



DISSERTAÇÃO DE MESTRADO

**A UTILIZAÇÃO DE RÁDIOS DEFINIDOS POR SOFTWARE,  
TÉCNICAS DE APRENDIZADO DE MÁQUINA E FERRAMENTAS  
DE ACESSO REMOTO NO APRIMORAMENTO DOS LABORATÓRIOS  
DE TELECOMUNICAÇÕES**

**THIAGO LUIZ PINHEIRO DE ALMEIDA**

**Brasília, Novembro de 2020**

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Departamento de Engenharia Elétrica

DISSERTAÇÃO DE MESTRADO

**A UTILIZAÇÃO DE RÁDIOS DEFINIDOS POR SOFTWARE,  
TÉCNICAS DE APRENDIZADO DE MÁQUINA E FERRAMENTAS  
DE ACESSO REMOTO NO APRIMORAMENTO DOS LABORATÓRIOS  
DE TELECOMUNICAÇÕES**

**THIAGO LUIZ PINHEIRO DE ALMEIDA**

*Dissertação de Mestrado submetido ao Departamento de Engenharia  
Elétrica como requisito parcial para obtenção  
do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof. Edson Mintsu Hung, D.Sc., FT/UnB  
*Orientador*

\_\_\_\_\_

Prof. Daniel Guerreiro e Silva, D.Sc., FT/UnB  
*Examinador Interno*

\_\_\_\_\_

Prof. Leonardo R. A. X. de Menezes, D. Sc.,  
FT/UnB  
*Examinador Interno*

\_\_\_\_\_

## FICHA CATALOGRÁFICA

ALMEIDA, T.L.P

A UTILIZAÇÃO DE RÁDIOS DEFINIDOS POR SOFTWARE, TÉCNICAS DE APRENDIZADO DE MÁQUINA E FERRAMENTAS DE ACESSO REMOTO NO APRIMORAMENTO DOS LABORATÓRIOS DE TELECOMUNICAÇÕES [Distrito Federal] 2020.

xvi, 56 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2020).

Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. GNU Radio

2. USRP

3. Machine Learning

4. EAD

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

ALMEIDA, THIAGO (2020). *A UTILIZAÇÃO DE RÁDIOS DEFINIDOS POR SOFTWARE, TÉCNICAS DE APRENDIZADO DE MÁQUINA E FERRAMENTAS DE ACESSO REMOTO NO APRIMORAMENTO DOS LABORATÓRIOS DE TELECOMUNICAÇÕES*. Dissertação de Mestrado, Publicação PPGEE n° 751/20, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 56 p.

## CESSÃO DE DIREITOS

AUTOR: THIAGO LUIZ PINHEIRO DE ALMEIDA

TÍTULO: A UTILIZAÇÃO DE RÁDIOS DEFINIDOS POR SOFTWARE, TÉCNICAS DE APRENDIZADO DE MÁQUINA E FERRAMENTAS DE ACESSO REMOTO NO APRIMORAMENTO DOS LABORATÓRIOS DE TELECOMUNICAÇÕES.

GRAU: Mestre em Engenharia Elétrica ANO: 2020

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa Dissertação de Mestrado pode ser reproduzida sem autorização por escrito do autor.

---

THIAGO LUIZ PINHEIRO DE ALMEIDA

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedico este trabalho à minha querida família que se sacrificou para que eu pudesse concluir essa tarefa. Angélica e Nicole, obrigado pelo amor de vocês!

---

## RESUMO

O Departamento de Defesa Norte Americano iniciou pesquisas para o desenvolvimento de um rádio que pudesse ser facilmente configurável e que permitisse a interoperabilidade de troca de informações entre as Forças Armadas em um cenário de combate. Essa necessidade impulsionou o desenvolvimento dos chamados *Software Defined Radios* (SDR), os quais podem ser atualmente empregados em diversas finalidades, além dos propósitos militares inicialmente visualizados. Tais rádios se mostram como uma alternativa economicamente mais barata para equipar os laboratórios didáticos das Universidades e para possibilitar pesquisas por novos métodos de comunicação, seja em experimentos presenciais ou em formato EAD. Este trabalho demonstra a aplicabilidade real de equipamentos baseados em SDR nos laboratórios universitários como suporte à ementa básica de cursos de engenharia, sendo uma ferramenta importante principalmente na modalidade EAD. Os resultados confirmam a possibilidade do emprego desses rádios como infraestrutura de pesquisa de novas técnicas de comunicação, como a aplicação de mecanismos de aprendizado de máquina ao contexto de modulação/demodulação em comunicação digital, apresentando resultados com desempenhos comparáveis ou, em algumas situações, até mesmo superiores aos métodos de modulação/demodulação tradicionais.

---

## ABSTRACT

The North American Department of Defense initiated research to develop a radio that could be easily configurable and that would allow interoperability of information exchange between the Armed Forces in a combat scenario. This need drove the development of the so-called Software Defined Radios (SDR), which can currently be used for several purposes, in addition to the military purposes initially envisaged. Such radios are shown as an economically cheaper alternative to equip the teaching laboratories of the Universities and to enable research by new communication methods, whether in face-to-face experiments or in distance learning format. This work demonstrates the real applicability of SDR-based equipment in university laboratories as a support for the basic menu of engineering courses, being an important tool mainly in distance learning. The results confirm the possibility of using these radios as a research infrastructure for new communication techniques, such as the application of machine learning mechanisms to the context of modulation / demodulation in digital communication, presenting results with comparable performances or, in some situations, even even superior to traditional modulation/demodulation methods.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	DEMANDAS POR UM NOVO MODELO DE LABORATÓRIO	1
1.2	UMA NOVA TECNOLOGIA	3
1.3	PROPOSTA DE UTILIZAÇÃO DE SDR NOS LABORATÓRIOS	4
1.4	OBJETIVO DA DISSERTAÇÃO	5
1.5	ESTRUTURA DO TRABALHO	5
<b>2</b>	<b>AMBIENTE EXPERIMENTAL</b>	<b>6</b>
2.1	METODOLOGIA	6
2.2	ARQUITETURA DO GNU RADIO COMPANION	7
2.3	ARQUITETURA DO UNIVERSAL SOFTWARE RADIO PERIPHERAL	8
2.3.1	ANACONDA E JUPYTER	10
2.3.2	SCIKIT-LEARN	10
2.3.3	TEAMVIEWER E CONJUNTO EXPERIMENTAL	11
<b>3</b>	<b>EXPERIMENTOS COM GNU RADIO E USRP</b>	<b>12</b>
3.1	EXPERIMENTOS DEMONSTRATIVOS	12
3.2	FILTRAGEM DE SINAIS	14
3.3	MODULAÇÃO EM AMPLITUDE - <i>Amplitude Modulation (AM)</i>	15
3.4	MODULAÇÃO EM FREQUÊNCIA - <i>Frequency Modulation (FM)</i>	18
3.5	PHASE LOOKED LOOP (PLL)	23
3.5.1	ESTRUTURA GERAL DO PLL	23
3.5.2	CALIBRAÇÃO DO PLL	26
3.6	TRANSMISSÃO DIGITAL EM BANDA BASE	29
3.6.1	DETECÇÃO ÓTIMA - FILTRO CASADO	31
3.6.2	CONTROLE ESPECTRAL - FORMATAÇÃO DE PULSO	32
<b>4</b>	<b>DEMODULADOR NEURAL</b>	<b>38</b>
4.1	REDES NEURAS EM TELECOMUNICAÇÕES	38
4.2	IDENTIFICAÇÃO DO CONTEXTO GERAL	39
4.2.1	FUNCIONAMENTO DE UMA REDE NEURAL	39
4.2.2	REDES NEURAS E A DEMODULAÇÃO DE SINAIS	43
4.3	OBTENÇÃO DOS DADOS	44
4.3.1	BLOCO EXTERNO - CLASSIFICADOR DE SÍMBOLOS TRADICIONAL	44
4.3.2	CRIAÇÃO DO <i>Dataset</i>	45
4.4	PREPARAÇÃO DOS DADOS	47
4.5	AVALIAÇÃO PRELIMINAR DOS DADOS	47

4.6	SELEÇÃO E TREINAMENTO DE MODELOS .....	47
4.7	APRESENTAÇÃO DA SOLUÇÃO .....	48
<b>5</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>52</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>55</b>
	<b>APÊNDICES.....</b>	<b>57</b>

# LISTA DE FIGURAS

1.1	Declaração da pandemia de COVID-19.....	1
1.2	Restrição de aulas de laboratório .....	1
1.3	Quantidade de cursos oferecidos em EAD no Brasil em 2017 e 2018 (Censo Digital EAD 2018) .....	2
1.4	Laboratório eletrônica - Lab NEI ( <a href="http://fga.unb.br/eletronica/infraestrutura">http://fga.unb.br/eletronica/infraestrutura</a> , acesso em 05 nov 2020).....	2
2.1	Arquitetura de funcionamento do GNU Radio .....	8
2.2	Vista superior da placa USRP modelo N210 .....	9
2.3	Placas LFTX (transmissão) e LFRX (recepção) .....	10
2.4	Bancada de testes para validação dos experimentos .....	11
3.1	Filtragem de onda quadrada .....	14
3.2	Principais tipos de filtragem .....	15
3.3	Modulação em frequência no tempo e no espectro (1) .....	16
3.4	Modulador AM no GNU Radio .....	17
3.5	Demodulador AM no GNU Radio .....	17
3.6	Resultado da modulação AM no tempo e na frequência .....	18
3.7	Diagrama esquemático da placa USRP e GRC.....	19
3.8	Espectro de frequência de um sinal senoidal modulado em frequência (1) .....	21
3.9	Modulador FM direto.....	22
3.10	Demodulador FM direto .....	22
3.11	Espectro de tom senoidal representado por função de Bessel .....	23
3.12	Diagrama de Blocos básicos de um PLL (2).....	23
3.13	PLL em blocos GNU Radio.....	27
3.14	Experimento PLL.....	28
3.15	Gráfico de erro sinal medido no osciloscópio digital (sink).....	29
3.16	Diagrama comunicação PAM (3).....	30
3.17	Pulso $p[nT_s - i(LT_s)] * p(-nT_s)$ Cosseno Elevado.....	34
3.18	Diagrama formatação de pulso .....	35
3.19	Modulação por Amplitude de Pulso com filtro RRC .....	36
3.20	Texto corrompido por ISI.....	36
4.1	Modelo gráfico do Perceptron (4) .....	40
4.2	Encadeamento de Perceptrons (4) .....	40
4.3	Agrupamento de vários neurônios em camadas (4) .....	42
4.4	Estrutura de arquivos dos blocos do GNU Radio (5) .....	44
4.5	Limiar de classificação de símbolos (1).....	45



4.6	Diagrama de ambiente de comunicação para treinamento do classificador baseado em rede neural (4) .....	46
4.7	Constelação sob baixa interferência versus Constelação sob elevada interferência ..	48
4.8	Matriz de confusão para diferentes estruturas de rede e funções de ativação .....	49
4.9	Constelação de símbolos após ruídos, interferências e reflexões .....	50
4.10	Matriz de confusão comparativa entre o demodulador tradicional e o demodulador baseado em rede neural .....	51

## LISTA DE TABELAS

3.1	Consumo de recursos no computador remoto e hospedeiro .....	13
4.1	Mapa de símbolos e amostras .....	46
4.2	Comparação de acurácias para as configurações da rede .....	50

# 1 INTRODUÇÃO

## 1.1 DEMANDAS POR UM NOVO MODELO DE LABORATÓRIO

Recentemente, o mundo foi acometido pela pandemia de COVID-19, conforme registrado na Fig. 1.1, que requereu medidas como quarentenas e isolamento social para o controle da proliferação do vírus. Tal situação gerou um enorme desafio para as Universidades, principalmente para a execução das aulas laboratoriais, que tiveram que ser flexibilizadas, conforme observado na Fig. 1.2.

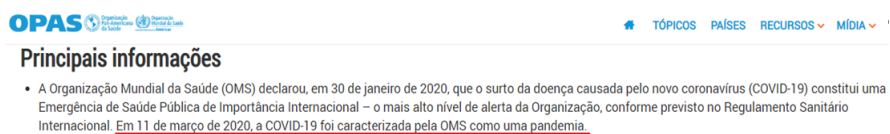


Figura 1.1: Declaração da pandemia de COVID-19

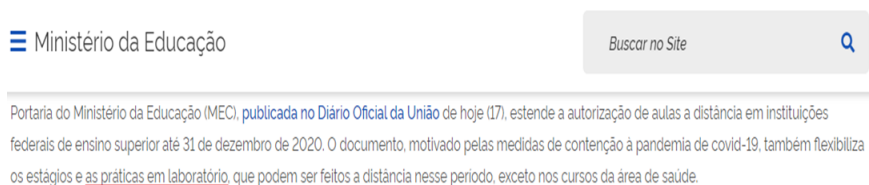


Figura 1.2: Restrição de aulas de laboratório

Os laboratórios universitários tradicionais, por apresentarem características de ensino fortemente presenciais, precisavam de uma alternativa para superar as restrições de locomoção e isolamento social. Os equipamentos rádios utilizados nos laboratórios telecomunicações, área específica a ser tratada nessa dissertação, não estavam adaptados para este novo contexto. Assim, surgiu a necessidade de que estes laboratórios pudessem ser operados à distância e de uma maneira que pudesse cumprir a ementa de um curso inicial de telecomunicações.

Cabe observar que essa demanda não foi decorrente exclusivamente da pandemia de COVID-19. A possibilidade de acesso remoto à estrutura laboratorial abre espaço para o surgimento de uma ferramenta em apoio às Instituições de Ensino à Distância, as quais vêm ganhando mercado no Brasil, conforme o gráfico apresentado na Fig. 1.3.

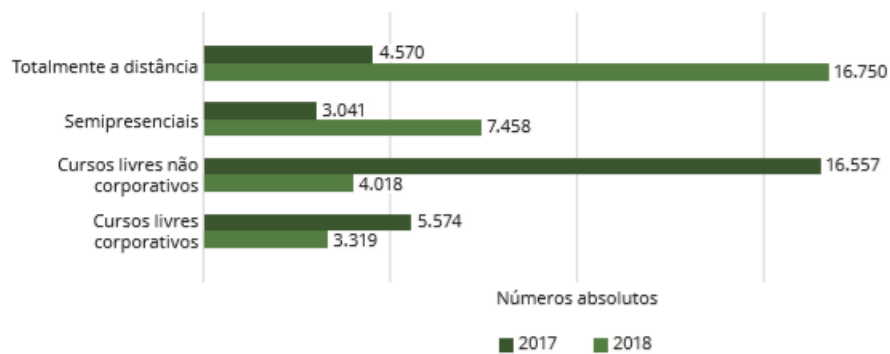


Figura 1.3: Quantidade de cursos oferecidos em EAD no Brasil em 2017 e 2018 (Censo Digital EAD 2018)

Entretanto, a principal limitação dos equipamentos empregados nos laboratórios de telecomunicações tradicionais é que eles normalmente apresentam hardware e software desenvolvidos para executarem um conjunto de funções específicas, de forma que qualquer outro tipo de sinal ou forma de onda não prevista inicialmente no escopo, dificilmente poderá ser trabalhada pelo equipamento. Assim, alterações na aplicação final do sistema podem requerer o desenvolvimento de um novo hardware ou um novo software para realizar a comunicação, o que eleva os custos do projeto devido aos desenvolvimentos e certificações, bem como diminui a flexibilidade do sistema (6).

Não obstante o aspecto de pouca flexibilidade dos equipamentos, observa-se também o aspecto custo. Os equipamentos listados no site do laboratório de eletrônica analógica, visualizado na Fig. 1.4, e do laboratório de telecomunicações, do curso de engenharia eletrônica da UNB-GAMA, possuem custos aproximados de mais de R\$ 30.000,00 por bancada.



Figura 1.4: Laboratório eletrônica - Lab NEI (<http://fga.unb.br/eletronica/infraestrutura>, acesso em 05 nov 2020)

Dessa forma, cabe verificar a disponibilidade de uma ferramenta capaz de atender a ementa de um curso de inicial de telecomunicações, que possa ser acessada remotamente e que possua

um custo razoável de implantação e utilização, frente a um laboratório tradicional.

## 1.2 UMA NOVA TECNOLOGIA

O espectro eletromagnético começou a ser utilizado com o surgimento do rádio proposto por Guglielmo Marconi no final do século XIX. Entretanto, em 1900, foi registrada a primeira interferência de radiofrequência durante a transmissão da corrida de Iates nos Estados Unidos, a qual estava sendo transmitida por Marconi e Lee de Forest na mesma frequência (7). Pouco tempo depois, a larga utilização de radiotransmissores na I Guerra Mundial despertou a percepção das autoridades para a regulação do Espectro eletromagnético, o qual tem sido um recurso bastante limitado, principalmente na área militar (8). Por ser um recurso escasso e caro, a utilização do espectro eletromagnético tem criado demandas para que os equipamentos de comunicação aproveitem ao máximo o espectro disponível nas suas transmissões (1) (9). Adicionalmente, no meio militar, o Departamento de Defesa Norte Americano encontrou como obstáculo não apenas as limitações de frequência, mas também a dificuldade de estabelecer comunicação de forma interoperável entre as Forças Armadas, o que originou a demanda por um rádio reconfigurável o suficiente para atuar como gateway entre os diversos sistemas de comando e controle na forma do *Joint Tactical Radio System (JTRS)* (10). Neste ponto estão destacados dois importantes requisitos para o desenvolvimento de um novo sistema de comunicação:

- Flexibilidade - Deveria ser reconfigurável para ser compatibilizado com diversos sistemas heterogêneos existentes nas Forças Armadas.
- Eficiência espectral - Deveria aproveitar ao máximo a capacidade do espectro de frequências que estivesse disponível.

Tal situação levou pesquisadores a desenvolverem o conceito de *Software Defined Radio (SDR)*, para definir um rádio em que todas ou quase todas as funções da camada física possam ser definidas por software (11). Assim, o hardware específico é substituído por um hardware geral que pode ser especializado, por meio de configuração de seu software de operação, para atuar com uma forma de onda desejada.

Esta flexibilidade permite atender novos e diversos contextos de emprego, incluindo modificações na frequência da portadora, banda de transmissão, forma de acesso ao canal, codificação de canal, tipo de modulação, dentre outros. Dessa forma, a finalidade e modo de operação do rádio pode ser alterada pela edição de um novo programa que estabeleceria novos mecanismos de comunicação para atender determinada demanda. Além disso, essa ferramenta, quando associada a um computador, pode ser configurada e operada remotamente para diversas finalidades, dentre elas, para a execução de práticas laboratoriais.

### 1.3 PROPOSTA DE UTILIZAÇÃO DE SDR NOS LABORATÓRIOS

A avaliação do contexto global da situação dos laboratórios universitários, considerando os custos, o aumento de cursos na modalidade EAD e as restrições de contato e locomoção decorrentes da pandemia de COVID-19, criou o ambiente propício para que os SDR se tornem uma ferramenta importante para a continuidade da atividade laboratorial nesse novo contexto. Dessa forma, vislumbrou-se a possibilidade de utilizar SDR nos laboratórios universitários, visto que possuem custos mais vantajosos que os equipamentos de bancada tradicionais e podem ser reconfigurados para atender diversos experimentos para cursos de graduação e pós-graduação, seja na modalidade presencial ou à distância.

Tanto nos cursos de graduação em Engenharia de Telecomunicações como em cursos correlatos, o primeiro contato do aluno com o formalismo dos conceitos de modulação, demodulação e desempenho de sistemas ocorre tipicamente em uma disciplina expositiva complementada por uma atividade laboratorial. O núcleo do escopo dos experimentos de laboratório geralmente consiste na reprodução, em bancada, da geração de formas de onda moduladas, no projeto de demoduladores e na medição de características essenciais dos sinais no contexto de transmissão de sinais. Como exemplo, têm-se as medições de largura de banda do sinal transmitido, frequência e potência de portadora, recuperação de sincronismo e alguma métrica de qualidade - como relação sinal-ruído ou taxa de erro de bits. Em nível de graduação, os SDR poderiam ser configurados para trabalharem as ementas tradicionais de um curso de telecomunicações como análise de espectro, modulação/demodulação AM e FM, análise de formas de pulso e estudo de modulação/demodulação de sinais digitais.

Os SDR amadureceram o suficiente para se tornarem uma alternativa mais barata no mercado em relação aos equipamentos tradicionais de bancada (12). Nesse sentido, evidencia-se que essas ferramentas podem ser usadas para equipar os laboratórios universitários a um custo menor que com itens tradicionais como osciloscópios, analisadores de espectros, dentre outros. Todas essas funções poderiam ser configuradas apenas por mudança de software, mantendo o hardware padronizado e ao mesmo tempo, flexível o suficiente para atender a diversos experimentos de nível de graduação e pós-graduação das Universidades. Um equipamento SDR, igual ao utilizado nessa dissertação, pode ser adquirido por valores inferiores a R\$ 16.000,00, tipicamente mais baratos do que o conjunto de bancada básico composto de fonte de tensão, osciloscópio, multímetro e componentes do transmissor e receptor.

Além do baixo custo, a flexibilidade de configuração de sistemas baseados em SDR permitem ao aluno e ao instrutor a realização de um passo além da simulação e da configuração de equipamentos de laboratório tradicionais. Já é de conhecimento a elaboração de uma estratégia geral de uso desse material no ensino de graduação (13) e, como mecanismos de demonstração da viabilidade de aplicação de SDR a tais estudos, este trabalho apresenta experimentos realizados com o software GNU Radio associado à plataforma Universal Software Radio Peripheral (USRP) para demonstração de diversos tipos de estudo normalmente vistos nas matérias de graduação, além da possibilidade de estender esses experimentos para o Ensino à Distância (EAD).

## 1.4 OBJETIVO DA DISSERTAÇÃO

Nesta dissertação são analisados os aspectos negativos e positivos da utilização de SDR em substituição aos equipamentos tradicionais de laboratório didático, para a execução de uma ementa de apoio laboratorial complementar às aulas teóricas de cursos de telecomunicações, em nível de graduação e pós-graduação, e que permita a utilização remota de um laboratório com equipamentos de transmissão reais para apoiar as instituições de ensino à distância.

Em nível de graduação demonstra-se a execução de experimentos básicos associados a uma disciplina introdutória de telecomunicações, após a aplicação de dois semestres consecutivos do uso parcial do ambiente proposto, como processos de translação em frequência, conceitos de modulação analógica de onda contínua e modulação digital. Algumas das ideias apresentadas neste trabalho já possuem estudos similares em outros locais do Brasil e do exterior (5) (14) (15) (12).

Em nível de pós-graduação, discute-se o emprego de novas técnicas de demodulação utilizando plataformas SDR, como o emprego de *machine learning* para a classificação de símbolos em um receptor adaptativo às variações do canal, a fim de demonstrar que um laboratório equipado com essa plataforma permite desde a elucidação de conceitos básicos de cursos de telecomunicações até pesquisas de novas tecnologias que podem ser aplicadas na melhoria de sistemas de telecomunicações.

## 1.5 ESTRUTURA DO TRABALHO

O capítulo 2 descreve a metodologia utilizada para o desenvolvimento deste trabalho, apresentando as ferramentas utilizadas para reforçar o embasamento teórico a ser complementado pelos experimentos. Além disso, o capítulo também contém a descrição geral do arcabouço de software utilizado para desenvolver os experimentos, bem como do hardware utilizado para estabelecer o acesso remoto e a comunicação física dos sinais.

O capítulo 3 procura delimitar o contexto de cada experimento a ser realizado, procurando colocá-los em grau crescente de dificuldade e de acordo com o andamento padrão da ementa de um curso introdutório de telecomunicações.

O capítulo 4 tem como objetivo dar uma visão inovadora para o emprego da teoria de *machine learning* nos processos de comunicação. Neste trabalho, o foco ocorre sobre o processo de demodulação de sinais digitais obtendo resultados bastante satisfatórios quando comparados com o processo de modulação tradicional.

Por fim, o capítulo 5 busca descrever as conclusões gerais e indicar as principais propostas para trabalhos futuros sobre o tema.

