



TRABALHO DE CONCLUSÃO DE CURSO

**ANÁLISE DE DESEMPENHO
DE ARQUITETURAS DE REDES NEURAIS
NO PROCESSO DE EQUALIZAÇÃO DE
CANAIS DE COMUNICAÇÃO DISPERSIVOS E NÃO LINEARES**

Letícia Alves Lima

Brasília, Dezembro de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE CONCLUSÃO DE CURSO

**ANÁLISE DE DESEMPENHO
DE ARQUITETURAS DE REDES NEURAIS
NO PROCESSO DE EQUALIZAÇÃO DE
CANAIS DE COMUNICAÇÃO DISPERSIVOS E NÃO LINEARES**

Letícia Alves Lima

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro Eletricista*

Banca Examinadora

Prof. João Paulo Leite, ENE/UnB
Orientador

Prof. Alexandre Ricardo S. Romariz, ENE/UnB
Examinador interno

Prof. Paulo Henrique P. de Carvalho, ENE/UnB
Examinador interno

Dedicatória

Dedico este trabalho aos meus pais e ao meu irmão. Juntos, vocês formam o melhor e mais completo trio de melhores amigos que alguém poderia ter.

Letícia Alves Lima

Agradecimentos

Agradeço aos meus pais, Luís e Deise, por todo o amor, apoio, carinho e dedicação à minha formação ao longo de todos esses anos, e por sempre acreditarem em mim.

Ao meu irmão, Bruno, por todo o companheirismo, compreensão e inspiração que trouxe para a minha vida desde que veio ao mundo.

Aos amigos Ana Flávia, Antônio, Caio, Fernanda, Gustavo, Luiza e Rodrigo, por tornarem a vida mais leve e pela parceria nos mais diversos momentos e fases da vida.

Aos amigos que a Engenharia Elétrica me deu, João Vítor, Ricardo e Vítor, por todas as conversas e por todos os conhecimentos e risadas compartilhadas.

À ENETEC, por transformar minha experiência de graduação e por contribuir para a minha, assim como para a de tantos outros estudantes do curso, formação pessoal e profissional.

Por fim, agradeço ao professor João Paulo Leite, pelo apoio, dedicação e paciência na orientação deste trabalho.

Letícia Alves Lima

RESUMO

O desenvolvimento das redes neurais surgiu com a ideia de se reproduzir o processamento apresentado por uma das máquinas mais potentes da história da humanidade: o cérebro humano. A ideia era tentar reproduzir o processamento paralelo apresentado pelo órgão mais importante do corpo humano, rompendo com o processamento serial apresentado pelos computadores até então e ampliando sua capacidade e versatilidade. Desde então, diversos tipos de redes neurais e algoritmos para o aprendizado de máquina destas redes foram desenvolvidos, apresentando viabilidade para uma ampla de aplicações; entre elas, dentro das comunicações digitais, o problema de equalização dos canais de comunicação. Neste trabalho, avaliou-se o desempenho do uso de um tipo de rede neural - o *perceptron* de múltiplas camadas - no problema da equalização de canal. Na análise, foram consideradas duas arquiteturas para o *perceptron* de múltiplas camadas: o equalizador *Fully Complex* e o equalizador *Split*, com a utilização do algoritmo da descida do gradiente adaptativo e do algoritmo de propagação resiliente em seus treinamentos. Além das redes utilizadas, analisou-se também o desempenho do equalizador clássico MMSE. Como resultado, as curvas de BER apresentadas pelos equalizadores foram obtidas, considerando-se sua aplicação a quatro canais de comunicação dispersivos nos esquemas de modulação BPSK, QPSK, 8-PSK, 16-PSK, 32-PSK e 64-PSK. A comparação das curvas obtidas elucidou a superioridade em desempenho da equalizador *Fully Complex* .

ABSTRACT

The development of neural networks came with the idea of reproducing the processing presented by one of the most powerful machines in human history: the human brain. The idea was to try to reproduce the parallel processing presented by the most important organ of the human body, breaking with the serial processing presented by the computers until then and increasing its capacity and versatility. Ever since, several types of neural networks and algorithms for machine learning of these networks have been developed, presenting viability for a wide range of applications; among them, in digital communications, the problem of equalization of communication channels. In this work, the performance of the use of one type of neural networks - the multilayer *perceptron* - in the channel equalization problem, was evaluated. In the analysis, two architectures were considered for the multilayer *perceptron*: the *Fully Complex* equalizer and the *Split* equalizer, with the use of the adaptive gradient descent algorithm and the resilient propagation algorithm in its training. In addition to the networks used, the performance of the classic MMSE equalizer was also analyzed. As a result, the BER curves presented by the equalizers were obtained, considering their application to four dispersive communication channels in the BPSK, QPSK, 8-PSK, 16-PSK, 32-PSK and 64-PSK modulation schemes. Comparison of the obtained curves elucidated the superiority in performance of the *Fully Complex* equalizer.

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | CONTEXTUALIZAÇÃO | 1 |
| 1.2 | OBJETIVOS DO TRABALHO | 3 |
| 1.3 | APRESENTAÇÃO DO TRABALHO | 4 |
| 2 | COMUNICAÇÕES DIGITAIS | 6 |
| 2.1 | INTRODUÇÃO | 6 |
| 2.2 | SISTEMAS DE COMUNICAÇÕES DIGITAIS | 6 |
| 2.3 | PROCESSOS DO SISTEMA DE COMUNICAÇÕES DIGITAIS | 7 |
| 2.3.1 | GERAÇÃO E RECEPÇÃO | 7 |
| 2.3.2 | DIGITALIZAÇÃO | 8 |
| 2.3.3 | CODIFICAÇÃO E DECODIFICAÇÃO DE CANAL | 16 |
| 2.3.4 | MODULAÇÃO E DEMODULAÇÃO | 16 |
| 2.3.5 | EQUALIZAÇÃO | 20 |
| 2.3.6 | CONCLUSÃO | 20 |
| 3 | DESEMPENHO DE SISTEMAS DIGITAIS E EQUALIZAÇÃO | 22 |
| 3.1 | INTRODUÇÃO | 22 |
| 3.2 | DESEMPENHO DE SISTEMAS DIGITAIS | 22 |
| 3.2.1 | DEMODULAÇÃO | 23 |
| 3.2.2 | REPRESENTAÇÃO VETORIAL DE SINAIS | 24 |
| 3.2.3 | RELAÇÃO SINAL RUÍDO PARA SISTEMAS DIGITAIS | 27 |
| 3.2.4 | RECEPTOR DE MÁXIMA VEROSSIMILHANÇA | 27 |
| 3.2.5 | PROBABILIDADE DE ERRO | 29 |
| 3.3 | EQUALIZAÇÃO | 30 |
| 3.3.1 | INTERFERÊNCIAS INTERSIMBÓLICAS (ISI) E EFEITOS | 31 |
| 3.3.2 | PRIMEIRO CRITÉRIO DE NYQUIST PARA ISI NULA | 32 |
| 3.3.3 | EQUALIZADORES | 36 |
| 3.4 | CONCLUSÕES | 39 |
| 4 | REDES NEURAIS | 40 |
| 4.1 | INTRODUÇÃO | 40 |
| 4.2 | INTRODUÇÃO ÀS REDES NEURAIS | 40 |

| | | |
|----------|---|-----------|
| 4.2.1 | APLICAÇÕES DO APRENDIZADO DE MÁQUINA | 41 |
| 4.3 | TIPOS DE REDES NEURAIIS | 42 |
| 4.3.1 | <i>Perceptrons</i> DE CAMADA ÚNICA..... | 42 |
| 4.3.2 | <i>Perceptron</i> DE MÚLTIPLAS CAMADAS (MLP) | 42 |
| 4.3.3 | FUNÇÕES RADIAIS DE BASE | 46 |
| 4.3.4 | MÁQUINA DE VETOR DE SUPORTE (SVM)..... | 48 |
| 4.4 | CONCLUSÕES | 53 |
| 5 | UTILIZAÇÃO DE REDES NEURAIIS NO PROCESSO DE EQUALIZAÇÃO..... | 54 |
| 5.1 | INTRODUÇÃO | 54 |
| 5.2 | EQUALIZAÇÃO COM O <i>Perceptron</i> DE MÚLTIPLAS CAMADAS (MLP) | 54 |
| 5.3 | EQUALIZAÇÃO <i>Fully Complex</i> | 55 |
| 5.4 | EQUALIZAÇÃO <i>Split</i> | 58 |
| 5.5 | RESULTADOS..... | 59 |
| 5.5.1 | ANÁLISE COMPARATIVA ENTRE O DESEMPENHO DE REDES NEURAIIS UTILIZANDO <i>perceptrons</i> DE MÚLTIPLAS CAMADAS E O EQUALIZADOR MMSE NO PROCESSO DE EQUALIZAÇÃO | 72 |
| 5.5.2 | ANÁLISE COMPARATIVA ENTRE OS EQUALIZADORES <i>Fully Complex</i> , <i>Split</i> E MMSE NO PROCESSO DE EQUALIZAÇÃO | 77 |
| 5.6 | CONCLUSÕES | 89 |
| 6 | CONCLUSÕES | 90 |
| 6.1 | PROPOSTAS DE CONTINUIDADE DO TRABALHO | 91 |
| | BIBLIOGRAFIA | 92 |

LISTA DE FIGURAS

| | | |
|------|---|----|
| 1.1 | Diagrama de blocos de um equalizador adaptativo. | 2 |
| 1.2 | Representação do processamento do sinal para equalização. | 3 |
| 2.1 | Sistema de comunicação digital: representação em blocos. | 7 |
| 2.2 | Representação do sinal original no domínio do tempo e da frequência. | 8 |
| 2.3 | Trem de impulsos. | 9 |
| 2.4 | Representações do sinal amostrado no domínio do tempo e da frequência. | 10 |
| 2.5 | Filtro passa-baixas | 11 |
| 2.6 | Reconstrução do sinal (adaptado de [1]). | 12 |
| 2.7 | Comparação entre sinal original/sinal quantizado e apresentação do erro de quantização. Adaptado de [2]. | 14 |
| 2.8 | Lei μ e Lei A . Adaptado de [3]. | 15 |
| 2.9 | Codificação do sinal quantizado. Adaptado de [4] | 16 |
| 2.10 | Diferentes tipos de modulação digital. | 17 |
| 2.11 | Exemplos para constelações M-PSK. | 18 |
| 2.12 | Modulação e Demodulação. | 21 |
| 3.1 | Representação da nuvem formada com a inserção do ruído ao redor do símbolo transmitido. Adaptado de [5]. | 25 |
| 3.2 | Probabilidade de erro para cada valor de E_b/N_o | 28 |
| 3.3 | Diagrama de Constelação para o 16-PSK: (a) $E_b/N_o = 10\text{dB}$; (b) $E_b/N_o = 20\text{dB}$; (c) $E_b/N_o = 30\text{dB}$ | 28 |
| 3.4 | Curva de BER para diferentes técnicas de modulação. | 31 |
| 3.5 | Forma de onda sugerida para $ P(f) $. Adaptado de [1]. | 34 |
| 3.6 | Exemplo de filtro adaptativo | 37 |
| 4.1 | Exemplo de <i>perceptron</i> de camada única. | 43 |
| 4.2 | Exemplo de MLP com m entradas, duas camadas ocultas com m neurônios cada e três neurônios de saída. | 44 |
| 4.3 | Exemplo de rede RBF. | 46 |
| 4.4 | Hiperplano ótimo na SVM. | 49 |
| 5.1 | Sistema implementado nas simulações. | 61 |
| 5.2 | Estrutura implementada no MATLAB para o equalizador MLP <i>Fully Complex</i> , considerando uma camada oculta com cinco neurônios. | 61 |

| | | |
|------|---|----|
| 5.3 | Estrutura implementada no MATLAB para o equalizador MLP <i>Split</i> , considerando uma camada oculta com cinco neurônios. | 62 |
| 5.4 | Constelação BPSK - Canal 1: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 63 |
| 5.5 | Constelação BPSK - Canal 2: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 63 |
| 5.6 | Constelação BPSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 64 |
| 5.7 | Constelação BPSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 64 |
| 5.8 | Constelação QPSK - Canal 1: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 65 |
| 5.9 | Constelação QPSK - Canal 2: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 65 |
| 5.10 | Constelação QPSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 66 |
| 5.11 | Constelação QPSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 66 |
| 5.12 | Constelação 8-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 67 |
| 5.13 | Constelação 8-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 67 |
| 5.14 | Constelação 16-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 68 |
| 5.15 | Constelação 16-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 68 |
| 5.16 | Constelação 32-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 69 |
| 5.17 | Constelação 32-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 69 |
| 5.18 | Constelação 64-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 70 |
| 5.19 | Constelação 64-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$ | 70 |
| 5.20 | Curva BER da modulação BPSK para o Canal 1 - Comparativo <i>Fully Complex</i> e MMSE. | 73 |
| 5.21 | Curva BER da modulação BPSK para o Canal 2 - Comparativo <i>Fully Complex</i> e MMSE. | 73 |
| 5.22 | Curva BER da modulação BPSK para o Canal 3 - Comparativo <i>Fully Complex</i> e MMSE. | 74 |
| 5.23 | Curva BER da modulação BPSK para o Canal 4 - Comparativo <i>Fully Complex</i> e MMSE. | 74 |

| | | |
|------|---|----|
| 5.24 | Curva BER da modulação QPSK para o Canal 1 - Comparativo <i>Fully Complex</i> e MMSE. | 75 |
| 5.25 | Curva BER da modulação QPSK para o Canal 2 - Comparativo <i>Fully Complex</i> e MMSE. | 75 |
| 5.26 | Curva BER da modulação QPSK para o Canal 3 - Comparativo <i>Fully Complex</i> e MMSE. | 76 |
| 5.27 | Curva BER da modulação QPSK para o Canal 4 - Comparativo <i>Fully Complex</i> e MMSE. | 76 |
| 5.28 | Curva BER da modulação BPSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 83 |
| 5.29 | Curva BER da modulação BPSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 83 |
| 5.30 | Curva BER da modulação QPSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 84 |
| 5.31 | Curva BER da modulação QPSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 84 |
| 5.32 | Curva BER da modulação 8-PSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 85 |
| 5.33 | Curva BER da modulação 8-PSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 85 |
| 5.34 | Curva BER da modulação 16-PSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 86 |
| 5.35 | Curva BER da modulação 16-PSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 86 |
| 5.36 | Curva BER da modulação 32-PSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 87 |
| 5.37 | Curva BER da modulação 32-PSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 87 |
| 5.38 | Curva BER da modulação 64-PSK para o Canal 3 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 88 |
| 5.39 | Curva BER da modulação 64-PSK para o Canal 4 - Comparativo <i>Split, Fully Complex</i> e MMSE. | 88 |

LISTA DE TABELAS

| | | |
|------|---|----|
| 5.1 | Tabela de Parâmetros BPSK/QPSK- Rede Neural <i>Fully Complex</i> : Descida do Gradiente Adaptativo..... | 72 |
| 5.2 | Tabela de Parâmetros BPSK/QPSK - MMSE: Descida do Gradiente Adaptativo | 72 |
| 5.3 | Tabela de Parâmetros BPSK/QPSK - Rede Neural <i>Fully Complex</i> : Propagação Resiliente | 78 |
| 5.4 | Tabela de Parâmetros BPSK/QPSK - Redes Neurais <i>Split</i> : Propagação Resiliente.... | 78 |
| 5.5 | Tabela de Parâmetros BPSK/QPSK - MMSE | 78 |
| 5.6 | Tabela de Parâmetros 8-PSK - Rede Neural <i>Fully Complex</i> : Propagação Resiliente... | 79 |
| 5.7 | Tabela de Parâmetros 8-PSK - Redes Neurais <i>Split</i> : Propagação Resiliente | 79 |
| 5.8 | Tabela de Parâmetros 8-PSK - MMSE..... | 79 |
| 5.9 | Tabela de Parâmetros 16-PSK - Rede Neural <i>Fully Complex</i> : Propagação Resiliente . | 80 |
| 5.10 | Tabela de Parâmetros 16-PSK - Redes Neurais <i>Split</i> : Propagação Resiliente..... | 80 |
| 5.11 | Tabela de Parâmetros 16-PSK - MMSE | 80 |
| 5.12 | Tabela de Parâmetros 32-PSK - Rede Neural <i>Fully Complex</i> : Propagação Resiliente . | 81 |
| 5.13 | Tabela de Parâmetros 32-PSK - Redes Neurais <i>Split</i> : Propagação Resiliente..... | 81 |
| 5.14 | Tabela de Parâmetros 32-PSK - MMSE | 81 |
| 5.15 | Tabela de Parâmetros 64-PSK - Rede Neural <i>Fully Complex</i> : Propagação Resiliente . | 82 |
| 5.16 | Tabela de Parâmetros 64-PSK - Redes Neurais <i>Split</i> : Propagação Resiliente..... | 82 |
| 5.17 | Tabela de Parâmetros 64-PSK - MMSE | 82 |
| 6.1 | Tabela Comparativa - Descida do Gradiente Adaptativo e Propagação Resiliente..... | 91 |

Abreviações / Glossário

| | |
|------|--|
| ANN | <i>Artificial Neural Networks</i> |
| ASK | <i>Amplitude Shift Keying</i> |
| AWGN | <i>Additive white Gaussian noise</i> |
| BER | <i>Bit Error Rate</i> |
| DF | <i>Decision Feedback</i> |
| FIR | <i>Finite-Impulse Response</i> |
| FSK | <i>Frequency Shift Keying</i> |
| IEEE | <i>Institute of Electrical and Electronics Engineers</i> |
| ITU | <i>International Telecommunications Union</i> |
| ISI | <i>Inter-symbolic Interference</i> |
| LMS | <i>Least Mean Square</i> |
| MAP | <i>Maximum a Posteriori</i> |
| MLP | <i>Multi-layer Perceptron</i> |
| MLSE | <i>Maximum-Likelihood Sequence Estimation</i> |
| MMSE | <i>Minimum Mean Square Error</i> |
| PSK | <i>Phase Shift Keying</i> |
| QAM | <i>Quadrature Amplitude Modulation</i> |
| RBF | <i>Radial Basis Functions</i> |
| RLS | <i>Recursive Least Square</i> |
| SVM | <i>Support Vector Machine</i> |
| VLSI | <i>Very Large Scale Integration</i> |
| ZF | <i>Zero-forcing</i> |

Capítulo 1

Introdução

1.1 Contextualização

O desenho de equalizadores para canais distorcivos é uma área ativa de pesquisa acadêmica e desenvolvimento. Nos últimos anos, a arte de utilização de redes neurais artificiais (ANN) para comunicações *wireless* têm ganho *momentum*. Equalizadores lineares usualmente empregam filtros lineares com estruturas transversais ou em treliça e algoritmos adaptativos como o mínimo quadrado recursivo (RLS) e o mínimo quadrado médio (LMS). No entanto, equalizadores lineares não são a melhor alternativa para equalização de determinados canais.

Com isto, surge o desenvolvimento de redes neurais com o intuito de operarem com equalização não-linear, dada a capacidade de redes neurais de formar limites de decisão não-lineares arbitrários para problemas de classificação complexos, como é o caso da equalização.

A equalização é a técnica em processamento de sinais voltada para o combate aos efeitos da interferência intersimbólica (ISI) em canais dispersivos. Técnicas de equalização padrão são baseadas na modelagem de um canal de comunicação como um filtro adaptativo com função de transferência específica. Com isto, o equalizador atua sobre os parâmetros desta função de transferência para desfazer os efeitos da distorção.

O equalizador baseado em rede neural extrai o sinal desejado do sinal distorcido com a aplicação de um algoritmo adaptativo, que minimiza o erro entre a saída do equalizador e o sinal de teste, como apresentado na Figura 1.1, que apresenta um diagrama de blocos de um equalizador adaptativo.

Para extrair as características de fase do canal a partir da informação recebido no equalizador, é necessário utilizar estatísticas de ordem maior do sinal recebido. A função não-linear da saída do equalizador baseado em rede neural é a responsável por dar origem a estas estatísticas.

O sinal $\bar{s}(n)$ na saída do canal é submetido também ao ruído branco aditivo gaussiano (AWGN) $e(n)$, com média zero e variância σ_n^2 , considerando-se suas representações por envoltória complexa. A Figura 1.2 apresenta os sinais nas saídas de cada bloco dos principais processos relacionados à equalização de canal. Maiores detalhes acerca dos sinais e de suas representações serão desenvol-

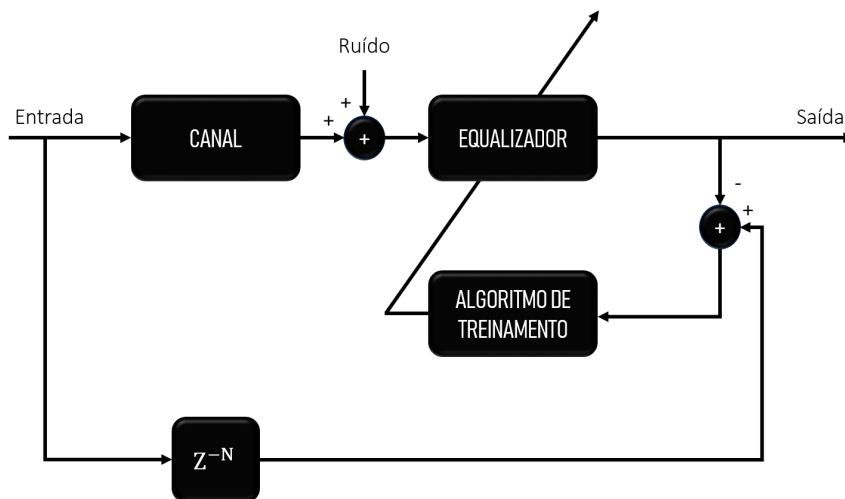


Figura 1.1: Diagrama de blocos de um equalizador adaptativo.

vidos ao longo do trabalho.

A entrada apresentada ao equalizador é, portanto, dada pela equação (1.2)

$$x(n) = \bar{s}(n) + e(n) \quad (1.1)$$

Existem diferentes estruturas de redes neurais e, por este motivo, há diferentes possíveis implementações destes equalizadores e seu histórico dos testes pode ser considerado recente na história das comunicações digitais [6].

Para as implementações de equalizadores baseados em redes neurais utilizando a estrutura do *perceptron* de múltiplas camadas, tem-se que, em 1990, Chen, Gibson e Cown propuseram o uso de redes neurais com estrutura de *perceptron* de múltiplas camadas para equalização adaptativa de canais não-lineares finitos [7], quatro anos depois do estudo de Rumelhart, Hinton e Williams sobre o aprendizado de representações internas por meio da propagação de erro [8].

Em 1992, Benvenuto e Piazza apresentaram um algoritmo de retropropagação complexo para a equalização de canal [9], assim como Georgiou e Koutsougeras [10]. As proposições por eles apresentadas permitiram a formulação de dois tipos de equalizadores de redes neurais principais: *Split* e *Fully Complex*.

Em 1993, Meyer e Pfeiffer apresentaram uma estrutura de equalizador baseado na aplicação de um *perceptron* de múltiplas camadas com *decision feedback* (DF)[11].

Nos anos 2000, 2001 e 2002, Kim e Adali propuseram o uso de redes neurais com retropropagação complexa para o processamento de sinais não-lineares [12] e para o processamento de sinais com envelope constante [13]. Além disto, apresentaram um estudo do desempenho destes algoritmos de retropropagação complexos utilizadas, considerando o uso de diferentes funções de ativação [14].

Em 2003, Ibnkahla e Yuan apresentaram uma rede neural baseada em *perceptron* de múltiplas camadas que reproduzia o comportamento de um equalizador convencional do tipo MLSE, para

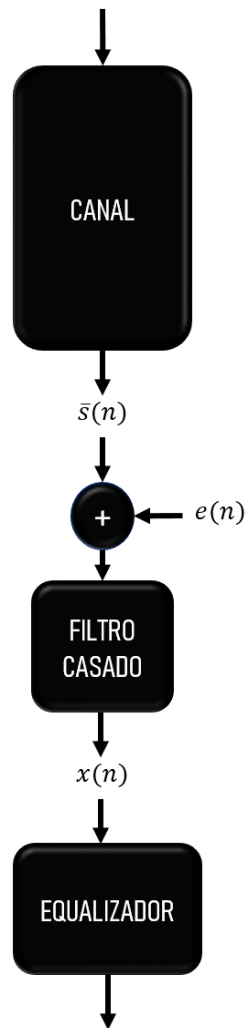


Figura 1.2: Representação do processamento do sinal para equalização.

comunicações via satélite, utilizando o algoritmo da descida do gradiente, com rápida convergência [15] .

Por fim, outro estudo relevante foi publicado em 2004, por Kantsila e Lehtokangas, apresentando um algoritmo de retropropagação resiliente, próprio para sistemas móveis realistas. Este algoritmo apresenta maior rapidez na convergência e, portanto, necessita de um tempo menor de treinamento [16].

Neste trabalho, objetiva-se analisar o desempenho de redes neurais enquanto dispositivos de equalização para contextos similares ao contexto apresentado em [13].

1.2 Objetivos do Trabalho

Este trabalho apresenta dois objetivos principais, contemplados no Capítulo 5 - Utilização de Redes Neurais no Processo de Equalização de Canal:

- Analisar a aplicabilidade do uso de redes neurais no processo de equalização de canal, comparando-as com um equalizador convencional, para modulações do tipo BPSK e QPSK, considerando as distorções provocadas por quatro canais diferentes e de complexidades crescentes: linear real, não-linear real, linear complexo e não-linear complexo, tomando como inspiração inicial os artigos mencionados neste trabalho, que apresentam resultados em algumas das condições das simulações deste trabalho;

- Comparar o desempenho das estruturas de equalizadores baseados em redes neurais *Split* e *Fully Complex* para equalização de canal e determinar qual delas apresenta melhor tratamento da distorção, considerando um canal com distorção linear e outro com distorção não-linear, para as modulações BPSK, QPSK, 8-PSK, 16-PSK, 32-PSK e 64-PSK, com o intuito de aprofundar os resultados apresentados nos artigos deste trabalho e verificar qual destas estruturas baseadas em redes neurais apresenta o melhor desempenho como equalizador.

1.3 Apresentação do Trabalho

Em um sistema de comunicação digital, é possível destacar dois fatores que comprometem sua funcionalidade: o ruído e a distorção; caso estes fatores não estejam dentro de uma faixa controlável para operação, haverá comprometimento da confiabilidade dos acertos sobre as decisões tomadas acerca dos *bits* enviados, isto é, um aumento na taxa de erro de *bits* (BER - *bit* error rate).

Este trabalho aprofunda os conceitos necessários acerca das operações básicas de um sistema de comunicação digital, como apresentado no Capítulo 2 – Comunicações Digitais, assim como das operações necessárias para a manutenção em níveis adequados da confiabilidade das decisões acerca dos *bits* enviados quando sob o efeito das distorções provocadas pelo canal de comunicação.

De forma simples e geral, a tomada de decisão após a passagem pelo sistema receptor é feita com a comparação sobre um limiar de decisão (por exemplo, se o limiar de decisão for a_0 , teremos a seleção para o *bit* 1 caso o sinal detectado seja maior ou igual ao limiar de decisão; caso contrário, a seleção será para o *bit* 0). Com os efeitos da distorção do canal, pode ocorrer a dispersão temporal de um determinado pulso enviado; o alongamento deste pulso, por sua vez, pode ultrapassar o limite da janela de outro símbolo, prejudicando seu processo de amostragem e, por conseguinte, sua identificação.

Este processo de interferência entre janelas de símbolos, que reduz o desempenho do sistema, aumentando sua BER, é denominado interferência intersimbólica e será abordado com maiores detalhes no Capítulo 3 – Desempenho de Sistemas Digitais e Equalização. Para evitar a ocorrência deste fenômeno, no momento de amostragem de um determinado símbolo, a interferência de outros pulsos referentes a outros símbolos deve ser, idealmente, zerada, mantendo apenas a informação relevante ao símbolo em questão.

O dispositivo utilizado para correção da distorção provocada pelo canal é denominado filtro equalizador e o processo que o filtro executa é o processo de equalização de canal, apresentado com maiores detalhes também no Capítulo 3, juntamente a outras métricas de desempenho dos

sistemas.

A busca pelo aperfeiçoamento dos filtros equalizadores se depara com o avanço de outra tecnologia que possui finalidade compatível: as redes neurais. Redes neurais, em suma, tem como finalidade principal a capacidade de operar com classificações, a partir do aprendizado de máquina – um treinamento prévio que indica como uma determinada máquina deve funcionar; neste caso, as redes neurais. No Capítulo 4 – Redes Neurais, tem-se uma breve introdução ao campo das redes neurais e do aprendizado de máquina.

Sendo assim, a aplicabilidade do uso de redes neurais para a operação de equalização apresenta-se como uma alternativa ao uso de equalizadores convencionais para o problema da equalização de canal, trazendo sua funcionalidade, de forma geral, para o campo da classificação de *bits*. Esta possibilidade é explorada com maiores detalhes no Capítulo 5 – Utilização de Redes Neurais no Processo de Equalização.

Por fim, apresentam-se as conclusões acerca dos resultados apresentados no Capítulo 5, assim como as sugestões de projetos futuros a serem implementados a partir do desenvolvimento deste trabalho, no Capítulo 6 - Conclusão.

Capítulo 2

Comunicações Digitais

2.1 Introdução

O Capítulo 1 introduziu as principais estruturas utilizadas neste trabalho, assim como os fins de utilização destas estruturas e os resultados desejados.

Neste capítulo, aprofunda-se o nível de detalhamento de um sistema de comunicação digital, apresentando um modelo para sua estrutura geral com divisão em blocos, assim como as vantagens e desvantagens de sua aplicação, apresentadas na Seção 2.2 – Sistemas de Comunicações Digitais. Em seguida, apresentam-se os processos dos blocos do modelo apresentado na Seção 2.3 – Processos do Sistema de Comunicações Digitais.

Na Seção 2.3.2, aborda-se o processo de conversão do sinal analógico em sinal digital, com a apresentação dos processos de amostragem, quantização e codificação. Na Seção 2.3.3, detalham-se os processos de modulação e demodulação, que serão de fundamental importância para compreensão da relevância dos resultados apresentados neste trabalho; por fim, na Seção 2.3.4, introduzimos o processo de equalização, que será aprofundado no Capítulo 3.

2.2 Sistemas de Comunicações Digitais

Um sistema de comunicação digital é um tipo de sistema projetado para a transmissão de um conjunto finito de informações através de um canal de comunicação. Sua estrutura pode ser apresentada por meio da divisão em blocos, cada um deles representando os processos aos quais a informação deve ser submetida, desde a transmissão até a recepção.

Entre algumas vantagens da aplicação de sistemas de comunicação digitais estão:

- A facilidade de regeneração do sinal digital, com o uso de repetidores regeneradores;
- Maior imunidade ao ruído, decorrentes de técnicas de processamento digital de sinais;
- Baixo custo e ocupação de espaço com implementação em circuito digital VLSI;

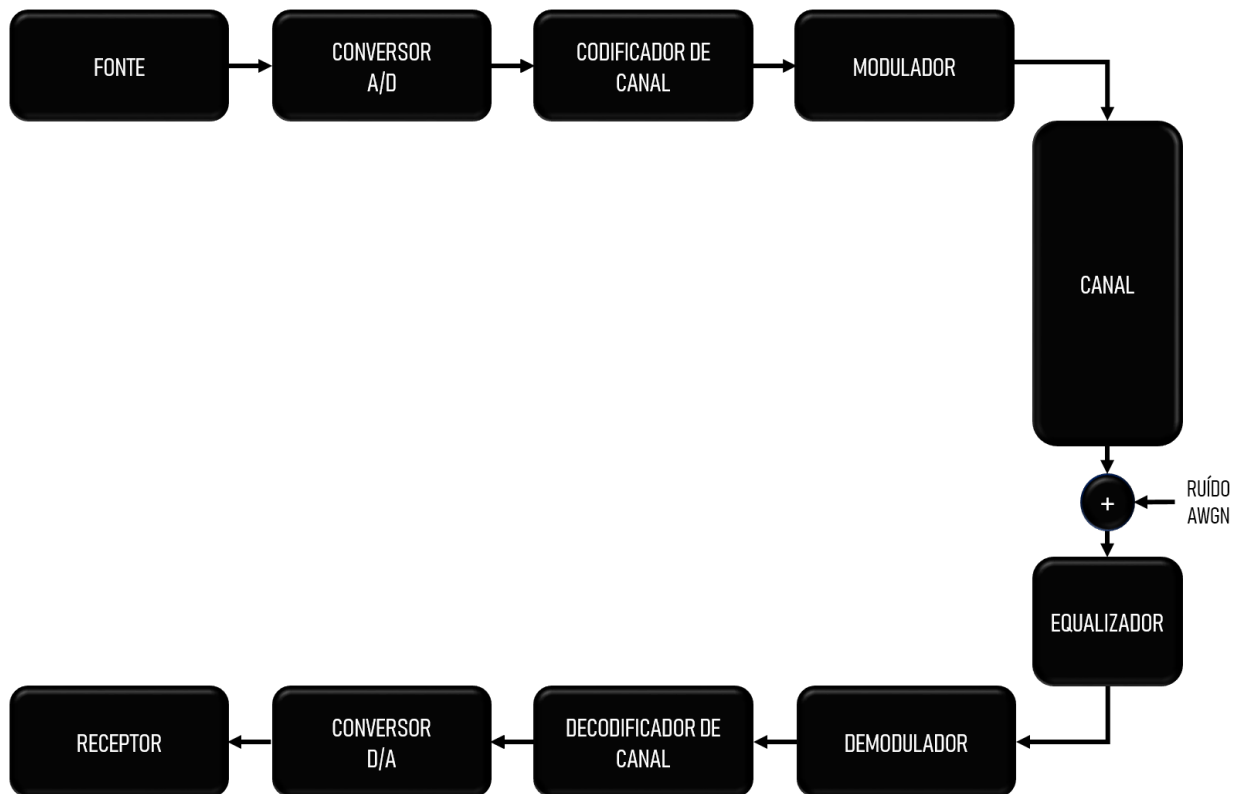


Figura 2.1: Sistema de comunicação digital: representação em blocos.

