

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JULIANA SIMÕES**

**IMPLEMENTAÇÃO DE EXPERIMENTOS DE SIMULAÇÃO COM RASPBERRY  
PI 3 MODEL B PARA A DISCIPLINA DE FUNDAMENTOS DE  
COMUNICAÇÕES**

**CAMPO MOURÃO**

**2025**

**JULIANA SIMÕES**

**IMPLEMENTAÇÃO DE EXPERIMENTOS DE SIMULAÇÃO COM RASPBERRY  
PI 3 MODEL B PARA A DISCIPLINA DE FUNDAMENTOS DE  
COMUNICAÇÕES**

**Implementation of simulation experiments with Raspberry Pi 3 Model B for  
the Fundamentals of Communications class**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Eletrônica  
do Curso de Bacharelado em Engenharia  
Eletrônica da Universidade Tecnológica Federal  
do Paraná.

Orientador: Prof. Dr. Roberto Wilhelm Krauss  
Martinez

**CAMPO MOURÃO**

**2025**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JULIANA SIMÕES**

**IMPLEMENTAÇÃO DE EXPERIMENTOS DE SIMULAÇÃO COM RASPBERRY  
PI 3 MODEL B PARA A DISCIPLINA DE FUNDAMENTOS DE  
COMUNICAÇÕES**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Eletrônica  
do Curso de Bacharelado em Engenharia  
Eletrônica da Universidade Tecnológica Federal  
do Paraná.

Data de aprovação: 10 de junho de 2025

---

Roberto Wilhelm Krauss Martinez  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Osmar Tormena Junior  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Lucas Ricken Garcia  
Doutorado  
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO  
2025**

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus, que sustenta minha existência e colocou em minha vida todas as pessoas que me ajudaram ao longo desta trajetória.

Aos meus pais, por terem me dado a vida e me proporcionado condições favoráveis para bem vivê-la. Agradeço, especialmente, pela paciência nos momentos mais difíceis, pelo apoio e compreensão inabaláveis.

Ao meu orientador, Dr. Roberto Krauss, por todo o suporte oferecido. Agradeço por sua dedicação, por se colocar sempre à disposição para sanar minhas dúvidas e por me incentivar nos momentos em que pensei em desistir.

Aos amigos que me acompanharam durante toda a graduação, Carlos Boduar de Alcântara, Felipe Vieceli Bernardes de Souza e Giovana Cavalli Ronqui. Mesmo com as mudanças de rota, com um oceano ou um mosteiro carmelita nos separando, sempre pude contar com cada um de vocês e a isso serei eternamente grata.

Por fim, agradeço ao meu noivo, Rafael Zolett Correa, que esteve ao meu lado na reta final desta jornada, me apoiou em todos os momentos e adicionou uma causa maior de motivação à conclusão deste trabalho.

Queres de verdade ser santo? Cumpre o  
pequeno dever de cada momento; faz o que  
deves e está no que fazes.  
— São Josemaria Escrivá

## RESUMO

Os métodos de aprendizagem ativa, nos quais o aluno assume um papel central no próprio processo de aprendizagem, têm-se difundido progressivamente no contexto educacional. Em disciplinas de engenharia, são constantemente desenvolvidos projetos onde o aluno pode aplicar os conhecimentos de forma prática, criando uma experiência de aprendizagem profunda. Entretanto, em matérias como Fundamentos de Comunicações, o nível de complexidade e o custo dos equipamentos podem dificultar o desenvolvimento de projetos que abordem todos os tópicos estudados. O presente trabalho consiste na utilização do software MATLAB® em conjunto com a placa Raspberry Pi 3 para a elaboração de experimentos que utilizam os conhecimentos aprendidos na disciplina de Fundamentos de Comunicações. Os experimentos foram realizados a partir de diagramas de blocos desenvolvidos através da ferramenta Simulink® e foram abordados temas como modulação analógica e digital e filtragem de sinais. A comunicação entre a Raspberry Pi 3 e o mundo externo é feita através de uma placa de som USB, que atua como um conversor AD e DA, e permite tanto a inserção de sinais externos quanto a visualização em osciloscópio dos sinais processados, de modo a levar o aluno a uma experiência além da simulação tradicional no MATLAB®/Simulink®. Os resultados obtidos indicam que a proposta é funcional e adequada para fins didáticos, uma vez que permite a implementação de experimentos de baixo custo e relativa simplicidade. Apesar das limitações encontradas, como a presença de ruídos, a abordagem adotada amplia as possibilidades pedagógicas no ensino de comunicações.

**Palavras-chave:** raspberry pi; comunicações; rádio definido por software.

## ABSTRACT

Active learning methods, in which the student becomes the protagonist in the process of learning, have become increasingly common in the educational context. In engineering classes, projects are constantly developed in which the student can apply knowledge in a practical way, creating a deep learning experience. However, in classes such as Fundamentals of Communications, complexity and equipment pricing can make it difficult to develop projects that cover all the studied topics. The present work consists of using the MATLAB® software together with the Raspberry Pi 3 board to develop a series of experiments that use the knowledge learned in the Fundamentals of Communications. The experiments were carried out using block diagrams developed using the Simulink® tool and topics such as analog and digital modulation and signal filtering were covered. Communication between the Raspberry Pi 3 and the external world is done through a USB sound card, which acts as an AD and DA converter, and allows both the insertion of external signals and the visualization of the processed signals on an oscilloscope, in order to take the student beyond the simulation environment. The results obtained demonstrate a functional proposal, suitable for teaching purposes, since it allows the implementation of simple and low-cost experiments. Despite the limitations faced, the chosen approach expands the pedagogical possibilities of communications teaching.

**Keywords:** raspberry pi; communications; software defined radio.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Sinal NRZ no tempo e sua respectiva densidade espectral de potência . . . . .                 | 19 |
| Figura 2 – Modulação 2-PAM e 4-PAM . . . . .   | 19 |
| Figura 3 – Sinal AM DSB-SC no domínio do tempo e da frequência . . . . .                                 | 21 |
| Figura 4 – Modulação FM no domínio do tempo . . . . .  | 23 |
| Figura 5 – Modulação FM no domínio da frequência . . . . .   | 24 |
| Figura 6 – Forma de onda para modulação BPSK . . . . .   | 25 |
| Figura 7 – Mapeamento Gray que representa a modulação QPSK . . . . .                                     | 26 |
| Figura 8 – Raspberry Pi 3 Model B . . . . .  | 27 |
| Figura 9 – Placa de som USB externa utilizada para entrada e saída de sinais no<br>experimento . . . . . | 28 |
| Figura 10 – Cabo P2-RCA com adaptador BNC . . . . .  | 29 |
| Figura 11 – Diagrama de conexões . . . . .   | 32 |
| Figura 12 – Modelo do Simulink® para geração de um sinal senoidal . . . . .                              | 33 |
| Figura 13 – Modelo do Simulink® para captura e reprodução de sinais . . . . .                            | 33 |
| Figura 14 – Modelo do Simulink® para geração de um sinal senoidal . . . . .                              | 34 |
| Figura 15 – Modelo do Simulink® para geração de um sinal de onda quadrada . . . . .                      | 34 |
| Figura 16 – Modelo do Simulink® do transmissor empregando modulação AM-DSB-<br>SC . . . . .              | 35 |
| Figura 17 – Modelo do Simulink® para demodulação AM . . . . .  | 35 |
| Figura 18 – Diagrama de conexões para o processo de demodulação . . . . .                                | 36 |
| Figura 19 – Modelo do Simulink® para modulação FM . . . . .  | 36 |
| Figura 20 – Modelo do Simulink® para modulação 2-PAM . . . . .   | 37 |
| Figura 21 – Diagrama de constelação da modulação 2-PAM . . . . .   | 38 |
| Figura 22 – Modelo do Simulink® para modulação 4-PAM . . . . .   | 38 |
| Figura 23 – Diagrama de constelação da modulação 4-PAM . . . . .   | 39 |
| Figura 24 – Modelo do Simulink® para modulação BPSK . . . . .  | 39 |
| Figura 25 – Diagrama de constelação da modulação BPSK . . . . .  | 40 |
| Figura 26 – Modelo do Simulink® para modulação QPSK . . . . .  | 41 |
| Figura 27 – Diagrama de constelação da modulação QPSK . . . . .  | 42 |
| Figura 28 – Diagrama de blocos para modulação QPSK com componentes separadas                             | 42 |

|  |           |
|--|-----------|
| <b>Figura 29 – Resposta em frequência do sistema em que a Raspberry Pi 3 atua como gerador de funções . . . . .</b>          | <b>43</b> |
| <b>Figura 30 – Resposta em frequência do sistema em que é utilizado um gerador de funções externo . . . . .</b>              | <b>44</b> |
| <b>Figura 31 – Forma de onda senoidal obtida . . . . .</b>   | <b>45</b> |
| <b>Figura 32 – Forma de onda quadrada obtida . . . . .</b>   | <b>46</b> |
| <b>Figura 33 – Sinal modulado AM DSB-SC no domínio do tempo . . . . .</b>  | <b>46</b> |
| <b>Figura 34 – Espectro do sinal modulado AM DSB-SC com ênfase nas componentes principais . . . . .</b>                      | <b>47</b> |
| <b>Figura 35 – Espectro do sinal modulado AM DSB-SC com ênfase nas componentes espectrais de menor intensidade . . . . .</b> | <b>48</b> |
| <b>Figura 36 – Sinal demodulado AM DSB-SC . . . . .</b>  | <b>48</b> |
| <b>Figura 37 – Sinal modulado FM no tempo . . . . .</b>  | <b>49</b> |
| <b>Figura 38 – Espectro de frequência do sinal modulado FM . . . . .</b>   | <b>49</b> |
| <b>Figura 39 – Sinal 2-PAM . . . . .</b>   | <b>50</b> |
| <b>Figura 40 – Diagrama de olho do sinal 2-PAM . . . . .</b>   | <b>50</b> |
| <b>Figura 41 – Sinal 4-PAM com indicação dos quatro níveis de tensão . . . . .</b>   | <b>51</b> |
| <b>Figura 42 – Diagrama de olho do sinal 4-PAM . . . . .</b>   | <b>51</b> |
| <b>Figura 43 – Sinal BPSK . . . . .</b>  | <b>52</b> |
| <b>Figura 44 – Diagrama de olho do sinal BPSK . . . . .</b>  | <b>52</b> |
| <b>Figura 45 – Forma de onda QPSK com componentes combinadas . . . . .</b>   | <b>53</b> |
| <b>Figura 46 – Forma de onda QPSK com componentes separadas . . . . .</b>  | <b>53</b> |
| <b>Figura 47 – Forma de onda QPSK com componentes somadas através do <i>math menu</i> . . . . .</b>                          | <b>54</b> |
| <b>Figura 48 – Configurações de rede e compartilhamento do Windows 10 . . . . .</b>  | <b>59</b> |
| <b>Figura 49 – Rede Ethernet estabelecida entre o computador e a Raspberry Pi 3 . . . . .</b>                                | <b>59</b> |
| <b>Figura 50 – Redes disponíveis no DHCP Server . . . . .</b>  | <b>60</b> |
| <b>Figura 51 – Protocolos suportados pelo DHCP Server . . . . .</b>  | <b>61</b> |
| <b>Figura 52 – Configuração DHCP . . . . .</b>   | <b>61</b> |
| <b>Figura 53 – Gravação do arquivo INI . . . . .</b>   | <b>62</b> |
| <b>Figura 54 – Conclusão da configuração DHCP . . . . .</b>  | <b>63</b> |
| <b>Figura 55 – Servidor DHCP . . . . .</b>   | <b>63</b> |
| <b>Figura 56 – Conexão entre a Raspberry Pi 3 e o MATLAB® . . . . .</b>  | <b>64</b> |

|  |    |
|--|----|
| Figura 57 – Instalação dos pacotes e bibliotecas necessários . . . . .           | 65 |
| Figura 58 – Informações exibidas na janela de comando do MATLAB® . . . . .       | 65 |
| Figura 59 – <i>Shell</i> do Linux . . . . .                                      | 66 |
| Figura 60 – Listagem dos dispositivos de reprodução e captura de áudio . . . . . | 67 |
| Figura 61 – <i>AlsaMixer v.1.2.4</i> . . . . .                                   | 68 |
| Figura 62 – Configurações de implementação do modelo . . . . .                   | 69 |
| Figura 63 – Definição dos parâmetros na aba <i>Build options</i> . . . . .       | 69 |
| Figura 64 – Botão utilizado para executar o modelo na Raspberry Pi . . . . .     | 70 |
| Figura 65 – Modelo utilizado para controle automático de ganho . . . . .         | 75 |
| Figura 66 – Parâmetros do bloco <i>Digital Filter Design</i> . . . . .           | 76 |
| Figura 67 – Parâmetros do bloco <i>FM Modulator Baseband</i> . . . . .           | 77 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 – Codificação de símbolos para modulação 2-PAM . . . . .   | 18 |
| Tabela 2 – Especificações técnicas da placa de som USB externa . . . . .  | 29 |
| Tabela 3 – Parâmetros de configuração do bloco <i>Sine Wave</i> . . . . .   | 72 |
| Tabela 4 – Parâmetros de configuração do bloco <i>Data Type Conversion</i> . . . . .                                    | 73 |
| Tabela 5 – Parâmetros de configuração do bloco <i>Matrix Concatenate</i> . . . . .                                      | 73 |
| Tabela 6 – Parâmetros de configuração do bloco <i>ALSA Audio Playback</i> . . . . .                                     | 74 |
| Tabela 7 – Parâmetros de configuração do bloco <i>Pulse Generator</i> . . . . .   | 74 |
| Tabela 8 – Parâmetros de configuração do bloco <i>Buffer</i> . . . . .  | 74 |
| Tabela 9 – Parâmetros de configuração do bloco <i>Multiport Selector</i> . . . . .                                      | 75 |
| Tabela 10 – Parâmetros de configuração do bloco <i>Sine Wave</i> para modulação FM .                                    | 77 |
| Tabela 11 – Parâmetros de configuração do bloco <i>Bernoulli Binary Generator</i> . . .                                 | 78 |
| Tabela 12 – Parâmetros de configuração do bloco <i>M-PAM Modulator Baseband</i> para<br>a modulação 2-PAM . . . . .     | 78 |
| Tabela 13 – Parâmetros de configuração do bloco <i>FIR Interpolator</i> . . . . .                                       | 79 |
| Tabela 14 – Parâmetros de configuração do bloco <i>M-PSK Modulator Baseband</i> para<br>modulação BPSK e QPSK . . . . . | 79 |
| Tabela 15 – Parâmetros de configuração do bloco <i>Sine Wave</i> para portadoras em<br>quadratura . . . . .             | 80 |

## SUMÁRIO

|            |  |           |
|------------|--|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                          | <b>13</b> |
| <b>1.1</b> | <b>Objetivos</b>                           | <b>15</b> |
| 1.1.1      | Objetivo geral                             | 15        |
| 1.1.2      | Objetivos específicos                      | 15        |
| <b>1.2</b> | <b>Justificativa</b>                       | <b>15</b> |
| <b>2</b>   | <b>REFERENCIAL TEÓRICO</b>                 | <b>17</b> |
| <b>2.1</b> | <b>Resposta em frequência</b>              | <b>17</b> |
| <b>2.2</b> | <b>Modulação em banda base</b>             | <b>18</b> |
| 2.2.1      | Pulse Amplitude Modulation (PAM)           | 18        |
| <b>2.3</b> | <b>Modulação analógica</b>                 | <b>20</b> |
| 2.3.1      | Modulação em amplitude                     | 20        |
| 2.3.2      | Modulação em frequência                    | 22        |
| <b>2.4</b> | <b>Modulação Digital em Banda Passante</b> | <b>24</b> |
| 2.4.1      | Phase Shift Keying (PSK)                   | 25        |
| <b>3</b>   | <b>MATERIAIS E MÉTODOS</b>                 | <b>27</b> |
| <b>3.1</b> | <b>Materiais</b>                           | <b>27</b> |
| 3.1.1      | Raspberry Pi                               | 27        |
| 3.1.2      | Placa de som USB externa                   | 28        |
| 3.1.3      | Cabos e adaptadores                        | 29        |
| 3.1.4      | Gerador de funções e osciloscópio          | 30        |
| 3.1.5      | MATLAB® e Simulink                         | 30        |
| 3.1.6      | ALSA                                       | 31        |
| <b>3.2</b> | <b>Métodos</b>                             | <b>31</b> |
| 3.2.1      | Resposta em frequência do sistema          | 32        |
| 3.2.2      | Gerador de funções                         | 33        |
| 3.2.3      | Transmissor e receptor AM                  | 34        |
| 3.2.4      | Modulador FM                               | 36        |
| 3.2.5      | Modulador 2-PAM e 4-PAM                    | 37        |
| 3.2.6      | Modulador BPSK e QPSK                      | 39        |
| <b>4</b>   | <b>RESULTADOS</b>                          | <b>43</b> |



## 1 INTRODUÇÃO

O estudo de Deslauriers *et al.* (2019) sugere que os métodos de ensino baseados em aulas expositivas passivas, onde o professor transmite conteúdo enquanto os alunos permanecem ouvindo, não promovem um aprendizado tão eficaz quanto o obtido em metodologias ativas. No estudo, alunos submetidos a aulas ativas, nas quais interagem diretamente com o conteúdo por meio de atividades e resolução de problemas, obtiveram resultados significativamente superiores em testes de conteúdo em comparação aos que participaram de aulas tradicionais. Na chamada aprendizagem ativa, o aluno se ocupa em fazer alguma coisa e, simultaneamente, pensar de forma lógica sobre a atividade desenvolvida, enquanto o professor atua como um orientador e facilitador do processo de aprendizagem (Barbosa; Moura, 2014).

Em cursos de engenharia, é frequente a utilização de ferramentas de aprendizagem baseada em problemas e aprendizagem baseada em projetos. Ambas as ferramentas contribuem para uma aprendizagem significativa, onde os conceitos aprendidos encontram-se contextualizados e alinhados à tecnologias contemporâneas, o que transforma o conhecimento em algo palpável e contribui para a formação profissional dos estudantes, que lidam durante a graduação com situações semelhantes às que encontrarão no mercado de trabalho (Barbosa; Moura, 2014).

Apesar de se demonstrarem excelentes ferramentas, a aprendizagem baseada em problemas e projetos possui aplicação limitada em determinados contextos, como é o caso dos Rádios Definidos por Software (SDRs, do inglês *Software Defined Radio*), cujo uso tem-se tornado dominante em sistemas de comunicações (Ramos *et al.*, 2024).

Um rádio é um dispositivo sem fio que recebe e transmite sinais de rádio; dispositivos de rádio físicos apresentam limitações e só podem ser modificados através de alterações no hardware. Por sua vez, em um SDR, componentes que seriam tipicamente implementados por hardware (como filtros, moduladores, demoduladores e amplificadores) são substituídos por algoritmos digitais, permitindo uma flexibilidade maior para configurar, modificar e atualizar o rádio (Ball; Naik; Jenkins, 2017).

Existe uma série de modelos comerciais de SDRs, dentre eles pode-se citar o HackRF One, um rádio de médio alcance que opera entre 1 MHz e 6 GHz e é frequentemente utilizado para aprendizagem e pesquisa (Gadgets, 2025). Uma outra opção popular é o RTL-SDR (do inglês, *Realtek Software Defined Radio*), da NOELEC, que opera em frequências entre 24 MHz e 1,7 GHz, o que permite captar diversos tipos de transmissão, como rádio FM, sinais de comunicação aeronáutica e rádio amador (Laufer, 2015).

Os experimentos realizados em sala de aula normalmente utilizam os dispositivos comerciais supracitados, uma vez que o ensino da implementação dessa tecnologia enfrenta uma série de desafios, como o custo de hardware, que inclui kits de processamento digital de sinais e FPGAs (do inglês, *field programmable gate array*), e a complexidade envolvida no desen-

volvimento, que passa por etapas de *design*, processamento digital de sinais, programação e implementação final do hardware (Pasolini; Bazzi; Zabini, 2017).

O cenário de dificuldades apresentado pode ser superado pela introdução no mercado de dispositivos programáveis e de propósito geral, como é o caso da placa Raspberry Pi, que apresenta características que vão de encontro às dificuldades apresentadas. A placa Raspberry Pi 3 apresenta um custo abaixo do de outros equipamentos, custando R\$ 400,00 em 2024, em média. O modelo também apresenta uma diminuição na complexidade dos projetos de SDR, uma vez que é capaz de operar com software gerado de forma automática a partir de diagrama de blocos (Pasolini; Bazzi; Zabini, 2017).

A empresa MathWorks®, desenvolvedora dos softwares MATLAB® e Simulink®, fornece um pacote de suporte ao hardware do Raspberry Pi que pode ser utilizado na conexão entre uma placa Raspberry Pi e um computador pessoal. Diversos modelos da placa Raspberry Pi são aceitos, a depender das versões dos softwares MATLAB® e Simulink® utilizados (MathWorks, 2024c).

O Simulink® permite a modelagem gráfica de sistemas através de diagramas de blocos e a tradução para software, que pode então ser carregado na placa Raspberry Pi. Esse procedimento permite o projeto de algoritmos complexos, ao passo que exime a necessidade de conhecimento de linguagens de programação específicas, o que torna o processo acessível para alunos com níveis de conhecimento distintos (Pasolini; Bazzi; Zabini, 2017).

Desta forma, o trabalho de Pasolini, Bazzi e Zabini (2017) propõe o desenvolvimento de uma plataforma didática que utiliza a placa Raspberry Pi 2 Model B em conjunto com o software MATLAB® em sua versão R2015a. Os autores desenvolveram experimentos na área de telecomunicações e processamento de sinais, incluindo experimentos de geração de sinais, filtragem digital, modulação PAM, ASK e transmissão OFDM.

Neste sentido, Tominaga (2018) propôs a reprodução dos experimentos apresentados por Pasolini, Bazzi e Zabini (2017), bem como o desenvolvimento de novos, com a utilização de uma versão mais recente de hardware, a Raspberry Pi 3. Em seu trabalho, o autor implementou também experimentos de modulação analógica.

Apesar da inovação proposta por Pasolini, Bazzi e Zabini (2017) e das atualizações propostas por Tominaga (2018), com o passar dos anos o software MATLAB® e a ferramenta Simulink® passaram por atualizações que modificaram o caminho para o desenvolvimento dos experimentos. Passos importantes para a utilização da Raspberry Pi com o pacote de suporte ao hardware do MATLAB® sofreram alterações consideráveis, sendo a principal delas a descontinuidade de diversos componentes e bibliotecas.

O presente trabalho propõe a reprodução dos experimentos apresentados por Pasolini, Bazzi e Zabini (2017) e por Tominaga (2018), adequando-os para as versões atuais dos softwares MATLAB® e Simulink® e do sistema operacional mais recente da placa Raspberry Pi 3. Além disso, são implementados novos experimentos não abordados nos demais trabalhos, como a demodulação AM e a análise da resposta em frequência do dispositivo.

Os novos experimentos implementados ampliaram as possibilidades de aplicação pedagógica na área de telecomunicações, utilizando uma plataforma acessível ao cenário universitário. Como sugestão para trabalhos futuros, propõe-se a implementação da transmissão sem fio das modulações desenvolvidas, o que permitirá um estudo mais completo sobre os desafios e potencialidades dos sistemas de comunicação que utilizam rádios definidos por software.

## 1.1 Objetivos

Nesta seção, são apresentados os objetivos gerais e específicos do presente trabalho.

### 1.1.1 Objetivo geral

Prover experimentos práticos aos alunos da disciplina de Fundamentos de Comunicações utilizando a placa Raspberry Pi e os softwares MATLAB® e Simulink®.

### 1.1.2 Objetivos específicos

- Reproduzir e adaptar os experimentos propostos por Pasolini, Bazzi e Mirabella (2016), como os de modulação PAM e PSK, adequando-os para as versões atuais dos softwares e sistemas operacionais;
- Elaborar experiências relacionadas à disciplina de Fundamentos de Comunicações, abordando tópicos como a filtragem de sinais, modulação analógica e digital e transmissão e recepção de sinais, de modo a permitir aos alunos a observação e análise de sinais com instrumentos reais;
- Apresentar os procedimentos necessários para a configuração das ferramentas utilizadas, com foco em instruções claras e acessíveis, que permitam fácil reprodução pelos usuários.

## 1.2 Justificativa

O presente trabalho surge da premissa de que a aprendizagem eficaz requer não somente a exposição teórica aos conceitos, mas também a aplicação prática e a análise de resultados (Barbosa; Moura, 2014). Como mostra a pesquisa realizada por Deslauriers *et al.* (2019), alunos submetidos à metodologias ativas de ensino enfrentam desafios e têm como resultado maiores pontuações em testes de conhecimento.

Como já exposto, a utilização de SDRs no ensino de telecomunicações frequentemente envolve a utilização de dispositivos comerciais e reduz o impacto do "aprender fazendo", ao

passo que os alunos não têm acesso de forma aprofundada ao desenvolvimento de seus diversos componentes, especialmente os de modulação e demodulação. Como citado por Pasolini, Bazzi e Zabini (2017), a elaboração de SDRs raramente é realizada em sala devido ao custo elevado de hardware e à complexidade envolvida nas diversas etapas de projeto, que demandam uma quantidade de tempo muitas vezes não disponível no período letivo.

A placa de desenvolvimento Raspberry Pi 3 surge como uma solução viável, uma vez que é de fácil acesso nos laboratórios da universidade, além de possuir custo relativamente baixo quando em comparação com outros dispositivos, como FPGAs. Além disso, a ferramenta Simulink® oferece uma interface intuitiva e dispensa a necessidade de conhecimentos de linguagens de programação específicas, o que diminui o nível de dificuldade e o tempo demandado para a execução dos experimentos. Assim, a realização dos experimentos propostos proporciona aos alunos a oportunidade de explorar conceitos importantes vistos na disciplina de forma prática e controlada.

Apesar da relevância dos trabalhos desenvolvidos por Pasolini, Bazzi e Zabini (2017) e por Tominaga (2018) e da inovação trazida em termos da utilização da placa Raspberry Pi em conjunto com o software MATLAB® e Simulink® novas versões de software foram lançadas após a publicação dos trabalhos, de modo que, para um aluno com pouca experiência, a reprodução dos experimentos pode se tornar difícil e demorada.

Do exposto, pretende-se trazer as atualizações necessárias a fim de tornar a reprodução simples e inclusiva, assim como elaborar novos experimentos que trabalhem tópicos não abordados anteriormente.

## 2 REFERENCIAL TEÓRICO

O presente capítulo discorre sobre os conceitos teóricos necessários para a compreensão do desenvolvimento dos experimentos propostos.

