

RODRIGO DE PAULA RODRIGUES

**RECONHECIMENTO DE CARGAS ELÉTRICAS
MONOFÁSICAS NÃO-LINEARES ATRAVÉS DA
DECOMPOSIÇÃO WAVELET E DE
REDES NEURAIS ARTIFICIAIS**

Itajubá, Março de 2009

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RODRIGO DE PAULA RODRIGUES

RECONHECIMENTO DE CARGAS ELÉTRICAS
MONOFÁSICAS NÃO-LINEARES ATRAVÉS DA
DECOMPOSIÇÃO WAVELET E DE
REDES NEURAS ARTIFICIAIS

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Área de Concentração:
Automação e Sistemas Elétricos Industriais

Orientador:
Prof. Luiz Eduardo Borges da Silva

Março de 2009
Itajubá-MG

A minha família, cuja presença constante e imprescindível envolve cada novo dia,
cada nova conquista, cada nova vitória.

Agradecimentos

A meus pais e a minha irmã, por proporcionarem a possibilidade de me engajar em mais este desafio.

A minha namorada, que soube entender minhas ausências devido ao desenvolvimento deste trabalho e mesmo assim sempre me amparou com carinho e compreensão.

Aos amigos e a todo o corpo docente da Universidade Federal de Itajubá, em especial aos professores Vitorio Alberto De Lorenci e Otávio Augusto Salgado Carpinteiro, que mesmo sem compartilharem de participação formal neste trabalho, muito me instruíram e direcionaram tanto em termos acadêmicos quanto de conduta.

Aos professores Germano Lambert-Torres e Fernando Nunes Belchior, face aos direcionamentos e ajuda cedidos.

E a meu orientador, prof. Luiz Eduardo Borges da Silva, pelo empenho e pela compreensão desprendidos.

Folha em branco

Resumo

O contínuo desenvolvimento e o crescente emprego de novas tecnologias agregam aos sistemas de distribuição de energia elétrica equipamentos das mais variadas naturezas e finalidades. A presença crescente das chamadas cargas elétricas monofásicas não-lineares em meio a esses equipamentos tem como consequência marcante a injeção da chamada poluição harmônica, que por ser capaz de degradar consideravelmente a qualidade da energia disponibilizada pelas redes de distribuição, fez com que o objetivo maior deste trabalho fosse propor uma metodologia de reconhecimento da presença dessas cargas elétricas. Essa proposta é composta por um sistema de reconhecimento de padrões que utilize redes neurais artificiais como mecanismo de inferência e a decomposição *wavelet* em multirresolução como técnica de extração de características das formas de onda de corrente em estado estacionário dessas cargas elétricas. Assim, este trabalho discorre sobre as principais características a respeito de cargas elétricas não-lineares, sobre sistemas de reconhecimento de padrões, redes neurais artificiais e também a respeito das transformadas de Fourier e *wavelet*, que serviram de base para a metodologia proposta. Os detalhes de concepção e de implementação prática de cada uma das técnicas utilizadas são apresentados em companhia aos resultados da aplicação e da comparação da proposta deste trabalho com uma metodologia similar que utilize a transformada de Fourier para também extrair características das formas de onda de corrente geradas por cargas elétricas monofásicas não-lineares.

Palavras-chave: energia elétrica, sistemas de distribuição, qualidade de energia elétrica, poluição harmônica, cargas elétricas não-lineares, reconhecimento de padrões, redes neurais artificiais, transformada de Fourier, transformada *wavelet*.

Abstract

The continuous development of new technologies based on electronic devices is responsible for adding to the electric power distribution systems an increasingly amount of electronic equipment that are intended to sustain, in some way, everyday life. The growing presence of these devices, which are also called nonlinear electric loads, has as its main consequence the injection of the well-known harmonic pollution that, for being capable of degrade the quality of the electricity that is available within these systems, sets as the prime objective of this study the proposal of a methodology for the recognition of electric single-phase nonlinear loads by means of the usage of a pattern recognition systems based in a multi-layer artificial neural network as its inference mechanism and also by means of a multiresolution *wavelet* decomposition as its feature extraction mechanism for the steady-state current waveforms generated by these electric loads. Thus, this work raises the key features related to the nonlinear electric loads, to the pattern recognition systems, to the artificial neural networks and also to the Fourier and *wavelet* transforms. The details of design and practical implementation of each of these techniques used by this proposal are shown together with the results of its application and its comparison with a similar methodology based on the Fourier transform as the feature extraction technique for the nonlinear current waveforms.

Keywords: electricity, distribution systems, power quality, harmonic pollution, nonlinear electric loads, pattern recognition, artificial neural networks, Fourier transform, wavelet transform.

Sumário

Lista de Figuras	V
Lista de Tabelas	IX
Introdução	1
Capítulo 1: Cargas Elétricas Não-Lineares	9
1.1 Poluição Harmônica	17
1.2 Modelos de Representação	27
Capítulo 2: Sistemas de Reconhecimento de Padrões	34
2.1 Processos de Reconhecimento e Classificação	36
2.2 Aplicação	40
2.3 Operação e Treinamento	41
Capítulo 3: Redes Neurais Artificiais	46
3.1 Neurônios Artificiais	47
3.2 Perceptrons de Múltiplas Camadas	51
3.3 Perceptrons como Reconhedores de Padrões	58
3.4 Treinamento	59
3.4.1 Propagação de Erros	61
3.5 Arquitetura das Redes Neurais	65
Capítulo 4: Transformações de Fourier	67
4.1 A Transformada de Fourier de Tempo Contínuo	68
4.2 A Série de Fourier de Tempo Contínuo	72
4.3 A Transformada de Fourier de Tempo Discreto	73
4.4 A Série de Fourier de Tempo Discreto	74
4.5 Aplicação	75
4.5.1 A Transformada Rápida de Fourier	77
4.6 Limitações da Representação por Séries de Fourier de Tempo Discreto	82
4.6.1 Reconhecimento de Falsas Frequências	83
4.6.2 Vazamento de Frequências	84
4.6.3 Presença de Descontinuidades	87
Capítulo 5: Transformações Wavelet	91
5.1 A Transformada Wavelet Contínua	93
5.2 A Transformada Wavelet Discreta	98
5.3 Análise em Multirresolução	99
5.3.1 Espaços de Multirresolução	100

	IV
5.4 Aplicação	106
5.4.1 Momentos Nulos	106
5.4.2 Variação ao Deslocamento	107
5.4.3 Algoritmo de Banco de Filtros	108
5.4.4 Famílias de Funções Wavelets	111
Capítulo 6: Modelo Proposto	113
6.1 Amostragem	115
6.2 Decomposição	115
6.3 Extração de Características	116
6.4 Normalização de Características	117
6.5 Reconhecimento e Treinamento	118
Capítulo 7: Implementação do Modelo	119
7.1 Sistema de Aquisição de Dados	120
7.2 Rede Neural Artificial de Múltiplas Camadas	123
7.2.1 Testes	124
7.3 Transformada Rápida de Fourier	127
7.3.1 Testes	127
7.4 Transformada Wavelet Discreta	128
7.4.1 Testes	131
Capítulo 8: Aplicação do Modelo	134
8.1 Levantamento de Formas de Onda de Corrente	135
8.2 Análise das Formas de Onda	137
8.3 Reconhecimento de Cargas Individuais	139
8.4 Reconhecimento de Associações de Cargas Elétricas	142
8.5 Comparação com um Sistema Baseado na Transformada de Fourier	148
Capítulo 9: Conclusão	155
9.1 Propostas de Trabalhos Futuros	158
Apêndice A	166
Apêndice B	175
Apêndice C	194
Apêndice D	198
Apêndice E	202

Lista de Figuras

Figura 1 - Produção mundial de eletricidade entre os anos de 1980 e 2005 e sua participação percentual frente à energia primária total produzida.	2
Figura 2 - Curvas pertencentes a uma carga linear (a), um resistor, e a uma carga não-linear (b), um diodo.....	13
Figura 3 - Exemplo de uma forma de onda monofásica ideal para uma tensão de 127 V eficaz a 60 Hz.....	15
Figura 4 - Exemplo de formas de ondas trifásicas ideais, cada qual com 60 Hz de frequência e 127 V eficaz.....	15
Figura 5 - Formas de onda de tensão e corrente obtidas em sistemas de distribuição de energia elétrica e relacionadas a cargas elétricas lineares e a cargas elétricas não-lineares.....	16
Figura 6 - Representação de formas de onda contendo alguns dos distúrbios possíveis para as formas de ondas presentes em um sistema de distribuição de energia elétrica.....	18
Figura 7 - Formas de onda de um sinal (a) e das respectivas primeira (b), terceira (c) e nona (d) harmônicas que o compõem.....	20
Figura 8 - Formas de onda de corrente pertencentes a algumas cargas elétricas não-lineares.	23
Figura 9 - Diagrama elétrico da seção de retificação presente em computadores pessoais típicos.....	29
Figura 10 - Modelo genérico para cargas elétricas proposto pelo IEEE.	30
Figura 11 - Modelo simplificado utilizado para se representar um aparelho de ar condicionado no modo de resfriamento.....	31
Figura 12 - Forma de onda obtida pela simulação do circuito que representa um aparelho de ar condicionado em modo de resfriamento.....	31
Figura 13 - Diagrama das etapas típicas presentes nos modelos de reconhecimento de cargas não-lineares baseados na análise de ondas de corrente.....	33
Figura 14 - Procedimentos presentes em um processo de reconhecimento de padrões.....	37
Figura 15 - Procedimentos presentes em um processo de classificação.....	38
Figura 16 - Sistema de reconhecimento de padrões em seu modo de operação.....	42

Figura 17 - Ilustração do processo de treinamento no ponto de vista dos conjuntos de vetores de características envolvidos.....	42
Figura 18 - Plano vetorial comportando uma distribuição de vetores (a) e uma tentativa de classificá-los (b).....	43
Figura 19 - Particionamento de um espaço vetorial através de linhas que separem regiões...	45
Figura 20 - Neurônio biológico (a) e seu respectivo modelo funcional (b) segundo McCulloch e Pitts.....	47
Figura 21 - Modelo neural denominado perceptron.....	48
Figura 22 - Exemplos de funções de ativação unipolares (a)(c) e bipolares (b)(d) mais comuns. Fonte: [64].....	50
Figura 23 - Exemplo de partição de um espaço vetorial bidimensional efetuada por um perceptron.....	51
Figura 24 - Camada de M perceptrons com N entradas e M saídas.....	52
Figura 25 - Divisão de um espaço vetorial bidimensional em quatro regiões devido à presença de 3 planos (linhas) não coincidentes e paralelos.....	53
Figura 26 - Distribuição de vetores em um espaço bidimensional (a) e a respectiva tentativa de separá-los (b) através de planos (linhas).....	54
Figura 27 - Estrutura neural contendo 2 elementos na camada de entrada e 3 perceptrons na camada de processamento.....	54
Figura 28 - Regiões resultantes do particionamento efetuado por uma camada de 3 perceptrons.....	56
Figura 29 - Arquitetura básica de uma rede de perceptrons de múltiplas camadas contendo uma camada de entrada e duas de processamento.....	57
Figura 30 - Espectros de amplitude (a) e fase (b) obtidos a partir da série de Fourier da função $x(t)$	73
Figura 31 - Sinal discreto com frequência fundamental de 1 Hz e seus respectivos espectros harmônicos de amplitude e fase.....	76
Figura 32 - Decomposição de um sinal de 8 elementos em 8 sinais de 1 elemento cada.....	80
Figura 33 - Estrutura em borboleta utilizada pela transformada rápida de Fourier.....	81
Figura 34 - Aplicação de uma transformada rápida de Fourier a um sinal contendo 8 elementos.....	81