## 2 AMBIENTE EXPERIMENTAL

Os SDR possuem como característica a capacidade de modificar seus parâmetros de comunicação por meio de software de modo a transmitir e receber formas de onda específicas. Aliado a isso, é possível encontrar equipamentos com essa tecnologia por preços mais baixos que os equipamentos de bancada de testes, conforme citado em (12). Perseguindo o objetivo de diminuir custos, os experimentos desta dissertação utilizaram o software GNU Radio Companion (GRC), o qual pode ser obtido gratuitamente na internet, associado um front-end para transmissão e recepção de sinais, denominado Universal Software Radio Peripheral (USRP), o qual pode ser adquirido por valores que variam a depender da capacidade de processamento do modelo e das faixas de frequências disponíveis.

De forma geral, podemos dividir a arquitetura de uma plataforma baseada em SDR em duas partes: o front-end, responsável pela transmissão e recepção do sinal eletromagnético bem como realizar as devidas conversões de frequência para modulação e demodulação; e o back-end, responsável pela realização do processamento efetivo da informação (5). Existem diversos softwares que podem atuar como back-end, como o matlab, o labview e GNU Radio Companion. Optou-se, neste trabalho, pelo GRC por se tratar de um software livre e isento de custos, fato que se mostra como um ponto positivo para a montagem do laboratório com menor custo. Existem também diversos front-end disponíveis no mercado como o HackHf e USRP, porém, neste trabalho, foi utilizado o USRP por já ser um equipamento disponível na Universidade.

### 2.1 METODOLOGIA

A metodologia utilizada nesta dissertação se baseia em pesquisa teórica e experimentos realizados em ambiente de laboratório real do curso de telecomunicações da Universidade de Brasília (UNB).

A descrição experimental tomou como referência principal a ementa do curso de telecomunicações da UNB - Campus GAMA e considerou estudos essencialmente teóricos de livros, artigos e periódicos sobre a aplicação de ferramentas de *Machine Learning* aos sistemas de comunicação.

A partir da ementa destacaram-se os conceitos chaves que os alunos deveriam conhecer e foram elaborados experimentos didáticos com a finalidade de possibilitar a interação prática do aluno com os sistemas de telecomunicações, partindo-se de conceitos básicos como transformadas de fourier e translação em frequência, passando por métodos de modulação de onda contínua, formatação de pulso e, finalmente, aplicando conceitos de modulação digital mais avançados.

Assim, utilizou-se todo o ferramental de software e hardware para construir uma série de experimentos, testados remotamente, no escopo do programa de pós-graduação, a fim de sugerir



a aplicação dos SDR na criação de laboratórios remotos para aplicações de EAD. Os resultados mostram pontos positivos e negativos que devem ser avaliados previamente a uma implantação prática desse tipo de prática em Universidades.

Este trabalho também procurou mostrar que bancadas de laboratório baseadas em SDR permitem pesquisas que vão além de demonstrações didáticas básicas. Para comprovar essa possibilidade, foram realizadas pesquisas teóricas sobre a incorporação de ferramentas de *machine learning* como forma de integração de matérias e, principalmente, para a melhoria dos sistemas de comunicação. Nesse sentido, foi criado um experimento para demonstrar a utilização de redes MLP para a classificação de símbolos na comunicação digital QPSK, demonstrando um ganho de robustez à distorções comparativamente a um classificador tradicional baseado em limiar de decisão.

## 2.2 ARQUITETURA DO GNU RADIO COMPANION

O *GNU Radio Companion* (GRC), em sua versão 3.7.12.0 empregada neste trabalho, é um software livre que utiliza blocos com funções de processamento de sinais, os quais podem ser encadeados para formar um sistema real (16). O software é gratuito e permite simular aplicações que podem ter contato com o meio físico por intermédio de um front-end de RF. Dessa forma, o GNU Radio atua como um back-end para uma plataforma baseada em SDR.

O GNU radio apresenta uma Interface Homem-Máquina (IHM) construída em Python e utiliza uma ampla biblioteca de processamento de sinais. Os blocos de processamento podem ser desenvolvidos em Python ou em C++, a critério do desenvolvedor da aplicação considerando a diferença de velocidade de execução das linguagens, sendo que a estrutura do GRC permite a chamada de funções em C através da interface Python.

Embora possua ambiente desenvolvido primordialmente em Python, os blocos, quando se utilizam de funções escritas em linguagem C, são integrados por meio do *Simplified Wrapper and Interface Generator* (SWIG), que visualiza os blocos em Python como grafos e os relacionam ao código em C++ para uma execução mais adequada à velocidade de processamento exigida, conforme descrito na Fig. 2.1.

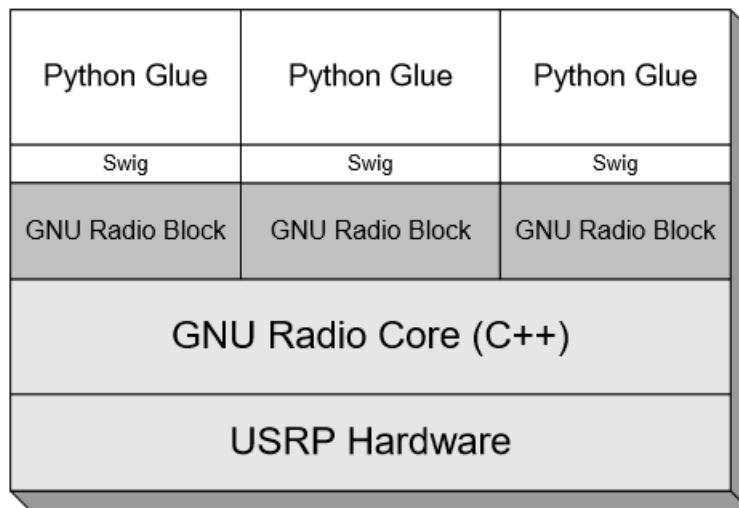


Figura 2.1: Arquitetura de funcionamento do GNU Radio

O aplicativo conta com blocos de processamento padrão como filtros digitais, fontes de sinais, analisadores de sinais no domínio da frequência e do tempo, além de ser uma ferramenta flexível a ponto de permitir a customização de blocos existentes ou mesmo a criação de blocos de processamentos novos.

A conexão do GRC com o front-end USRP é realizada fisicamente por meio de cabo Gigabit Ethernet e logicamente por meio de endereçamento IP e drivers específicos denominados *USRP Hardware Drivers* (UHD), os quais podem ser inseridos e configurados graficamente no ambiente GRC.

### 2.3 ARQUITETURA DO UNIVERSAL SOFTWARE RADIO PERIPHERAL

O *Universal Software Radio Peripheral* (USRP) é um hardware flexível que pode atuar como front-end de RF para os sinais processados no GRC ou pode atuar como o próprio processador de sinais caso a aplicação de processamento seja portada para a placa (17). O desenvolvimento de hardwares desse nível só foi possível graças ao avanço tecnológico de diversos componentes, dentre os quais se destacam os *Digital Signal Processors* (DSP), conversores digital-analógico e os *Field Programmable Gate Arrays* (FPGA).

O USRP possui uma arquitetura básica composta por uma placa mãe, placas filhas e conversores A/D e D/A e está apresentada na Fig. 2.2.

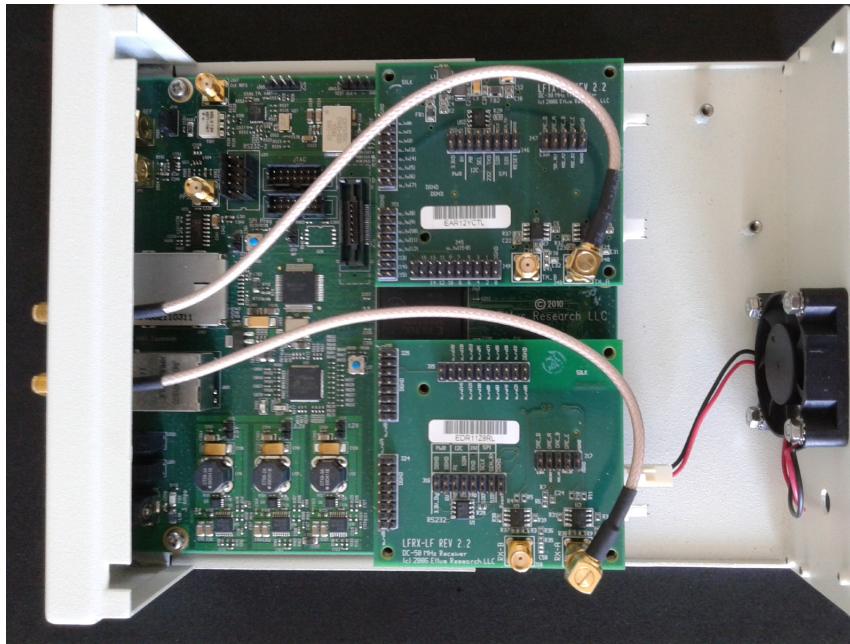


Figura 2.2: Vista superior da placa USRP modelo N210

A placa mãe é responsável pelo processamento da informação obtida nas placas filhas, pela alimentação elétrica dos componentes e por fazer a interface com o computador hospedeiro. O principal elemento deste componente é o FPGA Xilins Spartan-3A, o qual pela translação em frequência e pelos processos de convenção de taxas de amostragem do sinal (interpolação/decimação). A placa mãe admite sinal de clock externo, emprego de módulos de antenas *Multiple Input Multiple Output* (MIMO), além de disponibilizar portas para transmissão e recepção dos sinais. A comunicação com o computador é realizada por meio de uma interface de rede Gigabit Ethernet exclusivamente. Não há comunicação com interface de rede de velocidades menores como 10/100 Mbps, a menos que se utilize um switch com portas 10/100/1000 Mbps como intermediário da conexão. A necessidade de conexões de alta velocidade é justificada pela grande quantidade de amostras de sinais que são processadas nas aplicações.

Um SDR, pelo menos em nível teórico, deve ser capaz de receber e emitir sinais em uma ampla faixa de frequências por meio de um sistema irradiante que trabalhe com eficiência em diversas bandas de frequência, entretanto, muitos desenvolvedores utilizam placas de transmissão e recepção apenas em faixas específicas, papel desempenhado pelas citadas placas filhas. No experimento em questão, foram utilizadas placas LFTX para transmissão e LFRX para recepção, conforme mostrado na Fig. 2.3. O espectro de operação das placas é de DC até 30 MHz e funcionam bem para ambientes cabeados, mas são mais difíceis de serem utilizadas para irradiação em espaço livre devido à dificuldade de encontrar antenas nessa faixa de operação que sejam compactas o suficiente para um ambiente laboratorial. A utilização de placas na faixa de V/UHF possibilitaria uma maior capacidade de demonstração de exemplos didáticos utilizando sinais de televisão e de radiodifusão.

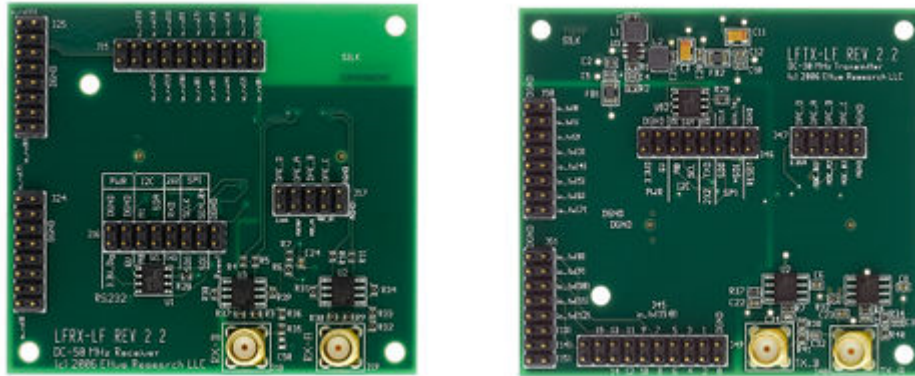


Figura 2.3: Placas LFTX (transmissão) e LFRX (recepção)

O USRP utiliza conversores AD9777 da Analog Devices, os quais conseguem realizar amostragens da ordem de 160 Msps e podem codificar o sinal amostrado em 8 ou 16 bits. Esse fluxo de bits é repassado ao computador hospedeiro responsável pelo tratamento do sinal. Dessa forma, há necessidade de um canal de grande capacidade entre o USRP e o computador hospedeiro, o que neste caso é realizado por meio de uma porta Gigabit Ethernet.

### 2.3.1 Anaconda e Jupyter

Para o desenvolvimento dos experimentos foi utilizado o software Anaconda 3 para criar um ambiente Python com bibliotecas científicas específicas como Numpy e Matplotlib (18)(19). O Anaconda permite a criação de ambientes virtuais isolados de forma que em cada um deles seja possível a instalação de bibliotecas de versões específicas para determinado experimento. No caso deste trabalho, não houve necessidade de alternar entre diversos ambientes, uma vez que não houve mudança de versão das ferramentas. Associado ao Anaconda, também foi utilizado o aplicativo Jupyter Notebook (versão 6.0.0) (20), o qual permitiu a edição de códigos Python, além de permitir a integração com as bibliotecas e ambientes do Anaconda.

### 2.3.2 Scikit-Learn

O Scikit-Learn (versão 0.21.2) é um software livre que dispõe de diversas bibliotecas com funções em Python que implementam variados algoritmos de aprendizado de máquina e análise de dados (21) e foi utilizado de forma conjunta com as demais bibliotecas do ambiente Anaconda. Neste trabalho, buscou-se verificar como a variação de parâmetros de uma rede neural Multilayer Perceptron (MLP) influencia no desempenho da classificação de símbolos sujeitos a ruídos e outras interferências externas. (4). Para tanto, foram empregadas funções de ajuste de dados utilizando-se classificadores baseados em redes MLP com variadas camadas ocultas e funções de ativação.

### 2.3.3 Teamviewer e Conjunto Experimental

Outra questão importante neste trabalho é demonstrar que os experimentos realizados com *software radio* podem ser utilizados no ensino à distância, tornando-se uma alternativa mais barata e prática de lecionar matérias de telecomunicações. Todos os softwares e hardwares apresentados nas seções anteriores foram instalados em um computador hospedeiro. Adicionalmente a esses softwares já mencionados foi utilizado também o software Teamviewer, que é capaz de fornecer o acesso remoto a um computador hospedeiro a partir de um computador ou dispositivo portátil remoto que seja compatível com o aplicativo, implicando que todas as simulações que utilizem ou não o USRP podem ser executadas a partir de um computador/dispositivo remoto. A Fig. 2.4 abaixo apresenta a estrutura de testes montada para validar os experimentos remotamente.

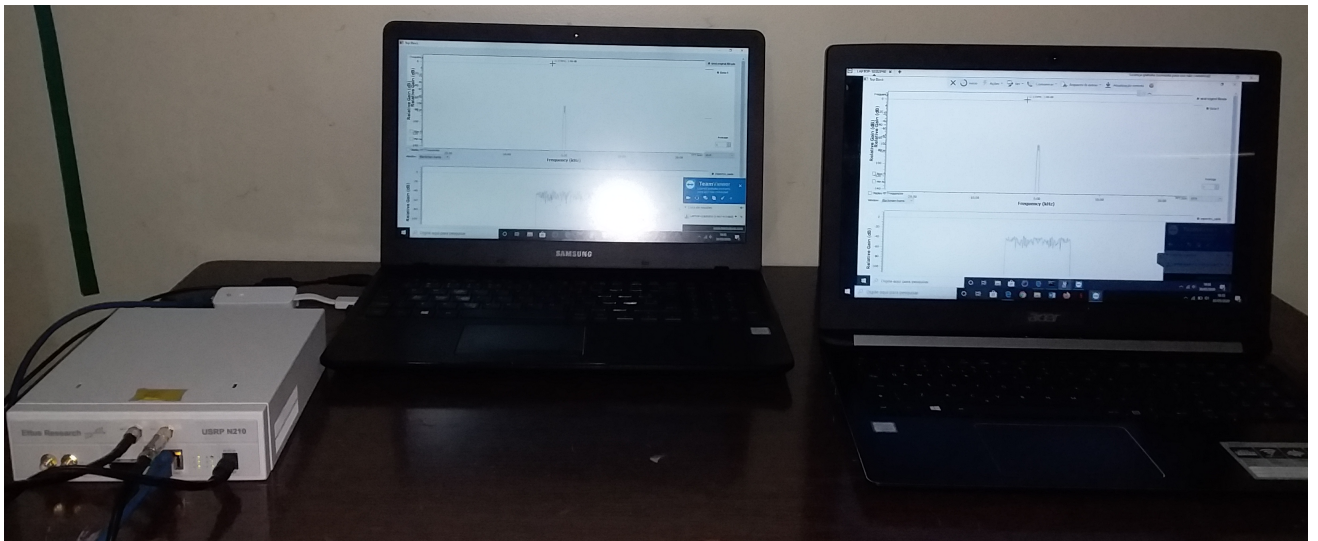


Figura 2.4: Bancada de testes para validação dos experimentos

O Teamviewer foi escolhido para essa demonstração por ser de instalação, configuração e operação fáceis e por ser gratuito para uso único e pessoal. Para outros tipos de acesso, o software requer uma assinatura paga com valores que variam de acordo com as quantidades de usuários para acesso. Entretanto, há uma variedade de softwares que podem ser utilizados para o acesso remoto com funções que podem contar com a conexão simultânea de vários usuários, com contas diferentes, em um mesmo computador hospedeiro. Outras configurações de segurança e restrição de acesso a pastas de acordo com as contas dos alunos podem ser requeridas e configuradas em softwares como o Remmina (similar do Teamviewer para Linux com algumas funcionalidades diferentes).

## 3 EXPERIMENTOS COM GNU RADIO E USRP

### 3.1 EXPERIMENTOS DEMONSTRATIVOS

O aparato experimental apresentado no capítulo anterior possibilitou o desenvolvimento de diversos experimentos com o objetivo de demonstrar a aplicabilidade dos SDR como ferramentas capazes de substituir os equipamentos tradicionais de laboratório em aplicações de telecomunicações analógicas e digitais. Foram elaborados experimentos partindo-se de situações simples, como filtragens, até transmissões digitais utilizando demoduladores não tradicionais com a utilização de ferramentas de *machine learning*. Dessa forma, foram realizados 06 (seis) experimentos que acompanham o fluxo tradicional de matérias ministradas em cursos introdutórios de telecomunicações:

- Filtragem de sinais
- Modulação AM
- Modulação FM
- *Phase Locked Loop* (PLL)
- Modulação em banda base
- Demodulação QPSK utilizando redes neurais

A execução desses experimentos foi realizada no USRP com auxílio da configuração do GRC, transmitindo por meio cabeado, da placa LFTX para a placa LFRX. Nessa configuração, todo o ruído inserido na transmissão foi colocado artificialmente por meio da utilização de um bloco específico do GRC.

Em um ambiente de laboratório, vislumbra-se que o aluno poderia realizar os experimentos utilizando um mesmo USRP para transmissão e recepção dos sinais ou poderia executar a comunicação com um segundo USRP, por exemplo, o do professor, o qual avaliaria a qualidade do transmissor elaborado pelo aluno a partir de seu próprio *setup* de recepção. Além da possibilidade de transmissão direta do rádio do aluno para o rádio do professor, há a possibilidade de se ter como *output* um arquivo com os resultados gerados pelo sistema proposto, o qual pode ser encaminhado ao professor para execução e avaliação *a posteriori*.

Outra grande importância da utilização de SDR nos laboratórios é a viabilidade para acessá-los remotamente, o que possibilita realizar experimentos à distância em um laboratório equipado com hardware SDR e computadores com softwares específicos que possam ser acessados remotamente, podendo inclusive filmar o experimento e observar o status dos LED do front-end.

As plataformas de ensino à distância têm ganhado grande penetração de mercado e poderiam ser complementadas por esse tipo de prática de laboratório e, em situações emergenciais, como as decorrentes da limitação de locomoção devido a pandemia do COVID-19, a estrutura de laboratório remoto permaneceria funcionando, a fim de possibilitar o acesso aos experimentos, os quais estariam ocorrendo de forma real no laboratório. Esse tipo de abordagem aumenta o alcance dos laboratórios universitários uma vez que qualquer usuário conectado a uma rede/internet poderia acessá-lo e obter resultados reais, assim o alcance seria praticamente mundial.

A disponibilidade dos laboratórios também seria aumentada visto que os sistemas poderiam ser mantidos em funcionamento praticamente 24h por dia, parando apenas para manutenções. O custo de implantação é competitivo com relação aos equipamentos tradicionais, além disso, a possibilidade do laboratório ser operado remotamente permite algumas flexibilidades, como ser compartilhado com outras instituições (ou campi) e a utilização em qualquer horário, sem a necessidade de um técnico ou professor, caso transmissor e receptor sejam conectados diretamente.

No âmbito deste trabalho, foi instalado o aplicativo TeamViewer para demonstrar a realização de testes de simulação utilizando o GRC a partir de um computador remoto e os resultados mostraram controle total de todos os parâmetros de simulação. Outra vantagem é que o processamento de todos os dados foi realizado pelo computador hospedeiro, o que viabiliza o teste de aplicações que requerem maior processamento sem a necessidade de planos de internet fora dos convencionais. Para demonstração disso, foi executado remotamente o treinamento de uma rede Multilayer Perceptron (MLP) 5 x 5, de cerca de 02 min de execução, cujos resultados de consumo de recursos podem ser observados na tabela 3.1 e comprovam que o processamento fica a cargo do hospedeiro.

O treinamento de redes neurais é um processo iterativo que consome muitos recursos computacionais, visto que, conforme será explorado em seções posteriores, realiza contas como multiplicação de matrizes e cálculo de derivada de funções custo. Se o processamento dependesse do computador que está acessando o SDR remotamente, a proposta de realizar os experimentos à distância poderia ser comprometida para dispositivos com baixo poder computacional. Porém, com o processamento se concentrando no SDR (hospedeiro), não há demanda computacional elevada para o dispositivo do usuário, facilitando a utilização do laboratório.

Tabela 3.1: Consumo de recursos no computador remoto e hospedeiro

PARÂMETRO	HOSPEDEIRO	REMOTO
CPU	48%	3%
RAM	4,8GB	5,3GB



## 3.2 FILTRAGEM DE SINAIS

Os filtros são elementos básicos utilizados em sistemas de comunicação e normalmente são empregados para separar sinais com componentes nas frequências interesse. Os filtros digitais podem ser implementados a partir de uma versão analógica considerando a frequência de amostragem, mas dependendo do método de síntese do filtro, eles podem apresentar características diferentes. Não é objetivo deste trabalho demonstrar cada tipo de síntese de filtro especificamente, mas apenas verificar o comportamento resultante de um sinal ao ser submetido a um filtro passa-baixa, passa-faixa e passa-alta, a fim de apresentar visualmente uma comprovação do funcionamento do filtro em um sinal real.

O GRC possui uma diversidade de filtros cujos principais parâmetros de configuração são as frequências de corte, frequência de amostragem, janelamento e tamanho da faixa de transição entre as bandas de passagem e de atenuação, os quais podem ser ajustados durante a execução. Uma onda quadrada  $s(t) = \text{rect}(t)$ , quando visualizada no domínio da frequência, é uma função  $S(f) = \text{sinc}(f)$  com infinitas componentes, o que torna mais fácil verificar o resultado dos filtros apenas pela observação dos resultados destes no espectro de frequência do sinal. Neste sentido, elaborou-se um experimento no GRC cujo objetivo consistiu em filtrar uma onda quadrada de 1V de amplitude e frequência de 1kHz, conforme apresentado na Fig. 3.1, a fim de verificar o comportamento do sinal após a filtragem no domínio da frequência.