- Flexibilidade para a transmissão de diferentes tipos de sinais digitais, aplicando-lhes diferentes taxas para sua representação em símbolos binários, podendo serem combinados usando técnicas de multiplexação por divisão de tempo;
- Baixo consumo de potência;
- Permite a aplicação de várias técnicas de múltiplo acesso.

Entre as desvantagens deste tipo de sistema estão o sincronismo necessário para operação adequada (implementação de circuitos de sincronização e, por conseguinte, aumento na complexidade) e a interferência intersimbólica, apresentada devido aos efeitos de distorção de canal), trazendo a necessidade de equalização, como verificado em [1], [17].

A Figura 2.1 ilustra um tipo geral de estrutura de um sistema de comunicação digital.

2.3 Processos do Sistema de Comunicações Digitais

2.3.1 Geração e Recepção

As informações apresentadas pela fonte do sistema de comunicação digital são usualmente símbolos discretos, contando com a adição de um bloco de digitalização (Figura 2.1 – Conversor A/D) caso a natureza do sinal emitido seja analógica.

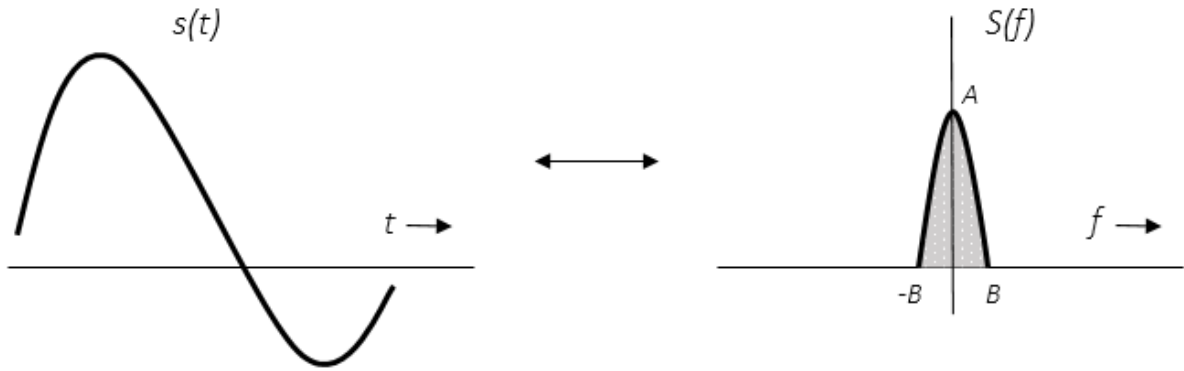


Figura 2.2: Representação do sinal original no domínio do tempo e da frequência.

Após todas as modificações, o receptor recebe o sinal e o reconstitui, aproximando-o do sinal originalmente enviado pela fonte.

2.3.2 Digitalização

Por vezes, a natureza da informação emitida pela fonte é analógica. Neste contexto, surge a necessidade de digitalizar a informação enviada, afim de processá-la em um sistema de comunicação digital.

A digitalização – conversão analógica-digital – pode ser dividida em três etapas: amostragem, quantização e codificação de fonte.

Amostragem

A amostragem é um processo que consiste na retirada de amostras do sinal originalmente emitido pela fonte, para processamento e posterior recuperação do sinal no receptor.

A teoria por trás do processo de amostragem de um sinal tem como base o teorema da amostragem de Nyquist. Este teorema define a frequência de amostragem de acordo com a largura de banda do sinal, afirmando que é possível reconstruir completamente um sinal a partir de suas amostras, se estas forem tomadas a uma taxa de igual ou maior que o dobro da largura de banda.

O teorema será demonstrado a seguir, de acordo com [1].

Para um sinal $s(t)$ cujo espectro é limitado em banda a B Hz, define-se como amostragem de $s(t)$ a retirada de f_s amostras por segundo do sinal. Na Figura 2.2, verifica-se a representação deste sinal.

O processo de amostragem ideal consiste na multiplicação do sinal $s(t)$ por um trem de impulsos $\delta_{T_s}(t)$ como o apresentado na Figura 2.3, em que cada impulso é apresentado a um intervalo de espaçamento T_s segundos – em que T_s é o intervalo de amostragem.

O resultado desta multiplicação é o sinal amostrado $\bar{s}(t)$. O sinal amostrado relaciona-se com o sinal original de acordo com a equação (2.1), em que o n -ésimo impulso é o impulso para o instante de amostragem nT_s .

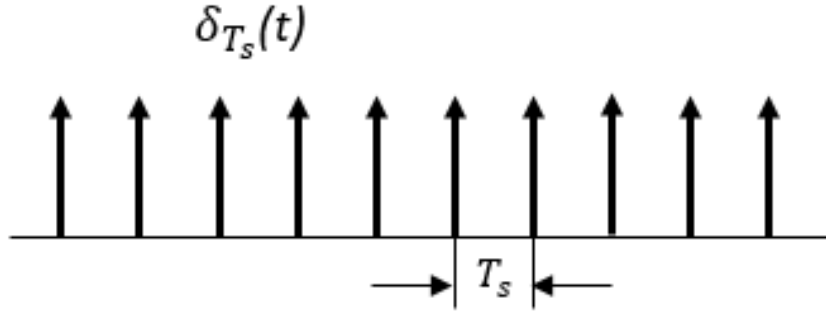


Figura 2.3: Trem de impulsos.

$$\bar{s}(t) = s(t)\delta_{T_s}(t) = \sum_n s(nT_s)\delta(t - nT_s). \quad (2.1)$$

O trem de impulsos utilizado na amostragem do sinal é um sinal periódico com período T_s . Isto permite que ele seja representado como uma série de Fourier exponencial, que assume a forma da equação (2.2)

$$\delta_{T_s}(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{jn2\pi f_s t}. \quad (2.2)$$

Com isto, é possível escrever uma nova expressão para o sinal amostrado na equação (2.3).

$$\bar{s}(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} s(t)e^{jn2\pi f_s t}. \quad (2.3)$$

Aplicando-se a propriedade de translação de frequência na transformada de Fourier do sinal amostrado, tem-se a equação (2.4)

$$\bar{S}(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} G(f - nf_s) \quad (2.4)$$

A Figura 2.4 apresenta as representações do sinal amostrado no domínio do tempo e no domínio da frequência.

Observando-se a equação (2.4), verifica-se que o sinal amostrado é um conjunto de amostras do sinal original espaçadas entre si por uma frequência f_s .

Com estes resultados, é necessário definir as condições necessárias para que o sinal original – agora amostrado – possa ser reconstituído sem distorções.

Para que o sinal reconstituído seja igual ao sinal original, é necessário que o processo de reconstituição seja realizado de forma a não permitir a sobreposição das amostras do sinal amostrado. Como dito anteriormente, a largura de banda do sinal é B e, portanto, para realizarmos a reconsti-

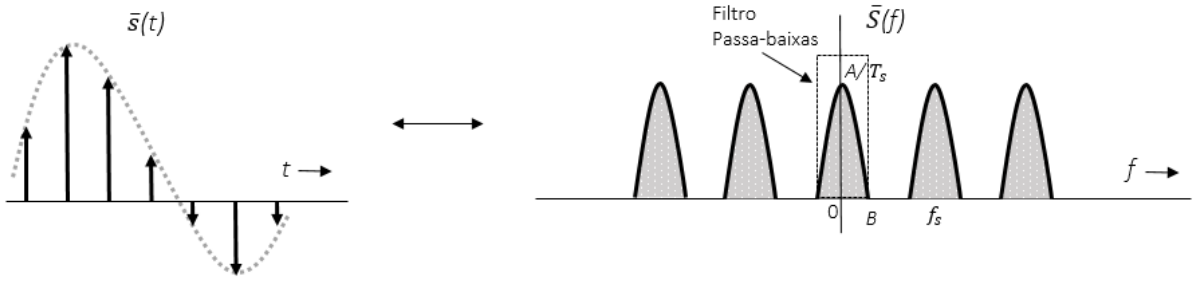


Figura 2.4: Representações do sinal amostrado no domínio do tempo e da frequência.

tuição sem sobreposição das amostras, a frequência de amostragem f_s deve ser, no mínimo, maior que $2B$.

As restrições para a frequência de amostragem e para o tempo de amostragem são representadas pelas expressões (2.5) e (2.6)

$$f_s \geq 2B \quad (2.5)$$

$$T_s \leq \frac{1}{2B} \quad (2.6)$$

Atendendo a estas condições, é possível recuperar o sinal original a partir do sinal amostrado, aplicando a ele um filtro passa-baixas com a largura de banda B do sinal. A partir destes resultados, define-se:

- Taxa de Nyquist: a taxa mínima para amostragem do sinal sem risco de sobreposição das réplicas espectrais (*aliasing*) na recuperação do sinal original;

$$f_s = 2B \quad (2.7)$$

- Intervalo de Nyquist: intervalo mínimo de amostragem;

$$T_s = \frac{1}{2B} \quad (2.8)$$

Como visto, para reconstruir o sinal original, aplicamos um filtro passa-baixas ao sinal amostrado. A função de transferência do filtro encontra-se na equação (2.9):

$$H(f) = T_s \Pi\left(\frac{f}{2B}\right) \quad (2.9)$$

A função retangular apresentada acima – também chamada de *rect* – é dada pelo conjunto de expressões em (2.10)

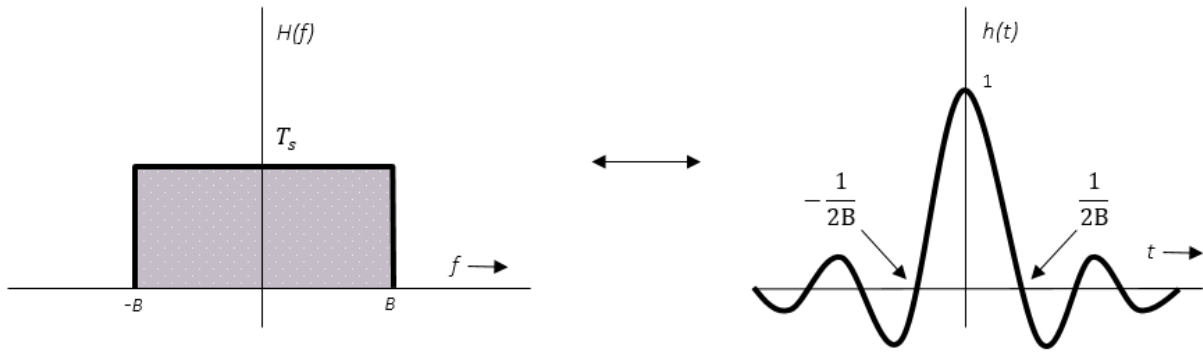


Figura 2.5: Filtro passa-baixas

$$rect(t) = \Pi(t) = \begin{cases} 1, & |t| \leq \frac{1}{2} \\ 0, & c.c \end{cases} \quad (2.10)$$

A transformada inversa da função de transferência do filtro passa-baixas representa a resposta ao impulso do filtro. Dado que uma função retangular no domínio da frequência está sendo utilizada, ao aplicarmos a transformada inversa, obtém-se a expressão (2.11)

$$h(t) = 2BT_s sinc(Bt) \quad (2.11)$$

A função sinc apresentada na equação (2.11) é dada pela expressão (2.12)

$$sinc(t) = \frac{sen(\pi t)}{\pi t} \quad (2.12)$$

De acordo com o resultado para o intervalo de Nyquist, é possível escrever a expressão em (2.13)

$$h(t) = sinc(Bt) \quad (2.13)$$

A Figura 2.5 apresenta as formas de onda da função de transferência do filtro no domínio do tempo e no domínio da frequência.

É interessante observar que a resposta impulsional do filtro passa-baixas vale zero em qualquer outro instante de amostragem, excetuando o instante $t = 0$.

Uma vez que cada amostra obtida é caracterizada por um impulso, a passagem do sinal amostrado pelo filtro-passa baixas produz um conjunto de funções do tipo sinc, cada uma destas funções tendo um valor de pico correspondente ao valor da amostra para um determinado instante de amostragem.

A soma destes pulsos permite a reconstituição do sinal original $s(t)$, como apresentado na Figura 2.6. A qualidade do sinal reconstruído depende de uma seleção adequada dos parâmetros de cada filtro do processo, dentro dos critérios de projeto.

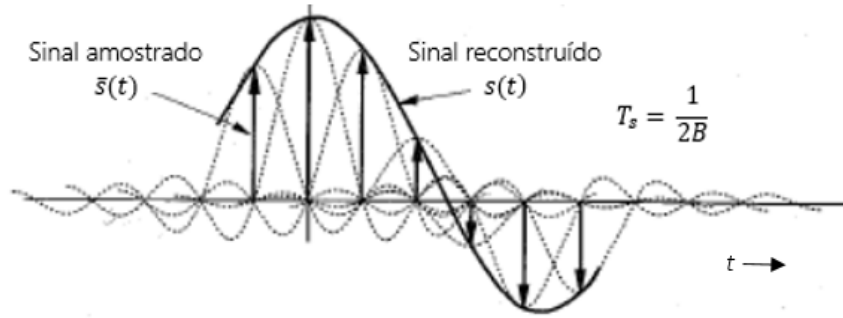


Figura 2.6: Reconstrução do sinal (adaptado de [1]).

Quantização

O processo de quantização consiste em um processo de mapeamento do sinal em um alfabeto finito e discreto, atuando com o arredondamento do sinal amostrado para níveis pré-estabelecidos de tensão que permitem a tomada de decisão acerca da informação enviada.

Dada uma oscilação do sinal em um intervalo definido entre, por exemplo, $[-A; A]$, o quantizador particiona esse intervalo em L níveis de quantização. Cada intervalo, portanto, tem comprimento igual a $2A/L$ e a amostra assume o valor médio deste comprimento de intervalo em que a amostra ocorre.

A aproximação aos níveis pré-estabelecidos por esta modulação gera um erro de quantização. Este erro pode ser minimizado de acordo com a quantidade de níveis L estabelecidos.

Dada uma amostra $s(nT_s)$ do sinal $s(t)$, denotamos $\hat{s}(nT_s)$ como sendo a amostra após a quantização. As expressões representativas dos dois sinais estão nas equações (2.14) e (2.15)

$$s(t) = \sum_n s(nT_s) \text{sinc}(Bt - n\pi) \quad (2.14)$$

$$\hat{s}(t) = \sum_n \hat{s}(nT_s) \text{sinc}(Bt - n\pi) \quad (2.15)$$

A componente de distorção do sinal reconstruído é dada pela diferença entre o sinal quantizado e o sinal original. Assim, a expressão para a componente de distorção é dada pela equação (2.16)

$$q(t) = \sum_n [\hat{s}(nT_s) - s(nT_s)] \text{sinc}(Bt - n\pi) \quad (2.16)$$

Como o sinal $q(t)$ é resultado da diferença entre os sinais, escreve-se a equação (2.17)

$$q(t) = \sum_n q(nT_s) \text{sinc}(Bt - n\pi) \quad (2.17)$$

O sinal $q(t)$ é um sinal indesejado; por este motivo, é denominado ruído de quantização. A potência do ruído de quantização é calculada conforme o desenvolvimento na equação (2.18)

$$\widetilde{q^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} q^2(t) dt \quad (2.18)$$

Aplicando-se a equação (2.17) na equação (2.18), tem-se a equação (2.19)

$$\widetilde{q^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \left[\sum_n q(nT_s) \text{sinc}(Bt - n\pi) \right]^2 dt \quad (2.19)$$

Prova-se que os sinais com argumentos diferentes entre si são ortogonais [1]. O resultado está nas expressões em (2.20)

$$\int_{-\infty}^{\infty} \text{sinc}(2\pi Bt - m\pi) \text{sinc}(2\pi Bt - n\pi) dt = \begin{cases} 0, m \neq n \\ \frac{1}{2B}, m = n \end{cases} \quad (2.20)$$

Com o resultado acima, é possível reescrever a potência do ruído de quantização na equação (2.21):

$$\widetilde{q^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \sum_n q^2(nT_s) \text{sinc}^2(Bt - n\pi) dt \quad (2.21)$$

Rearranjando o somatório e a integral, tem-se a equação (2.22)

$$\widetilde{q^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_n q^2(nT_s) \int_{-\frac{T}{2}}^{\frac{T}{2}} \text{sinc}^2(Bt - n\pi) dt \quad (2.22)$$

A partir do resultado da ortogonalidade entre sinais, segue-se o resultado apresentado na equação (2.23)

$$\widetilde{q^2(t)} = \lim_{T \rightarrow \infty} \frac{1}{2BT} \sum_n q^2(nT_s) \quad (2.23)$$

O intervalo entre os níveis de quantização é dado por Δv , descrito na expressão (2.24):

$$\Delta v = \frac{2A}{L} \quad (2.24)$$

Por este motivo, o máximo valor assumido pelo erro de quantização é de $\pm \Delta v/2$.

O erro de quantização quadrático médio, assumindo equiprobabilidade do erro no intervalo $[-\Delta v/2; \Delta v/2]$, é dado pela equação (2.25)

$$\widetilde{q^2} = \frac{1}{\Delta v} \int_{-\frac{\Delta v}{2}}^{\frac{\Delta v}{2}} q^2 dq = \frac{(\Delta v)^2}{12} q^2 dq = \frac{A^2}{3L^2} \quad (2.25)$$

Este resultado nos permite definir a potência do ruído de quantização, N_q , na equação (2.26).

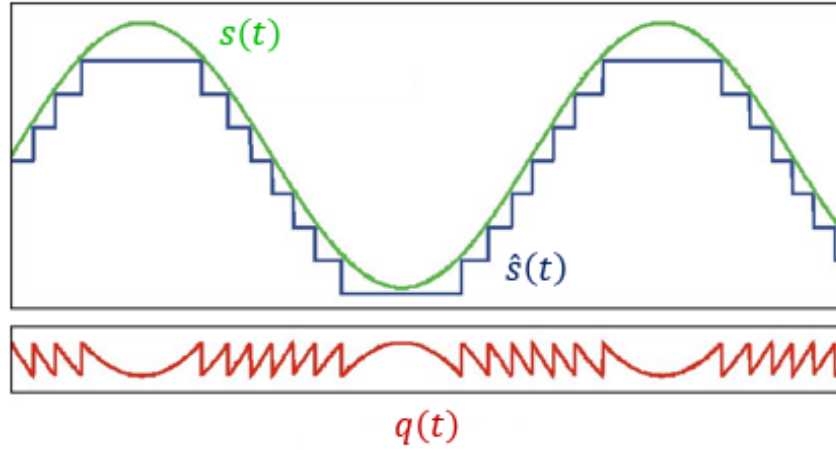


Figura 2.7: Comparação entre sinal original/sinal quantizado e apresentação do erro de quantização. Adaptado de [2]

$$N_q = \overline{q^2(t)} = \frac{A^2}{3L^2} \quad (2.26)$$

O sinal reconstruído na saída do receptor, portanto, é igual ao resultado na equação (2.27)

$$\widehat{s}(t) = s(t) + q(t) \quad (2.27)$$

O sinal desejado na saída é $s(t)$ e o ruído de quantização é $q(t)$. Dado que $\overline{s^2(t)}$ é potência do sinal mensagem $s(t)$, têm-se os resultados nas equações (2.28) e (2.29)

$$S_0 = \overline{s^2(t)} \quad (2.28)$$

$$N_0 = N_q = \frac{A^2}{3L^2} \quad (2.29)$$

A razão sinal-ruído (SNR), portanto, é uma função linear da potência do sinal mensagem, como vemos na equação (2.30).

$$\frac{S_0}{N_0} = 3L^2 \frac{\overline{s^2(t)}}{A^2} \quad (2.30)$$

A Figura 2.7 representa um ilustrativo dos sinais original e quantizado, além do erro de quantização.

Após a quantização, as amostras são codificadas e transmitidas no formato de pulsos binários.

Para sinais com baixa amplitude, faz-se necessário utilizar a quantização não-uniforme: neste caso, os níveis de quantização não necessariamente possuem o mesmo intervalo entre si. A quantização não-uniforme é dividida em duas etapas: a compressão não-linear e quantização uniforme.

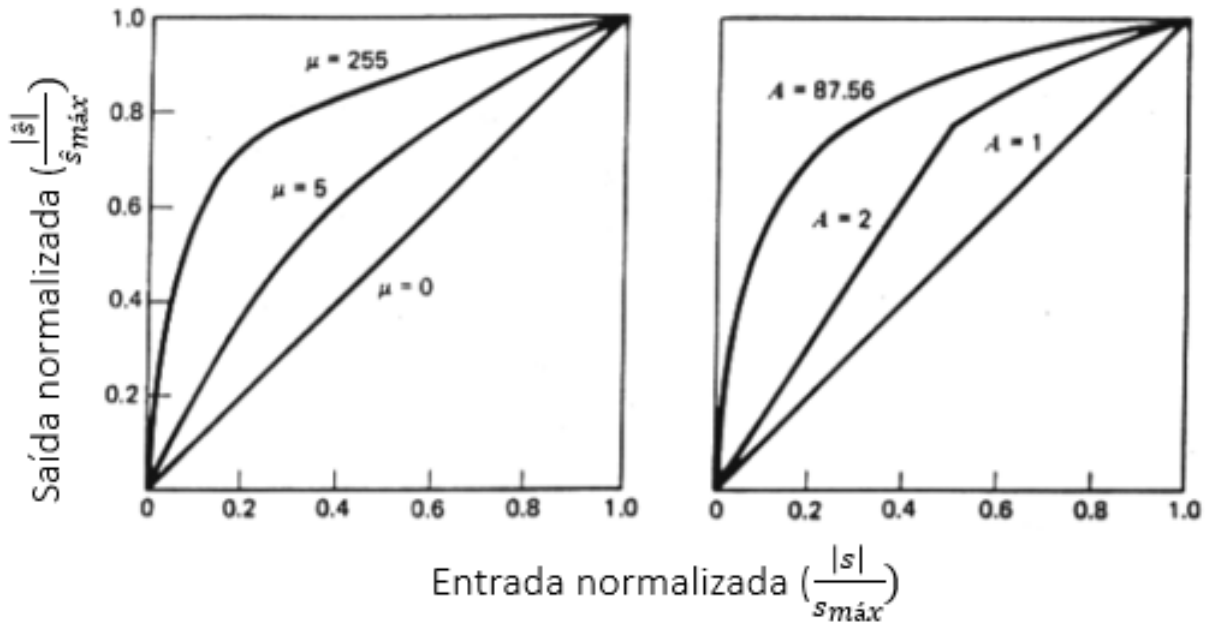


Figura 2.8: Lei μ e Lei A . Adaptado de [3]

Para o processo de compressão não-linear, há duas leis principais. As figuras a seguir apresentam os gráficos representativos da saída normalizada para uma entrada normalizada para cada uma destas leis:

- Lei μ , na América do Norte e Japão;

$$\hat{s} = \frac{\log(1 + \mu |s|)}{1 + \mu} \text{sign}(s) \quad (2.31)$$

- Lei A , na Europa;

$$\hat{s} = \begin{cases} \frac{A|s|}{1+\log A} \text{sign}(s), & 0 \leq |s| \leq \frac{1}{A} \\ \frac{1+\log(A|s|)}{1+\log A} \text{sign}(s), & \frac{1}{A} \leq |s| \leq 1 \end{cases} \quad (2.32)$$

Codificação de Fonte

A codificação de fonte tem a atribuição de formar descrições eficientes das informações emitidas pela fonte. Descrições eficientes permitem redução nos recursos de memória e largura de banda necessários para armazenar ou transportar amostras das informações emitidas pela fonte.

Para fontes discretas, a capacidade de formar descrições com taxa de informação reduzida é relacionada ao conteúdo da informação e a correlação estatística entre os símbolos da fonte.

O objetivo final da codificação de fonte é formar descrições fidedignas das informações enviadas pela fonte para uma determinada taxa de *bit* ou permitir baixas taxas de *bit* para uma determinada descrição da informação enviada pela fonte. O detalhamento deste processo, em seus diferentes formatos, pode ser encontrado em [18].

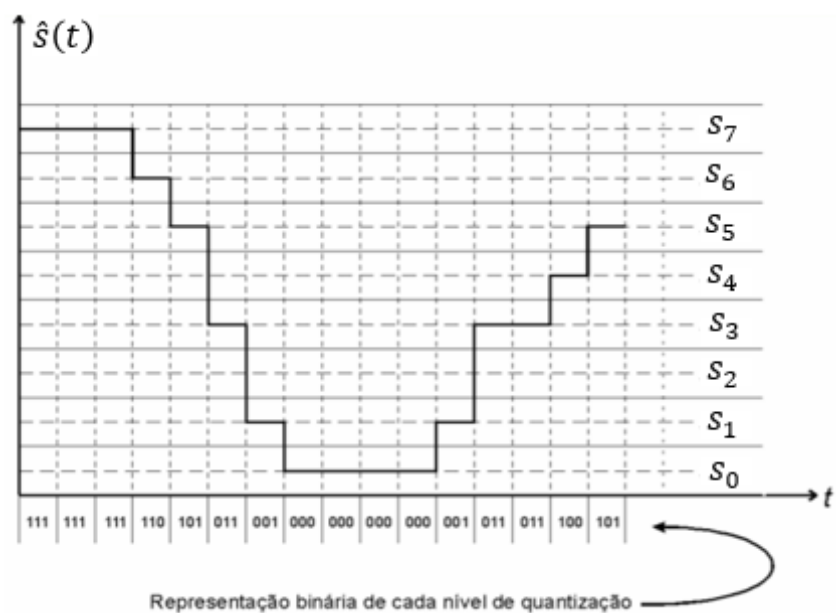


Figura 2.9: Codificação do sinal quantizado. Adaptado de [4]

Um resultado possível do processo de codificação é ilustrado na Figura 2.9.

2.3.3 Codificação e Decodificação de Canal

Além do codificador de fonte, há também o codificador de canal, que introduz redundância controlada na sequência de símbolos apresentada a ele. O decodificador de canal, por sua vez, utiliza esta redundância para controlar os efeitos dos ruídos e distorções causados pela transmissão do sinal pelo canal de comunicações.

2.3.4 Modulação e Demodulação

O modulador de um sistema de comunicação tem a função de mapear a sequência de símbolos enviados em uma sequência de sinais (formas de onda) próprios para transmissão pelo canal de comunicação, que possui natureza analógica. A seleção do tipo de modulação depende de algumas características desejadas, como formato e duração das formas de ondas dos sinais, potência média transmitida, taxa de erro de *bit*/símbolo (BER/SER) e largura de banda.

Além disso, é possível classificar cada tipo de modulação também de acordo com suas capacidades de detecção. Na detecção coerente, utiliza-se a fase da portadora utilizada na modulação para detectar a informação contida no sinal modulado. Na detecção não-coerente, não se utiliza a fase para detectar a informação.

Como modalidades de modulação digital disponíveis, é possível citar:

- ASK – Chaveamento de Amplitude (*Amplitude Shift Keying*): permite detecção coerente e não coerente. Este processo consiste em modificar o nível de amplitude da onda portadora a partir

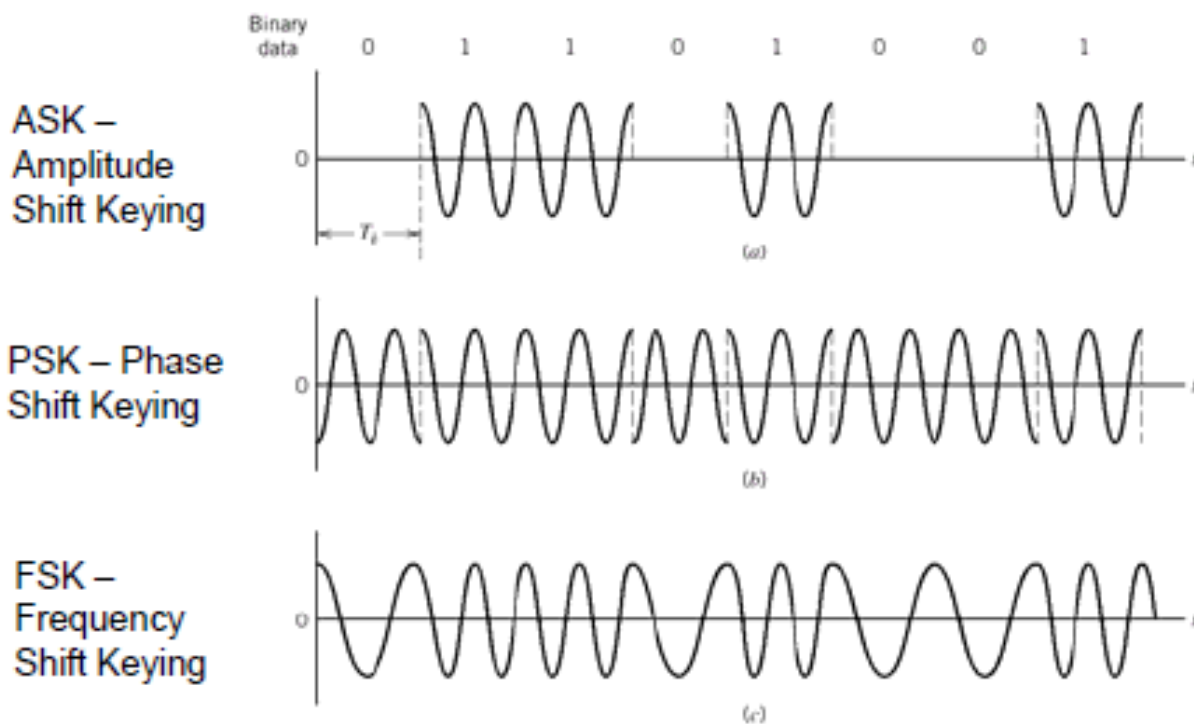


Figura 2.10: Diferentes tipos de modulação digital.

do sinal de entrada emitido pela fonte, que assume dois níveis lógicos discretos (0 ou 1).

- PSK – Chaveamento de Fase (*Phase Shift Keying*): permite detecção coerente. Este processo consiste na variação da fase da onda portadora em função do sinal emitido pela fonte. Com a mudança no nível lógico, muda-se a fase da onda portadora como forma de indicar a mudança no nível lógico do sinal mensagem que está sendo transmitido.

Os resultados apresentados neste trabalho foram realizados utilizando modulação do tipo PSK.

- FSK – Chaveamento de Frequência (*Frequency Shift Keying*): permite detecção coerente e não coerente. Este processo consiste em variar a frequência da onda portadora em função do sinal emitido pela fonte. A amplitude da onda portadora se mantém constante, variando-se a frequência de acordo com os níveis lógicos do sinal enviado.

As formas de onda destes tipos de modulação podem ser visualizadas na Figura 2.10.

A demodulação de um sinal significa a remoção de sua portadora após a sua passagem pelo canal. O demodulador, portanto, executa a função inversa do modulador, determinando uma sequência de símbolos discretos.

Os processos de modulação e demodulação do tipo PSK serão apresentados a seguir, seguindo as demonstrações apresentadas em [1], a fim de construir os conceitos necessários para compreensão do problema e do objetivo final desse trabalho. Na Figura 2.11, verificam-se alguns exemplos de constelação M -PSK. É importante notar nestas constelações a representação do sinal modulado em termos de sua envoltória complexa, em que cada símbolo da constelação de sinais é representado por uma componente em fase e uma componente em quadratura.

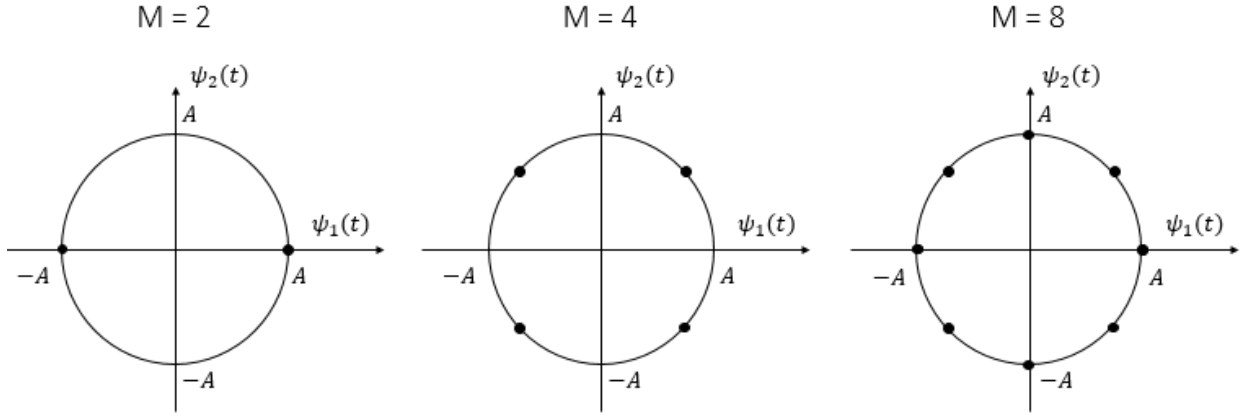


Figura 2.11: Exemplos para constelações M-PSK.