Figura 35 - Relação entre as frequências aparente, de amostragem e real em um sinal amostrado.....	84
Figura 36 - Espectro de amplitude de uma onda contendo as frequências de 60 Hz e 180 Hz (a) e o resultado (b) do acréscimo de uma componente frequência inter-harmônica de 110 Hz à onda inicial.....	86
Figura 37 - Sinal (a) e suas aproximações (b)-(e) obtidas através da série de Fourier de tempo discreto.....	89
Figura 38 - Funções wavelets da família Daubechies 2 com diferentes deslocamentos e escalas.....	95
Figura 39 - Janelas de tempo-frequência utilizadas pela transformada wavelet.	97
Figura 40 - Relação entre os subespaços de maior e menor resoluções.....	102
Figura 41 - Análise em multirresolução efetuada através dos subespaços A_s e W_s	104
Figura 42 - Exemplo de uma representação em multirresolução de 3 níveis.	105
Figura 43 - Estrutura básica da decomposição realizada pela análise em multirresolução efetuada através de bancos de filtros.....	110
Figura 44 - Estrutura básica de reconstrução realizada pela representação em multirresolução efetuada através de bancos de filtros.....	111
Figura 45 - Representação do modelo proposto, que busca relacionar os coeficientes da decomposição wavelet de uma forma de onda de corrente à classe de carga elétrica não-linear correspondente.....	114
Figura 46 - Diagrama em blocos do sistema proposto.....	114
Figura 47 - Interface de operação do equipamento de aquisição de dados desenvolvido.....	121
Figura 48 - Diagrama em blocos do equipamento de aquisição de dados desenvolvido.....	122
Figura 49 - Protótipo do equipamento de aquisição de dados desenvolvido.....	122
Figura 50 - Interface desenvolvida para se manipular a rede neural implementada.....	124
Figura 51 - Rede neural de múltiplas camadas utilizada para reproduzir a operação da função matemática ou-exclusivo para duas variáveis.....	125
Figura 52 - Erros de treinamento obtidos em dois processos de treinamento específicos onde se utilizou o algoritmo de mínimo passo descendente.....	125
Figura 53 - Erros de treinamento obtidos em dois processos de treinamento específicos onde se utilizou o algoritmo de Levenberg-Marquardt.....	126

Figura 54 - Interface de operação da transformada rápida de Fourier implementada para o modelo.....	127
Figura 55 - Forma de onda contendo as frequências de 1, 2 e 3 Hz e seu espectro de amplitudes obtido através da implementação da transformada rápida de Fourier.....	128
Figura 56 - Ampliação de formas de onda através do preenchimento das extremidades com valores 0 (a) e com valores simétricos (b).....	130
Figura 57 - Interface de operação da decomposição em multirresolução implementada para o modelo.....	131
Figura 58 - Forma de onda utilizada para se realizar um dos testes de validação da transformada wavelet implementada.....	132
Figura 59 - Aproximação em 5 níveis da forma de onda presente na Figura 58 e obtida através da transformada discreta wavelet implementada.....	133
Figura 60 - Exemplos de formas de ondas de corrente dos aparelhos analisados.....	136
Figura 61 - Decomposições wavelet em multirresolução aplicadas a cada forma de onda de corrente coletada.....	137
Figura 62 - Formas de onda de corrente originadas por associações entre um osciloscópio analógico, um monitor de cristal líquido e uma lâmpada fluorescente compacta.....	143
Figura 63- Divisão de um espectro de 128 frequências em N grupos de mesmo tamanho e que não se sobrepõem.....	149
Figura 64 - Circuito de detecção de passagem por zero.....	166
Figura 65 - Circuito de transdução dos sinais de tensão e corrente.....	167
Figura 66 - Circuito de filtragem dos sinais de tensão e corrente.....	167
Figura 67 - Circuito de operação.....	168
Figura 68 - Circuito de alimentação.....	168
Figura 69 - Exemplo da variação do erro de treinamento obtido pelo algoritmo de mínimo passo descendente para uma rede com estrutura 3-3-2-1.....	179
Figura 70 - Exemplo de variação do erro de treinamento obtido pelo algoritmo de Levenberg-Marquadt ao treinar uma rede com estrutura 3-2-2-1.....	180
Figura 71 - Exemplo de variações do erro de treinamento para uma rede com estrutura 2-2-2-1.....	180
Figura 72 - Exemplo de variação do erro de treinamento obtido pelo algoritmo de Levenberg-Marquadt ao treinar uma rede com estrutura 3-4-1.....	180

Lista de Tabelas

Tabela 1 - Alguns dos equipamentos elétricos e eletrônicos típicos a cada	11
Tabela 2 - Principais classes de circuitos e dispositivos responsáveis pela geração de distorções harmônicas	22
Tabela 3 - Distorções harmônicas totais e fatores de crista de algumas formas de onda.....	24
Tabela 4 - Principais conseqüências da presença de poluição harmônica em sistemas de distribuição de energia elétrica.	25
Tabela 5 - Resultados da adição de ondas harmônicas a uma de senoidal de 1A.	26
Tabela 6 - Modelos mais comuns e suas respectivas áreas de aplicação.	28
Tabela 7 - Espectros de amplitude e fase de um aparelho de ar condicionado no modo de resfriamento.....	30
Tabela 8 - Taxonomia dos métodos de reconhecimento de padrões.....	36
Tabela 9 - Principais famílias de wavelets ortogonais e suas respectivas características.....	112
Tabela 10 - Principais especificações do equipamento de aquisição de dados.....	120
Tabela 11 - Função matemática “ou-exclusivo” para duas variáveis.....	124
Tabela 12 - Equipamentos cujas formas de onda de corrente foram levantadas.....	135
Tabela 13 - Vetores de características obtidos a partir da forma de onda de corrente de um osciloscópio analógico.....	138
Tabela 14 - Conjuntos de formas de onda de corrente obtidos para o teste inicial do modelo proposto.....	140
Tabela 15 - Codificação dos vetores-alvo para o teste de reconhecimento individual.....	140
Tabela 16 - Resultados dos testes de identificação individual de cargas elétricas monofásicas não-lineares através do modelo proposto.....	141
Tabela 17 - Vetores de características obtidos para as associações de cargas elétricas analisadas.....	144
Tabela 18 - Vetores de identificação de classe e os vetores de saída da rede para cada um dos vetores de características relacionados aos vetores de identificação apresentados.....	145

Tabela 19 - Vetores de características obtidos para as associações entre um monitor (M), uma lâmpada fluorescente compacta (L), um gerador de funções (G) e um osciloscópio analógico (O).....	146
Tabela 20 - Vetores de identificação de classe e os vetores de saída da rede para cada um dos 15 vetores de características relacionados às associações entre um monitor (M), uma lâmpada fluorescente compacta (L), um gerador de funções (G) e um osciloscópio (O).....	147
Tabela 21 - Resultados dos testes de reconhecimento baseados na extração de características através da transformada rápida de Fourier.....	150
Tabela 22 - Vetores de características obtidos tanto através da transformada de Fourier quanto da decomposição em multirresolução wavelet para uma lâmpada fluorescente (L), um televisor de 17 polegadas (T) e um aparelho de som (S).....	152
Tabela 23 - Coeficientes dos filtros de decomposição de algumas famílias de funções wavelets.....	198

Introdução

Pode-se dizer que a disponibilidade de energia em uma sociedade e a conseqüente manutenção da relação entre sua oferta e sua demanda são resultados de questões que basicamente dependem da presença de fontes de energia e da técnica com qual essas fontes são manipuladas. A questão da disponibilidade de fontes de energia diz respeito à existência ou não de recursos naturais em um dado limite territorial, como o são os potenciais hidroelétricos, eólico, dentre outros, e a menos dos esforços relacionados à procura por novas fontes energéticas, essa questão é tida como definida. Já a questão técnica diz respeito à capacidade com a qual se trabalhar a energia de forma eficaz durante cada uma das atividades presentes em seus ciclos de utilização e, por conseguinte, também diz respeito ao desenvolvimento e à aplicação de conhecimentos específicos relacionados ao emprego de cada uma das formas de energia que são atualmente utilizadas e que têm, na eletricidade, uma de suas maiores e mais bem-sucedidas representantes.

A eletricidade é uma fonte de energia de descoberta remota e cuja utilização tornou-se possível e vem sendo aprimorada ao decorrer dos anos mediante contribuições de muitos pesquisadores [1]-[4]. Essa fonte de energia é classificada como uma fonte secundária, uma vez que não pode ser obtida diretamente na natureza mas somente a partir de transformações de outras fontes como a cinética e a térmica, por exemplo. Na atualidade, a eletricidade é tida como uma das fontes de recursos energéticos de maior utilização em todo o mundo, sendo que sua produção para o ano de 2005 foi da ordem de 11.240 Terawatt-hora (Twh)¹ [5] para o

¹ Com um consumo residencial mensal médio no Brasil de 144 Quilowatt-hora (kWh) em 2005 [6], a quantia de 11.240 TWh é capaz de suprir, por um período de 65 anos, 200 cidades brasileiras com 500.000 habitantes cada.

cenário mundial e de 354.514 Gigawatt-hora (Gwh) para o cenário brasileiro [6]. O primeiro montante representa um crescimento da ordem de 115% sobre o total de 5.209 Twh produzido em 1980 [5] e, ao se apurar o percentual que tais montantes representam frente à energia primária mundial total produzida para esses mesmos anos de 1980 e 2005, respectivamente, é observado um significativo crescimento de participação, o qual foi de 15,6 % em 1980 e passou para a marca de 25,7% em 2005. Ambas essas variações são melhor elucidadas pela Figura 1 e justificam, sem muita dificuldade, o porquê da eletricidade, mesmo de descoberta longínqua e já secular, continuar a inspirar numerosos esforços e campos de pesquisa [7]-[9].