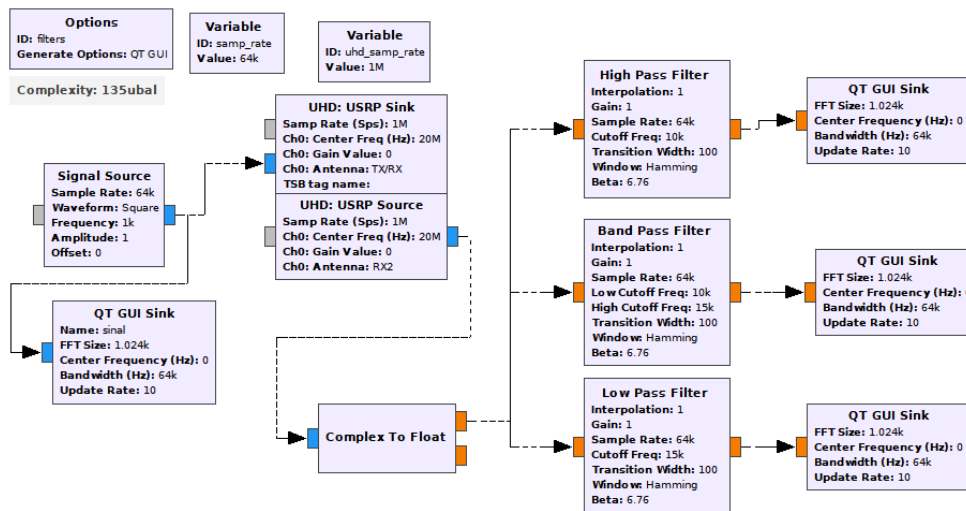


Figura 3.1: Filtragem de onda quadrada

Nesse contexto, foram observados os resultados apresentados na Fig. 3.2, em que uma onda quadrada foi filtrada por um filtro passa-baixa, um filtro passa-faixa e um filtro passa-alta.



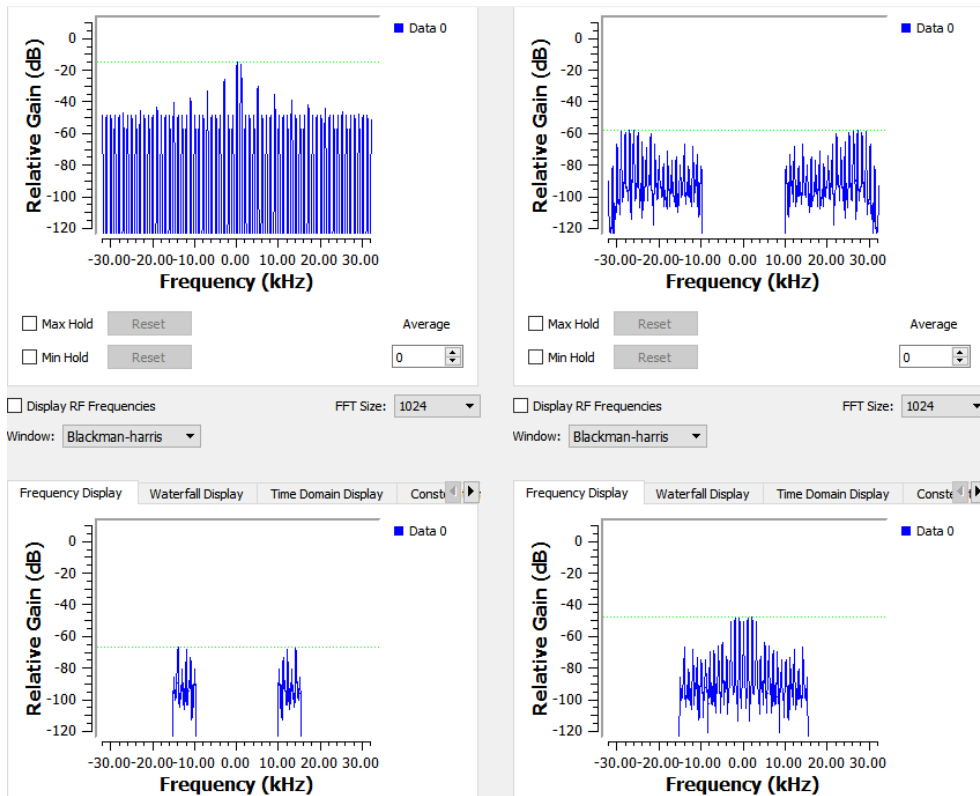


Figura 3.2: Principais tipos de filtragem

Observou-se que os filtros modificaram as componentes de frequência do sinal conforme sua característica de passa-baixa, passa-faixa ou passa-alta. A presença de componentes do sinal dentro da banda de rejeição é decorrente das características não ideais dos filtros e apesar disso, os valores dessas componentes são inferiores a -100 dB comparativamente à magnitude do sinal filtrado de interesse.

### 3.3 MODULAÇÃO EM AMPLITUDE - AMPLITUDE MODULATION (AM)

A técnica de AM consiste em variar linearmente a amplitude de um sinal portador de acordo com as características de um sinal de mensagem, em banda base, a ser transmitido. Por banda base entende-se o sinal antes da aplicação de qualquer processo de translação de frequência decorrente da multiplicação deste por um sinal senoidal/cossenoidal. Assim a expressão de um sinal modulado em amplitude é apresentado na equação 3.1 abaixo:

$$s(t) = A_c[1 + k_a m(t)] \cos(2\pi f_c t) \quad (3.1)$$

A modulação em amplitude considera a restrição  $|k_a m(t)| < 1$ , para evitar a sobremodulação, que pode ser observada na prática como uma distorção na envoltória do sinal, prejudicando a fase e conseqüentemente a demodulação do sinal mensagem. O índice  $K_a$ , portanto, é um fator que

ajusta o quanto da amplitude do sinal portador será utilizado para excursionar o sinal mensagem e é denominado sensibilidade à amplitude. A multiplicação de uma onda cossenoidal de frequência  $f_c$  por um sinal de mensagem translada todo o espectro de frequência deste da banda base para um região em frequência centralizada em  $f_c$ . Assim um sinal de mensagem  $m(t)$ , cuja frequência máxima de seu espectro seja  $W$ , teria a representação em frequência conforme apresentado na Fig. 3.3:

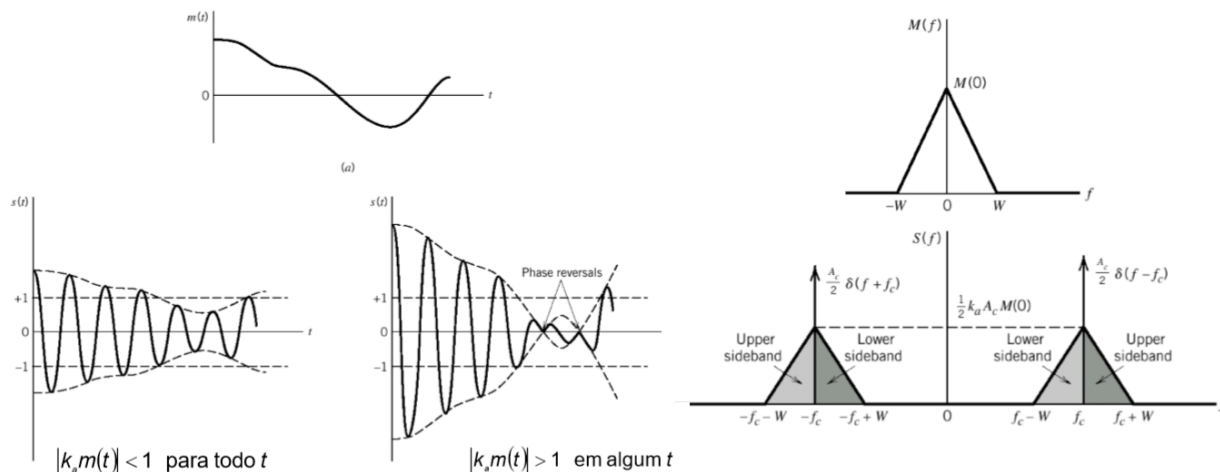


Figura 3.3: Modulação em frequência no tempo e no espectro (1)

Feitas essas definições básicas, podemos considerar o caso geral para modulação em amplitude. Toda a estrutura de modulação/demodulação do GRC e do USRP são baseados no mecanismo de fase e quadratura, assim é possível visualizar o formato geral de um sinal AM conforme a expressão abaixo.

$$s(t) = s_I(t) \cos(2\pi f_c t) - s_Q(t) \sin(2\pi f_c t) \quad (3.2)$$

O termo  $s_I$  e  $s_Q$  são as chamadas componente em fase e em quadratura do sinal.

Para a realização do experimento, foi produzido um sinal  $s(t)$ , utilizando um tom senoidal  $m(t) = A_m \cos(2\pi f_c t)$ , de amplitude 1 V e frequência 1 kHz, modulada na frequência portadora de 20 MHz, conforme o diagrama de bloco desenvolvido no GRC e apresentado na Fig. 3.4. É importante destacar que nem todos os blocos apresentados na figura são necessários para o funcionamento do modulador, mas ajudam no processo de construção e transição do ambiente de simulação para o ambiente real do USRP ou mesmo tornam o experimento mais realista, como por exemplo com a adição de blocos de fontes de som que podem ser reproduzidos e avaliados durante o experimento.

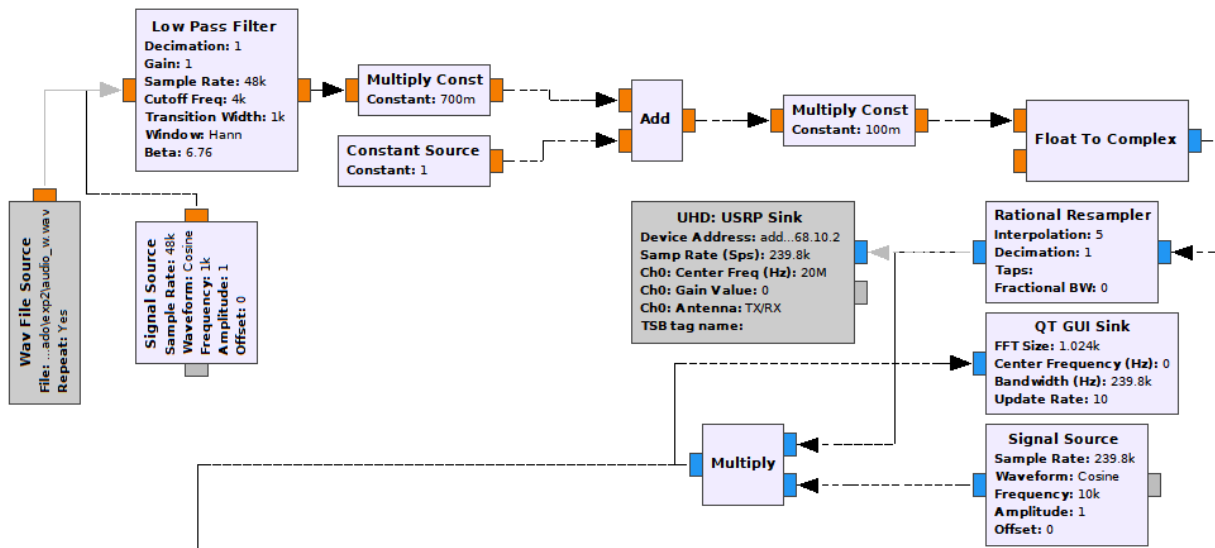


Figura 3.4: Modulador AM no GNU Radio

Para realizar a demodulação, optou-se pela técnica de detecção coerente visualizada na Fig. 3.5, mas poderia ser usada a técnica de detecção de envoltória. O demodulador consistiu em transladar em frequência, para banda base, o sinal  $m(t)$  original e utilizar um filtro passa-baixa para melhorar a relação sinal-ruído a partir da eliminação de sinais espúrios fora da frequência do tom senoidal de interesse  $m(t)$ .

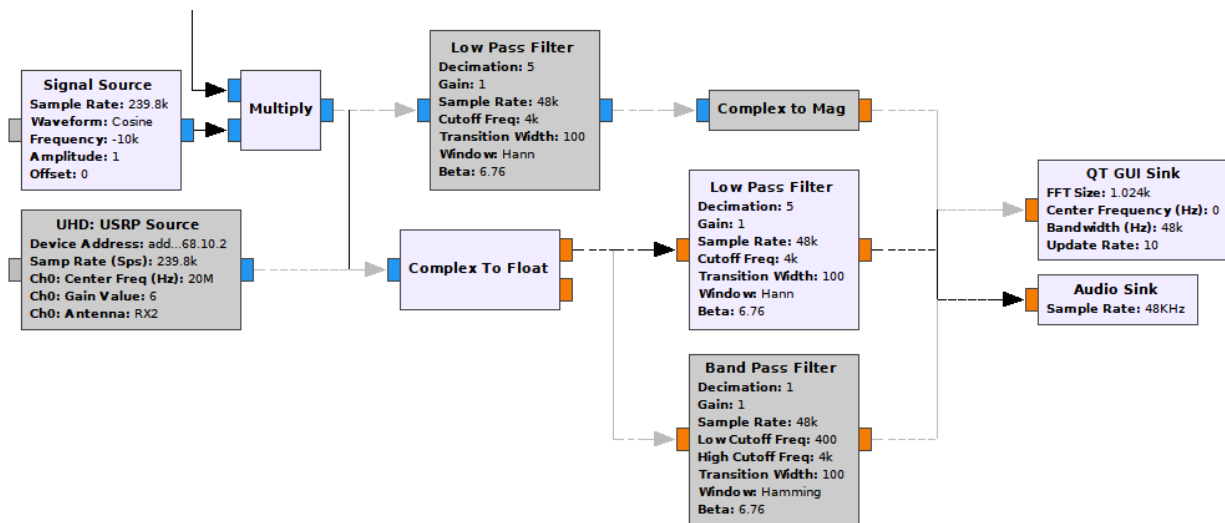


Figura 3.5: Demodulador AM no GNU Radio

No mesmo experimento, aplicou-se o sinal resultante da demodulação ao sistema de áudio do computador o que permitiu a escuta do tom senoidal enviado.

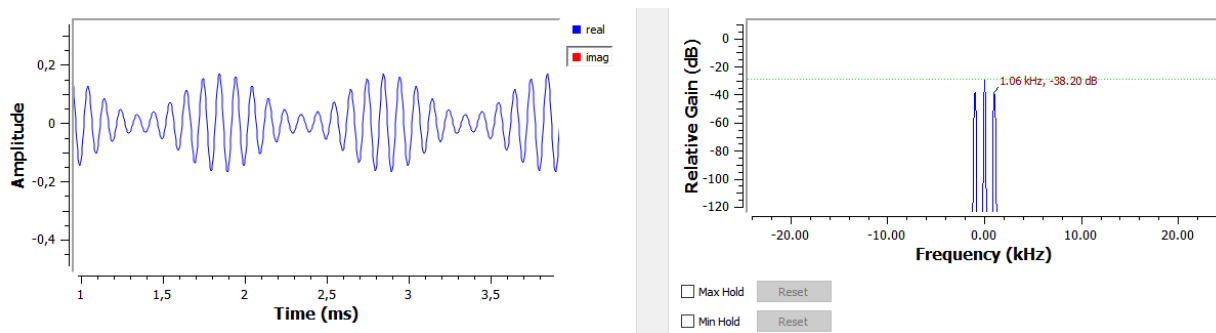


Figura 3.6: Resultado da modulação AM no tempo e na frequência

A análise da Fig. 3.6 mostra que o tom senoidal  $m(t)$  de 1kHz (À esquerda) foi recuperado com sucesso tendo apresentado apenas redução de amplitude em virtude das atenuações do processo de transmissão, demodulação e filtragem, o que pode ser compensado com um ganho no sinal recebido.

Acrescentando um pouco de complexidade ao tema, realizou-se a transmissão de um áudio na faixa de voz humana a fim de reproduzi-lo na saída de áudio do computador. O objetivo é trazer maior realidade para os experimentos realizados em aula e adicionar mecanismos novos como a utilização de arquivos de áudio, mecanismos de reprodução de som e reamostradores para compatibilizar as taxas de amostragem entre os diversos segmentos do sistema. O áudio foi inicialmente filtrado para que fosse limitado em uma banda de até 4kHz, a fim de ocupar menos banda no momento da transmissão e para exemplificar o que acontecia nos primórdios da comunicação analógica via rede de telefonia pública. Os sinal demodulado também foi reproduzido no sistema de áudio do computador e a qualidade do resultado é dependente de diversos fatores como taxa de amostragem, filtros e atenuações/ruídos do meios que não seriam observados facilmente em um sistema simulado. Embora em sala só tenha sido apresentado o mecanismo AM DSB+C, outros mecanismos podem ser realizados para demonstrar formas de AM.

### 3.4 MODULAÇÃO EM FREQUÊNCIA - *FREQUENCY MODULATION (FM)*

A modulação FM se caracteriza por ser mais robusta comparativamente a modulação AM visto que a mensagem é transmitida por variação no parâmetro frequência da onda portadora e dessa forma está menos suscetível a ação do ruído aditivo normalmente considerado no canal. Ou seja, embora a modulação em frequência seja afetada pelo ruído, não é tão prejudicada quanto o efeito do ruído sobre a modulação em amplitude.

A modulação em frequência é uma modulação da classe contínua não-linear e seu estudo pode ser facilitado quando escolhemos um sinal mensagem senoidal/cossenoidal (1). Para início do estudo, observa-se que o parâmetro frequência instantânea de um sinal modulado em frequência, para uma mensagem genérica  $m(t)$ , é dado pela expressão (3.3) a seguir:

$$f_i(t) = f_c + k_f m(t) \quad (3.3)$$

Aplicando essa equação 3.3 a um sinal cossenoidal observa-se a expressão abaixo:

$$s(t) = A \cos \phi = A \cos\left(\int 2\pi f_i t\right) = A \cos(2\pi f_c t + 2\pi k_f \int m(t) dt) \quad (3.4)$$

A expressão 3.4 apresenta o caso geral da modulação FM para qualquer sinal  $m(t)$ .

$$s(t) = A \cos(2\pi k_f \int m(t) dt) \cos(2\pi f_c t) - A \sin(2\pi k_f \int m(t) dt) \sin(2\pi f_c t) \quad (3.5)$$

Comparando-se a expressão acima, com a descrição canônica da modulação em fase e quadratura, observa-se que:

$$s(t) = I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t) \quad (3.6)$$

$$I(t) = A \cos(2\pi k_f \int m(t) dt) \quad (3.7)$$

$$Q(t) = A \sin(2\pi k_f \int m(t) dt) \quad (3.8)$$

A expressão em fase e em quadratura é bastante relacionada a estrutura de modulação e demodulação do GRC e da placa USRP N210, uma vez que esta utiliza o esquema de modulação em fase e quadratura, conforme apresentado na figura 3.7.

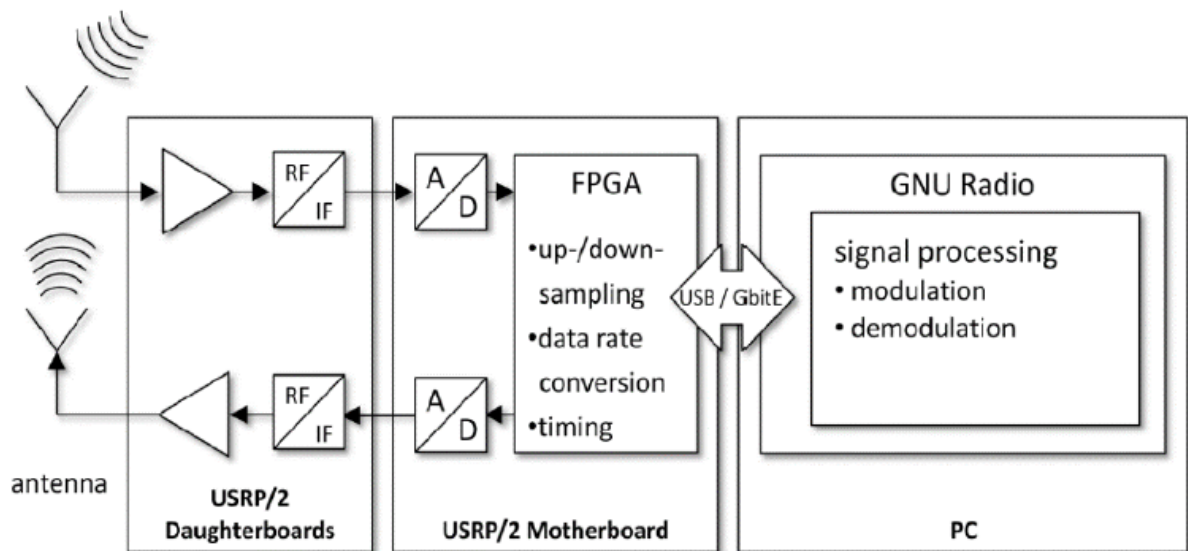


Figura 3.7: Diagrama esquemático da placa USRP e GRC

É comum simplificar o estudo da modulação avaliando os seus efeitos para o caso em que  $m(t)$  é uma onda senoidal/cossenoidal. Assim tem-se:

$$m(t) = A_m \cos(2\pi f_m t) \quad (3.9)$$

De 3.9 em 3.8 vem que:

$$s(t) = A \cos(\beta \sin(2\pi f_m t)) \cos(2\pi f_c t) - A \sin(\beta \sin(2\pi f_m t)) \sin(2\pi f_c t) \quad (3.10)$$

O parâmetro  $\Delta f$ , da expressão abaixo, é o desvio de frequência e também surge o parâmetro  $\beta$ , denominado índice de modulação em frequência, dado conforme a expressão 3.11.

$$\beta = \frac{A_m k_f}{f_m} = \frac{\Delta f}{f_m} \quad (3.11)$$

A partir de então os estudos teóricos normalmente se dividem em considerar o grau de modulação do sinal frente ao valor da frequência da onda portadora. Quando o sinal transmitido atende a condição  $A_m k_f = \Delta f \ll f_c$ , ou seja, quando  $\beta \ll 1$ , tem-se a modulação FM de banda estreita ou *Narrow Band Frequency Modulation* (NBFM). A consideração oposta para  $\beta$  é considerado como o caso FM banda larga ou *Wide Band Frequency Modulation* (WBFM).

Na modulação em frequência de uma mensagem senoidal banda larga observa-se um espectro formado por componentes com amplitudes que variam de acordo com uma função de Bessel no domínio da frequência (1), decorrente da expansão em série de Fourier do sinal modulado, conforme 3.12, bem como seu espectro de frequência definido em 3.13 .

$$S(f) = A_c \sum_{i=-\infty}^{\infty} J_n(\beta) \cos(2\pi(f_c + n f_m)t) \quad (3.12)$$

$$S(f) = A_c \sum_{i=-\infty}^{\infty} J_n(\beta) [\delta(f - f_c - n f_m) + \delta(f + f_c + n f_m)] \quad (3.13)$$

Para avaliar o espectro de frequência, emprega-se a regra de Carson como aproximação prática para determinar a banda consumida pelo sinal na qual, para a modulação em frequência de um tom senoidal  $m(t) = A_m \cos(2\pi f_m t)$ , é dado por:

$$B_T = 2(\Delta f) + 2f_m \quad (3.14)$$

Dessa forma, verifica-se que o espectro de frequência de um tom de mensagem senoidal é um conjunto de funções impulsos espaçados por uma frequência de  $f_m$  e de amplitude ajustada pelo

valor da função de Bessel naquela frequência, conforme a Fig.3.8. Por meio do experimento mostrado, os alunos podem variar a frequência do sinal  $m(t)$  e verificar o comportamento do espectro de frequência do sinal modulado, comprovando a teoria. Na Fig. 3.8, quando observada comparativamente entre linhas, observa-se a variação do  $\delta f$ , para um mesmo valor  $f_m$ , quando se modifica o valor de  $\beta$ . Quando observada comparativamente entre colunas, verifica-se a modificação do espectro, para um mesmo valor de  $\beta$ , ao se variar apenas  $f_m$ .