É possível escrever o sinal modulante em banda-base $s(t)$ conforme apresentado na equação (2.33).

$$s(t) = \sum a_m p(t - mT_b), p(t) = \Pi\left(\frac{t - \frac{T_b}{2}}{T_b}\right) \quad (2.33)$$

Se o sinal $s(t)$ é polar, considerando $p(t)$ o pulso básico, para um PSK binário (BPSK), transmite-se “1” por um pulso do tipo $p(t)\cos(2\pi f_c t)$ e “0” por $-p(t)\cos(2\pi f_c t) = p(t)\cos(2\pi f_c t + \pi)$. Deste modo, fica explícito que os pulsos estão defasados entre si em π radianos.

A informação transmitida, portanto, encontra-se na fase ou no sinal do pulso e denomina-se este tipo de modulação de chaveamento por deslocamento de fase (PSK – phase shift keying). Sua forma geral é apresentada na equação (2.34).

$$\varphi_{PSK}(t) = s(t)\cos(2\pi f_c t) \quad (2.34)$$

Para amplitude constante, é possível apresentar o formato para o sinal modulado em PSK na equação (2.35), com amplitude constante [1].

$$\varphi_{PSK}(t) = A\cos(2\pi f_c t + \theta_m), mT_b \leq t < mT_b + T_b \quad (2.35)$$

Expandindo-se a equação (2.35), tem-se a equação (2.36):

$$\varphi_{PSK}(t) = A\cos(\theta_m)\cos(2\pi f_c t) - A\sin(\theta_m)\sin(2\pi f_c t), mT_b \leq t < mT_b + T_b \quad (2.36)$$

Sendo $a_m = A\cos(\theta_m)$ e $b_m = -A\sin(\theta_m)$, reescreve-se a equação (2.36), obtendo-se a equação (2.37)

$$\varphi_{PSK}(t) = a_m\cos(2\pi f_c t) - b_m\sin(2\pi f_c t), mT_b \leq t < mT_b + T_b \quad (2.37)$$

O sinal modulado em PSK pode ser escrito como apresentado na equação (2.38).

$$\varphi_{PSK}(t) = a_m \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t) - b_m \sqrt{\frac{2}{T_b}} \sen(2\pi f_c t), 0 \leq t \leq T_b \quad (2.38)$$

Os sinais $\sqrt{(2/T_b)}\cos(2f_c t)$ e $\sqrt{(2/T_b)}\sen(2f_c t)$ são mutuamente ortogonais e normalizados no intervalo $[0, T_b]$. Estes dois resultados permitem a representação dos símbolos PSK em um sinal bidimensional, utilizando como funções de base ψ_1 e ψ_2 , como apresentado nas equações (2.39) e (2.40).

$$\psi_1(t) = \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t) \quad (2.39)$$

$$\psi_2(t) = \sqrt{\frac{2}{T_b}} \sen(2\pi f_c t) \quad (2.40)$$

A representação do sinal modulado, utilizando-se as funções de base, encontra-se na equação (2.41).

$$\varphi_{PSK}(t) = a_m \psi_1(t) - b_m \psi_2(t) \quad (2.41)$$

A equação (2.42) apresenta condições para o sinal PSK. Em suma, todos os pontos devem estar inseridos em uma circunferência de raio A .

$$[a_m^2 + b_m^2 = A^2 \cos^2(\theta_m) + (-A)^2 \sen^2(\theta_m) = A^2 = constante] \quad (2.42)$$

Os sinais estão igualmente espaçados entre si e isto facilita o processo de equalização, pela uniformidade da classificação. Por este motivo, outra condição para o sinal PSK reside na seleção de ângulos uniformemente espaçados, como apresentado na equação (2.43).

$$[a_m^2 + b_m^2 = A^2 \cos^2(\theta_m) + (-A)^2 \sen^2(\theta_m) = A^2 = constante] \quad (2.43)$$

A demodulação coerente do sinal consiste na multiplicação do sinal modulado pela portadora, seguida pela aplicação de um filtro passa-baixas e um tomador de decisão, que decide qual símbolo foi originalmente enviado.

Para realizar a demodulação coerente, é necessário gerar uma portadora. Um método para regeneração da portadora utilizada no processo de modulação é o método quadrático, que pode ser encontrado em [1].

A representação PSK é um caso especial da modulação em amplitude em quadratura (QAM), com a diferença de que o sinal modulado apresente magnitude constante.

O sinal transmitido por um sistema M -QAM pode ser escrito como mostrado na equação (2.44).

$$p_i(t) = a_i p(t) \cos(2\pi f_c t) + b_i p(t) \sin(2\pi f_c t) = r_i p(t) \cos(2\pi f_c t - \theta_i), i = 1, 2, \dots, M \quad (2.44)$$

Em que

$$r_i(t) = \sqrt{a_i^2 + b_i^2} \quad (2.45)$$

$$\theta_i = \tan^{-1}\left(\frac{b_i}{a_i}\right) \quad (2.46)$$

Escolhendo-se o pulso retangular $p(t)$, tem-se a equação (2.47)

$$p(t) = \sqrt{\frac{2}{T_b}} [u(t) - u(t - T_b)] \quad (2.47)$$

Definem-se os sinais $s_1(t) = a_i p(t)$ e $s_2(t) = b_i p(t)$ – sinais PAM em banda base, em que $s_1(t)$ é a componente em fase e $s_2(t)$ é a componente em quadratura.

O sistema de blocos que apresenta a modulação e a demodulação dos sinais (como visto, tanto para o QAM quanto para o PSK) está na Figura 2.12.

2.3.5 Equalização

Em sua passagem pelo canal, dado seu caráter não-ideal, o sinal é distorcido. Além da distorção do canal, o sinal também é corrompido por ruído. Estas alterações dificultam a composição de uma réplica fidedigna do sinal originalmente transmitido para entrega ao receptor.

Neste sentido surge o processo de equalização de canal, responsável por minimizar os efeitos da distorção do canal e do ruído introduzidos na transmissão. Os detalhes sobre as causas dessas distorções, seus impactos e as melhores formas de retirá-las serão detalhados no próximo capítulo.

2.3.6 Conclusão

Neste capítulo, viram-se os processos que um sinal de informação percorre ao longo de todo o sistema de comunicação digital de forma mais detalhada, desde em que é emitido pela fonte até o momento em que é recebido pelo receptor.

Verificou-se que, neste processamento, há uma etapa em que ocorre uma distorção do sinal originalmente emitido pela fonte, podendo prejudicar a veracidade da informação originalmente enviada ao receptor e comprometendo a confiabilidade do sistema.

No próximo capítulo, detalha-se de forma mais aprofundada o processo de equalização, observando as causas da distorção, seus impactos e apresentando processos convencionais de equalização, que têm o intuito de eliminar as distorções apresentadas.

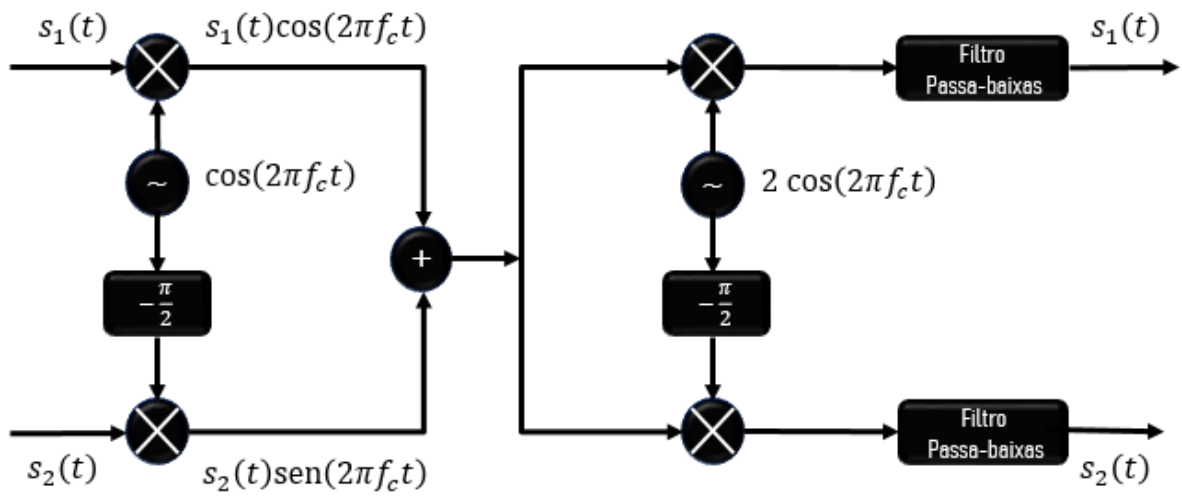


Figura 2.12: Modulação e Demodulação.

Capítulo 3

Desempenho de Sistemas Digitais e Equalização

3.1 Introdução

No Capítulo 2, apresentou-se um modelo geral para um sistema de comunicação digital, assim como um detalhamento dos processos de cada bloco do modelo apresentado sobre o sinal de informação enviado.

Neste capítulo, apresentam-se conceitos fundamentais para algumas métricas do desempenho de um sistema de comunicação digital na Seção 3.1 – Desempenho de Sistemas Digitais, assim como as principais causas e impactos da interferência intersimbólica, motivadora do processo de equalização na Seção 3.2.1 – Primeiro Critério de Nyquist para ISI Nula. Por fim, apresentam-se ainda dois tipos de equalizadores: zero-forcing e MMSE.

3.2 Desempenho de Sistemas Digitais

O desempenho de sistemas digitais é determinado a partir de uma comparação entre o sinal na entrada do receptor e o sinal emitido pela fonte. Por isto, examinar as interferências dos processos aos quais um sinal é submetido em sua passagem por um sistema de comunicação digital no desempenho do sistema é de fundamental relevância.

As seções a seguir apresentam conceitos fundamentais para o entendimento das métricas utilizadas na questão do desempenho de sistemas digitais, assim como as tomadas de decisão que ocorrem no sistema e que contribuem para a determinação de seu desempenho. As demonstrações nas seções 3.1.1 a 3.1.5 são baseadas em [5].

3.2.1 Demodulação

Considerando-se um sistema binário, em um intervalo de sinalização com duração T , o sistema é capaz de transmitir duas formas de ondas possíveis por vez: $s_1(t)$ e $s_2(t)$. O sinal transmitido, portanto, pode ser representado como na equação (3.1), em que $s_1(t)$ é o símbolo 1 e $s_2(t)$ é o símbolo 0.

$$s_i(t) = \begin{cases} s_1(t), 0 \leq t \leq T_b \\ s_2(t), 0 \leq t \leq T_b \end{cases} \quad (3.1)$$

Como visto, $h_c(t)$ é a resposta ao impulso do canal. $n(t)$, por sua vez, é o ruído, aqui representado como uma variável aleatória definida sobre o processo, com média 0 e variância σ^2 , conforme apresentado em [1].

O ruído incorporado ao sinal é do tipo ruído branco aditivo gaussiano (AWGN). Este ruído é modelado como um processo aleatório gaussiano com média zero e densidade espectral de potência $S_n(f) = \frac{N_0}{2} = \sigma^2[W/Hz]$.

Com isto, o sinal recebido $r(t)$ é dado pela equação (3.2)

$$r(t) = s_i(t) * h_c(t) + n(t), i = 1, 2 \quad (3.2)$$

Nos capítulos anteriores, viu-se que o processo de demodulação consiste na recuperação do sinal, com a detecção sendo a tomada de decisão acerca do símbolo transmitido, tomando por base o valor de uma amostra. Após cada intervalo de sinalização T_b , gera-se uma amostra $z(T_b)$. Esta amostra tem amplitude proporcional à energia do sinal no intervalo em questão. A amostra pode ser escrita conforme a equação (3.3)

$$z(T_b) = a_i(T_b) + n(T_b), i = 1, 2 \quad (3.3)$$

Em que $a_i(T_b)$ é a componente desejada do sinal; $n(T_b)$ é a componente do ruído, cuja função de distribuição de probabilidade é dada pela equação (3.4)

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2} \quad (3.4)$$

As funções de distribuição de probabilidade condicionais $p(z | s_1)$ e $p(z | s_2)$, que indicam a probabilidade de $z(T_b)$ ser gerada, dado que s_i foi enviado, são apresentadas nas equações (3.5) e (3.6)

$$p(z | s_1) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-a_1}{\sigma}\right)^2} \quad (3.5)$$

$$p(z | s_2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-a_2}{\sigma}\right)^2} \quad (3.6)$$

Portanto, para o processo de detecção, a informação relevante é o valor de $z(T_b)$. O detector deve escolher entre duas hipóteses: a primeira hipótese - H_1 - indica que o sinal s_1 foi enviado; a segunda hipótese - H_2 - indica que o sinal s_2 foi enviado. A escolha é feita seguindo as expressões em (3.7)

$$H_i = \begin{cases} z(T_b) > \gamma, H_1 \\ z(T_b) < \gamma, H_2 \end{cases} \quad (3.7)$$

3.2.2 Representação Vetorial de Sinais

Os sinais enviados podem ser representados como vetores. Para tanto, é necessário definir um espaço ortogonal com N dimensões a partir de um conjunto de N funções linearmente independentes $\psi_j(t)$. Este conjunto é denominado como o conjunto de base de funções do espaço ortogonal de sinais.

Os sinais da base $\psi_j(t)$ devem satisfazer as condições apresentadas na equação (3.8), com $j, k = 1, \dots, N$

$$\int_0^{T_b} \psi_j(t)\psi_k(t)dt = K_j\delta_{jk}, 0 \leq t \leq T_b \quad (3.8)$$

Em que

$$\delta_{jk} = \begin{cases} 1, j = k \\ 0, c.c \end{cases} \quad (3.9)$$

A constante K_j apresentada na equação (3.8) representa a energia do sinal $\psi_j(t)$. Se esta constante for diferente de zero, o espaço de sinais é ortogonal; se for igual a 1, o espaço de sinais é ortonormal.

$$E_j = \int_0^{T_b} \psi_j^2(t)dt = K_j \quad (3.10)$$

Um conjunto arbitrário de formas de ondas com duração T_b , dado por $s_i(t)$ ($i = 1, \dots, M$), pode ser expresso como uma combinação linear de funções ortogonais $\psi_1(t), \psi_2(t), \dots, \psi_N$ ($N \leq M$). Portanto, de forma geral, tem-se a representação das formas de onda na equação (3.11), para $N \leq M$

$$s_i(t) = \sum_{j=1}^N a_{ij}\psi_j(t), i = 1, \dots, M \quad (3.11)$$

Os coeficientes a_{ij} são calculados como na equação (3.12), considerando-se $i = 1, \dots, M$ e $j = 1, \dots, N$

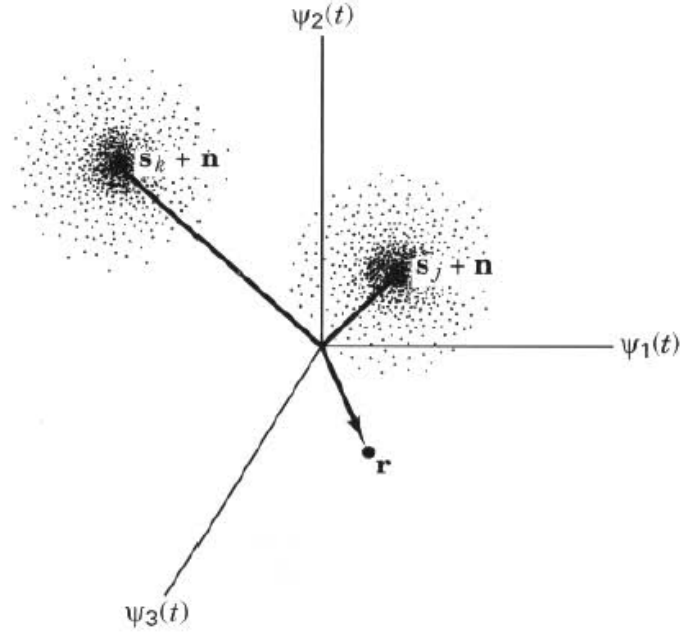


Figura 3.1: Representação da nuvem formada com a inserção do ruído ao redor do símbolo transmitido. Adaptado de [5].

$$a_{ij} = \int_0^{T_b} s_i(t)\psi_j(t)dt \quad (3.12)$$

Com isto, tem-se que um sinal $s_i(t)$ pode ser representado por um vetor $s_i = (a_{i1}, a_{i2}, \dots, a_{iN})$ e s_i é caracterizado pelas componentes a_{ij} em relação a uma base de funções ortogonais.

Usualmente, vetores s_j e s_k são representantes dos sinais de referência transmitidos em um conjunto com M possibilidades, considerando um esquema de modulação M -ário. O sinal recebido, portanto, consiste em uma combinação do sinal de referência juntamente com o ruído, podendo ser representado por uma nuvem de pontos ao redor de s_j , como vemos na Figura 3.1.

A função do receptor, mediante a apresentação do sinal recebido, consiste na decisão acerca da nuvem a qual o sinal recebido \mathbf{r} pertence, levando em consideração a menor distância entre \mathbf{r} e os sinais \mathbf{s}_i .

A energia de um sinal $s_i(t)$ pode ser apresentada em termos de suas componentes, como na equação (3.13).

$$E_i = \int_0^{T_b} s_i^2(t)dt = \int_0^{T_b} \left[\sum_{j=1}^N a_{ij}\psi_j(t) \right]^2 dt \quad (3.13)$$

Separando o termo quadrático em dois termos lineares em k e em j , tem-se a equação (3.14)

$$E_i = \int_0^{T_b} \sum_{j=1}^N a_{ij} \psi_j(t) \sum_{k=1}^N a_{ik} \psi_k(t) dt \quad (3.14)$$

Retirando-se os coeficientes da integral, apenas as funções em t permanecem. Tem-se a equação (3.15)

$$E_i = \sum_{j=1}^N \sum_{k=1}^N a_{ij} a_{ik} \int_0^T \psi_j(t) \psi_k(t) dt \quad (3.15)$$

O resultado da integral já é conhecido e foi apresentado na equação (3.8). Com isto, escreve-se a equação (3.16)

$$E_i = \sum_{j=1}^N \sum_{k=1}^N a_{ij} a_{ik} K_j \delta_{jk} = \sum_{j=1}^N a_{ij}^2 K_j dt, i = 1, \dots, M \quad (3.16)$$

Considerando $K_j = 1$, tem-se uma base ortonormal. Por isso, a energia do sinal é relacionada com a norma do vetor s_i como apresentado na equação (3.17)

$$E_i = \sum_{j=1}^N a_{ij}^2 = \|s_i\|^2 \quad (3.17)$$

A partir de um conjunto de M sinais $s_i(t)$, é possível obter uma base ortonormal utilizando o procedimento de Gram-Schmidt, como visto em [5].

Assim como os sinais, o ruído também pode ser representado no espaço de sinais em função das funções de base ortogonais. O ruído apresentado na entrada do receptor é dado pela equação (3.18)

$$n(t) = \hat{n}(t) + \tilde{n}(t), \hat{n}(t) = \sum_{j=1}^N n_j \psi_j(t) \quad (3.18)$$

Em que $\tilde{n}(t)$ é o ruído descartado pelo receptor, dado que ele não pertence ao espaço de sinais. Na equação (3.19), vê-se a representação de $\tilde{n}(t)$

$$\tilde{n}(t) = n(t) - \hat{n}(t) \quad (3.19)$$

As componentes do ruído n_j são obtidas fazendo-se projeções nas funções de base. Na equação (3.20), tem-se

$$n(t) = \sum_{j=1}^N n_j \psi_j(t) + \tilde{n}(t) \quad (3.20)$$

Em que

$$n_j = \frac{1}{K_j} \int_0^T n(t)\psi_j(t)dt, \forall j \quad (3.21)$$

E, ainda,

$$\int_0^{T_b} \tilde{n}(t)\psi_j(t)dt = 0, \forall j \quad (3.22)$$

Com as considerações acima, é possível simplificar a notação apresentada, fazendo com que $\hat{n}(t) = n(t)$. No espaço de sinais, $n(t)$ pode ser representado pelo vetor $\mathbf{n} = (n_1, n_2, \dots, n_N)$, em que \mathbf{n} é um vetor aleatório, com média nula, distribuição gaussiana e com componentes n_j independentes entre si, como apresentado anteriormente. Para o ruído AWGN com densidade espectral de potência igual a $N_o/2$, a variância é dada pela equação (3.23).

$$\int_0^{T_b} \tilde{n}(t)\psi_j(t)dt = 0, \forall j \quad (3.23)$$

3.2.3 Relação Sinal Ruído para Sistemas Digitais

A razão sinal-ruído (SNR) é uma métrica de desempenho de sistemas de comunicação. Em sistemas analógicos, esta razão é dada pela razão entre a potência média do sinal e a potência média do ruído; em sistemas digitais, utiliza-se uma versão normalizada desta relação, denotada por E_b/N_o , em que E_b é a energia média do *bit* e N_o é o valor da densidade espectral de potência do ruído. Sendo S a potência média do sinal, N a potência média do ruído, W a largura de banda e R a taxa de *bit*, verifica-se que E_b/N_o é uma versão da SNR dos sistemas analógicos S/N normalizada pela largura de banda e pela taxa de *bit*

$$\frac{E_b}{N_o} = \frac{ST_b}{N/W} = \frac{S/R_b}{N/W} = \frac{S}{N} \frac{W}{R} \quad (3.24)$$

A Figura 3.2 apresenta a relação entre a probabilidade de erro para cada E_b/N_o . Esta curva é denominada curva de taxa de erro de *bit* (BER). Vale ressaltar que esta curva apresenta comportamentos diferentes para diferentes modulações.

Na Figura 3.1, viu-se a influência do ruído no sinal transmitido. Esta influência é diferente para cada E_b/N_o . Denomina-se diagrama de constelação o diagrama que representa os sinais transmitidos para uma determinada modulação. Na Figura 3.3, tem-se os diagramas de constelação para modulação do tipo 16-PSK sob a influência de ruído AWGN e sem distorção de canal, para diferentes E_b/N_o .

3.2.4 Receptor de Máxima Verossimilhança

Na seção 3.1.1, viu-se como é feita a escolha da hipótese com base no valor de $z(T_b)$. O valor é selecionado de forma a atender a algum critério de desempenho do projeto, sendo um dos critérios mais utilizados a minimização da probabilidade de erro de detecção.

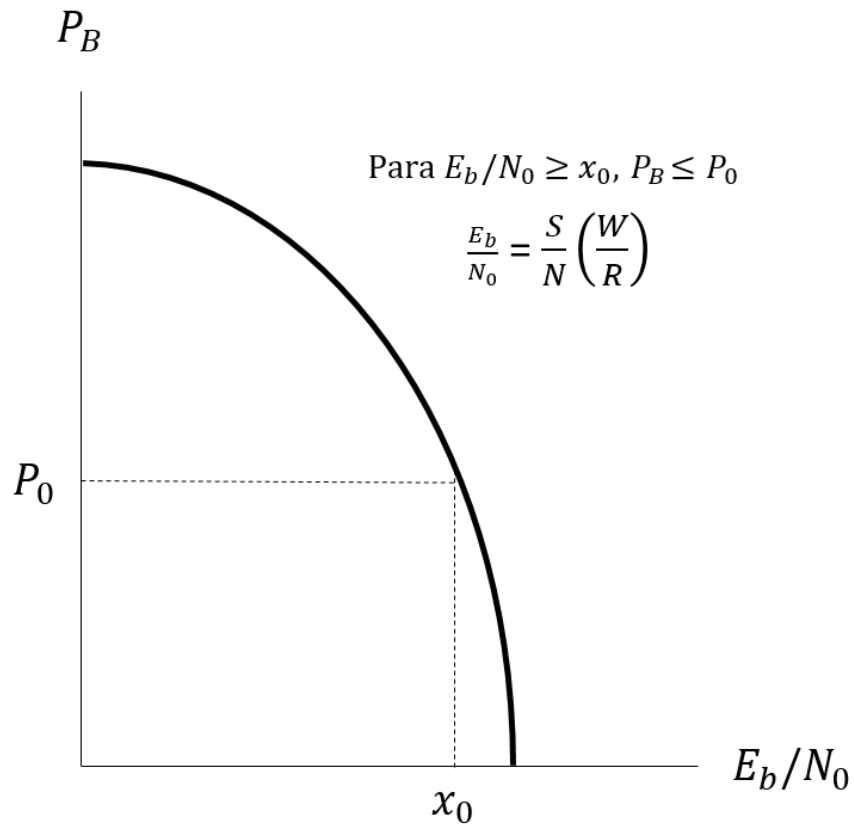


Figura 3.2: Probabilidade de erro para cada valor de E_b/N_0 .

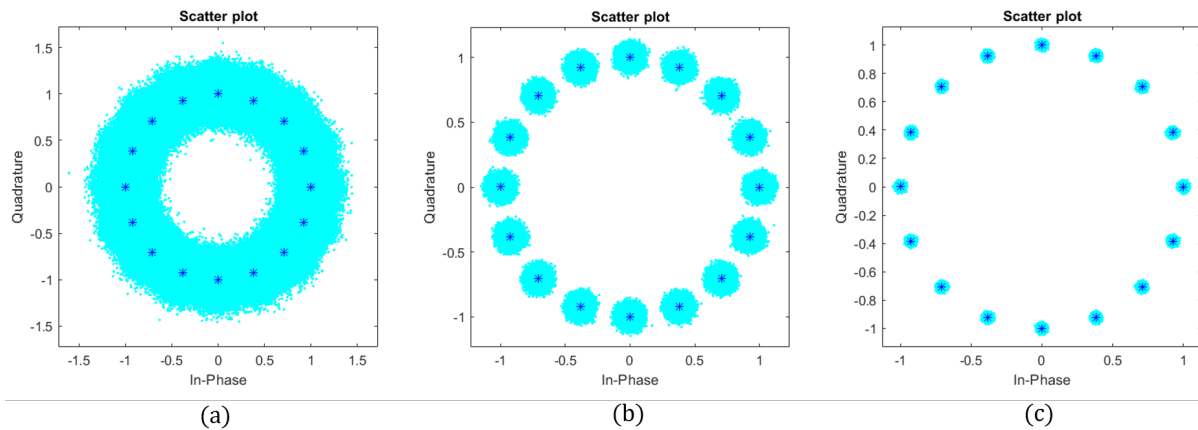


Figura 3.3: Diagrama de Constelação para o 16-PSK: (a) $E_b/N_0 = 10\text{dB}$; (b) $E_b/N_0 = 20\text{dB}$; (c) $E_b/N_0 = 30\text{dB}$.

A aplicação deste critério resulta na regra MAP (Máxima Probabilidade a Posteriori), que consiste em tomar a decisão pela hipótese que segue as equações (3.25) e (3.26)

$$H_1 = P(s_1)p(z | s_1) > P(s_2)p(z | s_2) \quad (3.25)$$

$$H_2 = P(s_1)p(z | s_1) < P(s_2)p(z | s_2) \quad (3.26)$$

Além disto, a regra MAP pode ser escrita em função da razão de verossimilhança $\Lambda(z)$, como nas equações (3.27) e (3.28)

$$\Lambda(z) = \frac{p(z | s_1)}{p(z | s_2)} > \frac{P(s_2)}{P(s_1)} \rightarrow H_1 \quad (3.27)$$

$$\Lambda(z) = \frac{p(z | s_1)}{p(z | s_2)} < \frac{P(s_2)}{P(s_1)} \rightarrow H_2 \quad (3.28)$$

Em que $P(s_1)$ e $P(s_2)$ são as probabilidades a priori dos sinais s_1 e s_2 . Considerando-se o caso em que s_1 e s_2 são equiprováveis, a partir da regra MAP, reescreve-se

$$z(T_b) > \frac{a_1 + a_2}{2} = \gamma_0 \rightarrow H_1 \quad (3.29)$$

$$z(T_b) < \frac{a_1 + a_2}{2} = \gamma_0 \rightarrow H_2 \quad (3.30)$$

Essa estratégia é conhecida como critério do erro mínimo e γ_0 representa o limiar ótimo a fim de minimizar a probabilidade de uma tomada de decisão equivocada.

3.2.5 Probabilidade de Erro

O erro na questão de desempenho de um sistema digital consiste no evento em que a tomada de decisão falha, resultando na opção por um símbolo diferente do transmitido. Matematicamente, tem-se a probabilidade de erro condicional para s_1 e s_2 nas equações (3.31) e (3.32)

$$P(e | s_1) = P(H_2 | s_1) = \int_{-\infty}^{\gamma_0} p(z | s_1) dz \quad (3.31)$$

$$P(e | s_2) = P(H_1 | s_2) = \int_{\gamma_0}^{\infty} p(z | s_2) dz \quad (3.32)$$

Portanto, a probabilidade de erro é definida pela equação (3.33)

$$P_B = \sum_{i=1}^2 P(e, s_i) = \sum_{i=1}^2 P(e | s_i)P(s_i) \quad (3.33)$$

Expandindo-se a equação (3.33)

$$P_B = P(e | s_1)P(s_1) + P(e | s_2)P(s_2) = P(H_2 | s_1)P(s_1) + P(H_1 | s_2)P(s_2) \quad (3.34)$$

Considerando-se novamente o caso em que s_1 e s_2 são equiprováveis e levando em consideração a simetria entre $p(z | s_1)$ e $p(z | s_2)$, tem-se novamente a probabilidade de erro na equação (3.35).

$$P_B = \frac{1}{2}P(H_2 | s_1) + \frac{1}{2}P(H_1 | s_2) = P(H_2 | s_1) = P(H_1 | s_2) \quad (3.35)$$

De acordo com (3.32)

$$P_B = \int_{\gamma_0 = \frac{(a_1+a_2)}{2}}^{\infty} p(z | s_2) dz \quad (3.36)$$

Substituindo-se $p(z | s_2)$ pelo resultado apresentado na equação (3.6)

$$P_B = \int_{\gamma_0 = \frac{(a_1+a_2)}{2}}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{1}{2}\left(\frac{z-a_2}{\sigma}\right)^2\right]} dz \quad (3.37)$$

Fazendo-se $u = \frac{z-a_2}{\sigma}$

$$P_B = \int_{u = \frac{(a_1-a_2)}{2\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\left[-\frac{u^2}{2}\right]} du = Q\left(\frac{a_1 - a_2}{2\sigma}\right) \quad (3.38)$$

A função $Q(x)$ apresentada na equação (3.38) é denominada função de erro complementar.

$$Q(x) = \frac{1}{\sqrt{2\pi}} e^{\left[-\frac{u^2}{2}\right]} \quad (3.39)$$

A partir desta função, desenha-se a curva de desempenho de um sistema digital a partir de sua BER para cada E_b/N_o . A Figura 3.4 apresenta exemplos de curvas de BER teóricas (sem interferência da distorção de um canal não-ideal) para três modulações diferentes.

3.3 Equalização

Como visto, o canal utilizado como meio de transmissão é um canal não-ideal, corrompendo o sinal transmitido em conjunto com o ruído AWGN. Com isto, surge a necessidade de implementar um processo no sistema de comunicação digital que tenha como intuito reduzir os efeitos da distorção provocada pelo canal.

A seguir, apresentam-se as causas do surgimento da interferência intersimbólica dada a distorção provocada pelo canal (responsável pela necessidade de equalização), assim como alguns tipos de equalizadores que podem ser utilizados no processo. As demonstrações apresentadas nas seções 3.3.1 e 3.3.2 têm como referência [1].

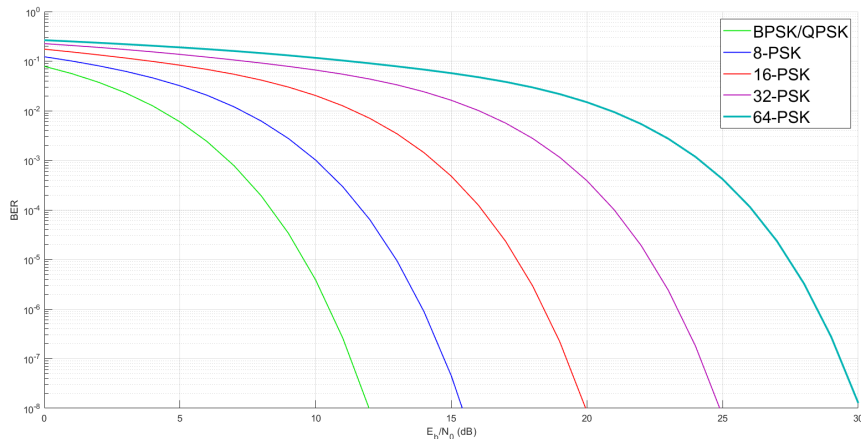


Figura 3.4: Curva de BER para diferentes técnicas de modulação.