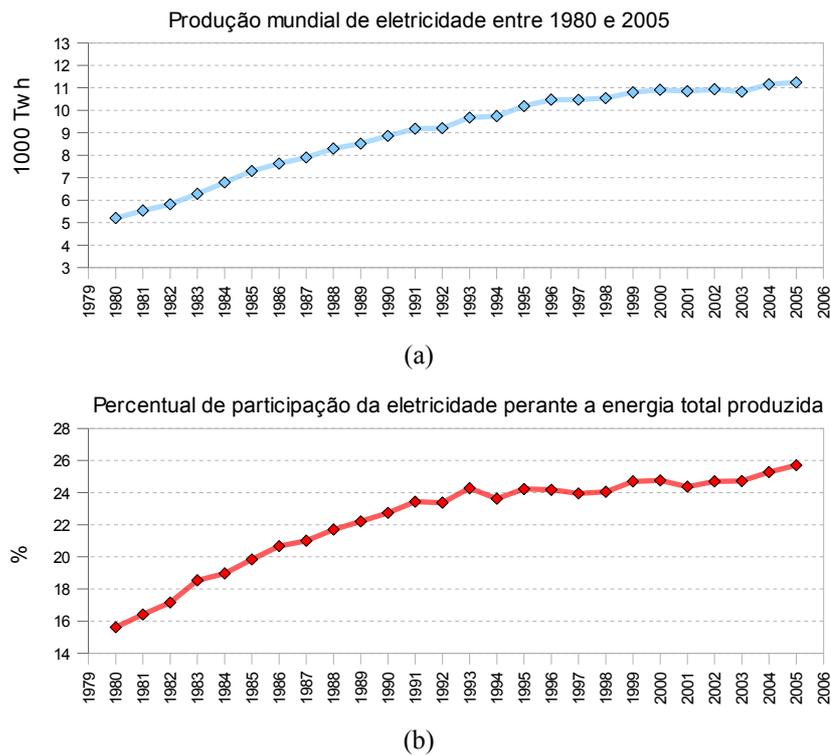


Figura 1 - Produção mundial de eletricidade entre os anos de 1980 e 2005 e sua participação percentual frente à energia primária total produzida.

Fonte: IEA [5]

Os campos de estudos relacionados à aplicação de energia elétrica são numerosos e abrangem atividades de pesquisa que visam a aprimorar cada um dos aspectos ligados à utilização desse tipo de energia. Entre esses campos está o da qualidade de energia elétrica [10], ou QEE, que trata das características de qualidade relacionadas aos serviços e aos produtos que são disponibilizados pelos diversos agentes do setor de energia elétrica a seus clientes. Essa área de investigação tem recebido crescente atenção, sendo que em meio a seus temas de interesse, os relacionados à continuidade do fornecimento e aos níveis da tensão elétrica são os mais difundidos. Esse interesse especial se deve principalmente ao fato de que a correta operação de uma grande e crescente variedade de processos e atividades contemporâneos dependa do emprego de componentes eletro-eletrônicos cada vez mais sensíveis a variações bruscas na tensão que os alimenta. Como consequência dessa sensibilidade, tanto atividades cotidianas e simples como o mero emprego de equipamentos eletrônicos e eletrodomésticos como computadores, rádios-despertadores, aparelhos de televisão, dentre outros, quanto atividades mais complexas e de maior impacto como a operação de instalações e aparelhos de auxílio à vida e a operação de complexas e amplamente automatizadas plantas industriais e de prestação de serviços podem ter suas operações comprometidas frente a problemas no potencial elétrico do qual dependem. Ambas essas questões tratadas pela QEE, a continuidade e a qualidade da tensão de fornecimento, são de tamanha importância que em grande parte dos países os agentes reguladores do setor elétrico já as normatizam mediante o estabelecimento de padrões a serem respeitados. No Brasil, que integra tal grupo de nações, a Agência Nacional de Energia Elétrica, ou ANEEL, regulamenta essas questões através de duas resoluções normativas que discorrem sobre a continuidade do fornecimento [11] e sobre limites para as variações no nível da tensão elétrica fornecida [12].

O interesse e a importância cedidos às questões da continuidade do fornecimento e da qualidade do nível da tensão elétrica direcionam o desenvolvimento de um campo de estudos da QEE que trata a questão da poluição nos sistemas de distribuição de potencial elétrico. Essa área de pesquisa é considerada como um aprofundamento natural dos estudos relacionados às questões da continuidade e da qualidade citados e suas atividades de investigação abrangem, em meio a temas como a mensuração de índices de poluição [13]-[15] e o levantamento das consequências devido à presença dessa mesma poluição [16][17], o estudo das cargas elétricas não-lineares, que são consideradas como um dos agentes poluidores dos próprios sistemas de distribuição aos quais são conectadas.

As atividades de pesquisa a respeito de cargas elétricas não-lineares têm como objetivo desenvolver técnicas e metodologias que sejam capazes de identificar a presença de tais agentes e caracterizar o tipo de poluição que os mesmos agregam às redes de distribuição, sendo que, à data de elaboração deste trabalho, as atividades ligadas à identificação de cargas não-lineares podiam ser situadas em duas possíveis categorias: a de métodos invasivos (intrusivos) e a de métodos não-invasivos (não-intrusivos).

Os métodos de caracterização de cargas não-lineares chamados invasivos são baseados na injeção de um sinal de prova e na análise do sinal de resposta captado, para então se obter as características desejadas de uma carga elétrica ou de um conjunto delas [18][19]. Os métodos não-invasivos, por sua vez, aferem grandezas correlatas à presença dessas cargas, como tensões, correntes e potências, as quais são utilizadas junto a modelos físicos e métodos computacionais voltados a levantar características dessas cargas como, por exemplo, o perfil das potências elétricas de cargas industriais presentes em uma dada localidade.

Os estudos relacionados a métodos não-invasivos de monitoramento e reconhecimento de cargas elétricas não-lineares são mais presentes na literatura e seus esforços parecem suplantam aqueles relacionados aos métodos invasivos de mesmo fim. Dentro dessa classe de estudos, estão presentes abordagens que se dividem entre a vertente que toma como parâmetro de classificação a relação entre a tensão aplicada sobre uma dada carga e a respectiva corrente produzida [20]; a vertente que faz a análise das potências elétricas relacionadas [21]-[24]; a que parametriza as classes de cargas elétricas segundo as curvas de carga (de potências) temporais geradas pelos clientes dos sistemas de distribuição [25]-[27] e, como uma das mais encontradas na literatura, a vertente que busca identificar cargas elétricas não-lineares através da análise das formas de onda da corrente que são produzidas por esses agentes poluidores [28]-[34]. Algumas outras vertentes como a que se baseia na admitância de cargas elétricas como parâmetro de identificação [35] e mesmo esforços correlatos também são presentes [36]-[41], mas, até o momento, a publicação de seus resultados traz avanços menos expressivos que aqueles apresentados pela vertente que tem base na análise das formas de onda de corrente.

O reconhecimento de cargas elétricas não-lineares monofásicas através da análise das formas de onda de corrente tem como base as premissas de que tais cargas produzem formas de onda de corrente com características que representem as classes de cargas das quais fazem parte e o fato de que a representação de uma dada classe seja única perante as demais. Essa mesma característica também é tida como base a um modelo para cargas elétricas não-lineares desenvolvido pelo *Institute Of Electrical And Electronics Engineers* [42], o IEEE, segundo o qual um agente poluidor pode ser representado através dos espectros harmônicos de amplitude e fase da forma de onda de sua corrente e, assim, sua utilização em circuitos elétricos destinados a estudos correlatos, por exemplo, pode ser obtida através da utilização da

associação em paralelo de ramos resistor-capacitor (RC) ou resistor-indutor (RL) cada qual representando uma dentre as componentes harmônicas dos espectros que representem a carga. Ainda segundo tal modelo, o estudo de associações de cargas não-lineares pode, então, ser feito mediante a associação em paralelo entre as representações de cada uma das cargas a compor a associação a ser estudada.

Esse modelo de representação de cargas não-lineares através dos conteúdos harmônicos de suas formas de onda de corrente naturalmente sugere o desenvolvimento de técnicas de identificação baseados na decomposição de tais formas de onda por meio da transformada discreta de Fourier [43][44], para, então, se fazer uso dos resultados da decomposição junto a algoritmos de reconhecimentos de padrões e afins [28]-[34]. Contudo, a aplicação da transformada discreta de Fourier a sinais que apresentam descontinuidades, as quais são comuns às formas de onda de corrente de cargas elétricas não-lineares [45], contém limitações [46][47] que dificultam a obtenção de uma representação em tempo-discreto desses sinais por parte de suas decomposições sem que haja um certo erro na representação, o que faz com que seja esperado que isso degrade a eficiência dos métodos de reconhecimento baseados na abordagem citada.

No entanto, a necessidade de se trabalhar com formas de onda que contenham descontinuidades e que sejam até mesmo não-estacionárias, uma constante em áreas de processamento de sinais, propiciou o desenvolvimento de técnicas como a transformada *wavelet* [48]. Essa classe de decomposições possui vantagens sobre a transformada de Fourier como o fato de conter suporte local [46], ao contrário do suporte global representado pelas funções seno e co-seno utilizados como base da transformada de Fourier, e a capacidade de melhor compactar as características de um sinal [47], fatos estes que a tornam mais apropriada

para se trabalhar junto a descontinuidades e às classes de sinais que as apresentem e, assim, junto a sinais de corrente de cargas elétricas não-lineares.

Nesse contexto, este trabalho tem como objetivo propor um modelo de reconhecimento de cargas elétricas monofásicas não-lineares que seja composto por um sistema de reconhecimento de padrões que faça uso de redes neurais artificiais como mecanismo de inferência e da transformada *wavelet* como técnica de extração de características presentes em formas de onda de corrente em estado estacionário, tendo-se como base a referência [49], onde se discorre sobre a aplicação da transformada *wavelet* a curvas de corrente de cargas elétricas não-lineares e se sugere, devido às características dessa transformada, a sua utilização como mecanismo de extração de características para sistemas de reconhecimento dessas cargas. Dessa forma, este trabalho propõe (i) um sistema de reconhecimento de cargas elétricas monofásicas e não-lineares e pretende testá-lo tanto para se (ii) reconhecer cargas elétricas presentes em formas de corrente geradas por uma única carga elétrica quanto para se (iii) reconhecer as classes de cargas elétricas presentes em uma forma de onda de corrente resultante das associações dessas cargas, na tentativa de se verificar se o modelo proposto é capaz de ser aplicado em situações mais próximas das reais, e, por fim, (iv) comparar algumas implementações da proposta com implementações similares que utilizem a transformada rápida de Fourier como mecanismo de extração de características.

