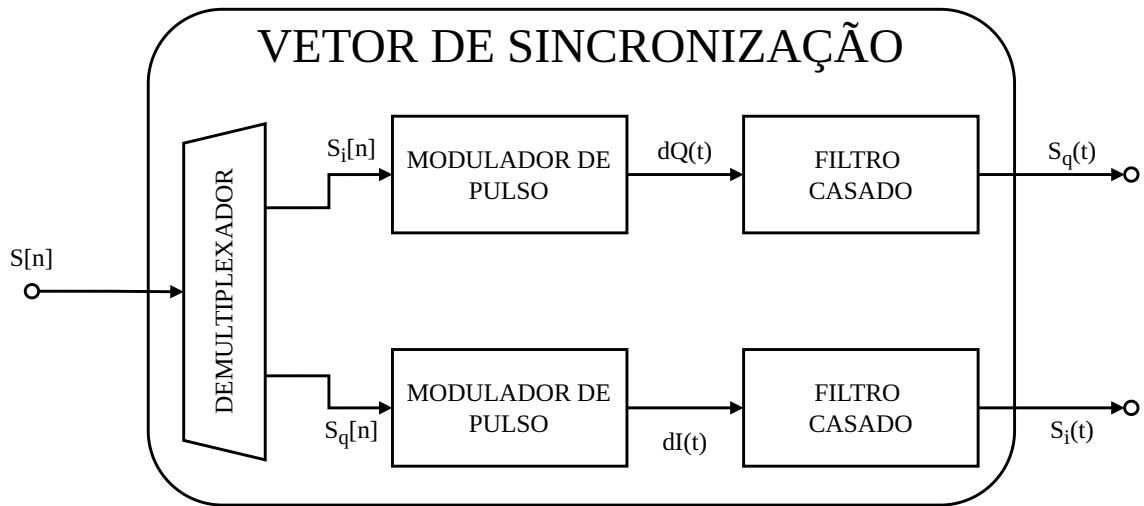


Assim, os sinais $I'(t)$ e $Q'(t)$ estão prontos para a sincronização de símbolos, conforme detalhado na seção 3.4.3.

3.4.3 Sincronização de símbolos

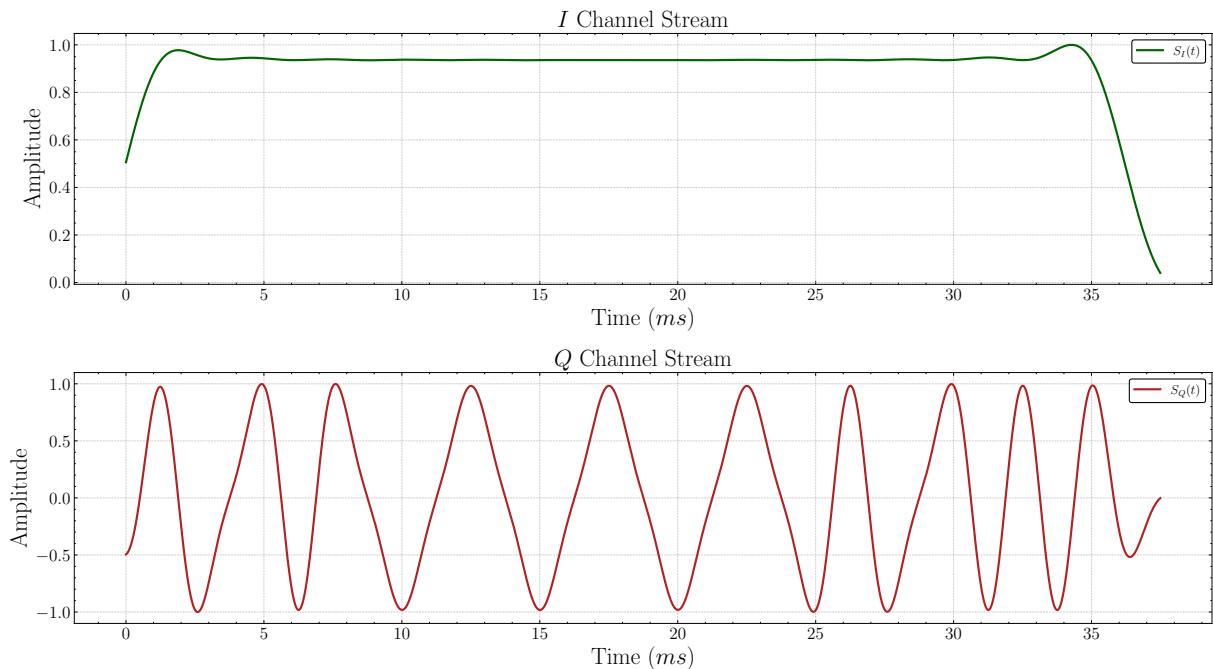
Para poder recuperar os bits transmitidos, é necessário realizar a sincronização de símbolos para saber o delay inicial de amostragem τ , e então utilizar esse valor para iniciar a amostragem com tempo de T_b até o fim do sinal. Para a sincronização ocorrer, precisa-se iniciamente saber qual sequência esperar, e para isso deve-se montar o vetor de amostras esperadas para o sincronismo, esse processo é ilustrado na Figura 47.