### 2.1 Resposta em frequência

A resposta em frequência é uma característica de sistemas lineares invariantes no tempo (LIT) que descreve como o sistema responde a diferentes frequências de entrada. Com aplicações em diversas áreas da engenharia, como controle e telecomunicações, a resposta em frequência permite avaliar a capacidade de um sistema amplificar, atenuar ou filtrar sinais em diferentes faixas de frequência (Oppenheim; Willsky, 2010).

Matematicamente, a resposta em frequência é obtida através da transformada de Fourier da resposta ao impulso apresentada pelo sistema LIT e é dada pela Equação 1.

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)}, \quad (1)$$

em que  $j$  é a unidade imaginária,  $\omega$  é a frequência angular,  $Y(j\omega)$  é a transformada da saída e  $X(j\omega)$  é a transformada da entrada de um sistema LIT. Dessa forma, a resposta em frequência  $H(j\omega)$  possui magnitude e fase, onde a magnitude determina o ganho ou atenuação aplicada a cada frequência, enquanto a fase indica o atraso ou avanço temporal aplicado ao sinal de entrada (Oppenheim; Willsky, 2010).

A partir da Equação 1, verifica-se que o efeito do sistema sobre a entrada é de multiplicá-la pela resposta em frequência, motivo pelo qual a resposta em frequência é também chamada de ganho do sistema (Oppenheim; Willsky, 2010).

A representação gráfica da resposta em frequência é comumente feita em diagramas de magnitude em função da frequência, onde utiliza-se a escala logarítmica para a magnitude da transformada de Fourier. Para sistemas de tempo contínuo, é frequente a utilização de escala de frequência logarítmica. O diagrama mais utilizado para essa representação é o chamado diagrama de Bode, que facilita a visualização das regiões em que o sistema amplifica (ganho maior que 1), atenua (ganho menor que 1) ou apresenta características específicas, como a ressonância (Oppenheim; Willsky, 2010).

Nos sistemas de telecomunicações, a resposta em frequência desempenha um papel importante, pois garante que os sinais transmitidos sejam amplificados ou filtrados de forma adequada. Um canal de comunicação, por exemplo, deve possuir uma resposta em frequência adequada para preservar o sinal transmitido e minimizar distorções, o que evita a perda de informações.

## 2.2 Modulação em banda base

A modulação em banda base é feita atuando-se diretamente sobre o sinal, sem o uso de uma portadora de alta frequência, de modo que o sinal resultante permanece centrado em torno de 0 Hz.

### 2.2.1 Pulse Amplitude Modulation (PAM)

A modulação por amplitude de pulso (PAM, do inglês *pulse amplitude modulation*) é uma técnica de modulação digital onde a amplitude de pulsos regularmente espaçados é variada de forma proporcional às informações contidas na mensagem (Haykin; Moher, 2008).

Matematicamente, dado um sinal digital de entrada  $a_i$ , representado por uma sequência de símbolos associados a bits binários, a sequência de pulsos que representa o sinal PAM pode ser expressa pela Equação (2) (Tominaga, 2018).

$$s(t) = \sum_{i=-\infty}^{\infty} a_i p(t - iT_s), \quad (2)$$

em que  $T_s$  é o período da amostragem e  $p(t)$  é a forma de onda de um pulso unitário, geralmente assumido como um pulso retangular para simplificação.

Em um sistema de modulação por amplitude de pulso  $M$ -ária (M-PAM), cada símbolo  $a_i$  é escolhido entre  $M$  possíveis níveis de amplitude, ou número de símbolos (sempre em potência de 2), o que permite que cada símbolo represente um número  $b_{bit} = \log_2 M$  de bits. Por exemplo, em uma modulação 2-PAM, os bits  $[0,1]$  podem ser mapeados nos níveis de amplitude (símbolos)  $a_i \in [-1, +1]$ , como expresso na Tabela 1. Já em uma modulação 4-PAM, os pares de bits 00, 01, 10, 11 são mapeados para os níveis  $a_i \in [-1, -1/2, +1/2, +1]$  (Tominaga, 2018).

**Tabela 1 – Codificação de símbolos para modulação 2-PAM**

| Bit de entrada | Símbolo de saída |
|----------------|------------------|
| 0              | $-S$             |
| 1              | $+S$             |

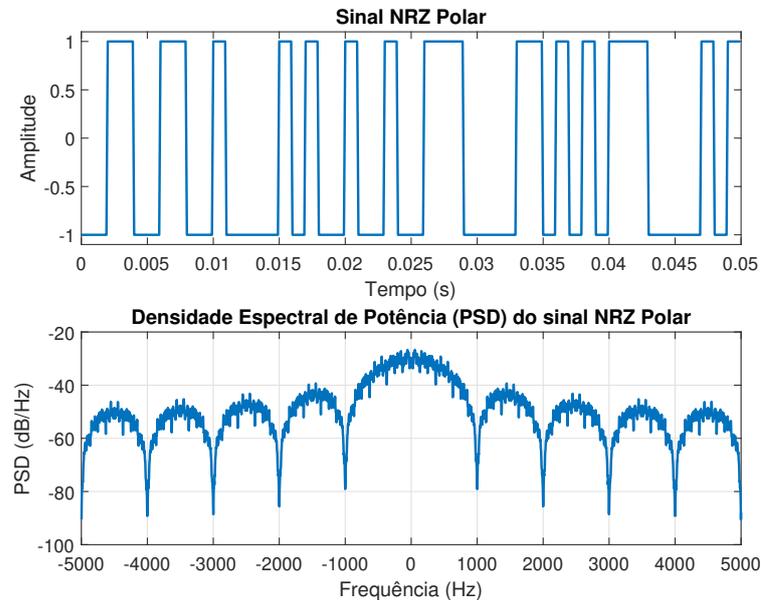
**Fonte: Pasolini, Bazzi e Mirabella (2016).**

O caso da modulação PAM com dois níveis de amplitude, ou seja,  $M = 2$ , representa um caso particular conhecido como codificação NRZ (NRZ, do inglês *non-return to zero*), onde os bits são modulados em formas de onda que não retornam ao nível de tensão 0 V antes do final do tempo de bit, ou codificação RZ (RZ, do inglês *return to zero*), em que a forma de onda retorna para o nível de tensão 0 V antes do final do tempo de bit (Sklar, 2001).

A Figura 1 mostra um trecho de um sinal NRZ polar gerado a partir de uma sequência binária aleatória, onde o sinal se altera entre os níveis  $[+1, -1]$ , que representam respectiva-

mente os bits 1 e 0. Na parte inferior da imagem, verifica-se a PSD do sinal, estimada através do método de Welch. Nota-se um pico central em 0 Hz, bem como a ocorrência de zeros periódicos a cada  $n \cdot f_s$ , em que  $n = \{0,1,2,\dots\}$  e  $f_s$  é a taxa de transmissão do sinal.

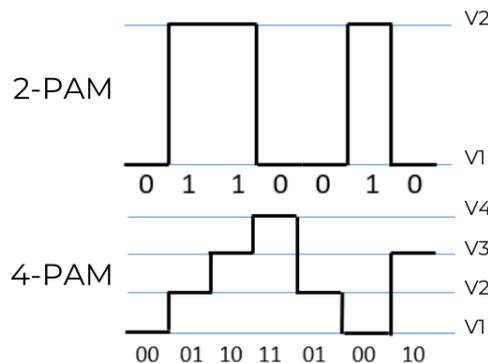
**Figura 1 – Sinal NRZ no tempo e sua respectiva densidade espectral de potência**



**Fonte: Autoria própria (2025).**

A Figura 2 mostra uma comparação entre a modulação 2-PAM, com dois níveis de tensão, e 4-PAM, com 4 níveis de tensão.

**Figura 2 – Modulação 2-PAM e 4-PAM**



**Fonte: Adaptado de Kim *et al.* (2021, p. 3).**

A escolha do número  $M$  de níveis afeta diretamente a eficiência espectral e a resistência ao ruído. Um maior número de níveis oferece maior densidade de informações por símbolo, porém se tornam mais suscetíveis a erros devido a ruídos e interferências, uma vez que os níveis de amplitude ficam mais próximos um do outro, o que reduz a margem de erro na detecção de símbolos (Sklar, 2001).

## 2.3 Modulação analógica

O processo de modulação de um sinal consiste em alterar uma ou mais propriedades de uma onda portadora de acordo com a informação do sinal de mensagem a ser transmitida. Nesse processo, o sinal de mensagem é deslocado para uma faixa específica de frequência, que facilita sua transmissão. O processo de modulação pode ser classificado como analógico ou digital, a depender do tipo de sinal transmitido. No contexto da modulação analógica, utilizada em sistemas tradicionais, como rádio, as técnicas de modulação em amplitude (AM, do inglês *amplitude modulation*) e em frequência (FM, do inglês *frequency modulation*) desempenham um importante papel na transmissão de informações (Lathi; Ding, 2012).

Enquanto a modulação AM varia a amplitude da onda portadora de acordo com o sinal da mensagem, a modulação FM varia a frequência da onda portadora. A escolha da técnica de modulação depende de uma série de fatores, como a largura de banda disponível, a resistência a ruídos e interferências e a aplicação desejada.

### 2.3.1 Modulação em amplitude

A modulação AM é uma das técnicas mais conhecidas e utilizadas de modulação analógica, onde um sinal de mensagem,  $m(t)$ , é deslocado para uma nova faixa de frequências centrada em  $f_c$  Hz. Esse deslocamento é obtido através da propriedade da translação na frequência da transformada de Fourier, que afirma que a multiplicação de um sinal por uma onda senoidal no domínio do tempo resulta em um deslocamento do espectro desse sinal no domínio da frequência. Assim, na modulação em amplitude, um sinal de mensagem  $m(t)$  é multiplicado por uma portadora de frequência  $c(t)$  (Lathi; Ding, 2012). Assim, o sinal modulado é representado no tempo através da Equação (3).

$$s_1(t) = m(t)c(t)$$

$$s_1(t) = A_m A_c \cos(2\pi f_m t) \cos(2\pi f_c t) \quad (3)$$

em que  $A_m$  é a amplitude da mensagem,  $A_c$  é a amplitude da portadora,  $f_m$  é a frequência da mensagem e  $f_c$  é a frequência da portadora.

Pela propriedade trigonométrica da multiplicação entre dois cossenos, obtém-se a Equação 4.

$$s_1(t) = \frac{A_m A_c}{2} \cos(2\pi(f_m - f_c)t) + \cos(2\pi(f_m + f_c)t) \quad (4)$$

No domínio da frequência, essa operação resulta em um espectro com duas bandas laterais: em  $+f_c$ , encontra-se uma componente a sua direita, denominada banda lateral superior

(USB, do inglês *upper sideband*) e outra a sua esquerda, denominada banda lateral inferior (LSB, do inglês *lower sideband*), como descrito pela Equação 5. Já em  $-f_c$ , a USB encontra-se à esquerda e a LSB encontra-se à direita.

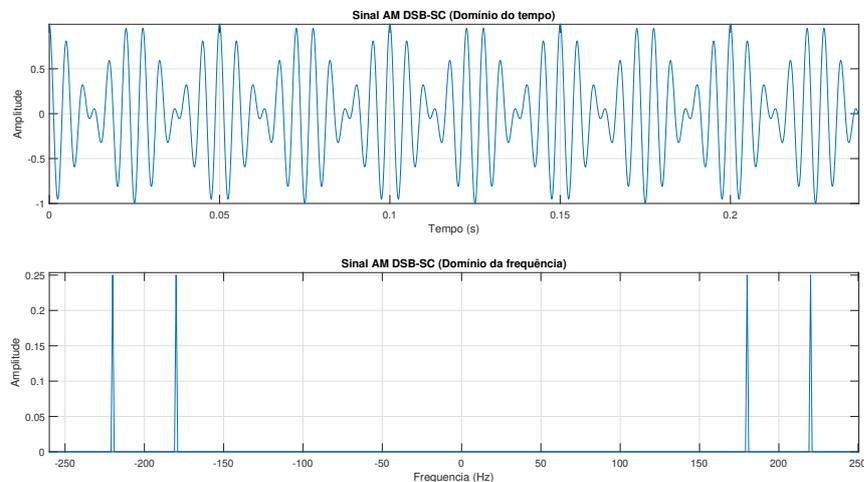
$$S_1(f) = \frac{1}{2}M(f - f_c) + \frac{1}{2}M(f + f_c) \quad (5)$$

em que  $M(f - f_c)$  é a transformada de Fourier do sinal  $m(t)$ .

Essa forma de modulação é chamada modulação com banda lateral dupla e portadora suprimida (DSB-SC, do inglês *double-sideband suppressed carrier*), uma vez que a frequência central da portadora não possui uma componente discreta no espectro (Lathi; Ding, 2012).

A Figura 3 mostra um sinal AM DSB-SC no domínio do tempo (acima) e da frequência (abaixo). Neste exemplo, a mensagem  $m(t)$  é um sinal senoidal de frequência  $f_m = 20$  Hz e amplitude unitária, enquanto a portadora é um sinal senoidal de frequência  $f_c = 200$  Hz e amplitude também unitária. No domínio do tempo, percebe-se uma variação na amplitude do sinal em alta frequência, denominado portadora, que varia de acordo com o sinal da mensagem. Já no domínio da frequência, o sinal possui duas bandas laterais deslocadas, uma à direita e outra à esquerda de  $\pm f_c$ .

**Figura 3 – Sinal AM DSB-SC no domínio do tempo e da frequência**



**Fonte: Autoria Própria (2025).**

A demodulação de sinais AM DSB-SC é feita através de um demodulador coerente, no qual o sinal modulado é multiplicado por um sinal idêntico ao da portadora utilizada para a modulação (Lathi; Ding, 2012), como demonstrado pela Equação 6.

$$m_2(t) = s(t)A_c \cos(2\pi f_c t)$$

$$m_2(t) = \frac{A_m A_c}{2} [\cos(2\pi(f_c - f_m)t) + \cos(2\pi(f_c + f_m)t)] \cdot A_c \cos(2\pi f_c t) \quad (6)$$

Pela identidade trigonométrica do produto entre dois cossenos, obtém-se a Equação 7.

$$m_2 = \frac{A_m A_c^2}{4} [\cos(-2\pi f_m t) + \cos(2\pi(2f_c - f_m)t) + \cos(2\pi f_m t) + \cos(2\pi(2f_c + f_m)t)] \quad (7)$$

No demodulador, o sinal  $m_2$  é aplicado a um filtro passa-baixas, onde são retiradas as componentes de alta frequência do sinal, o que resulta na Equação 8.

$$m_{rec}(t) = \frac{(A_m \cdot A_c^2)}{4} \quad (8)$$

em que  $m_{rec}(t)$  é o sinal demodulado.

Para a correta demodulação do sinal, é necessário que a portadora gerada no receptor esteja sincronizada em frequência e em fase com a portadora transmitida, o que dificulta o processo.

### 2.3.2 Modulação em frequência

A modulação em frequência é um tipo de modulação angular, onde a frequência instantânea do sinal portador varia de maneira proporcional ao sinal da mensagem (Lathi; Ding, 2012). Por apresentar alta imunidade a ruídos e interferências, a modulação FM é frequentemente utilizada em sistemas de comunicação analógicos, como radiodifusão e telecomunicações (Haykin; Moher, 2008).

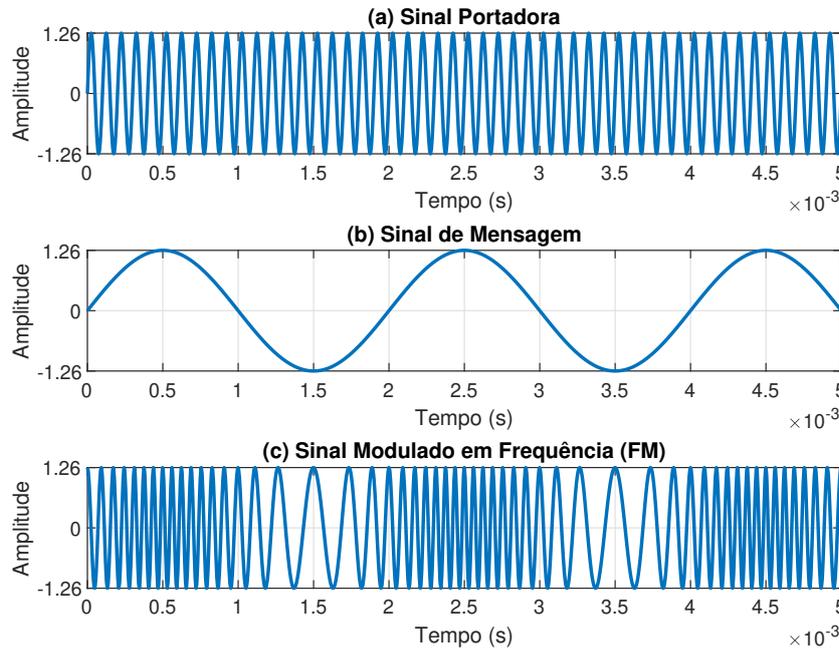
Dado o sinal de uma mensagem  $m(t)$ , o sinal modulado em frequência pode ser matematicamente representado pela Equação (9) (Haykin; Moher, 2008).

$$s(t) = A_c \cos[2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau], \quad (9)$$

em que  $A_c$  é a amplitude do sinal da portadora,  $f_c$  é a frequência da portadora e  $k_f$  é o fator de sensibilidade de frequência do modulador, que determina o desvio da frequência do sinal modulado em relação à portadora. O termo  $2\pi k_f \int_0^t m(\tau) d\tau$  representa o desvio de fase acumulado devido ao sinal da mensagem  $m(t)$ , o que resulta em uma variação proporcional da frequência instantânea do sinal transmitido.

A Figura 4 ilustra a modulação FM. O sinal portador encontra-se em 4(a), com frequência  $f_c = 10$  kHz e amplitude  $A_c = 1,26$ , a mensagem é representada em 4(b), com frequência  $f_m = 500$  Hz e amplitude  $A_m = 1,26$ , e o sinal modulado em 4(c). Percebe-se a variação da frequência do sinal modulado em decorrência da variação de amplitude do sinal da mensagem.

**Figura 4 – Modulação FM no domínio do tempo**



**Fonte: Autoria própria (2025).**

A partir dos dados utilizados para a simulação, é possível reescrever a Equação 9, obtendo-se a Equação 10.

$$s(t) = 1,26 \cos\left[2\pi 10^3 \cdot t + 2\pi k_f \int_0^t m(\tau) d\tau\right] \quad (10)$$

A Figura 5 mostra o sinal modulado no domínio da frequência. Ao se analisar o espectro de frequência do sinal FM, observa-se que a energia do sinal não se concentra apenas na frequência da portadora  $f_c$ , mas se espalha em várias bandas laterais, localizadas em múltiplos da frequência do sinal de mensagem  $f_m$ .

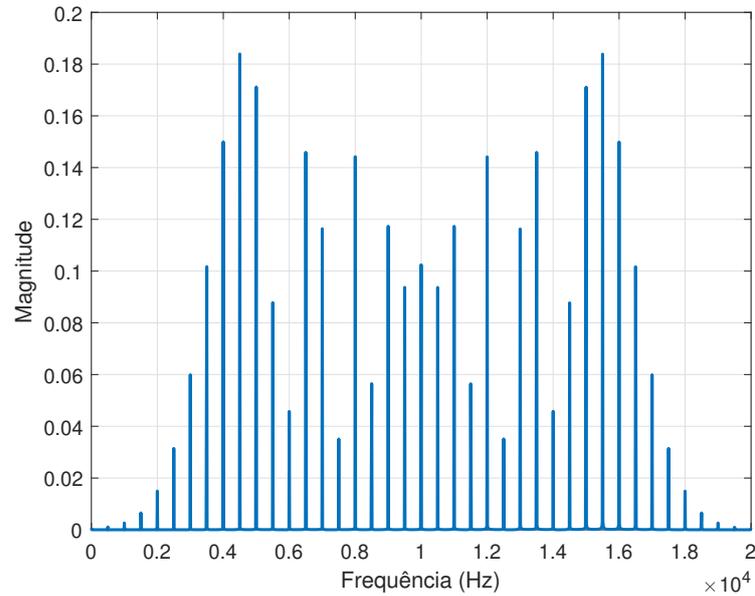
Uma característica importante na modulação FM é o desvio de frequência, que corresponde à máxima variação da frequência instantânea em relação à frequência da portadora. Representado pela Equação 11, percebe-se que o desvio de frequência independe da frequência do sinal modulante, variando apenas em função da amplitude do mesmo (Haykin; Moher, 2008).

$$\Delta_f = k_f A_m, \quad (11)$$

em que  $A_m$  é a amplitude do sinal modulante.

A partir do desvio de frequência, obtém-se o índice de modulação  $\beta$ , representado pela Equação 12. O índice de modulação trata-se de uma grandeza adimensional que indica o grau de modulação aplicado ao sinal, onde valores baixos de  $\beta$  correspondem à modulação de banda estreita, enquanto valores altos resultam em modulação de banda larga.

**Figura 5 – Modulação FM no domínio da frequência**



**Fonte: Autoria própria (2025).**

$$\beta = \frac{\Delta f}{f_m} \quad (12)$$

O presente trabalho apresentará uma modulação tonal, logo, a partir da Equação 9, pode-se reescrever sua fase instantânea através da Equação 13. Resolvendo a integral, obtém-se a Equação 13.

$$\theta_i = 2\pi f_c t + 2\pi k_f \int [A_m \cos(2\pi f_m t)] dt \quad (13)$$

em que  $\theta_i$  é o ângulo do cosseno,  $A_m$  é a amplitude do sinal da mensagem e  $f_m$  é a frequência do sinal da mensagem.

$$\theta_i = 2\pi f_c t + \frac{\Delta f}{f_m} \sin(2\pi f_m t) \quad (14)$$

Portanto, substituindo  $\frac{\Delta f}{f_m}$  por  $\beta$ , a Equação 9 que representa o sinal modulado pode ser reescrita em função de  $\beta$ , o que resulta na Equação 15.

$$s(t) = A_c \cos [2\pi f_c t + \beta \sin(2\pi f_m t)] \quad (15)$$

## 2.4 Modulação Digital em Banda Passante

Na modulação digital, o sinal da mensagem é composto por uma sequência de dados binários que representam a informação a ser transmitida. Aqui, os parâmetros da onda porta-

dora, ou seja, amplitude, frequência ou fase, são alterados em níveis discretos correspondentes aos valores da mensagem. A modulação digital é amplamente utilizada em sistemas modernos de telecomunicações, como redes de dados, televisão digital e telefonia celular.

#### 2.4.1 Phase Shift Keying (PSK)

A modulação por chaveamento de fase (PSK, do inglês *phase shift keying*) é um tipo de modulação digital onde a fase de uma portadora senoidal é alterada em função das mudanças do sinal modulante. Essa técnica de modulação é amplamente utilizada em sistemas de comunicação digital devido à sua eficiência na transmissão de dados (Sklar, 2001).

A mensagem transmitida é descrita pela Equação (16).

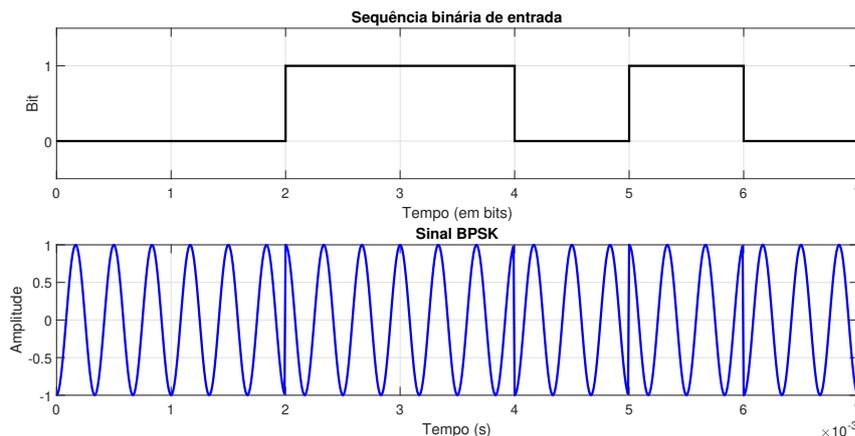
$$s_i(t) = \sqrt{\frac{2E}{T}} \cos[\omega_0 t + \phi_i(t)], \quad (16)$$

em que  $E$  é a energia de cada símbolo,  $T$  é a duração de cada símbolo,  $i = 1, \dots, M$ . A fase  $\phi$  de cada símbolo é determinada pela Equação (17).

$$\phi_i(t) = \frac{2\pi i}{M} \quad (17)$$

A Figura 6 ilustra a forma de onda gerada pela modulação BPSK (PSK binária), onde  $M = 2$ . Neste caso, a cada mudança no sinal de entrada, a fase do sinal modulado é modificada em  $\pi$  radianos, representando as transições entre os dois estados possíveis do sinal.

**Figura 6 – Forma de onda para modulação BPSK**



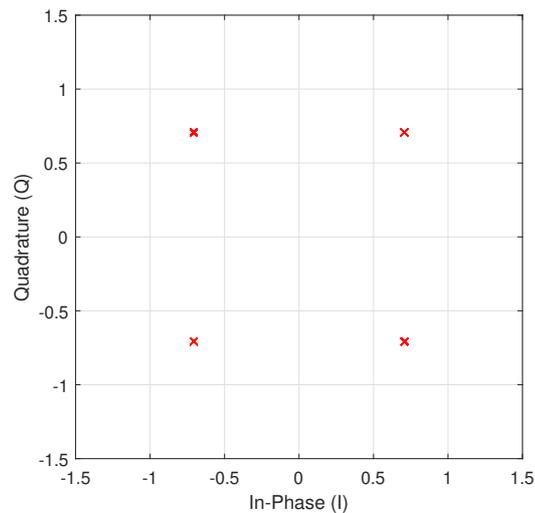
**Fonte: Autoria própria (2025).**

Na modulação PSK, os bits de entrada são agrupados para formar cada símbolo do sinal transmitido. Assim como na modulação PAM, a quantidade de bits por símbolo é dada por  $\log_2(M)$ . Por exemplo, em uma modulação 4-PSK, também chamada modulação PSK por quadratura (QPSK, do inglês *quadrature-phase-shift keying*), os bits são agrupados em pares. Nesse caso, o sinal da portadora pode ser modulado para assumir uma das quatro fases distintas:

$\pi/4$ ,  $3\pi/4$ ,  $5\pi/4$  e  $7\pi/4$  radianos. Com isso, é possível que dois bits sejam transmitidos a cada símbolo, o que aumenta a eficiência da modulação (Haykin; Moher, 2008).