3.3.1 Interferências Intersimbólicas (ISI) e Efeitos

Interferência intersimbólica (ISI) é o nome que se dá ao fenômeno em que ocorre o espalhamento de um pulso além de sua janela de tempo T_b , considerando-se um pulso a transmissão de um *bit* (o tempo de transmissão de um símbolo é denominado T_s e é dado por nT_b). Este espalhamento interfere em pulsos vizinhos, causando distorção e prejudicando a detecção do pulso.

Este espalhamento é provocado pela não-idealidade dos canais de comunicação; um canal não-ideal é um canal que não é livre de distorção em toda a largura de banda do sinal.

A seguir, ilustra-se o processo de geração da ISI, de acordo com a apresentação de [1].

Escolhe-se transmitir um pulso a cada T_b segundos, sendo o k -ésimo pulso definido pela expressão $a_k p(t - kT_b)$. O intuito é que seja possível detectar a amplitude do pulso a_k de forma correta após sua passagem por um canal de largura de banda finita.

O pulso considerado até aqui é um pulso limitado no tempo. Um pulso limitado no tempo implica, por definição, em um pulso não-limitado no domínio da frequência e, portanto, uma parte de seu espectro será retirada, devido ao caráter finito da largura de banda do canal não-ideal. Esta supressão implica no espalhamento dos pulsos, provocando ISI e distorcendo os pulsos.

Uma solução possível consistiria na limitação do pulso em frequência para passagem pelo canal. Esta limitação em frequência, no entanto, não é uma alternativa viável, uma vez que um pulso limitado em frequência é, por definição, um pulso não-limitado no domínio do tempo. Pulsos não-limitados no tempo se sobrepõem e, novamente, têm-se ISI.

Verifica-se, portanto, que a limitação do pulso, seja no domínio da frequência ou no domínio do tempo não é capaz de eliminar a distorção. Isto indica que a ISI não pode ser evitada e é um fenômeno inerente à transmissão em largura de banda finita.

Apesar disto, há uma forma de operar de forma a minimizar os impactos da interferência intersimbólica: para detectar as amplitudes de pulso sem distorção mesmo com o espalhamento

ou a sobreposição dos pulsos, é preciso garantir que, no momento da tomada de decisão acerca da amplitude de pulso, não haja interferência intersimbólica. Isto é possível com a formatação correta de pulsos limitados em frequência.

A formatação adequada para os pulsos deve obedecer a três critérios estabelecidos por Nyquist, assumindo que a sobreposição de pulsos ocorra. Esta formatação é feita de forma a garantir que a interferência seja nula em todos os outros pulsos no instante da tomada de decisão. Desta forma, com a exigência da interferência nula apenas para os instantes de tomada de decisão, elimina-se a necessidade de que os pulsos sejam totalmente não-sobrepostos.

Neste trabalho, aborda-se apenas o primeiro critério, apresentado a seguir.

3.3.2 Primeiro Critério de Nyquist para ISI Nula

O primeiro critério consiste na escolha de uma forma de pulso que apresente amplitude não nula no tempo $t=0$ e nula nos instantes de tempo $t = \pm nT_b$ (em que $n = 1, 2, 3, \dots$), sendo T_b a janela de tempo entre a transmissão de cada pulso. Este pulso pode ser representado pela equação (3.40).

$$p(t) = \begin{cases} 1, & t = 0 \\ 0, & t = \pm nT_b \end{cases} \quad (3.40)$$

O pulso que satisfizer os critérios acima causa ISI zero nos centros de todos os outros pulsos transmitidos. Na Figura, fica explícito que, nos instantes de amostragem, tem-se apenas a amplitude de um pulso, centralizado no instante de amostragem e sem interferência de outros pulsos.

A transmissão de R_b bits/s requer uma largura de banda mínima teórica de $R_b/2$ Hz, como visto no capítulo anterior. Portanto, é necessário que o pulso $p(t)$ escolhido atenda a esta condição. O pulso que atende às duas condições apresentadas é a função $\text{sinc}(R_b t)$, como vê-se na equação (3.41), em que $T_b = 1/R_b$.

$$\text{sinc}(R_b t) = \begin{cases} 1, & t = 0 \\ 0, & t = \pm nT_b \end{cases} \quad (3.41)$$

Cuja transformada de Fourier é dada pela equação (3.42)

$$P(f) = \frac{1}{R_b} \text{rect}\left(\frac{f}{2B}\right) \quad (3.42)$$

Este pulso, no entanto, é impraticável: sua duração é ilimitada com decaimento lento e truncá-lo para torna-lo praticável não é uma opção, dado que isto aumentaria a largura de banda e faria com que o pulso deixasse de atender ao critério da largura de banda. Além disto, o pulso decai a uma taxa lenta, de $1/t$.

Com isto, torna-se necessário encontrar um pulso que decaia de forma mais rápida. Nyquist mostrou que um pulso como esse tem largura de banda $(1+r)R_b/2$, sendo r o fator de *roll-off*, definido posteriormente neste texto.

Prova-se a aplicabilidade deste pulso a seguir, de acordo com [1].

Considera-se $p(t) \Leftrightarrow P(f)$, com largura de banda de $P(f)$ no intervalo $(R_b/2, R_b)$, um pulso que atende às condições apresentadas nas expressões (3.1) .

Em seguida, amostra-se $p(t)$ a cada T_b segundos, multiplicando-o por um trem de impulsos $\delta_{(T_b)}(t)$. De acordo com as propriedades em 3.1, todas as amostras, excetuando a amostra em $t = 0$. Desta forma, o sinal amostrado $\bar{p}(t)$ é dado pela equação (3.43)

$$\bar{p}(t) = p(t)\delta_{(T_b)}(t) = \delta(t) \quad (3.43)$$

Aplica-se a transformada de Fourier em ambos os lados da equação e obtém-se a equação (3.44), com $R_b = 1/T_b$

$$\frac{1}{T_b} \sum_{n=-\infty}^{\infty} P(f - nR_b) = 1 \quad (3.44)$$

Alternativamente, escreve-se a equação (3.45) que mostra que a soma dos espectros formados pela repetição de $P(f)$ a uma frequência R_b é uma constante T_b .

$$\sum_{n=-\infty}^{\infty} P(f - nR_b) = T_b \quad (3.45)$$

Para o espectro em um intervalo $0 < f < R_b$, tem-se equação (3.46)

$$P(f) + P(f - R_b) = T_b \quad (3.46)$$

Para $x = f - R_b/2$, reescreve-se a equação (3.46) e obtém-se a equação (3.47)

$$P\left(x + \frac{R_b}{2}\right) + P\left(x - \frac{R_b}{2}\right) = T_b, |x| < \frac{R_b}{2} \quad (3.47)$$

Em seguida, aplica-se a propriedade da simetria conjugada na equação (3.47) e obtém-se a equação (3.48).

$$P\left(\frac{R_b}{2} + x\right) + P^*\left(\frac{R_b}{2} - x\right) = T_b, |x| < \frac{R_b}{2} \quad (3.48)$$

Escolhe-se $P(f)$ com valores reais e positivos e então basta que $|P(f)|$ satisfaça a equação (3.48). Dado que $P(f)$ é real, então tem-se a equação (3.49).

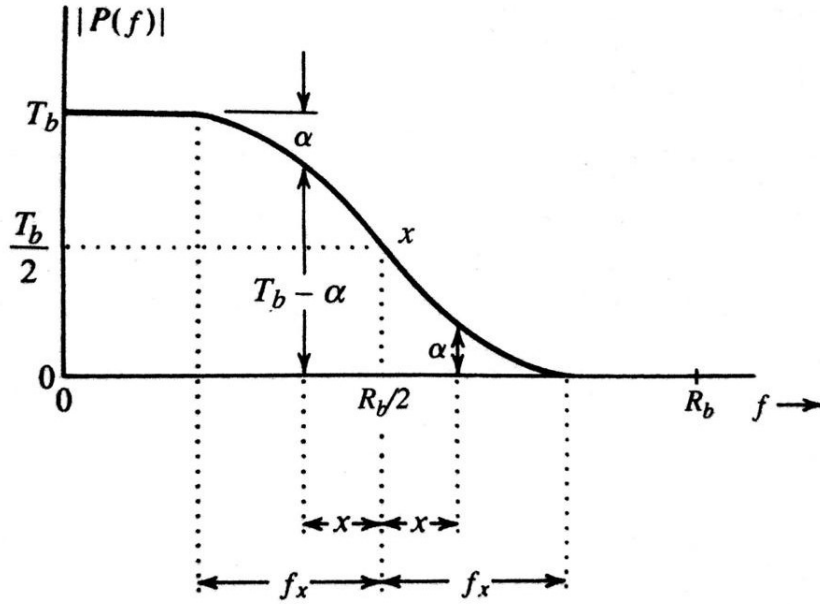


Figura 3.5: Forma de onda sugerida para $|P(f)|$. Adaptado de [1].

$$\left| P\left(\frac{R_b}{2} + x\right) \right| + \left| P\left(\frac{R_b}{2} - x\right) \right| = T_b, |x| < \frac{R_b}{2} \quad (3.49)$$

Segue que a forma de $|P(2\pi f)| = |P(\omega)|$ deve ser uma curva com simetria ímpar em relação ao conjunto de eixos que se cruzam no ponto α (ponto na curva $P(f)$ em que $f = R_b/2$, como visto na Figura 3.5). Isto requer que a condição apresentada na equação (3.50) seja válida.

$$\left| P\left(\frac{R_b}{2}\right) \right| = \frac{|P(0)|}{2} \quad (3.50)$$

A largura de banda de $P(f)$ em Hertz é $R_b/2 + f_x$, em que f_x é a largura de banda além da largura de banda mínima $R_b/2$.

Sendo r a razão entre o excesso de largura de banda e a largura de banda mínima proposta na teoria, escreve-se a equação (3.51)

$$r = \frac{f_x}{R_b/2} = 2f_x T_b \quad (3.51)$$

Dado que o excesso de largura de banda não pode ser superior a $R_b/2$, tem-se a equação (3.52)

$$0 \leq r \leq 1 \quad (3.52)$$

Levando-se em consideração a mínima largura de banda teórica $R_b/2$ e o excesso de largura de banda $f_x = R_b/2$, a largura de banda de $P(f)$ é dada pela expressão (3.53)

$$B_T = \frac{R_b}{2} + r \frac{R_b}{2} = \frac{(1+r)R_b}{2} \quad (3.53)$$

O fator r apresentado é denominado fator de decaimento (*roll-off*) e indica a porcentagem de acréscimo na largura de banda de $P(f)$ para um espectro que segue o primeiro critério de Nyquist.

Uma família de espectros que satisfazem o primeiro critério de Nyquist é dada pela equação (3.54).

$$P(f) = \begin{cases} 1, & |f| < \frac{R_b}{2} - f_x \\ \frac{1}{2}, & \left| f - \frac{R_b}{2} \right| < f_x \\ 0, & |f| > \frac{R_b}{2} + f_x \end{cases} \quad (3.54)$$

Para o caso em que $r = 1$, a equação (3.54) se reduz à expressão em (3.55)

$$P(f) = \frac{1}{2}(1 + \cos(\pi f T_b)) \Pi\left(\frac{f}{2R_b}\right) = \cos^2\left(\frac{\pi f T_b}{2}\right) \Pi\left(\frac{f T_b}{2}\right) \quad (3.55)$$

A equação (3.54) é conhecida como cosseno levantado, dado que representa um cosseno levantado por sua amplitude de pico. A equação (3.55), por sua vez, é denominada por decaimento de cosseno amplo.

A transformada de Fourier do espectro é dada pela expressão (3.56).

$$p(t) = R_b \frac{\cos(\pi R_b t)}{1 - 4R_b^2 t^2} \text{sinc}(R_b t) \quad (3.56)$$

Algumas características do pulso cosseno levantado são:

- Possui largura de banda R_b Hz, com valor R_b em $t = 0$;
- Apresenta valor nulo não só em todos os outros instantes de amostragem, assim como nos pontos médios entre todos os instantes de sinalização;
- O pulso decai mais rapidamente, a uma taxa de $1/t^3$;
- Possui relativa insensibilidade a desvios de R_b e da taxa de amostragem, em decorrência dos pontos citados acima;
- É realizável com boa aproximação;
- Possui característica de fase praticamente linear, eliminando a necessidade de equalização de fase adicional.

Portanto, os pulsos recebidos na entrada do detector são os que devem apresentar ISI nula. Sendo assim, considerando um canal não-ideal e que introduz distorção no sinal, os pulsos transmitidos devem ser formatados para que, após sua passagem pelo canal, sejam recebidos com a forma de um pulso cosseno levantado.

Sendo $H_c(f)$ a função de transferência do canal, o pulso transmitido $p_i(t)$ deve satisfazer a equação (3.57).

$$P_i(f)H_c(f) = P(f) \quad (3.57)$$

3.3.3 Equalizadores

Um equalizador é um dispositivo utilizado em sistemas de comunicações digitais para retirar a distorção introduzida pelo canal, que tem a forma de dispersão, provocada pela atenuação de componentes em frequências críticas.

Idealmente, a característica em frequência do equalizador deve ser dada pelo inverso da característica em frequência do canal, assim as distorções seriam anuladas. Com isto, as componentes em frequências críticas seriam restauradas e a dispersão dos pulsos seria eliminada. No entanto, esta modelagem também provocaria um fenômeno chamado amplificação de ruído: o ruído de canal recebido seria aumentado, como consequência da restauração das componentes em frequências críticas.

A dispersão dos pulsos em um sistema de comunicação digital gera ISI, que, por sua vez, aumenta o erro de detecção. O aumento do ruído em decorrência do processo de equalização (considerando-se o trem de impulsos e a função sinc utilizados na amostragem do sinal, é fácil visualizar o aumento das frequências que não correspondem à banda do sinal, dada a modelagem ideal apresentada de inversão da função de transferência do canal) também provoca aumento na probabilidade do erro de detecção. Portanto, o projeto de um equalizador deve levar em consideração não só a atenuação da interferência intersimbólica, mas também a atenuação do ruído de canal [1].

Entre os tipos de filtros equalizadores, podemos citar:

- Estimador de Sequência de Máxima Verossimilhança (*MLSE* com uso do Algoritmo de Viterbi);
- *Zero-forcing* (ZF);
- Erro Quadrático Médio Mínimo (*MMSE*);
- *Decision Feedback* (DF).

Nesta seção, serão demonstrados os procedimentos necessários para a definição do projeto da estrutura de alguns equalizadores convencionais, com a determinação dos seus vetores de coeficientes \mathbf{c} . As demonstrações apresentadas para os equalizadores têm como referência [19].

A Figura 3.6 apresenta um modelo geral de filtro adaptativo para os equalizadores detalhados nesta seção.

Primeiramente, detalham-se os princípios de funcionamento dos filtros equalizadores do tipo *zero-forcing*, que são equalizadores lineares. Estes equalizadores correlacionam o vetor do sinal recebido com o vetor de coeficientes \mathbf{c} do equalizador para produzir uma estatística de decisão, a partir da qual será selecionado o símbolo enviado.

Este processo fica evidenciado na equação (3.58), em que $r[n]$ é o sinal recebido, $b[n]$ é o sinal

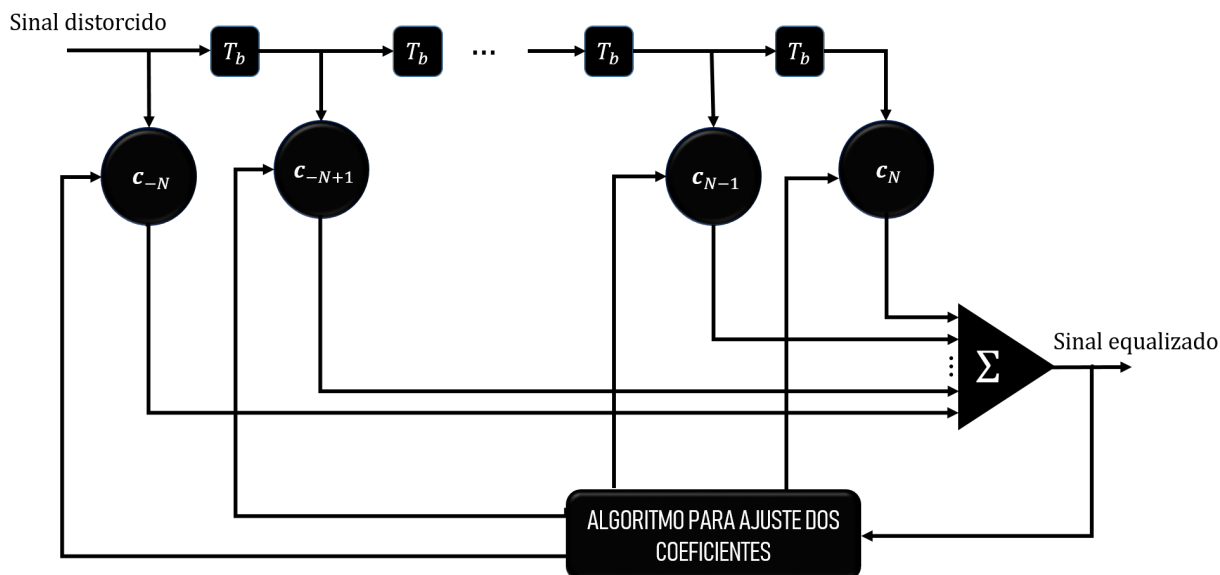


Figura 3.6: Exemplo de filtro adaptativo

enviado e $Z[n]$ é a estatística de decisão, sendo n o instante de amostragem, definido em função do tempo de símbolo, T (a amostragem ocorre à taxa de símbolo).

É importante notar que a estatística de decisão $Z[n]$ é dada para um determinado sinal amostrado $z(T_b)$ em um instante nT_b , como apresentado na equação (3.3).

$$Z[n] = \mathbf{c}^H r[n] = b[n] (\mathbf{c}^H u_0) + \sum_{i \neq 0} b[n+i] (\mathbf{c}^H u_i) + \mathbf{c}^H w[n] \quad (3.58)$$

Na equação (3.58), todos os termos da soma são termos referentes à interferência intersimbólica e, portanto, indesejados. O termo referente ao sinal enviado (com $i = 0$) deve ser maximizado em detrimento da minimização dos termos da soma. O último termo $\mathbf{c}^H w[n]$ é resultado dos efeitos do ruído no sistema. Os termos u são referentes ao vetor sinal, que correspondem a cada símbolo individual. Com a comparação da estatística de decisão com um limiar de decisão, obtemos o símbolo transmitido.

Diferenciamos cada equalizador de acordo com os coeficientes utilizados na solução.

O equalizador do tipo *zero-forcing* é o equalizador ideal, dado que, como o próprio nome já diz, ele força todos os termos referentes à interferência intersimbólica na saída do correlator a zero. Portanto, a solução deste equalizador envolve zerar todos os termos da soma na equação (3.58), e fazer com que o termo referente ao símbolo $b[n](\mathbf{c}^H u_0[n])$ seja igual a 1.

Para obter estes resultados, é necessário definir os termos do vetor de taps do equalizador de forma adequada. Sendo \mathbf{U} o vetor de sinais para cada símbolo, tem-se a equação (3.59)

$$\mathbf{c}^H \mathbf{U} = (0, \dots, 1, \dots, 0) = \mathbf{e}^T \quad (3.59)$$

Portanto, escreve-se a equação (3.60)

$$\mathbf{U}^H \mathbf{c} = \mathbf{e} \quad (3.60)$$

Define-se $\mathbf{c} = \mathbf{U}a$, com $a = (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{e}$. Por fim, define-se os coeficientes do ZF na equação (3.61):

$$\mathbf{c}_{ZF} = \mathbf{U} (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{e} \quad (3.61)$$

A desvantagem da modelagem do ZF reside no fato de que ele não leva em consideração a contribuição do ruído na saída do equalizador para os efeitos de corrompimento do sinal.

O equalizador MMSE atua sobre uma das desvantagens do ZF: na saída do equalizador, é feito um *trade-off* entre os efeitos do ruído e os efeitos da interferência intersimbólica. Diferentemente, do *zero-forcing*, o MMSE não anula a influência de símbolos anteriormente amostrados em um determinado instante de amostragem; seu foco está na redução máxima do erro quadrático médio entre o valor na saída do equalizador e o valor enviado.

O erro quadrático médio é função de cada um dos coeficientes do equalizador. Abaixo, verifica-se que o erro quadrático médio é menor valor esperado para o quadrado da diferença entre o que foi transmitido e o que está na saída do equalizador. O valor quadrático evidencia a minimização entre a diferença das energias de cada sinal. A equação (3.62) indica a forma de cálculo do erro quadrático médio.

$$MSE = J(\mathbf{c}) = \left[|\mathbf{c}^H r[n] - b[n]|^2 \right] \quad (3.62)$$

Os coeficientes do MMSE, dado que os termos são minimizados e não anulados, são calculados de forma diferente daquele do *zero-forcing*, levando em consideração a autocorrelação entre o sinal recebido e a correlação cruzada entre o sinal transmitido e o sinal recebido.

Para máxima confiabilidade do sistema, tem-se que deve valer a identidade apresentada na equação (3.63):

$$b[n] = \mathbf{c}^H r[n] \quad (3.63)$$

Agora com a multiplicação em ambos os lados pelo fator $r[n]$ e considerando os valores esperados, definindo a autocorrelação como $R_{rr} = E \left[r[n] (r[n])^H \right]$ e a correlação cruzada como $R_{rb} = E [b[n] (r[n])]$ e fazendo os ajustes de dimensão matricial necessários, tem-se a equação (3.64):

$$R_{rr} \mathbf{c} = R_{rb} \quad (3.64)$$

Reescrevendo, tem-se a expressão para o correlator dos coeficientes do MMSE na equação (3.65):

$$\mathbf{c}_{MMSE} = R_{rr}^{-1} R_{rb} \quad (3.65)$$

3.4 Conclusões

Neste capítulo, aprofundou-se em um nível maior de detalhes o efeito da interferência das não-idealidades presentes na transmissão de um sistema de comunicação digital em seu desempenho final, tais como o ruído AWGN, a distorção de canal e a introdução de interferência intersimbólica. Além disto, métricas para a visualização dos efeitos desta distorção, como o diagrama de constelação e a curva de BER também foram apresentadas.

Em seguida, o processo de equalização, voltado para a redução dos efeitos da ISI, foi detalhado, junto à apresentação de dois modelos principais de equalizadores convencionais: o zero-forcing e o MMSE, este último sendo de fundamental relevância nos resultados que serão apresentados neste trabalho.

No próximo capítulo, serão apresentados conceitos gerais acerca de redes neurais, o objeto de análise principal nos resultados aqui apresentados.

Capítulo 4

Redes Neurais

4.1 Introdução

Nos capítulos anteriores, apresentou-se um modelo para um sistema de comunicação digital, bem como os processos básicos de seu funcionamento. Além disso, apresentou-se ainda algumas métricas de performance destes sistemas e um aprofundamento no processo de equalização, apresentando as causas e impactos da distorção de canal, o processo de geração de interferência intersimbólica e a conseqüente necessidade de equalização; por fim, descreveu-se o processo de operação de alguns tipos de equalizadores.

Neste capítulo, serão abordados os conceitos principais relacionados ao campo das redes neurais, tais como os componentes básicos de uma rede neural e seu princípio de funcionamento, os tipos de aprendizado possíveis para o treinamento das redes e os principais problemas aos quais o uso destas redes é aplicável, apresentados na Seção 4.2 – Introdução às Redes Neurais. Por fim, na Seção 4.3 – Tipos de Redes Neurais, apresentam-se alguns tipos de redes neurais e seus princípios de funcionamento

4.2 Introdução às Redes Neurais

Redes neurais artificiais são técnicas computacionais formuladas com o intuito de reproduzir a operação do cérebro humano, que apresenta processamento não-linear e paralelo, diferente do processamento convencional dos computadores [20].

A célula-base de uma rede neural – sua unidade básica de processamento - é o neurônio. Basicamente, um neurônio é responsável por produzir uma determinada saída adequada, utilizando uma função de ativação sobre as entradas apresentadas. As entradas são ponderadas por pesos sinápticos e são somadas a um fator de ajuste (*bias*), antes de serem apresentadas à função de ativação. A função de ativação é a responsável pela definição final da classificação da entrada para a saída correspondente.

Para uma rede neural desempenhar sua função, ela deve ser treinada para aprender a apresen-

tar as saídas adequadas, fazendo o ponderamento adequado (ajustando os pesos sinápticos) das entradas. O treinamento de uma rede neural é denominado aprendizado de máquina. É possível dividir o aprendizado de máquina em duas classificações principais:

- **Aprendizado Supervisionado:** neste tipo de aprendizado, há uma pré-classificação das saídas que esperamos que a rede apresente. A rede é treinada com um conhecimento prévio das classificações;
- **Aprendizado Não-Supervisionado:** nesta modalidade, não há informação prévia sobre as características que definem as saídas desejadas para classificação. A rede aprende a realizar a classificação de acordo com o agrupamento de características em comum das entradas.

4.2.1 Aplicações do Aprendizado de Máquina

As redes neurais apresentam versatilidade em seu leque de aplicações. Nesta seção, apresentamos alguns problemas em que elas se apresentam como alternativa [20],[21] e [22].

- **Aprendizado Associativo**

Neste tipo de problema, a rede neural é responsável por armazenar um conjunto de padrões (vetores) que são apresentados a ela. Em seguida, após o armazenamento, apresenta-se à rede uma descrição incompleta ou distorcida/ruidosa de um padrão armazenado e a tarefa da rede consiste em recuperar o padrão originalmente armazenado.

Basicamente, para determinar uma regra de associação, é necessário aprender uma probabilidade condicional da forma $P(X | Y)$, que determina a probabilidade de um evento X ocorrer, dado que o evento Y ocorreu.

Um exemplo para esta aplicação seria a localização de potenciais consumidores de um produto. Dado que consumidores que tipicamente compram X também compram Y , então a partir dos consumidores de X , é possível encontrar um nicho de potenciais consumidores de Y .

- **Classificação**

O problema de classificação é também um problema que envolve a atribuição a padrões. A função do classificador é associar uma entrada a ele apresentada a uma classe; a função utilizada para separar os exemplos de diferentes classes é denominada discriminante.

Tendo a função discriminante, a aplicação principal do problema é a previsão. Com uma função que consegue descrever o comportamento das informações apresentadas anteriormente, é possível prever o comportamento das novas informações apresentadas. Um exemplo de classificação seria o problema do cálculo do risco de empréstimo ao cliente, em que a partir de dados sobre um cliente (como profissão, salário, investimentos e histórico financeiro, por exemplo), torna-se possível inferir se é um risco ou não para o banco realizar empréstimos ao cliente em questão.

- **Regressão**

O problema da regressão consiste na determinação de um estimador de função, em que a partir de um conjunto de dados da variável de entrada e da saída calculada pela função $(x_i, f(x_i))$,

calcula-se a função representativa do comportamento mais aproximado ao da função desejada.

Um exemplo para o problema de regressão seria a aproximação do valor sugerido de venda para uma casa considerando os valores das outras casas em seu condomínio.

- Clusterização

O problema de clusterização consiste no treinamento da rede neural para identificar as características das entradas apresentadas de forma a agrupá-las em clusters com outras entradas que possuem as mesmas características que as entradas em questão. Um exemplo para o problema de clusterização seria o agrupamento de clientes de uma determinada empresa, para determinação do perfil de consumidores que seus produtos usualmente atraem, permitindo o agrupamento de consumidores com características em comum. Este agrupamento auxilia nas questões do relacionamento com o consumidor.

Na próxima seção, breves caracterizações de alguns tipos comuns de redes neurais são apresentadas, de acordo com as demonstrações apresentadas em [20].

4.3 Tipos de Redes Neurais

4.3.1 *Perceptrons* de Camada Única

Este tipo de rede apresenta, como o próprio nome já diz, apenas uma camada de neurônios. Isto significa que as entradas serão processadas por apenas uma camada de neurônios antes da apresentação da saída. O processo de aprendizagem envolve a atualização dos pesos sinápticos (inicializados de forma arbitrária) apresentados à entrada de acordo com um sinal de erro. Sendo $w[n]$ um determinado peso aplicado à entrada para um instante de treinamento e $e[n]$ o erro apresentado na saída, então tem-se, como função-base para a técnica de otimização do *perceptron* de camada única, a expressão para atualização dos pesos expressa na equação (4.1)

$$w[n+1] = w[n] + \eta e[n] \quad (4.1)$$

onde η representa um fator de aprendizado. O aprendizado ocorre até que o erro seja minimizado (algoritmo LMS). Técnicas de otimização comum para esta rede – e que seguem funções similares à equação 4.1 – são: o método da descida mais íngreme, o método de Newton e o método de Gauss-Newton, em ordem crescente de rapidez na convergência do algoritmo [20].

A figura 4.1 apresenta um exemplo de *perceptron* de camada única.

4.3.2 *Perceptron* de Múltiplas Camadas (MLP)

O *perceptron* de múltiplas camadas indica que a entrada será processada por uma ou mais camadas ocultas de neurônios até que sejam apresentadas à camada de saída. Assim, o treinamento da rede deve ser diferente daquele realizado para *perceptrons* que não possuem camadas ocultas.

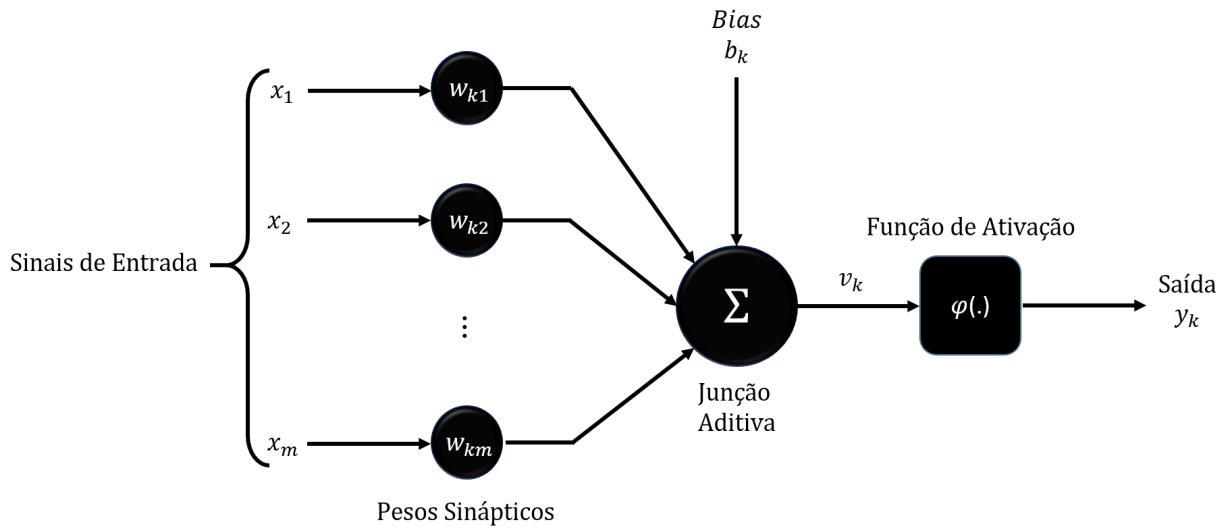


Figura 4.1: Exemplo de *perceptron* de camada única.

Um dos algoritmos mais utilizados nos MLP é o algoritmo de retropropagação, considerado uma generalização do algoritmo LMS utilizado nos *perceptrons* de camada única: ele consiste no acúmulo das mudanças que ocorrem durante uma época de treinamento (passagem completa das entradas aplicadas por todas as camadas do *perceptron*).

A Figura 4.2 apresenta um exemplo de MLP.