A apresentação dos estudos e das atividades que compõem este trabalho, assim como cada um dos resultados obtidos, faz uso de uma organização que o estrutura segundo quatro áreas principais, sendo que a primeira delas tem o objetivo de apresentar os estudos executados e discorre, ao longo de todo o capítulo 1, sobre cargas elétricas não-lineares, apresentando sua definição e suas características mais relevantes. Os capítulos 2, 3, 4 e 5

também compõem esta primeira área e apresentam os principais aspectos de definição, construção e aplicação a respeito das respectivas temáticas sobre sistemas de reconhecimento de padrões, sobre redes neurais artificiais e sobre as transformações de Fourier e *wavelet*. A segunda área faz uso do capítulo 6 para discorrer a respeito do modelo proposto, levantando suas principais características e peculiaridades, enquanto a terceira área trata de aspectos de implementação e aplicação das técnicas utilizadas e do modelo proposto, englobando, para tanto, os respectivos capítulos 7 e 8. Por sua vez, a apresentação de resultados finais e de discussões correlatas é de responsabilidade da última área, a qual é desenvolvida ao longo do capítulo 9.

CAPÍTULO 1

Cargas Elétricas Não-Lineares

A busca pela melhora da qualidade de vida e pelo aprimoramento de processos de geração de valor econômico é um dos principais insumos ao desenvolvimento de tecnologias que são destinadas a amparar os afazeres humanos. Essa busca e seu conseqüente desenvolvimento disponibilizam ao mercado, quase que pertinentemente, uma grande variedade de novidades que ao serem absorvidas pelo cotidiano, permitem com que a execução de grande parte das atividades contemporâneas seja amparada por algum tipo de dispositivo destinado a mitigar esforços e dificuldades ou até mesmo a substituir a presença humana.

Frente a essa absorção paulatina de novas tecnologias, as sociedades passaram a cultivar uma forte dependência para com o emprego de equipamentos em quase todas as suas atividades, desde àquelas relacionadas ao lazer e ao bem-estar, até àquelas destinadas à prestação de serviços e à geração de mercadorias através de atividades industriais e afins. Tal

dependência, que parece se ampliar ainda mais ao longo dos tempos, traz consigo o importante fato de que, nos dias de hoje, as áreas atendidas pelos sistemas de distribuição de energia elétrica estejam povoadas por dispositivos das mais variadas naturezas e finalidades e que são, perante o ponto de vista desses mesmos sistema de distribuição, cargas elétricas.

Nesse sentido, as cargas elétricas são consideradas como todos e quaisquer equipamentos conectados aos sistemas de distribuição e cuja operação dependa da própria energia proveniente desses mesmos sistemas, indiferentemente de serem eles simples lâmpadas incandescentes, equipamentos mais complexos e sensíveis empregues em sistemas de automação e de auxílio à vida ou mesmo dispositivos mais robustos e de maior consumo como fornos, caldeiras, bombas e demais maquinários industriais e rurais.

A definição de carga elétrica segundo o ponto de vista dos sistemas de distribuição lhe concede um caráter bem amplo, que, na prática, sugere uma primeira classificação segundo os grupos residencial, comercial e industrial. O grupo de cargas residenciais engloba as cargas mais comuns em residências, enquanto o comercial representa os equipamentos comuns às instalações comerciais e o de cargas industriais remonta aos equipamentos de consumo mais elevado e corriqueiros às indústrias e afins. Essa classificação é costumeiramente utilizada junto a documentos de regulamentação do setor elétrico, como resoluções [11][12] e demais nos quais as principais considerações de classificação são a área de aplicação e o consumo de energia de cada um das cargas elétricas.

A Tabela 1 apresenta os dispositivos mais comuns para cada uma das classes de cargas residencial, comercial e industrial.

Tabela 1 - Alguns dos equipamentos elétricos e eletrônicos típicos a cada uma das classes de cargas elétricas residencial, comercial e industrial

Residenciais	Comerciais	Industriais
Chuveiros	Aparelhos de ar condicionado	Caldeiras
Computadores	Computadores	Fornos
Fornos	Fornos	Conversores
Lâmpadas	Iluminação	Iluminação
Lavadoras de roupas	Sistemas de refrigeração	Inversores
Lavadores de louças		Geradores
Microondas		Motores
Refrigeradores		Sistemas de automação
Televisores		Transformadores

A classificação de cargas elétricas segundo os grupos residencial, comercial e industrial, mesmo embora corriqueira às áreas de regulamentação de sistemas elétricos, não é capaz de fornecer e nem mesmo sugerir informações de maior importância aos estudos de pesquisa relacionados aos sistemas de distribuição de energia elétrica, como o são, por exemplo, a descrição do comportamento de uma carga elétrica no ponto de vista da corrente consumida e as contribuições que sua presença agrega aos sistemas de distribuição.

Para se estudar as cargas elétricas a partir de um ponto de vista mais técnico e que melhor descreva suas características de interesse aos sistemas de distribuição, os estudos relacionados adotam a descrição de um equipamento conectado a um sistema de distribuição, ou mesmo de um conjunto deles, do ponto de vista elétrico, com o qual são relacionadas as correntes que consomem e as tensões que os alimentam [13]. Essa relação é expressa pela equação (1.1), que descreve a corrente elétrica i consumida por uma carga elétrica como sendo uma função da tensão v aplicada sobre a mesma e a obedecer à lei matemática ρ , que é utilizada para se caracterizar a relação corrente-tensão inerente a cada classe de dispositivos.

$$i = \rho(v) \quad (1.1)$$

Essa abordagem de descrição de cargas elétricas fornece, entre outras informações, a possibilidade de se modelar o comportamento de tais cargas como se fossem funções de transferência entre a corrente consumida e a tensão aplicada, bem como a possibilidade de se agrupar as carga elétricas segundo a linearidade da corrente que consumem em relação à tensão a elas aplicada, o que se resume, em termos práticos, a classificar as cargas elétricas como cargas lineares ou cargas não-lineares.

Uma carga elétrica linear é tida como uma carga cuja relação expressa pela equação (1.1) toma a forma

$$i = kv, \quad (1.2)$$

na qual a função de transferência ρ é modelada como sendo uma combinação linear k de operadores também lineares como o integral, o derivativo, e constantes a expressar a corrente consumida pela carga como sendo uma proporção do potencial elétrico a ela aplicado. Essa relação claramente define uma dependência linear entre a corrente e a tensão de uma carga elétrica e é amplamente utilizada para se modelar a relação corrente-tensão de cargas elétricas puramente lineares como resistores, transformadores lineares, lâmpadas incandescentes, dentre outras.

Um exemplo da relação citada é apresentado na Figura 2 (a), que exhibe a relação corrente-tensão para um resistor de 5Ω e para um diodo. Este último, uma carga elétrica não-linear típica.

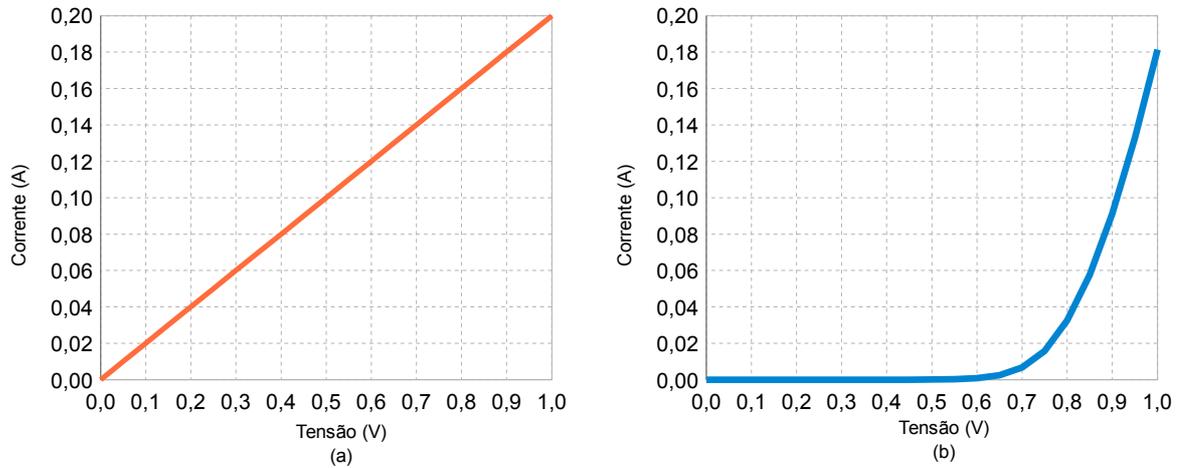


Figura 2 - Curvas pertencentes a uma carga linear (a), um resistor, e a uma carga não-linear (b), um diodo.

Ambas as curvas (a) e (b) presentes na Figura 2 foram obtidas através do programa de simulação *Spice* [50] e apresentam as relações corrente-tensão de um resistor de 5Ω , em (a), e de um diodo, em (b). A relação para o resistor é da forma $i = (1/2)*v$, o que está em concordância com a equação (1.1) e exprime uma representação gráfica similar às demais pertencentes a cargas elétricas lineares. Já a relação em (b), própria de um diodo da família 1N4148, exprime a relação corrente-tensão de uma das classes de componentes elétricos mais comuns à grande parte dos circuitos de potência presentes em dispositivos eletrônicos. Esse é um componente não-linear clássico que ajuda a conferir a própria característica não-linear apresentada por equipamentos eletro-eletrônicos.

Diferentemente das cargas lineares, as cargas não-lineares não possuem uma representação única para a equação (1.1) que seja capaz de expressar a relação corrente-tensão para todas as suas subclasses e, assim, cada uma delas está relacionada a um dado tipo de relação derivada de (1.1).

Demais relações corrente-tensão para cargas elétricas não-lineares podem ser obtidas na referência [20], sendo que para o caso do diodo utilizado como exemplo de carga não-linear, sua relação corrente-tensão é da forma [51]

$$i_d = k_1(e^{v/k_2} - 1), \quad (1.3)$$

em que k_1 e k_2 são constantes a serem adaptadas segundo as características do diodo a ser estudado.

A utilização da equação (1.1) para se classificar as cargas elétricas requer um esforço prévio no sentido de se levantar a equação analítica que represente a relação corrente-tensão resultante das características eletro-eletrônicas de cada classe de carga, para que, somente então, se possa submeter a classe de cargas em estudo à parcela de cargas lineares ou à parcela de cargas não-lineares. No entanto, ao se fazer menção aos equipamentos conectados e alimentados por sistemas de distribuição de energia elétrica em corrente alternada (CA), um procedimento prático e até mesmo intuitivo pode ser utilizado para se proceder com a identificação de tais dispositivos como sendo cargas elétricas lineares ou não-lineares. Essa possibilidade toma como base a operação ideal dos sistemas de distribuição nos quais as tensões disponibilizadas às cargas elétricas assumem formas de ondas senoidais de frequência e amplitude constantes que devem apresentar distorções mínimas [12] e, assim, se assemelharem às formas de ondas ideais apresentadas tanto pela Figura 3 quanto pela Figura 4. Dessa forma, ao se aplicar uma forma de onda de tensão em condições ideais a um equipamento, a forma de onda de sua corrente conterá amplitude e frequência modeladas por algumas de suas características e, assim, se essa forma de onda resultante contiver a mesma frequência e o mesmo formato senoidal da onda de tensão aplicada, mesmo perante a existência de um atraso angular entre ambas as ondas, então o equipamento será classificado

como uma a carga elétrica linear. Já para os casos nos quais a onda de corrente apresente distorções ou em sua forma ou em sua frequência, fala-se, então, de uma carga elétrica não-linear.