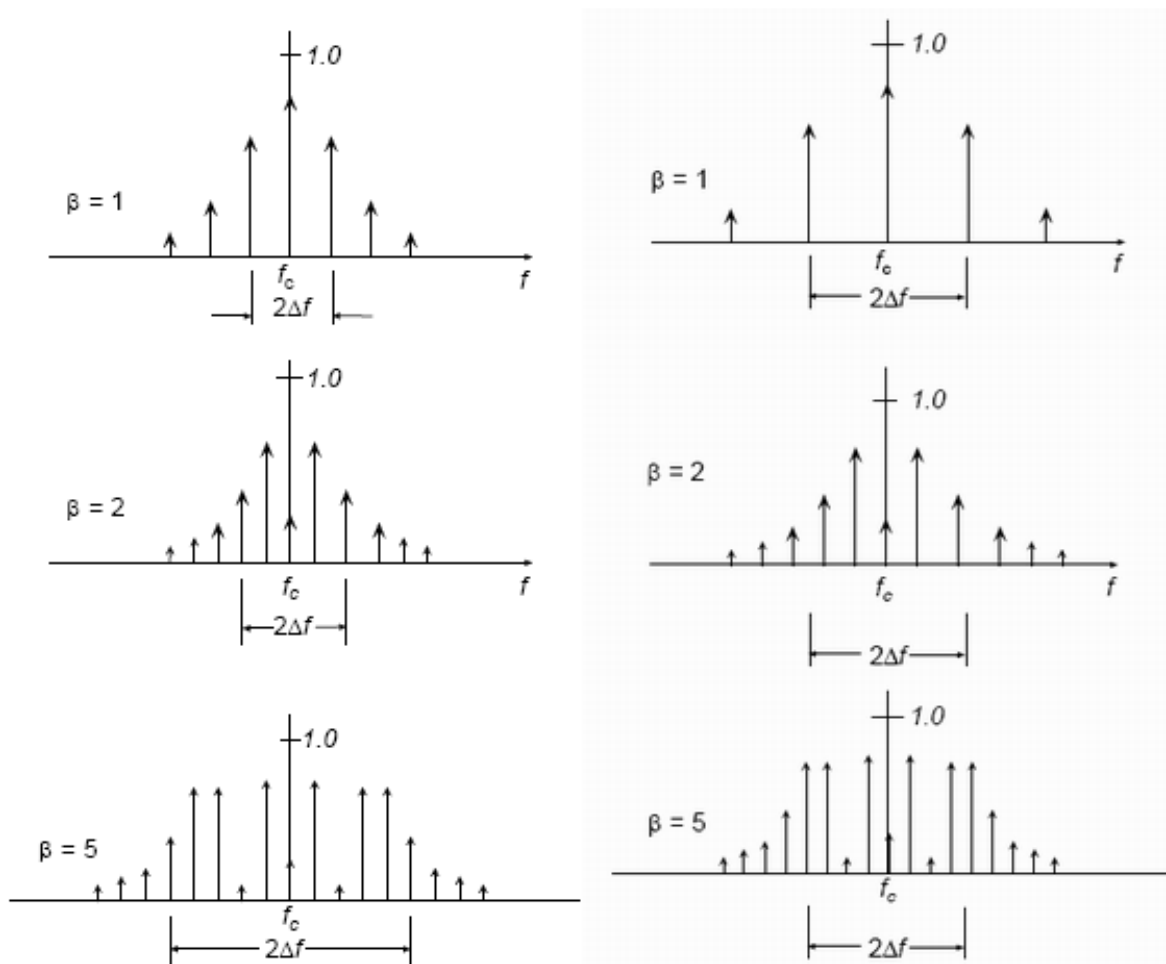


Figura 3.8: Espectro de frequência de um sinal senoidal modulado em frequência (1)

Para um sinal de forma geral limitado em banda, é razoável assumir, para efeitos de consumo de banda, a modulação de um tom senoidal da frequência mais elevada que compõe o sinal para se ter uma idéia do comportamento do sistema no pior caso possível. É conveniente, na prática, aplicar o sinal a um filtro limitador de banda (normalmente passa-baixa), de forma a se ter o controle da máxima frequência do sinal a ser modulado.

Neste trabalho, utilizou-se o método de sintetização digital, no qual as expressões matemáticas de modulação e demodulação são executadas diretamente por meio de processamento digital de sinais, sem recorrer a considerações específicas para simplificação do projeto.

A criação de circuito de modulação e demodulação FM consistiu em interligar blocos ele-

mentares de telecomunicações do GNU radio conforme apresentado na Fig. 3.9. O sinal  $m(t)$  consistiu em uma onda senoidal aplicada filtro passa-faixa limitador de banda para reduzir o conteúdo espectral da mensagem, de forma a se ter controle sobre qual a frequência máxima do sinal no momento de avaliar o consumo de banda. Para o sinal cossenoidal o filtro não tem influência, mas o experimento também foi realizado para limitar o sinal de voz de um arquivo de audio de entrada.

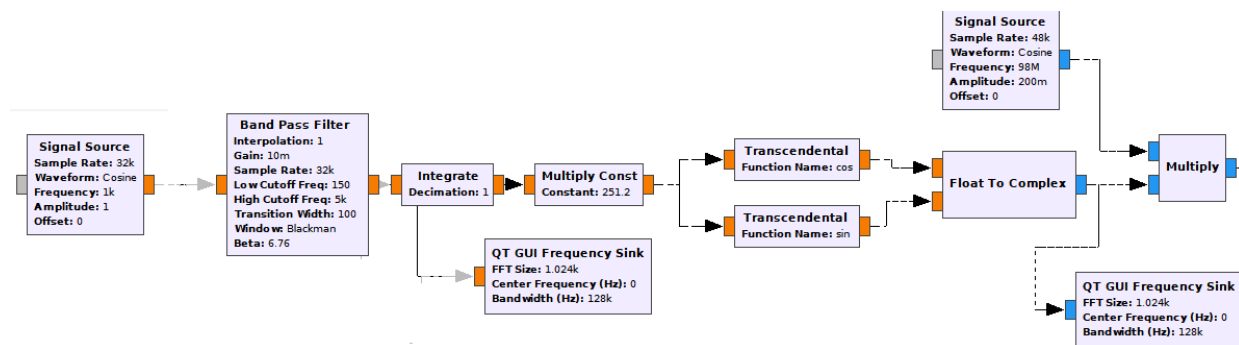


Figura 3.9: Modulador FM direto

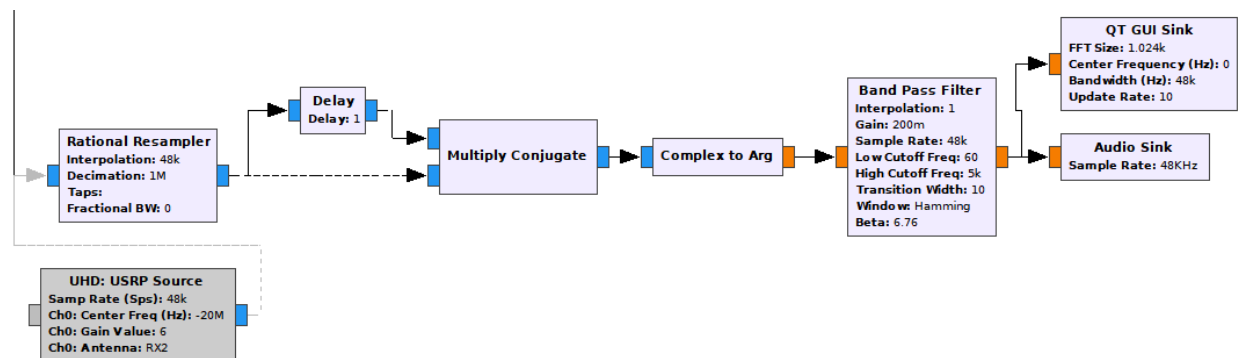


Figura 3.10: Demodulador FM direto

A saída do modulador é dada por  $I(t) + jQ(t)$ , enquanto que a saída da fonte senoidal é dada por  $\cos 2\pi f_c t + j \sin 2\pi f_c t$ . O bloco de multiplicação por uma portadora senoidal só foi utilizado para a simulação, uma vez que em teste no ambiente real foi usado o bloco (UHD) da placa de transmissão, que já executa a translação em frequência.

Para o sinal de entrada  $m(t)$ , o demodulador consiste em um diferenciador discreto baseado na equação:

$$\arg(s(t)s(t-1)) = \arg[\exp(-j2\pi\phi(t)) \exp(j2\pi\phi(t-1))] = \phi(t) - \phi(t-1) = 2\pi f_c m(t) \quad (3.15)$$

Foi verificado que o conteúdo espectral do sinal modulado se comportou de maneira muito similar ao esperado teoricamente. A Fig. 3.11 apresenta os resultados observados no experimento. O aluno pode realizar modificações em parâmetros do sinal mensagem e verificar o resultado disso



no sinal modulado.

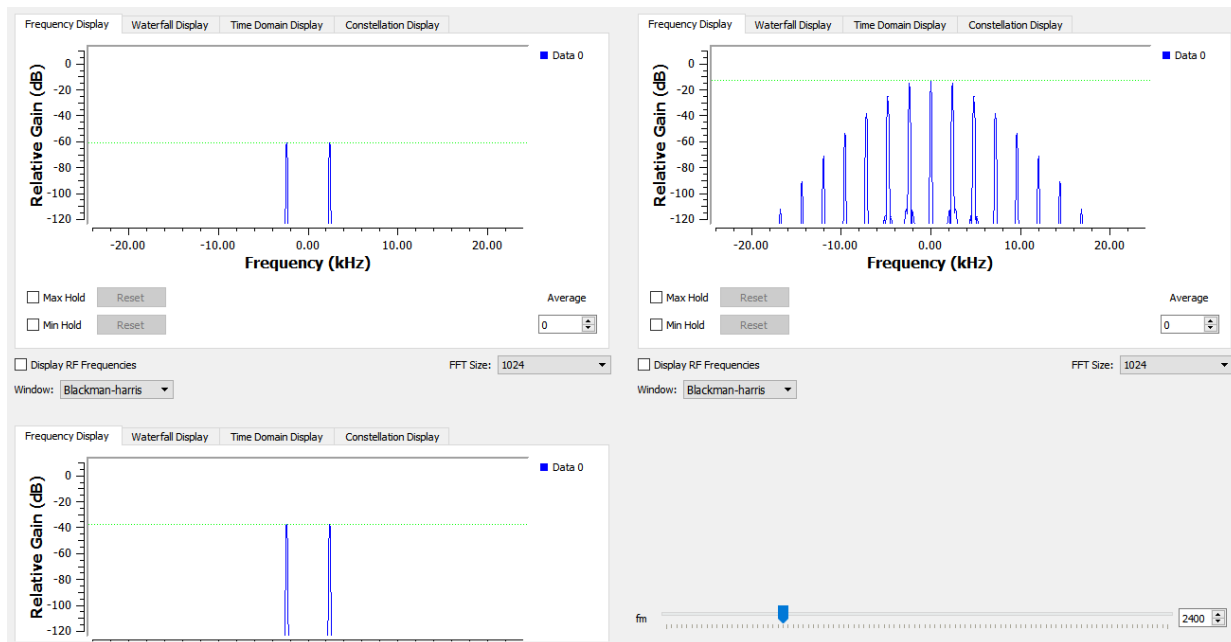


Figura 3.11: Espectro de tom senoidal representado por função de Bessel

### 3.5 PHASE LOCKED LOOP (PLL)

O PLL é um circuito com a função de sintonizar a frequência/fase de um oscilador local em relação a frequência/fase de um sinal de referência aplicado à entrada do sistema. Esse circuito é muito importante em sistema de telecomunicações para manter a coerência de fase e frequência entre o transmissor e o receptor.

#### 3.5.1 Estrutura geral do PLL

O PLL é composto por três blocos principais (2), conforme apresentado na Fig. 3.12 :

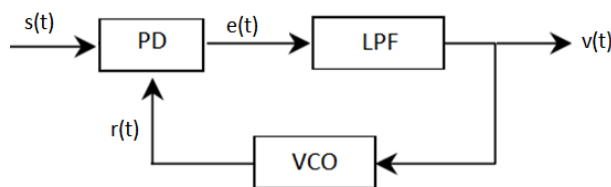


Figura 3.12: Diagrama de Blocos básicos de um PLL (2)

O *Phase Detector* (PD), também encontrado na literatura com a nomenclatura de comparador de fase, é o responsável por gerar o sinal de erro entre a fase do sinal de entrada e a fase do sinal local. A implementação prática desse circuito consiste de um misturador que multiplica o sinal senoidal local pelo sinal recebido, gerando componentes de alta e baixa frequências que podem

ser selecionadas por um filtro.

Seja  $s(t)$  o sinal de entrada conforme:

$$s(t) = A_c \sin(2\pi f_c t + \phi_1(t)) \quad (3.16)$$

Seja  $r(t)$  o sinal gerado localmente pelo VCO.

$$r(t) = A_v \cos(2\pi f_c t + \phi_2(t)) \quad (3.17)$$

Observa-se que o sinal descrito na saída do comparador de fase após uma adaptação trigonométrica de produto de cossenos em soma de senos é dado conforme a equação 3.19.

$$e(t) = s(t)r(t) = A_c A_v \sin(2\pi f_c t + \phi_1(t)) \cos(2\pi f_c t + \phi_2(t)) \quad (3.18)$$

$$e(t) = \frac{1}{2} A_c A_v \sin(4\pi f_c t + \phi_1(t) + \phi_2(t)) + \frac{1}{2} A_c A_v \sin(\phi_1(t) - \phi_2(t)) \quad (3.19)$$

Para facilitar o estudo do PLL e proceder a simplificações pertinentes serão feitas duas considerações básicas sobre o sistema:

a) A componente de maior frequência do sinal ( $2f_c$ ) será atenuada no filtro passa-baixas no próximo bloco, portanto é possível desconsiderá-la, restando o resultado a seguir:

$$e(t) = k_d \sin(\phi_1(t) - \phi_2(t)) \quad (3.20)$$

onde  $k_d$  é um parâmetro que representa as multiplicações de amplitude e fatores decorrentes das relações trigonométricas.

b) Linearização – O PLL considera como hipótese que a variação  $\phi_1(t) - \phi_2(t)$  é muito pequena de forma a tornar verdadeira a aproximação  $\sin \phi = \phi$ . Dessa forma, o resultado final da saída do comparador dado na equação 3.20 pode ser reescrito como:

$$e(t) = K_d(\phi_1(t) - \phi_2(t)) \quad (3.21)$$

Assim, dadas as considerações observa-se que a saída do comparador é um sinal de erro proporcional a diferença de fase entre o sinal local e o sinal de referência.

O *Low-pass filter* (LPF) é um filtro passa-baixa responsável por eliminar a componentes de alta-frequência resultante do processo de mistura dos sinais, atenuar ruídos provenientes do canal e configurar a resposta dinâmica do sistema. A literatura cita comumente a utilização de filtros proporcionais integrativos para realização dessa função (1). Sendo  $e(t)$  o sinal de entrada no filtro, observa-se em sua saída, um sinal  $e_f(t)$  formado pela composição do efeito de proporção ( $e_p$ ) e

do efeito de integração ( $e_i$ ) resultante de cada ramo do filtro visualizado na figura 3.13, onde é um atraso de 01 amostra. Assim, verifica-se matematicamente que:

$$e_f[t] = e_p[t] + e_i[t] \quad (3.22)$$

$$e_p[t] = k_p e[t] \quad (3.23)$$

$$e_i[t] = e_i[t - 1] + k_i e[t] \quad (3.24)$$

No domínio da frequência, tem-se

$$e(s) = k_p + \frac{k_i}{s} \quad (3.25)$$

Dessa forma, é possível verificar que os parâmetros  $k_p$  e  $k_i$  podem ser utilizados para modificar a resposta dinâmica do sistema. O controle PI foi escolhido por sua simplicidade, além de realizar a filtragem do sinal de maneira praticamente plana até uma determinada frequência denominada  $\omega_n$ , que será definida adiante.

O *Voltage Controlled Oscillator* (VCO) é um dispositivo que consiste em variar a frequência de um oscilador local em função de uma tensão de entrada. Em circuitos digitais é comum o uso de um *Numerically Controlled Oscillator* (NCO), que é a versão digital do VCO, o qual varia a fase de um oscilador local por meio de uma entrada numérica. O NCO é dividido em duas etapas principais, a saber, o acumulador de fase e a *look-up table* (LUT).

O circuito acumulador pode ser visualizado na figura 3.13 e apresenta a seguinte representação matemática, em que  $n$  é o número da amostra:

$$\theta_{NCO}[n] = \sum_{i=0}^{i=n-1} e_f[n] \quad (3.26)$$

A *look-up table* apenas usa o valor da fase como argumento para gerar o sinal senoidal/cossenoidal correspondente, conforme equação abaixo:

$$S_I[n] = \cos(\theta[n]) \quad (3.27)$$

$$S_Q[n] = \sin(\theta[n]) \quad (3.28)$$

Para uso prático, há necessidade apenas do sinal  $s_i[n]$ .

### 3.5.2 Calibração do PLL

O PLL, ao tentar ajustar o sinal de saída conforme a frequência e fase do sinal de entrada, acaba por usar valores médios de medições passadas nos seus filtros o que o torna, de maneira geral, um filtro passa-baixa. Os principais parâmetros a serem determinados para se obter uma resposta dinâmica específica do PLL são o fator de amortecimento ( $\eta$ ) e a frequência natural de oscilação ( $\omega_n$ ).

O PLL consegue se ajustar com mais facilidade à diferença de frequência e fase do sinal de entrada se a frequência deste for inferior a  $\omega_n$ , pois a curva de atenuação do sistema é mais plana nesse intervalo, distorcendo menos o sinal, e também para poder usar a aproximação linear na diferença entre os sinais. Uma definição mais precisa de largura de banda para o PLL (22), considerando o fator de amortecimento e um filtro PI, é dada pela expressão:

$$B_n = \frac{\omega_n}{2} \left( \eta + \frac{1}{4\eta} \right) \quad (3.29)$$

Os ganhos do controlador PI também podem ser derivados a partir de  $B_n$  e  $\eta$  a partir das seguintes relações:

$$k_p = \frac{1}{k_d} \frac{4\eta}{\eta + \frac{1}{4\eta}} \frac{B_n}{F_s} \quad (3.30)$$

$$k_i = \frac{1}{k_d} \frac{4}{\eta + \frac{1}{4\eta}} \frac{B_n^2}{F_s} \quad (3.31)$$

O GRC já utiliza como padrão o valor de  $\eta = 0.707$  de forma que apenas o fator  $B_n$  é ajustável em função da taxa de amostragem. Uma largura de banda pequena realiza um acompanhamento mais lento do sinal de entrada, porém consegue eliminar mais ruído do sinal. Ao contrário, uma largura de banda mais ampla permite um acompanhamento mais rápido das variações do sinal de entrada, porém permite a passagem de mais ruído. Em casos práticos, uma  $B_n$  variando entre 1% e 5% da taxa da amostragem apresenta resultados satisfatórios quanto ao compromisso entre acompanhamento do sinal de entrada e passagem de ruído. A partir da escolha do fator de amortecimento  $\eta$ , do fator de ganho do detetor de fase  $k_d$  e da banda equivalente em função da taxa de amostragem, é possível derivar todos os demais parâmetros da malha de controle.

O GNU Radio possui blocos que implementam um PLL e os parâmetros considerados para a configuração do bloco PLL de interesse são  $B_n$  e  $w_{max}$  e  $w_{min}$ , ou seja, o intervalo de desvio de frequência para aquisição. O objetivo do experimento é variar a frequência do sinal de entrada e verificar o alinhamento realizado pelo circuito de regeneração do sinal, acompanhando os sinais de entrada e saída propriamente ditos bem como o sinal diferença entre estes, conforme apresentado na figura 3.13. Na referida figura, foi inserida uma onda senoidal a um bloco altera a frequência do sinal. Esse sinal modificado, foi inserido em um bloco PLL, cuja saída foi compa-

rada ao sinal original enviado e ao sinal após o bloco de alteração de frequência, além de serem gerados os sinais de erro.

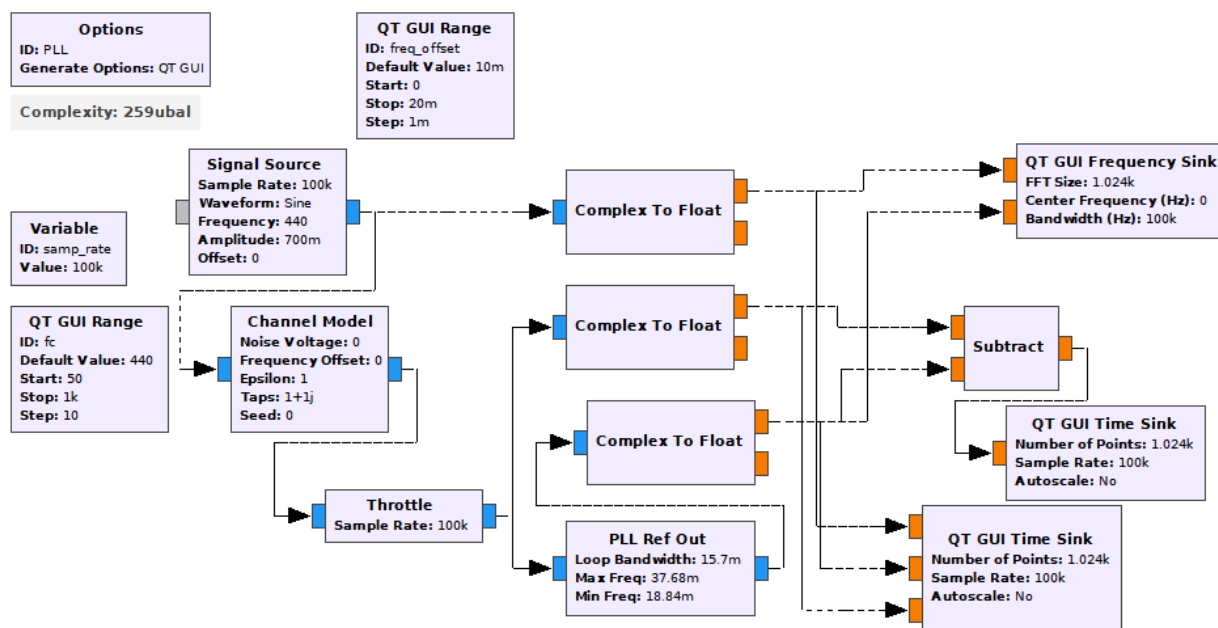


Figura 3.13: PLL em blocos GNU Radio

O primeiro teste consistiu em observar o comportamento do sinal original, o sinal modificado pelo canal na entrada do PLL e o sinal ajustado na saída do PLL. Observou-se que, como esperado, o PLL tenta se ajustar ao sinal modificado pelo meio, conforme verificado na Figura 3.15. O sinal apresentado na figura foi gerado em uma frequência de 660Hz, fora da faixa de acompanhamento do PLL, apenas para que fosse possível verificar, visualmente e didaticamente, o acompanhamento da entrada e saída do PLL

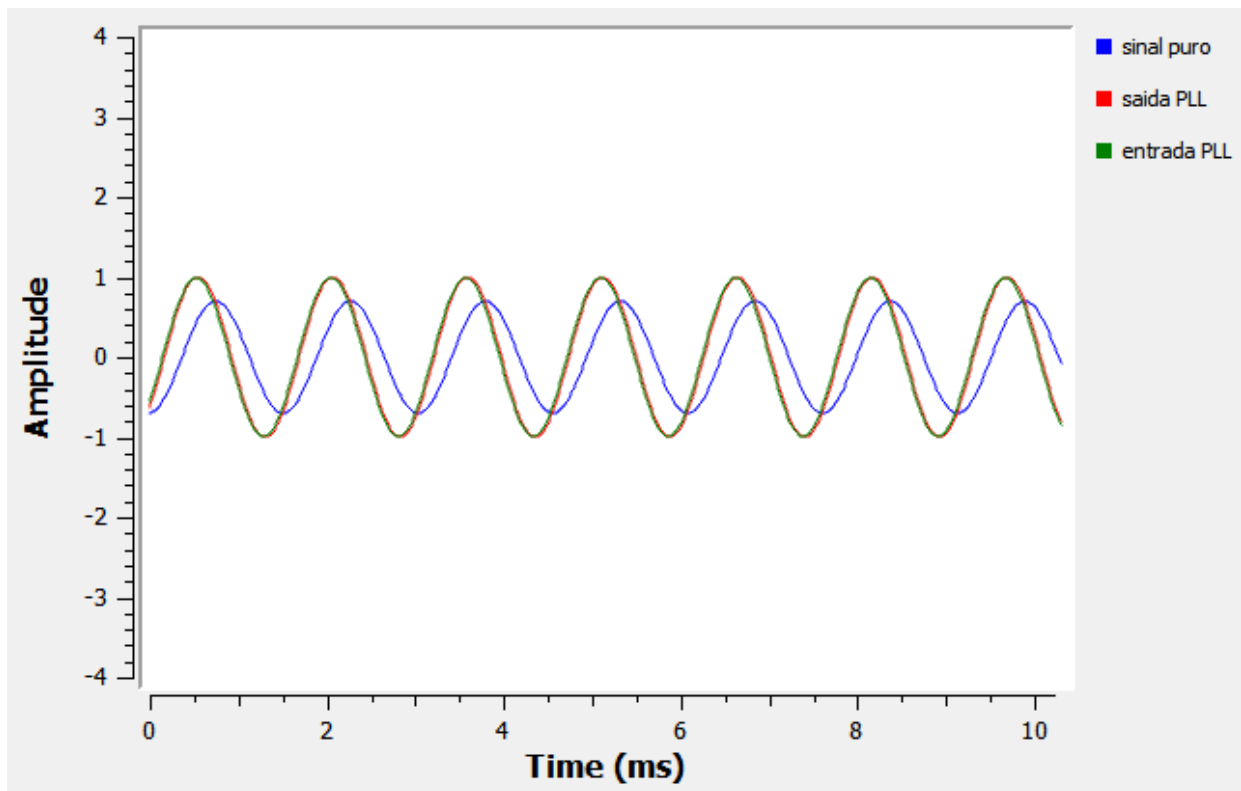


Figura 3.14: Experimento PLL

O segundo teste consistiu em verificar a faixa de acompanhamento do PLL através da variação da frequência do sinal original. Foi utilizado o bloco PLL-freq-detect constante no GNU radio, o qual gera uma portadora na mesma frequência do sinal de entrada, se este sinal estiver com os parâmetros dentro da faixa de acoplamento do circuito PLL.