Figura 47 – Diagrama de blocos para montagem do vetor de sincronismo



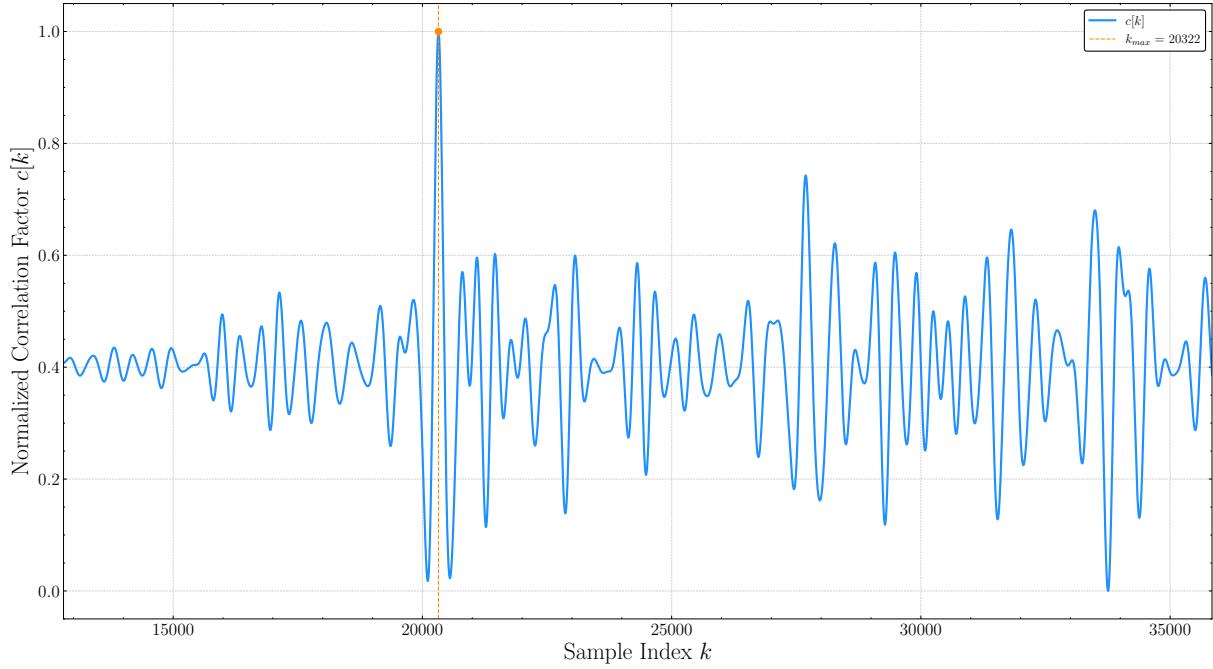
Após a montagem do vetor de amostras esperadas para o sincronismo, que consiste em, gerar a sequência, superamostra-la, aplicar formatação de pulso nos canais I e Q , e em seguida filtragem casada, obtém-se o vetor de amostras esperadas para o sincronismo de cada canal, conforme ilustrado na Figura 48.

Figura 48 – Vetor de amostras esperadas para sincronismo com I e Q



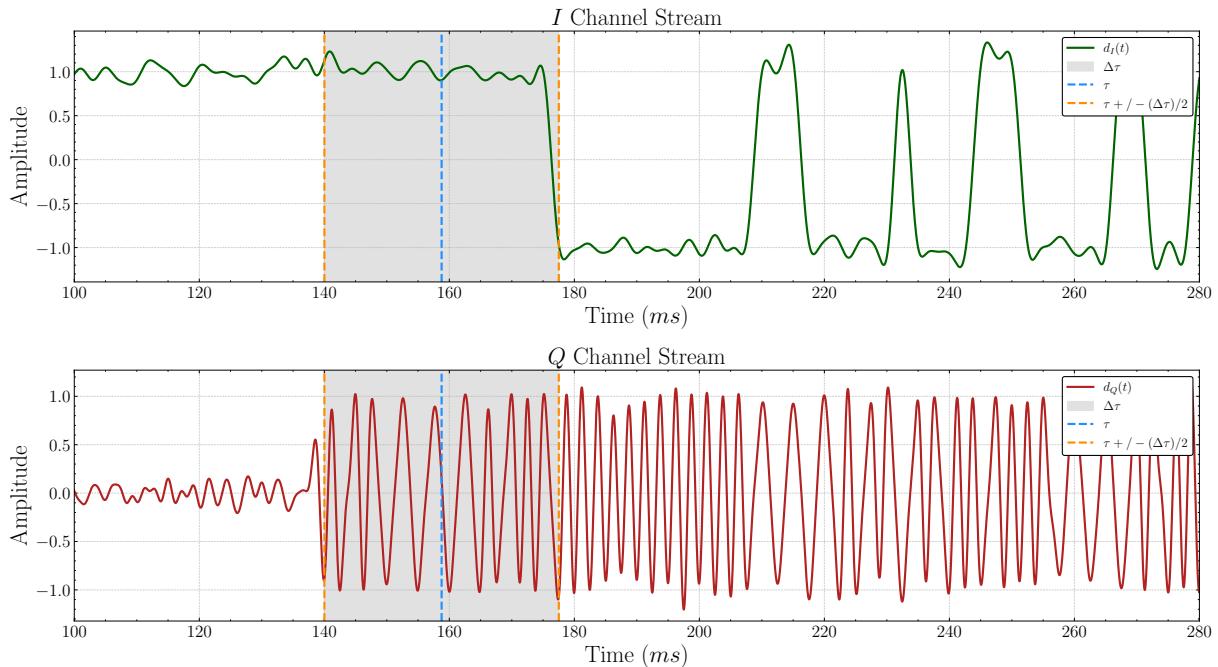
Com o vetor de amostras esperadas para o sincronismo, é possível realizar a correlação, isto é, comparar o quanto se parece uma seção dos canais I e Q com o vetor de sincronismo, resultando no módulo de correlação, onde o pico máximo do módulo de correlação k_{max} indica o ponto de início da transmissão. O processo de correlação para encontrar a amostra de máxima correlação é ilustrado nas Figura 49 e Figura 50.