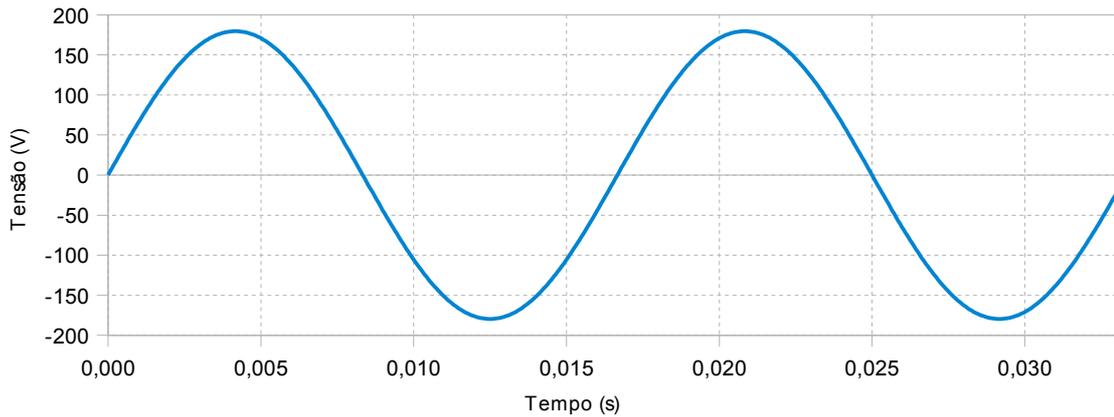


Figura 3 - Exemplo de uma forma de onda monofásica ideal para uma tensão de 127 V eficaz a 60 Hz.

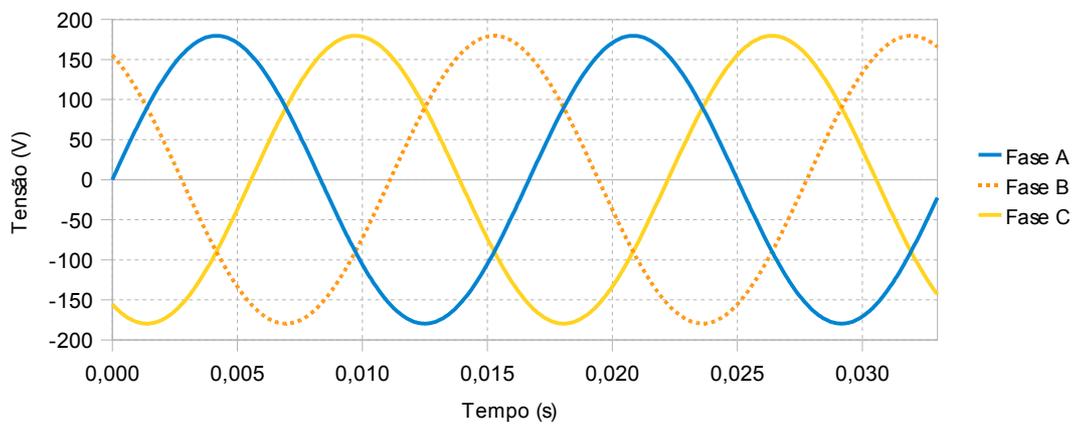


Figura 4 - Exemplo de formas de ondas trifásicas ideais, cada qual com 60 Hz de frequência e 127 V eficaz

Nesse sentido, as curvas temporais de tensão e corrente de cargas lineares ligadas aos sistemas de distribuição se assemelham às curvas (a.1) e (a.2) presentes na Figura 5, em que tanto o formato senoidal quanto a frequência presentes na forma de onda de tensão são

herdados pela forma de onda de corrente, mesmo perante a perceptível presença de um atraso de fase nesta última.

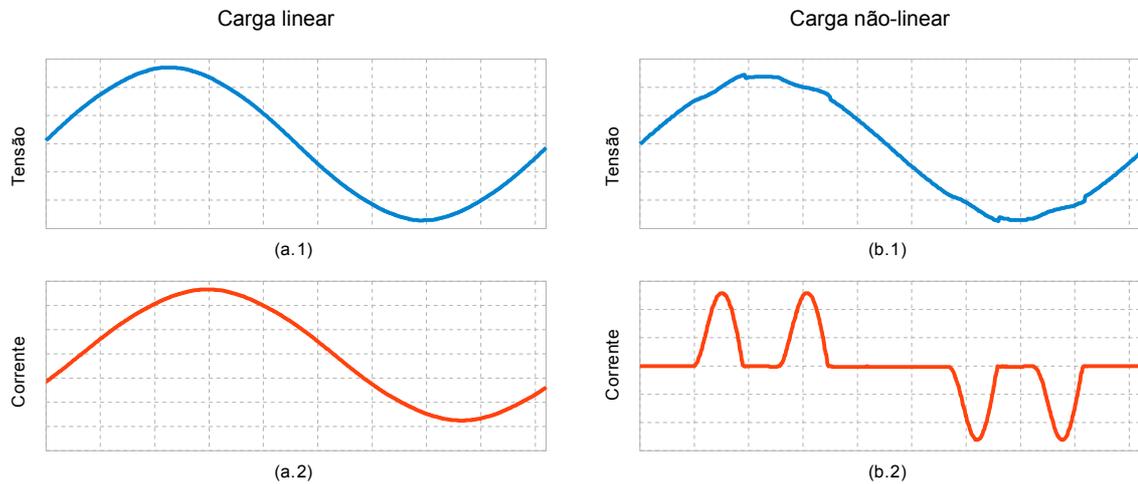


Figura 5 - Formas de onda de tensão e corrente obtidas em sistemas de distribuição de energia elétrica e relacionadas a cargas elétricas lineares e a cargas elétricas não-lineares.

Já quanto às curvas (b.1) e (b.2) também presentes na Figura 5, essas são referentes a uma das fases de uma carga trifásica não-linear típica, um retificador trifásico de onda completa com filtro capacitivo. Nesse caso, assim como o que ocorre com as demais cargas não-lineares, prevalece uma notória ausência de linearidade da forma de onda de corrente para com uma onda senoidal, assim como também pode ser notada uma leve degradação da forma de onda de tensão que alimenta a carga e que ocorre como resultado da presença da não-linearidade da própria carga elétrica exemplificada.

Ainda a respeito das formas de ondas presentes na Figura 5, as distorções presentes nas curvas (b.1) e (b.2) são exemplos de uma das principais questões que justificam a importância dada à classificação das cargas elétricas como lineares e não-lineares, pois tais distorções são características únicas das cargas não-lineares que, dentre outras coisas, injetam

a chamada poluição harmônica nos sistemas de distribuição e assim são consideradas como um dos principais agentes responsáveis por níveis de poluição capazes de degradar a qualidade do potencial elétrico consumido tanto por elas mesmas quanto pelas demais cargas que compartilhem dos mesmos ramais de distribuição.

1.1 Poluição Harmônica

As cargas elétricas não-lineares representam uma classe de cargas elétricas que notoriamente despertam maior interesse aos campos de estudos relacionados, se comparadas às cargas lineares. E isso se deve ao fato de que as cargas lineares não representam maiores problemas aos sistemas elétricos de potência do que daqueles já bem conhecidos e estudados problemas relacionados ao elevado consumo de potência elétrica e ao baixo fator de potência que podem ser associados a algumas dessas cargas elétricas. Mas já em termos das cargas elétricas não-lineares, no entanto, a importância cedida aos esforços ligados aos estudos de sua presença nos sistemas de distribuição se deve ao fato de que essa classe de cargas elétricas é responsável pela emissão da chamada poluição harmônica, o que ocorre como resultado da existência de distorções presentes em suas correntes e que por estarem impregnadas por conteúdo harmônico que se propaga pela própria rede de distribuição, são responsáveis por uma considerável parcela da degradação da qualidade do potencial elétrico distribuído [52].

O termo poluição harmônica é utilizado para designar um tipo de poluição sujeita a ocorrer em sistemas de distribuição devida à presença de cargas elétricas cujas formas de onda

de corrente apresentam um tipo de distúrbio elétrico que é caracterizado por distorções conhecidas como distorções harmônicas. Em meio a outros distúrbios elétricos também passíveis de ocorrerem nas redes distribuição, um exemplo típico de poluição harmônica é apresentada pela Figura 6.

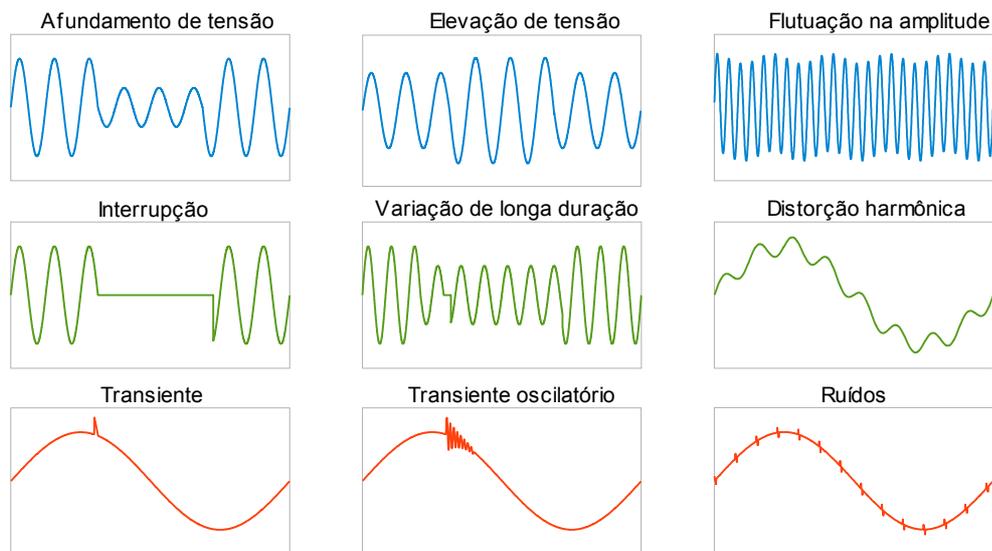


Figura 6 - Representação de formas de onda contendo alguns dos distúrbios possíveis para as formas de ondas presentes em um sistema de distribuição de energia elétrica.

Fonte: [52].

O conceito de distorção harmônica é tido como um tipo de distúrbio característico às formas de onda de corrente de cargas elétricas não-lineares e cujo significado está estritamente ligado ao conceito matemático de ondas harmônicas, que se trata de um conceito matemático que devido aos grandes sucesso e emprego junto a campos de estudos em eletricidade, obteve ares de conceito físico. Dessa forma, o conceito de distorção harmônica é derivado de mais uma das bem-sucedidas aplicações da transformada de Fourier e permite que, acima de tudo, as poluições advindas de cargas elétricas não-lineares sejam modeladas e estudadas através de um ferramental que tem base na série de Fourier, uma técnica muito bem conhecida e dominada pela literatura [44].