O circuito PLL foi configurado do seguinte modo:

$$B_w = 440Hz$$

$$W_{min} = 300Hz$$

$$W_{max} = 600Hz$$

Foi observado que o sinal diferença entre a entrada e a referência do PLL obteve variação de módulo e fase conforme a Fig. 3.15, após a variação da portadora no intervalo de 200Hz a 700Hz].

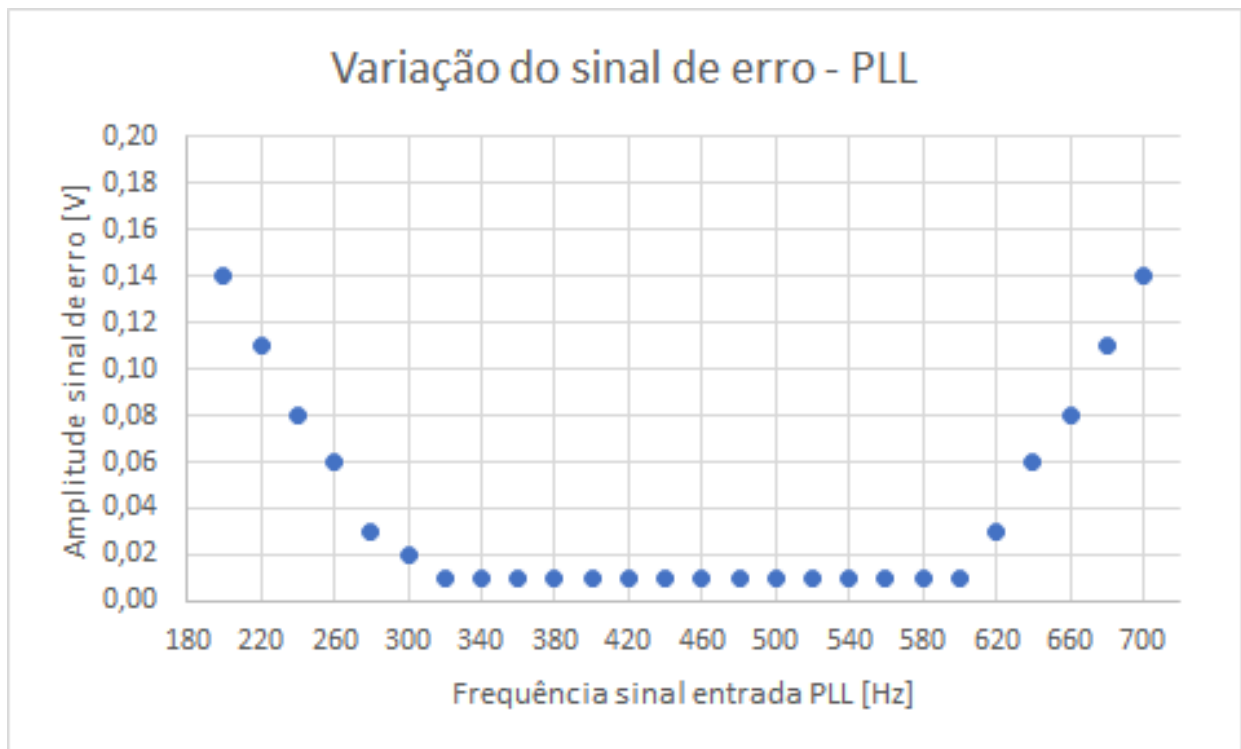


Figura 3.15: Gráfico de erro sinal medido no osciloscópio digital (sink)

Observa-se que a variação da magnitude do sinal de erro se mantém na ordem de 0,01V na faixa de acompanhamento do PLL, verifica-se um aumento na magnitude do sinal do erro quando o sinal utiliza frequência fora da faixa de acompanhamento e constata-se, de forma pratica, que o PLL consegue adequar a frequência do oscilador local a variações de frequência do sinal de original causadas pelo canal. Nesse experimento, o aluno tem contato com uma interface para alterar a frequência de entrada e observar todos os efeitos de ajuste do PLL bem como realizar medições utilizando o osciloscópio digital para tornar mais dinâmica a interação com o ambiente gráfico.

### 3.6 TRANSMISSÃO DIGITAL EM BANDA BASE

A comunicação digital em banda base se caracteriza quando uma informação inerentemente digital é convertida para um formato analógico e transmitida por um meio inerentemente passa-baixa, cujas componentes em frequência do sinal não são transladadas em frequência por meio de modulação. Tais mecanismos normalmente são empregados em meios cabeados que não requeiram irradiação por antenas.

Os dois principais processos de conversão do domínio contínuo (analógico) para o domínio discreto (digital) são denominados amostragem e quantização. O sinal analógico é amostrado e quantizado por um conversor analógico-digital com a função de discretizar o sinal e representá-lo digitalmente por uma sequência de bits para representar uma informação.

A principal utilização da comunicação em banda base ocorre em um cenário em que uma fonte (transmissor), inerentemente digital, converte os bits da mensagem em pulsos que serão detectados no receptor e convertidos novamente em bits. A fig. 3.16 apresenta um diagrama esquemático desse processo.

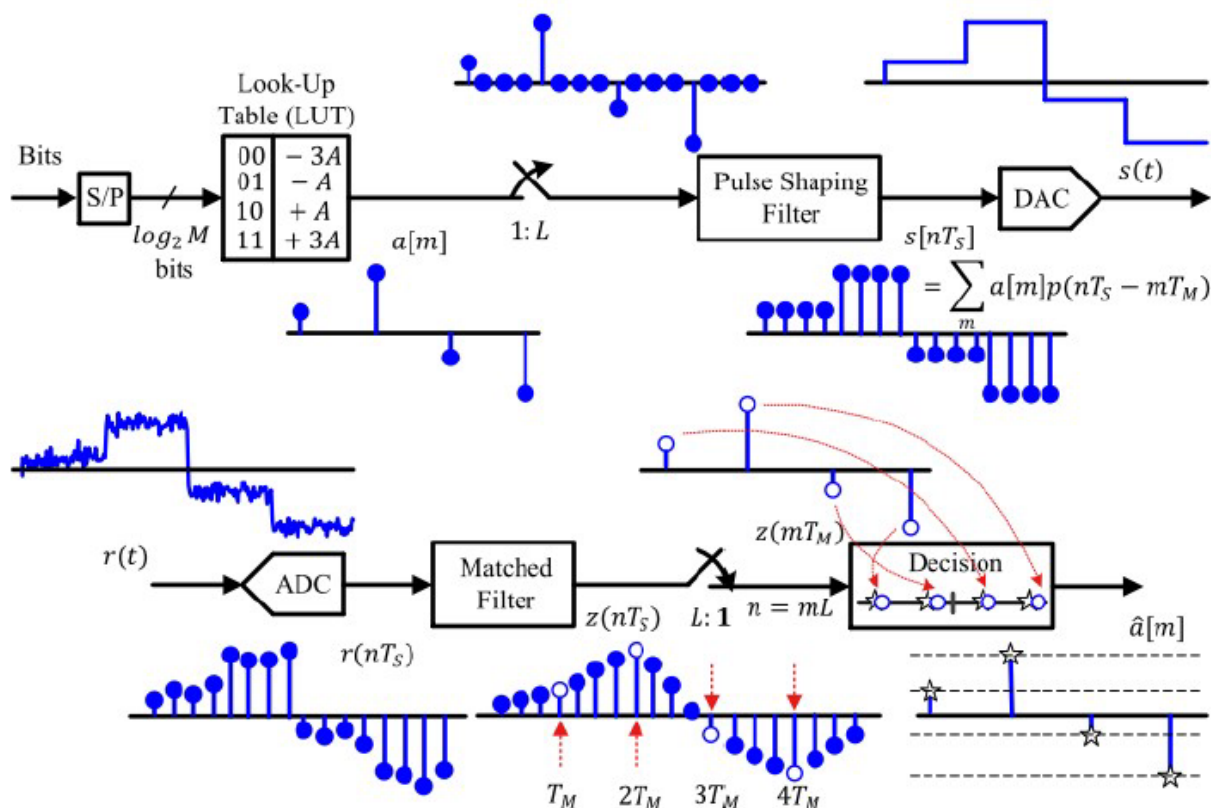


Figura 3.16: Diagrama comunicação PAM (3)

Do ponto de vista físico, observam-se alguns desafios a serem superados quando do projeto de um sistema de comunicação. O sinal digital é convertido em um sinal analógico e sofre distorções decorrentes dos ruídos presentes no canal. Existem formatos específicos de pulsos para serem transmitidos de forma que o receptor consiga ter condições ótimas para identificar qual o bit transmitido tomando por base o pulso recebido. Adicionalmente, o sinal transmitido deve atender a requisitos de limitação em banda, visto que o espectro disponível é bastante regulado e escasso. No aspecto físico, quanto maior a limitação em banda de um sinal, maior será a duração (temporal) do pulso que representa aquele sinal, o que pode levar, em uma situação ideal, à construção de um pulso de duração infinita, irrealizável na prática. Em síntese, os principais aspectos a serem considerados em um sistema de comunicação são:

- construir pulsos e filtros que maximizem a probabilidade de recuperação dos bits em um canal ruidoso
- estabelecer um formato de pulso que possa atender as limitações de banda disponíveis e minimizar o efeito do *aliasing*, o qual será melhor detalhado mais a frente.



### 3.6.1 Detecção ótima - Filtro casado

O grande problema observado no receptor é encontrar a correspondência entre o sinal recebido do meio ruidoso e a sequência de bits transmitida, a fim de se recuperar a informação original. Dessa forma, considerando a Fig. 3.16, seja  $r(t)$  o sinal recebido na entrada do receptor antes do filtro de recepção, conforme a equação:

$$r(t) = a[t]p(t) * h(t) + w(t), 0 \leq t \leq T \quad (3.32)$$

Na equação 3.32,  $p(t)$  é o pulso enviado pelo transmissor,  $h(t)$  é a representação do canal e  $w(t)$  é o ruído do sistema, que para efeitos de simplificação de demonstração, será considerado *Additive White Gaussian Noise* (AWGN). O objetivo é desenvolver um filtro de recepção  $c(t)$  que maximize a relação sinal ruído na saída do filtro receptor exatamente no instante de amostragem, desconsiderando o efeito do canal ( $h(t) = 1$ ). A primeira consideração para facilitar o problema é considerar que o filtro de recepção é linear. Dessa forma, observa-se o seguinte sinal de saída:

$$z(t) = p(t) * h(t) * c(t) + w(t) * c(t) \quad (3.33)$$

Para maximizar a relação sinal ruído conforme

$$\eta = \frac{|p(t) * c(t)|^2}{E[n^2(t)]} \quad (3.34)$$

Da transposição de 3.34 para o domínio da frequência, observa-se que o objetivo é maximizar a expressão abaixo:

$$\eta = \frac{|\int P(f)C(f)e^{j2\pi fT}df|^2}{\frac{N_o}{2} \int |C(f)|^2 df} \quad (3.35)$$

Aplicando a desigualdade de Schwarz na equação 3.35 observa-se que

$$\eta \leq \frac{\int |C(f)|^2 df \int |P(f)|^2 df}{\frac{N_o}{2} \int |C(f)|^2 df} \leq \frac{2}{N_o} \int |G(f)|^2 df \quad (3.36)$$

O valor máximo da desigualdade de Schwarz é atingido na condição:

$$C(f) = kP^*(f)e^{-j2\pi fT} \quad (3.37)$$

No domínio do tempo observa-se o filtro de recepção dado em 3.37 é uma versão invertida do pulso original enviado pelo transmissor.

$$c(t) = kp(T - t) \quad (3.38)$$

No domínio do tempo discreto observa-se que a equação 3.38 pode ser reescrita como na equação 3.39, no tempo discreto, onde  $L$  é o número de amostras tomadas do pulso de transmissão.

$$c(nT_s) = kp(LT_s - nT_s) \quad (3.39)$$

O resultado apresentado em 3.39 é a especificação do filtro que maximiza a relação sinal-ruído para um pulso transmitido em um canal sujeito a interferência AWGN. Essa especificação resolve o primeiro problema relacionado a obter a melhor probabilidade de detecção do símbolo transmitido frente ao ruído, porém não considera as restrições de banda de frequência disponível para a transmissão em um meio real. Para atender essa última condição, há necessidade de avaliar o comportamento espectral do pulso  $p(t)$ , o que será verificado na seção seguinte.

### 3.6.2 Controle espectral - Formatação de Pulso

A modulação por amplitude de pulsos é uma técnica utilizada para converter uma informação digital em um formato analógico que ocupe uma banda de frequência aceitável, dentro das restrições de um sistema de comunicações e que consiga ser, ao mesmo tempo, realizável fisicamente.

A informação digital é representada por bits “0” e “1”, os quais são mapeados para um símbolo caracterizado por um valor que representa um conjunto de bits. Aos valores definidos pelos símbolos são atribuídos um formato de pulso, também chamados de códigos de linha, que representam eletricamente o sinal. Os formatos de pulsos mais conhecidos são os listados a seguir:

- Return to Zero (RZ) - Neste código os bits 0 e 1 são representados por pulsos retangulares de amplitudes opostas, mas que retornam ao valor zero no meio do período de amostragem.
- Non Return to Zero (NRZ) - Neste código os bits 0 e 1 são representados por pulsos retangulares de amplitudes opostas, mas que não retornam ao valor zero antes do período da próxima amostragem. A amplitude do pulso só se modifica quando um novo pulso é gerado no período seguinte.
- Manchester - Na representação Manchester os bits são definidos por um nível alto durante meio ciclo de clock e um nível baixo no ciclo de clock seguinte. Os bits 0 e 1 apresentam inversões de polaridade um em relação ao outro de tal forma que o nível DC médio de um bit é zero.

Normalmente a transmissão em banda base, quando explicada teoricamente, utiliza formatos de pulsos retangulares NRZ para facilitar a compreensão e os cálculos matemáticos. A vantagem desse tipo de simplificação é que não há interferência de um símbolo sobre o outro no domínio do tempo. Entretanto, esse tipo de abordagem possui limitações práticas como o surgimento de um espectro de frequências infinito, de forma que um pulso interfere no espectro de outro. Adicionalmente, a potência do lóbulo secundário do espectro de um pulso retangular não é atenuado o

suficiente para ser desconsiderado frente ao lóbulo principal (a diferença é da ordem de 13dB), o que compromete a banda disponível no canal. Além disso, o filtro casado para um sinal retangular tem uma banda muito grande, o que permite maior passagem de ruído.

A partir do espectro de frequência de um sinal retangular observa-se que quanto menor a duração de um pulso, maior é a largura de banda consumida no espectro. Dessa forma, para obter ganhos de largura de banda, uma solução seria criar um pulso com maior duração de tempo, de forma que cada pulso possa se sobrepor aos outros, a fim de manter uma velocidade razoável de comunicação do sistema, mas que ao mesmo tempo permita a determinação do símbolo transmitido em meio à interferência de um pulso sobre o outro.

Considerando que uma fonte digital emitiu bits que foram transformados em pulsos modulados em amplitude (PAM), observa-se o que o sinal que chega ao receptor é da forma:

$$r(nT_s) = \sum a[i]p[nT_s - i(LT_s)] \quad (3.40)$$

Em que  $L$  é o número de amostras que compõem o pulso,  $LT_s$  é a duração do símbolo ( $T_M$ ) e  $a[i]$  é a amplitude do pulso decorrente da *look up table* e que vai determinar qual símbolo foi transmitido.

Aplicando-se o sinal  $r(nT_s)$  ao filtro casado de recepção equivalente obtém-se o seguinte resultado:

$$c(nT_s) = \sum a[i]p[nT_s - i(LT_s)] * p(-nT_s) = \sum a[i]r_p[nT_s - i(LT_s)] \quad (3.41)$$

Como a saída do filtro casado só interessa no instante de tempo equivalente ao final da duração do símbolo ( $T_M$ ), observa-se que é possível tomar  $n = mL$  na equação 3.41. Nesse processo, as amostras intermediárias são ignoradas e o resultado acontece conforme a expressão:

$$c(mLT_s) = a[0]r_p[0LT_s] + \sum (a[i]r_p[(m - i)LT_s]) \quad (3.42)$$

O segundo termo da expressão 3.41 é o efeito da interferência dos outros pulsos no sinal de interesse e não um ruído do canal. Da expressão observa-se que o filtro ideal normalizado para minimizar a ISI deve satisfazer as seguintes condições:

$$r_p[m] = 1, m = 0 \quad (3.43)$$

$$r_p[m] = 0, m \neq 0 \quad (3.44)$$

A menor largura de banda seria obtida com um pulso de formato retangular na frequência, o que implica que, no tempo, o pulso teria um formato  $\text{sinc}(x)$ , o que é no mínimo muito difícil

de ser executado na prática, uma vez que possui infinitas componentes. Dessa forma, outros formatos de pulso foram adotados para poder promover um equilíbrio adequado entre a ISI e o consumo de largura de banda, sendo que o mais utilizado é o pulso cosseno elevado, que pode ser observado na Fig. 3.17.

$$-0.5(1 - \alpha)R_M \leq F \leq +0.5(1 - \alpha)R_M.$$

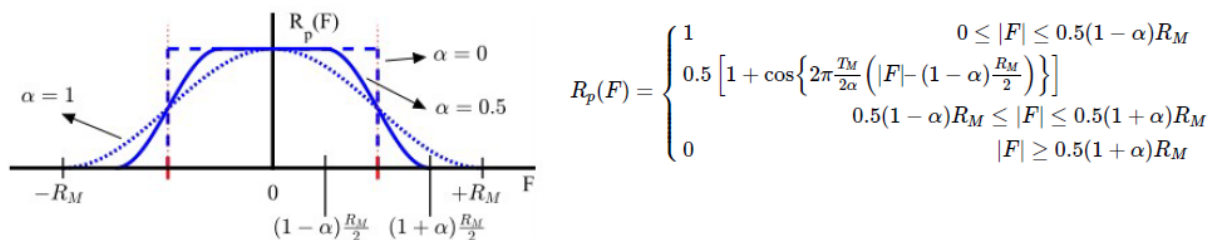


Figura 3.17: Pulso  $p[nT_s - i(LT_s)] * p(-nT_s)$  Cosseno Elevado

É necessário notar que o pulso cosseno elevado definido nas equações acima são resultado da convolução entre o filtro de transmissão e o filtro casado de recepção. Assim, cada filtro  $P(f)$  utilizado na transmissão e seu equivalente casado na recepção devem ter a forma raiz do cosseno elevado conforme demonstrados na equação 3.45, derivada da equação 3.41

$$R_p(f) = P(f).C(f), C(f) = P(f) \implies P(f) = \sqrt{R_p(f)} \quad (3.45)$$

Nesse contexto, buscou-se desenvolver um experimento onde foi utilizada uma técnica de formatação de pulso para transmissão em banda base de um texto. O processo consistiu na aplicação de um trem de impulsos a um filtro passa-baixas configurado como raiz do cosseno elevado. Para a demonstração do processo foi utilizado o diagrama de blocos apresentado na Fig. 3.18.

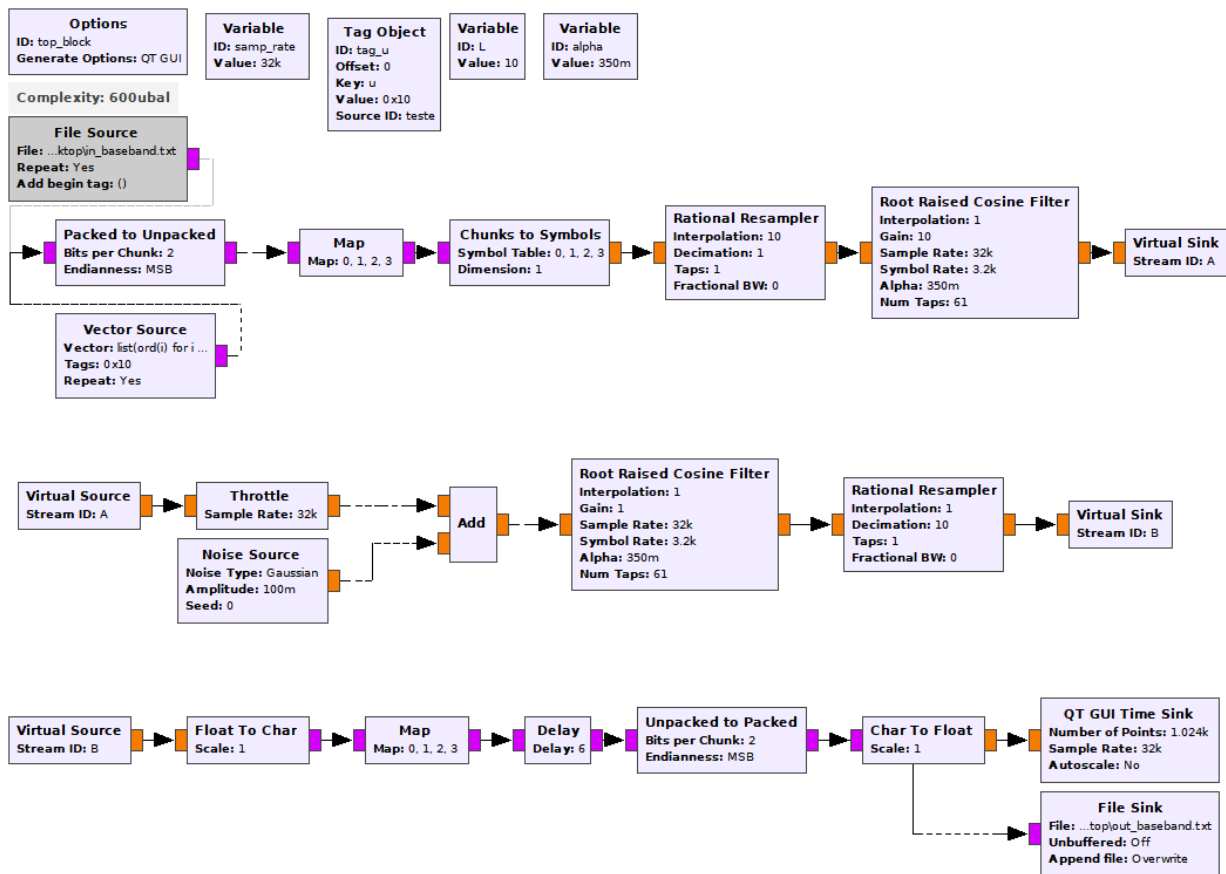


Figura 3.18: Diagrama formatação de pulso

O bloco *vector source* é o responsável por ser a fonte discreta de informação, configurada para gerar o texto "unb\_ppgee", e foi conectada aos blocos *packed to unpacked*, *map* e *chunks to symbols*, que realizam o mapeamento dos bit em símbolos (para este experimento específico, o bloco *map* não é necessário no transmissor). Na sequência há um bloco *rational resampler* responsável por elevar a taxa de amostragem para fins de executar o processo de formatação de pulso no filtro *root raised cosine*. No experimento apresentado, a taxa de transmissão de símbolos selecionada no filtro foi de  $\frac{1}{T_M} = 3200$  símbolos/s, onde  $T_M$  é o período do símbolo. Dada a relação entre a taxa de símbolos e a taxa de amostragem  $samp\_rate = \frac{1}{T_s} = 32ksp_s$ , cada pulso enviado possuía um total de  $f_s T = L = 10$  amostras por símbolo (valor propositalmente configurado no reamostrador para essa condição). O reamostrador gera a primeira amostra com o mesmo valor do sinal de entrada no bloco ( $a[k]$  pertencente ao conjunto 0,1,2,3) e valor zero para as demais 9 amostras do mesmo símbolo, totalizando as 10 interpolações. O resultado é uma aproximação de um trem de impulsos que pode ser aplicado a um filtro com o exato formato do pulso que se deseja para atender as condições de limitação de banda e *aliasing*.