Figura 49 – Módulo de correlação entre I e Q e vetor de sincronismo



Uma vez com o fator máximo de correlação, podemos facilmente determinar o delay inicial de amostragem como $\tau = f_s/k_{\max}$, visto que a f_s é conhecida. Com o valor de τ determinado, os canais I e Q estão prontos para serem amostrados a cada T_b a partir do ponto inicial τ . O processo de sincronização de símbolos é ilustrado na Figura 50.

Figura 50 – Sincronização de I e Q com vetor de sincronismo



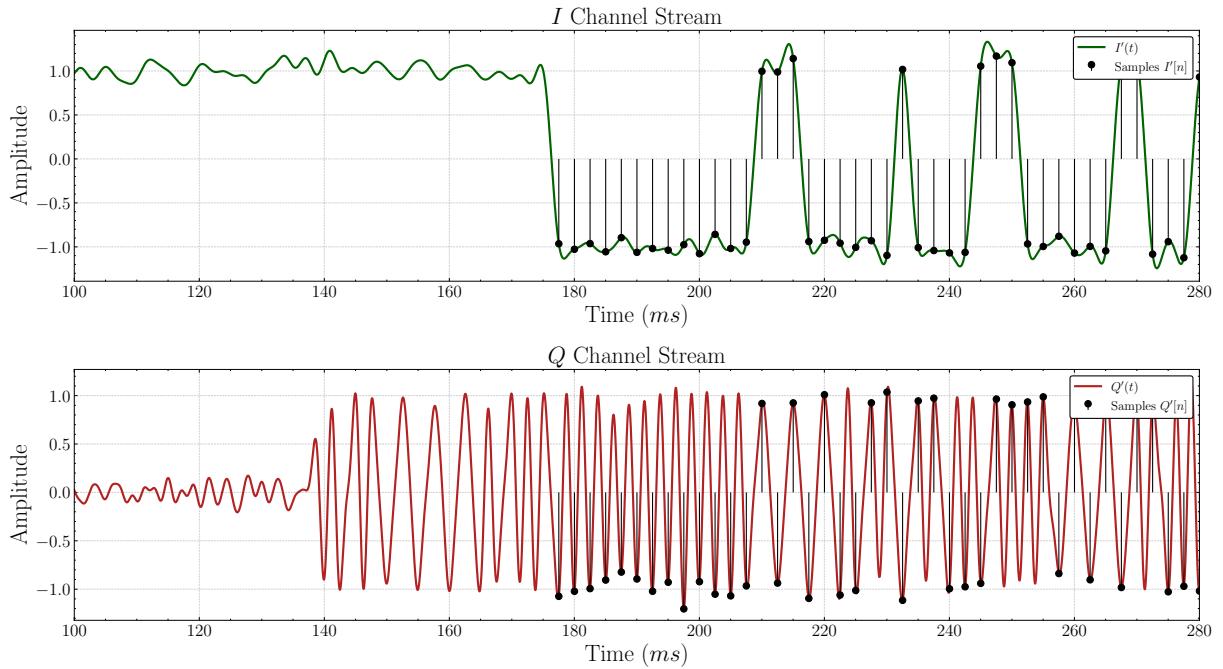
Na Figura 50 a região em cinza representa o período de preâmbulo, onde é possível observar que o pico máximo de correlação ocorre exatamente na metade do preâmbulo,

ou seja, deve-se iniciar a amostragem a partir de $\tau + (\Delta\tau)/2$, onde $\Delta\tau$ é o comprimento do preâmbulo em amostras ($\Delta\tau = L_{sync} \cdot f_s / R_b$).