A constelação apresentada pela Figura 7 representa a modulação QPSK, em que cada ponto corresponde a um símbolo representado como um número complexo com fase fixa. As posições dos pontos no plano IQ (*In-Phase* e *Quadrature*, do inglês "em fase" e "em quadratura", respectivamente) indicam as fases ideais da modulação, o que evidencia a natureza discreta e a separação angular entre os símbolos da modulação QPSK.

**Figura 7 – Mapeamento Gray que representa a modulação QPSK**



**Fonte: Autoria própria (2025).**

### 3 MATERIAIS E MÉTODOS

O presente capítulo apresenta as ferramentas utilizadas para o desenvolvimento do trabalho, bem como as etapas de desenvolvimento.

A sequência de experimentos implementados se inicia com experimentos introdutórios de geração de sinais, com o objetivo de introduzir assuntos básicos e familiarizar os alunos com as ferramentas utilizadas, seguidos de experimentos de modulação analógica e digital.

As ferramentas de hardware e software utilizadas estão descritas na subseção 3.1, enquanto a descrição do trabalho realizado encontra-se na subseção 3.2.

#### 3.1 Materiais

Nesta seção, são apresentadas as ferramentas de hardware e de software utilizadas para a implementação dos experimentos.

##### 3.1.1 Raspberry Pi

A Raspberry Pi é um *single board computer* de dimensões reduzidas desenvolvido pela Raspberry Pi Foundation. Equipada com um *System on a Chip* (SoC), ela integra em um único circuito integrado todos os componentes essenciais de um computador convencional, como uma unidade central de processamento (CPU, do inglês *central processing unit*) e uma unidade de processamento gráfico (GPU, do inglês *graphics processing unit*) (Upton *et al.*, 2016).

A versão utilizada neste trabalho é a Raspberry Pi 3 Model B, que pode ser vista na Figura 8. A versão utiliza o SoC Broadcom BCM2837 e possui funcionalidades como Wi-Fi e Bluetooth integrados, além de CPU ARMv8 quad-core de 64 bits (Upton *et al.*, 2016).

**Figura 8 – Raspberry Pi 3 Model B**



Fonte: Adaptado de Upton *et al.* (2016).

A alimentação recomendada para a Raspberry Pi 3 Model B é feita através de uma fonte micro USB de 5 V com capacidade mínima de 2,5 A (Raspberry Pi Ltd, 2024a). A recomendação de corrente mínima se deve à possibilidade de conexão de periféricos à Raspberry Pi, o que leva a um aumento no consumo de corrente na mesma; entretanto, verificou-se através de testes que, caso não haja a conexão de periféricos, uma fonte de alimentação com capacidade de 0,5 A é suficiente.

A placa possui 40 pinos de entrada e saída de propósito geral (GPIO, do inglês *general-purpose input/output*), sendo dois reservados para tensão de 5 V, dois reservados para tensão de 3,3 V e oito reservados para referência. Os demais pinos GPIO podem ser configurados como entradas ou saídas digitais (Raspberry Pi Ltd, 2024b). A Raspberry Pi não possui entradas analógicas, de modo que é necessário utilizar um conversor ADC externo para a leitura de sinais analógicos (Tominaga, 2018).

A Raspberry Pi opera utilizando o sistema operacional Raspberry Pi OS, que deve ser instalado em um cartão micro SD de no mínimo 8 GB (Raspberry Pi Ltd, 2024a).

### 3.1.2 Placa de som USB externa

A Raspberry Pi 3 possui uma saída analógica de áudio, entretanto não possui nenhum tipo de entrada analógica, de modo que foi utilizada uma placa de som USB externa para a entrada de sinais, bem como para a saída de sinais gerados pela Raspberry Pi 3. A placa de som utilizada é da marca Vinik, conforme mostra a Figura 9, e possui entrada para microfone, que desempenha um papel de conversor analógico-digital para os sinais de entrada, e saída para fones de ouvido, que desempenha um papel de conversor digital-analógico para os sinais de saída (Pasolini; Bazzi; Zabini, 2017). A Tabela 2 mostra as especificações técnicas da placa utilizada.

**Figura 9 – Placa de som USB externa utilizada para entrada e saída de sinais no experimento**



Fonte: Adaptado de Eletroinfo Cia (2024).

**Tabela 2 – Especificações técnicas da placa de som USB externa**

| <b>Característica</b> | <b>Especificação</b> |
|-----------------------|----------------------|
| Versão do USB         | 2.0                  |
| Taxa de amostragem    | 48 kHz               |
| Tipo de conexão       | P2                   |

**Fonte: Adaptado de Eletroinfo Cia (2024).**

Uma vez que o foco principal de uso da placa de som USB é para utilização com sinais de áudio, sua taxa de amostragem está limitada a 48 kHz. Assim, de acordo com a taxa de Nyquist, a largura de banda máxima dos sinais utilizados nos experimentos deve estar entre 0 e 24 kHz.

### 3.1.3 Cabos e adaptadores

#### **Cabo P2-RCA e adaptador BNC**

Para a visualização dos sinais gerados na saída da placa de som USB ou para a aplicação de sinais à entrada da mesma, são utilizados cabos P2 macho (estéreo) para RCA macho em conjunto com adaptadores RCA fêmea para BNC macho, como mostra a Figura 10. O uso do adaptador é necessário uma vez que osciloscópios e geradores de funções possuem conexão do tipo BNC.

**Figura 10 – Cabo P2-RCA com adaptador BNC**



**Fonte: Autoria própria (2025).**

#### **Cabo Ethernet**

A comunicação entre a Raspberry Pi e o computador é feita através de um cabo Ethernet. O cabo é da categoria CAT 5E, que suporta uma largura de banda de até 1 Gb/s, sendo adequado para redes locais (LAN) que demandam alta velocidade de transmissão de dados.

Caso a entrada Ethernet do computador já esteja sendo utilizada, é possível utilizar um adaptador USB-Ethernet para realizar a conexão; neste projeto, a conexão do computador à internet foi realizada via Wi-Fi, de modo que não se fez necessária a utilização do adaptador para a comunicação entre a Raspberry Pi e o computador.

#### 3.1.4 Gerador de funções e osciloscópio

Um gerador de funções é um equipamento eletrônico capaz de gerar formas de onda de diversos tipos, como ondas senoidais, quadradas e triangulares. Em alguns experimentos, o gerador de funções será utilizado para gerar sinais de entrada para a Raspberry Pi. O modelo utilizado é o MGF-4201A, da marca Minipa.

O osciloscópio será utilizado em todos os experimentos, uma vez que permite a visualização e análise de sinais em tempo real. O modelo utilizado é o TDS2002B, da marca Tektronix, que possui também a função de analisador de espectro, o que permite a análise do sinal no domínio da frequência. Este modelo apresenta uma impedância de entrada de  $1\text{ M}\Omega$  em paralelo com uma capacitância de  $20\text{ pF}$ , característica que garante medições precisas sem interferir no circuito analisado (Tektronix, 2009).

#### 3.1.5 MATLAB® e Simulink

MATLAB® é um software de programação e computação numérica utilizado em diversas áreas, como engenharia, matemática e computação (MathWorks, 2024a). Desenvolvida pela MathWorks, a plataforma possui como extensão o Simulink, um ambiente de desenvolvimento para diagrama de blocos, que permite a modelagem de sistemas, a simulação de diagramas e a implementação sem desenvolvimento de código (MathWorks, 2024b). A versão dos softwares utilizada neste trabalho é a R2023b, que é compatível com sistemas operacionais Windows, Linux e Mac.

Os softwares MATLAB® e Simulink® possuem diversos pacotes, denominados *toolboxes*, com ferramentas desenvolvidas para lidar com áreas específicas, como processamento de sinais, aprendizagem de máquina e estatística. Para a execução dos experimentos aqui propostos, é necessário que, além dos softwares MATLAB® e Simulink® sejam instalados os seguintes *toolboxes*: *Signal Processing Toolbox*, *DSP System Toolbox* e *Communications System Toolbox*.

Para a execução dos experimentos na Raspberry Pi, é necessária a instalação do *MATLAB Support Package for Raspberry Pi Hardware*, que é compatível com diversos modelos de Raspberry Pi, incluindo a Raspberry Pi 3 Model B, utilizada neste projeto, para a qual é necessário que a versão dos softwares MATLAB® e Simulink® seja no mínimo a R2016a, sendo compatível com todas as demais versões mais recentes (MathWorks, 2024c).

### 3.1.6 ALSA

A arquitetura ALSA (*Advanced Linux Sound Architecture*, do inglês Arquitetura Avançada de Som para Linux), trata-se de uma estrutura de software que possui um conjunto *drivers* de hardware, bibliotecas e utilitários que provê funcionalidades de áudio para o sistema operacional Linux (ALSA Project, 2020).

Uma vez que o sistema operacional utilizado na Raspberry Pi é baseado em Linux, a utilização de dispositivos de gravação e reprodução de áudio requer a instalação do pacote ALSA *utilities*. Ao instalar o Raspberry Pi OS, sistema operacional oficial da Raspberry Pi, o pacote é instalado de forma automática, não sendo necessária nenhuma ação do usuário. O pacote de suporte ao hardware do MATLAB® realiza a verificação dos pacotes necessários para a operação e, caso seja verificado que o ALSA *utilities* ou algum outro ainda não tenha sido instalado, sua instalação é realizada.

O pacote ALSA *utilities* permite a configuração dos dispositivos de áudio, sendo possível alterar a prioridade dos dispositivos (alternando entre a placa de som externa e a saída de áudio nativa da Raspberry Pi, por exemplo) e os níveis de volume. A versão utilizada neste projeto é a 1.2.4 e os procedimentos para configuração encontram-se descritos no apêndice B.

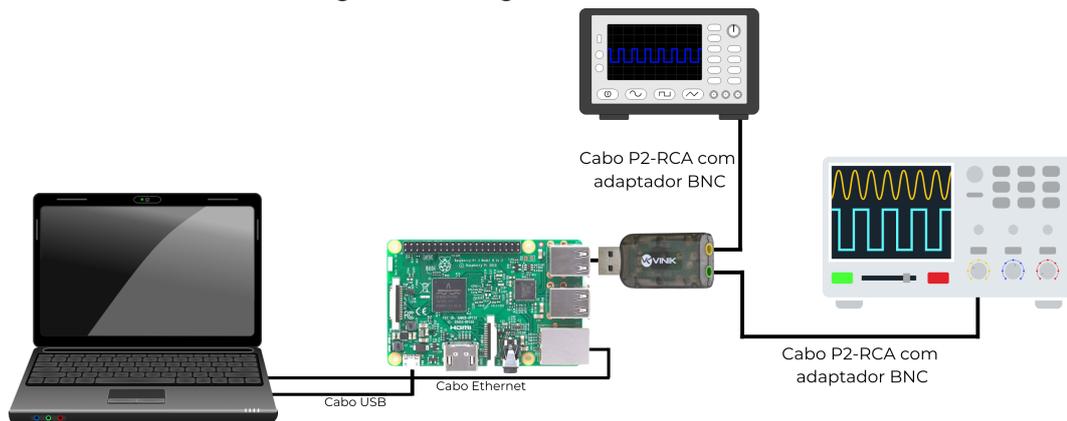
## 3.2 Métodos

Os experimentos desenvolvidos foram organizados em etapas progressivas, com início em tópicos básicos que visam familiarizar os alunos com a plataforma e uma evolução gradativa para abordar os tópicos da disciplina de Fundamentos de Comunicações. Cada experimento foi estruturado em etapas que incluíram a configuração do ambiente de simulação no Simulink® a implementação do diagrama de blocos e a execução do experimento através da Raspberry Pi.

Inicialmente, a Raspberry Pi 3 foi conectada a um computador pessoal através de um cabo Ethernet, o que possibilita o acesso remoto e o controle do dispositivo. Foi realizada a configuração inicial de uma porta IP estática, de modo que, após a primeira configuração, o mesmo IP fosse reutilizado em todas as execuções subsequentes. Todas as configurações realizadas anteriormente à simulação encontram-se no Apêndice A.

Para a visualização dos sinais resultantes dos experimentos, conforme mostra a Figura 11, uma placa USB externa foi conectada à Raspberry Pi 3, através da qual os sinais de saída foram transmitidos ao osciloscópio por meio de cabo P2 - RCA conectado com adaptador BNC.

**Figura 11 – Diagrama de conexões**



**Fonte: Autoria Própria (2024).**

Uma vez finalizadas todas as configurações, deu-se início à implementação de cada um dos experimentos, desenvolvidos de modo a explorar diferentes técnicas de modulação e de processamento de sinais utilizando a Raspberry Pi 3. Os detalhes das configurações de cada um dos blocos presentes nos modelos encontram-se no Apêndice B.

### 3.2.1 Resposta em frequência do sistema

O primeiro experimento teve como objetivo determinar a resposta em frequência do sistema utilizando sinais senoidais em diversas frequências. Essa análise é importante para decisões de projeto nos experimentos subsequentes, bem como para a análise de seus resultados.

O experimento foi conduzido por meio de duas abordagens distintas. Na primeira, gerou-se um sinal senoidal através do Simulink® com amplitude fixa definida como  $2^{15} - 1$ , e variou-se progressivamente sua frequência a fim de observar a resposta na saída do sistema. Como não foi adotada uma referência de tensão para o cálculo do ganho, a resposta em frequência foi representada em termos de tensão de saída em função da frequência.

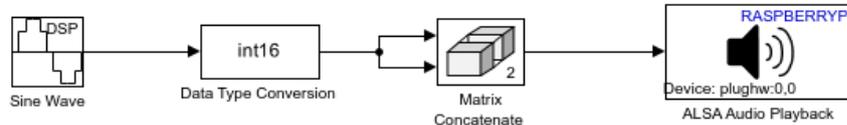
Na segunda abordagem, o sinal foi gerado por um gerador de funções, com amplitude fixa de 1 V e variação progressiva da frequência, sendo então processado pela Raspberry Pi via Simulink® e visualizado em um osciloscópio.

O modelo apresentado na Figura 12 apresenta a primeira abordagem, na qual a Raspberry Pi 3 é utilizada como um gerador de sinal senoidal. O sinal é produzido de forma discreta pelo bloco *Sine Wave*, configurado com amplitude de  $2^{15} - 1$  e frequência variando de 10 Hz a 14 kHz. Frequências superiores a 14 kHz apresentaram distorções significativas na forma de onda observada no osciloscópio, inviabilizando sua análise como sinal senoidal. Por esse motivo, tais frequências foram desconsideradas na construção da resposta em frequência.

Após a geração, o sinal é convertido para o tipo `int16` (inteiro de 16 bits), compatível com dispositivos de áudio. Em seguida, é duplicado para formar dois canais, organizados

em uma matriz estéreo, e enviado ao osciloscópio por meio da interface de áudio ALSA. As configurações detalhadas de cada bloco estão descritas no Apêndice B.

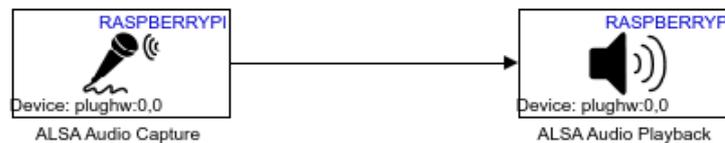
**Figura 12 – Modelo do Simulink® para geração de um sinal senoidal**



**Fonte: Autoria própria (2025).**

Na segunda abordagem, os sinais senoidais foram gerados por um gerador de funções, com amplitude fixa de 1 V e frequência variando de 10 Hz a 19 kHz. Assim como na abordagem anterior, o sinal observado no osciloscópio apresentou fortes distorções em frequências acima de 19 kHz, não sendo possível reconhecer o caráter senoidal no mesmo, o que impossibilitou a análise precisa da forma de onda, e por isso foram igualmente desconsideradas. Os sinais foram então aplicados à entrada para microfone da placa de som USB. Em seguida, os sinais foram transmitidos para um osciloscópio utilizando a saída de áudio da placa de som USB, como mostra a Figura 13.

**Figura 13 – Modelo do Simulink® para captura e reprodução de sinais**



**Fonte: Autoria própria (2025).**

Logo, os sinais foram visualizados no osciloscópio e salvos em arquivos CSV com dados de tempo e amplitude. Para o processamento dos sinais, os arquivos foram lidos através de um *script* do MATLAB® e suas respectivas amplitudes foram convertidas para unidades de ganho em decibels e, em sequência, foi gerada a curva de resposta em frequência em escala logarítmica para cada uma das abordagens. As demais configurações realizadas encontram-se descritas no Apêndice B.

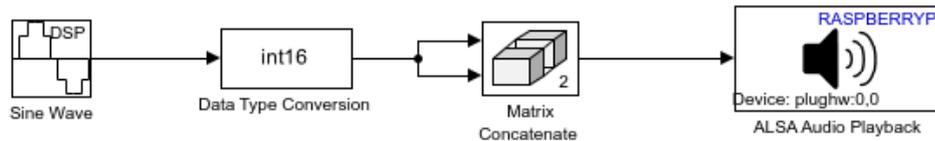
### 3.2.2 Gerador de funções

O segundo experimento desenvolvido busca familiarizar os estudantes com blocos do Simulink® que serão utilizados posteriormente em experimentos mais complexos, além de apresentar a execução em tempo real dos modelos. O experimento é uma adaptação do primeiro experimento proposto por Pasolini, Bazzi e Mirabella (2016) e consiste em utilizar a Raspberry Pi 3 como um gerador de funções.

O modelo apresentado na Figura 14, idêntico ao utilizado na primeira abordagem para obtenção da função de transferência, transforma a Raspberry Pi 3 em um gerador de sinal

senoidal. No diagrama, o sinal senoidal é gerado de forma discreta pelo bloco *Sine Wave*, configurado com os parâmetros especificados. Em seguida, o sinal é convertido para o tipo `int16`, ou seja, um número inteiro com 16 bits, compatível com dispositivos de áudio. Posteriormente, o sinal é reorganizado em formato de matriz, onde é duplicado para criar dois canais de áudio e, por fim, transmitido para um dispositivo de reprodução conectado à Raspberry Pi 3 utilizando a interface ALSA.

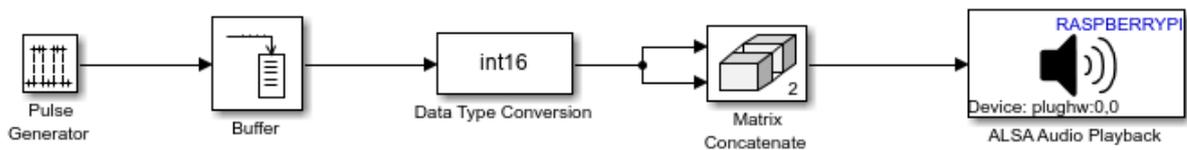
**Figura 14 – Modelo do Simulink® para geração de um sinal senoidal**



**Fonte: Autoria própria (2024).**

Alterando apenas o bloco *Sine Wave* e mantendo a configuração dos demais, é possível gerar outras formas de onda. A Figura 15 mostra o diagrama utilizado para a geração de um sinal de onda quadrada, onde nota-se a substituição do bloco *Sine Wave* pelos blocos *Pulse Generator*, que gera pulsos retangulares em intervalos regulares, e *Buffer*, que agrupa as amostras do sinal em quadros.

**Figura 15 – Modelo do Simulink® para geração de um sinal de onda quadrada**



**Fonte: Autoria própria (2024).**

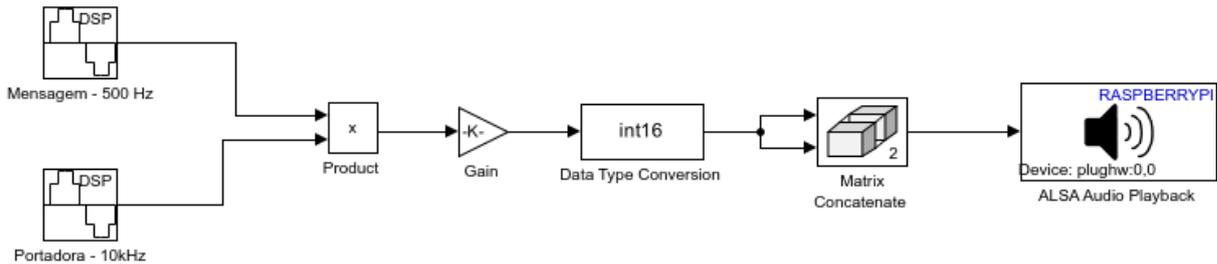
As demais configurações realizadas encontram-se descritas no Apêndice B.

### 3.2.3 Transmissor e receptor AM

O experimento de modulação e demodulação AM teve como objetivo ilustrar o processo de transmissão e recuperação de um sinal utilizando a técnica de modulação em amplitude com banda lateral dupla e portadora suprimida, uma das mais tradicionais e amplamente aplicadas nas telecomunicações. O experimento foi dividido em duas partes principais: a implementação do modulador AM e a recuperação do sinal original pelo demodulador AM.

O modelo apresentado na Figura 16 transforma a Raspberry Pi 3 em um modulador AM-DSB-SC. No diagrama, o sinal da mensagem, configurado com frequência de 500 Hz e amplitude de 1,26 V, é gerado pelo bloco *Sine Wave*; simultaneamente, o sinal da portadora, com frequência de 10 kHz e amplitude de 1,26 V é gerado por outro bloco *Sine Wave*.

**Figura 16 – Modelo do Simulink® do transmissor empregando modulação AM-DSB-SC**



Fonte: Autoria própria (2024).

A modulação é realizada de acordo com a equação da modulação AM-DSB-SC, expressa pela Equação 18.

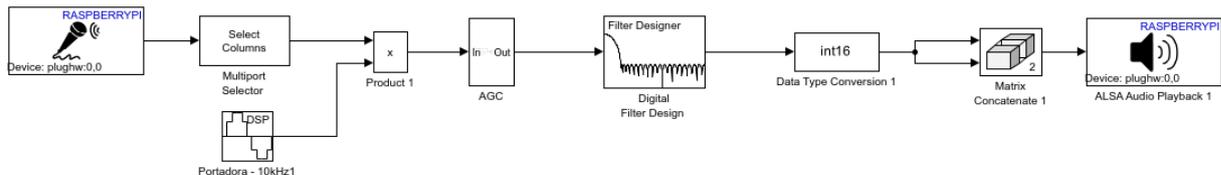
$$s(t) = 1,26 \cos(2\pi \cdot 500t) \cdot 1,26 \cos(2\pi \cdot 10000t) \quad (18)$$

Os sinais são multiplicados pelo bloco *Product* para realizar a modulação. Em seguida, o bloco *Gain* ajusta a amplitude do sinal, que é então convertido para o tipo de dado *int16* e reorganizado em formato de matriz de modo a criar dois canais de áudio. Por fim, o sinal é transmitido para o dispositivo de reprodução conectado à Raspberry Pi 3 por meio da interface ALSA.

Após o processo de modulação, realizou-se a demodulação coerente com o objetivo de recuperar o sinal original, conforme a Equação 6.

Neste experimento, todo o processo de modulação e demodulação foi implementado em uma única Raspberry Pi 3, como apresentado pelo modelo na Figura 17.

**Figura 17 – Modelo do Simulink® para demodulação AM**



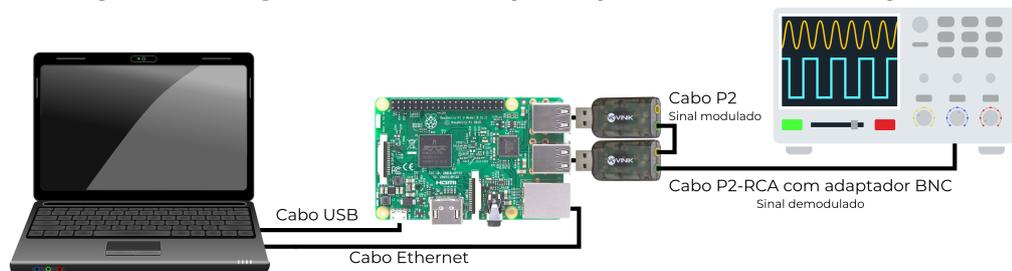
Fonte: Autoria própria (2025).

No modelo apresentado, o sinal modulado é capturado pela entrada de microfone da placa de som externa, representada pelo bloco *ALSA Audio Capture* e processado pelo bloco *Multiport Selector*, que converte o sinal de entrada estéreo em um sinal mono. Finalmente, o sinal é multiplicado por uma onda senoidal de 10 kHz, gerada pelo bloco *Sine Wave*, de mesma frequência e fase da portadora original, conforme expresso pela Equação 6.

O sinal tem então sua amplitude ajustada por um bloco de controle automático de ganho (AGC, do inglês *Automatic Gain Control*), que ajusta dinamicamente sua amplitude. Em seguida, o sinal é filtrado, eliminando-se o segundo termo da Equação 6. Por fim, o sinal é convertido para o tipo *int16* e para o formato estéreo e transmitido para o osciloscópio por meio da interface ALSA.

A Figura 18 apresenta as conexões realizadas para o experimento, onde o sinal modulado foi transmitido pela saída de áudio de uma das placas externas e conectado à entrada de microfone de uma segunda placa de som externa. Após a demodulação, realizada no Simulink® o sinal recuperado é transmitido pela saída de áudio da segunda placa de som externa e visualizado no osciloscópio.

**Figura 18 – Diagrama de conexões para o processo de demodulação**

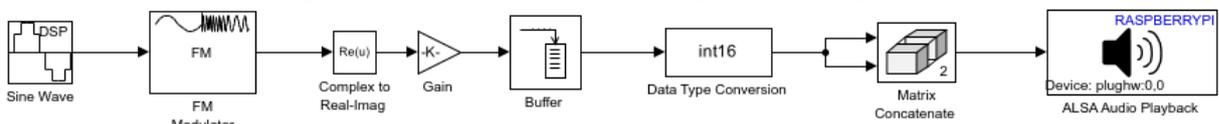


Fonte: Autoria própria (2025).

### 3.2.4 Modulador FM

O experimento de modulação FM teve como objetivo implementar um modulador FM utilizando ferramentas disponibilizadas pelo *Communications Toolbox* do Simulink®. A Figura 19 apresenta o modelo desenvolvido.

**Figura 19 – Modelo do Simulink® para modulação FM**



Fonte: Autoria própria (2024).

No modelo da Figura 19, o bloco *Sine Wave* é responsável por gerar um sinal senoidal que funciona como sinal modulante. Esse sinal é aplicado ao bloco *FM Modulator Baseband*, que realiza a modulação em frequência do sinal, produzindo um sinal modulado em banda base. A utilização da modulação em banda base neste trabalho visa otimizar o processamento do sinal no ambiente de simulação, uma vez que o sinal não será transmitido em meio físico. Em trabalhos futuros, é possível incluir um estágio final de conversão de banda base para banda passante.