Define-se o sinal de erro e_j na saída de um neurônio j como sendo a diferença entre a saída desejada e a saída apresentada, conforme apresentado na equação (4.2).

$$e_j(n) = d_j(n) - y_j(n) \quad (4.2)$$

A energia total do erro $E(n)$ em uma camada de neurônios para uma determinada época n é dada pela equação (4.3)

$$E(n) = \frac{1}{2} \sum e_j^2(n) \quad (4.3)$$

Classifica-se a energia média do erro quadrado como a função de custo. Sua redução significa a redução da energia do erro e, portanto, configura-se como um objetivo almejado. Ela é dada pela média das energias totais de erro para cada época, como apresentado na equação (4.4)

$$E_{mdia} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (4.4)$$

O termos v_j e y_j representam, respectivamente, o campo local induzido na entrada de um neurônio j e sua saída. O campo local induzido é função da multiplicação dos pesos sinápticos que conectam o neurônio à camada anterior a ele pela saída do neurônio i da camada anterior, com

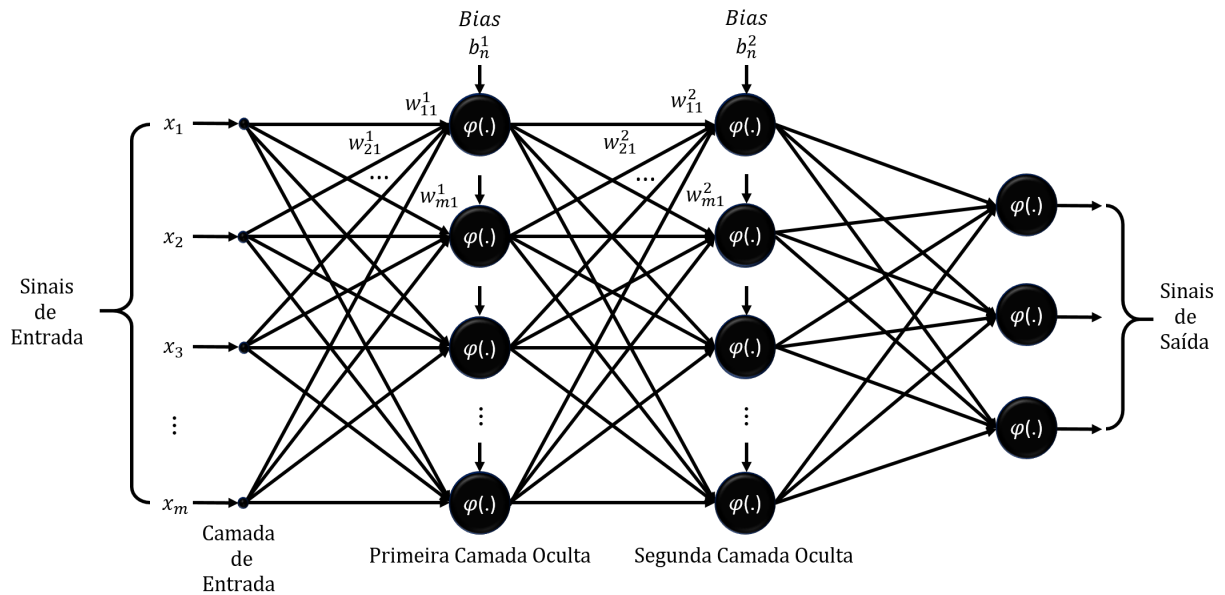


Figura 4.2: Exemplo de MLP com m entradas, duas camadas ocultas com m neurônios cada e três neurônios de saída.

m sendo o número total de entradas (sem considerar o *bias*) aplicadas ao neurônio j . Em uma iteração n , tem-se a equação (4.5):

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (4.5)$$

A saída do neurônio, por sua vez, é função do campo local induzido. Na equação (4.6), f é a função de ativação do MLP. Neste trabalho, a função de ativação utilizada é a tangente hiperbólica. Maiores detalhes serão abordados no próximo capítulo.

$$y_j(n) = f(v_j(n)) \quad (4.6)$$

Com isto, é possível agora definir o fator de sensibilidade, que nos permite executar os ajustes de correção do peso sináptico juntamente a um fator de aprendizado η , seguindo lógica similar à do *perceptron* de camada única. A equação (4.7) representa a derivada da energia do erro com relação aos pesos.

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (4.7)$$

Com a diferenciação de ambos os lados da equação (4.3), tem-se a equação (4.8)

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (4.8)$$

Em seguida, diferenciando-se ambos os lados da equação (4.2), tem-se a equação (4.9)

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (4.9)$$

Diferenciando a equação (4.6) em relação a $v_j(n)$, tem-se a equação (4.10). O apóstrofo em f indica a diferenciação em relação ao argumento e, portanto, apresenta a necessidade de que a função de ativação utilizada seja diferenciável para a aplicabilidade do algoritmo de retropropagação.

$$\frac{\partial y_j(n)}{\partial v_j(n)} = f'(v_j(n)) \quad (4.10)$$

Por fim, a diferenciação da equação (4.5) em relação a $w_{ji}(n)$ produz a equação (4.11)

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad (4.11)$$

Substituindo-se as expressões (4.8) a (4.11) na equação (4.7), tem-se a equação (4.12)

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n) f'(v_j(n)) y_i(n) \quad (4.12)$$

O ajuste dos pesos é, portanto, dado pela regra delta, apresentada na equação (4.13). η é a taxa de aprendizagem do algoritmo de retropropagação e o uso do sinal negativo na equação indica a descida do gradiente no espaço de pesos, isto é, indica a busca por uma direção para a mudança do peso que reduza o valor de $E(n)$.

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (4.13)$$

É possível reescrever a equação (4.13) utilizando a equação (4.12), formando a equação (4.14)

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (4.14)$$

Onde define-se o gradiente local $\delta_j(n)$, responsável por indicar as modificações necessárias para o ajuste dos pesos sinápticos, como expresso na equação (4.15)

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} \quad (4.15)$$

Prossegue-se com a regra da cadeia, formando a equação (4.16)

$$\delta_j(n) = -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \quad (4.16)$$

A partir dos resultados apresentados anteriormente, tem-se a equação (4.17):

$$\delta_j(n) = e_j(n) f'(v_j(n)) \quad (4.17)$$

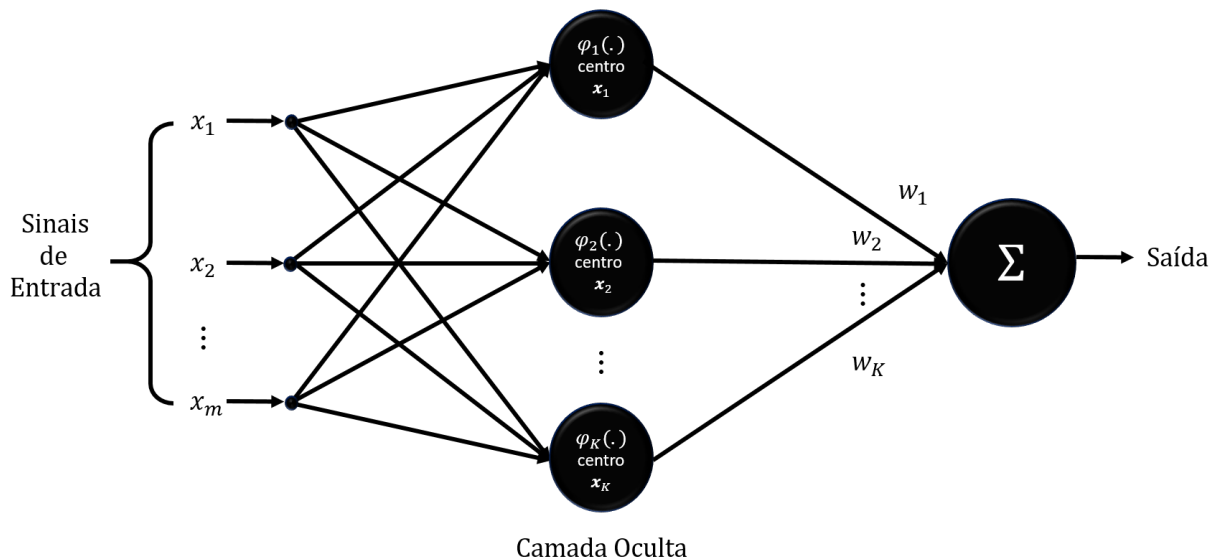


Figura 4.3: Exemplo de rede RBF.

Após o cálculo do fator de ajuste dos pesos, os pesos são reajustados, realiza-se novamente a propagação (geração do sinal de saída a partir dos sinais de entrada) e a retropropagação, até que a energia do erro seja minimizada [20].

4.3.3 Funções Radiais de Base

As redes para funções radiais de base possuem estrutura semelhante à de um MLP, contando com apenas uma camada oculta, com as seguintes características:

- A camada de entrada possui nós de fonte (pontos de entrada de conexão aos neurônios) em quantidade equivalente ao tamanho do vetor de entrada. As conexões de entrada não apresentam pesos sinápticos;

- A amostra de treinamento desta rede é composta de N vetores x ;

- A camada oculta possui um número de neurônios inferior ao tamanho da amostra de treinamento. Cada um deles é descrito por uma função radial de base (para fins de estudo, uma função gaussiana), em que, para um neurônio j , o centro da função será x_j : $f_j(x)$, sendo

$$f_j(x) = f(x - x_j) = e^{\left(-\frac{1}{2\sigma_j^2} \|x - x_j\|^2\right)};$$

- A camada de saída é composta por um neurônio, que recebe todas as saídas da camada oculta ponderadas por um vetor de peso w .

A Figura 4.3 apresenta um exemplo para as redes RBF.

Em suma, cada entrada apresentada é atribuída a um determinado cluster de classificação de acordo com a distância euclidiana entre a entrada e o centro de uma determinada função.

Para o treinamento das redes RBF, utiliza-se o método da Clusterização K-Médias para treinamento das camadas ocultas da rede e o algoritmo RLS para o treinamento da camada de saída.

Por este motivo, diz-se que as redes RBF possuem aprendizagem híbrida [20].

- **Clusterização K -Médias:** a atribuição de uma determinada entrada a um cluster é feita com o uso de um codificador. Este codificador possui uma função de custo, que, novamente, deve ser minimizada. Para um conjunto com N entradas e K clusters, o codificador atribui uma determinada entrada observada a um cluster se sua distância observada até a média estimada daquele cluster for mínima. Assim, sendo a função de custo $J(C)$, tem-se, para $j = C(i)$ e $i = 1, 2, \dots, N$, a equação (4.18)

$$J(C) = \sum_{j=1}^K \sum_{C(i)=j} \|x_j - \mu_j\|^2 \quad (4.18)$$

A função de custo é o somatório das variâncias de cada cluster j resultante das atribuições do codificador a cada cluster, como visto na equação (4.9). Com isto, tem-se a equação (4.19):

$$\sigma_j^2 = \sum_{C(i)=j} \|x_j - \mu_j\|^2 \quad (4.19)$$

- **Estimativa Recursiva dos Mínimos Quadrados (RLS):** para o cálculo do vetor-peso, procede-se a partir de uma equação normal, apresentada na equação (4.20):

$$R(n)\mathbf{w}(n) = r(n) \quad (4.20)$$

Em que $R(n)$, apresentada na equação (4.21), é a função de autocorrelação das saídas das unidades ocultas e $r(n)$, apresentada na equação (4.22) é o vetor de correlação cruzada das saídas das unidades ocultas.

$$R(n) = \sum_{i=1}^n \phi(x_i)\phi^T(x_i) \quad (4.21)$$

$$r(n) = \sum_{i=1}^n \phi(x_i)(i) \quad (4.22)$$

Onde tem-se as funções radiais de base, na equação (4.23):

$$\phi(x_i) = [\varphi(x_i, \mu_1), \varphi(x_i, \mu_2), \dots, \varphi(x_i, \mu_K)]^T \quad (4.23)$$

Em que as funções de cada elemento do vetor são determinadas de acordo com a equação (4.24)

$$\varphi(x_i, \mu_j) = e^{\left(-\frac{1}{2\sigma_j^2} \|x_i - \mu_j\|^2\right)} \quad (4.24)$$

Prossegue-se com as manipulações algébricas para encontrar uma estimativa de atualização para os vetores-peso, com o resultado apresentado na equação (4.25).

$$r(n) = \sum_{i=1}^{n-1} \phi(x_i)(i) + \phi(x_n)(n) = r(n-1) + \phi(x_n)(n) = R(n-1)\mathbf{w}(n-1) + \phi(x_n)(n) \quad (4.25)$$

Adicionando o termo $\phi(n)\phi^T(n)\mathbf{w}(n-1)$, reescreve-se a equação (4.25), obtendo a equação (4.26):

$$r(n) = [R(n-1) + \phi(n)\phi^T(n)] \mathbf{w}(n-1) + \phi(n) [d(n) - \phi^T(n)\mathbf{w}(n-1)] \quad (4.26)$$

Obtém-se a função de correlação, apresentada na equação (4.27):

$$R(n) = R(n-1) + \phi(n)\phi^T(n) \quad (4.27)$$

Define-se $\alpha(n)$ na equação (4.28):

$$\alpha(n) = d(n) - \mathbf{w}^T(n-1)\phi(n) \quad (4.28)$$

A partir da equação (4.28), reescreve-se as equações (4.26) e (4.27), obtendo-se as equações (4.29) e (4.30):

$$r(n) = R(n)\mathbf{w}(n-1) + \phi(n)\alpha(n) \quad (4.29)$$

$$R(n)\mathbf{w}(n) = R(n)\mathbf{w}(n-1) + \phi(n)\alpha(n) \quad (4.30)$$

Finalmente, tem-se a expressão para atualização dos pesos na equação (4.31):

$$\mathbf{w}(n) = \mathbf{w}(n-1) + R(n)^{-1}\phi(n)\alpha(n) \quad (4.31)$$

Dado que a média dos clusters e o vetor-peso são estimativas, estes valores são usualmente representados como estimadores.

4.3.4 Máquina de Vetor de Suporte (SVM)

Máquinas de Vetor de Suporte são capazes de construir um hiperplano de separação entre as classes de forma que a distância entre elas seja a maior possível. O algoritmo principal utilizado é o Método de Kernel: o kernel do produto interno entre um vetor-suporte – retirado da amostra de treinamento – e um vetor do conjunto de vetores de entrada; este algoritmo pode ser utilizado na classificação de padrões e em casos de regressão não-linear.

A definição do hiperplano ótimo de separação envolve a definição de uma margem de separação ρ entre cada classe de definição e o hiperplano que separa as classes. Para a otimização, esta margem de separação deve ser maximizada.

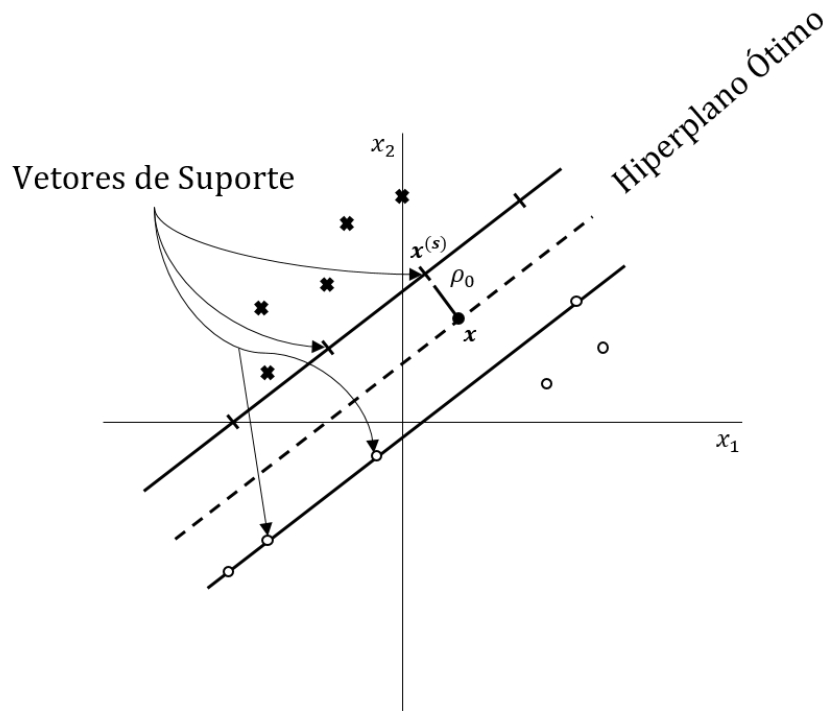


Figura 4.4: Hiperplano ótimo na SVM.

A Figura 4.4 apresenta o hiperplano ótimo utilizado na modelagem das SVM.

Define-se função discriminante $g(x)$, com x sendo as entradas apresentadas, w_0 o vetor peso e b_0 o *bias*, na equação (4.32):

$$g(x) = w_0^T x + b_0 \quad (4.32)$$

A partir da função discriminante, podemos obter os vetores de suporte. Estes vetores são obtidos para os pontos que satisfazem as condições apresentadas nas equações (4.33) e (4.34), para as classes de cada lado do hiperplano.

$$w_0^T x + b_0 \geq 1, d_i = +1 \quad (4.33)$$

$$w_0^T x + b_0 \leq -1, d_i = -1 \quad (4.34)$$

A margem de separação é dada pela equação (4.35):

$$\rho = \frac{2}{\|w_0\|} \quad (4.35)$$

Utiliza-se a otimização quadrática para determinação do hiperplano ótimo, o vetor-peso e o *bias* que satisfazem a equação (4.36) (combinação da equação (4.23)), ao mesmo tempo em que minimizam a função de custo, apresentada na equação (4.37).

$$d_i(w^T x_i + b) \geq 1 \quad (4.36)$$

$$\Phi(w) = \frac{1}{2}w^T w \quad (4.37)$$

• Método dos multiplicadores de Lagrange: para calcular os pesos e o *bias* adequados, prosseguimos com o método dos multiplicadores de Lagrange. Na equação (4.38), tem-se a função de Lagrange, com α sendo os multiplicadores de Lagrange.

$$J(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^N \alpha_i [d_i(w^T x_i + b) - 1] \quad (4.38)$$

Nas equações (4.39), (4.40), (4.41) e (4.42), tem-se as quatro condições para otimização da função:

$$\frac{\partial J(w, b, \alpha)}{\partial w} = 0 \quad (4.39)$$

$$\frac{\partial J(w, b, \alpha)}{\partial b} = 0 \quad (4.40)$$

$$w = \sum_{i=1}^N \alpha_i d_i x_i \quad (4.41)$$

$$\sum_{i=1}^N \alpha_i d_i = 0 \quad (4.42)$$

Todos os multiplicadores que não satisfizerem às condições serão nulos. Sendo assim, os multiplicadores não nulos possíveis serão aqueles que obedecerem à condição apresentada em na equação (4.43):

$$\alpha_i [d_i(w^T x_i + b) - 1] = 0 \quad (4.43)$$

Expande-se a equação (4.38), formando-se a equação (4.44).

$$J(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^N \alpha_i d_i w^T x_i - b \sum_{i=1}^N \alpha_i d_i + \sum_{i=1}^N \alpha_i \quad (4.44)$$

Reescrevem-se os termos função de custo apresentada na equação (4.37), de acordo com a equação (4.41), formando a equação (4.45), com termos i e j diferenciando os termos do vetor \mathbf{w} e do vetor \mathbf{w}^T , respectivamente.

$$w^T w = \sum_{i=1}^N \alpha_i d_i w^T x_i = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j \quad (4.45)$$

As equações (4.44) e (4.45) permitem a definição da função objetivo, apresentada na equação (4.46).

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j x_i^T x_j \quad (4.46)$$

Os multiplicadores de Lagrange devem maximizar a Função Objetivo seguindo as duas condições apresentadas nas equações (4.47) e (4.48), considerando-se $i = 1, 2, \dots, N$

$$\sum_{i=1}^N \alpha_i d_i = 0 \quad (4.47)$$

$$\alpha_i \geq 0 \quad (4.48)$$

Com os multiplicadores determinados para as condições atendidas, é possível encontrar o vetor peso, conforme apresentado nas equações (4.49) e (4.50), e o *bias*, a partir das condições apresentadas na equação (4.51) e das manipulações apresentadas nas equações (4.52) e (4.53), ótimos

$$w = \sum_{i=1}^N \alpha_i d_i x_i \quad (4.49)$$

$$w_0 = \sum_{i=1}^{N_s} \alpha_{0,i} d_i x_i \quad (4.50)$$

$$g(x^{(s)}) = w_0^T x^{(s)} + b_0 = \pm 1 \quad (4.51)$$

$$b_0 = 1 - w_0^T x^{(s)}, d^{(s)} = 1 \quad (4.52)$$

$$b_0 = 1 - \sum_{i=1}^{N_s} \alpha_{0,i} d_i x_i^T x^{(s)} \quad (4.53)$$

A máquina de vetor de suporte pode ser vista como uma máquina de Kernel. Para tanto, definimos a superfície de decisão, com $\phi(x)$ sendo a imagem induzida em um espaço de características pelo vetor de entradas e $\phi(x_i)$ sendo o vetor de características (correspondente ao padrão de entrada x_i em um i -ésimo exemplo; o conjunto $\{\varphi_j(x)\}_{j=1}^m$ representa um conjunto de transformações não-lineares do espaço de entrada para o espaço de características, com dimensão m – basicamente, uma transformação aplicada à entrada pelo vetor peso). Com isto, escreve-se a equação (4.54)

$$w^T \phi(x) = \sum_{i=1}^{N_s} \alpha_i d_i \phi^T(x_i) \phi(x) = 0 \quad (4.54)$$

Em que

$$w = \sum_{i=1}^{N_s} \alpha_i d_i \phi(x_i) \quad (4.55)$$

E

$$\phi(x_i) = [\varphi_1(x_i), \varphi_2(x_i)]^T \quad (4.56)$$

Verifica-se que a equação (4.54) independe do vetor-peso. Portanto, não é necessário calculá-lo para aplicá-la.

O núcleo do produto interno (dado que $\phi^T(x_i)\phi(x)$ é um produto interno entre vetores induzidos no espaço de características – um pelo vetor de entrada x e outro pelo padrão de entrada x_i em um i -ésimo exemplo) é dado pela equação (4.57)

$$k(x, x_i) = \phi^T(x_i)\phi(x) = \sum_{j=1}^{\infty} \phi_j(x_i)\phi_j(x) \quad (4.57)$$

O núcleo do produto interno pode ser utilizado para construir o hiperplano ótimo no espaço de características. O hiperplano é definido pela equação (4.58)

$$\sum_{i=1}^{N_s} \alpha_i d_i k(x, x_i) = 0 \quad (4.58)$$

Os resultados acima são um caso especial do Teorema de Mercer, teorema que determina se um determinado kernel é um produto interno kernel, estando habilitado para uso em uma máquina de vetor de suporte. Além disso, o teorema define o limite do número de kernels admissíveis. Por conta dele, o produto interno kernel é denominado kernel de Mercer.

Expandindo o núcleo $k(x, x')$ em uma série, com λ_i positivo para todo i , tem-se a equação (4.59):

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x)\phi_i(x') \quad (4.59)$$

Esta expressão será válida, se for atendida as condições das equações (4.47) e (4.48). Para todo $\int_b^a \psi^2(x)dx < \infty$, escreve-se a equação (4.60)

$$\int_b^a \int_b^a k(x, x')\psi(x)\psi(x')dx dx' \geq 0 \quad (4.60)$$

Com isto, define-se se um núcleo candidato é um núcleo de produto interno em algum espaço, sendo admissível para uma máquina de vetor de suporte.

4.4 Conclusões

Neste capítulo, apresentou-se um panorama geral sobre redes neurais e seu treinamento por aprendizado de máquina: exemplos clássicos de aprendizado, aplicações do aprendizado de máquina e, por fim, alguns modelos clássicos de redes neurais.

No próximo capítulo, apresentar-se-á uma possibilidade de aplicação do uso do *perceptron* de múltiplas camadas em um problema clássico em sistemas de comunicações digitais, apresentado nos Capítulos 2 e 3: a equalização de canal, com a possibilidade de aplicação de duas técnicas principais de treinamento dos pesos por retropropagação – *Split Complex* e *Fully Complex backpropagation*.

Capítulo 5

Utilização de Redes Neurais no Processo de Equalização

5.1 Introdução

No capítulo anterior, apresentou-se um panorama geral sobre as redes neurais: principais estruturas e suas formas de treinamento, aprendizado de máquina para o treinamento das redes neurais e suas classificações, além de exemplos de aplicações possíveis para a implementação de redes neurais.

Neste capítulo, serão apresentados os fundamentos para a implementação de redes neurais como alternativa para a solução de um problema clássico da transmissão de sinais por sistema de comunicação digital, previamente apresentado: a geração de interferência intersimbólica devido à distorção provocada por canais não ideais.

Na Seção 5.2 – Equalização com o *Perceptron* de Múltiplas Camadas (MLP), apresenta-se a possibilidade de aplicação de uma estrutura comum de rede neural para a equalização de canal, identificando o tipo de aplicação destas redes que condiz com a função de um equalizador. Nas Seções 5.3 – Equalização *Fully Complex* e 5.4 – Equalização *Split*, duas possíveis abordagens para a implementação de um MLP para equalização de canal são demonstradas.

Por fim, apresentam-se os resultados referentes à análise da aplicabilidade do uso de redes neurais para equalização de canais com complexidades crescentes: linear real (canal 1), não linear real (canal 2), linear complexo (canal 3) e não linear complexo (canal 4), assim como o comparativo de desempenho entre as equalizações *Split*, *Fully Complex* e MMSE para os canais 3 e 4.

5.2 Equalização com o *Perceptron* de Múltiplas Camadas (MLP)

Na Seção 4.3, apresentou-se o problema da classificação, em que o classificador tem a função de associar a entrada apresentada a ele a uma determinada classe. Com isto, é razoável fazer um paralelo entre a entrada apresentada a um classificador e a classe a que a entrada é atribuída

com os *bits*/símbolos distorcidos apresentados por um sistema digital e as classes de símbolos possíveis antes da distorção, dado o esquema de modulação utilizado. Desta forma, verifica-se a aplicabilidade do uso de redes neurais para o problema da equalização.

Entre as modelagens de redes neurais apresentadas na Seção 4.3, uma possibilidade para o equalizador é o *perceptron* de múltiplas camadas, dada a facilidade de implementação do algoritmo de retropropagação.

Para a implementação de um equalizador por rede neural, é necessário levar em consideração a natureza da informação que será apresentada à rede. Para modulações binárias, a implementação é mais simples, dado que não há parte complexa na representação do sinal após a modulação; para modulações M -árias em que $M > 2$, no entanto, é necessário adaptar o algoritmo utilizado para treinar a rede. Nas seções 5.2 e 5.3, são apresentadas alternativas para implementação do treinamento da rede considerando um *perceptron* de múltiplas camadas complexo [6].

5.3 Equalização *Fully Complex*

Em [10], identificam-se cinco propriedades desejáveis na escolha de uma função de ativação *fully complex*:

- $f(z) = u(x, y) + jv(x, y)$ é não-linear em x e em y ;
- $f(z)$ é limitada;
- As derivadas parciais u_x , u_y , v_x e v_y existem e são limitadas;
- $f(z)$ é não-inteira;
- $u_x v_y \neq v_x u_y$, em que $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $v_x = \frac{\partial v}{\partial x}$ e $v_y = \frac{\partial v}{\partial y}$.

A última restrição é imposta para garantir aprendizagem contínua, prevenindo uma situação em que uma entrada não-nula aplicada à função de ativação force o gradiente da função de erro (cujo erro é relacionado ao peso complexo) a zero.

Para o caso da função tangente hiperbólica $\tanh(z)$, utilizada no desenvolvimento deste trabalho, esta condição não é necessária, uma vez que a função atende às equações de Cauchy-Riemann. O atendimento a estas equações é condição necessária para que uma função complexa seja analítica em um ponto z .

Para chegar às equações de Cauchy-Riemann, é importante notar que as derivadas parciais de $f(z)$ ao longo dos eixos real e imaginário devem ser iguais, como apresentado na equação (5.1)

$$f'(z) = u_x + jv_x = v_y - ju_y \quad (5.1)$$

Igualando as partes reais e imaginárias de ambos os lados da equação, obtêm-se as equações de Cauchy-Riemann: $u_x = v_y$ e $v_x = -u_y$. Estes resultados permitem escrever $f'(z)$ como a equação (5.2)

$$f'(z) = f_x = -j f_y \quad (5.2)$$

Com o atendimento às equações de Cauchy-Riemann, a restrição $u_x v_y \neq v_x u_y$ pode ser substituída por $u_x^2 \neq u_y^2$. Esta desigualdade é sempre válida, exceto para $u_x = u_y = 0$. Neste caso, dado que as quatro derivadas parciais são reais, $v_x = v_y = 0$. Este evento só ocorre quando o processo de aprendizagem acaba em um extremo local e, portanto, a última condição só se mostra necessária para funções analíticas *split complex*, mas não para funções analíticas *fully complex*.

A seguir, demonstra-se o desenvolvimento do algoritmo de retropropagação *Fully Complex*, de acordo com [13].

Considera-se a função de ativação complexa da equação (5.3) em que $z = x + jy$.

$$f(z) = u(x, y) + jv(x, y) \quad (5.3)$$

O erro na camada de saída é definido por E , conforme a equação (5.4)

$$E = \frac{1}{2} \sum_n |e_n|^2, e_n = d_n - o_n \quad (5.4)$$

Em que d_n é a n -ésima saída esperada na camada de saída e o_n , a n -ésima saída apresentada pelo neurônio na camada de saída. O resultado apresentado pelo neurônio de saída é dado pela equação (5.5)

$$o_n = f(z_n) = u_n + jv_n, z_n = x_n + jy_n = \sum_k W_{nk} X_{nk} \quad (5.5)$$

É possível reescrever a entrada z_n aplicada à função de ativação, expandindo-se os pesos e as entradas de cada neurônio em parte real e parte imaginária, conforme a equação (5.6)

$$z_n = \sum_k (W_{nkR} + jW_{nkI})(X_{nkR} + jX_{nkI}) \quad (5.6)$$

A regra de adaptação de pesos da retropropagação requer o gradiente. Desta forma, o gradiente da função erro com respeito aos componentes real e imaginário dos pesos W_{nk} pode ser escrita como apresentado na equação (5.7)

$$\frac{\partial E}{\partial W_{nk}} = \nabla_{w_{nk}} E = \frac{\partial E}{\partial W_{nkR}} + j \frac{\partial E}{\partial W_{nkI}} \quad (5.7)$$

Ao aplicar a regra da cadeia ao gradiente, têm-se as equações (5.8) e (5.9)

$$\frac{\partial E}{\partial W_{nkR}} = \frac{\partial E}{\partial u_n} \left(\frac{\partial u_n}{\partial x_n} \frac{\partial x_n}{\partial W_{nkR}} + \frac{\partial u_n}{\partial y_n} \frac{\partial y_n}{\partial W_{nkR}} \right) + \frac{\partial E}{\partial v_n} \left(\frac{\partial v_n}{\partial x_n} \frac{\partial x_n}{\partial W_{nkR}} + \frac{\partial v_n}{\partial y_n} \frac{\partial y_n}{\partial W_{nkR}} \right) \quad (5.8)$$

$$\frac{\partial E}{\partial W_{nkI}} = \frac{\partial E}{\partial u_n} \left(\frac{\partial u_n}{\partial x_n} \frac{\partial x_n}{\partial W_{nkI}} + \frac{\partial u_n}{\partial y_n} \frac{\partial y_n}{\partial W_{nkI}} \right) + \frac{\partial E}{\partial v_n} \left(\frac{\partial v_n}{\partial x_n} \frac{\partial x_n}{\partial W_{nkI}} + \frac{\partial v_n}{\partial y_n} \frac{\partial y_n}{\partial W_{nkI}} \right) \quad (5.9)$$

Definindo-se o gradiente $\delta_n = -\frac{\partial E}{\partial u_n} - j\frac{\partial E}{\partial v_n}$, em que $\delta_{nR} = -\frac{\partial E}{\partial u_n}$ e $\delta_{nI} = -\frac{\partial E}{\partial v_n}$, e aplicando as derivadas parciais identificáveis das equações (5.5) e (5.6), também é possível definir as equações (5.10)

$$\frac{\partial x_n}{\partial W_{nkR}} = X_{nkR}, \frac{\partial y_n}{\partial W_{nkR}} = X_{nkI}, \frac{\partial x_n}{\partial W_{nkI}} = -X_{nkI}, \frac{\partial y_n}{\partial W_{nkI}} = X_{nkR} \quad (5.10)$$

Com isto, reescrevem-se as expressões para o gradiente, como visto nas equações (5.11) e (5.12)

$$\frac{\partial E}{\partial W_{nkR}} = -\delta_{nR} \left(\frac{\partial u_n}{\partial x} X_{nkR} + \frac{\partial u_n}{\partial y} X_{nkI} \right) - \delta_{nI} \left(\frac{\partial v_n}{\partial x} X_{nkR} + \frac{\partial v_n}{\partial y} X_{nkI} \right) \quad (5.11)$$

$$\frac{\partial E}{\partial W_{nkI}} = -\delta_{nR} \left(\frac{\partial u_n}{\partial x} (-X_{nkI}) + \frac{\partial u_n}{\partial y} X_{nkR} \right) - \delta_{nI} \left(\frac{\partial v_n}{\partial x} (-X_{nkI}) + \frac{\partial v_n}{\partial y} X_{nkR} \right) \quad (5.12)$$

Combinando-se as equações do gradiente real e o do componente imaginário, é possível simplificar o gradiente da função erro para a equação (5.13)

$$\nabla w_{nk} E = -\bar{X}_{nk} \left\{ \left(\frac{\partial u_n}{\partial x} + j \frac{\partial u_n}{\partial y} \right) \delta_{nR} + \left(\frac{\partial v_n}{\partial x} + j \frac{\partial v_n}{\partial y} \right) \delta_{nI} \right\} \quad (5.13)$$

Ainda, é possível apresentar uma expressão ainda mais compacta para o gradiente da função erro, aplicando as equações de Cauchy-Riemann e reduzindo os componentes real e imaginário uma forma geral com a forma derivativa apresentada na equação (5.2), como vê-se na equação (5.14)

$$\nabla w_{nk} E = -\bar{X}_{nk} \left\{ \left(\frac{\partial \bar{f}}{\partial x} \right) \delta_{nR} - j \left(-j \frac{\partial \bar{f}}{\partial y} \right) \delta_{nI} \right\} = -\bar{X}_{nk} \bar{f}'_{nk} \bar{\delta}(n) \quad (5.14)$$

A atualização dos pesos complexos ΔW_{nk} é proporcional ao gradiente negativo, em que α é uma taxa de aprendizado constante real positiva, conforme a equação (5.15)

$$\Delta W_{nk} = \alpha \bar{X}_{nk} \bar{f}'_{nk} \bar{\delta}_n \quad (5.15)$$

Quando o peso complexo pertence ao neurônio de saída, o gradiente é dado pela equação (5.16)

$$\delta_n = e_n = d_n - o_n \quad (5.16)$$

Quando W_{mk} pertence à m -ésima camada oculta, a entrada da rede z_m ao neurônio m é dada pela equação (5.17).

$$z_m = x_m + jy_m = \sum_k (u_k + jv_k)(W_{mkR} + jW_{mkI}) \quad (5.17)$$

Em que o índice k pertence a todo neurônio que alimenta o neurônio m . Com a aplicação da regra da cadeia, tem-se a equação (5.18) para a parte real do gradiente δ_m

$$\delta_{mR} = -\frac{\partial E}{\partial u_m} = -\sum_k \frac{\partial E}{\partial u_k} \left(\frac{\partial u_k}{\partial x_k} \frac{\partial x_k}{\partial u_m} + \frac{\partial u_k}{\partial y_k} \frac{\partial y_k}{\partial u_m} \right) - \sum_k \frac{\partial E}{\partial v_k} \left(\frac{\partial v_k}{\partial x_k} \frac{\partial x_k}{\partial u_m} + \frac{\partial v_k}{\partial y_k} \frac{\partial y_k}{\partial u_m} \right) \quad (5.18)$$

Substituindo-se $-\frac{\partial E}{\partial u_m} = \delta_{mR}$, $\frac{\partial x_k}{\partial u_m} = W_{mkR}$, $\frac{\partial y_k}{\partial u_m} = W_{mkI}$ e $-\frac{\partial E}{\partial v_k} = \delta_{kI}$, tem-se a equação (5.19)

$$\delta_{mR} = -\frac{\partial E}{\partial u_m} = -\sum_k \delta_{kR} \left(\frac{\partial u_m}{\partial x_k} W_{mkR} + \frac{\partial u_m}{\partial y_k} W_{mkI} \right) - \sum_k \delta_{kI} \left(\frac{\partial v_m}{\partial x_k} W_{mkR} + \frac{\partial v_m}{\partial y_k} W_{mkI} \right) \quad (5.19)$$

Para a parte imaginária do gradiente δ_m , tem-se a equação (5.20)

$$\delta_{mI} = -\frac{\partial E}{\partial v_m} = -\sum_k \frac{\partial E}{\partial u_k} \left(\frac{\partial u_k}{\partial x_k} \frac{\partial x_k}{\partial v_m} + \frac{\partial u_k}{\partial y_k} \frac{\partial y_k}{\partial v_m} \right) - \sum_k \frac{\partial E}{\partial v_k} \left(\frac{\partial v_k}{\partial x_k} \frac{\partial x_k}{\partial v_m} + \frac{\partial v_k}{\partial y_k} \frac{\partial y_k}{\partial v_m} \right) \quad (5.20)$$

Com as substituições feitas para a parte real e com $\frac{\partial x_k}{\partial v_m} = -W_{kmI}$ e $\frac{\partial y_k}{\partial v_m} = W_{kmR}$, tem-se a equação (5.21)

$$\delta_{mI} = -\frac{\partial E}{\partial v_m} = -\sum_k \delta_{kR} \left(\frac{\partial u_k}{\partial x_k} (-W_{kmI}) + \frac{\partial u_k}{\partial y_k} W_{kmR} \right) - \sum_k \delta_{kI} \left(\frac{\partial v_k}{\partial x_k} (-W_{kmI}) + \frac{\partial v_k}{\partial y_k} W_{kmR} \right) \quad (5.21)$$

Combinando-se δ_{mR} e δ_{mI} , tem-se a equação (5.22) para o gradiente δ_m

$$\delta_m = \delta_{mR} + j\delta_{mI} = \sum_k \bar{W}_{km} \bar{f}'_m(z_m) \bar{\delta}_k \quad (5.22)$$

5.4 Equalização *Split*

A equalização do tipo *Split* consiste em uma alternativa mais simples e convencional para implementação do equalizador por rede neural.