3.4.4 Decisão de símbolos

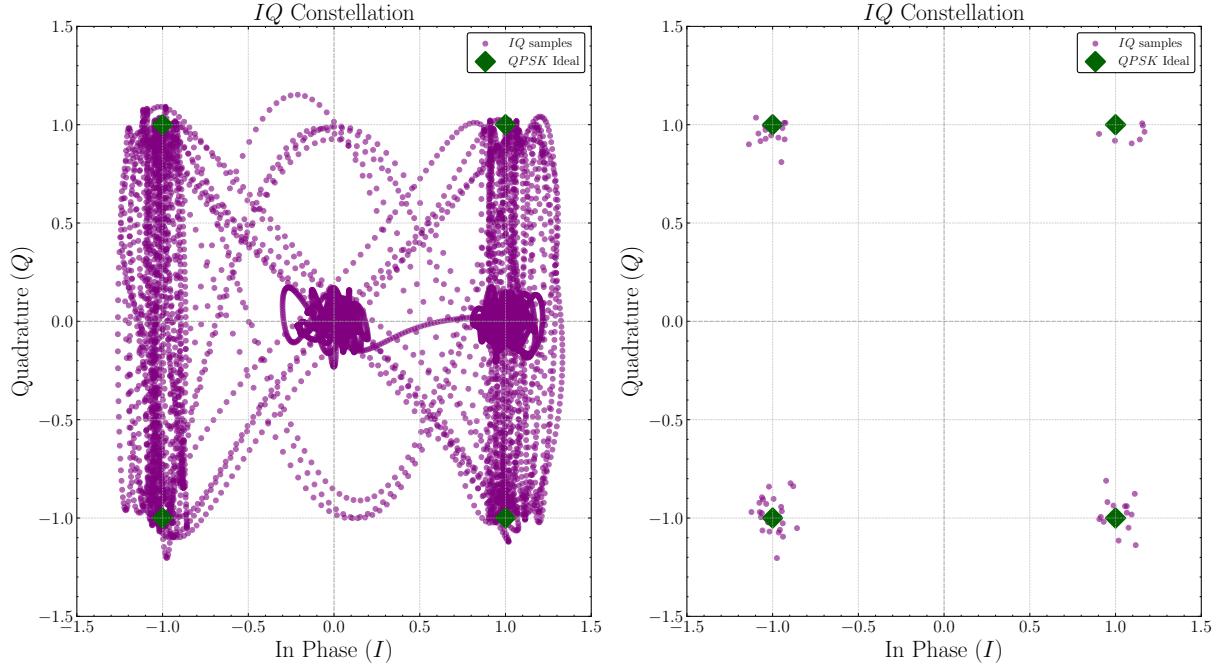
Com o valor de τ determinado, os canais I e Q estão prontos para serem amostrados a cada T_b a partir do ponto inicial τ , resultando nos vetores de amostras $I'[n]$ e $Q'[n]$. A decisão de símbolos é realizada através da comparação das amostras com um limiar definido, onde valores positivos são interpretados como ‘1’ e valores negativos como ‘0’. O processo de amostragem e decisão de símbolos é ilustrado nas Figura 51 e Figura 52.

Figura 51 – Amostragem dos canais I e Q em T_b a partir de τ



Tendo os sinais $d'_I(t)$ e $d'_Q(t)$ e seus correspondentes vetores de amostras $I'[n]$ e $Q'[n]$, é possível observar a constelação dos canais I e Q no Plano IQ, antes e após a amostragem, conforme ilustrado na Figura 52, nota-se que na direita da imagem, a constelação é plotada de maneira contínua, ou seja, cada ponto representa um instante de tempo t , enquanto que na esquerda da imagem, a constelação é plotada apenas com os pontos amostrados, ou seja, cada ponto representa um símbolo sobre o tempo T_b .