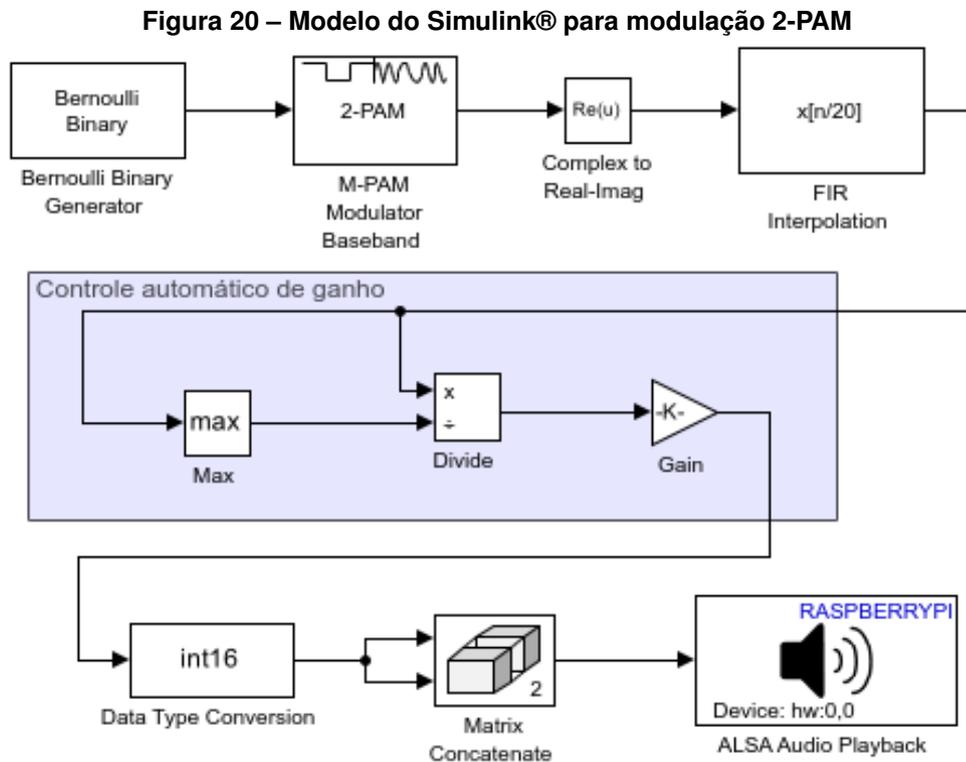
Em seguida, o bloco *Complex to Real-Imag* extrai a componente real do sinal modulado, que tem seu ganho ajustado para atender os requisitos de transmissão e, em seguida, é organizado em quadros de 1024 amostras por meio de um bloco de buffer.

Posteriormente, o sinal é convertido para o formato de dados `int16` e duplicado para formar um sinal estéreo, utilizando o bloco de concatenação de matrizes. Por fim, o sinal é transmitido através da interface ALSA, permitindo que seja visualizado através do osciloscópio.

### 3.2.5 Modulador 2-PAM e 4-PAM

No experimento de modulação PAM, dois sistemas foram implementados, sendo um de modulação 2-PAM e outro de modulação 4-PAM. A primeira configuração envolve a modulação 2-PAM, que pode ser visualizada na Figura 20, onde um sinal binário é gerado por um bloco *Bernoulli Binary Generator*, que gera uma sequência de bits aleatórios. Esses bits são então modulados em dois níveis distintos de amplitude através do bloco *M-PAM Modulator Baseband*.

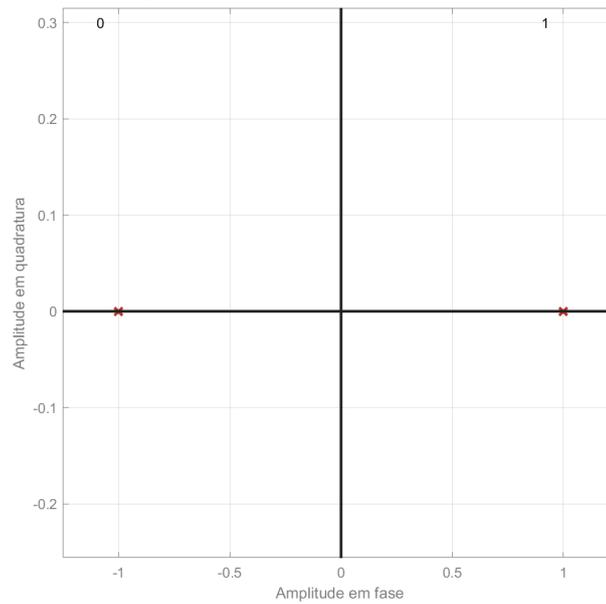
A parte real do sinal é extraída pelo bloco *Complex to Real-Imag* e em seguida interpolada através de um filtro FIR de modo a ajustar a taxa de amostragem e suavizar o espectro do sinal. Por fim, o sinal é convertido para o tipo de dado `int16`, duplicado para o formato estéreo e externalizado pela interface ALSA.



Fonte: Autoria própria (2024).

O bloco *M-PAM Modulator Baseband* gera como saída uma sequência de símbolos mapeada em um alfabeto M-ário. A Figura 21 representa a constelação Gray obtida para a modulação 2-PAM, representando os dois níveis de amplitude possíveis ( $-1$  e  $+1$ ), dispostos no eixo real.

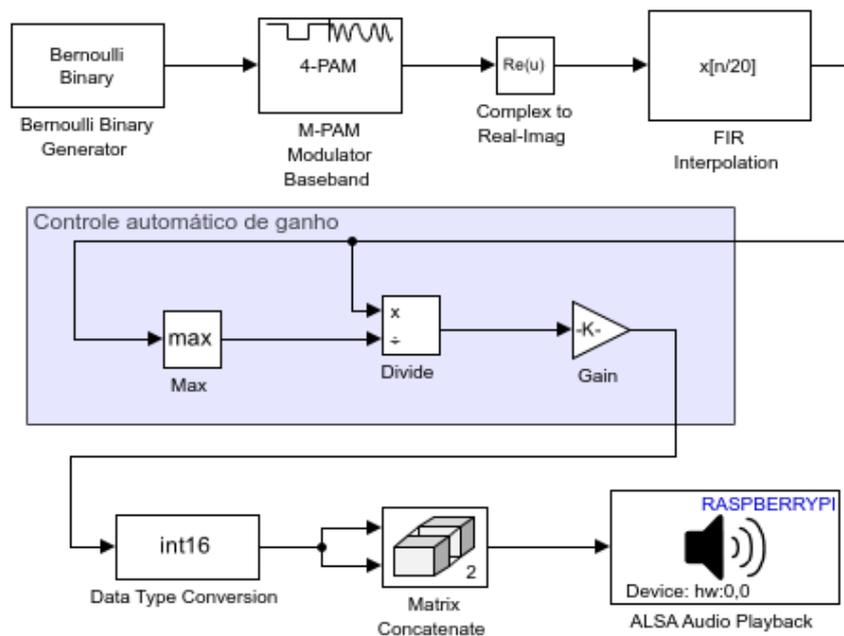
**Figura 21 – Diagrama de constelação da modulação 2-PAM**



**Fonte: Autoria própria (2025).**

Após a implementação da modulação 2-PAM, foi realizada a implementação da modulação 4-PAM. Diferentemente da modulação anterior, aqui o sinal é mapeado em quatro níveis de amplitude, o que permite a transmissão de mais informações a cada símbolo modulado. A Figura 22 mostra o modelo implementado, onde o sinal passa pelos mesmos processos de conversão de componentes real e imaginária e interpolação por filtro FIR para ajuste de taxa de amostragem e suavização do espectro. A configuração de todos os blocos é idêntica à realizada para a modulação 2-PAM, com exceção do parâmetro Número M-ário do bloco *M-PAM Modulator Baseband*, aqui definido como 4.

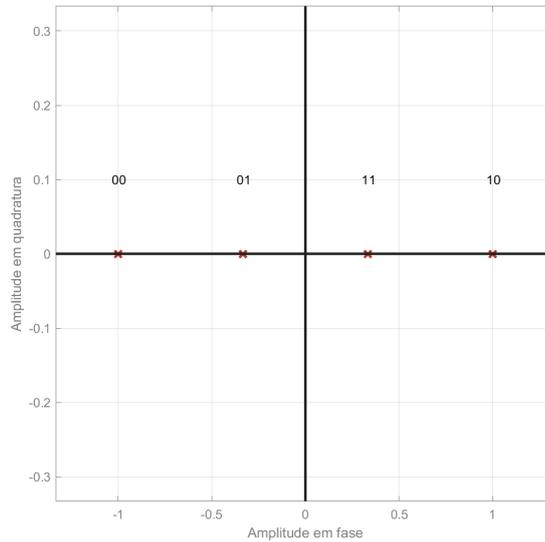
**Figura 22 – Modelo do Simulink® para modulação 4-PAM**



**Fonte: Autoria própria (2024).**

Com a alteração realizada no bloco *M-PAM Modulator Baseband*, o diagrama de constelação é modificado, como mostra a Figura 23, onde verifica-se a existência de quatro níveis de amplitude ( $-1$ ,  $-0,5$ ,  $+0,5$  e  $+1$ ), dispostos no eixo real.

**Figura 23 – Diagrama de constelação da modulação 4-PAM**



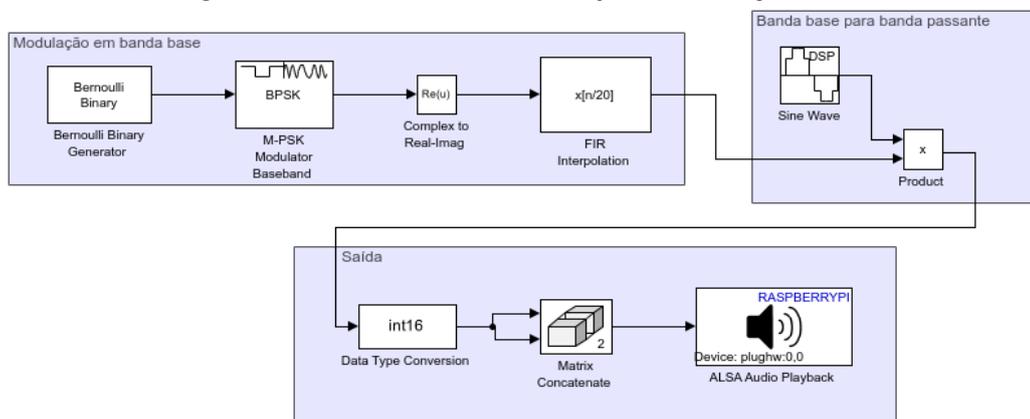
**Fonte: Autoria própria (2024).**

### 3.2.6 Modulador BPSK e QPSK

No experimento de modulação PSK, dois sistemas foram implementados, sendo um de modulação BPSK e outro de modulação QPSK. A modulação BPSK utiliza duas fases,  $0$  rad e  $\pi$  rad, para representar os bits 0 e 1, respectivamente, enquanto a modulação QPSK utiliza quatro fases:  $0$ ,  $\pi/2$ ,  $\pi$ , e  $3\pi/2$ , o que permite a codificação de dois bits por símbolo.

A primeira configuração implementada foi a do modulador BPSK, conforme ilustrado na Figura 24. No diagrama, o bloco *Bernoulli Binary Generator* gera uma sequência aleatória de bits, que são então modulados em duas fases distintas pelo bloco *M-PSK Modulator Baseband*.

**Figura 24 – Modelo do Simulink® para modulação BPSK**



**Fonte: Autoria própria (2024).**

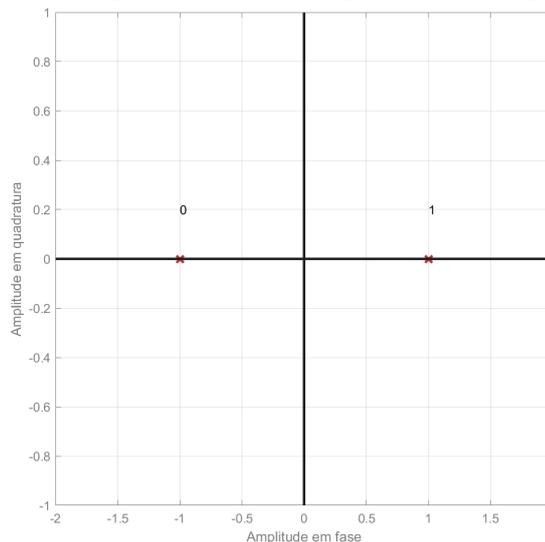
O sinal modulado é então interpolado por um filtro FIR, que ajusta a taxa de amostragem e suaviza o espectro do sinal, e tem sua parte real extraída pelo bloco *Complex to Real-Imag*. Em seguida, o sinal é modulado em banda passante pelo produto entre o sinal modulado e uma senoide gerada pelo bloco *Sine Wave*.

O bloco *M-PSK Modulator Baseband* mapeia os bits de entrada em duas fases distintas, sendo 0 rad para o bit 0 e  $\pi$  rad para o bit 1, como expresso pela Equação 19.

$$s(t) = \sqrt{2E_b} \cos(2\pi \cdot 10000t + \pi b), \quad b \in \{0,1\} \quad (19)$$

Esses dois pontos são dispostos sobre o eixo real, como mostrado no diagrama de constelação da Figura 25. Esse comportamento reflete que, no BPSK, toda a energia do sinal está concentrada na componente real, enquanto a componente imaginária é nula.

**Figura 25 – Diagrama de constelação da modulação BPSK**



**Fonte: Autoria própria (2024).**

No caso da modulação QPSK, duas abordagens distintas foram utilizadas para o processamento das componentes do sinal. Na primeira abordagem, as componentes em fase e em quadratura foram somadas diretamente no modelo do Simulink® o que possibilita a visualização do sinal modulado QPSK em sua forma final. Já na segunda abordagem, as componentes foram mantidas separadas e examinadas individualmente.

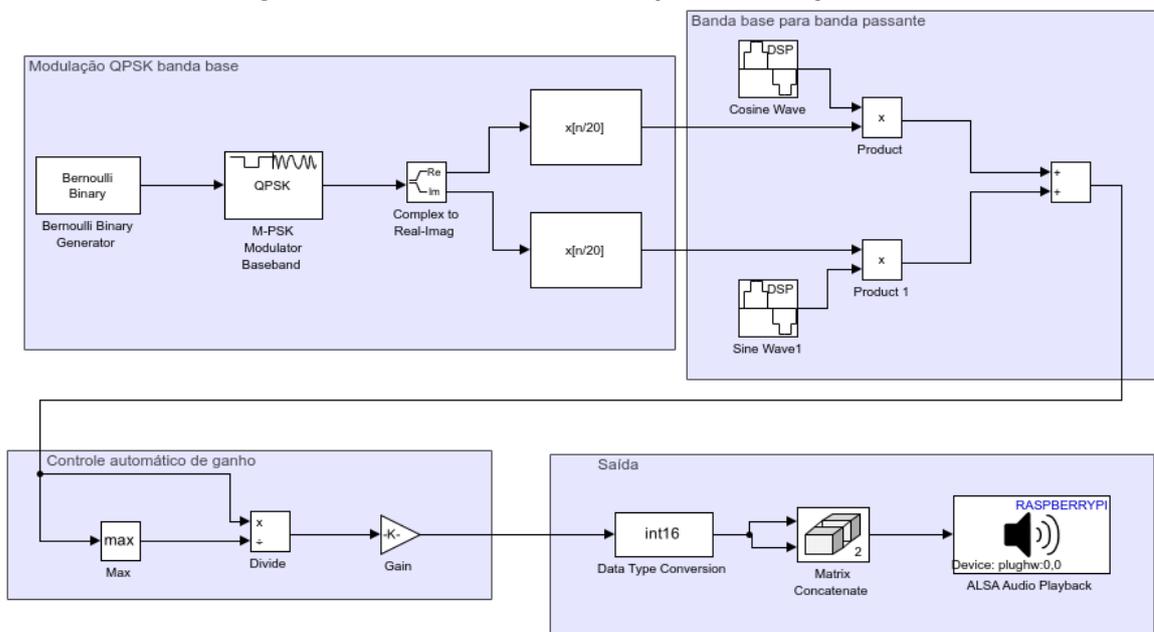
A separação das componentes em fase e em quadratura desempenha um papel no entendimento didático dessa técnica de modulação. Na modulação QPSK, o sinal modulado é formado pela soma de duas ondas ortogonais que transportam informações distintas, onde cada par de bits representa uma das quatro fases possíveis, como expresso pela Equação 20. Quando as componentes são examinadas separadamente, o entendimento da técnica de modulação se torna mais fácil pois cada componente pode ser analisado como um sinal independente.

$$s(t) = \sqrt{2E_s} \cos(2\pi f_c t + \phi_i), \quad \phi_i \in \left\{ \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4} \right\} \quad (20)$$

A Figura 26 apresenta o modelo do Simulink® para a primeira abordagem, onde o sinal é transmitido em sua forma final. O diagrama é implementado de forma similar àquele apresentado para a modulação BPSK, tendo como diferença a separação em componentes reais e imaginárias da saída do bloco *M-PSK Modulator Baseband*, configurado para mapear os bits em quatro fases distintas. Essa separação em componente real e imaginária corresponde às componentes em fase e em quadratura apresentadas por essa modulação.

Na conversão de banda base para banda passante, uma das componentes é multiplicada por um sinal cosseno e a outra por um sinal seno, utilizados como portadoras, o que garante a ortogonalidade entre os sinais. Em seguida, os sinais são somados para compor o sinal a ser transmitido.

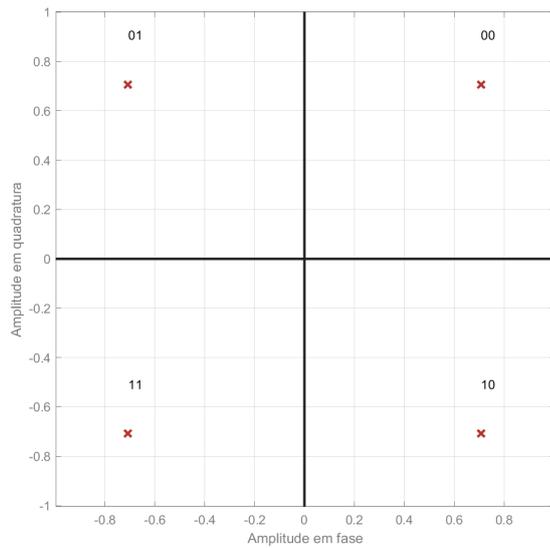
**Figura 26 – Modelo do Simulink® para modulação QPSK**



**Fonte: Autoria própria (2024).**

A Figura 27 apresenta o diagrama de constelação Gray da modulação QPSK. Diferentemente da modulação BPSK, na modulação QPSK os pontos não ficam sobre a reta real pois o sinal modulado utiliza duas componentes independentes: uma em fase (componente real) e outra em quadratura (componente imaginária).

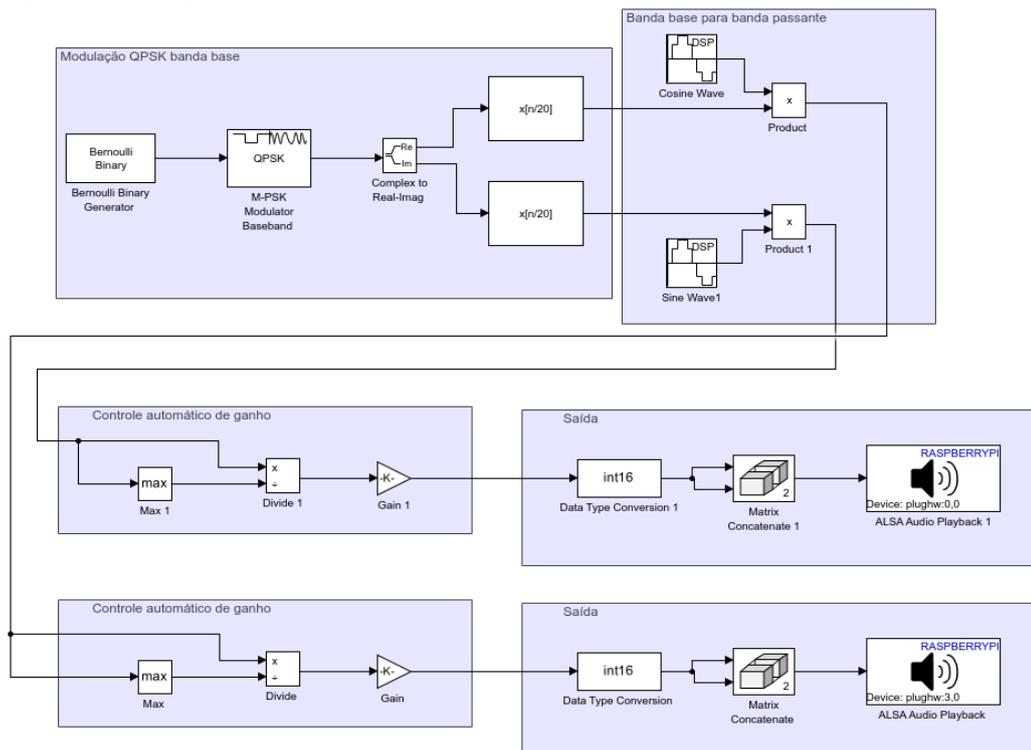
**Figura 27 – Diagrama de constelação da modulação QPSK**



**Fonte: Autoria própria (2024).**

Na segunda abordagem de modulação QPSK, as componentes foram mantidas separadas e examinadas individualmente. Neste caso, o bloco de soma não foi utilizado e as demais etapas de conversão para banda passante, ajuste de ganho e saída são implementadas de forma duplicada, com cada uma delas sendo aplicada a cada componente do sinal, como mostra a Figura 28. As configurações utilizadas em cada bloco são as mesmas apresentadas no experimento anterior, com a distinção de que aqui são necessárias duas saídas de áudio distintas, cada uma delas conectada a um canal do osciloscópio.

**Figura 28 – Diagrama de blocos para modulação QPSK com componentes separadas**



**Fonte: Autoria própria (2024).**

## 4 RESULTADOS

Os modelos desenvolvidos no Simulink® se destacam pela facilidade de uso e pela clareza na visualização dos conceitos teóricos. Um dos desafios encontrados foi a baixa tolerância do sistema a frequências mais altas (acima de 14 kHz), o que impôs uma restrição significativa à implementação de modulações com frequências de portadora típicas no mundo real, como as apresentadas em sistemas de radiodifusão. Para contornar essa limitação, os experimentos foram realizados com frequência de portadora de 10 kHz e amplitude de 1,26 V. Esse ajuste atende ao propósito didático, porém limita a execução de cenários reais de telecomunicações.

Outro ponto identificado foi o alto nível de ruído apresentado pelos sinais analisados, que pode ser atribuído a fatores como a qualidade da placa de som externa utilizada, mau contato entre os cabos e conectores utilizados e baixa resolução dos sinais digitais gerados.

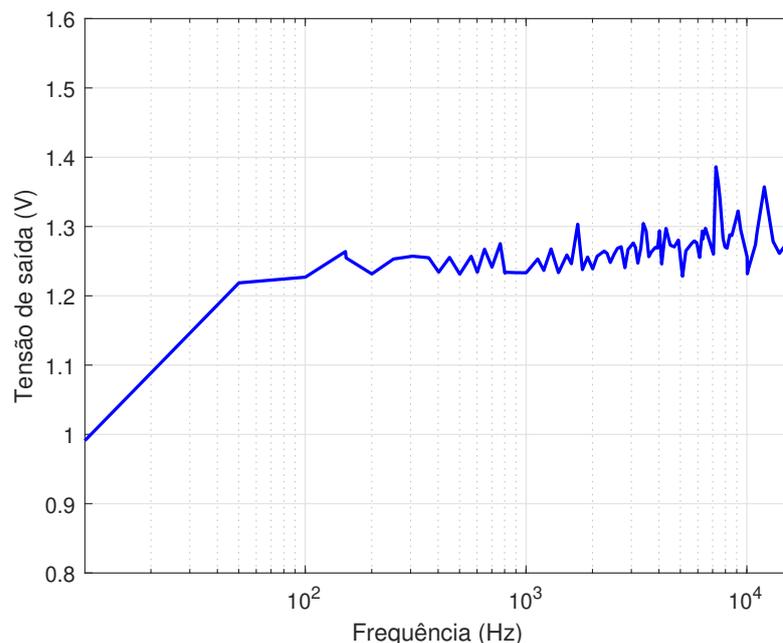
Os resultados obtidos em cada um dos experimentos encontram-se detalhados abaixo.

### 4.1 Resposta em frequência

A partir do processamento dos dados salvos nos arquivos CSV através do osciloscópio, foram gerados os gráficos de resposta em frequência referentes a cada uma das abordagens.

A Figura 29 apresenta o gráfico da resposta em frequência referente à primeira abordagem adotada, na qual gerou-se um sinal senoidal através do Simulink®, com amplitude fixa e variação progressiva da frequência.

**Figura 29 – Resposta em frequência do sistema em que a Raspberry Pi 3 atua como gerador de funções**

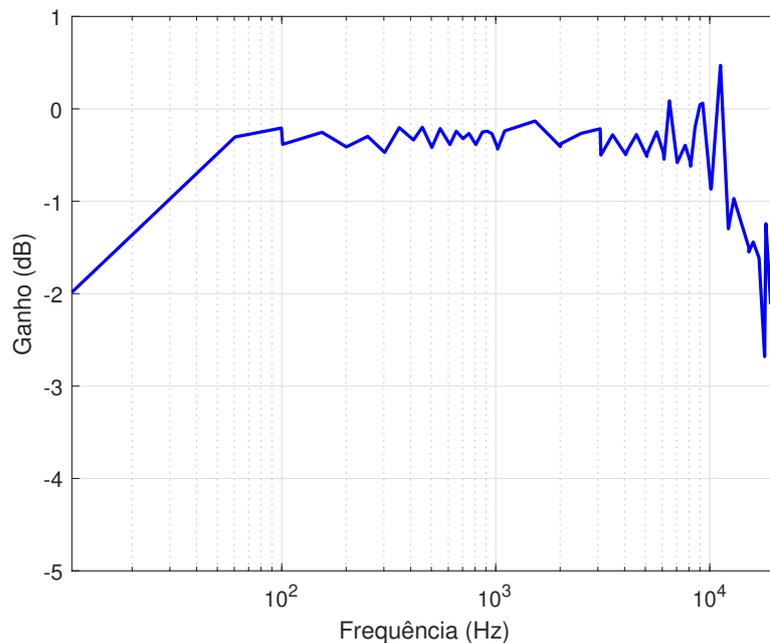


Fonte: Autoria própria (2025).