O filtro *rrc* teve o parâmetro *taps* configurado para 61 o que significa que para cada sinal de impulso na entrada do filtro seriam geradas 61 amostras na saída. Como cada pulso é representado por  $T_M = 10$  amostras, o filtro na verdade gera um pulso com duração 6 vezes maior do que o intervalo de tempo disponível para acomodar o pulso, causando sobreposição e a criação do

efeito da ISI. Por outro lado, embora haja interferências entre símbolos adjacentes no domínio do tempo, esta é mais controlada que no caso de um *pulse shaping* retangular e tal situação permite uma limitação em banda do sinal no domínio da frequência.

O bloco *Root Raised Cossine* (RRC) não precisa de interpolador nem repetidor, uma vez que é possível configurar a taxa de símbolos desejada em seus parâmetros. O gráfico de demonstração do conceito para os alunos está apresentado na Fig. 3.19.

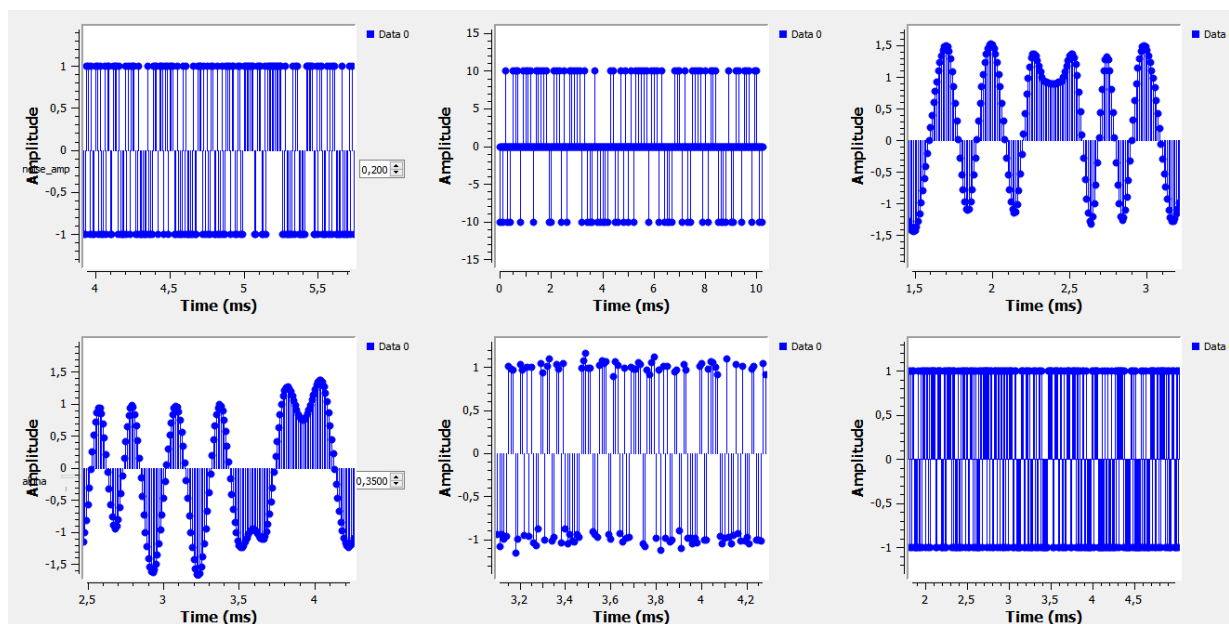


Figura 3.19: Modulação por Amplitude de Pulso com filtro RRC

O aluno pode ajustar o nível de ruído e o nível do parâmetro  $\alpha$  do filtro RRC de forma a verificar o comportamento do sinal transmitido, como ocupação de largura de banda, de forma a tornar mais claro o entendimento do processo de comunicação em banda base. Na prática, é possível observar o efeito da ISI sobre o texto transmitido conforme observado na Fig. 3.20.

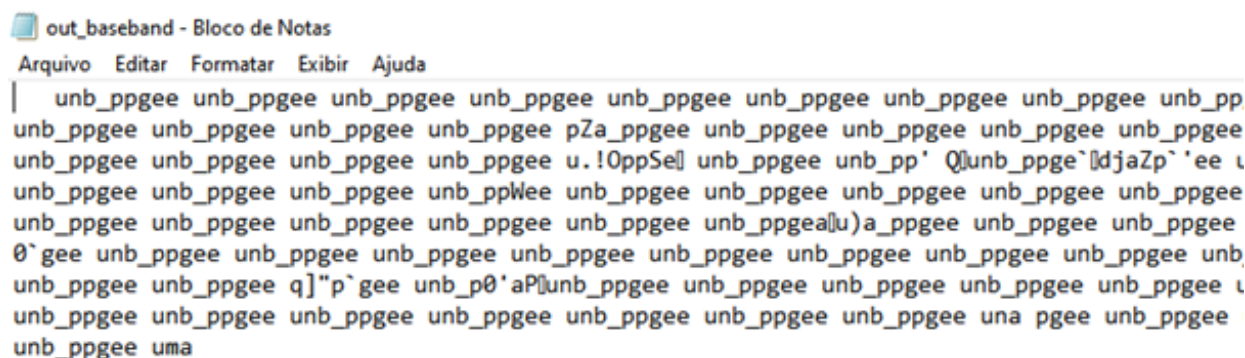


Figura 3.20: Texto corrompido por ISI

Em resumo, este experimento propiciou o amadurecimento dos conceitos de formação de pulso, ajuste de banda para uma transmissão real, ajuste de sincronização de bits devido ao pro-

cesamento da informação e demonstrou, prática e didaticamente, o efeito da ISI sobre o trâmite de uma informação textual.

## 4 DEMODULADOR NEURAL

### 4.1 REDES NEURAS EM TELECOMUNICAÇÕES

Esse trabalho demonstrou em seções anteriores a viabilidade de se utilizar bancadas baseadas em SDR para substituir os equipamentos tradicionais de um laboratório viabilizando, principalmente, a execução de experimentos acessados remotamente, em apoio ao Ensino à Distância. Esse acesso à distância de um laboratório completo também pode tornar mais eficiente os recursos destinados à pesquisa, visto que o mesmo ferramental também pode ser utilizado para pesquisas na área de telecomunicações.

Para dar um foco de pós-graduação ao emprego do laboratório baseado em SDR, foi desenvolvido um experimento que procura inovar a forma de tratar os sistemas atuais de comunicações, buscando empregar novas tecnologias ou novos conceitos ao desenvolvimento de sistemas.

Ao mesmo tempo em que a tecnologia de rádios veio se desenvolvendo, as técnicas de *machine learning* (ML), como as redes neurais, também obtiveram grande avanço em virtude da maior capacidade de processamento dos hardwares, bem como devido à evolução de diversas aplicações voltadas para o aprendizado de máquina como tensorflow (23) e Scikit-Learn (21). A questão colocada neste ponto é a utilização dessas técnicas de aprendizado de máquina aplicadas ao contexto de processamento de sinais, utilizando plataformas SDR como base para os experimentos (24) (25).

No desenvolvimento de sistemas de comunicação, observa-se como problema a necessidade de um amplo estudo de modelos matemáticos e estatísticos do canal a fim de possibilitar a melhor técnica de modulação/demodulação a ser utilizada. Essa abordagem tradicional exige um grande esforço de engenharia para solucionar problemas complexos. Nesse ponto, destaca-se que o emprego de redes neurais apresenta-se como uma forma diferente de demodular o sinal e classificar os bits enviados a partir do sinal recebido, visto que as redes podem ser treinadas com um conjunto de dados (*dataset*) de treinamento construído por amostras reais (26)(27)(28)(29). A vantagem desse processo é que essas amostras já contêm todas as perturbações reais que o sistema pode apresentar, como ruído e distorções não lineares dos componentes, permitindo que o classificador baseado em rede neural se ajuste melhor às imperfeições do meio produzindo um desempenho comparável ou superior em relação aos mecanismos tradicionais baseados em limiar de decisão. Além disso, esse método não tem a necessidade de amplas considerações matemáticas sobre o canal, além de permitir a adaptação dinâmica do demodulador a novos ambientes não previstos inicialmente, por meio de ajustes dos hiperparâmetros da rede.

Utilizando o GRC foi possível gerar um dataset de amostras IQ de um modulador PSK associado a distorções causadas por ruído branco, de forma a possibilitar o treinamento de um classificador baseado em uma rede neural. Assim, foi possível avaliar o desempenho do demo-



dulador utilizando técnicas de ML, indicando que esse tipo de procedimento apresenta vantagens em relação às técnicas tradicionais.

Conceitualmente, o GRC pode emular qualquer tipo de modulação/mecanismo de comunicação, mas a utilização da modulação QPSK facilita a simulação dada a disponibilidade de documentação sobre o assunto. A utilização desse tipo de modulação, em ambiente ruidoso e com interferência AM, constitui um cenário importante, por exemplo, para a avaliação de enlaces digitais de dados entre aeronaves e Centros de Controle, utilizando técnicas de ML, em um ambiente de comunicação aeronáutica, onde o principal tipo de interferência é oriundo das comunicações AM das demais aeronaves e centros de controle que operam nessa faixa de frequência.

O demodulador proposto neste trabalho basicamente é um caso específico de experimento de *machine learning* e para o desenvolvimento completo do experimento, foram utilizados os procedimentos descritos a seguir (30), com objetivo de dar uma visão mais clara de desenvolvimento de projeto de aprendizado de máquina:

- Identificação do contexto geral
- Obtenção dos dados
- Avaliação preliminar dos dados para obter informações
- Preparação dos dados para emprego nos algoritmos
- Seleção e treinamento de modelos
- Ajuste do modelo (caso necessário)
- Apresentação da solução

## 4.2 IDENTIFICAÇÃO DO CONTEXTO GERAL

### 4.2.1 Funcionamento de uma Rede Neural

Uma rede neural é um modelamento matemático para o processamento de informações baseado no funcionamento do cérebro humano, que procura encontrar um resultado de interesse a partir da análise de informações de entrada, processadas por milhares de neurônios conectados por sinapses. No campo da engenharia, os neurônios podem ser representados matematicamente pelos denominados *Perceptrons*, os quais são encadeados em camadas que recebem dados de entrada e entregam um resultado especificado para atingir determinado objetivo. A Fig. 4.1 apresenta o modelo gráfico de um perceptron.

A ferramenta de aprendizado de máquina que será avaliada será uma rede *Multiplayer Perceptron* (MLP) (4), as quais consistem em encadear os Perceptrons em arranjos de forma a criar

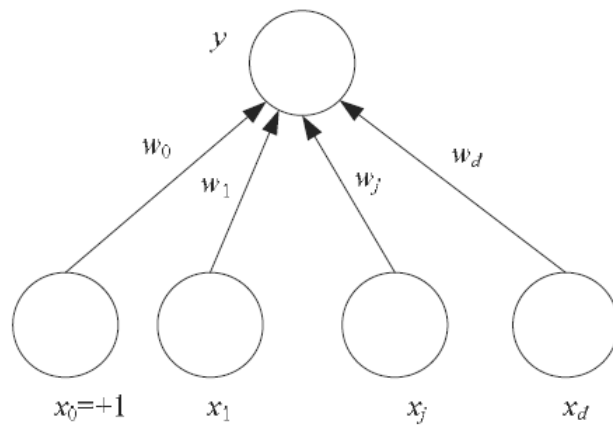


Figura 4.1: Modelo gráfico do Perceptron (4)

sinapses semelhantes ao arranjo de neurônios biológicos. As sinapses são matematicamente representadas por um conjunto de pesos pelos quais os neurônios multiplicam o valor dos dados de entrada, conforme representado na Fig. 4.2.

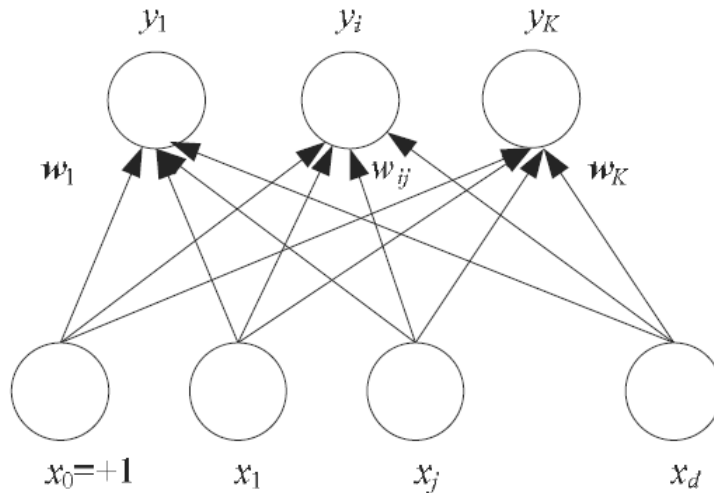


Figura 4.2: Encadeamento de Perceptrons (4)

A rede MLP foi utilizada para a função de classificar um conjunto de dados em classes que são definidas pelo desenvolvedor. Outras técnicas de ML poderiam ter sido utilizadas, mas a rede MLP possui vantagens práticas gerais como economia de energia, ocupação de menos espaço no FPGA caso o código seja portado e normalmente consome menos recursos computacionais comparativamente a outras técnicas. Adicionalmente, a rede atingiu resultados satisfatórios para a demonstração de usabilidade, não havendo necessidade de recorrer a outros mecanismos de aprendizado de máquina. A título de exemplo, pode ser criada uma rede neural que classifica, a partir de um conjunto de parâmetros de um veículo, qual é o tipo ou marca do carro (classe), ou distingui-lo dentre diversas outras classes de veículos como ônibus, caminhão, moto e etc. A partir do exemplo apresentado, observa-se que há necessidade de informar um conjunto de características iniciais, denominadas *features*, como cor, tamanho, motorização, dentre outros, para que a rede decida a qual classe o objeto pertence. Os tipos de classes disponíveis podem ser

passados a priori pelo projetista ou deixar para que a própria rede crie as classes existentes, a partir de algoritmos específicos de identificação de classes. Para que haja a correta classificação, os pesos da rede, também denominados parâmetros, devem ser calibrados para atingir determinado grau de acerto na classificação.

Neste trabalho, utilizou-se um processo de desenvolvimento de rede neural que se baseia em duas fases, a saber, a de treinamento e a de validação. Existem diversas formas de treinamento de uma rede, cujo foco é ajustar os parâmetros da rede a fim de melhorar as classificações. Este experimento se baseou no treinamento supervisionado para a calibração de pesos, em que se utiliza um conjunto de dados com valores de dados de entrada (features) e classes de saída previamente conhecidos. A partir de então, força-se que os pesos da rede variem interativamente para diversos dados de features e classes do conjunto de treinamento até que essa atinja um índice de acertos tolerável, por meio da otimização de uma determinada métrica de qualidade de classificações. Finalizado o treinamento, segue-se a fase de validação em que a acurácia da rede será testada em um segundo conjunto, denominado conjunto de validação, o qual serve para verificar se a rede está classificando as informações de maneira aceitável. Este segundo conjunto também possui features e classes previamente conhecidas, mas na validação não há mais calibração de parâmetros. Utilizam-se as features como entrada da rede para gerar uma saída que será comparada com a saída esperada já determinada no conjunto de validação. Após a avaliação dos resultados, o projetista verifica ou não a necessidade de ajustes na rede.

De forma matemática, descreve-se o funcionamento do perceptron baseado na Fig. 4.1 conforme a seguir:

$$y = \sum_{j=1}^d w_j x_j + w_0 \quad (4.1)$$

A equação 4.1 é uma representação simples de sinapses de perceptrons, porém vários perceptrons podem ser encadeados em diversas camadas, conforme Fig. 4.3, de onde se observa a forma geral de uma rede neural cuja saída  $o_i$  de cada perceptron é dada pela equação 4.2 a seguir:

$$o_i^t = \sum_{h=1}^H v_{ih} z_h^t + v_{i0} \quad (4.2)$$

Da saída de K perceptrons escolhe-se a classe que apresenta o maior resultado dentre as saídas representadas pela função abaixo:

$$y_i^t = \frac{\exp o_i^t}{\sum_k \exp o_k^t} \quad (4.3)$$

A escolha da classe se dá pelo atendimento do critério:  $C_i$  se  $y_i = \max(y_k)$ . A grande vantagem das redes neurais é que elas adaptam seus parâmetros a partir de um conjunto de observações ou dados denominado dataset, o que significa que não há necessidade de tecer hipóteses ou modelos matemáticos sobre o problema para iniciar a classificação em um primeiro momento.

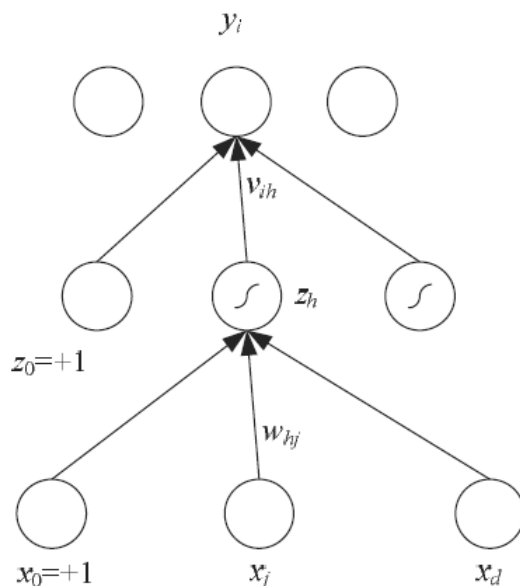


Figura 4.3: Agrupamento de vários neurônios em camadas (4)

Obviamente, essa consideração pode não ser válida em outros tipos de experimentos, mas foi funcional neste trabalho.

A métrica utilizada para avaliar o andamento do treinamento, definida conforme a equação 4.4, será a minimização da entropia cruzada entre o vetor de saída da rede  $y_i^t$ , para cada entrada "t" e um vetor que contém o rótulo  $r_i^t$  de cada classe, o qual assume um valor unitário para a classe correta e zero para as demais. O valor  $y_i^t$  corresponderá à probabilidade de o conjunto de dados  $x^t$  das *features* indicarem a classe  $C_i$ , ou seja,  $y_i^t = P(C_i|x^t)$ .

$$E(W, V|X) = - \sum_t \sum_i r_i^t \log y_i^t \quad (4.4)$$

Dessa forma, a atualização dos parâmetros considerando a entropia dada em 4.4 é descrita pelas expressões abaixo, formuladas a partir do método conhecido como *backpropagation* (4):

$$\Delta v_{ih} = \eta \sum_t (r_i^t - y_i^t) z_h^t \quad (4.5)$$

$$\Delta w_{hj} = \eta \sum_t \left[ \sum_y (r_i^t - y_i^t) v_{ih} \right] z_h^t (1 - z_h^t) x_j^t \quad (4.6)$$

Para duas camadas intermediárias de neurônios de uma rede neural, observa-se a seguinte expressão para os parâmetros:

$$z_{1h} = \text{sigmoid}(w_{1h}^T x), h = 1, \dots, H_1 \quad (4.7)$$

$$z_{2l} = \text{sigmoid}(w_{2l}^T z_1), l = 1, \dots, H_2 \quad (4.8)$$

$$y = v^T z_2 \quad (4.9)$$

Onde  $H_1$  e  $H_2$  são o número de neurônios antes da camada de saída, também chamadas de camadas ocultas, e seus respectivos pesos  $w_{1h}$  e  $w_{2l}$ . A expressão geral pode ser derivada de modo similar às expressões demonstradas acima, acrescentando que o erro de cada camada deve ser propagado pela rede inteira, tornando-se uma tarefa computacionalmente dispendiosa. Durante os experimentos, essas expressões e outros mecanismos de otimização destes cálculos já estão incorporadas no códigos utilizados para realizar o cálculo das redes no Scikit-learn.

#### 4.2.2 Redes neurais e a demodulação de sinais

Os mecanismos de comunicação digital requerem um amplo estudo estatístico do comportamento do símbolo quando submetido aos ruídos e distorções oriundos do canal de comunicação e de não linearidades dos próprios equipamentos transmissores e receptores (31)(32). No caso da modulação/demodulação QPSK, submetida a um ruído no canal, é preciso criar suposições quanto às características do ruído (branco ou colorido), quanto à probabilidade da emissão de determinado símbolo na fonte para, a partir de então, tecer considerações sobre o limiar de decisão que será utilizado para distinguir um símbolo de outro no receptor (1). Uma vez definido o critério de decisão, a demodulação consiste em tratar o sinal e definir qual o símbolo recebido em relação aos limiares de decisão. Entretanto, o ponto negativo deste mecanismo é que o sistema fica limitado a considerações estabelecidas antes do seu emprego real. Essas considerações podem não ser integralmente válidas nos diversos ambientes em que o sistema será empregado, fazendo com que o sistema não atue de forma eficiente em situações específicas.

Como mencionado, os métodos tradicionais não conseguem administrar com eficiência a comunicação em ambientes que não se comportem conforme as hipóteses inicialmente estabelecidas para aquele modelo (33). Neste sentido, propõem-se a utilização de um dataset com símbolos pré-estabelecidos entre o transmissor e o receptor de forma que seja possível o treinamento supervisionado de uma rede neural capaz de classificar os símbolos transmitidos a partir das amostras em fase e quadratura (IQ) do sinal recebido. O treinamento pode ser executado periodicamente de forma a adaptar os parâmetros da rede para as condições reais do ambiente.

A utilização de redes neurais é viável e vantajosa para a determinação dos símbolos recebidos por diversos motivos. O primeiro é que as amostras do sinal real já contemplam todas as distorções presentes no sistema sem a necessidade de criar modelos matemáticos de aproximação para cada efeito físico a que o sinal está submetido, como suposição de ruído, modelos de reflexão, dentre outros (31). O segundo motivo para a utilização das redes neurais é a grande disponibilidade de aplicativos que, em virtude do aumento da capacidade computacional disponível

atualmente, conseguiram viabilizar o treinamento e utilização de redes neurais com resultados bastante satisfatórios para as aplicações de comunicação (34). Além disso, o GNU Radio se baseia amplamente em linguagem Python, fato este que facilita a integração com as ferramentas de *machine learning* utilizadas neste experimento, por utilizarem essa linguagem como padrão. O terceiro ponto relevante é que as aplicações de comunicação são tradicionalmente separadas em blocos com funções específicas, embora nem sempre tal separação seja ótima (35). Essa modularidade permite a inserção de blocos de redes neurais, sem ter que repensar completamente o sistema. Adicionalmente, para adaptar o demodulador para outro tipo de modulação digital basta alterar o conjunto de treinamento, dando flexibilidade ao sistema.

Assim, a metodologia de avaliação do processo de demodulação proposto consistiu em comparar um demodulador tradicional com um demodulador que incorpora técnicas de ML, denominado, neste trabalho, de demodulador neural.