Figura 52 – Constelação dos canais I e Q filtrados e amostrados

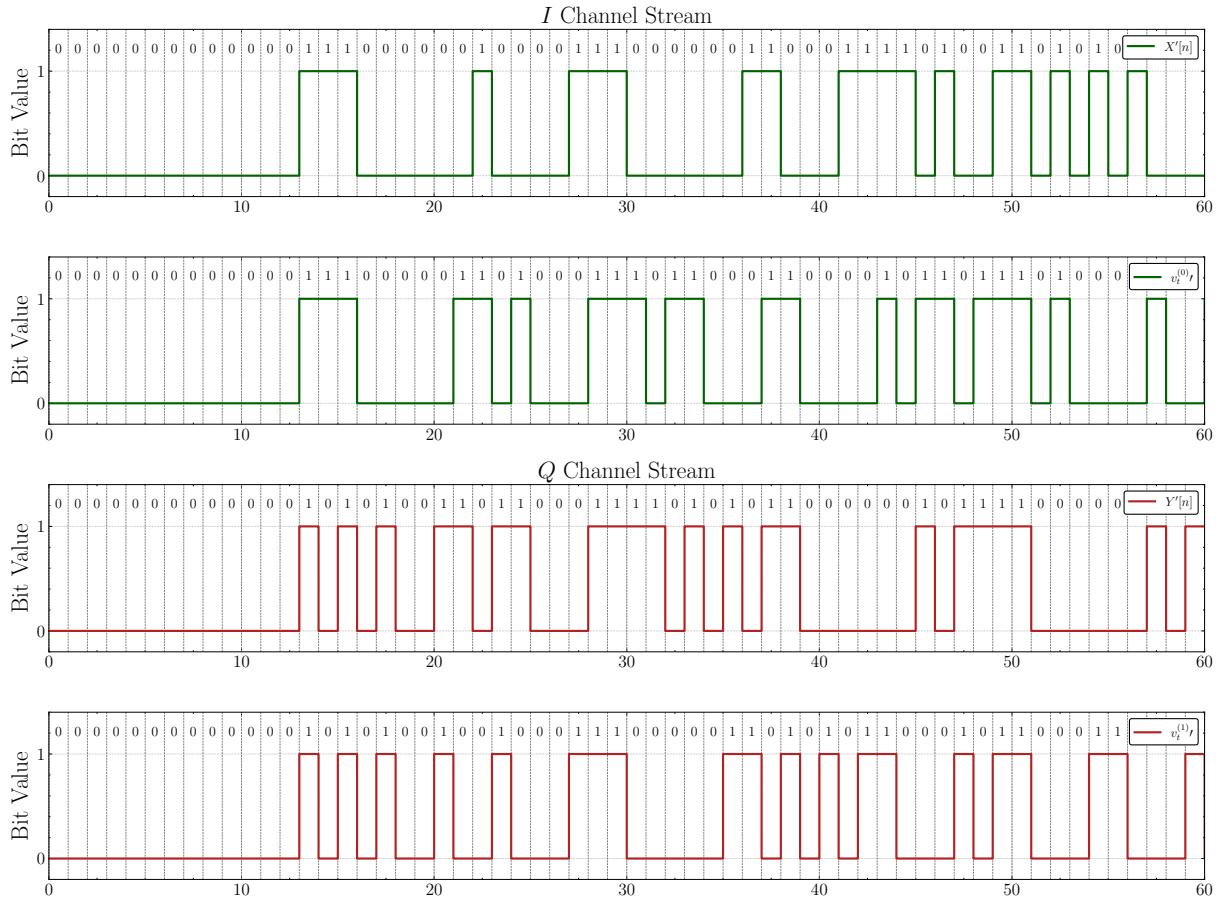


Os vetores de amostras $I'[n]$ e $Q'[n]$ estão agora prontos para a recuperação do datagrama, conforme detalhado na seção 3.4.5.

3.4.5 Recuperação do Datagrama

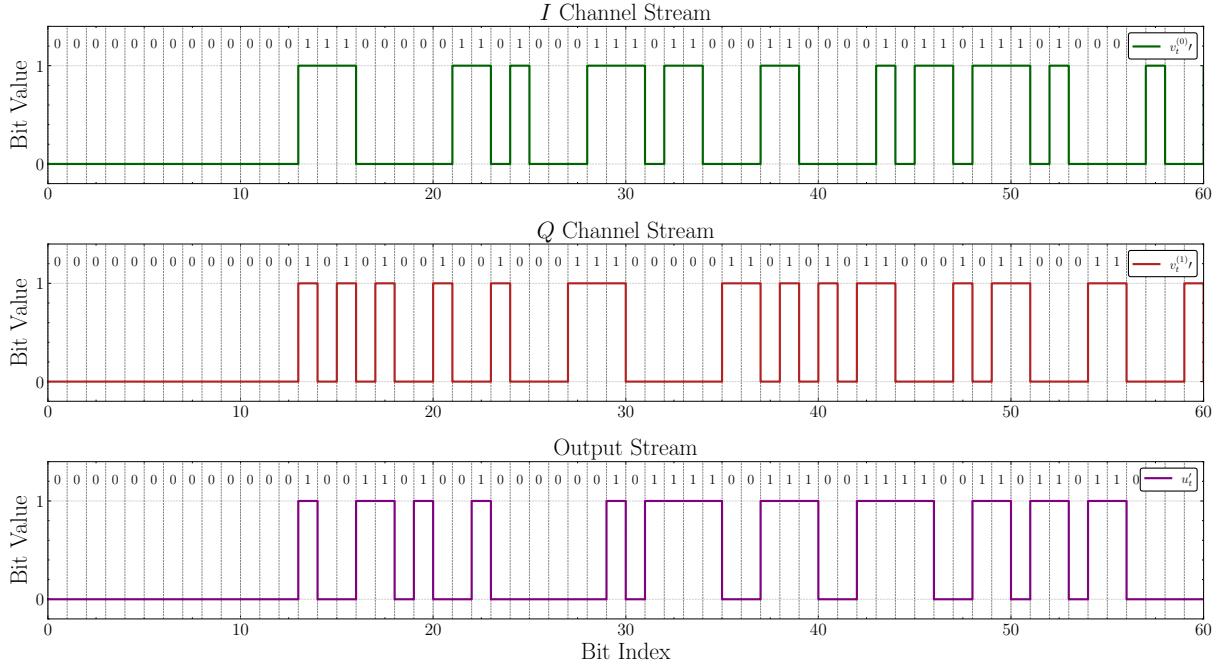
O último passo na cadeia de recepção é a recuperação do datagrama transmitido a partir dos vetores de amostras $I'[n]$ e $Q'[n]$. O primeiro passo é o descodificador de linha, onde os vetores de amostras são convertidos de volta para bits, utilizando a técnica inversa da codificação de linha aplicada na transmissão, conforme detalhado na seção 2.5.3, resultando nos vetores de bits $X'[n]$ e $Y'[n]$, que precisam ser desembaralhados, conforme ilustrado na Figura 53.