A série de Fourier se trata de uma derivação da transformada de Fourier e é apropriada para se trabalhar com sinais periódicos assim como o são as formas de onda de corrente e tensão das cargas elétrica não-lineares. Segundo essa técnica, um sinal $f(t)$ de período fundamental T e, portanto, com uma frequência angular fundamental $\omega_0=2\pi/T$, pode ser decomposto através da equação (1.1.1) em um somatório de ondas senoidais em que cada umas delas contenha uma frequência múltipla inteira da frequência angular fundamental ω_0 e uma amplitude que se relacionada ao sinal original através do conjunto de equações (1.1.2) [52].

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{+\infty} [a_n \cos(n\omega_0 t) + b_n \text{sen}(n\omega_0 t)] \quad (1.1.1)$$

$$a_0 = \frac{2}{T} \int_0^T f(t) dt \quad a_n = \frac{2}{T} \int_0^T f(t) \cos(n\omega_0 t) dt \quad b_n = \frac{2}{T} \int_0^T f(t) \text{sen}(n\omega_0 t) dt \quad (1.1.2)$$

Quando aplicada a um sinal periódico de tensão $v(t)$, a série de Fourier gera um conjunto de formas de ondas que são chamadas de ondas harmônicas. Essas ondas possuem, cada qual, uma frequência múltipla inteira da frequência do que é chamada de componente fundamental da tensão $v(t)$ e uma amplitude específica. A designação das harmônicas presentes em uma decomposição respeita o índice n presente na transformação (1.1.1), que estabelece tanto a ordem da harmônica quanto a multiplicidade de sua frequência com relação à frequência da componente fundamental do sinal decomposto. Como exemplo, caso a frequência fundamental de uma tensão não decomposta fosse, por exemplo, de 60 Hz, ao decompô-la, sua primeira harmônica ($n=1$) conteria também 60 Hz e seria designada por harmônica ou componente fundamental, enquanto a segunda harmônica ($n=2$) conteria 120

Hz, a terceira ($n=3$), 180 Hz, a quarta ($n=4$), 240 Hz, a quinta ($n=5$), 300 Hz, e assim sucessivamente.

Um exemplo de aplicação de decomposição que leva a se utilizar o conceito de distorção harmônica é apresentado nas formas de onda da Figura 7, na qual a curva (b) é tida como a primeira harmônica, ou fundamental, enquanto as ondas (c) e (d) representam a terceira e a nona harmônicas que, somadas à fundamental, permitem reconstruir a forma de onda original representada tanto pela curva (a) da mesma Figura 7 quanto pela formulação analítica na equação (1.1.3).

$$v(t) = 1.03 \sin(2\pi 60 t) + 0.11 \sin(3(2\pi 60 t)) + 0.23 \sin(9(2\pi 60 t)) \quad (1.1.3)$$

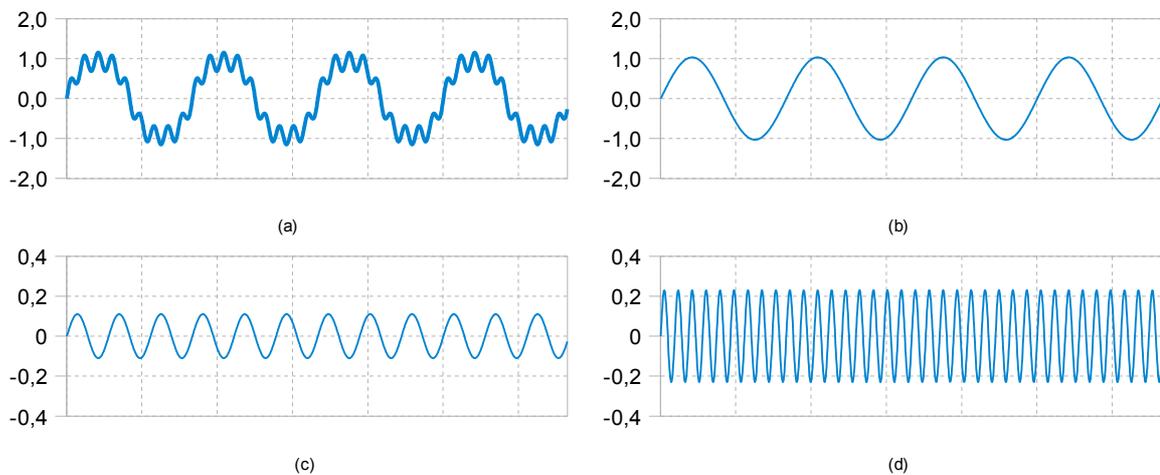


Figura 7 - Formas de onda de um sinal (a) e das respectivas primeira (b), terceira (c) e nona (d) harmônicas que o compõem.

Esse método de representação de sinais através das componentes harmônicas não é tido como um método universal no sentido de que possa ser aplicado à classe de cargas elétricas não-lineares em sua totalidade, uma vez que não é capaz de representar corretamente as formas de onda de cargas não-lineares como circuitos de chaveamento em alta frequência, de correção do fator de potência e de controle ajustável de velocidade, pois tais cargas

elétricas, assim como alguns poucos outros circuitos não-lineares [53][54], são responsáveis pela geração de ondas de corrente que contêm componentes inter-harmônicas, que são ondas cujas frequências não são múltiplas inteiras da frequência da rede elétrica associada e que, por assim se portarem, não são corretamente representadas pela série discreta de Fourier.

No entanto, a deficiência da série de Fourier em representar cargas não-lineares geradoras de ondas inter-harmônicas não compromete totalmente o método de análise de distorções harmônicas apresentado, pois, além do fato de que a quantidade de cargas elétricas geradoras de frequências inter-harmônicas ainda ser pequeno em comparação à quantidade de cargas não-lineares que não o fazem, também existe o fato de que a presença de distorções harmônicas em áreas residenciais e comerciais ser hoje também grande [45], o que outrora era uma característica quase que restrita às áreas industriais. Isso se dá como resultado dos crescentes desenvolvimento e emprego de novos dispositivos eletrônicos que incorporam, em sua grande parte, circuitos retificadores que ao serem utilizados para se adaptar a forma de energia disponibilizada pela sistema de distribuição para uma forma internamente aproveitável pelo equipamento, geram grandes quantidades de distorções harmônicas.

Circuitos elétricos contendo retificadores, conversores de corrente alternada em corrente contínua ou mesmo conversores de corrente contínua para corrente contínua são muito comuns a equipamentos que operam em baixa tensão, como computadores, televisores, aparelhos de som e afins, e geram grandes quantidades de distorção harmônica [45]. Mesmo circuitos como controladores de fase presentes em sistemas de aquecimento e de ajuste da intensidade em lâmpadas, assim como as próprias lâmpadas fluorescentes, também contribuem para a degradação da qualidade nas redes de distribuição pois também geram distorções harmônicas que são injetadas nas próprias redes às quais são conectados.

Ao lado dos equipamentos eletrônicos citados, algumas outras classes de circuitos e dispositivos são considerados fontes de poluição harmônica e são listados, junto aos primeiros, na Tabela 2.

Tabela 2 - Principais classes de circuitos e dispositivos responsáveis pela geração de distorções harmônicas

Fontes de distorções harmônicas		
Conversores a diodos Conversores a tiristores Controladores de velocidade ajustável Inversores	Fornos elétricos Conversores CC-CC Inversores Lâmpadas fluorescentes Reatores eletrônicos	Transformadores Fontes chaveadas Controladores de fase Lâmpadas mistas

Cada uma das classes de circuitos e dispositivos presentes na Tabela 2 gera, quando associados a cada um dos tipos de equipamentos nos quais são empregues, uma forma de onda de corrente com distorções que são diferentes daquelas geradas por uma segunda combinação entre o mesmo circuito e uma outra classe de equipamentos que o empregue. Essa diferença, em alguns casos, pode ser sutil e aqui é exibida pela Figura 8, que evidencia a pequena diferença existente entre as formas de onda das curvas de corrente de um aparelho de som e de um microondas e a grande diferença existente entre essas duas curvas e uma curva pertencente a um televisor, todos esses equipamentos que empregam uma mesma classe de circuitos retificadores monofásicos a diodos. Essa mesma figura ainda exhibe a palpável diferença existente entre as curvas já citadas e a forma de onda de uma lâmpada fluorescente com reator eletromagnético, que se trata de uma classe de circuitos diferentes da classe utilizada pelas três primeiras cargas citadas.

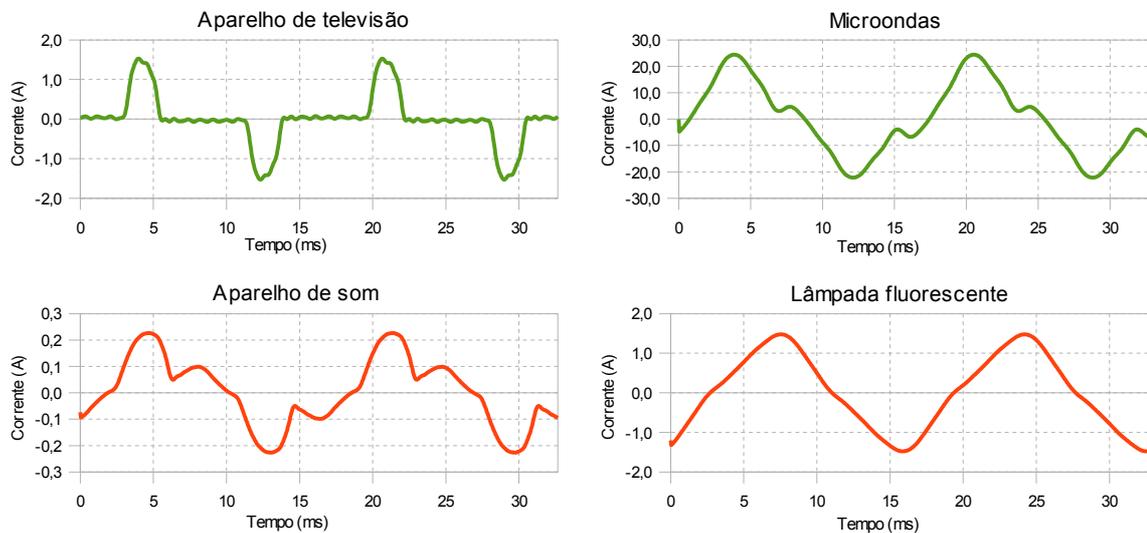


Figura 8 - Formas de onda de corrente pertencentes a algumas cargas elétricas não-lineares.
 Fonte: [45].