Como não foi adotada uma tensão de referência para o cálculo do ganho, optou-se por representar diretamente a amplitude da tensão de saída em função da frequência. Nota-se uma variação na tensão de saída, que permanece, na maior parte do tempo, entre 1,2 V e 1,3 V.

Por sua vez, a Figura 30 apresenta o gráfico de resposta em frequência obtido a partir da segunda abordagem, onde o sinal gerado por um gerador de funções é capturado pela entrada em microfone e externalizado pela saída de áudio de uma placa de som USB conectada à Raspberry Pi 3.

**Figura 30 – Resposta em frequência do sistema em que é utilizado um gerador de funções externo**



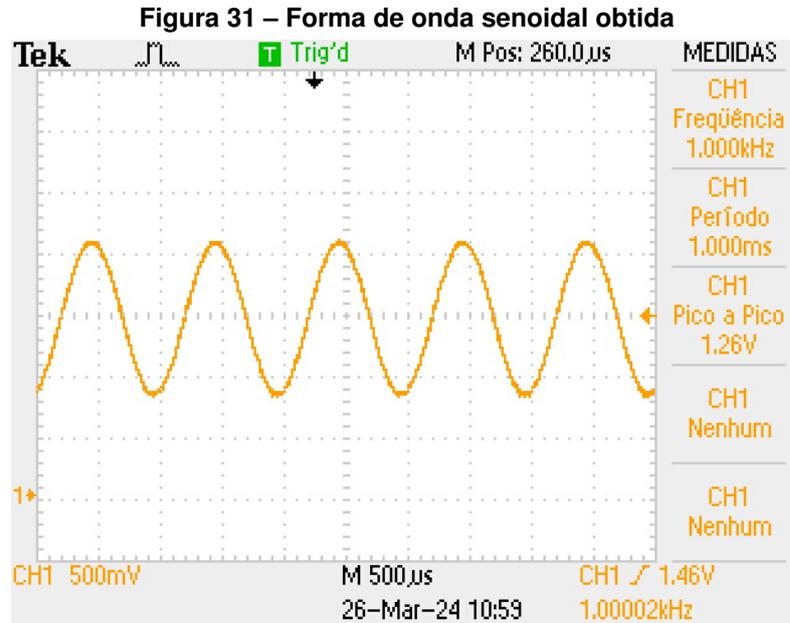
**Fonte: Autoria própria (2025).**

Assim como na abordagem anterior, a análise da resposta em frequência demonstra que o sistema apresenta pequenas variações ao longo da faixa de frequências analisada, onde o ganho permanece próximo de 0 dB entre 50 Hz e 10 kHz. A atenuação observada a partir de 10 kHz provavelmente ocorre devido aos componentes que se encontram entre a entrada e a saída do sinal, ou seja, os cabos e conectores utilizados, a placa de som USB e a própria Raspberry Pi.

Observou-se, a partir dos experimentos realizados, que a amplitude do sinal obtido variou de acordo com a frequência definida, evidenciando que a resposta do sistema não é perfeitamente plana. Essa característica pode ser visualizada nos gráficos das Figuras 29 e 30, os quais apresentam variações de amplitude ao longo do espectro de frequências analisado. Assim, constata-se que a resposta em frequência do sistema apresenta distorções que não foram consideradas neste estudo. Deve-se procurar, em um trabalho futuro, melhores respostas a fim de se compreender melhor as limitações e possibilidades do sistema analisado.

## 4.2 Gerador de funções

Dois sinais foram gerados a partir do segundo experimento: um sinal de onda senoidal e um sinal de onda quadrada. A Figura 31 mostra a forma de onda senoidal, com frequência de 1 kHz



Fonte: Autoria própria (2024).

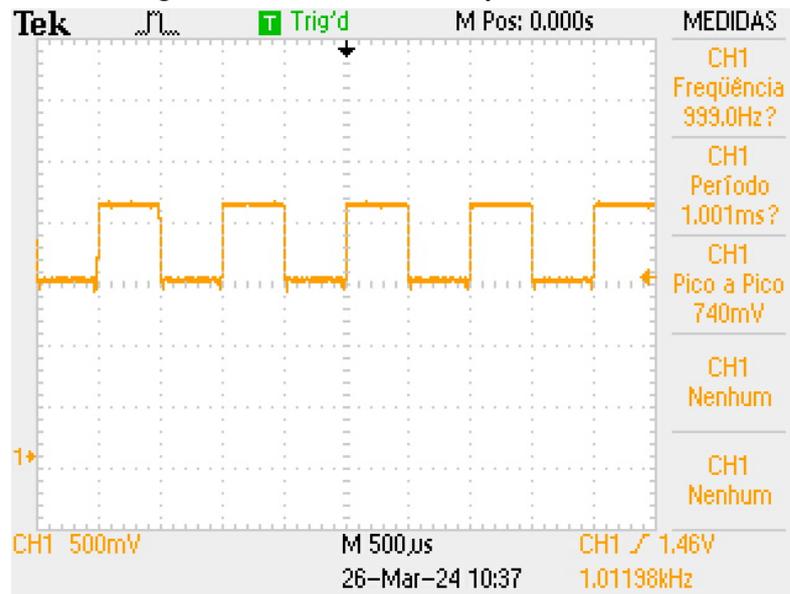
A partir do sinal senoidal de 1000 Hz gerado, verificou-se no osciloscópio que, quando a amplitude máxima no bloco *Sine Wave* é definida como  $2^{15} - 1$ , obtém-se no osciloscópio uma amplitude de 1,26 V de pico a pico. No Simulink®, a amplitude da senoide corresponde ao máximo valor de pico, ou seja, ao defini-la como  $2^{15} - 1$ , o valor de pico digital do sinal será de 32767 bits. A partir disso, é possível definir um fator de conversão, em Volts, para cada unidade digital, expresso na Equação 21, que pode ser utilizado para definir os valores de saída reais a partir das definições realizadas no Simulink®.

$$\kappa = \frac{1,26}{32767} = 38,45\mu\text{V} \quad (21)$$

em que  $\kappa$  é o fator de conversão. Para trabalhos futuros, sugere-se verificar se todos os valores de conversão são reproduzíveis.

A Figura 32 mostra a forma de onda quadrada, também com frequência de 1 kHz. Para a geração de onda quadrada, observou-se que mesmo aplicando uma amplitude de  $2^{15} - 1$ , a tensão de pico obtida foi de 740 mV.

Figura 32 – Forma de onda quadrada obtida

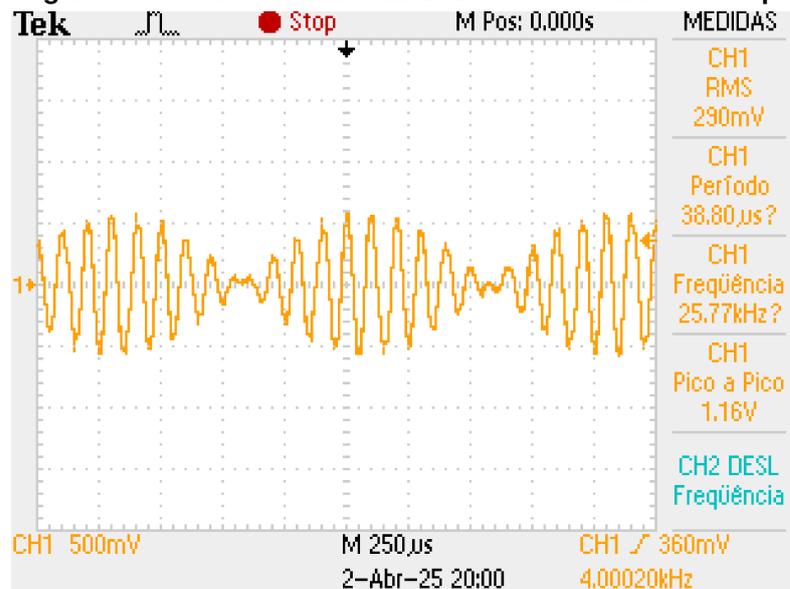


Fonte: Autoria própria (2024).

### 4.3 Transmissor e receptor AM

A Figura 33 mostra o sinal modulado no domínio do tempo. O gráfico mostra a modulação em amplitude, onde a envoltória do sinal segue o formato do sinal da mensagem com frequência de 500 Hz e amplitude configurada como  $2^{15} - 1$ , enquanto a portadora, com frequência de 10 kHz e amplitude  $2^{15} - 1$ , é visível dentro da envoltória, modulada em termos de amplitude e com amplitude máxima de 1,16 V.

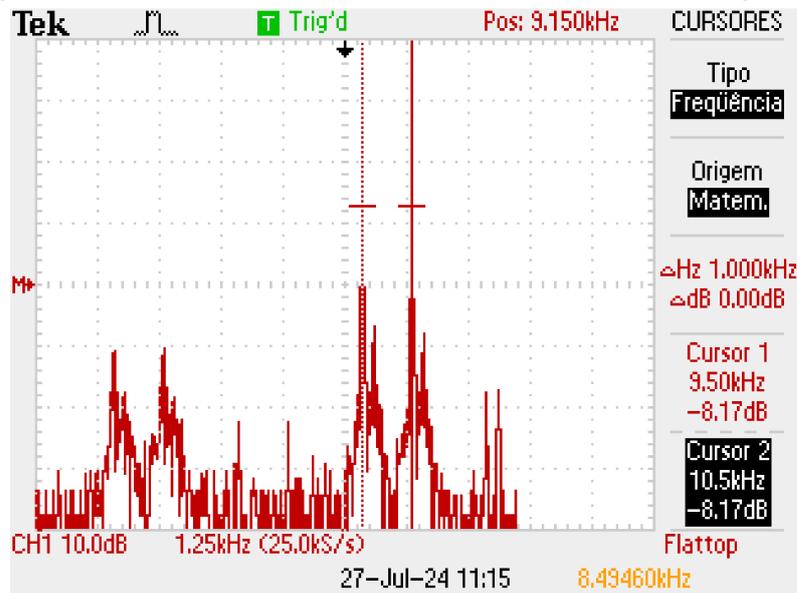
Figura 33 – Sinal modulado AM DSB-SC no domínio do tempo



Fonte: Autoria própria (2024).

Através da função matemática FFT do osciloscópio, observou-se o espectro de frequência apresentado na Figura 34, que apresenta duas componentes de frequência principais com amplitude de  $-8,17 \text{ dB}V_{rms}$ , conforme o manual do osciloscópio. Esse valor corresponde a  $1,1 V_{pp}$ , valor aproximado da tensão de pico a pico do sinal da mensagem. O cursor 1 indica a banda lateral inferior, em  $9,5 \text{ kHz}$ , enquanto o cursor 2 indica a banda lateral superior, em  $10,5 \text{ kHz}$ . A portadora em si está ausente, o que caracteriza a modulação com portadora suprimida, e também pode-se observar que as bandas laterais estão afastadas de  $500 \text{ Hz}$ , o que indica uma modulação tonal com frequência de  $500 \text{ Hz}$ .

**Figura 34 – Espectro do sinal modulado AM DSB-SC com ênfase nas componentes principais**



Fonte: Autoria própria (2024).

Destaca-se no espectro de frequência a presença de componentes espectrais adicionais de menor intensidade, como mostra a Figura 35, em que os cursores estão posicionados sobre as frequências de  $4,5 \text{ kHz}$  e  $5,5 \text{ kHz}$ , com amplitude de  $-35,4 \text{ dB}$ , uma diferença superior a  $27 \text{ dB}$  em comparação às bandas laterais principais. Esses sinais espúrios podem ter origem em componentes de frequência maiores que a frequência de Nyquist, intermodulação de segunda ordem e *fading* não uniforme, e sua análise não faz parte do escopo deste trabalho.

A Figura 36 mostra o sinal demodulado AM-DSB-SC, em que verifica-se a forma de onda da mensagem transmitida, com frequência de  $497,5 \text{ Hz}$  e amplitude  $368 \text{ mV}$ . Conforme apresentado pela Equação 8, a amplitude do sinal demodulado é de aproximadamente um quarto da amplitude do sinal original.

Figura 35 – Espectro do sinal modulado AM DSB-SC com ênfase nas componentes espectrais de menor intensidade

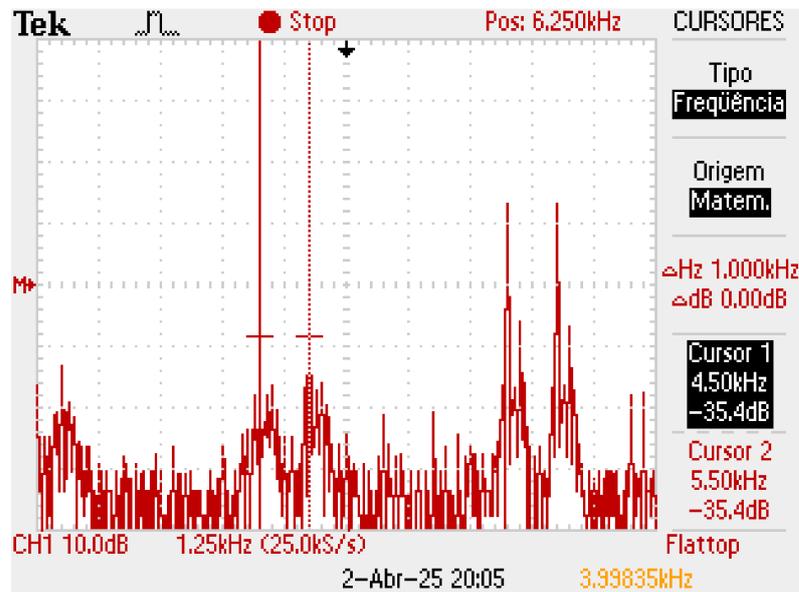
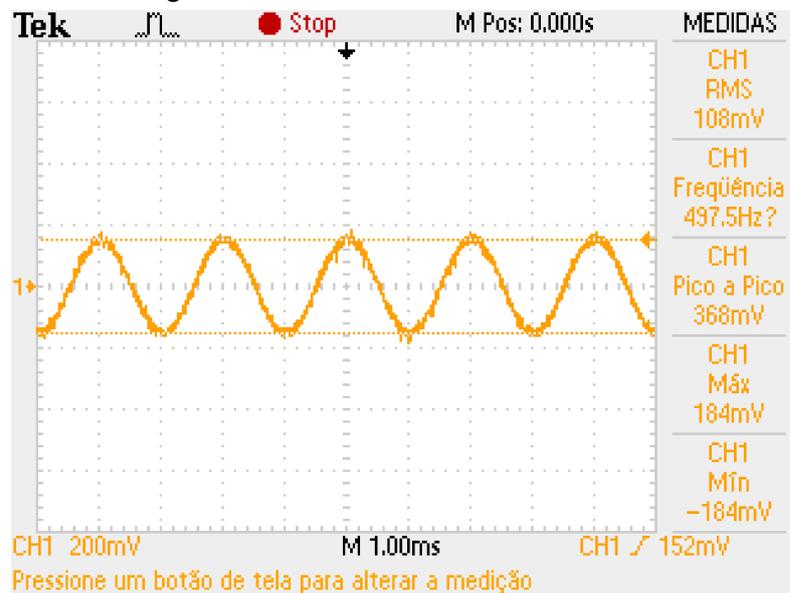


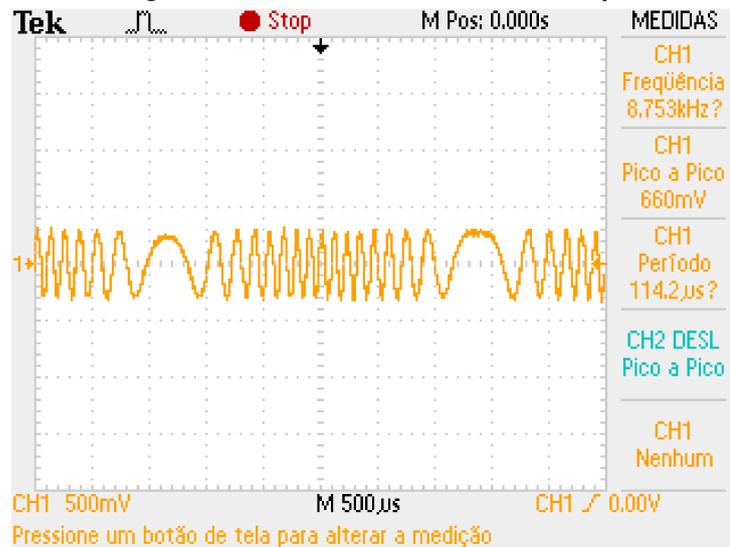
Figura 36 – Sinal demodulado AM DSB-SC



#### 4.4 Modulador FM

A Figura 37 apresenta o sinal modulado em frequência visto no osciloscópio após a execução do modelo apresentado na Figura 19. O gráfico exibe um sinal característico da modulação FM, onde a frequência do sinal transmitido varia em função do sinal modulante, com desvio de frequência de 5 kHz e amplitude de pico a pico de 660 mV.

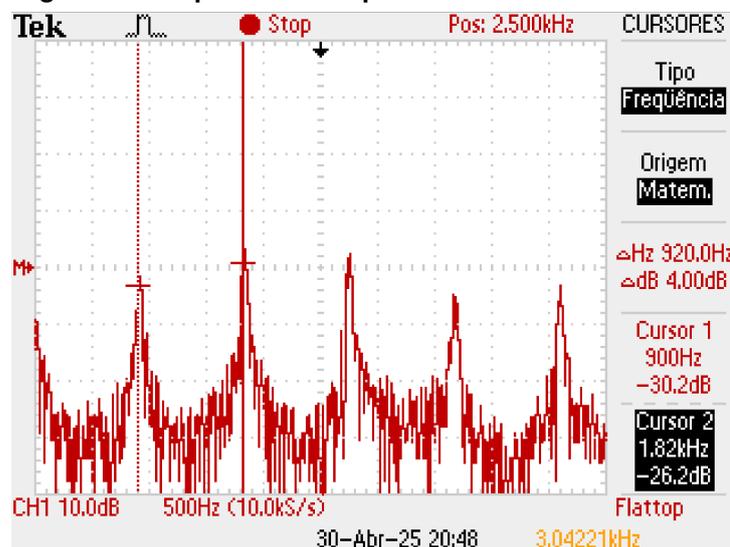
Figura 37 – Sinal modulado FM no tempo



Fonte: Autoria própria (2024).

Logo, a Figura 38 apresenta o espectro de frequência do mesmo sinal modulado em FM. É possível observar um espectro característico da modulação FM, com múltiplas componentes distribuídas ao redor da frequência central do sinal. Como o modelo não inclui uma etapa de translação para banda passante, o sinal modulado encontra-se ainda em banda base, razão pela qual as componentes espectrais aparecem próximas de 0 Hz.

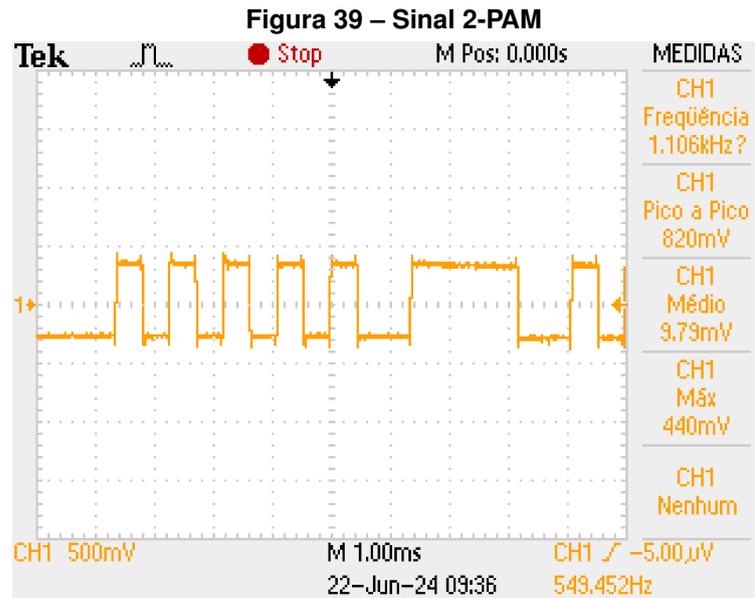
Figura 38 – Espectro de frequência do sinal modulado FM



Fonte: Autoria própria (2024).

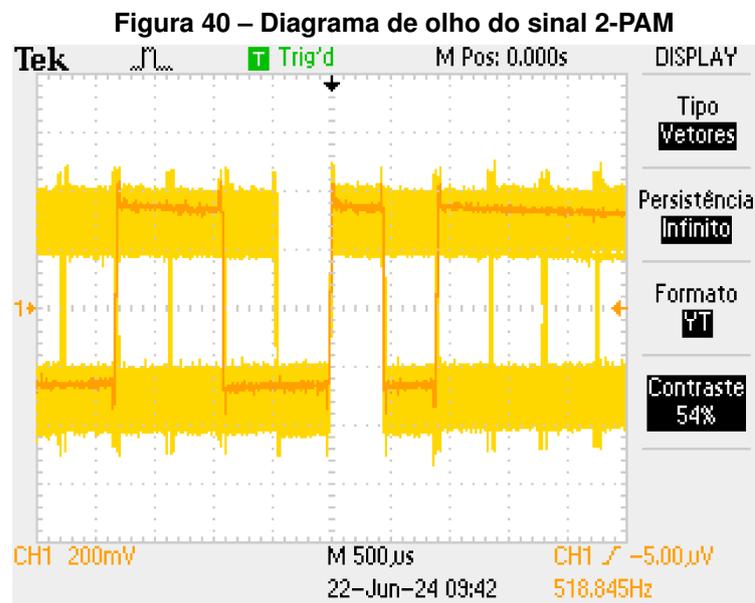
#### 4.5 Modulador 2-PAM e 4-PAM

A Figura 39 apresenta o resultado da execução do modulador 2-PAM. O sinal oscila entre dois níveis de tensão, um negativo, representando o bit 0, e outro positivo, representando o bit 1.



Fonte: Autoria própria (2024).

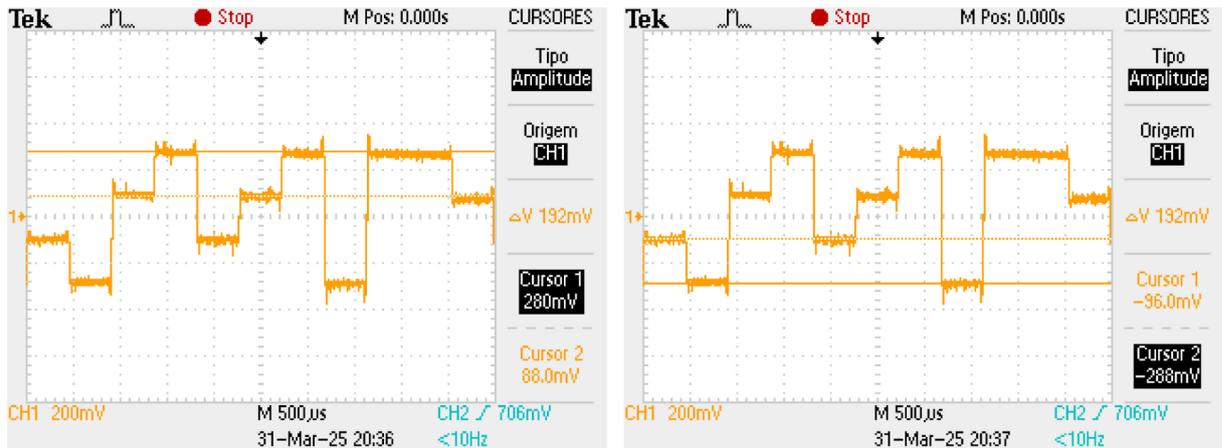
Para uma melhor análise do sinal obtido, utilizou-se da ferramenta gráfica diagrama do olho, que reflete as transições de um bit para outro no tempo e mostra como os sinais em diferentes instantes de tempo se sobrepõem. Na captura de tela do osciloscópio, exibida na Figura 40, é possível perceber que, mesmo com a presença de ruído, o sinal possui uma boa abertura do olho, o que permite a correta detecção entre os níveis do sinal.



Fonte: Autoria própria (2024).

Já a Figura 41 mostra o resultado da execução do modulador 4-PAM. Observa-se a existência de quatro níveis de amplitude, referentes aos dois bits para cada símbolo, sendo eles 280 mV para o par de bits {11}, 88 mV para {10}, -96 mV para {01} e -288 mV para {11}.

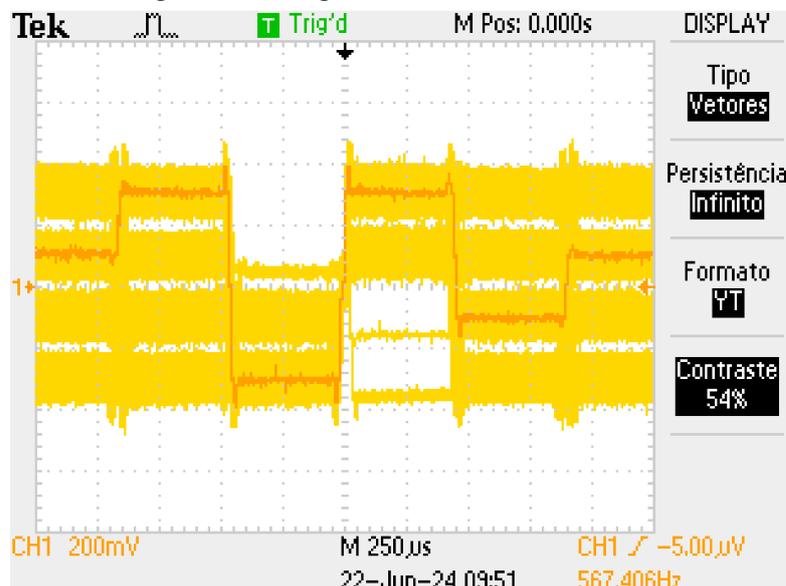
Figura 41 – Sinal 4-PAM com indicação dos quatro níveis de tensão



Fonte: Autoria própria (2025).

A Figura 42 mostra o diagrama do olho da modulação 4-PAM. Verifica-se que o aumento no número de níveis de amplitude impacta diretamente a abertura vertical do olho, que é menor em comparação com o 2-PAM. Esse comportamento indica uma maior sensibilidade à presença de ruídos e, por consequência, haverá mais erros na detecção do sinal.