Figura 53 – Desembaralhamento dos canais I e Q



Os vetores de bits $X'[n]$ e $Y'[n]$ desembaralhados, são então decodificados utilizando o decodificador Viterbi, conforme detalhado na seção 2.5.1, resultando no vetor de bits u'_t , que representa o datagrama recuperado. O processo de decodificação convolucional é ilustrado na Figura 54.

Figura 54 – Decodificação convolucional dos canais I e Q



Finalmente, o vetor de bits u'_t é processado para extrair o datagrama, verificando a integridade dos dados através do campo de verificação *Cyclic Redundancy Check* (CRC), e então extraíndo os campos de interesse, como o número de identificação da PCD PCD_{ID} e o payload. O processo de recuperação do datagrama é apresentado no código de exemplo 3.2.

Código 3.2 – Parse do datagrama de $s'(t)$

```

1 import argos3
2
3 datagramTX = argos3.Datagram(pcdnum=1234, numblocks=1)
4 t, s = argos3.Transmitter().transmit(datagramTX)
5
6 datagramRX, success = argos3.Receiver().receive(s)
7 print(datagramRX.parse_datagram())
8 {
9     "msglength": 1,
10    "pcdid": 1234,
11    "data": {
12        "bloco_1": {
13            "sensor_1": 37,
14            "sensor_2": 198,
15            "sensor_3": 9
16        }
17    },
18    "tail": 7
19 }
```

4 CONCLUSÃO

O desenvolvimento e a simulação do sistema de comunicação compatível com o padrão PTT-A3 do ARGOS-3 permitiram compreender de forma detalhada o funcionamento de cada etapa da cadeia de transmissão e recepção. A modelagem proposta para o simulador abordou desde a geração do streambits do datagrama e a codificação convolucional até a modulação QPSK, modelagem de canal, detecção de portadora e toda a cadeia de recepção utilizada no lado do satélite para recepção dos sinais transmitidos pelas PCD. Essa abordagem possibilitou avaliar o desempenho do sistema em condições controladas de ruído e interferência.

Os resultados demonstraram que o uso combinado de modulação QPSK, codificação convolucional e embaralhamento de dados proporciona ganhos significativos em robustez e confiabilidade, em relação a modulação QPSK pura. A presença dessas técnicas reduz a *BER* em regimes de baixa SNR mostrando a importância da codificação de canal. Além disso, a análise do sinal no tempo e frequência confirmou que o formato do sinal transmitido está de acordo com as especificações do padrão PTT-A3, assegurando a compatibilidade do modelo implementado com sistemas reais de coleta de dados ambientais (CNES, 2006)

Durante o processo de simulação, verificou-se que a implementação proposta oferece grande flexibilidade para experimentação e aprimoramento. O ambiente desenvolvido em Python permite a rápida reconfiguração de parâmetros, como taxa de bits, filtros de pulso e limiares de detecção, facilitando o estudo de diferentes cenários operacionais. Essa modularidade torna o simulador uma ferramenta útil tanto para fins de pesquisa quanto para atividades de ensino relacionadas à comunicação digital e à engenharia de sistemas espaciais.

Por fim, o trabalho atingiu seu objetivo principal de reproduzir e analisar o comportamento do sistema de transmissão ARGOS-3, contribuindo para o domínio nacional das tecnologias envolvidas na recepção de dados ambientais via satélite. As propostas de trabalhos futuros, incluindo a aplicação de modelos de canal mais realistas, a implementação de decodificação Viterbi com soft-decision e o estudo de desempenho *BER* vs E_b/N_0 , abrem caminhos para melhorar o simulador como uma plataforma de estudos.

4.0.1 Trabalhos futuros

Com o desenvolvimento do projeto, alguns pontos que podem ser melhorados foram identificados:

- Aplicar técnicas de filtragem para a matriz de detecção, eliminando ruídos detecta-

dos e mantendo apenas as transmissões de interesse;

- Alterar cadeia de recepção invertendo a ordem do decodificador convolucional para trabalhar com Viterbi soft-decision;
- Aplicar modelagem de canal mais realista, adicionando desvio de frequência e efeito Doppler, implementar sincronização de fase por PLL;
- Aplicar modelagem de canal mais realista, adicionando atenuação por distância e ACG (Automatic Gain Control);
- Aplicar modelagem de canal não-linear (transmissão por amplificadores reais), simulando efeitos de distorção;
- Realizar comparativo de desempenho da curva $BER \times E_b/N_0$ variando parâmetros do sistema, como palavra de sincronismo, vetores geradores, tipo de codificação de linha, tipo de modulação de pulso, etc;
- Implementar suporte a realtime nas classes do simulador, para possibilitar a integração com SDR;
- Implementar o sistema em GNU Radio, utilizando SDR para validação do modelo proposto;
- Implementar blocos do codificador convolucional e modulador QPSK em VHDL;
- Analisar de eficiência energética e consumo de potência do sistema, otimizar consumo para aplicações em PCD;
- Simulação de tráfego de múltiplos dispositivos (colisões de uplink), avaliando impacto na performance do sistema, e calcular a densidade máxima de dispositivos suportada em uma determinada região para a passada do satélite;
- Analisar modulações alternativas, como $8 - PSK$ e $16 - QAM$, avaliando custo benefício da eficiência espectral, robustez e complexidade de implementação em relação ao QPSK;