A Figura 8 exemplifica de forma visual a existência de diferentes formas de distorções harmônicas que estão associadas a combinações entre classes de circuitos e de equipamentos, mas, no entanto, na prática são utilizados métodos formais para se comparar, segundo o ponto de vista da distorção harmônica, as curvas geradas por cada equipamento. Esses métodos são baseados nos conceitos de distorção harmônica total e de fator de crista de uma onda [52], que são utilizados para se mensurar e classificar o grau de distorção e também de poluição harmônica associados a uma carga não-linear.

A distorção harmônica total, ou DHT, mensura a distorção presente em um sinal $f(t)$ como sendo o percentual que os valores eficazes de suas componentes harmônicas $f_{i=2..n}$ representam perante o valor eficaz de sua componente fundamental f_1 . Sua definição assume a forma apresentada pela equação (1.1.4) e associa um valor de distorção harmônica total 0 (zero) para uma onda senoidal pura.

$$DHT = \frac{\sqrt{\sum_{i=2}^n f_i}}{f_1} \quad (1.1.4)$$

Por sua vez, o fator de crista, ou FC, exprime a relação entre o valor de pico f_{pico} de uma onda $f(t)$ e seu respectivo valor eficaz f_{eficaz} , segundo a equação (1.1.5). Com essa formulação, uma curva senoidal pura com valor de pico 1 e conseqüente valor eficaz 0.707 tem um fator de crista de 1.414.

$$FC = \frac{f_{pico}}{f_{eficaz}} \quad (1.1.5)$$

Para o caso das curvas presentes na Figura 8, os respectivos fatores de crista e distorções harmônicas totais são apresentados na Tabela 3. Esses índices, ao serem analisados, permitem concluir que além de auxiliarem a diferenciar distorções aparentemente iguais, como é o caso das curvas relacionadas a um aparelho de som e a um aparelho de microondas, também permitem quantificar os tipos de distorções harmônicas e, assim, melhor classificar as cargas não-lineares.

Tabela 3 - Distorções harmônicas totais e fatores de crista de algumas formas de onda.

Carga elétrica	DHT	FC
Aparelho de televisão	103,29%	1,67
Aparelho de som	42,80%	1,89
Microondas	29,24%	1,64
Lâmpada fluorescente	12,20%	1,05

A presença de poluição harmônica em sistemas de distribuição de energia elétrica gera conseqüências que dependem do grau de distorção presente e da susceptibilidade de cada equipamento à mesma [45], mas mesmo embora haja tal diferenciação, todas essas conseqüências culminam em prejuízos tanto econômicos devido ao desperdício de recursos ligados à geração, à transmissão e à distribuição de um potencial elétrico que não chega a ser utilizada em sua totalidade pelos clientes dos sistemas de distribuição, quanto em conseqüências técnicas que podem até mesmo comprometer a correta operação dos dispositivos conectados às redes de distribuição e, assim, prejudicar atividades e tarefas do cotidiano.

As principais conseqüências técnicas ligadas à poluição harmônica são listadas na Tabela 4 e agrupadas segundo as classes de equipamentos a elas susceptíveis.

Tabela 4 - Principais conseqüências da presença de poluição harmônica em sistemas de distribuição de energia elétrica.
Fonte: [45].

Equipamento	Conseqüências
Máquinas rotativas	Torque pulsante, ruído audível , sobreaquecimento e perda de vida útil no isolamento
Condutores	Sobreaquecimento e conseqüente diminuição na capacidade de condução de corrente
Medidores de energia	Erros de medição que levam a tarifações indevidas
Relés de proteção	Operação indevida e inatividade
Capacitores	Ressonância e perda de vida útil do dielétrico
Transformadores	Aumento nas perdas em enrolamentos e demais componentes e sobreaquecimento
Equipamentos eletrônicos	Maior susceptibilidade a afundamentos de tensão, inoperância, perda de sincronismo em contadores digitais, geração de imagens distorcidas em tubos catódicos, dentre outros
Iluminação	Diminuição da vida útil em lâmpadas incandescentes, geração de ruídos audíveis em lâmpadas fluorescentes e variações nas

	intensidades de iluminação
Sistemas de telefonia	Geração de ruídos audíveis

Um exemplo claro das conseqüências da presença de distorção harmônica em uma forma de onda pode ser observada na Tabela 5, que apresenta a influência de ondas harmônicas junto ao acréscimo de corrente e à degradação do fator de potência de uma curva cuja fundamental foi fixada em 1A. O fator de potência apresentado foi obtido como sendo o quociente entre a potência ativa e a potência aparente [52].

Tabela 5 - Resultados da adição de ondas harmônicas a uma de senoidal de 1A.
Fonte: [55].

DHT (%)	Acréscimo (%)	Fator de potência total
0	0,00	1,000
10	0,50	0,995
30	4,40	0,958
50	11,80	0,894
70	22,07	0,819
90	34,54	0,743
100	41,42	0,707
120	56,20	0,640
150	80,28	0,555

A análise dos dados presentes na Tabela 5 permite constatar que já à presença de uma DHT da ordem de 50%, cerca de 10% do potencial elétrico consumido é desperdiçado, e esse desperdício pode facilmente ser aumentado caso a DHT seja também aumentada. No caso de uma DHT de ainda 90%, que é típica de aparelhos comuns como televisores, lâmpadas fluorescentes compactas, dentre outros [45], já são desperdiçados cerca de 1/4 da potência elétrica total disponibilizada. Nessas condições, a presença de distorções e da conseqüente

poluição harmônicas em um sistema de distribuição são fatores que facilmente evidenciam o quanto importantes são os esforços destinados a se modelar e a se estudar as cargas elétricas não-lineares.

1.2 Modelos de Representação

A diversidade e o emprego crescente de cargas elétricas não-lineares são fatores que reiteram os esforços relacionados aos estudos sobre a presença desses agentes nas redes de distribuição de energia elétrica e sobre as muitas conseqüências que permeiam essa mesma presença.

O fato de as carga não-lineares serem geradoras de distorções harmônicas e da conseqüente e importante poluição associada permite com que sejam encontradas, na literatura, diversas linhas de estudos relacionadas à presença dessas fontes de poluição. Algumas dessas linhas de pesquisa estão associadas à identificação das fontes de harmônicas em sistemas elétricos, enquanto outras, à mensuração de índices de poluição e à caracterização das conseqüências da presença da mesma, mas, no entanto, todas elas partem do princípio comum de que um modelo de cargas elétricas que seja capaz de agregar as distorções harmônicas deva fazer parte de suas formulações.

As linhas de pesquisa destinadas a estudar os vários aspectos envolvidos às cargas não-lineares são a de caracterização das fontes de poluição harmônica, a que estuda a propagação de ondas harmônicas nas redes de distribuição, a de mensuração de índices de poluição, a que

busca identificar a responsabilidade pela presença de distorção harmônica em pontos de conexão comum aos quais os clientes dos sistemas de distribuição se conectam ao sistema, a que realiza estudos dos impactos inerentes à poluição harmônica, a de modelagem de cargas não-lineares destinadas a serem utilizadas em simulações de sistemas de potência e a linha de pesquisa que busca desenvolver técnicas de reconhecimento das cargas não-lineares presentes nas redes de distribuição. Uma característica presente nessas áreas de investigação é o fato de que cada uma delas utiliza um modelo de cargas elétricas do qual faça parte a questão da distorção harmônica e do qual possam ser extraídas as informações mais interessantes segundo a abordagem que adotem.

A Tabela 6 apresenta uma relação entre os modelos de carga elétricas não-lineares mais comuns e suas respectivas áreas de aplicação.

Tabela 6 - Modelos mais comuns e suas respectivas áreas de aplicação.

Modelo	Área de aplicação
Modelo de circuitos equivalentes (diagramas)	Caracterização das fontes de poluição harmônica
Modelo generalizado e suas derivações (fontes de correntes e de tensão)	Mensuração de índices de poluição Estudos da propagação de ondas harmônicas Estudos dos impactos inerentes à poluição harmônica Identificação das responsabilidades pela emissão harmônica
Modelo baseado na relação corrente-tensão	Caracterização das fontes de poluição harmônica
Modelos baseados em características da forma de onda de corrente	Reconhecimento das cargas não-lineares em redes de distribuição

Dentre os modelos mais comuns presentes na literatura, o modelo de circuitos equivalentes busca representar as cargas elétricas não-lineares através de seus diagramas

elétricos. Esse modelo pode ser considerado como uma abordagem natural e até intuitiva para se representar cargas elétricas, uma vez que os diagramas eletro-eletrônicos são partes integrantes dos projetos de todos os equipamentos disponibilizados ao mercado e, assim, costumam abranger todos os aspectos elétricos das cargas que representam. Exemplos para essa metodologia são apresentados pelas referências [28] e [56] que utilizam, em meio a outros diagramas, a representação presente na Figura 9 para se estudar a poluição harmônica gerada por um computador pessoal típico.

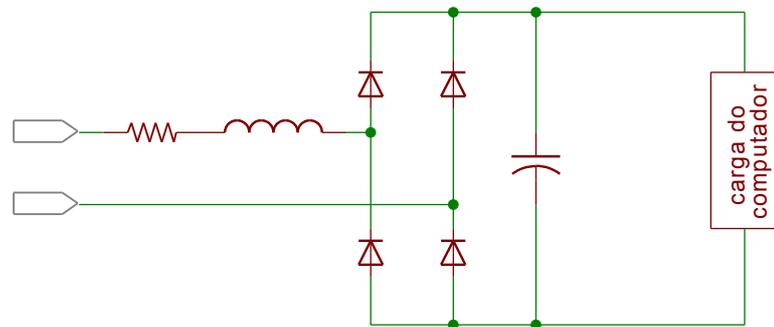


Figura 9 - Diagrama elétrico da seção de retificação presente em computadores pessoais típicos.

Embora essa primeira abordagem citada pareça ser um procedimento intuitivo e possa fornecer resultados satisfatórios [28], um outro modelo para cargas elétricas, chamado de modelo genérico [57], é mais presente na literatura. Esse modelo é apresentado na Figura 10 e tem base no emprego de componentes elétricos passivos para representar as impedâncias equivalentes de equipamentos com predominância linear e engloba a capacitância devida à presença de corretores de fator potência nas redes de distribuição. Já para contemplar cargas não-lineares, esse modelo utiliza fontes de corrente para representar cada uma das frequências harmônicas de interesse para a representação.

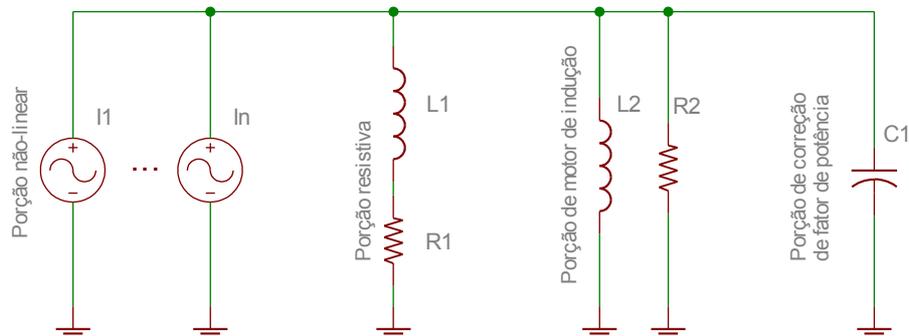


Figura 10 - Modelo genérico para cargas elétricas proposto pelo IEEE.
Fonte: [57].