Em termos estruturais, consiste na formulação de duas redes neurais separadas: uma para a parte real do símbolo recebido e outra para sua parte imaginária. Após a equalização, somam-se as saídas de cada uma das redes de forma a reconstituir o sinal equalizado na forma $z = x + jy$, para processamento do sinal no próximo bloco do sistema digital.

Esta alternativa simplifica a operação necessária do algoritmo de retropropagação, dado que cada rede neste caso recebe somente uma entrada e, com isto, a função de ativação atua somente sobre uma variável. Esta simplificação poderia trazer vantagens sobre a equalização com relação ao algoritmo *Fully Complex*, que aplica funções de duas variáveis.

Em comparação com a função de ativação *fully complex*, a função de ativação *split complex* assume a forma $f(z) = u(x) + jv(y)$, o que indica que $u_y = v_x = 0$ na composição do algoritmo de retropropagação *Split* [13].

Portanto, escreve-se a equação (5.23) para o algoritmo de atualização dos pesos

$$\Delta W_{nk} = \alpha \bar{X}_{nk} \left(\frac{\partial u_n}{\partial x} \delta_{nR} + j \frac{\partial v_n}{\partial y} \delta_{nI} \right) \quad (5.23)$$

Assim como no algoritmo *Fully Complex*, para o neurônio de saída vale a equação (5.14), em que $\delta_n = e_n = d_n - o_n$. Para as camadas de entrada e oculta, tem-se a equação (5.24)

$$\delta_m = \sum_k \bar{W}_{lm} \left(\frac{\partial u_m}{\partial x_k} \delta_{kR} + j \frac{\partial v_m}{\partial y_k} \delta_{kI} \right) \quad (5.24)$$

5.5 Resultados

Esta seção apresenta os resultados apresentados pelas redes neurais e equalizadores implementados, comparando seu desempenho para diferentes modulações e canais dispersivos.

Vale ressaltar que o produto principal deste projeto é um dispositivo de equalização, que facilite o processo de demodulação; não há, portanto, alteração no demodulador.

Além disto, utilizou-se a modulação do tipo PSK devido ao fato de o processo de classificação para este esquema de modulação ser mais delicado, dada a proximidade dos símbolos entre si e sua equidistância do centro do diagrama de constelação, tornando a análise dos resultados mais interessante.

A Figura 5.1 apresenta o sistema de comunicação digital implementado para as simulações.

A Figura 5.2 apresenta a estrutura implementada no MATLAB para o equalizador MLP *Fully Complex*, ao passo que a Figura 5.3 apresenta a estrutura implementada no MATLAB para o equalizador MLP *Split*, em que x_n e y_n são as componentes em fase e em quadratura do sinal modulado e \bar{x}_n e \bar{y}_n são as componentes em fase e em quadratura do sinal equalizado.

É importante notar que os termos de atraso utilizados nas entradas das redes são necessários para as convoluções presentes nas funções representativas de cada canal.

Os canais utilizados foram:

- Canal 1 - Linear Real: $\bar{s}(n) = [(0.707)s(n) + (0.101)s(n-1)]$
- Canal 2 - Não-linear Real: $\bar{s}(n) = [(0.707)s(n) + (0.101)s(n-1)] + 0.2 [(0.707)s(n) + (0.101)s(n-1)]^3$

- Canal 3 - Linear Complexo: $\bar{s}(n) = [(0.707 + j0.707)s(n) + (0.101 - j0.101)s(n - 1)]$
- Canal 4 - Não-linear Complexo: $\bar{s}(n) = [(0.707 + j0.707)s(n) + (0.101 - j0.101)s(n - 1)] + 0.2 [(0.707 + j0.707)s(n) + (0.101 - j0.101)s(n - 1)]^3$

É interessante observar que os termos sobre cada um dos símbolos indicam a potência que aquele símbolo apresenta. Desta forma, quanto maior o termo que multiplica um determinado símbolo, maior sua potência na transmissão; o caso ideal seria aquele em que a potência do símbolo desejado ($s(n)$) seria maximizada, ao passo em que a potência do símbolo da interferência intersimbólica ($s(n - 1)$) seria minimizada. Portanto, o processo de equalização do canal apresenta seu melhor desempenho quando as potência dos símbolos apresenta comportamento próximo ao caso ideal descrito.

Os resultados das distorções dos canais nas constelações de cada modulação são apresentados nas Figuras 5.4 a 5.19, para as duas análises comparativas apresentadas.

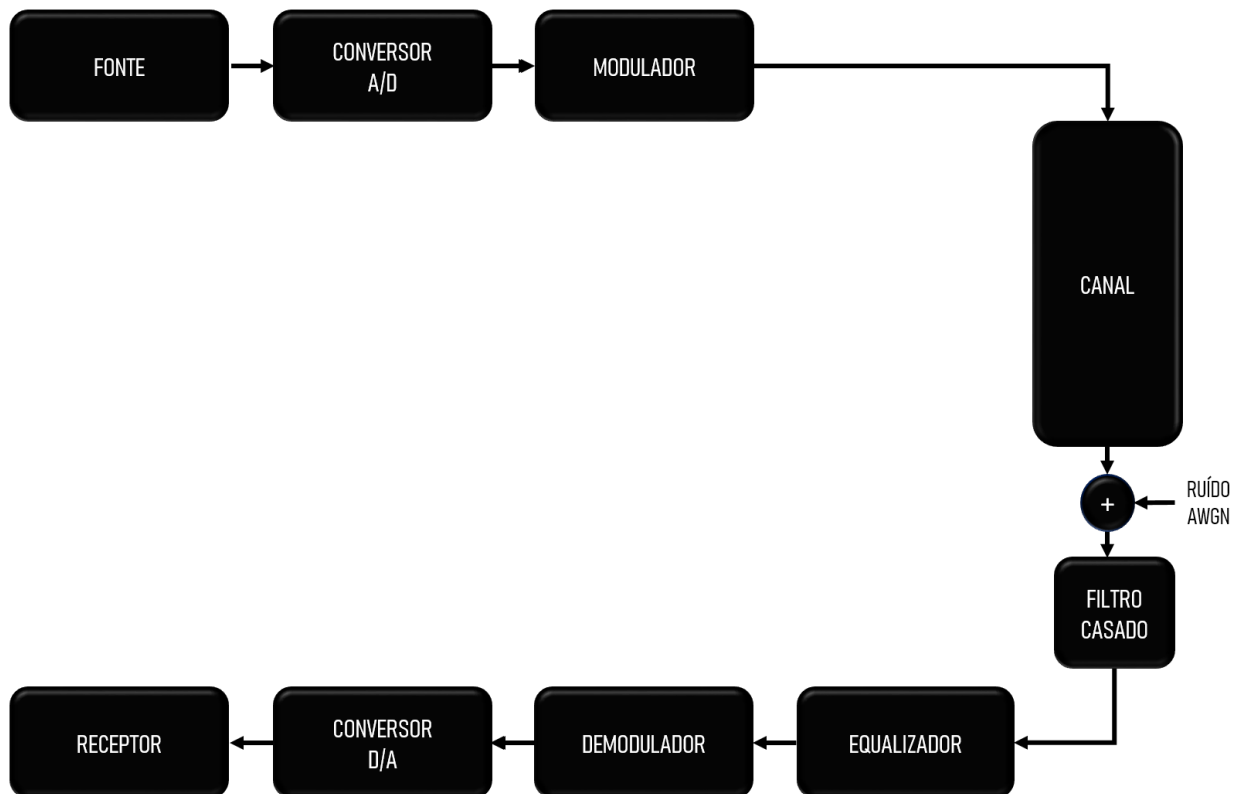


Figura 5.1: Sistema implementado nas simulações.

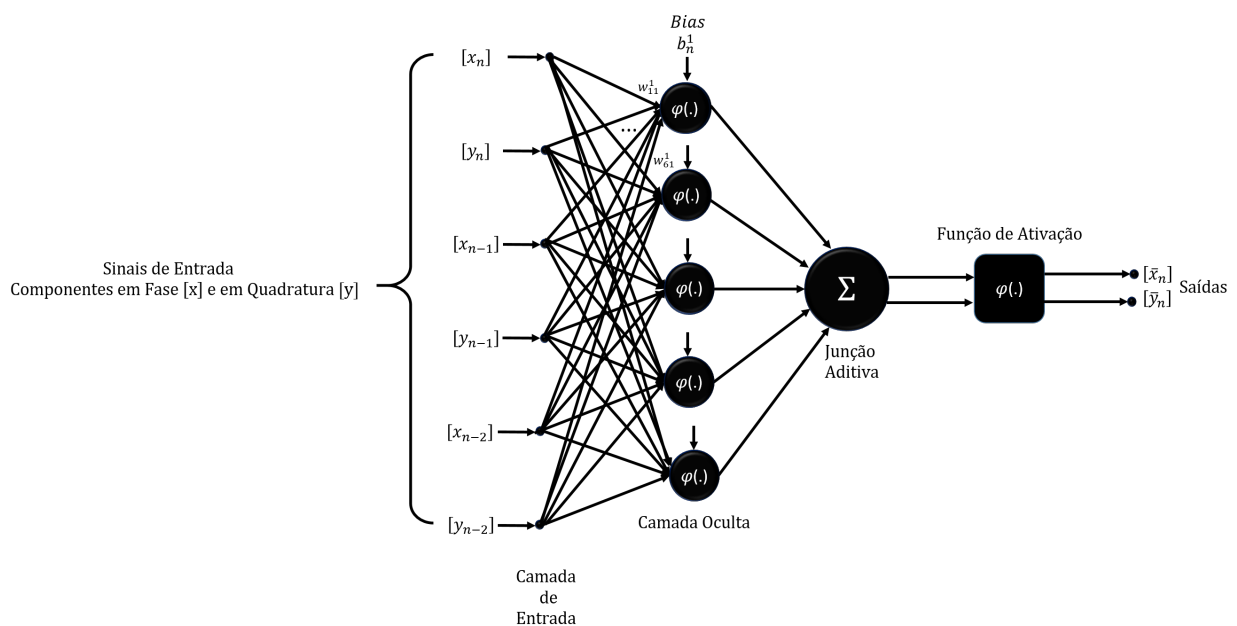


Figura 5.2: Estrutura implementada no MATLAB para o equalizador MLP *Fully Complex*, considerando uma camada oculta com cinco neurônios.

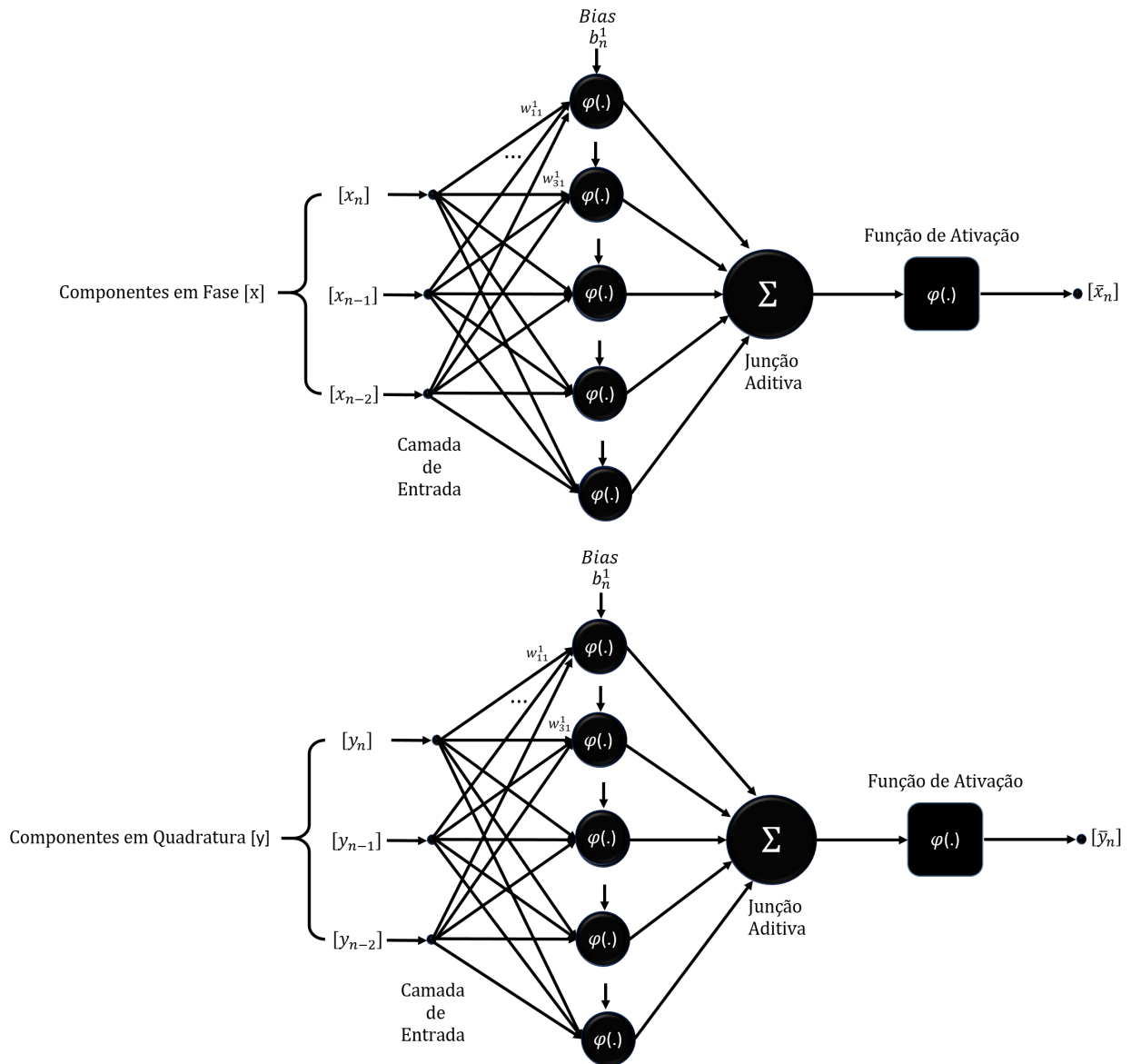


Figura 5.3: Estrutura implementada no MATLAB para o equalizador MLP *Split*, considerando uma camada oculta com cinco neurônios.

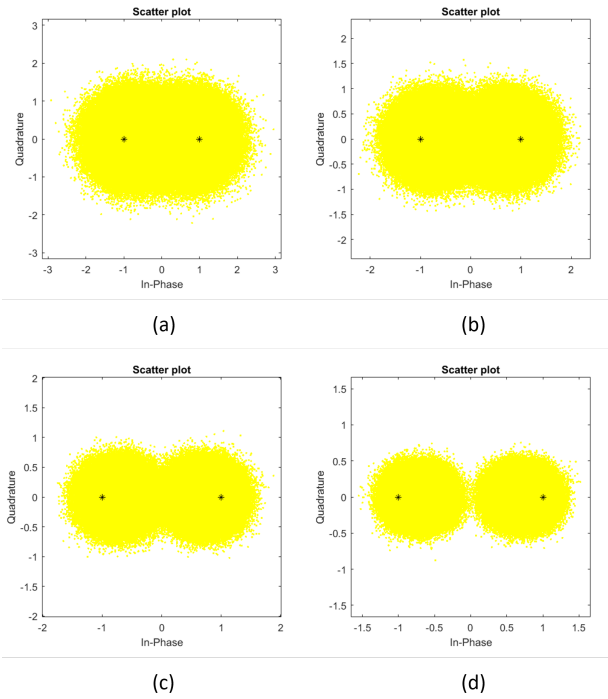


Figura 5.4: Constelação BPSK - Canal 1: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

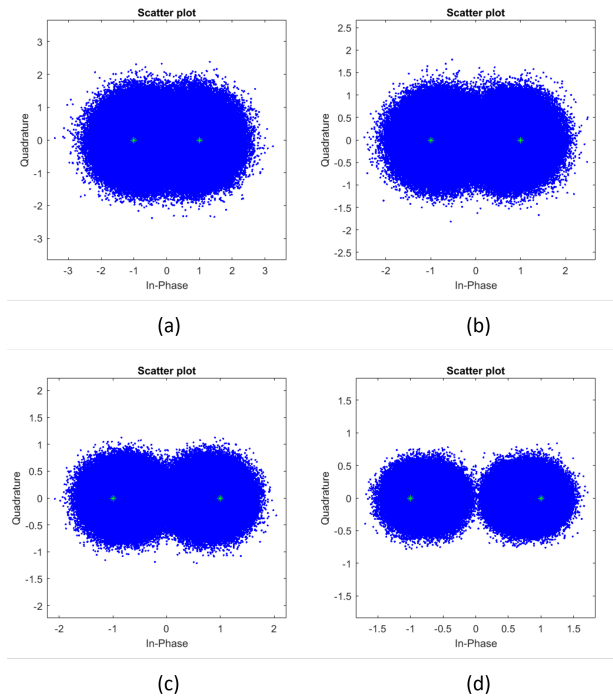


Figura 5.5: Constelação BPSK - Canal 2: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

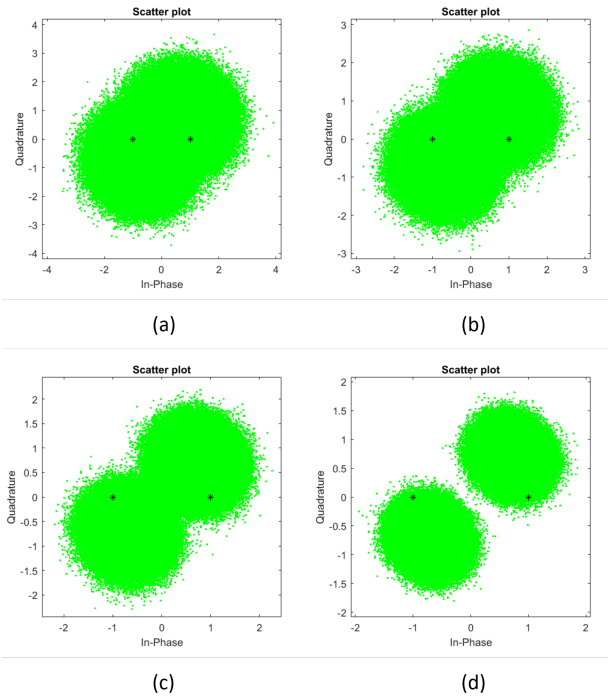


Figura 5.6: Constelação BPSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

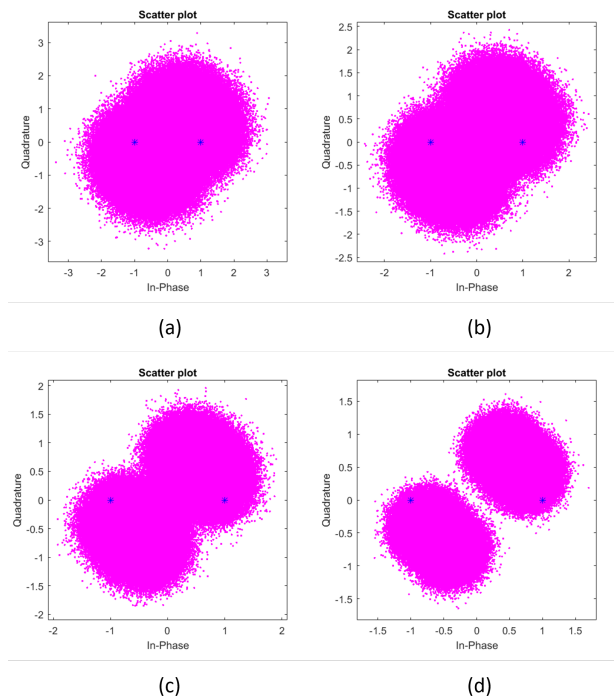


Figura 5.7: Constelação BPSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

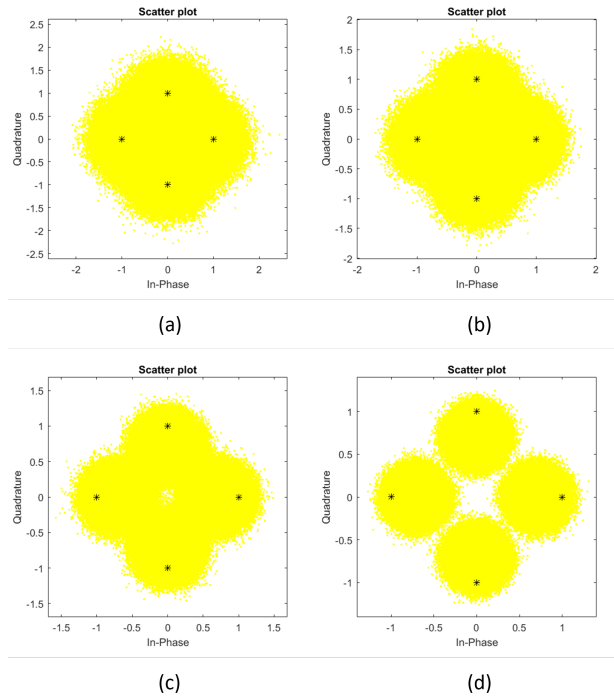


Figura 5.8: Constelação QPSK - Canal 1: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

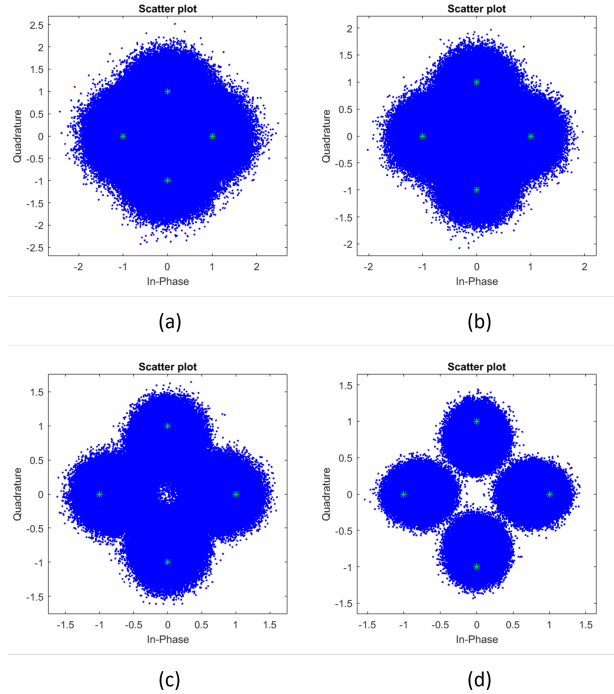


Figura 5.9: Constelação QPSK - Canal 2: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

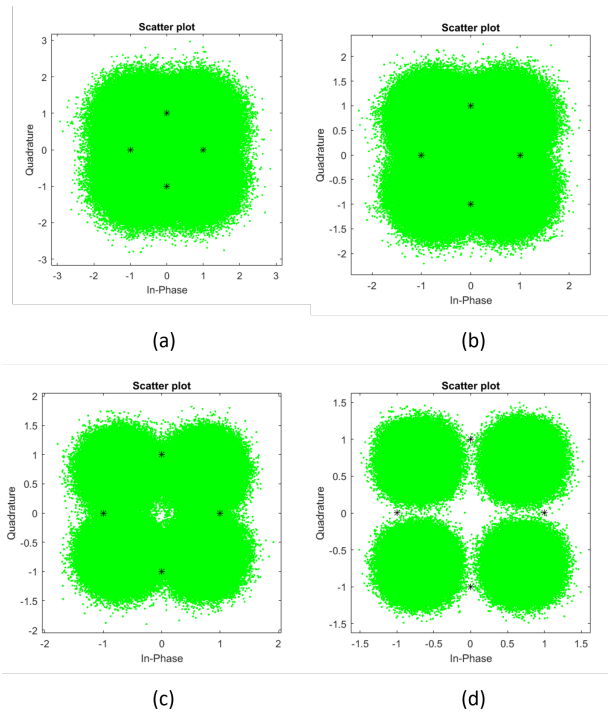


Figura 5.10: Constelação QPSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

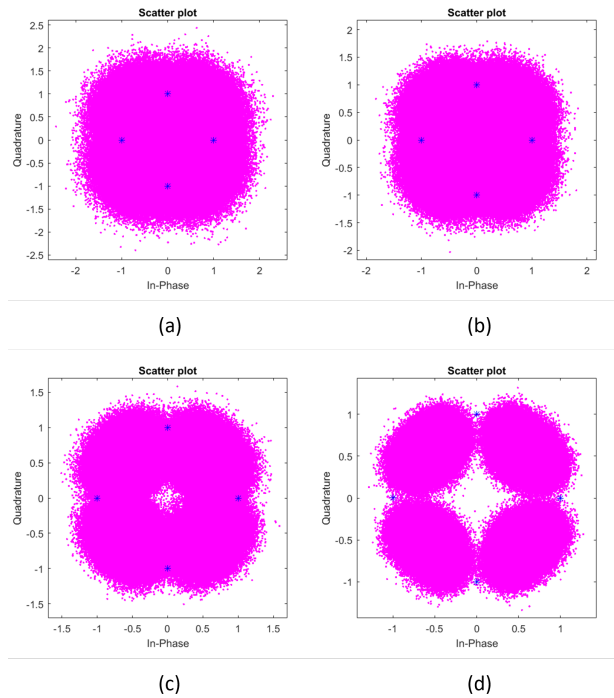


Figura 5.11: Constelação QPSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

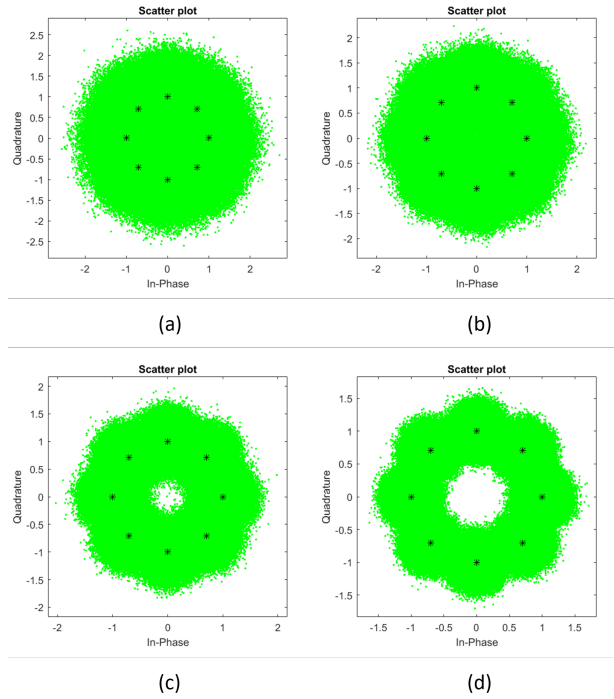


Figura 5.12: Constelação 8-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

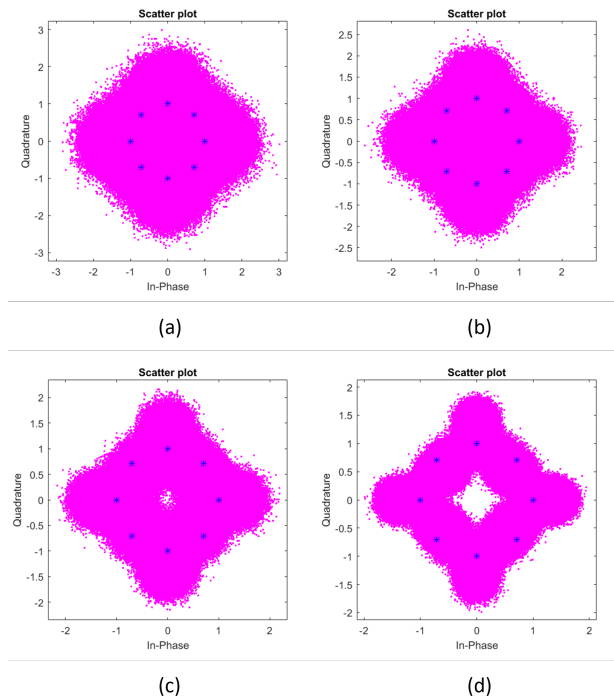


Figura 5.13: Constelação 8-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

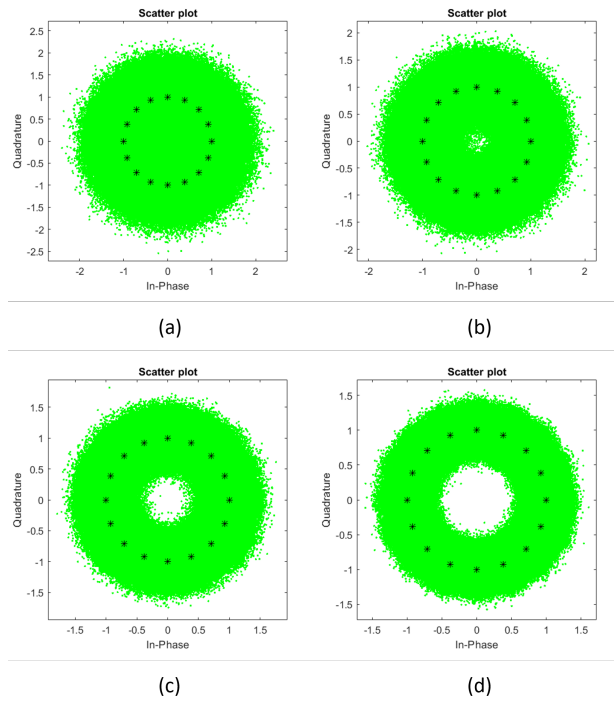


Figura 5.14: Constelação 16-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