REFERÊNCIAS

- CADORE, Arthur. **argos3**. GitHub, 2025. Accessed: 2025-10-05. Disponível em: <https://github.com/arthurcadore/argos3/>.
- CENTENARO, Marco *et al.* A Survey on Technologies, Standards and Open Challenges in Satellite IoT. **IEEE Communications Surveys & Tutorials**, v. 23, n. 3, p. 1693–1720, 2021. DOI: 10.1109/COMST.2021.3078433.
- CNES, Project Team. Services and Message Formats Ed2 Rev2. en, p. 80, 2006.
- DUARTE, José Marcelo L. *et al.* A multiuser decoder based on spectrum analysis for the Brazilian Environmental Data Collecting System. en. **International Journal of Satellite Communications and Networking**, v. 39, n. 2, p. 205–220, 2021. Publisher: John Wiley & Sons, Ltd. ISSN 1542-0981. DOI: 10.1002/sat.1381. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/sat.1381>.
- FRAIRE, Juan A.; CÉSPEDES, Sandra; ACCETTURA, Nicola. Direct-To-Satellite. en. In: AD-HOC, Mobile, and Wireless Networks. Springer, Cham, 2019. p. 241–258. ISSN: 1611-3349. ISBN 978-3-030-31831-4. DOI: 10.1007/978-3-030-31831-4_17. Disponível em: https://link.springer.com/chapter/10.1007/978-3-030-31831-4_17.
- GOMES, Guilherme Peinador. Otimização de órbita para comunicação de nanossatélites da Constelação Catarina. pt, 2024. Accepted: 2024-07-11. Publisher: Joinville, SC. Disponível em: <https://repositorio.ufsc.br/handle/123456789/255917>.
- LIMA, Raffael Sadite Cordoville Gomes de. **A parallel software-defined ultra-low-power receiver for a satellite message forwarding system**. 2021. Diss. (Mestrado) – Universidade Federal do Rio Grande do Norte. Accepted: 2022-02-02. Disponível em: <https://repositorio.ufrn.br/handle/123456789/45768>.
- RAE, João Carlos Pécal. **Detector de sinais para os satélites do Sistema Brasileiro de Coleta de Dados usando análise espectral digital**. 2005. Dissertação de Mestrado – Nome da Instituição, Brasil. Disponível conforme normas APA, Chicago e MLA.
- RODRIGUES, José Antonio. Demodulador ARGOS III Compatível com Sinais PTT-A2 e PTT-A3. pt, p. 104, 2018. Disponível em: <http://mtc-m21c.sid.inpe.br/col/sid.inpe.br/mtc-m21c/2018/08.01.12.38/doc/publicacao.pdf>.
- SHU, L. **Error Control Coding**. Pearson India, 2011. ISBN 9788131734407. Disponível em: https://books.google.com.br/books?id=BG08Zylo_7gC.

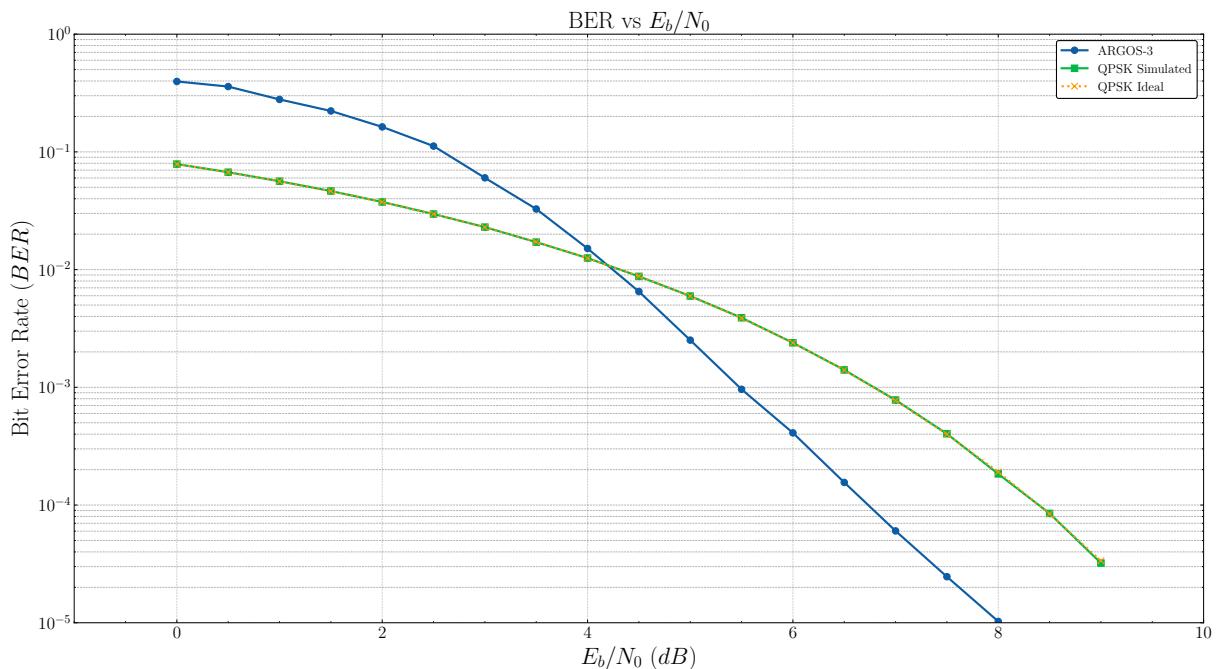
SILVA, Ramon Brandi da; ALMEIDA, Eugenio Sper de. Um framework para a identificação e diferenciação de plataformas de coleta de dados ambientais do Sistema Nacional de Dados Ambientais (SINDA). pt. In: SIMPÓSIO Brasileiro de Sistemas de Informação (SBSI). SBC, 2022. p. 17–20. ISSN: 0000-0000. DOI: 10.5753/sbsi_estendido.2022.222715. Disponível em: https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/21563.