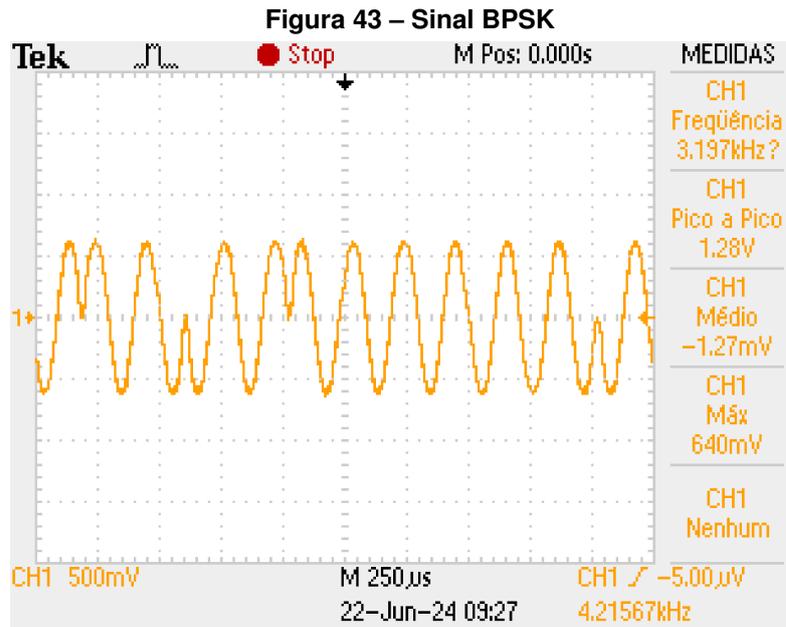
Figura 42 – Diagrama de olho do sinal 4-PAM



Fonte: Autoria própria (2024).

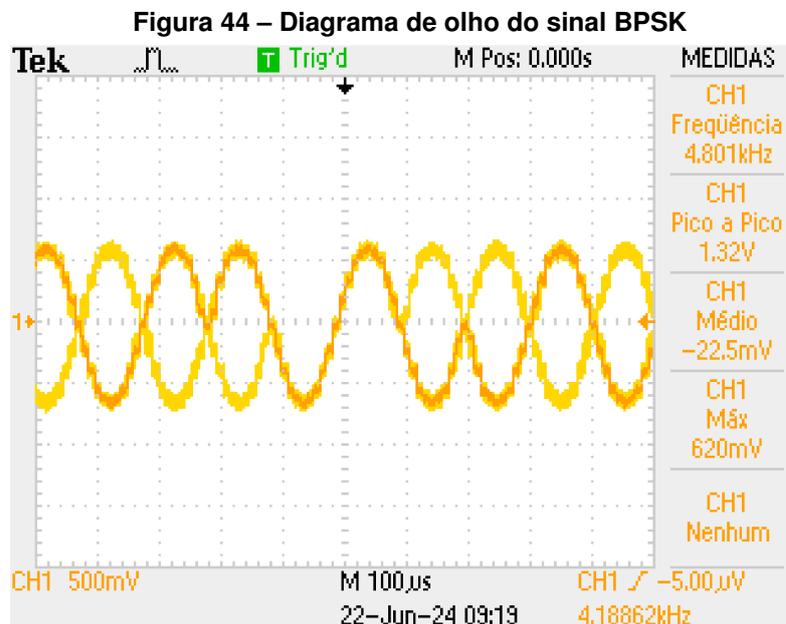
#### 4.6 Modulador BPSK e QPSK

Após a configuração do sistema BPSK, o sinal foi visualizado no osciloscópio, como mostra a Figura 43, onde se observa claramente as inversões de fase de  $180^\circ$  associadas à troca de bits entre 0 e 1. O sinal da portadora foi configurado para uma frequência de 5000 Hz e amplitude de  $2^{15} - 1$ .



Fonte: Autoria própria (2024).

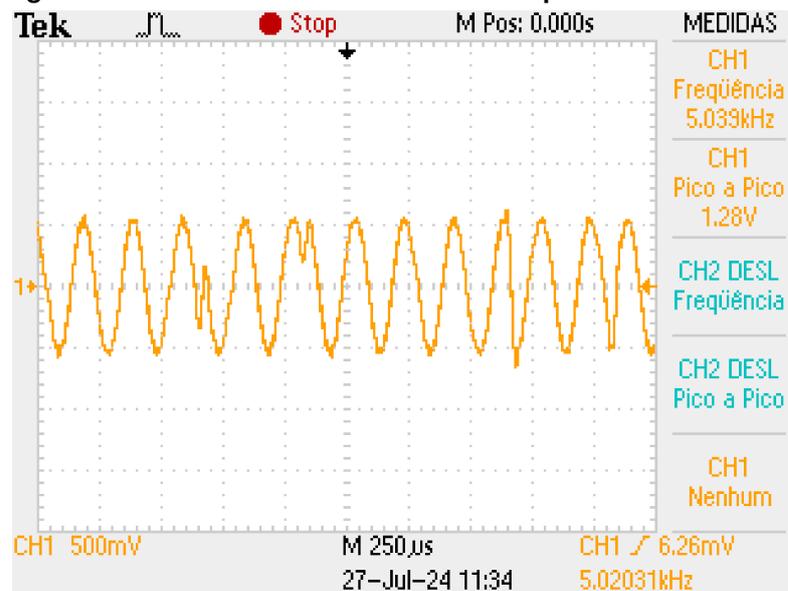
Com a utilização da ferramenta gráfica de diagrama de olho, exibida na Figura 44, verifica-se uma boa abertura do olho, o que indica que o sinal apresenta uma boa resistência ao ruído e pode ser facilmente detectado.



Fonte: Autoria própria (2024).

A execução da primeira abordagem de modulação QPSK teve como resultado o sinal exibido na Figura 45, onde ambas as componentes, em fase e em quadratura, representadas, respectivamente, por um seno de amplitude  $2^{14} - 1$  e frequência 5 kHz e por um cosseno de amplitude  $2^{14} - 1$  e frequência 5 kHz, são somadas no modelo Simulink® e transmitidas juntas. As mudanças de fase do sinal refletem a mudança entre os bits transmitidos (00, 01, 10 e 11).

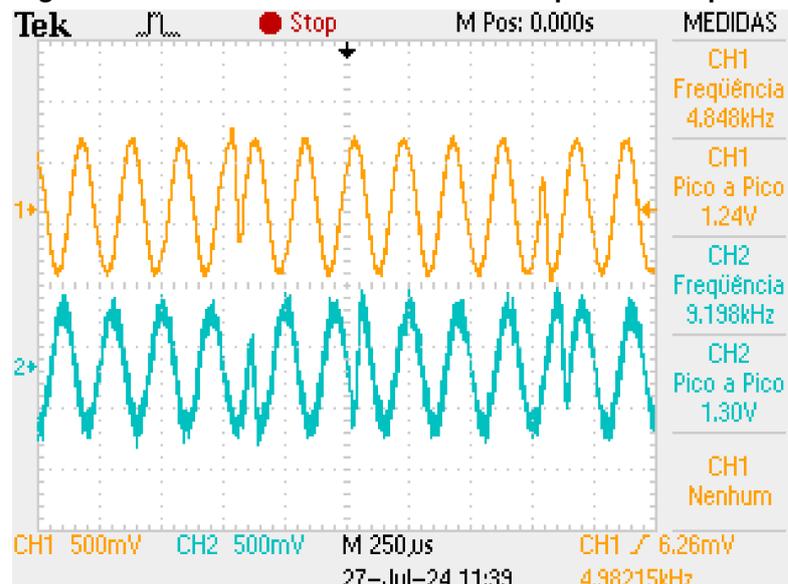
**Figura 45 – Forma de onda QPSK com componentes combinadas**



Fonte: Autoria própria (2024).

A segunda abordagem da modulação QPSK apresenta como resultado os sinais exibidos na Figura 46, onde as componentes em fase e em quadratura são visualizadas em canais distintos. É possível verificar como cada uma das ondas portadoras (seno e cosseno) modula os bits de entrada separadamente, sendo possível observar as mudanças nos dados transmitidos através das transições observadas em cada componente.

**Figura 46 – Forma de onda QPSK com componentes separadas**

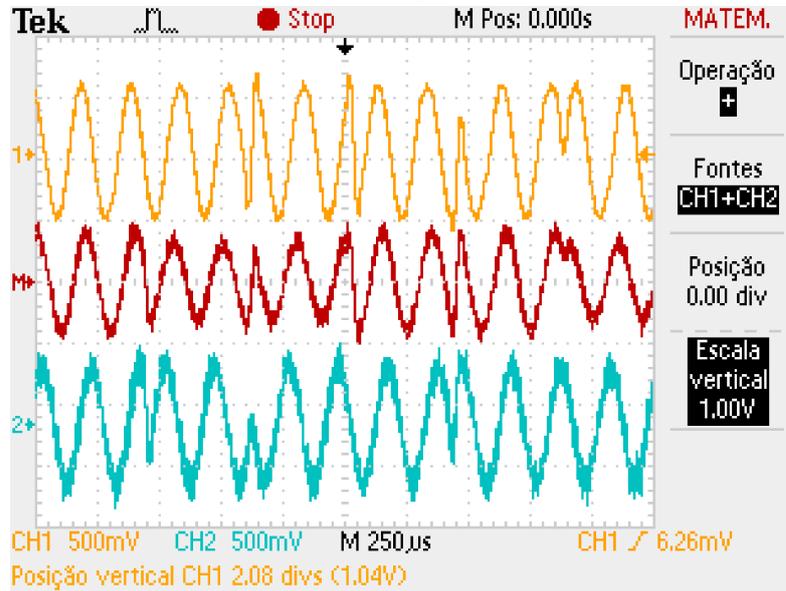


Fonte: Autoria própria (2024).

Através do uso da ferramenta *math menu* do osciloscópio, as componentes foram somadas para a obtenção de um sinal composto, como aquele obtido na Figura 45, semelhante ao que seria observado em um sistema de recepção real. Em azul, observa-se a componente em fase, em amarelo a componente em quadratura, e em vermelho a soma das duas componentes,

que gera um sinal que reflete os quatro estados de fase característicos da modulação QPSK, como mostra a Figura 47.

Figura 47 – Forma de onda QPSK com componentes somadas através do *math menu*



Fonte: Autoria própria (2024).

## 5 CONCLUSÃO

O presente trabalho teve como objetivo a implementação de experimentos práticos envolvendo tópicos da disciplina de Fundamentos de Comunicações, utilizando a Raspberry Pi 3 em conjunto com os softwares MATLAB® e Simulink®. Através de ferramentas de hardware e software acessíveis para a comunidade acadêmica, os conteúdos abordados na disciplina foram explorados de forma simples.

Diferente da abordagem frequentemente utilizada em disciplinas de processamento de sinais e comunicações, onde os experimentos didáticos são desenvolvidos apenas através de ferramentas computacionais, com a abordagem deste trabalho foi possível analisar os sinais gerados através de instrumentos reais e comuns ao ambiente de estudo da engenharia eletrônica. O osciloscópio, por exemplo, é um dos principais instrumentos utilizados ao longo deste trabalho e encontra-se presente na maioria dos laboratórios de engenharia eletrônica.

Os experimentos aqui propostos complementam a abordagem tradicional, atuando como uma ferramenta didática que auxilia o processo de entendimento do objeto de estudo. A abordagem utilizada, com a construção de modelos através de diagramas de blocos, permite que sistemas complexos sejam implementados de maneira fácil e rápida.

Além dos experimentos de simulação, este trabalho também incluiu o estudo da resposta em frequência do sistema por meio da observação do espectro dos sinais gerados e processados. Essa análise permitiu avaliar o comportamento dos sistemas e identificar as limitações da faixa de operação da placa de som utilizada.

As tecnologias utilizadas apresentaram vantagens, como a interface intuitiva do Simulink®, que não exige o conhecimento de linguagens de programação, a flexibilidade apresentada pelo software para o desenvolvimento de diversos tipos de modulação e processamento de sinais, e o baixo custo das ferramentas utilizadas.

Entretanto, algumas dificuldades foram evidentes, como a restrição das faixas de frequência suportadas pela placa de som externa e a presença de ruídos. As limitações, embora importantes, não comprometeram a implementação dos experimentos, que demonstraram de forma prática os conteúdos estudados na disciplina.

Como perspectiva futura, sugere-se a implementação de experimentos de transmissão sem fio, que ampliem a área de estudo e permitam a análise de características importantes, como a propagação de sinais em ambientes reais. Uma forma de desenvolver essa abordagem é através da utilização do RTL-SDR, um dispositivo de rádio definido por software de baixo custo que pode ser utilizado nas Raspberry Pi 3 com modelos desenvolvidos no Simulink®.

## REFERÊNCIAS

- ALSA Project. **Advanced Linux Sound Architecture (ALSA) project homepage**. ALSA Project, 2020. Acesso em: 23 de abril de 2024. Disponível em: [https://www.alsa-project.org/wiki/Main\\_Page](https://www.alsa-project.org/wiki/Main_Page).
- BALL, D.; NAIK, N.; JENKINS, P. Lightweight and Cost-Effective Spectrum Analyser Based on Software Defined Radio and Raspberry Pi. In: **2017 European Modelling Symposium (EMS)**. [S.l.: s.n.], 2017. p. 260–266.
- BARBOSA, E. F.; MOURA, D. d. Metodologias ativas de aprendizagem no ensino de engenharia. In: **Anais International Conference on Engineering and Technology Education, Cairo, Egito**. [S.l.: s.n.], 2014. v. 13, p. 110–116.
- DESLAURIERS, L. *et al.* Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. In: **Proceedings of the National Academy of Sciences**. [S.l.]: National Academy Sciences, 2019. v. 116, n. 39, p. 19251–19257.
- ELETROINFO CIA. **Placa de Som USB 5.1 Virtual AUSB51 Vinik**. 2024. Disponível em: <https://www.eletoinfocia.com.br/placa-de-som-usb-5-1-virtual-ausb51-vinik>. Acesso em: 13 jan. 2025.
- GADGETS, G. S. **HackRF One - An open source SDR platform**. 2025. Disponível em: <https://greatscottgadgets.com/hackrf/>. Acesso em: 17 abr. 2025.
- HAYKIN, S.; MOHER, M. **Introdução aos Sistemas de Comunicação**. [S.l.]: Bookman Companhia Editora, 2008. ISBN 978-85-7780-329-3.
- KIM, J.-W. *et al.* Design and Implementation Scheme of QSFP28 Optical Transceiver for Long-Reach Transmission Using PAM4 Modulation. **Applied Sciences**, v. 11, n. 6, 2021. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/11/6/2803>.
- LATHI, B.; DING, Z. **Sistemas de Comunicações Analógicos e Digitais Modernos**. [S.l.]: Grupo Editorial Nacional, 2012. ISBN 978-85-216-2024-3.
- LAUFER, C. **The Hobbyist's Guide to the RTL-SDR: Really cheap Software Defined Radio**. Upper Saddle River, Nova Jersey, EUA: CreateSpace Independent Publishing Platform, 2015. ISBN 9781514716694.
- MathWorks. **MATLAB: Math. graphics. programming**. MathWorks, 2024. Acesso em: 23 de abril de 2024. Disponível em: [https://www.mathworks.com/products/matlab.html?s\\_tid=hp\\_ff\\_p\\_matlab](https://www.mathworks.com/products/matlab.html?s_tid=hp_ff_p_matlab).
- MathWorks. **MATLAB: Design. simulate. deploy**. MathWorks, 2024. Acesso em: 23 de abril de 2024. Disponível em: [https://www.mathworks.com/products/simulink.html?s\\_tid=hp\\_ff\\_p\\_simulink](https://www.mathworks.com/products/simulink.html?s_tid=hp_ff_p_simulink).
- MathWorks. **Raspberry Pi Support from MATLAB: Acquire sensor and image data from your connected raspberry pi**. MathWorks, 2024. Acesso em: 19 de março de 2024. Disponível em: <https://www.mathworks.com/hardware-support/raspberry-pi-matlab.html>.
- OPPENHEIM, A. V.; WILLISKY, A. S. **Sinais e Sistemas**. São Paulo, São Paulo, Brasil: Pearson, 2010. ISBN 978-85-7605-504-4.

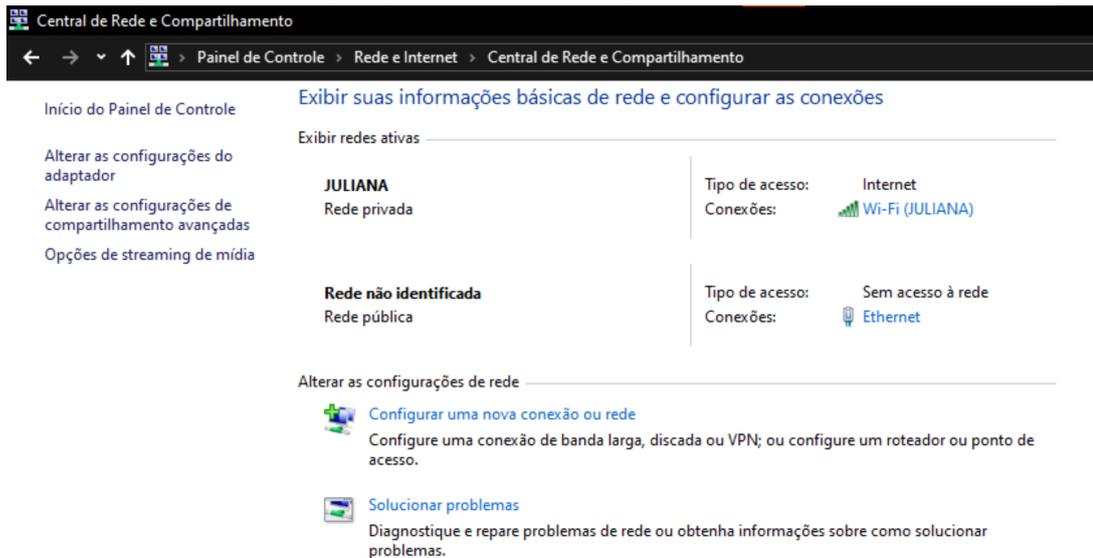
- PASOLINI, G.; BAZZI, A.; MIRABELLA, M. **Simulink Defined Radio with Raspberry Pi2**. 2016. Acesso em: 19 de março de 2024. Disponível em: <https://www.simulinkdefinedradio.com/>.
- PASOLINI, G.; BAZZI, A.; ZABINI, F. A raspberry pi-based platform for signal processing education [sp education]. **IEEE Signal Processing Magazine**, v. 34, n. 4, p. 151–158, 2017.
- RAMOS, M. A. *et al.* Software defined radio, a perspective from education. **Frontiers in Education**, Volume 8 - 2023, 2024. ISSN 2504-284X. Disponível em: <https://www.frontiersin.org/journals/education/articles/10.3389/educ.2023.1228610>.
- Raspberry Pi Ltd. **Raspberry Pi Documentation: Getting Started**. Raspberry Pi Ltd, 2024. Acesso em: 04 de abril de 2024. Disponível em: <https://www.raspberrypi.com/documentation/computers/getting-started.html>.
- Raspberry Pi Ltd. **Raspberry Pi Documentation: Raspberry Pi hardware**. Raspberry Pi Ltd, 2024. Acesso em: 04 de abril de 2024. Disponível em: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>.
- RUTTKAMP, U. A. **Welcome to the DHCP Server for Windows**. 2025. Disponível em: <https://www.dhcpserver.de/cms/>. Acesso em: 13 jan. 2025.
- SKLAR, B. **Digital Communications: Fundamentals and applications**. Upper Saddle River, Nova Jersey, EUA: Prentice Hall, 2001. ISBN 0130847887.
- TEKTRONIX. **Digital Storage Oscilloscopes**. 2009. Disponível em: [https://br.mouser.com/datasheet/2/403/3GW\\_15314\\_6-2308218.pdf](https://br.mouser.com/datasheet/2/403/3GW_15314_6-2308218.pdf). Acesso em: 14 abr. 2025.
- TOMINAGA, E. N. **Desenvolvimento de experimentos com o Raspberry Pi para o ensino de comunicações**. 2018. Monografia (Trabalho de conclusão de curso) — Universidade Federal de Santa Catarina, 2018.
- UPTON, E. *et al.* **Learning Computer Architecture with Raspberry Pi**. Indianapolis, Indiana, EUA: John Wiley Sons, Inc, 2016. ISBN 978-1-119-18394-5.

## **APÊNDICE A – Configurações preliminares**

## A.1 Conexão da Raspberry Pi 3 através de cabo Ethernet

Primeiramente, a Raspberry Pi 3 deve ser conectada ao computador utilizando o cabo Ethernet, e em seguida deve-se acessar a central de rede e compartilhamento do Windows 10, como mostra a Figura 48. Uma rede local é criada, identificada como Ethernet, que deve ser selecionada para que suas propriedades sejam acessadas.

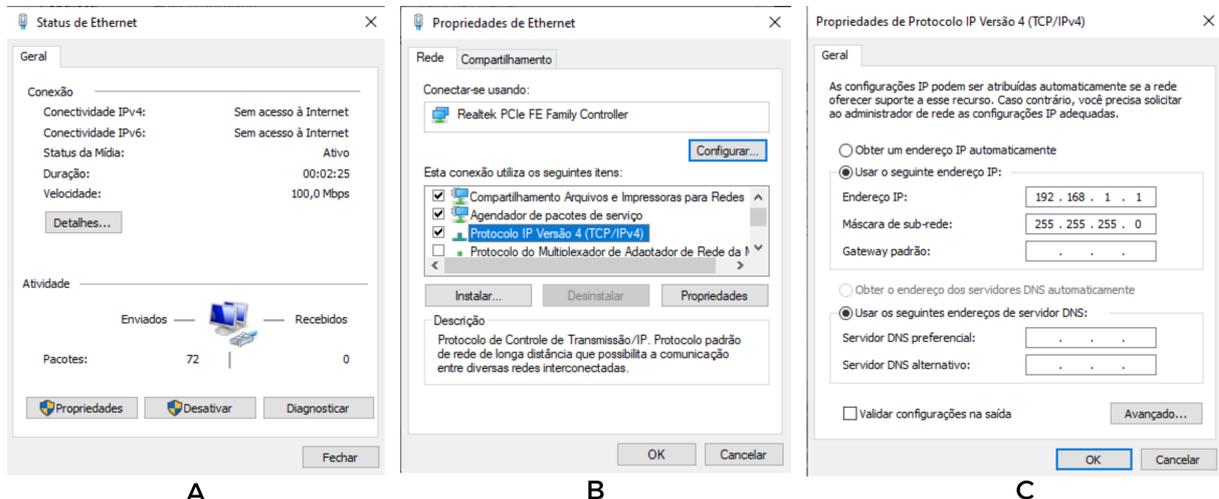
**Figura 48 – Configurações de rede e compartilhamento do Windows 10**



**Fonte: Autoria Própria (2025).**

Ao selecionar a rede Ethernet, a janela de status exibida na Figura 49A será aberta. Nela, deve-se selecionar o botão “Propriedades”, e em sequência, a janela exibida na Figura 49B será aberta. É preciso então selecionar a opção “Protocolo IP Versão 4 (TCP/IPv4)” e clicar em “Propriedades”. Na janela de propriedades, exibida na Figura 49C, deve-se selecionar a opção “Usar o seguinte endereço IP” e, em seguida, digitar um endereço IP adequado.

**Figura 49 – Rede Ethernet estabelecida entre o computador e a Raspberry Pi 3**



A

B

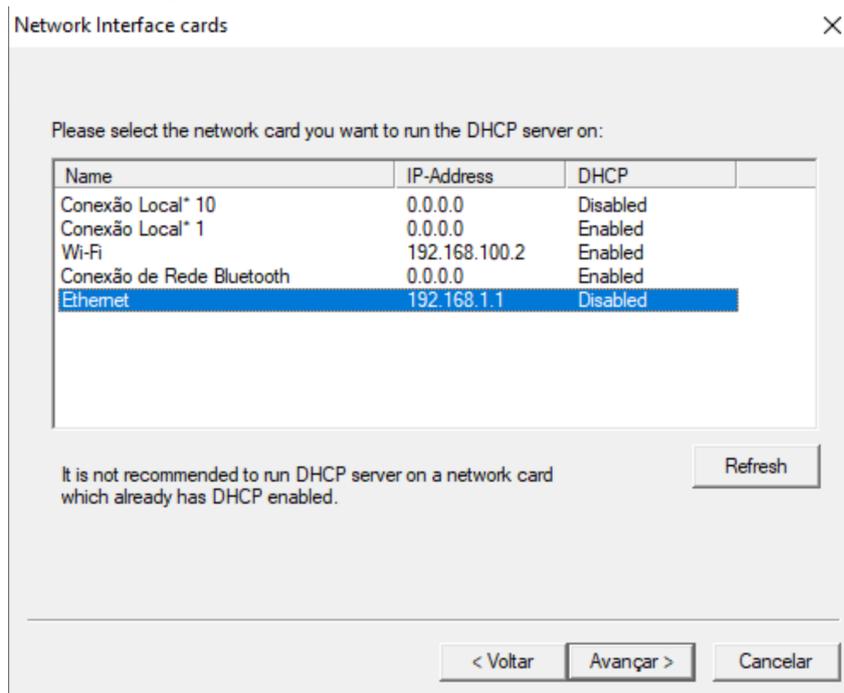
C

**Fonte: Autoria Própria (2025).**

Realizados esses procedimentos, um endereço IP deve ser atribuído através de um servidor DHCP. Para isso, foi utilizado o software DHCP Server for Windows, disponível gratuitamente para download (Ruttkamp, 2025).

Após o download do DHCP Server, deve-se abrir o arquivo executável do assistente de configuração. Na primeira janela, deve-se selecionar a conexão Ethernet que possui o mesmo endereço IP definido nas configurações de rede do Windows, como mostra a Figura 50.