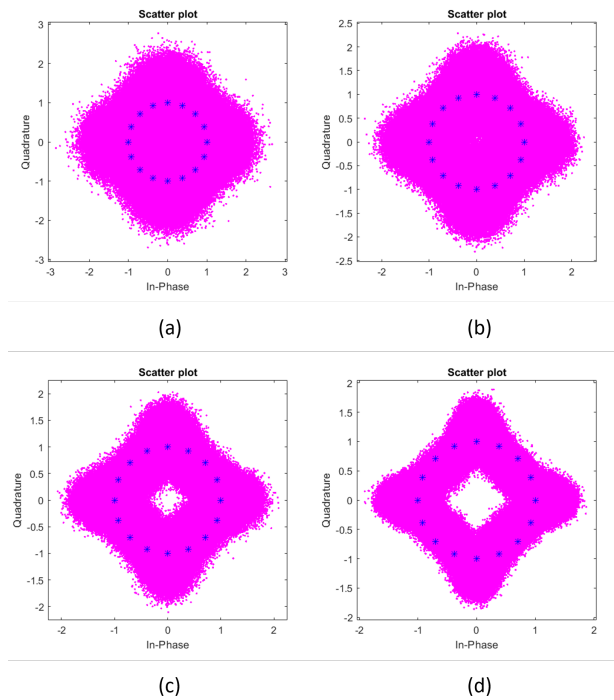


Figura 5.15: Constelação 16-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

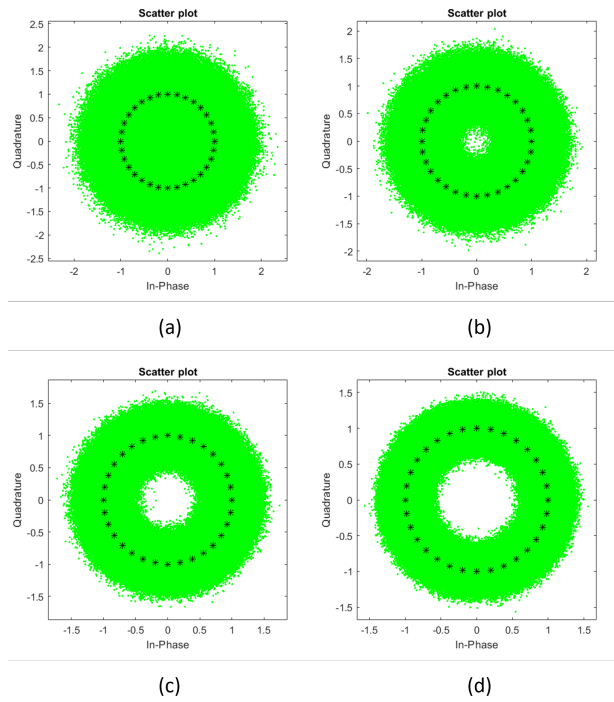


Figura 5.16: Constelação 32-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

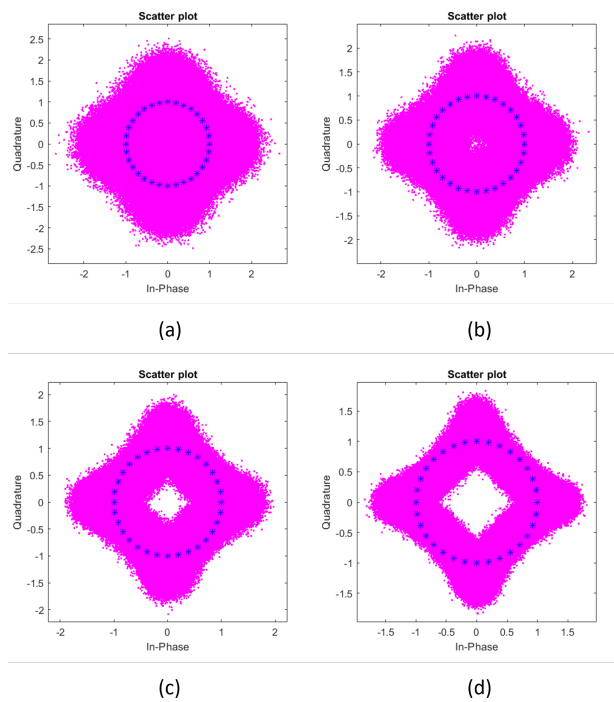


Figura 5.17: Constelação 32-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

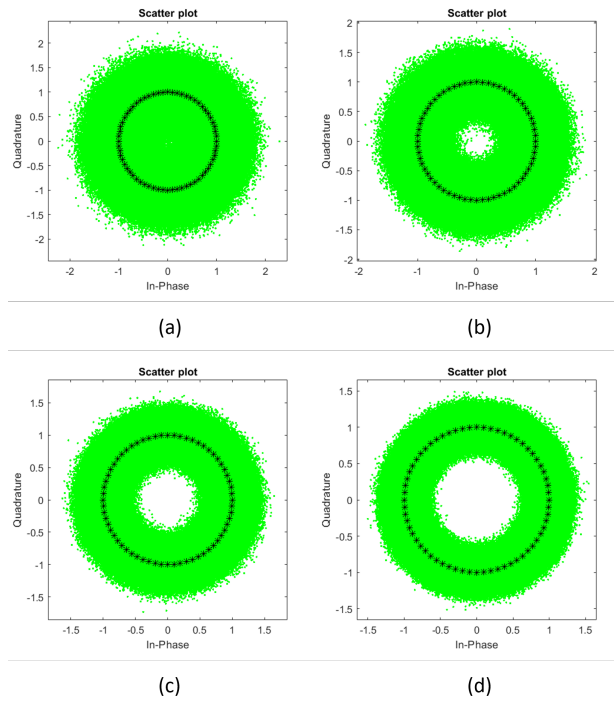


Figura 5.18: Constelação 64-PSK - Canal 3: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

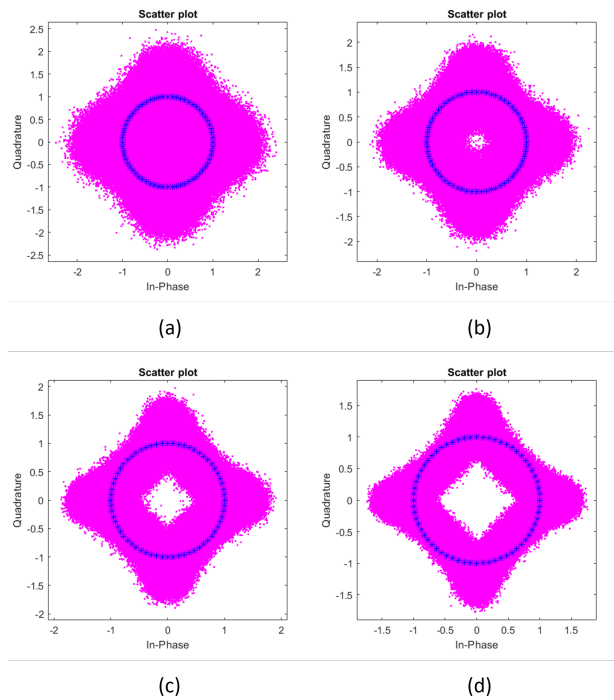


Figura 5.19: Constelação 64-PSK - Canal 4: (a) $E_b/N_0 = 1\text{dB}$; (b) $E_b/N_0 = 4\text{dB}$; (c) $E_b/N_0 = 7\text{dB}$ e (d) $E_b/N_0 = 10\text{dB}$.

É importante lembrar que o sinal distorcido apresentado ao equalizador é resultado da convolução do sinal distorcido com a função de transferência do canal dispersivo. A convolução de duas funções é dada pela equação (5.25)

$$x_1(t) * x_2(t) = \int_{-\infty}^{\infty} x_1(\tau) \cdot x_2(t - \tau) d\tau \quad (5.25)$$

Uma propriedade da convolução é a propriedade distributiva, dada pela equação (5.26)

$$x_1 * (x_2 + x_3) = (x_1 * x_2) + (x_1 * x_3) \quad (5.26)$$

Ao observar as características do sinal transmitido, que possui componentes em fase e componentes em quadratura, analisando-se a aplicação da propriedade distributiva da convolução na distorção provocada pelo canal, torna-se evidente a relação entre cada uma das componentes do sinal e as componentes $s(n)$ e $s(n - 1)$ da função de cada canal (estas últimas ponderadas por termos reais ou complexos). Esta relação é fundamental para as tomadas de decisão acerca da estrutura dos equalizadores a serem utilizados no projeto de um sistema de comunicação, como será visualizado na análise dos resultados apresentados na seção 5.5.2.

Tendo isto, é possível analisar os diferentes impactos de cada um dos canais nas constelações a seguir. As análises feitas apresentam melhor visibilidade para as modulações BPSK e QPSK, dada a quantidade reduzida de símbolos. Além disto, somente estes dois esquemas de modulação foram analisados para os quatro canais, considerando os intuitos da análise apresentada na seção 5.5.1. Por estes motivos, as análises apresentadas para as constelações serão feitas para o BPSK e o QPSK.

Comparando-se as Figuras 5.4 e 5.8 com as Figuras 5.5 e 5.9, verifica-se que a introdução da não-linearidade no Canal 2, em comparação com o Canal 1, provoca aumento no diâmetro da nuvem de distorção ao redor de cada um dos símbolos das constelações.

A introdução dos termos complexos no Canal 3 produz os diagramas de constelação apresentados nas Figuras 5.6 e 5.10. Quando comparado às Figuras 5.4, 5.8, 5.5 e 5.9, verifica-se uma rotação do conjunto de nuvens ao redor dos símbolos no sentido anti-horário.

A introdução da não-linearidade no Canal 4, por sua vez, apresentada nas Figuras 5.7 e 5.11, quando comparada às Figuras 5.6 e 5.10, provoca uma deformação nas nuvens de distorção ao redor dos símbolos, fazendo com que assumam um formato de elipse.

Por fim, é importante observar que a alteração no formato destas nuvens, bem como a rotação de suas posições no plano, é diretamente proporcional ao aumento na complexidade da equalização, dado que este processo depende do reconhecimento de padrões entre os símbolos distorcidos para classificação, com base na comparação com os símbolos transmitidos. Em suma, quanto maiores e mais significativas as alterações de forma e as rotações, maiores as dificuldades nas comparações com os símbolos não-distorcidos.

Tabela 5.1: Tabela de Parâmetros BPSK/QPSK- Rede Neural *Fully Complex*: Descida do Gradiente Adaptativo

| Parâmetro | Valor/Nome |
|---|-----------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 2500 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>traingda</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.2: Tabela de Parâmetros BPSK/QPSK - MMSE: Descida do Gradiente Adaptativo

| Parâmetro | Valor/Nome |
|------------------------|------------|
| Número de Coeficientes | 8 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

5.5.1 Análise comparativa entre o desempenho de redes neurais utilizando *perceptrons* de múltiplas camadas e o equalizador MMSE no processo de equalização

Nesta seção, apresentam-se os resultados obtidos para o processo de equalização utilizando um MLP implementado na *toolbox* de redes neurais do MATLAB inspirado no MLP *Fully Complex* apresentado na seção 5.2, comparando os resultados com um equalizador MMSE tradicional com número de coeficientes próximo ao número de neurônios das rede.

A Tabela 5.1 apresenta os parâmetros da rede neural inspirada no MLP *Fully Complex*, ao passo que a Tabela 5.2 apresenta os parâmetros do MMSE. Todos os parâmetros foram utilizados tanto para a modulação BPSK quanto para a modulação QPSK, assim como para todos os canais utilizados.

Os resultados das simulações são apresentados nas Figuras 5.20 a 5.27

Nestes resultados, verifica-se que para os dois esquemas de modulação utilizados e para sua aplicação aos quatro canais descritos, o desempenho do equalizador *Fully Complex* é superior ao desempenho do equalizador clássico MMSE sobre o efeito de distorção dos canais, apresentando curva de BER com características mais próximas à curva de BER teórica apresentada (que considera somente o efeito do ruído na distorção do sinal, desconsiderando a distorção de canal).

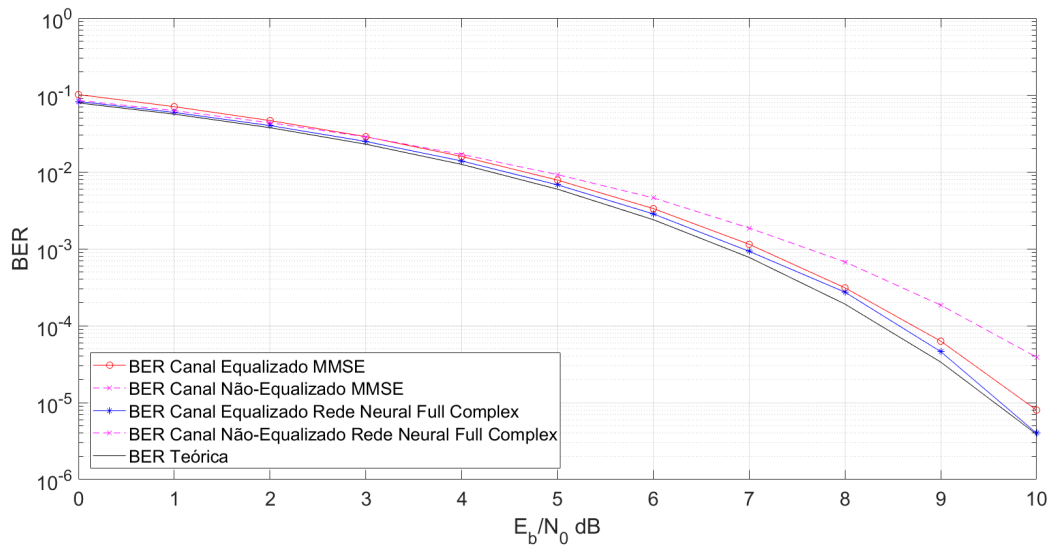


Figura 5.20: Curva BER da modulação BPSK para o Canal 1 - Comparativo *Fully Complex* e MMSE.

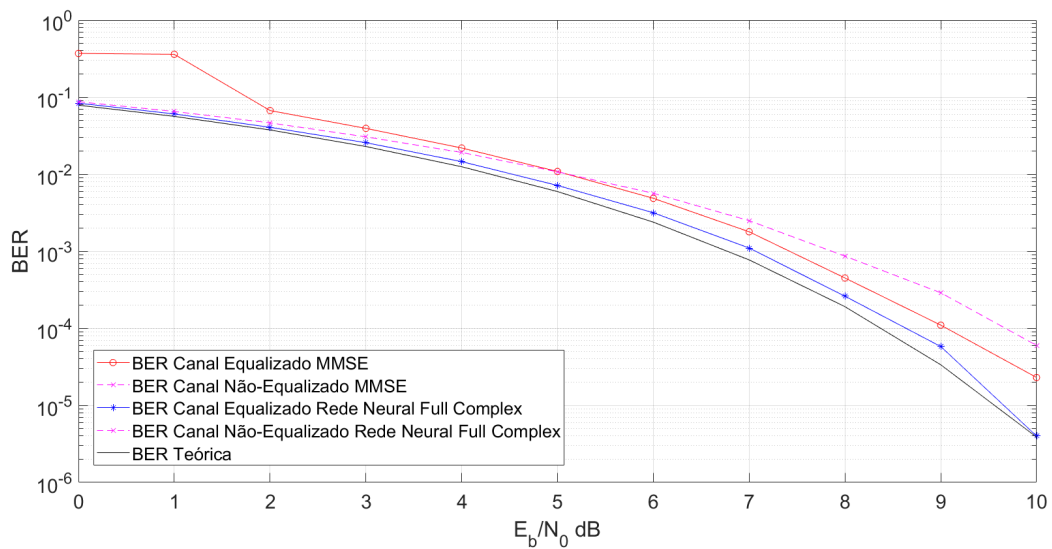


Figura 5.21: Curva BER da modulação BPSK para o Canal 2 - Comparativo *Fully Complex* e MMSE.

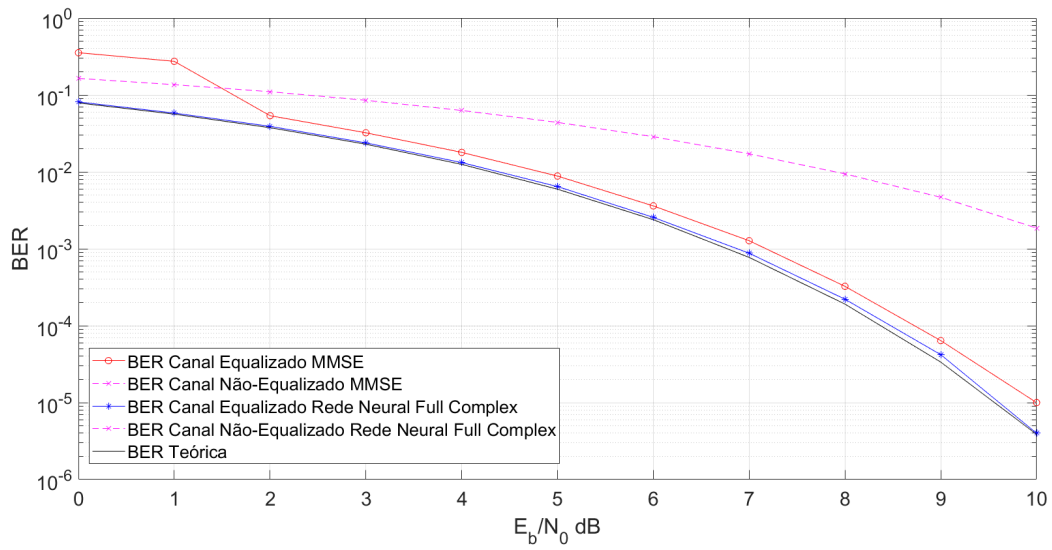


Figura 5.22: Curva BER da modulação BPSK para o Canal 3 - Comparativo *Fully Complex* e MMSE.

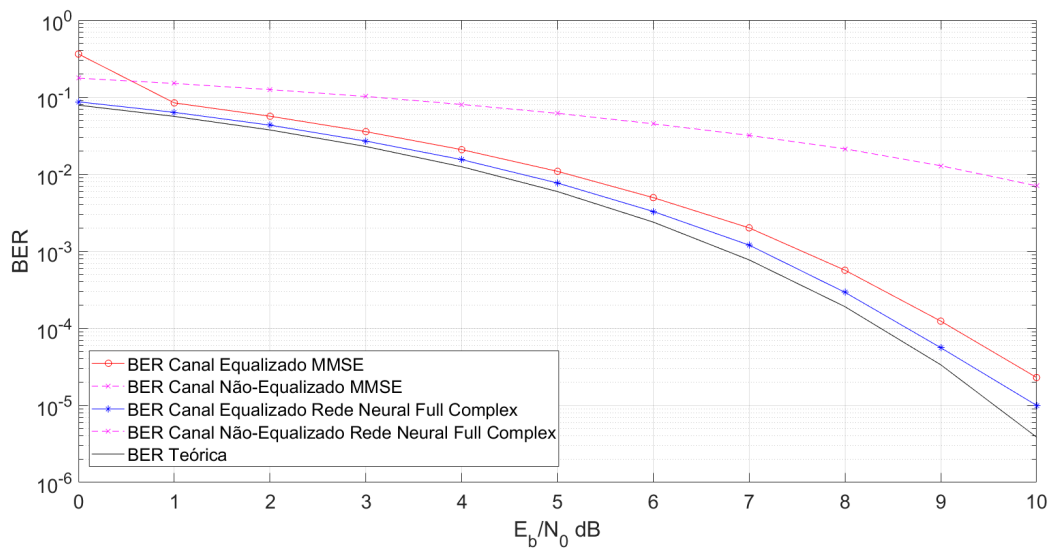


Figura 5.23: Curva BER da modulação BPSK para o Canal 4 - Comparativo *Fully Complex* e MMSE.

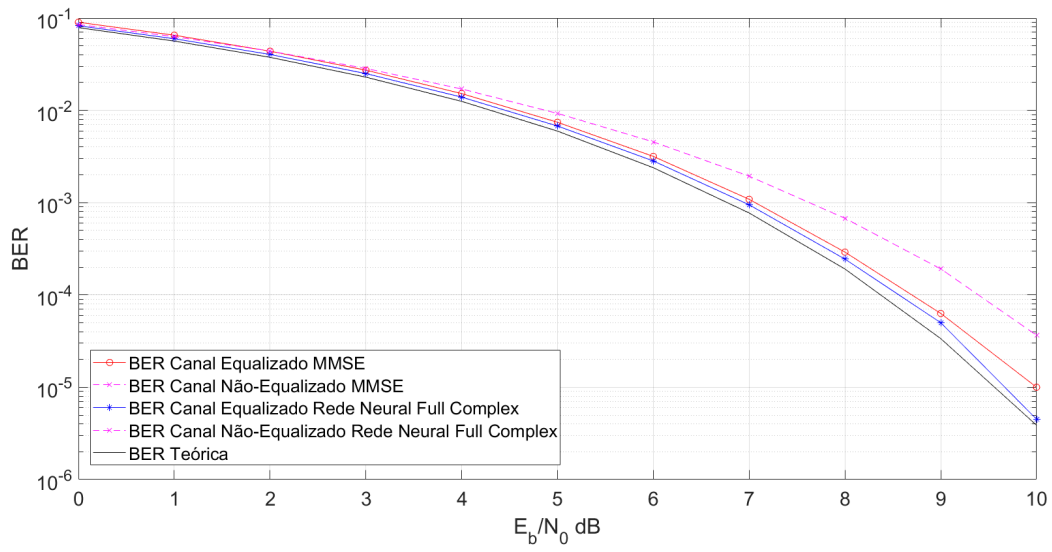


Figura 5.24: Curva BER da modulação QPSK para o Canal 1 - Comparativo *Fully Complex* e MMSE.

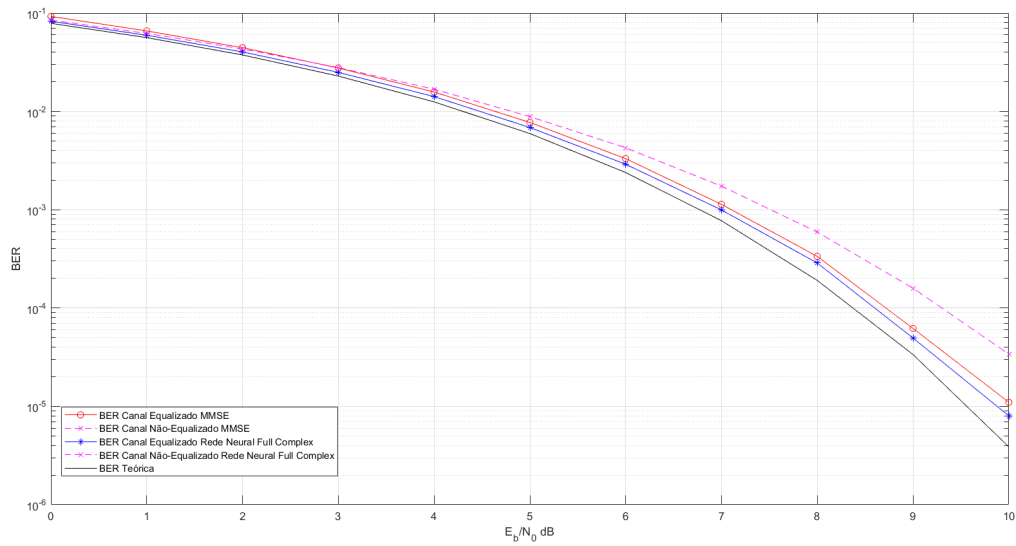


Figura 5.25: Curva BER da modulação QPSK para o Canal 2 - Comparativo *Fully Complex* e MMSE.

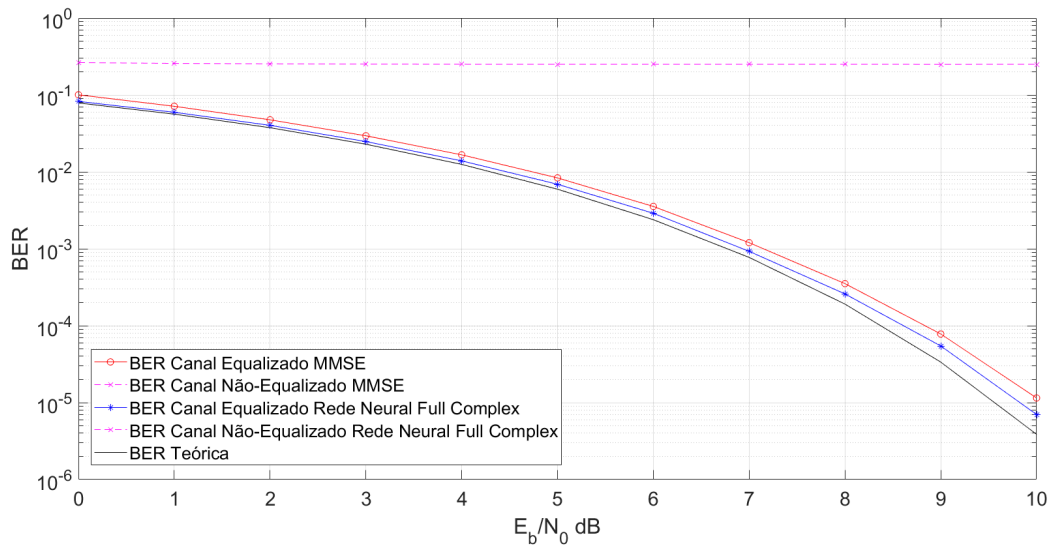


Figura 5.26: Curva BER da modulação QPSK para o Canal 3 - Comparativo *Fully Complex* e MMSE.

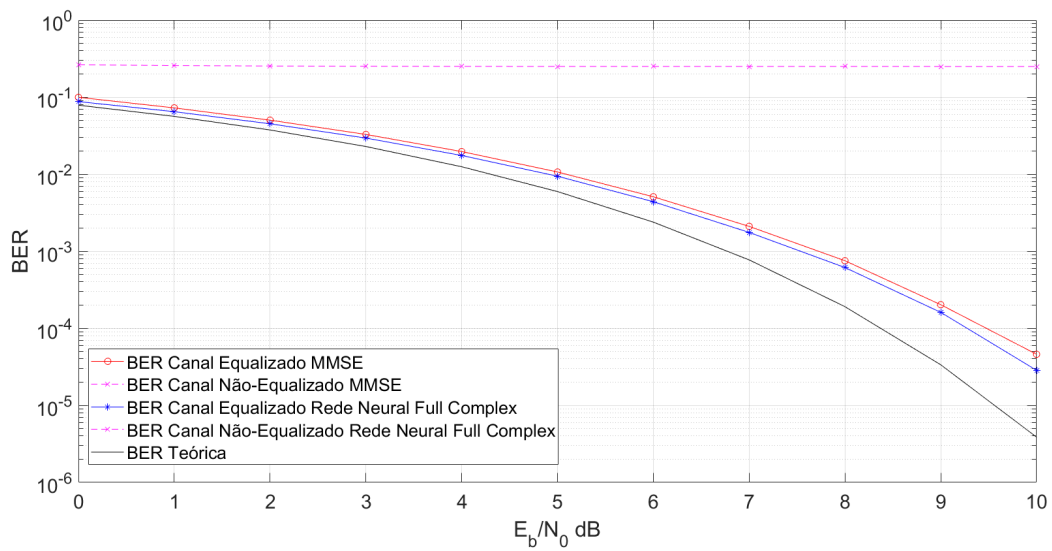


Figura 5.27: Curva BER da modulação QPSK para o Canal 4 - Comparativo *Fully Complex* e MMSE.

5.5.2 Análise comparativa entre os equalizadores *Fully Complex*, *Split* e MMSE no processo de equalização

Após os testes iniciais da aplicabilidade do modelo *Fully Complex*, simulou-se em conjunto o modelo *Split*, para os canais 3 e 4, que apresentam maior complexidade.

O intuito era verificar uma possível superioridade de desempenho da rede com o tratamento específico e das componentes em fase e em quadratura. Desta vez, utilizou-se o algoritmo de propagação resiliente, proposto em [16], por sua rapidez iterativa.

Para o equalizador *Fully Complex*, aumentou-se a complexidade das redes implementadas de forma diretamente proporcional ao esquema de modulação: para os esquemas de modulação BPSK e QPSK, utilizou-se um MLP com apenas uma camada oculta e cinco neurônios; para os esquemas de modulação 8-PSK, 16-PSK e 32-PSK, duas camadas ocultas foram necessárias, com, respectivamente, 8 e 10 neurônios cada; por fim, para o esquema de modulação 64-PSK, três camadas ocultas foram implementadas, com, respectivamente, 6, 8 e 10 neurônios.

O número de coeficientes do equalizador clássico MMSE utilizado para os esquemas de modulação foi definido aproximando-se o número total de neurônios das camadas ocultas implementadas para o equalizador *Fully Complex* do esquema de modulação em questão. Com isto, para as modulações BPSK/QPSK, o equalizador foi implementado com 8 coeficientes; para as modulações 8-PSK, 16-PSK e 32-PSK, o equalizador foi implementado com 18 coeficientes; por fim, para a modulação 64-PSK, o equalizador foi implementado com 24 coeficientes.

O equalizador *Split* não apresentou melhorias em seu desempenho com a mudança no número de neurônios implementados na rede, tanto para a diminuição quanto para o aumento de sua complexidade, considerando, respectivamente, 1 e 5 neurônios. Implementou-se uma camada oculta para a rede das componentes em fase e uma camada oculta para a rede das componentes em quadratura.

As Tabelas 5.3 a 5.17 apresentam os parâmetros das redes implementadas nas simulações desta seção e os resultados das simulações estão apresentados nas Figuras 5.28 a 5.39.

Verifica-se, nas simulações, que o equalizador *Fully Complex* é o que mais aproxima a curva de BER à curva teórica, apresentando o melhor desempenho sobre os efeitos de distorção dos canais 3 e 4. Ainda, verifica-se que o equalizador *Split* apresenta desempenho inferior não somente ao equalizador *Fully Complex*, mas também ao equalizador clássico MMSE.

A explicação para o desempenho do equalizador *Split* está na distorção do sinal: como visto, a distorção é expressa pela convolução do sinal transmitido com a função de transferência do canal. Na seção 5.5, os diagramas de constelação de cada um dos canais utilizados para cada esquema de modulação foram apresentados e comentou-se sobre os efeitos da convolução na composição final destes diagramas. O equalizador *Split* não leva em consideração as relações existentes entre as componentes em fase e em quadratura, considerando a distorção do sinal. Por este motivo, o tratamento isolado das componentes perde informações relevantes para o processo de equalização, comprometendo seu desempenho de forma severa. Além disto, viu-se que o equalizador MMSE perde sua funcionalidade para equalização em esquemas de modulação superiores ao 8-PSK.