SKLAR, Bernard. **Digital communications: fundamentals and applications**. USA: Prentice-Hall, Inc., 1988. ISBN 0132119390.

APÊNDICE A – BER VS SNR

Com base na implementação do receptor e transmissor apresentada nas seções 2.6 e 2.5, foi realizado um estudo de desempenho do sistema, variando a relação E_b/N_0 e medindo a *BER* para cada valor, comparando o desempenho do sistema ARGOS-3 com modulação QPSK (tanto teórica quanto simulada), conforme apresentado na Figura 55.

Figura 55 – Comparação BER vs SNR - ARGOS3 e QPSK



Nota-se que o desempenho do sistema ARGOS-3 está melhor do que o desempenho teórico e simulado da modulação QPSK pura, o que pode ser explicado pela presença do código convolucional, que adiciona redundância aos dados transmitidos, permitindo a correção de erros no receptor, além da presença da técnica de codificação de linha Manchester, que também contribui para a melhoria das características do sinal modulado.

APÊNDICE B – SIMULADOR OPEN SOURCE ARGOS-3

A implementação do transmissor e receptor ARGOS-3, conforme descrito nas seções 2.5 e 2.6, foi realizada em Python, utilizando bibliotecas open source como NumPy, SciPy e Matplotlib. O código-fonte completo do simulador está disponível no repositório GitHub (Cadore, 2025), permitindo que outros alunos, professores e pesquisadores possam utilizar, modificar e contribuir para o desenvolvimento do projeto. A biblioteca ARGOS-3 utilizada na implementação pode ser instalada via ‘pip’, conforme mostrado no exemplo de instalação na Código B.1.

Código B.1 – Exemplo de instalação da biblioteca

```

1 python -m venv .venv
2 source .venv/bin/activate
3 pip install argos3

```

O exemplo apresentado no Código B.2 demonstra o uso dos principais módulos da biblioteca, incluindo a criação de um transmissor, a transmissão de um datagrama, a detecção do canal e a recepção do datagrama.

Código B.2 – Exemplo de uso dos principais módulos da biblioteca

```

1 import argos3
2 import numpy as np
3
4 # Frequência da portadora (entre 1kHz e 8kHz)
5 fc = np.random.randint(10,80)*100
6
7 # Exibe a frequência da portadora
8 print(fc)
9 2400
10
11 # Cria o transmissor e transmite o datagrama gerando o sinal s(t)
12 transmitter = argos3.Transmitter(fc=fc)
13 t, s = transmitter.transmit(argos3.Datagram(pcdnum=1234, numblocks=1))
14
15 # Cria o canal com duração de 1 segundo
16 channel = argos3.Channel(duration=1, noise_mode="ebn0", noise_db=20)
17 # Adiciona o sinal no meio do canal
18 channel.add_signal(s, position_factor=0.5)
19 channel.add_noise()
20 st = channel.channel
21
22 # Cria o detector de portadora e detecta as transmissões contidas no canal
23 detector = argos3.CarrierDetector(seg_ms=10, threshold=-15)

```

```
24 detector.detect(st.copy())
25 detections = detector.return_channels()
26
27 # Exibe as detecções no formato (fc, start_seg, end_seg)
28 print(detections)
29 [(np.float64(2400.0), 41, 65)]
30
31 # Isola o sinal recebido s'(t) a partir das amostras de início e fim da detecção
32 first_sample = int((detections[0][1] - detector.history) * detector.fs * detector.
33     seg_s)
33 last_sample = int(detections[0][2] * detector.fs * detector.seg_s)
34 st_prime = st[first_sample:last_sample]
35
36 # Cria o receptor, recebe o sinal s'(t) e tenta decodificar o datagrama
37 receiver = argos3.Receiver(fc=detections[0][0])
38 datagramRX, success = receiver.receive(st_prime)
39
40 # Exibe se a recepção foi bem sucedida e o datagrama recebido
41 print(success)
42 True
43 print(datagramRX.parse_datagram())
44 {
45     "msglength": 1,
46     "pcdid": 1234,
47     "data": {
48         "bloco_1": {
49             "sensor_1": 37,
50             "sensor_2": 198,
51             "sensor_3": 9
52         }
53     },
54     "tail": 7
55 }
```