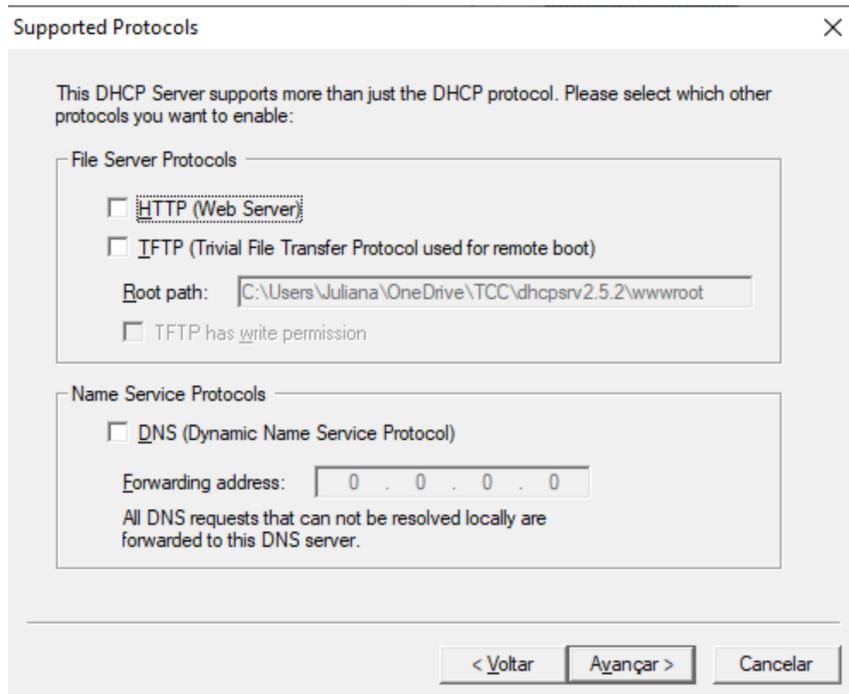
**Figura 50 – Redes disponíveis no DHCP Server**



**Fonte: Autoria Própria (2025).**

Em seguida, é possível selecionar outros protocolos, como HTTP, TFTP e DNS, como mostra a Figura 51. Para essa aplicação, o único protocolo necessário é o DHCP, de modo que nenhum outro foi selecionado.

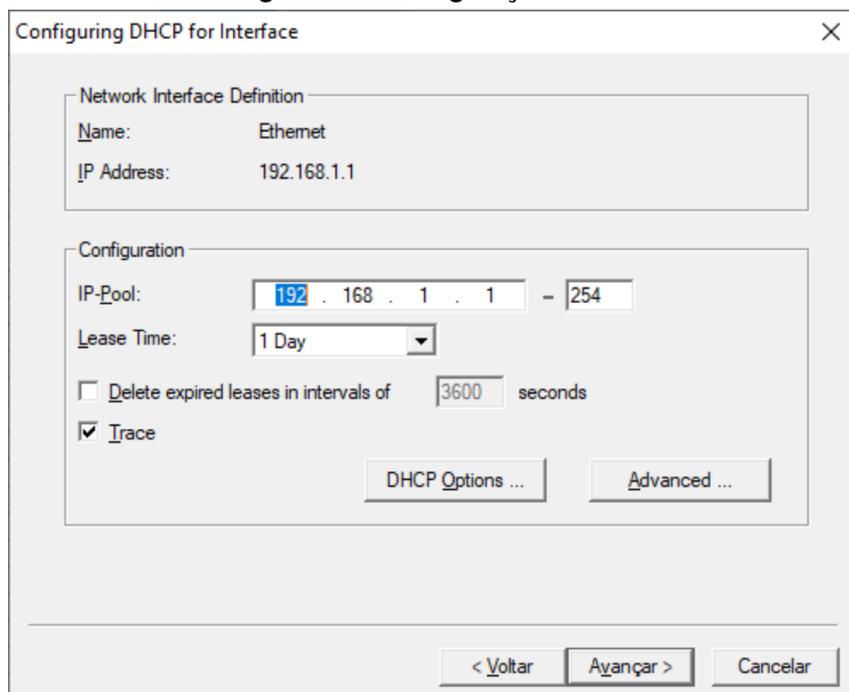
**Figura 51 – Protocolos suportados pelo DHCP Server**



Fonte: Autoria Própria (2025).

Na janela seguinte, exibida na Figura 52, é definida a faixa de endereços IP, onde um cliente receberá um dos endereços disponíveis contidos nessa faixa. Aqui, definiu-se o conjunto de “192.168.1.1” até “192.168.1.254”.

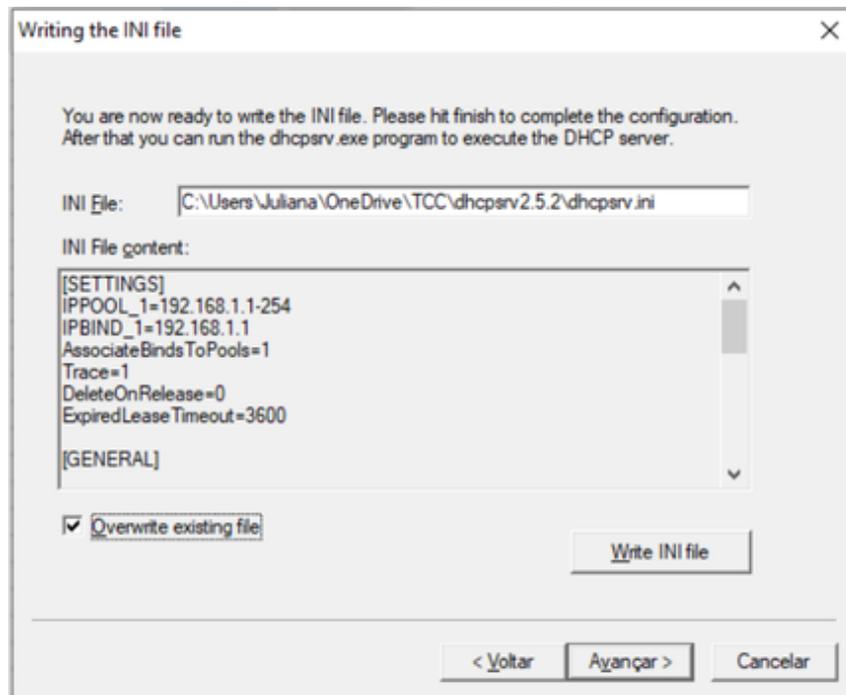
**Figura 52 – Configuração DHCP**



Fonte: Autoria Própria (2025).

As configurações resultantes são gravadas em um arquivo INI, que deve ser salvo na mesma pasta onde se encontra o arquivo executável do servidor. A Figura 53 mostra a janela de gravação do arquivo INI, onde deve-se selecionar a opção “Overwrite existing file” e, em seguida, “Write INI file”.

**Figura 53 – Gravação do arquivo INI**

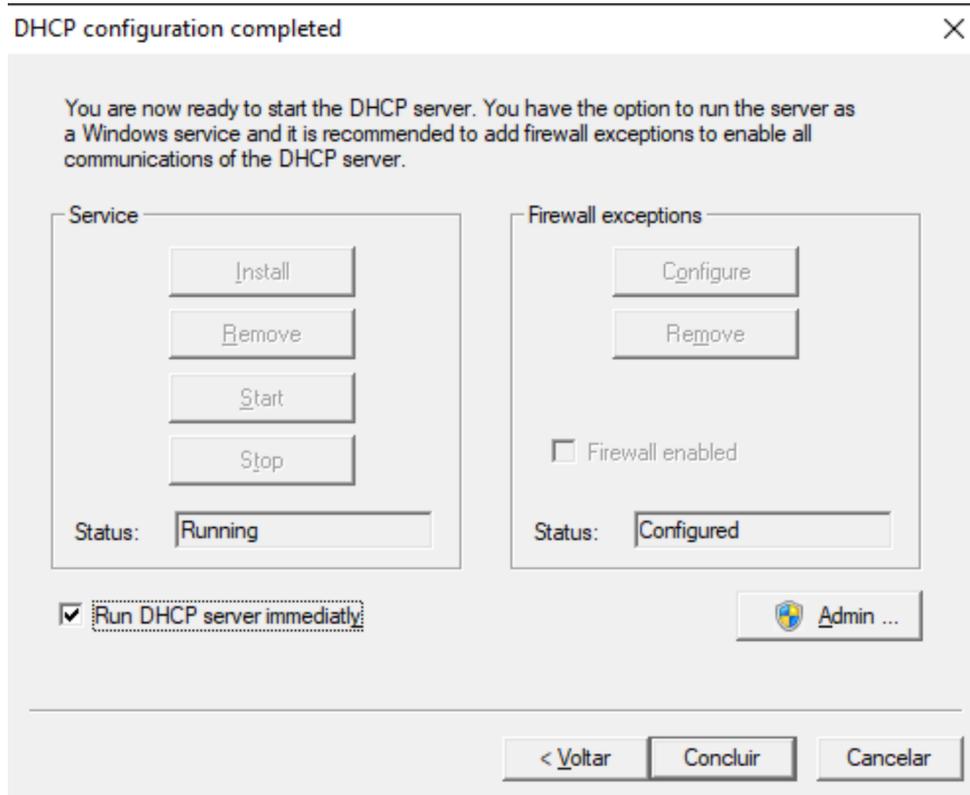


**Fonte: Autoria Própria (2025).**

Após a gravação do arquivo INI, a configuração DHCP encontra-se concluída. Na Figura 54, deve-se selecionar os botões “Install” e “Configure” e aguardar até que os status em cada um dos campos sejam “Running” e “Configured”. Feito isso, pode-se selecionar a caixa “Run DHCP server immediatly” e clicar em “Concluir”.

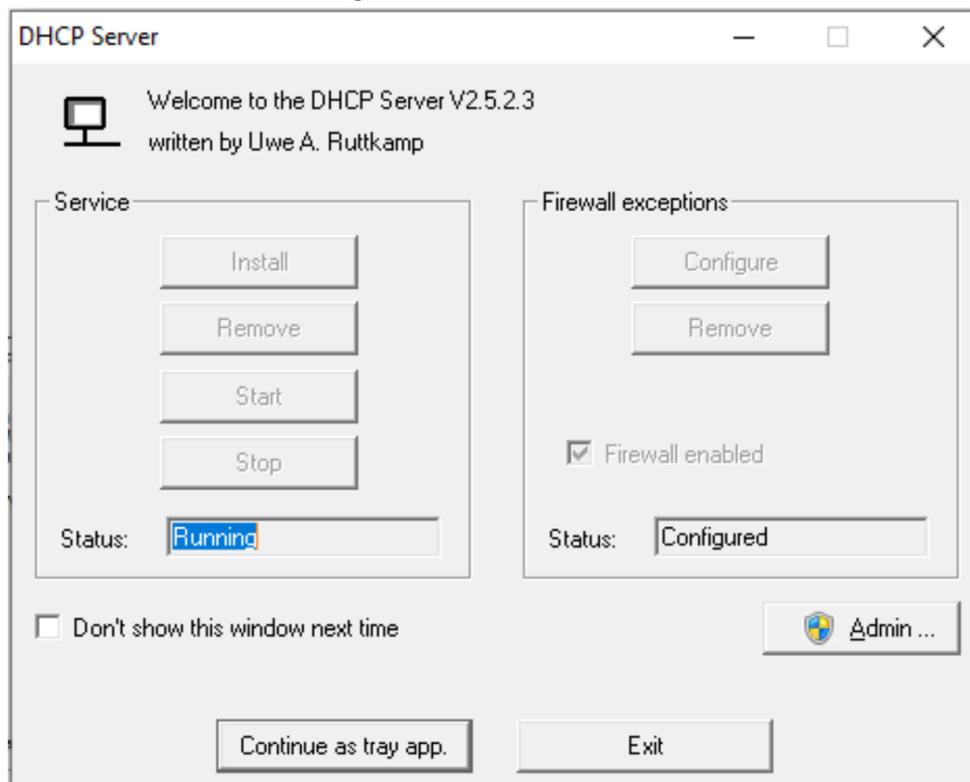
Após a conclusão, o servidor DHCP será aberto, como mostra a Figura 55, onde é necessário selecionar a opção “Continue as a tray app”. Feito isso, toda a configuração está concluída. O Windows receberá uma notificação informando qual endereço IP foi atribuído e esse será o endereço utilizado sempre que for necessário se conectar à Raspberry Pi 3.

**Figura 54 – Conclusão da configuração DHCP**



Fonte: Autoria Própria (2025).

**Figura 55 – Servidor DHCP**

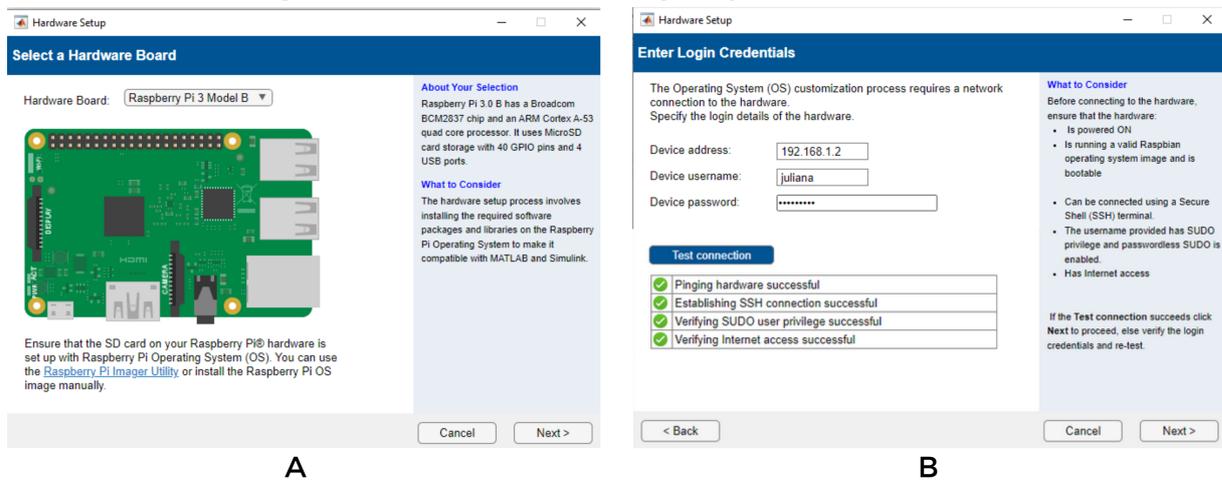


Fonte: Autoria Própria (2025).

## A.2 Configuração do pacote de suporte ao hardware

Após estabelecer a conexão entre a Raspberry Pi e o computador, a configuração inicial do dispositivo foi feita utilizando o *MATLAB Support Package for Raspberry Pi*. Com a instalação do pacote de suporte ao hardware concluída, o modelo da placa foi selecionado conforme mostrado na Figura 56A. Em seguida, a conexão entre a Raspberry Pi e o MATLAB® foi estabelecida utilizando as credenciais do dispositivo (endereço IP, nome de usuário e senha), conforme ilustrado na Figura 56B.

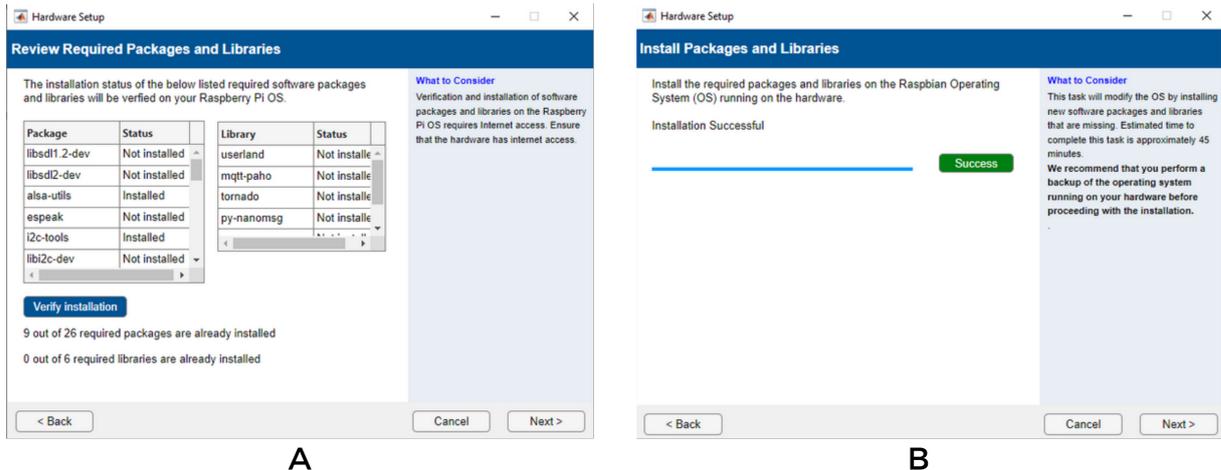
Figura 56 – Conexão entre a Raspberry Pi 3 e o MATLAB®



Fonte: Autoria Própria (2024).

Posteriormente, foi realizada a verificação dos pacotes e bibliotecas necessários para o correto funcionamento, conforme mostrado na Figura 57A, e a instalação dos mesmos, como indicado na Figura 57B. Após a instalação dos pacotes e bibliotecas faltantes, a Raspberry Pi poderia ser configurada para o desenvolvimento em *deep learning* e visão computacional, assim como para o uso de periféricos; entretanto, essas configurações não foram necessárias para os experimentos desenvolvidos neste trabalho. Com a conclusão de todas as configurações, foi necessário reiniciar a Raspberry Pi.

**Figura 57 – Instalação dos pacotes e bibliotecas necessários**



Fonte: Autoria Própria (2023).

### A.3 Configuração da placa de som USB externa

Após a instalação dos pacotes e bibliotecas necessários, é necessário realizar a conexão entre a Raspberry Pi e o MATLAB® todas as vezes que o MATLAB® for iniciado. Esse procedimento é realizado na janela de comando do MATLAB® através do comando `mypi = raspi('endereço IP', 'usuário', 'senha')`, onde o endereço IP, o usuário e a senha devem ser os mesmos utilizados na configuração do pacote de suporte ao hardware. Caso a conexão seja bem-sucedida, será exibido em tela as propriedades do dispositivo, como mostra a Figura 58.

**Figura 58 – Informações exibidas na janela de comando do MATLAB®**

```
Command Window
>> mypi = raspi('192.168.1.2','juliana','ldforever')

mypi =

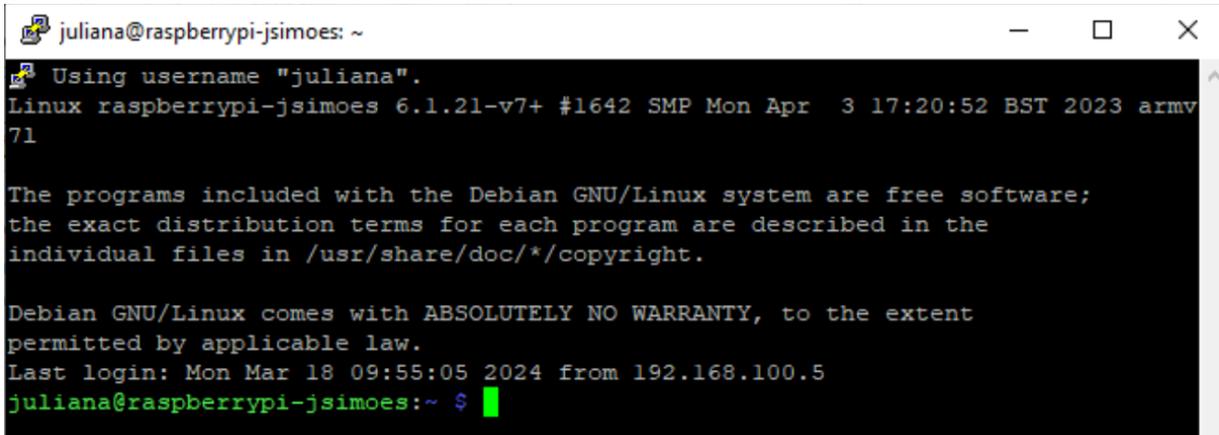
  raspi with properties:
    DeviceAddress: '192.168.1.2'
    Port: 18734
    BoardName: 'Raspberry Pi 3 Model B'
    AvailableLEDs: {'led0'}
    AvailableDigitalPins: [2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]
    AvailableSPIChannels: {}
    AvailableI2CBuses: {}
    AvailableWebcams: {}
    I2CBusSpeed:
    AvailableCANInterfaces: {}

  Supported peripherals
```

Fonte: Autoria própria (2024).

Para realizar a configuração da placa de som USB externa, o *Shell* do Linux, exibido na Figura 59 deve ser acessado, o que pode ser feito através do comando `myypi.openShell` na janela de comando do MATLAB®.

Figura 59 – *Shell* do Linux



```
juliana@raspberrypi-jsimoes: ~  
Using username "juliana".  
Linux raspberrypi-jsimoes 6.1.21-v7+ #1642 SMP Mon Apr 3 17:20:52 BST 2023 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Mar 18 09:55:05 2024 from 192.168.100.5  
juliana@raspberrypi-jsimoes:~ $
```

Fonte: Autoria própria (2024).

Com a placa de som conectada a uma das entradas USB da Raspberry Pi, é possível verificar sua presença como dispositivo de entrada e de saída através dos comandos `aplay -l`, que lista todos os dispositivos de reprodução de áudio, e `arecord -l`, que lista todos os dispositivos de captura de áudio.

A Figura 60 mostra todos os dispositivos conectados à Raspberry Pi. O primeiro dispositivo, denominado *bcm2835 Headphones*, trata-se da saída de áudio nativa da Raspberry Pi e possui por padrão o parâmetro *card 0*. Observa-se outros dois dispositivos de reprodução de áudio, sendo *card 1* a placa de som USB externa e *card 2* o conector HDMI. Caso mais de uma placa de som externa sejam conectadas, elas também serão listadas.

Figura 60 – Listagem dos dispositivos de reprodução e captura de áudio

```

julia@raspberrypi-jsimoes: ~
julia@raspberrypi-jsimoes:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: Headphones [bcm2835 Headphones], device 0: bcm2835 Headphones [bcm2835 Headphones]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 2: vc4hdmi [vc4-hdmi], device 0: MAI PCM i2s-hifi-0 [MAI PCM i2s-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
julia@raspberrypi-jsimoes:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
julia@raspberrypi-jsimoes:~ $ █

```

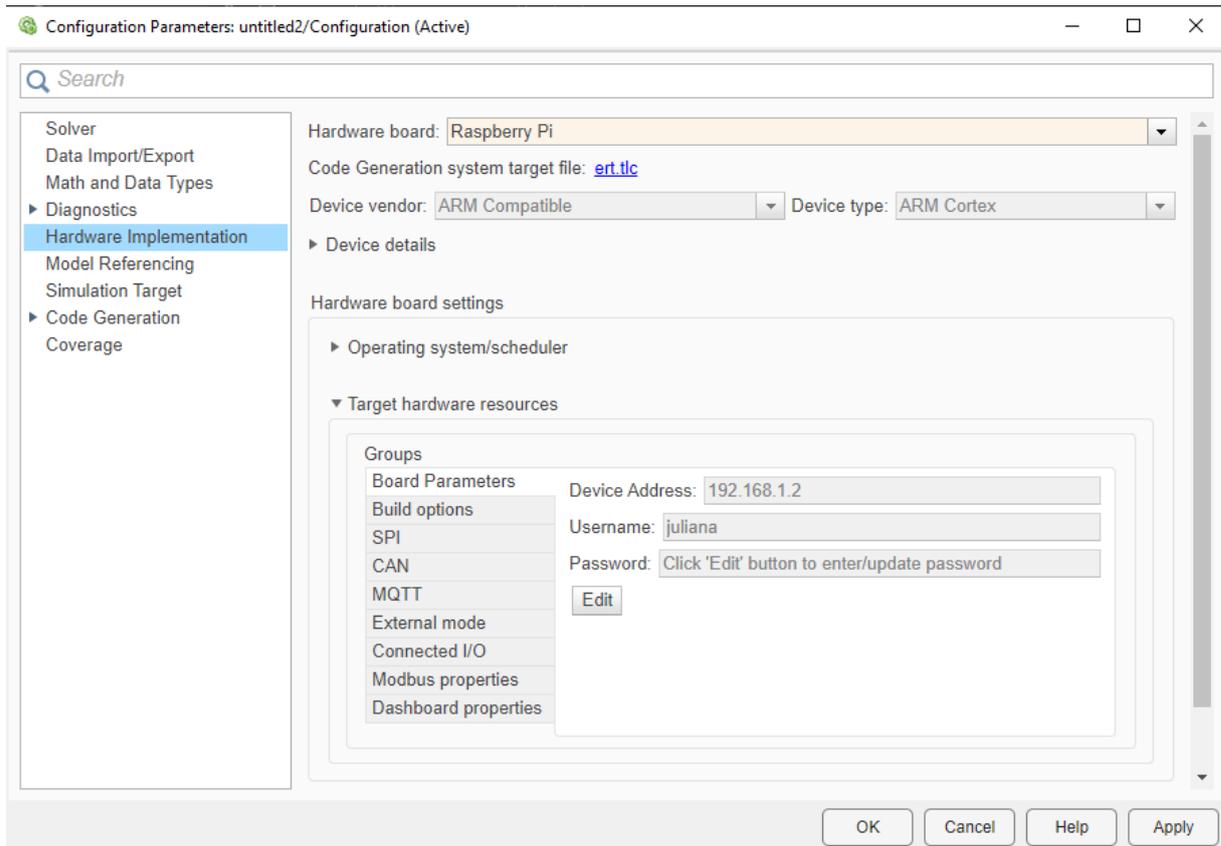
Fonte: Autoria própria (2024).

A ordem de prioridade é definida pela ordem crescente dos *cards*, sendo por padrão a saída de áudio própria da Raspberry Pi seu dispositivo primário. É possível alterar a ordem de prioridade dos dispositivos primários alterando o parâmetro do *card*, entretanto esse procedimento não se faz necessário. Durante os experimentos, é necessário respeitar a sintaxe dos blocos *ALSA Audio Playback* e *ALSA Audio Capture*, que solicita os parâmetros de *card* e *device* correspondentes à placa de som externa.

Para a correta utilização da placa de som externa, é necessário ajustar seus níveis de volume, que por padrão são definidos como zero. Esse ajuste pode ser feito através do comando *alsamixer* no *shell*, que exibirá a janela apresentada na Figura 61. A seleção do *card* é feita pressionando a tecla F6 e os níveis de volume de captura (*mic*) e reprodução (*speaker*) podem ser ajustados através das setas do teclado.



**Figura 62 – Configurações de implementação do modelo**

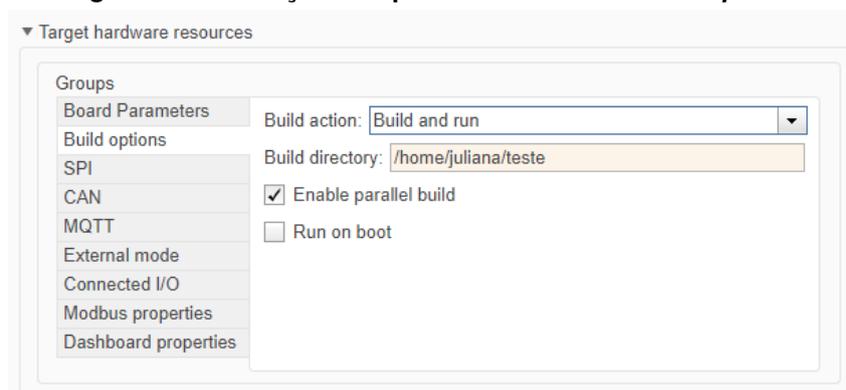


Fonte: Autoria própria (2024).