Tabela 5.3: Tabela de Parâmetros BPSK/QPSK - Rede Neural *Fully Complex*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 2000 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.4: Tabela de Parâmetros BPSK/QPSK - Redes Neurais *Split*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [1] |
| Número de Épocas | 500 |
| Taxa de Aprendizagem | 0,05 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.5: Tabela de Parâmetros BPSK/QPSK - MMSE

| Parâmetro | Valor/Nome |
|------------------------|-------------------|
| Número de Coeficientes | 8 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

Tabela 5.6: Tabela de Parâmetros 8-PSK - Rede Neural *Fully Complex*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 2 |
| Número de Neurônios nas Camadas Ocultas | [8 10] |
| Número de Épocas | 2000 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.7: Tabela de Parâmetros 8-PSK - Redes Neurais *Split*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 500 |
| Taxa de Aprendizagem | 0,05 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.8: Tabela de Parâmetros 8-PSK - MMSE

| Parâmetro | Valor/Nome |
|------------------------|-------------------|
| Número de Coeficientes | 18 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

Tabela 5.9: Tabela de Parâmetros 16-PSK - Rede Neural *Fully Complex*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 2 |
| Número de Neurônios nas Camadas Ocultas | [8 10] |
| Número de Épocas | 1500 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.10: Tabela de Parâmetros 16-PSK - Redes Neurais *Split*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 500 |
| Taxa de Aprendizagem | 0,05 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.11: Tabela de Parâmetros 16-PSK - MMSE

| Parâmetro | Valor/Nome |
|------------------------|-------------------|
| Número de Coeficientes | 18 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

Tabela 5.12: Tabela de Parâmetros 32-PSK - Rede Neural *Fully Complex*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 2 |
| Número de Neurônios nas Camadas Ocultas | [8 10] |
| Número de Épocas | 1000 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.13: Tabela de Parâmetros 32-PSK - Redes Neurais *Split*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 500 |
| Taxa de Aprendizagem | 0,05 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.14: Tabela de Parâmetros 32-PSK - MMSE

| Parâmetro | Valor/Nome |
|------------------------|-------------------|
| Número de Coeficientes | 18 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

Tabela 5.15: Tabela de Parâmetros 64-PSK - Rede Neural *Fully Complex*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 3 |
| Número de Neurônios nas Camadas Ocultas | [6 8 10] |
| Número de Épocas | 1000 |
| Taxa de Aprendizagem | 0,07 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.16: Tabela de Parâmetros 64-PSK - Redes Neurais *Split*: Propagação Resiliente

| Parâmetro | Valor/Nome |
|---|-------------------|
| Número de Camadas Ocultas | 1 |
| Número de Neurônios nas Camadas Ocultas | [5] |
| Número de Épocas | 500 |
| Taxa de Aprendizagem | 0,05 |
| Função de Transferência | <i>tansig</i> |
| Erro | 0.0001 |
| Algoritmo de Treinamento | <i>trainrp</i> |
| Número de Amostras | 1 000 000 |
| Tamanho da Amostra de Teste | 150 000 |
| Tamanho da Amostra de Validação | 150 000 |
| Tamanho da Amostra de Treinamento | 700 000 |

Tabela 5.17: Tabela de Parâmetros 64-PSK - MMSE

| Parâmetro | Valor/Nome |
|------------------------|-------------------|
| Número de Coeficientes | 24 |
| Taxa de Aprendizado | 0,01 |
| Amostras para Treino | 333 333 |

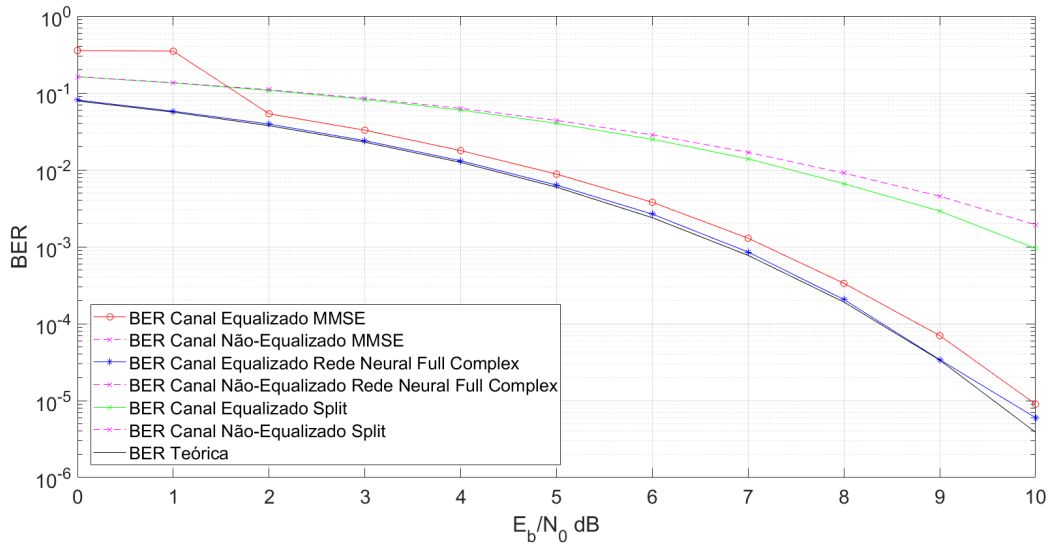


Figura 5.28: Curva BER da modulação BPSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

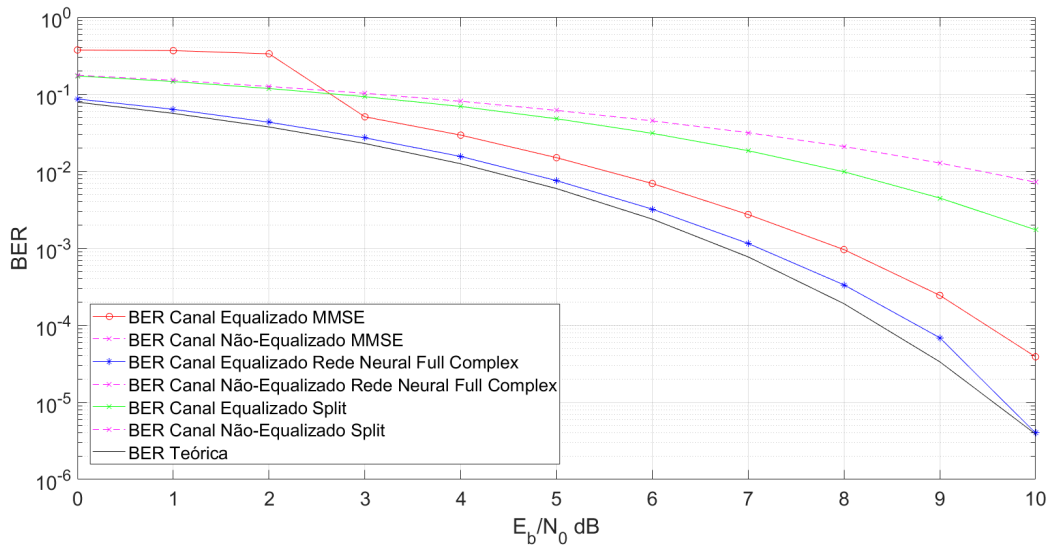


Figura 5.29: Curva BER da modulação BPSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

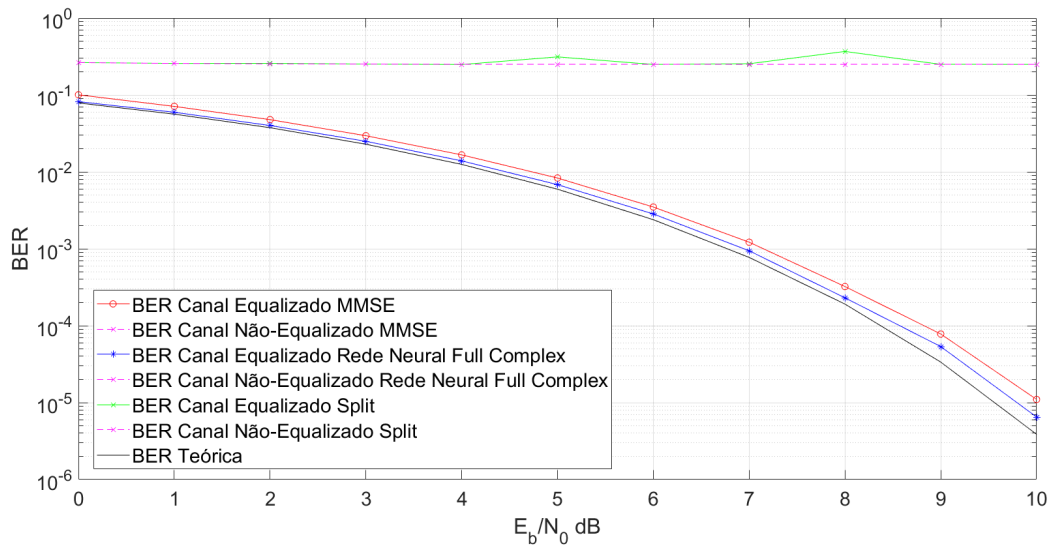


Figura 5.30: Curva BER da modulação QPSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

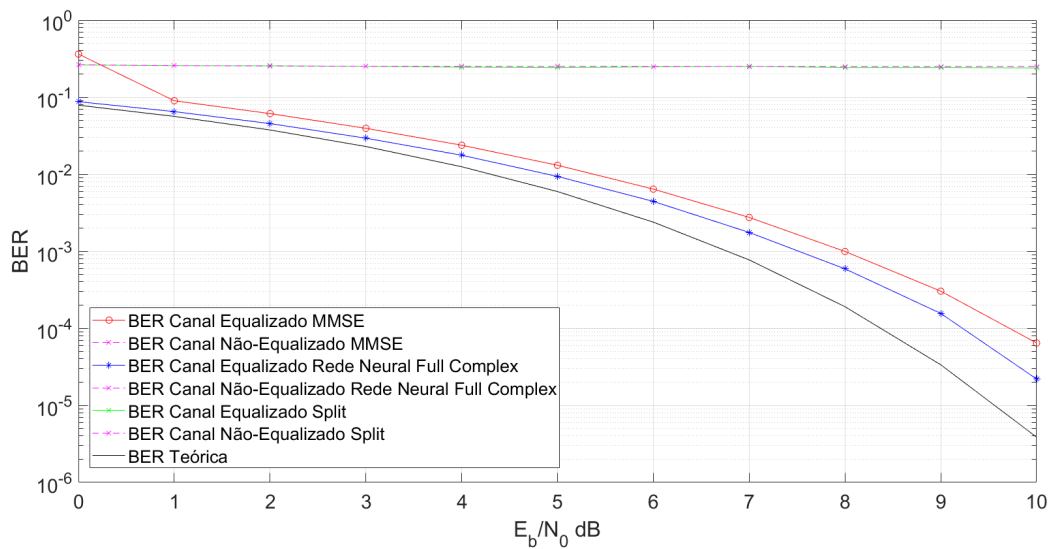


Figura 5.31: Curva BER da modulação QPSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

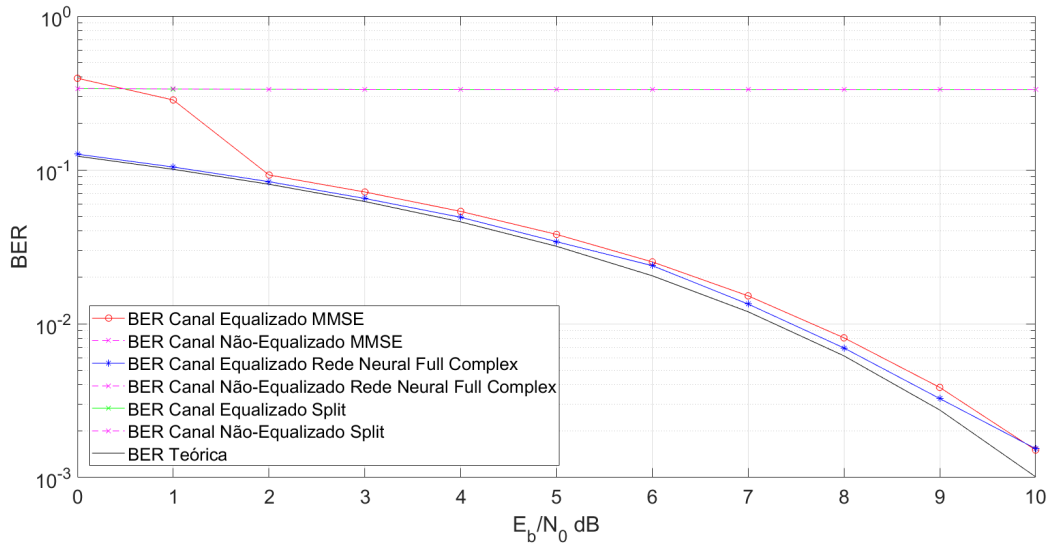


Figura 5.32: Curva BER da modulação 8-PSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

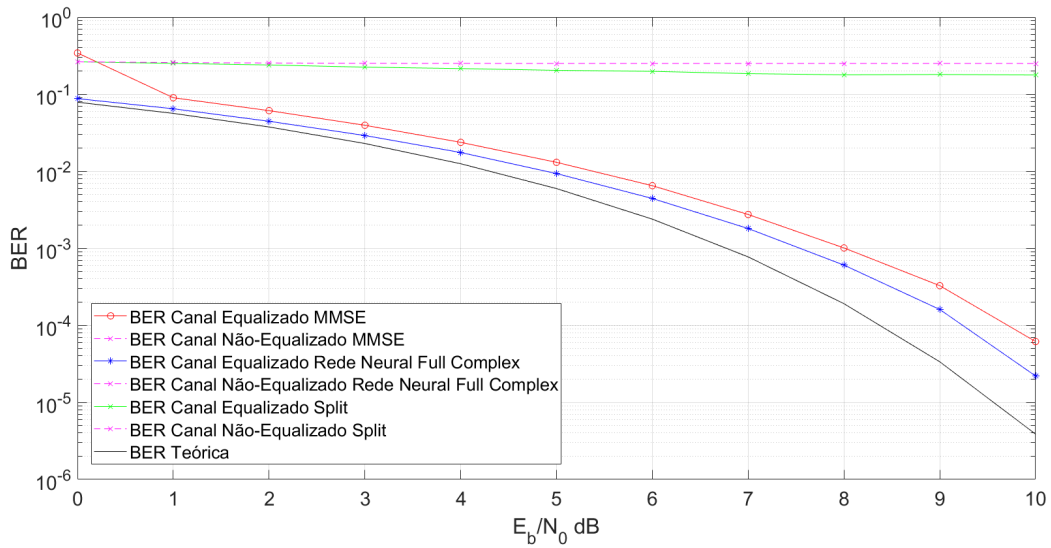


Figura 5.33: Curva BER da modulação 8-PSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

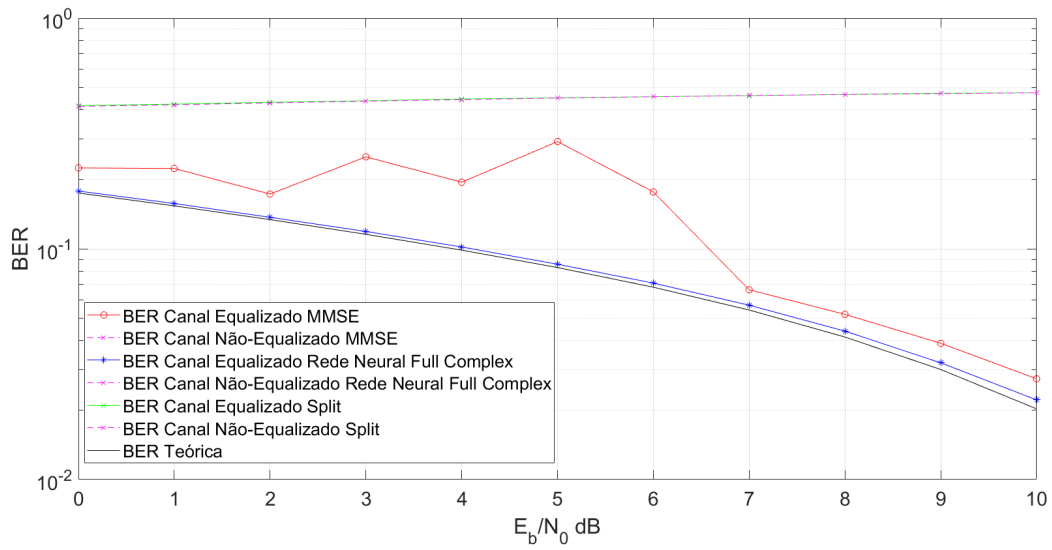


Figura 5.34: Curva BER da modulação 16-PSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

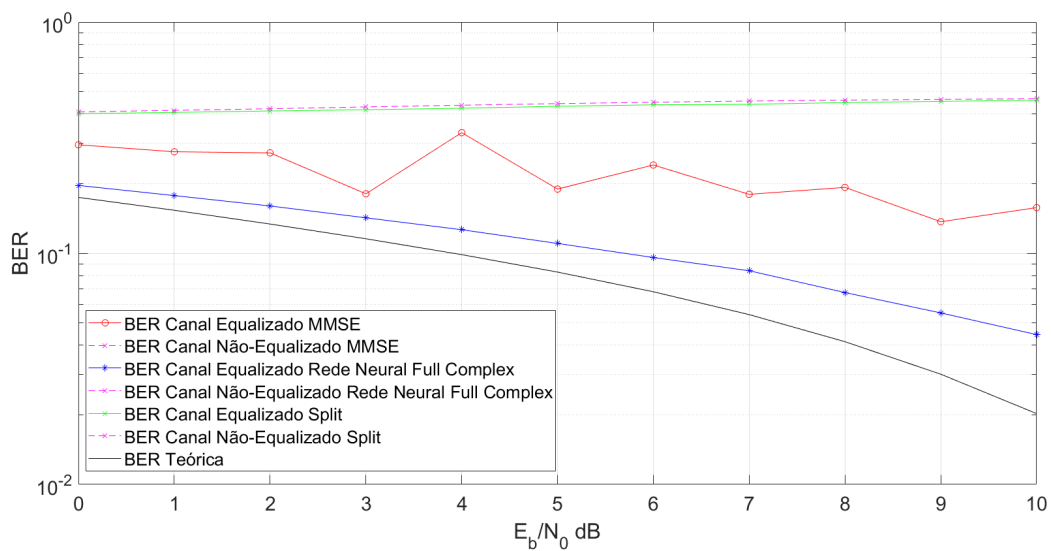


Figura 5.35: Curva BER da modulação 16-PSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

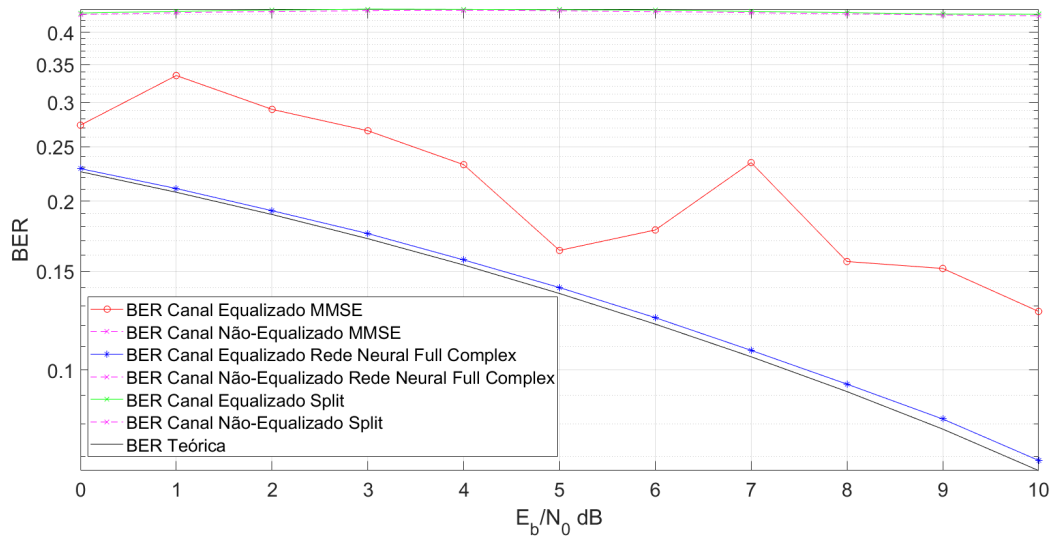


Figura 5.36: Curva BER da modulação 32-PSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

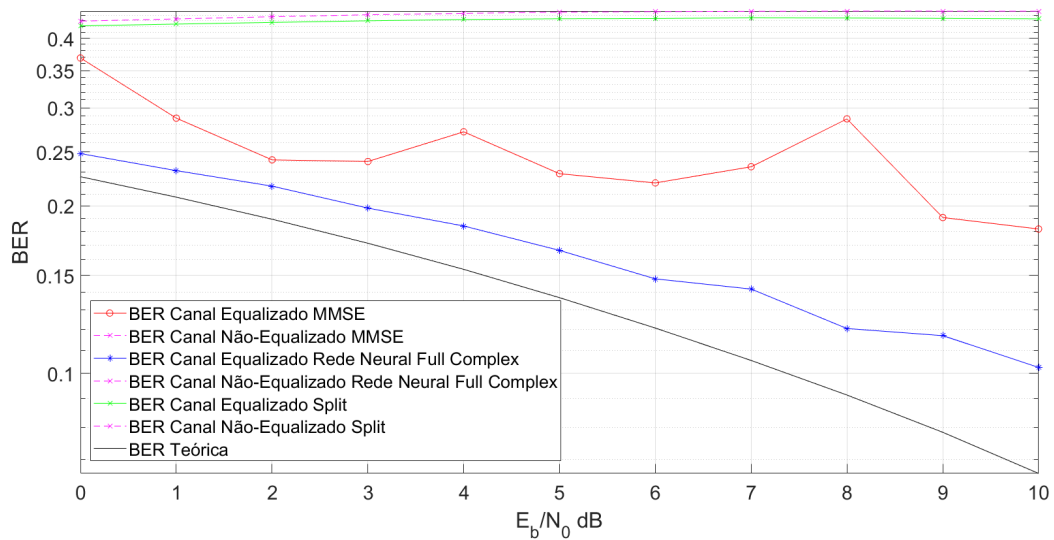


Figura 5.37: Curva BER da modulação 32-PSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

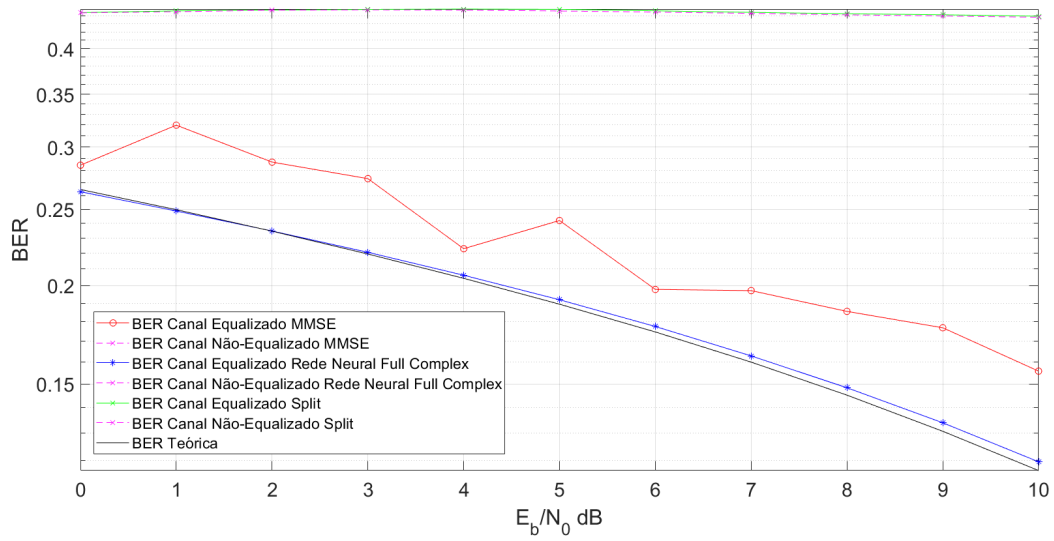


Figura 5.38: Curva BER da modulação 64-PSK para o Canal 3 - Comparativo *Split*, *Fully Complex* e MMSE.

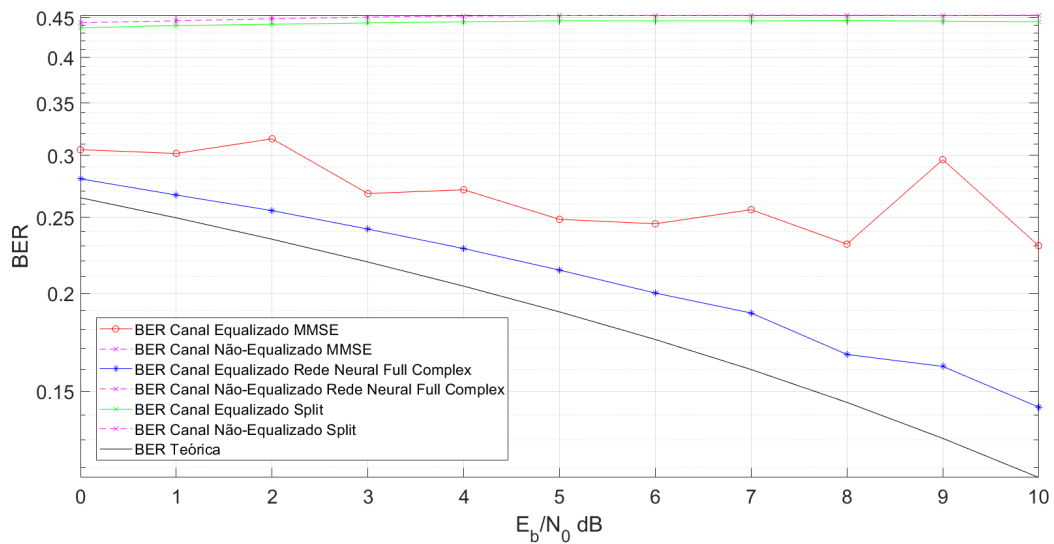


Figura 5.39: Curva BER da modulação 64-PSK para o Canal 4 - Comparativo *Split*, *Fully Complex* e MMSE.

5.6 Conclusões

Neste capítulo, foram apresentadas duas possíveis modelagens de redes neurais próprias para aplicação no processo de equalização de canal: *Fully Complex* e *Split*, apresentando o desenvolvimento do algoritmo destas modelagens para a atualização dos pesos da rede implementada.

Estas modelagens serviram de inspiração para os modelos utilizados para a obtenção dos resultados apresentados na seção 5.5, que apresenta a aplicabilidade das redes neurais no problema da equalização para diferentes modulações submetidas a canais com complexidades diferentes e verifica se o algoritmo *Split*, que separa o problema da equalização por redes neurais com o uso de duas redes neurais com estrutura mais simples, apresenta desempenho superior ao *Fully Complex* para dois tipos de canais.

No próximo capítulo, serão apresentadas as conclusões acerca dos resultados das simulações apresentadas na seção 5.5.

Capítulo 6

Conclusões

Neste trabalho, introduziu-se brevemente a história das redes neurais com o processo de equalização dos sistemas de comunicações digitais no Capítulo 1. Maiores detalhes sobre um sistema de comunicação digital e seus processos, além de um aprofundamento acerca das métricas de desempenho destes sistemas e de seu processo de equalização, foram apresentados nos Capítulos 2 e 3.

Em seguida, apresentou-se uma introdução às redes neurais, com seus diferentes tipos de aplicações, aprendizados e estruturas, no Capítulo 4. Por fim, a aplicação dos *perceptrons* de múltiplas camadas, um dos tipos de redes neurais, com adaptação para duas estruturas diferentes, *Fully Complex* e *Split*, teve seu desempenho analisado para o processo de equalização, considerando-se diferentes canais e esquemas de modulação, no Capítulo 5.

Os resultados das simulações apresentadas demonstraram que o uso de *perceptrons* de múltiplas camadas com modelagem inspirada no MLP *Fully Complex* mostra-se como alternativa viável ao processo de equalização de canais dispersivos.

Para todas as simulações apresentadas na seção 5.5, as redes implementadas apresentaram resultados de desempenho superior ao equalizador MMSE tradicional, considerando a questão da mitigação dos impactos da distorção provocada no sinal transmitido pelos diferentes canais dispersivos apresentado.

O tratamento isolado das componentes em fase e em quadratura dos sinais modulados proposto pelo equalizador inspirado no MLP *Split*, com a implementação de uma rede neural para cada uma das componentes, apresenta desempenho inferior tanto ao equalizador inspirado no MLP *Fully Complex*, quanto ao equalizador MMSE tradicional, para os canais 3 e 4 e todos os diferentes esquemas de modulação. Isto ocorre porque o tratamento isolado das componentes não leva em consideração a relação existente entre elas e a relação existente entre as componentes e os termos dos canais dispersivos apresentados, dadas pela convolução entre o sinal transmitido e o canal dispersivo. Ainda, o tratamento isolado das componentes não leva em consideração o comportamento do ruído sobre os símbolos.

Além dos resultados das redes, também foi possível comprovar a rapidez do algoritmo de

Tabela 6.1: Tabela Comparativa - Descida do Gradiente Adaptativo e Propagação Resiliente.

| Treinamento | Épocas |
|---|--------|
| Descida do Gradiente Adaptativo (<i>traingda</i>) | 2500 |
| Propagação Resiliente (<i>trainrp</i>) | 2000 |

propagação resiliente, quando comparado ao algoritmo do gradiente adaptativo, no quesito rapidez de treinamento das redes.

Este último resultado pode ser verificado na Tabela 6.1, que apresenta o número de épocas utilizadas para cada algoritmo de treinamento.

6.1 Propostas de continuidade do trabalho

Neste trabalho, as simulações apresentadas foram realizadas sobre um contexto restrito de aplicações, estimulando o aprofundamento das análises feitas e a realização de novos trabalhos sobre a aplicação de redes neurais no problema da equalização de canal.

Portanto, é possível considerar o desempenho de redes neurais dentro dos seguintes contextos:

- Análise comparativa do desempenho de outros tipos de redes neurais no problema da equalização de canal, como as redes RBF e SVM, com o desempenho do MLP *Fully Complex*;
- Análise do desempenho do MLP *Fully Complex* para esquemas de modulação superiores ao 64-PSK;
- Análise do desempenho do MLP *Fully Complex* para a variação da potências de cada símbolo do canal 4;
- Análise do desempenho do MLP *Fully Complex* para canais de complexidade superior ao canal 4 apresentado;
- Análise do desempenho do MLP *Fully Complex* utilizando diferentes funções de ativação, tais como as apresentadas em [14];
- Análise do desempenho do MLP *Fully Complex* e de outros tipos de redes neurais comparado ao desempenho de outros equalizadores tradicionais. Em especial, o equalizador tradicional MLSE, dadas as questões de custo computacional apresentadas por sua estrutura de equalização;
- Análise do desempenho do MLP *Fully Complex* e de outros tipos de redes neurais para canais variantes no tempo;
- Análise do desempenho do MLP *Fully Complex* e de outros tipos de redes neurais para canais móveis, que possuem influência mais acentuada da distorção por múltiplos percursos;
- Análise da rapidez de convergência das redes MLP *Fully Complex* quando comparadas aos equalizadores clássicos;

BIBLIOGRAFIA

- [1] B. P. Lathi and Z. Ding, *Sistemas de Comunicações Analógicas e Digitais Modernos*. LTC - Livros Técnicos e Científicos, 2012.
- [2] “Amostragem e Quantização.” <https://www.inatel.br/docentes/ynoguti/graduacao-sp-2113502489/50-amostragem-e-quantizacao>. Acessado em 19/11/2018.
- [3] Mendonça, G. V, “PCM, DPCM, DM, ADM.” <http://www02.smt.ufrj.br/gelson/ComDigitais/COMDIGIT3.pdf> - Acessado em 19/11/2018.
- [4] De Castro, F.C.C e De Castro, M.C.F, “Capítulo II - Codificação de Fonte.” <http://www.politecnica.pucrs.br/decastro/pdf/cd2.pdf> - Acessado em 19/11/2018.
- [5] Nascimento, E.J., “Demodulação e Detecção em Banda Básica - Tópicos Avançados em Engenharia Elétrica I.” http://www.univasf.edu.br/edmar.nascimento/topicos1/com_digital_aula04.pdf - Acessado em 19/11/2018.
- [6] K. Burse, R. N. Yadav, and S. C. Shrivastava, “Channel equalization using neural networks: A review,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, pp. 352–357, May 2010.
- [7] S. Chen, G. Gibson, C. Cowan, and P. Grant
- [8] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*. MITP, 1987.
- [9] N. Benvenuto and F. Piazza, “On the complex backpropagation algorithm,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 967–969, April 1992.
- [10] G. M. Georgiou and C. Koutsougeras, “Complex domain backpropagation,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, pp. 330–334, May 1992.
- [11] M. Meyer and G. Pfeiffer, “Multilayer *perceptron* based decision feedback equalisers for channels with intersymbol interference,” *IEE Proceedings I - Communications, Speech and Vision*, vol. 140, pp. 420–424, Dec 1993.

- [12] T. Kim and T. Adali, “Fully Complex multi-layer Perceptron network for nonlinear signal processing,” *VLSI Signal Processing*, vol. 32, pp. 29–43, 08 2002.
- [13] T. Kim and T. Adali, “Fully Complex backpropagation for constant envelope signal processing,” in *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501)*, vol. 1, pp. 231–240 vol.1, Dec 2000.
- [14] T. Kim and T. Adali, “Complex backpropagation neural network using elementary transcendental activation functions,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, pp. 1281–1284 vol.2, May 2001.
- [15] M. Ibnkahla and J. Yuan, “A neural network mlse receiver based on natural gradient descent: application to satellite communications,” in *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, vol. 1, pp. 33–36 vol.1, July 2003.
- [16] A. Kantsila, M. Lehtokangas, and J. Saarinen, “Complex rprop-algorithm for neural network equalization of gsm data bursts,” *Neurocomputing*, vol. 61, pp. 339 – 360, 2004. Hybrid Neurocomputing: Selected Papers from the 2nd International Conference on Hybrid Intelligent Systems.
- [17] “Introdução à Comunicação Digital.” <http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/238530.pdf>. Acessado em 19/11/2018.
- [18] B. Sklar and P.K.Ray, *Digital Communications - Fundamentals and Applications*. Pearson Education, 2014.
- [19] Madhow, U., *Fundamentals of Digital Communication*. Cambridge University Press, Nova Iorque, E.U.A, 2008.
- [20] Haykin, S., *Redes Neurais: Princípios e Práticas*. Bookman, Porto Alegre, RS, Brasil, 2008.
- [21] Flach, P., *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. Cambridge University Press, Nova Iorque, E.U.A, 2012.
- [22] Alpaydin, E., *Introduction to Machine Learning*. The MIT Press, Cambridge, Massachussets, E.U.A, 2014.