### 4.3 OBTENÇÃO DOS DADOS

Dada a dificuldade de obter dados reais submetidos a todos os efeitos de distorção inserido pelo canal, a aquisição dos dados foi realizada de forma simulada por meio da criação de um diagrama para gerar amostras com imperfeições do meio e criação de um bloco demodulador tradicional QPSK para servir como base de comparação para o demodulador a ser treinado.

#### 4.3.1 Bloco externo - classificador de símbolos tradicional

Para a construção do demodulador QPSK foi utilizado o procedimento descrito nos tutoriais do GNU Radio sobre criação de módulos *out-of-tree* (36). O GNU Radio já possui uma ferramenta ("gr\_modtool") com todos os arquivos necessários para a criação de um bloco customizado e a Fig. 4.4 mostra a estrutura geral de arquivos que podem ser modificados para a criação de um bloco (5).

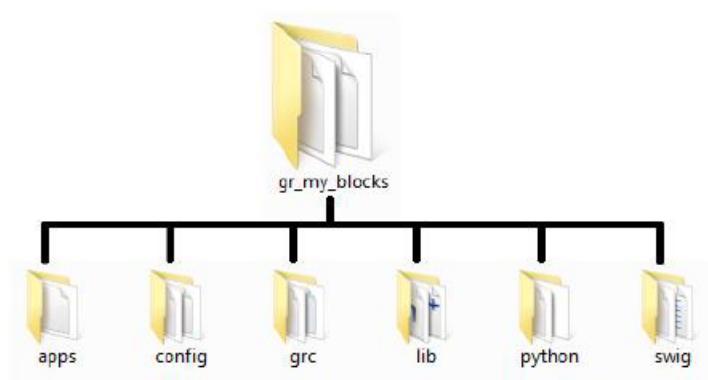


Figura 4.4: Estrutura de arquivos dos blocos do GNU Radio (5)

Com a árvore criada segue-se a execução do comando "gr\_modtool add nome\_do\_bloco", a

partir de então, deve-se escolher o tipo de bloco, a linguagem que o bloco será escrito e os campos dos blocos que serão apresentados ao usuário no ambiente gráfico quando o bloco for incorporado à árvore do GNU Radio. A intenção é criar um bloco que realize a classificação de símbolos baseadas na teoria básica de telecomunicações como limiares de decisão pré-estabelecidos, a fim de compará-lo com o demodulador baseado em rede neural.

Neste sentido, foi criado um bloco externo com o objetivo de ler as amostras em fase e quadratura (IQ) de um sinal QPSK contaminado com ruído branco simulado, interferência AM intencional e reflexões, a fim de descobrir qual foi o símbolo transmitido. O critério de escolha do símbolo foi a comparação do sinal recebido com os limiares de decisão tradicionalmente pré-estabelecidos na constelação de símbolos na presença de ruído AWGN, conforme exemplificado na Fig. 4.5.

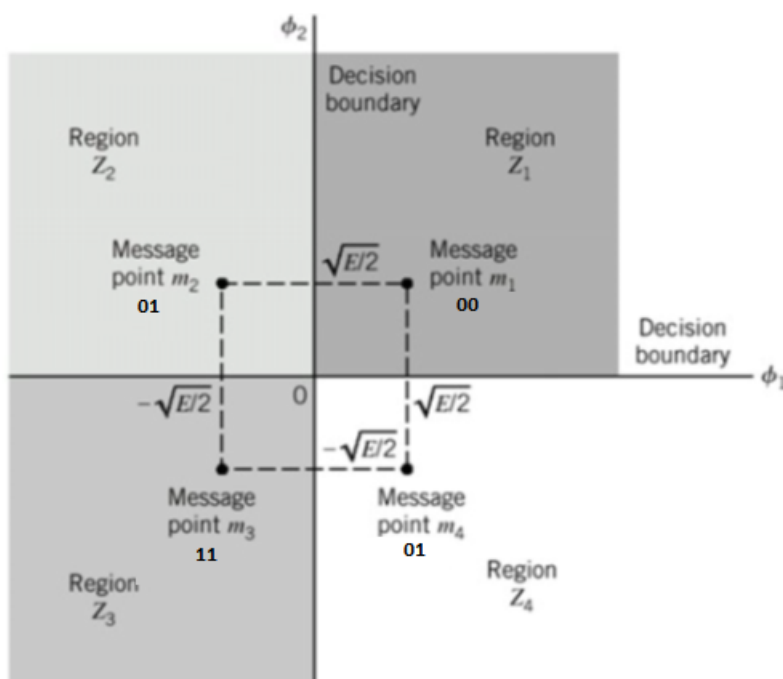


Figura 4.5: Limiar de classificação de símbolos (1)

De posse do *dataset* gerado pelo GRC, toda a análise, treinamento e validação da rede neural foi implementada utilizando o Scikit-Learn e bibliotecas científicas do Python como Numpy e Matplotlib.

### 4.3.2 Criação do Dataset

Os experimentos realizados consistiram em construir simulações de um sistema de comunicação com modulador e demodulador QPSK utilizando blocos do GRC. O ambiente de simulação demandou a configuração de duas classes de parâmetros no GRC, aqueles necessários para realizar a demodulação propriamente dita como sincronização de tempo de amostragem e filtros específicos; e os parâmetros necessários para simular a variação do ambiente de comunicação, como intensidade do ruído, do sinal interferente e dos efeitos de multipercursos/reflexão. Como

os símbolos transmitidos são definidos a priori, foi possível construir um *dataset* associando as amostras IQ recebidas na saída do sistema com os rótulos de cada símbolo já conhecidos. O sistema foi configurado de tal forma que a representação esperada para a constelação de símbolos transmitidos correspondesse aos valores da Tabela 4.1 em um meio ideal.

Tabela 4.1: Mapa de símbolos e amostras

símbolo	valor esperado da amostra	rótulo ( $r^t$ )
0	$0.707+0.707j$	$[1,0,0,0]$
1	$0.707-0.707j$	$[0,1,0,0]$
2	$-0.707+0.707j$	$[0,0,1,0]$
3	$-0.707-0.707j$	$[0,0,0,1]$

O diagrama de blocos desenvolvido no GRC está representado na Figura 4.6, e foi utilizado para gerar um conjunto de dados composto de 160 mil amostras (40 mil para cada símbolo), das quais 140 mil foram utilizadas para treinamento e 20 mil para teste.

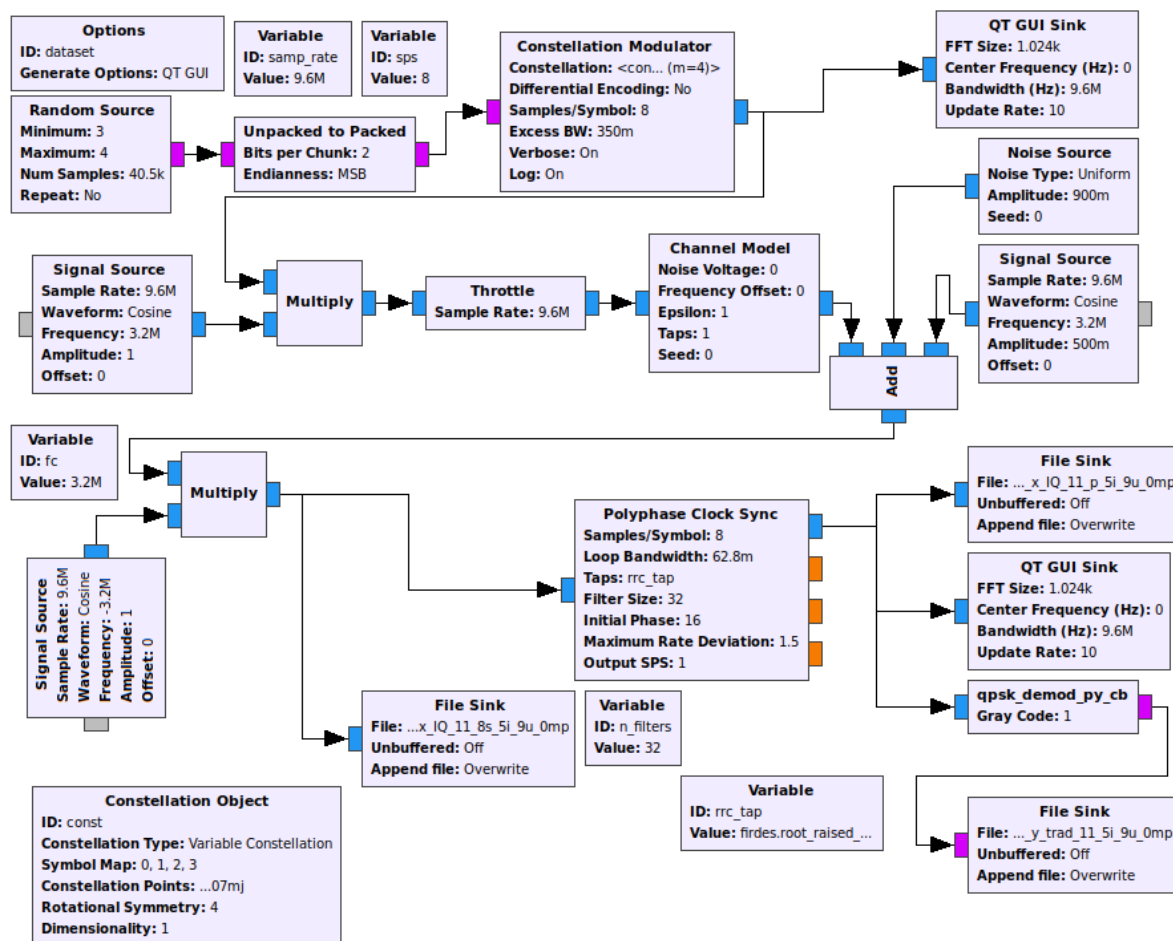


Figura 4.6: Diagrama de ambiente de comunicação para treinamento do classificador baseado em rede neural (4)

Dessa forma, as componentes IQ medidas do sinal recebido já contemplam as distorções do canal. Em um caso não simulado, não haveria a necessidade de realizar modelamentos matemáticos do ruído, uma vez que este já estaria inserido naturalmente nas amostras IQ medidas.



Caberia à rede neural se adaptar à melhor maneira de classificar o símbolo recebido, considerando a minimização da função de entropia cruzada e rótulos dos símbolos, critério utilizado para otimização.

Ao fim da simulação, as amostras IQ complexas (16 bits I e 16 bits Q) foram salvas em um arquivo específico e, como os símbolos gerados são conhecidos a priori, foi possível determinar a relação entre cada amostra IQ ao rótulo do símbolo transmitido. Os arquivos de rótulo e amostras foram carregados em um notebook Jupyter, a fim de realizar o treinamento adequado das redes, conforme o código anexo.

#### 4.4 PREPARAÇÃO DOS DADOS

Não foram necessários maiores ajustes nos dados após a criação do *dataset*. Apenas foi dimensionamento dos vetores com as features para o *shape* adequado para o processamento das funções do scikit-learn. Como as features I e Q possuíam ordem de grandeza equivalentes, não foi necessário realizar procedimentos de normalização para melhorar a convergência dos algoritmos.

#### 4.5 AVALIAÇÃO PRELIMINAR DOS DADOS

Para a avaliação preliminar dos dados foram plotados no plano IxQ os valores das amostras obtidas no GRC a fim de verificar o comportamento da constelação à medida que se modificavam os parâmetros dos sinais interferidores. A variação do sinal AM interferente ocasionou translação em alguns pontos da constelação enquanto a variação do efeito de multipercurso ocasionou rotação na constelação. Além desses, o ruído causou variações em torno do ponto esperado para o sinal. Dessa forma, a rede neural tinha que ser capaz de criar fronteiras para classificar cada símbolo em meio a essas interferências. A figura 4.7 apresenta os efeitos das interferências sob a constelação em um cenário de pouco ruído, interferência externa e reflexão, frente a um cenário de elevado ruído, interferência AM e reflexão.

#### 4.6 SELEÇÃO E TREINAMENTO DE MODELOS

Utilizando as ferramentas disponíveis no Scikit-Learn foi possível realizar o treinamento das redes utilizando variações de parâmetros para pesquisar melhorias nos resultados. Foram realizados treinamentos utilizando-se redes MLP de 02 camadas ocultas (ou *Hidden Layer* - HL) variando entre 01 ou 05 neurônios cada uma e empregando-se o mecanismo de Gradiente Descendente estocástico para a otimização dos parâmetros. A variável que controla a taxa de aprendizado foi mantida constante ( $\eta = 0.001$ ), porém foi empregado o método de gradiente acelerado

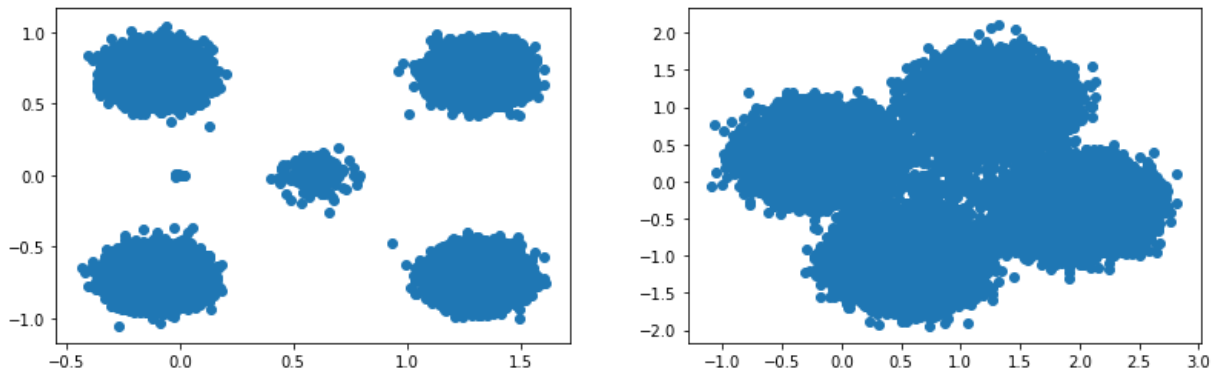


Figura 4.7: Constelação sob baixa interferência versus Constelação sob elevada interferência

de Nesterov para melhorar a convergência do algoritmo (30) (4), mas que não mudam a essência do processo já apresentado. Além disso, procurou-se modificar a função de ativação, entre função logística e tangente hiperbólica, para verificar se tal mudança ocasionaria melhoria significativa nos resultados. Observa-se que o termo “melhoria”, no aspecto desta dissertação, é a diminuição de erros de símbolos medidos a partir da análise da acurácia dos resultados do treinamento e da análise da matriz de confusão das classificações.

O classificador MLP utilizou o parâmetro que controle a taxa de aprendizado constante ( $\eta=0.001$ ), mas empregou o método de nesterov para otimização com momento de 0,9. O número máximo de iterações para esse caso foi configurado para 1000, valor que garantiu a convergência do algoritmo para todos os casos testados.

## 4.7 APRESENTAÇÃO DA SOLUÇÃO

Os resultados observados para a matriz de confusão de cada combinação de neurônios na camada oculta e função de ativação estão apresentados na Fig. 4.8 abaixo, considerando um nível de interferência senoidal AM DSB-SC de amplitude 0,9V e ruído AWGN de amplitude 0,3V.

A matriz de confusão para as camadas ocultas com 05 neurônios não apresentou resultados significativamente diferentes do método tradicional, considerando apenas a mudança na função de ativação. Entre as redes com 01 neurônio em cada camada, observou-se uma piora na matriz de confusão e nas acurácias ao passar da função de ativação tangente hiperbólica para a função logística. A configuração de camadas ocultas com 05 neurônios foi a que apresentou melhor resultado dentre todas as configurações testadas.

Foi possível validar o modelo por meio da utilização do método K-fold (com  $K=4$  subconjuntos). A técnica do K-fold divide o *dataset* original em K subconjuntos, utilizando um conjunto  $k_i$  para teste e o conjunto restante  $\{dataset\} - \{k_i\}$  para treinamento, repetindo o processo para todos os subconjuntos  $i \in \{1, \dots, k\}$  até que todos tenham sido testados. Um valor muito discrepante de acurácia em algum subconjunto do K-fold pode sinalizar a detecção de um viés ou outlier

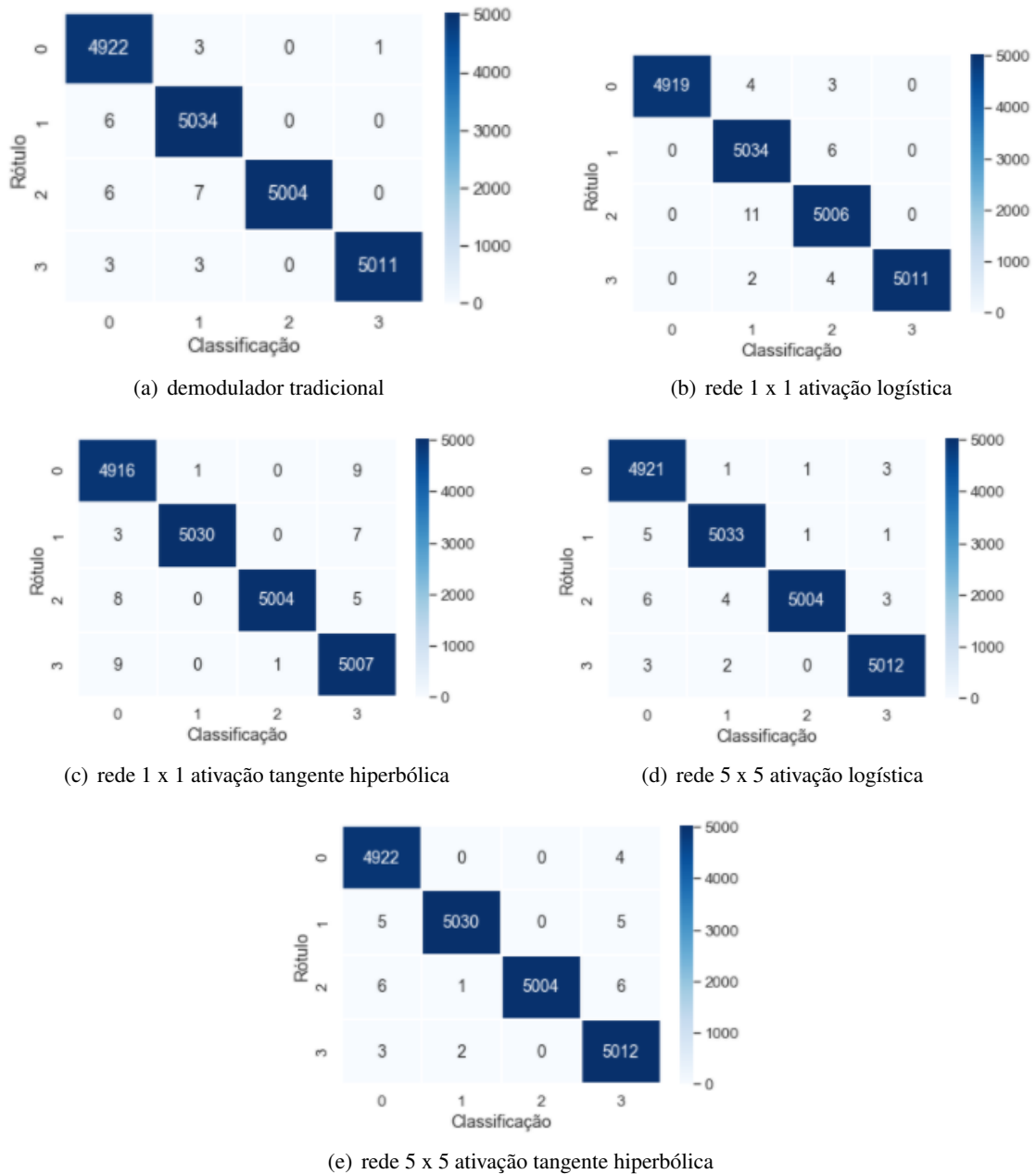


Figura 4.8: Matriz de confusão para diferentes estruturas de rede e funções de ativação

em um conjunto de treinamento específico.

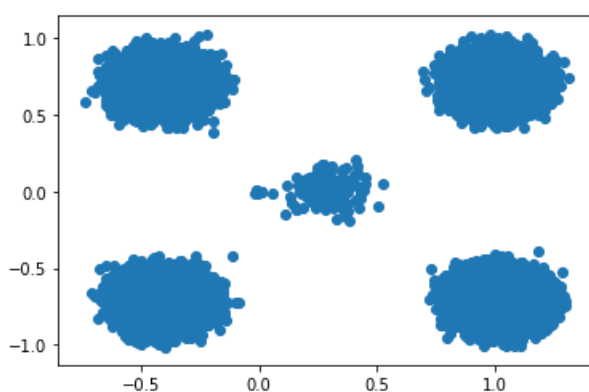
Foi verificado que a acurácia do modelo se manteve acima de 99% na maioria das configurações, conforme apresentado na Tabela 4.2. Na tabela as linhas representam o número de neurônios em cada camada oculta ( $H_1 \times H_2$ ), seguidos da função de ativação utilizada no experimento. Como o K-fold utiliza subconjuntos diferentes de uma mesmo *dataset* para treinamento e teste, a pouca variação da acurácia em todos os testes indica que o modelo se ajustou bem ao conjunto de dados, não sendo detectada a ocorrência de *overfitting*.

Em um segundo experimento, procurou-se realizar uma simulação com o objetivo de avaliar o comportamento do demodulador quanto a variações do ambiente considerando a presença de sinal

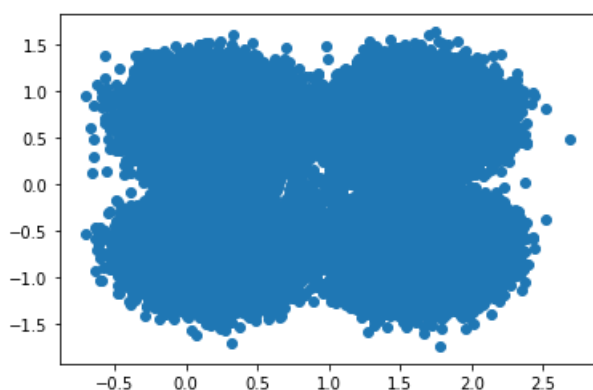
Tabela 4.2: Comparação de acurácias para as configurações da rede

Ativação	Acurácias			
55log	0.999	0.9993	0.99905	0.99885
55tanh	0.9989	0.9994	0.9989	0.9989
11log	0.9988	0.9986	0.9991	0.9988
11tanh	0.9987	0.9989	0.9988	0.9987

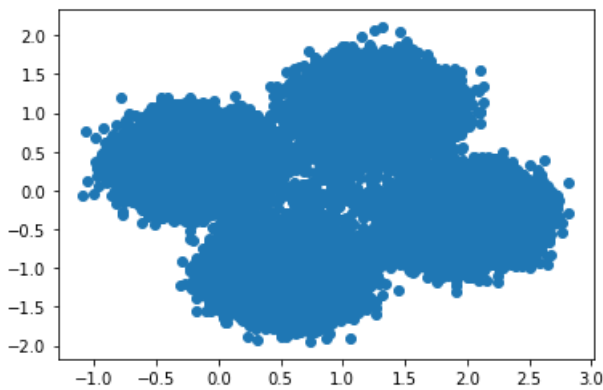
AM DSB-SC interferente em diversas amplitudes ( $A_{inter}$ ), a presença de ruído ( $A_{am}$ ) e a presença de sinal refletido decorrente de multipercursos (MulPath). Nas figuras, mulPath=x corresponde a passar o sinal por um filtro  $1+(0.1*x)^j$ , o que é equivalente a rotacionar a constelação de sianis. As figuras 4.9 e 4.10 mostram os resultados comparativos entre um demodulador tradicional e um demodulador baseado em uma rede neural com 02 camadas ocultas de 05 neurônios cada uma (5x5), utilizando função de ativação tangente hiperbólica.