Em seguida, no campo *Target Resources*, devem ser definidas as configurações da placa. Em *Board Parameters*, as informações de endereço IP, usuário e senha da Raspberry Pi são preenchidas automaticamente, podendo ser alteradas caso necessário.

Por fim, em *Build options*, é definido o diretório onde cada arquivo executável será salvo. Também é possível modificar a ação realizada pelo botão de compilação, sendo possível defini-lo apenas como compilação (*build*) ou compilação e execução *build and run*. A Figura 63 mostra as configurações realizadas. Concluídas as configurações, elas podem ser aplicadas utilizando o botão *Apply*.

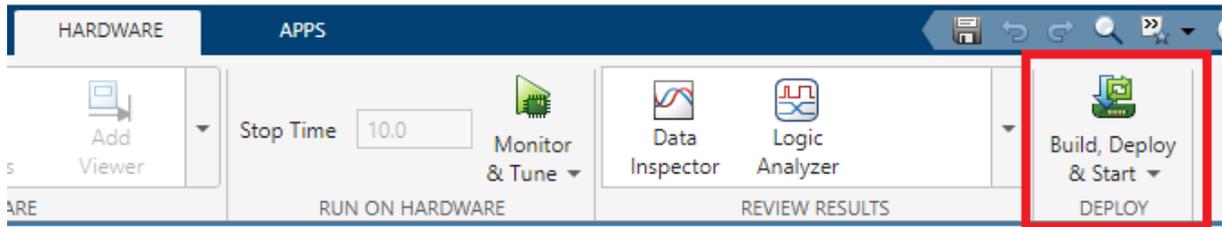
**Figura 63 – Definição dos parâmetros na aba *Build options***



Fonte: Autoria própria (2024).

Uma vez que o modelo foi configurado para a execução na Raspberry Pi, já é possível executá-lo. Para isso, na aba *Hardware*, basta selecionar o botão *Build, Deploy & Start*, como indicado pela Figura 64.

**Figura 64 – Botão utilizado para executar o modelo na Raspberry Pi**



**Fonte: Autoria própria (2024).**

É possível parar a execução de um modelo através da janela de comando do MATLAB®. Para isso, primeiramente cria-se um objeto através do comando `h = raspberrypi;` e, em sequência, utiliza-se o comando `h.stopModel('Nome do Modelo');`.

**APÊNDICE B – Configurações dos modelos Simulink® para os  
experimentos propostos**

## B.1 Gerador de funções

O bloco *Sine Wave* faz parte do *DSP System Toolbox* e gera em sua saída um sinal senoidal amostrado; os parâmetros utilizados para a configuração do bloco encontram-se na Tabela 3.

**Tabela 3 – Parâmetros de configuração do bloco *Sine Wave***

| Parâmetro                               | Configuração            |
|---|-------------------------|
| Amplitude                               | $2^{15} - 1$            |
| Frequência (Hz)                         | 1000                    |
| Deslocamento de fase (rad)              | 0                       |
| Modo de amostra                         | Discreto                |
| Complexidade de saída                   | Real                    |
| Método de computação                    | Função trigonométrica   |
| Tempo de amostra                        | 1/48000                 |
| Amostras por quadro                     | 1024                    |
| Redefinição de estados quando reativado | Reiniciar no tempo zero |

**Fonte: Autoria própria (2024).**

O primeiro parâmetro a ser configurado é a amplitude do sinal, cuja determinação é limitada pelo bloco *Alsa Audio Playback*, que possui representação no formato `int16`. Desse modo, o valor máximo permitido é  $2^{15} - 1$ , correspondente ao maior valor inteiro de 15 bits com um bit reservado para o sinal (Pasolini; Bazzi; Mirabella, 2016).

Os dois parâmetros subsequentes, frequência e fase, são definidos de acordo com as especificações desejadas. Neste caso, a frequência foi configurada em 1 kHz e o deslocamento de fase foi mantido em zero, podendo ser ajustado conforme o objetivo do modelo.

O modo de amostra é definido como discreto para gerar uma senoide discreta na saída, e a complexidade de saída é definida como real para que o sinal gerado seja senoidal, e não uma exponencial complexa. Adicionalmente, o método de computação foi mantido com a configuração padrão, baseada em funções trigonométricas.

Outro aspecto fundamental é o período de amostragem, que foi definido com base na frequência de amostragem da placa de som utilizada. Uma vez que a frequência máxima suportada pela placa é de 48000 amostras por segundo, o período de amostragem utilizado deve ser definido como 1/48000. Essa definição também assegura que a frequência do sinal gerado respeite a taxa de Nyquist, limitando-se a um máximo de 24 kHz. Além disso, a configuração das amostras por quadro, definida como 1024, estabelece a quantidade de dados transmitidos simultaneamente para o próximo bloco. Tal escolha foi baseada no fato de que a Raspberry Pi 3, assim como outros sistemas embarcados, opera com quadros de dados em vez de dados unitários (Pasolini; Bazzi; Mirabella, 2016).

Quanto ao último parâmetro do bloco *Sine Wave*, a configuração padrão foi mantida para que a geração da onda reinicie no tempo zero quando o bloco for reativado, preservando a coerência do sinal gerado.

O segundo bloco apresentado no modelo, *Data Type Conversion*, realiza a conversão do tipo de dado de entrada em qualquer outro tipo de dado suportado pelo Simulink®. Nesse caso, como o bloco *ALSA Audio Playback* exige dados de entrada no formato `int16` e a saída do bloco *Sine Wave* é do tipo `double`, essa conversão se torna essencial. A Tabela 4 mostra os parâmetros utilizados para a configuração do bloco *Data Type Conversion*, como a saída mínima e máxima e o tipo de dado de saída. Todos os parâmetros foram mantidos com sua configuração padrão, com exceção do tipo de dado de saída, configurado para o tipo de dado desejado, nesse caso `int16`.

**Tabela 4 – Parâmetros de configuração do bloco *Data Type Conversion***

| Parâmetro  | Configuração        |
|--|---------------------|
| Saída mínima                                       | []                  |
| Saída máxima                                       | []                  |
| Tipo de dado de saída                              | int16               |
| Bloquear a configuração do tipo de dado de saída   | Não                 |
| Entradas e saídas iguais a                         | Valor do mundo real |
| Método de arredondamento de inteiro                | Floor               |
| Saturar em caso caso de <i>overflow</i> de inteiro | Não                 |

**Fonte: Autoria própria (2024).**

Em sequência, o bloco *Matrix Concatenate* realiza a concatenação de múltiplos sinais de entrada em um único sinal de saída, que pode ser um vetor ou um arranjo multidimensional. Uma vez que o bloco *ALSA Audio Playback* exige que o sinal de entrada seja estéreo, é necessário transformar o sinal de saída do bloco *Data Type Conversion*, que possui apenas um canal, em um sinal de dois canais. Assim, utilizam-se duas entradas para o bloco *Matrix Concatenate*, cada uma recebendo o mesmo sinal mono, que serão concatenadas de modo a gerar um sinal estéreo na saída. A Tabela 5 mostra os parâmetros utilizados para a configuração do bloco, onde são definidos o número de entradas, o modo de concatenação e a dimensão da concatenação.

**Tabela 5 – Parâmetros de configuração do bloco *Matrix Concatenate***

| Parâmetro                | Configuração             |
|--------------------------|--------------------------|
| Número de entradas       | 2                        |
| Modo                     | Arranjo multidimensional |
| Dimensão da concatenação | 2                        |

**Fonte: Autoria própria (2024).**

Por fim, deve-se configurar o bloco *ALSA Audio Playback*, que representa a saída analógica da Raspberry Pi 3 e está relacionado ao *driver* de som ALSA. O bloco possui apenas dois parâmetros para configuração: nome do dispositivo e frequência de amostragem de áudio. O nome do dispositivo deve seguir a sintaxe “`plughw:card,device`”, onde “*card*” e “*device*” são dados pela configuração da placa de som externa apresentada no apêndice A. A frequência de amostragem é definida pelo valor máximo da placa de som externa. A Tabela 6 mostra os parâmetros utilizados para configuração, onde são inseridos o nome do dispositivo e a frequência de amostragem.

**Tabela 6 – Parâmetros de configuração do bloco *ALSA Audio Playback***

| Parâmetro                              | Configuração |
|--|--------------|
| Nome do dispositivo                    | 'plughw:0,0' |
| Frequência de amostragem de áudio (Hz) | 48000        |

**Fonte: A autoria própria (2024).**

Para alterar o tipo de sinal gerado, é necessário substituir o bloco *Sine Wave* por outro que gere a saída desejada. Aqui, ele é substituído pelo bloco *Pulse Generator* de modo a gerar um sinal de onda quadrada. A Tabela 7 mostra as configurações do bloco *Pulse Generator*, onde o primeiro parâmetro a ser configurado, tipo de pulso, pode ser definido como baseado em amostra ou baseado em tempo, sendo que na segunda opção, utilizada neste caso, o bloco calcula a saída em intervalos fixos especificados. Em sequência, o segundo parâmetro, tempo, especifica se será utilizado o tempo de simulação ou um sinal externo. Já no caso da amplitude do sinal, a configuração é feita da mesma forma que a apresentada para o bloco *Sine Wave*, sendo mantida como  $2^{15} - 1$ .

**Tabela 7 – Parâmetros de configuração do bloco *Pulse Generator***

| Parâmetro                             | Configuração              |
|---------------------------------------|---------------------------|
| Tipo de pulso                         | Baseado em amostra        |
| Tempo (t)                             | Usar o tempo de simulação |
| Amplitude                             | $2^{15} - 1$              |
| Período (número de amostras)          | 48                        |
| Largura de pulso (número de amostras) | 24                        |
| Desvio de fase (número de amostras)   | 0                         |
| Período de amostragem                 | 1/48000                   |

**Fonte: A autoria própria (2024).**

O período do sinal gerado é definido em números de amostras e deve ser configurado com base no período de amostragem. Uma vez que o período de amostragem é de 1/48000 Hz, o período de 48 amostras gera um sinal com período de 1 ms. Por fim, a largura de pulso, também dada em número de amostras, é o que define o *duty cycle* do sinal. Neste caso, para obter um *duty cycle* de 50%, definiu-se a largura de pulso como 24 amostras, metade do valor do período.

Como o bloco *Pulse Generator* não possui a opção de definir o número de amostra por quadros, esse processo é realizado de forma externa através do bloco *Buffer*. As configurações apresentadas na Tabela 8 são utilizadas para agrupar o sinal em quadros de 1024 amostras.

**Tabela 8 – Parâmetros de configuração do bloco *Buffer***

| Parâmetro                              | Configuração |
|--|--------------|
| Tamanho da saída do buffer (por canal) | 1024         |
| Sobreposição de buffer                 | 0            |
| Condições iniciais                     | 0            |

**Fonte: A autoria própria (2024).**

## B.2 Transmissor e receptor AM

Para a geração dos sinais de mensagem e de portadora, foram utilizados dois blocos *Sine Wave*. As configurações dos blocos seguem os mesmos parâmetros apresentados na Tabela 3, diferindo apenas na amplitude e na frequência. Aqui, tanto para o sinal da mensagem quanto da portadora, foi utilizada uma amplitude unitária. As frequências utilizadas foram  $f_m = 500$  Hz para a mensagem e  $f_c = 10$  kHz para a portadora.

A modulação foi realizada através da multiplicação entre o sinal da mensagem e o sinal da portadora através do bloco *Product*, cujo único parâmetro a ser configurado é a quantidade de entradas, definido como duas. Em sequência, o sinal modulado é aplicado a um bloco de ganho, onde  $K = 2^{15} - 1$ , responsável por ajustar o sinal para a amplitude máxima de 15 bits.

Por fim, os blocos *Data Type Conversion*, *Matrix Concatenate* e *ALSA Audio Playback* possuem as mesmas funções e seguem as mesmas configurações das apresentadas anteriormente.

Para o processo de recepção do sinal, o sinal entra pela placa de som externa, representada no modelo Simulink® pelo bloco *ALSA Audio Capture*, que segue as mesmas configurações apresentadas na Tabela 6.

Posteriormente, o sinal do tipo estéreo é convertido para mono através do bloco *Multiport Selector*, cujas configurações encontram-se na Tabela 9.

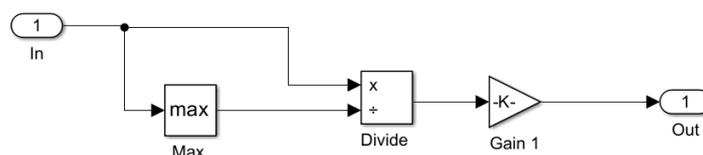
**Tabela 9 – Parâmetros de configuração do bloco *Multiport Selector***

| Parâmetro         | Configuração |
|-------------------|--------------|
| Select            | Columns      |
| Indices to output | 2            |
| Invalid Index     | Clip Index   |

**Fonte: Autoria própria (2024).**

O bloco AGC foi desenvolvido no Simulink® para ajustar dinamicamente a amplitude do sinal. A Figura 65 mostra o modelo que realiza o controle automático de ganho, onde o sinal tem sua amplitude máxima calculada através do bloco *Max*, e é normalizado utilizando o bloco *Divide*, onde o valor do sinal é dividido pelo valor da amplitude máxima. Em seguida, aplica-se um ganho para definir a amplitude do sinal de saída de acordo com o valor desejado, aqui definido como  $K = 2^{15} - 1$ . Esse processo evita distorções do sinal que ocorrem em função de flutuações na amplitude.

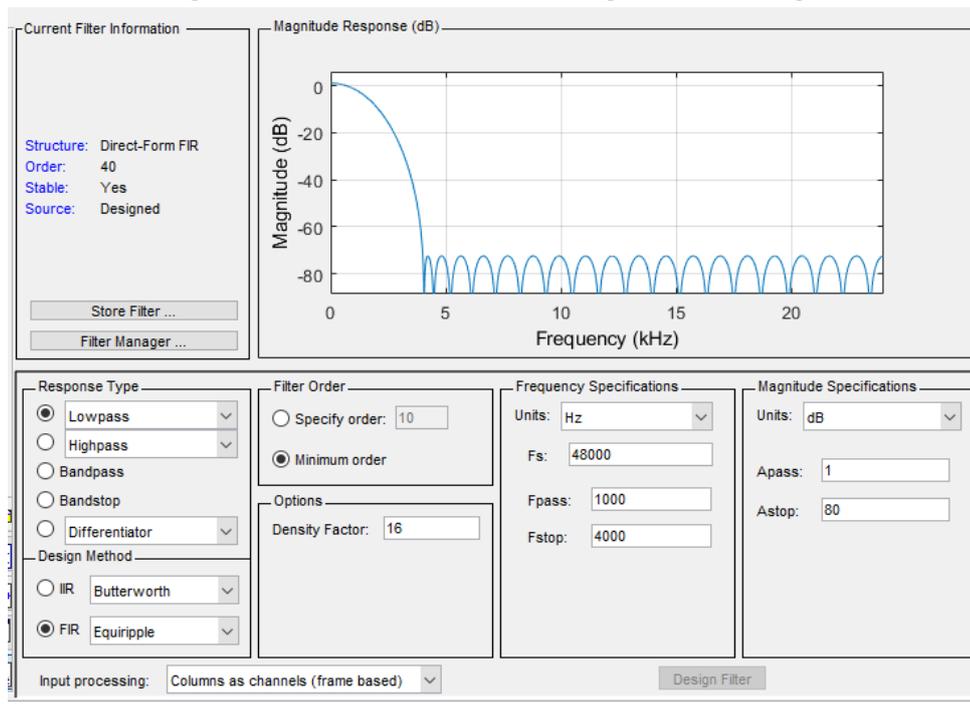
**Figura 65 – Modelo utilizado para controle automático de ganho**



**Fonte: Autoria própria (2024).**

Após o ajuste de ganho, o sinal é aplicado a um filtro digital projetado através do bloco *Digital Filter Design*, que atua como um filtro passa-baixas, onde a componente de alta frequência da portadora é removida e o sinal original é preservado. Conforme mostrado na Figura 66, o filtro passa-baixa foi projetado com uma taxa de amostragem de 48 kHz, uma frequência de passagem de 1 kHz, e uma frequência de rejeição de 4 kHz. Utilizou-se o método Equiripple para garantir uma resposta de magnitude uniforme na banda de passagem e uma atenuação de 80 dB na banda de rejeição.

**Figura 66 – Parâmetros do bloco *Digital Filter Design***



Fonte: Autoria própria (2024).

Após a filtragem, o sinal é novamente convertido para o formato `int16` e concatenado pelo bloco *Matrix Concatenate*, como apresentado para os experimentos anteriores. O sinal recuperado é então transmitido por uma segunda saída de áudio, também representada por um bloco *ALSA Audio Playback*, o que permite a visualização do sinal demodulado em um osciloscópio conectado.

### B.3 Modulador FM

Para a modulação FM, o sinal de mensagem foi gerado a partir do bloco *Sine Wave*, que faz parte do *DSP System Toolbox*, que segue as configurações apresentadas na Tabela 10.

Para a modulação FM utilizou-se o bloco *FM Modulator Baseband*, parte do *Communications Toolbox* e configurado para realizar a modulação em frequência em banda base do sinal de entrada. Como mostra a Figura 67, a configuração desse bloco inclui dois parâmetros: desvio de frequência, definido em Hz, e modo de simulação.

**Tabela 10 – Parâmetros de configuração do bloco *Sine Wave* para modulação FM**

| Parâmetro                               | Configuração            |
|---|-------------------------|
| Amplitude                               | 1                       |
| Frequência (Hz)                         | 100                     |
| Deslocamento de fase (rad)              | 0                       |
| Modo de amostra                         | Discreto                |
| Complexidade de saída                   | Real                    |
| Método de computação                    | Função trigonométrica   |
| Tempo de amostra                        | 1/24000                 |
| Amostras por quadro                     | 1                       |
| Redefinição de estados quando reativado | Reiniciar no tempo zero |

**Fonte: Autoria própria (2025).**

**Figura 67 – Parâmetros do bloco *FM Modulator Baseband***

**Fonte: Autoria própria (2024).**

O desvio de frequência determina o máximo desvio de frequência a ser aplicado ao sinal da portadora e foi configurado com um valor de 5000 Hz, de modo que a frequência do sinal da portadora pode variar até  $\pm 5000$  Hz em relação à sua frequência central, a depender da amplitude do sinal modulante. O bloco foi configurado para operar no modo “geração de código”, ideal para implementações que envolvem geração de código para hardware.

O bloco *Complex to Real-Imag* permite a separação das componentes de sinais complexos em real e imaginária, sendo possível obter na saída ambas as componentes ou apenas uma delas. O único parâmetro de configuração desse bloco é o tipo de saída, que pode ser definido como real, imaginária ou real e imaginária. O bloco *FM Modulator Baseband* gera na saída um sinal complexo, entretanto, a única componente necessária para a transmissão é a real, de modo que a saída do bloco *Complex to Real-Imag* é definida como real.

Os blocos *Gain*, *Buffer*, *Data Type Conversion*, *Matrix Concatenate* e *ALSA Audio Playback* apresentam as mesmas configurações utilizadas nos experimentos anteriores.

#### **B.4 Modulador 2-PAM e 4-PAM**

Nas duas abordagens de modulação PAM, o sinal a ser modulado consiste em uma sequência aleatória de bits gerada pelo bloco *Bernoulli Binary Generator*, cujas configurações encontram-se na Tabela 11.

**Tabela 11 – Parâmetros de configuração do bloco *Bernoulli Binary Generator***

| Parâmetro                | Configuração |
|--------------------------|--------------|
| Probabilidade de zero    | 0.5          |
| Fonte da semente inicial | Parâmetro    |
| Semente inicial          | 25           |
| Tempo de amostragem      | 1/2400       |
| Amostras por quadro      | 1000         |
| Tipo de dado de saída    | Double       |

**Fonte: Autoria própria.**

O primeiro parâmetro configurado foi a probabilidade de zero, definida como 0,5, ou seja, 50%. Em sequência, o parâmetro “semente inicial” recebe o valor inicial da sequência binária, a partir do qual são gerados números aleatórios. O tempo de amostragem corresponde a uma taxa de bits de  $R_b = 2400$  bits/s, agrupados em quadros de 1000 amostras.

A sequência de bits é encaminhada para o bloco *M-PAM Modulator Baseband*, cujas configurações encontram-se na Tabela 12. O número M-ário determina a quantidade de símbolos em que os bits são mapeados, sendo o único parâmetro a ser modificado para alterar o tipo de modulação. Por exemplo, para a modulação 2-PAM, o número M-ário é definido como 2, enquanto para a modulação 4-PAM ele é definido como 4.

**Tabela 12 – Parâmetros de configuração do bloco *M-PAM Modulator Baseband* para a modulação 2-PAM**

| Parâmetro                | Configuração     |
|--------------------------|------------------|
| Número M-ário            | 2                |
| Tipo de entrada          | Bit              |
| Ordem da constelação     | Gray             |
| Método de normalização   | Potência de pico |
| Potência de pico (watts) | 1                |

**Fonte: Autoria própria (2024).**

O sinal modulado passa, então, pelo bloco *Complex to Real-Imag*, que extrai a parte real do sinal complexo gerado pelo bloco *M-PAM Modulator Baseband*. Como apresentado no diagrama de constelação, a modulação, tanto a modulação 2-PAM quanto 4-PAM, os símbolos encontram-se dispostos sobre o eixo real, não possuindo uma componente imaginária. Como saída, o bloco *Complex to Real-Imag* apresenta uma sequência de números reais, ou símbolos; uma vez que a taxa de bits definida pelo bloco *Bernoulli Binary Generator* é  $R_b = 2400$  bits/s, a taxa de símbolos será  $R_s = 2400$  símbolos/s.

Após a conversão, o sinal é submetido a uma interpolação utilizando um filtro FIR, que ajusta a taxa de amostragem e suaviza o espectro do sinal. A Tabela 13 mostra as configurações utilizadas para implementar um filtro NRZ, em que o sinal não retorna a zero. Para isso, os parâmetros foram digitados manualmente. O fator de interpolação foi definido como 20, de modo que o sinal de entrada será *sobreamostrado* em 20 vezes, ou seja, para cada símbolo de entrada é atribuído um pulso de transmissão constituído por 20 amostras. Dessa forma, o sinal de saída possui taxa de amostragem de 48000 amostras/s.

**Tabela 13 – Parâmetros de configuração do bloco *FIR Interpolator***

| Parâmetro                   | Configuração                             |
|-----------------------------|--|
| Fonte dos coeficientes      | Parâmetros digitados (caixa de diálogo)  |
| Coefficientes do filtro FIR | ones(1,20)                               |
| Fator de interpolação       | 20                                       |
| Processamento da entrada    | Colunas como canais (baseado em quadros) |
| Opções de taxa              | Aplicar processamento de taxa única      |

**Fonte: Autoria própria (2024).**

A última etapa do processo de modulação envolve o ajuste de ganho, realizado por um subsistema AGC idêntico ao apresentado na Figura 65, onde o ganho é ajustado de acordo com o valor máximo do sinal de entrada. Isso garante que o sinal tenha uma amplitude uniforme antes de ser reproduzido. Após a modulação, o sinal passa pelos blocos *Data Type Conversion*, *Matrix Concatenate* e *ALSA Audio Playback*, cujas configurações são as mesmas apresentadas anteriormente.

## B.5 Modulador BPSK e QPSK

Na modulação PSK, o diagrama foi construído de maneira semelhante àquela apresentada na modulação PAM, com a sequência de bits gerada pelo bloco *Bernoulli Binary Generator* seguindo os mesmos parâmetros definidos na Tabela 11. A principal diferença para a modulação PAM foi a utilização do bloco *M-PSK Modulator Baseband*, cujos parâmetros encontram-se descritos na Tabela 14.

**Tabela 14 – Parâmetros de configuração do bloco *M-PSK Modulator Baseband* para modulação BPSK e QPSK**

| Parâmetro            | Configuração (BPSK) | Configuração (QPSK) |
|----------------------|---------------------|---------------------|
| Número M-ário        | 2                   | 4                   |
| Tipo de entrada      | Bit                 | Bit                 |
| Ordem da constelação | Gray                | Gray                |
| Desvio de fase       | Pi                  | pi/4                |

**Fonte: Autoria própria (2024).**

Após a modulação, o sinal complexo passa pelo bloco *Complex to Real-Imag*, utilizado para separar as componentes real e imaginária. No caso da modulação BPSK, o bloco é configurado para ter como saída um sinal real, enquanto na modulação QPSK o bloco é configurado para ter como saída dois sinais, um real e outro imaginário.

O bloco *FIR Interpolation*, utilizado tanto na modulação BPSK quanto QPSK, segue o mesmo objetivo e configurações apresentados no modelo de modulação PAM. No caso da modulação QPSK, após a separação do sinal em duas componentes, cada uma delas passa por uma interpolação.

Após a interpolação, o sinal modulado é convertido para banda passante através da multiplicação por um sinal senoidal. No caso da modulação BPSK, as configurações são as

mesmas apresentadas na Tabela 3, enquanto na modulação QPSK uma das componentes é multiplicada por um seno e a outra por um cosseno, conforme as configurações apresentadas na Tabela 15.

**Tabela 15 – Parâmetros de configuração do bloco *Sine Wave* para portadoras em quadratura**

| <b>Parâmetro</b>                        | <b>Configuração (Seno)</b> | <b>Configuração (Cosseno)</b> |
|---|----------------------------|-------------------------------|
| Amplitude                               | 1                          | 1                             |
| Frequência (Hz)                         | 5000                       | 5000                          |
| Deslocamento de fase (rad)              | pi                         | pi/2                          |
| Modo de amostra                         | Discreto                   | Discreto                      |
| Complexidade de saída                   | Real                       | Real                          |
| Método de computação                    | Função trigonométrica      | Função trigonométrica         |
| Tempo de amostra                        | 1/48000                    | 1/48000                       |
| Amostras por quadro                     | 1024                       | 1024                          |
| Redefinição de estados quando reativado | Reiniciar no tempo zero    | Reiniciar no tempo zero       |

**Fonte: Autoria própria (2024).**

As demais etapas do sistema seguem as mesmas configurações utilizadas nos experimentos anteriores.