(a) Constelação  $A_{am}=0.3V$   $A_{inter}=0.3V$



(b) Constelação  $A_{am}=0.9V$   $A_{inter}=0.9mV$



(c) Constelação  $A_{am}=0.9V$   $A_{inter}=0.5mV$  MulPath=5

Figura 4.9: Constelação de símbolos após ruídos, interferências e reflexões

A partir da análise dos resultados observa-se que para ambientes com baixa interferência e ruído, o demodulador neural apresenta resultados bastante similares ao demodulador tradicional. Porém, em ambientes com forte variação de ruído, interferência severa e presença de sinais espúrios decorrentes de multipercursos, observa-se que o demodulador neural mantém sua capacidade de demodulação enquanto que o demodulador tradicional apresenta um número significativo de



(a) tradicional  $A_{am}=0.9V$   $A_{inter}=0.3mV$



(b) rede 5x5 tanh  $A_{am}=0.9V$   $A_{inter}=0.3V$



(c) tradicional  $A_{am}=0.9V$   $A_{inter}=0.9V$



(d) Rede 5x5 tanh  $A_{am}=0.9V$   $A_{inter}=0.9V$



(e) tradicional  $A_{am}=0.9V$   $A_{inter}=0.9V$  MulPath=5



(f) Rede 5x5 tanh  $A_{am}=0.9V$   $A_{inter}=0.9V$  MulPath=5

Figura 4.10: Matriz de confusão comparativa entre o demodulador tradicional e o demodulador baseado em rede neural

erros de símbolo. A diferença se deve à capacidade de adaptação que o demodulador neural possui, uma vez que pode utilizar o *dataset* padrão no novo ambiente e ajustar os parâmetros da rede para operarem de forma ótima naquela situação.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho buscou evidenciar a utilização de ferramentas baseadas em SDR com a finalidade de fornecer suporte laboratorial a uma disciplina introdutória de telecomunicações e demonstrar a possibilidade de utilizar essa mesma estrutura para pesquisas e inovações em telecomunicações. A estrutura proposta atende as aulas, seja de forma presencial, no laboratório, seja por meio de acesso remoto à plataforma GNU Radio, tornando-se uma alternativa relativamente barata ao uso de um laboratório convencional.

Conforme observado pela execução dos diversos experimentos e pela utilização prática de variações destes em sala de aula, ficou demonstrado, de forma local e remota, que esse método de laboratório conseguiu explorar diversos conceitos abordados no currículo de uma matéria de fundamentos de telecomunicações. Conseguiu cobrir pontos iniciais no conceitos de frequências, transformadas de Fourier, filtros, modulações analógicas contínuas e modulações digitais. Dessa forma, conclui-se que, do ponto de vista da abrangência do conteúdo, essa metodologia baseada no emprego de SDR, se mostrou adequada a um currículo de uma disciplina introdutória.

Quanto à possibilidade de emprego do mesmo ferramental aplicado a pesquisas, verificou-se que foi possível desenvolver melhorias em um demodulador QPSK tradicional, inserindo elementos de aprendizado de máquina no sistema, aumentando a complexidade do experimento, associando outros softwares de apoio e inserindo um contexto de comunicação mais realista. O contexto mais complexo envolveu elementos como defasagem de sincronismo, efeito de reflexões, interferências externas, redes neurais e matrizes de confusão, ou seja, assuntos que normalmente não são explorados em um curso introdutório de telecomunicações. Com o ferramental utilizado, foi possível explorar uma abordagem mais complexa ao estudo de comunicações normalmente observada em nível de pós-graduação. Dessa forma, conclui-se que esse ambiente experimental possibilita o emprego de técnicas mais complexas de computação para aplicações de comunicações em nível de pós-graduação.

Observou-se também que tanto os experimentos de nível introdutório quanto os experimentos de nível de pós-graduação puderam ser criados, configurados e executados remotamente a partir de uma outra estação. Essa funcionalidade oferece a vantagem de empregar laboratórios que podem ser acessados pelo aluno remotamente, entregando a este não apenas um contexto de simulação, mas também um contexto realista de transmissão. A configuração do USRP, como exemplo de front-end apresentado neste trabalho, impõe limitações aos experimentos, tornando-os mais próximos da realidade de um projetista de sistemas. A funcionalidade de emprego dos equipamentos à distância, possibilitando a execução de experimentos desse nível, é uma grande vantagem para as instituições EAD e todos os experimentos aqui demonstrados foram testados de forma remota a fim de validar essa funcionalidade.

No momento em que este trabalho foi escrito, a epidemia causada pelo COVID-19 inviabilizou

grande parte dos serviços de ensino e provocou a decretação de quarentenas em diversas cidades, inviabilizando a locomoção de alunos e professores. Muitos dos regimes de aula passaram para o formato EAD e a existência desse mecanismo de laboratório remoto seria uma alternativa bem realista para esse tipo de cenário. Pelo exposto, conclui-se que esse laboratório didático remoto é viável e funcional para testes no ensino regular.

Foram identificadas muitas limitações de documentação sobre o funcionamento do GRC e do USRP, as quais apresentam explicações, em alguns casos bastante superficiais ou pouco claras sobre o funcionamento de determinado bloco, requerendo o desenvolvimento paralelo de outros experimentos apenas para a compreensão do funcionamento de alguns blocos, de forma a fornecer ao projetista um melhor entendimento sobre o sistema. Adicionalmente, como a maioria do código das funções de processamento utilizadas no GRC é baseada em Python e em C++, termina-se por exigir do aluno um conhecimento desse tipo de linguagem de programação para ter maior afinidade com a documentação técnica. Tal condição se apresenta como uma desvantagem para um laboratório que se propõe a ser um suporte para uma disciplina introdutória, onde muitos alunos ainda não têm familiaridade com linguagens de programação.

O contexto de emprego prático também impõe sérias restrições de tempo, visto que muitas tarefas que parecem rudimentares na teoria acabam se mostrando um grande obstáculo quando colocado em uso no USRP, ou seja, em ambiente prático. Ajustes de frequência, adaptação de taxas e sincronismo de tempo de amostragem, normalmente ignorados em simulações convencionais, se mostram um desafio para o funcionamento do sistema de comunicação. A ausência de uma referência concisa que trate do assunto, acaba por levar o projetista a pesquisar em fóruns com informações incompletas, até que todas as limitações sejam superadas e a comunicação possa de fato ocorrer. Em muitos momentos, o professor não disporá de tempo para desenvolver sua aula com antecedência de resolver os problemas observados na construção dos diagramas de telecomunicações, o que ocasiona o atraso da execução das aulas de laboratório ou leva à utilização somente de simulações. Essa limitação diminui o seu impacto sobre as aulas, à medida que a curva de aprendizado do professor sobre como utilizar os equipamentos melhora, o que proporcionará experimentos cada vez mais maduros com o passar do tempo.

Outro ponto que deve ser enfatizado é que embora as simulações tenham sido testadas remotamente, verificou-se que a interação com os softwares é um pouco mais lenta, dependendo da capacidade da rede e da capacidade de processamento dos computadores utilizados. Em momentos de pesquisa por problemas ou em testes de funcionamento de diagramas mais complexos esse atraso pode incomodar o projetista. De qualquer forma, a lentidão não chega a ser um fator inviabilizante para o emprego da plataforma, não sendo crítica em demonstrações de simulações prontas, mas tendo efeitos potencializados durante a montagem dos diagramas.

O emprego de rádios definidos por software nos laboratórios didáticos não apresenta óbices intransponíveis que justifiquem descartar o conceito. Os pontos positivos possibilitam agregação de conhecimento, o desenvolvimento de tarefas multidisciplinares, o favorecimento da pesquisa e permitem um contato do aluno com um ambiente de comunicações mais prático e a um custo

menor que um laboratório de bancada tradicional, além de ser capaz de ser usufruído por uma quantidade maior de alunos, a qualquer momento do dia, quando operado remotamente.



# REFERÊNCIAS BIBLIOGRÁFICAS

- 1 HAYKIN, S.; MOHER, M. *Introdução aos sistemas de comunicação*. [S.l.]: Bookman Editora, 2009.
- 2 QASIM, C. *Phase Locked Loop (PLL) in a Software Defined Radio (SDR)*. 2019. <<https://wirelesspi.com/phase-locked-loop-pll-in-a-software-defined-radio-sdr/>>. Acesso em: 20 jan. 2020.
- 3 CHAUDHARI, Q. *Pulse Amplitude Modulation (PAM)*. 2019. <<https://wirelesspi.com/pulse-amplitude-modulation-pam/>>. Acesso em: 20 jan. 2020.
- 4 ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2009.
- 5 DINIZ, P. C. A. et al. Utilização de rádio definido por software para análise de sinais aplicados ao ensino em engenharia. Universidade Federal de Uberlândia, 2013.
- 6 WIRELESS Innovation Forum - What is SDR. <[https://www.wirelessinnovation.org/what\\_is\\_sdr](https://www.wirelessinnovation.org/what_is_sdr)>. Acesso em: 24 jan. 2020.
- 7 COOLEY, J. E. *A Day in the Life of the RF Spectrum*. Tese (Doutorado) — Massachusetts Institute of Technology, 2005.
- 8 WATSON, S.; LAROUCHE, J.-B. Design of a software defined radio-based tactical dsa network. In: IEEE. *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. [S.l.], 2017. p. 738–743.
- 9 CHAUDHARI, Q. *How to Design Nyquist and Square-Root Nyquist Pulse Shaping Filters*. 2019. <<https://wirelesspi.com/design-nyquist-square-root-nyquist-pulse-shaping-filters/>>. Acesso em: 20 jan. 2020.
- 10 Koski, E.; Linn, C. The jtrs program: software-defined radios as a software product line. In: *10th International Software Product Line Conference (SPLC'06)*. [S.l.: s.n.], 2006. p. 10 pp.–191.
- 11 MITOLA, J. The software radio architecture. *IEEE Communications magazine*, IEEE, v. 33, n. 5, p. 26–38, 1995.
- 12 LINN, Y. An ultra low cost wireless communications laboratory for education and research. *IEEE Transactions on Education*, IEEE, v. 55, n. 2, p. 169–179, 2011.
- 13 PETROVA, M.; ACHTZEHN, A.; MÄHÖNEN, P. System-oriented communications engineering curriculum: teaching design concepts with sdr platforms. *IEEE Communications Magazine*, IEEE, v. 52, n. 5, p. 202–209, 2014.
- 14 XU, Z.; KUI, W.; HEI, X.; CHENG, W. Towards a student-centered lab design for learning principles of communications. In: IEEE. *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2017. p. 475–478.
- 15 WELCH, T. B.; WRIGHT, C. H.; MORROW, M. G. Software defined radio: Inexpensive hardware and software tools. In: IEEE. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2010. p. 2934–2937.
- 16 GNU Radio About. <<https://www.gnuradio.org/about/>>. Acesso em: 20 jan. 2020.
- 17 USRP N210 Software Defined Radio. <<https://www.ettus.com/all-products/UN210-KIT/>>. Acesso em: 20 jan. 2020.

- 18 ANACONDA Why Anaconda. <<https://www.anaconda.com/why-anaconda/>>. Acesso em: 20 jan. 2020.
- 19 GNU Radio About. <<https://www.gnuradio.org/about/>>. Acesso em: 20 jan. 2020.
- 20 THE Jupyter Notebook Introduction. <<https://jupyter-notebook.readthedocs.io/en/stable/notebook.html#introduction>>. Acesso em: 20 jan. 2020.
- 21 WELCOME to scikit-learn. <<https://scikit-learn.org/stable/preface.html>>. Acesso em: 20 jan. 2020.
- 22 BEST, R. E. *Phase locked loops: design, simulation, and applications*. [S.l.]: McGraw-Hill Professional, 2007.
- 23 MS Windows NT Kernel Description. <<http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>> Acesso em: 20 jan. 2020.
- 24 JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. In: ACM. *Proceedings of the 22nd ACM international conference on Multimedia*. [S.l.], 2014. p. 675–678.
- 25 ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- 26 AI and military RF systems. <[mil-embedded.com/articles/ai-military-systems/](http://mil-embedded.com/articles/ai-military-systems/)>. Acesso em: 24 jan. 2020.
- 27 KIRAR, V. P. S. Artificial neural networks for cognitive radio network: a survey. *International Journal of Electronics and Communication Engineering*, v. 9, n. 1, 2015.
- 28 WANG, T.; WEN, C.-K.; WANG, H.; GAO, F.; JIANG, T.; JIN, S. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*, IEEE, v. 14, n. 11, p. 92–111, 2017.
- 29 BKASSINY, M.; LI, Y.; JAYAWEERA, S. K. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials*, IEEE, v. 15, n. 3, p. 1136–1159, 2012.
- 30 GÉRON, A. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. [S.l.]: " O'Reilly Media, Inc.", 2017.
- 31 SCHENK, T. *RF imperfections in high-rate wireless systems: impact and digital compensation*. [S.l.]: Springer Science & Business Media, 2008.
- 32 O'SHEA, T. J.; ROY, T.; CLANCY, T. C. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, IEEE, v. 12, n. 1, p. 168–179, 2018.
- 33 O'SHEA. "an introduction to deep learning for the physical layer". *Journal of Academic Works in Science*, v. 1, p. 1–2, 2015.
- 34 MCCASKEY, M.; FEYDT, A.; CORRIGAN, R.; BHASIN, K.; CHELMINS, D. Implementation of a machine learning based modulation scheme in gnuradio for over-the-air packet communications. IET, 2018.
- 35 ZEHAVID, E. 8-psk trellis codes for a rayleigh channel. *IEEE Transactions on Communications*, IEEE, v. 40, n. 5, p. 873–884, 1992.
- 36 TUTORIAL: Working with GNU Radio in Python. <[https://wiki.gnuradio.org/index.php/Guided\\_Tutorial\\_GNU\\_Radio\\_in\\_Python](https://wiki.gnuradio.org/index.php/Guided_Tutorial_GNU_Radio_in_Python)>. Acesso em: 30 set. 2018.

## APÊNDICES

## Anexo 1 - Códigos

## I.1 TREINAMENTO E TESTE DAS REDES NEURAIS

```
1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # # Códigos para o processo de treinamento e validade de rede neural para
   classificação de símbolos
5
6 # In[ ]:
7
8
9 #Importação de amostras IQ geradas pelo GNU Radio
10 import numpy as np
11 import pandas as pd
12 import os
13
14 symbol_complex_00 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927
   \mod_x_IQ_00_p_9i_9na_10mp', dtype=np.float32)
15 n_elements00 = symbol_complex_00.size
16 n_elements_x00= int(np.divide(n_elements00, 2))
17
18 symbol_complex_01 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927
   \mod_x_IQ_01_p_9i_9na_10mp', dtype=np.float32)
19 n_elements01 = symbol_complex_01.size
20 n_elements_x01= int(np.divide(n_elements01, 2))
21
22 symbol_complex_10 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927
   \mod_x_IQ_10_p_9i_9na_10mp', dtype=np.float32)
23 n_elements10 = symbol_complex_10.size
24 n_elements_x10= int(np.divide(n_elements10, 2))
25
26 symbol_complex_11 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927
   \mod_x_IQ_11_p_9i_9na_10mp', dtype=np.float32)
27 n_elements11 = symbol_complex_11.size
28 n_elements_x11= int(np.divide(n_elements11, 2))
29
30
31 X00=symbol_complex_00.reshape(n_elements_x00,2)
32 X01=symbol_complex_01.reshape(n_elements_x01,2)
33 X10=symbol_complex_10.reshape(n_elements_x10,2)
34 X11=symbol_complex_11.reshape(n_elements_x11,2)
35
36 X = np.concatenate([X00[:40000], X01[:40000], X10[:40000], X11[:40000]])
37
38 n_elements_x00, n_elements_x01, n_elements_x10, n_elements_x11, X, X.shape[0]
39
40
41 # In[ ]:
42
43
```

```

44 #Verificação preliminar dos dados no plano I x Q
45
46 import matplotlib
47 import matplotlib.pyplot as plt
48
49 get_ipython().run_line_magic('matplotlib', 'inline')
50
51 fig,ax= plt.subplots()
52 plt.scatter(X[:,0], X[:,1])
53 plt.show()
54
55 fig3,ax3=plt.subplots()
56 plt.hist(X[:,0], bins
           =[-1.6,-1.4,-1.2,-1.0,-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,0.6,0.8,1.0,1.2,1.4,1.6])
57 plt.show()
58
59
60 # In[ ]:
61
62
63 #criação do arquivo de rótulo
64
65 y00=np.full((X00.shape[0]),fill_value=[0], dtype=np.int8)
66 y01=np.full((X01.shape[0]),fill_value=[1], dtype=np.int8)
67 y10=np.full((X10.shape[0]),fill_value=[2], dtype=np.int8)
68 y11=np.full((X11.shape[0]),fill_value=[3], dtype=np.int8)
69
70 y = np.concatenate([y00[:40000], y01[:40000], y10[:40000], y11[:40000]])
71
72 y_trad_00 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927\
                        mod_y_trad_00_9i_9na_10mp', dtype=np.int8)
73 y_trad_01 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927\
                        mod_y_trad_01_9i_9na_10mp', dtype=np.int8)
74 y_trad_10 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927\
                        mod_y_trad_10_9i_9na_10mp', dtype=np.int8)
75 y_trad_11 = np.fromfile(r'C:\Users\thiago\Desktop\compart_vbox\ds_08030927\
                        mod_y_trad_11_9i_9na_10mp', dtype=np.int8)
76
77 y_t = np.concatenate([y_trad_00[:40000],y_trad_01[:40000],y_trad_10[:40000],
                        y_trad_11[:40000]])
78
79 y, y.size, X.shape, y.shape, y_t, y_t.shape
80
81
82 # In[ ]:
83
84
85 #Permutação dos vetores de entrada e de rótulos
86 shuffle_index=np.random.permutation(160000)
87
88 X=X[shuffle_index]
89 y=y[shuffle_index]

```

```

90 y_t=y_t[shuffle_index]
91
92 X_train, X_test, y_train, y_test, y_t_train, y_t_test = X[:140000],X[140000:],y
    [:140000],y[140000:], y_t[:140000], y_t[140000:]
93
94 X_train, X_test, y_train, y_test
95
96
97 # In[ ]:
98
99
100 #Matriz de confusão para o caso tradicional
101 from sklearn.model_selection import cross_val_predict
102 from sklearn.metrics import confusion_matrix
103
104 import seaborn as sn
105
106 conf_mx_trad=confusion_matrix(y_test,y_t_test)
107 df_cm = pd.DataFrame(conf_mx_trad, range(4), range(4))
108 sn.set(font_scale=1.2)
109 sn.heatmap(df_cm, annot=True, annot_kws={"size": 14}, fmt='g', cmap = plt.get_cmap
    ('Blues'), linewidths=1)
110
111 plt.xlabel('Classificação')
112 plt.ylabel('Rótulo')
113
114 plt.show()
115
116
117 # In[ ]:
118
119
120 #Classificador e matriz de confusão para o caso de rede neural 5x5 com função de
    ativação logística
121 from sklearn.neural_network import MLPClassifier
122
123 mlp_clf_55log=MLPClassifier(solver='sgd', alpha=1e-5, hidden_layer_sizes=(5,5),
    random_state=42, max_iter=1000, activation='logistic')
124 mlp_clf_55log.fit(X_train, y_train)
125 y_test_pred = mlp_clf_55log.predict(X_test)
126 conf_mx_mlp55log=confusion_matrix(y_test, y_test_pred)
127
128
129 # In[ ]:
130
131
132 #Matriz de confusão 5x5 logística
133 df_cm = pd.DataFrame(conf_mx_mlp55log, range(4), range(4))
134 sn.set(font_scale=1.2)
135 sn.heatmap(df_cm, annot=True, annot_kws={"size": 14}, fmt='g', cmap = plt.get_cmap
    ('Blues'), linewidths=1) # font size
136

```

```

137 plt.xlabel('Classificação')
138 plt.ylabel('Rótulo')
139
140 plt.show()
141
142
143 # In[ ]:
144
145
146 #validação cruzada (K-fold =5) para o caso de rede neural 5x5 com função de
    ativação logística
147 from sklearn.model_selection import cross_val_score
148 cross_val_score(mlp_clf, X, y, cv=5, scoring="accuracy")
149
150
151 # In[ ]:
152
153
154 #Classificador e matriz de confusão para o caso de rede neural 1x1 com função de
    ativação tangente hiperbólica
155 mlp_clf_11tanh=MLPClassifier(solver='sgd', alpha=1e-5, hidden_layer_sizes=(1,1),
    random_state=42, max_iter=1000, activation='tanh')
156 mlp_clf_11tanh.fit(X_train, y_train)
157 y_test_pred = mlp_clf_11tanh.predict(X_test)
158 conf_mx_mlp11tanh=confusion_matrix(y_test, y_test_pred)
159 conf_mx_mlp11tanh
160
161
162 # In[ ]:
163
164
165 #Matriz de confusão 1x1 tangente hiperbólica
166 df_cm = pd.DataFrame(conf_mx_mlp11tanh, range(4), range(4))
167 sn.set(font_scale=1.2)
168 sn.heatmap(df_cm, annot=True, annot_kws={"size": 14}, fmt='g', cmap = plt.get_cmap
    ('Blues'), linewidths=1) # font size
169
170 plt.xlabel('Classificação')
171 plt.ylabel('Rótulo')
172
173 plt.show()
174
175
176 # In[ ]:
177
178
179 #validação cruzada (K-fold =5) para o caso de rede neural 1x1 com função de
    ativação tangente hiperbólica
180 cross_val_score(mlp_clf_11tanh, X, y, cv=4, scoring="accuracy")
181
182
183 # In[ ]:

```



```

184
185
186 #Classificador e matriz de confusão para o caso de rede neural 5x5 com função de
      ativação tangente hiperbólica
187 mlp_clf_55tanh=MLPClassifier(solver='sgd', alpha=1e-5, hidden_layer_sizes=(5,5),
      random_state=42, max_iter=1000, activation='tanh')
188 mlp_clf_55tanh.fit(X_train, y_train)
189 y_test_pred = mlp_clf_55tanh.predict(X_test)
190 conf_mx_mlp55tanh=confusion_matrix(y_test, y_test_pred)
191 conf_mx_mlp55tanh
192
193
194 # In[ ]:
195
196
197 #Matriz de confusão 5x5 tangente hiperbólica
198 df_cm = pd.DataFrame(conf_mx_mlp55tanh, range(4), range(4))
199 sn.set(font_scale=1.2)
200 sn.heatmap(df_cm, annot=True, annot_kws={"size": 14}, fmt='g', cmap = plt.get_cmap
      ('Blues'), linewidths=1) # font size
201
202 plt.xlabel('Classificação')
203 plt.ylabel('Rótulo')
204
205 plt.show()
206
207
208 # In[ ]:
209
210
211 #validação cruzada (K-fold =5) para o caso de rede neural 5x5 com função de
      ativação tangente hiperbólica
212 cross_val_score(mlp_clf_55tanh, X, y, cv=5, scoring="accuracy")
213
214
215 # In[ ]:
216
217
218 #Classificador e matriz de confusão para o caso de rede neural 5x5 com função de
      ativação logística
219 mlp_clf_11log=MLPClassifier(solver='sgs', alpha=1e-5, hidden_layer_sizes=(1,1),
      random_state=1, max_iter=1000, activation='logistic')
220 mlp_clf_11log.fit(X_train, y_train)
221 y_test_pred = mlp_clf_11log.predict(X_test)
222 conf_mx_mlp11log=confusion_matrix(y_test, y_test_pred)
223 conf_mx_mlp11log
224
225
226 # In[ ]:
227
228
229 #Matriz de confusão 1x1 logística

```

```

230 df_cm = pd.DataFrame(conf_mx_mlp11log, range(4), range(4))
231 sn.set(font_scale=1.2)
232 sn.heatmap(df_cm, annot=True, annot_kws={"size": 14}, fmt='g', cmap = plt.get_cmap
    ('Blues'), linewidths=1) # font size
233
234 plt.xlabel('Classificação')
235 plt.ylabel('Rótulo')
236
237 plt.show()
238
239
240 # In[ ]:
241
242
243 #validação cruzada (K-fold =5) para o caso de rede neural 1x1 com função de
    ativação logística
244 cross_val_score(mlp_clf_11log, X, y, cv=5, scoring="accuracy")

```

Listing 1: Treinamento e Teste Demodulador Neural