Para se representar uma carga elétrica não-linear através do modelo genérico de cargas, suas porções relacionadas à representação de um motor de indução e à reconstituição da presença de correção de fator de potência devem ser negligenciadas, o que faz com que a representação inicial seja simplificada para um modelo no qual a porção resistiva seja substituída por uma ramo paralelo R-L ou R-C utilizado para se gerar a harmônica fundamental da carga a representar, enquanto as fontes de corrente gerem cada uma das “n” frequência de importância à representação.

Um exemplo de representação de equipamento eletrônico a partir da simplificação do modelo genérico é apresentada através da Tabela 7 e da Figura 11. O resultado da simulação da representação desse equipamento eletrônico através do aplicativo de simulações ATP [58] é apresentado na Figura 12.

Tabela 7 - Espectros de amplitude e fase de um aparelho de ar condicionado no modo de resfriamento.
Fonte: [45].

Harmônica	Frequência (Hz)	Amplitude (A)	Fase (°)
1	60	13,73	-2,4

2	120	1,72	125,3
3	180	1,65	104,0
5	300	0,22	-163,9
7	420	0,10	138,8
9	540	0,11	113,5

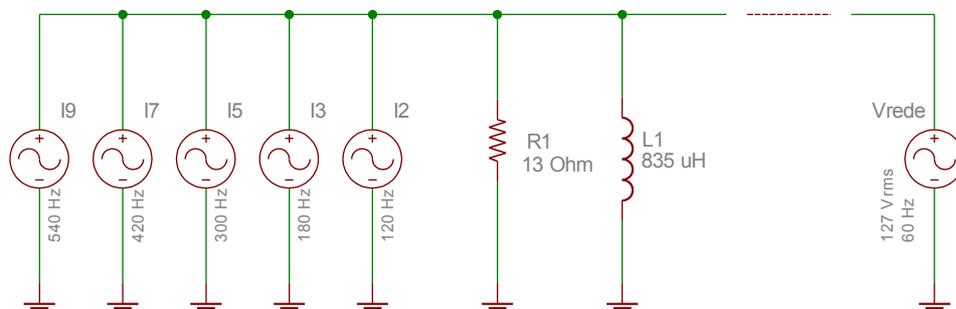


Figura 11 - Modelo simplificado utilizado para se representar um aparelho de ar condicionado no modo de resfriamento

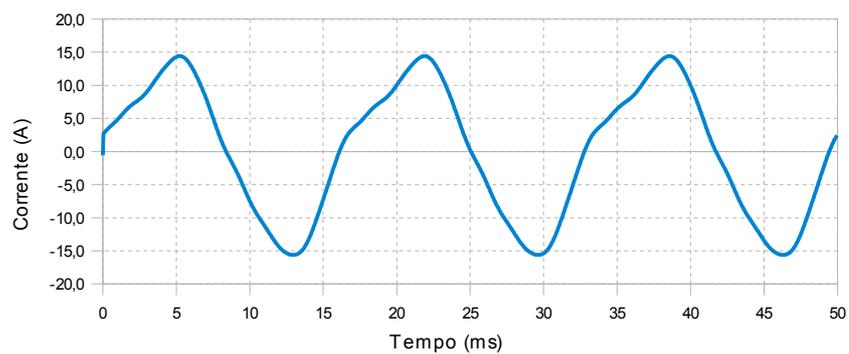


Figura 12 - Forma de onda obtida pela simulação do circuito que representa um aparelho de ar condicionado em modo de resfriamento.

Uma importante questão quanto à representação de cargas geradoras de poluição harmônica através modelo genérico é a necessidade de se utilizar as informações provenientes tanto dos espectros de amplitude quanto dos espectros de fase obtidos das ondas de corrente das cargas a serem representadas, e mesmo embora existam algumas propostas que dispensem a presença desses espectros [59][60], a grande maioria dos trabalhos que utilizam esse modelo fazem uso dos espectros citados.

As áreas de aplicação mais comuns para o modelo genérico e suas derivações são os estudos da propagação de ondas harmônicas [42][61], a mensuração de níveis de poluição harmônica [62], o levantamento dos impactos inerentes à poluição harmônica [16][17], a simulação de cargas emissoras de distorções harmônicas em sistemas de potência e a identificação das responsabilidades pela emissão harmônica nos pontos de conexão comum dos sistemas de distribuição [63].

Uma outra área de estudos que faz uso de modelos de cargas elétricas é a que busca caracterizar as fontes de distorções harmônicas, sendo que um exemplo de modelo para esta área é o que propõe caracterizar as fontes de distorções harmônicas através da relação entre as correntes que consomem e a tensão a elas aplicada [20].

Por sua vez, a área de reconhecimento da presença de cargas elétricas não-lineares em redes de distribuição tem como objetivo levantar quais são as fontes harmônicas ativas em uma determinada localidade e durante um dado período de tempo. Nessa área são encontrados, em meio a alguns outros modelos de mesmo fim [21][25], modelos que buscam representar as fontes harmônicas através da análise de suas formas de onda de corrente [28]-[34].

Um exemplo de aplicação desse modelo é apresentado pela referência [33], na qual a transformada rápida de Fourier [44] é aplicada às formas de onda de corrente de algumas cargas não-lineares para se extrair os respectivos espectros de amplitude e de fase que são, então, utilizados como assinaturas das fontes de distorções harmônicas estudadas.

De uma forma geral, os modelos utilizados para se identificar cargas elétricas não-lineares através da análise de suas formas de onda de corrente são modelos nos quais são empregues técnicas de processamento de sinais que permitam com que as características presentes em uma forma de onda sejam extraídas no formato de índices que possam ser posteriormente utilizados como assinaturas das cargas elétricas em análise.

Essa característica sugere que as propostas voltadas a identificar as cargas não-lineares através de suas formas de onda de corrente compartilhem de um estrutura similar ao diagrama apresentado na Figura 13, assim como o faz a proposição deste trabalho, e que assim contemplem pelo menos um procedimento de processamento das formas de ondas de corrente destinado a extrair características a serem, então, inferidas por um mecanismo baseado em alguma técnica de reconhecimento de padrões que seja capaz de levantar a qual classe de equipamentos pertence uma dada forma de onda de corrente analisada.

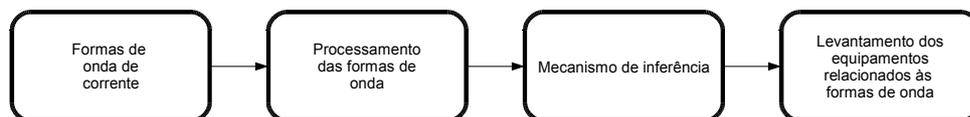


Figura 13 - Diagrama das etapas típicas presentes nos modelos de reconhecimento de cargas não-lineares baseados na análise de ondas de corrente

CAPÍTULO 2

Sistemas de Reconhecimento de Padrões

Sem que se perceba, grande parte das atividades cotidianas são executadas de uma forma que naturalmente induz ou é finalizada por algum tipo de conclusão, como o que ocorre ao se ouvir uma música e então reconhecer parte de seus trechos, ou o que ocorre ao se iniciar a leitura de um romance e mesmo ainda à frente de alguns poucos parágrafos finalizados, reconhecer seu autor devido à forma como as idéias são encadeadas ou mesmo concluídas, ou, ao se passar corriqueiramente por uma rua ou bairro, notar que certas “coisas” presentes em quase todas as residências se ausentam somente nesta ou naquela casa, ou, ainda, notar o que ocorre espontaneamente frente ao passar dos anos, que faz com que se aprenda que grande partes dos eventos naturais se repetem sempre de uma mesma forma ou em uma mesma época do ano, como se obedecessem a alguma regra, roteiro ou padrão pré-estabelecido.

Esses exemplos de situações corriqueiras nas quais se concluem fatos através de observações e conhecimentos prévios são exemplos de aplicações de mecanismos que são

utilizados naturalmente no cotidiano e que envolvem conceitos como o de aprendizagem, o de classificação e o de reconhecimento, considerados como conceitos-chave para a área da inteligência artificial que trabalha com o reconhecimento de padrões.

A área de reconhecimento de padrões é uma área da inteligência artificial na qual os esforços de pesquisa têm como objetivo desenvolver técnicas que sejam capazes tanto de extrair padrões a partir de grupos de atributos de eventos, objetos e afins, quanto implementar mecanismos baseados nos conceitos de reconhecimento, de classificação e aprendizagem que, unidos à extração de padrões, permitam com sejam criados sistemas artificiais capazes de reproduzir ou mesmo suplantar a habilidade humana em se reconhecer alguns padrões específicos.

O conceito de padrões, no entanto, não segue nenhuma definição clássica, pois pode representar entidades que estão de alguma forma relacionadas a significados estritamente associados ao contexto nos quais são empregues, como é o exemplo dos padrões musicais, que classificam tanto a forma de vibração com a qual as cordas de alguns instrumentos vibram ao produzirem certas notas quanto os gêneros de música clássico, erudito e popular; ou mesmo como é o caso das seqüências através das quais os sintomas de uma doença se intercalam e que permitem, assim, com que os profissionais da área de saúde possam ter estimativas de que a respectiva seqüência de sintomas é uma padrão inerente à presença dessa ou daquela doença.

A Tabela 8 apresenta a relação de métodos até então empregues junto ao desenvolvimento de sistemas de reconhecimento de padrões.

Tabela 8 - Taxonomia dos métodos de reconhecimento de padrões.
Fonte: [64].

Decisão teórica	Mapeamento associativo
Estatístico	Redes neurais artificiais sem retroalimentação
Método paramétrico	Perceptron de múltiplas camadas
Método não-paramétrico	Ligações funcionais
Estimação bayesiana	Funções de bases radiais
Teoria dos grafos	Redes auto-organizáveis
Baseados em regras	Teoria adaptativa ressonante
Regras lógicas binárias	Mapas de características auto-organizáveis
Regras lógicas difusas	Mapas auto-organizáveis difusos
Estrutural/sintático	Redes híbridas
Autômatos	Aprendizagem por vetor de quantificação
Determinístico	Redes neurais probabilísticas
Estocástico	Mapas associativos difusos
Redes neurais recorrentes	Aprendizagem difusa por vetor de
Mapas associativos bidirecionais	quantificação

2.1 Processos de Reconhecimento e Classificação

Os mecanismos de reconhecimento de padrões consistem em se associar um dado elemento de uma população P a uma subpopulação S_i de P que contenha somente elementos cujos atributos mensurados gerem padrões que se aproximem daqueles pertencentes ao elemento inicial.

Em uma notação formal, pode-se representar um padrão como sendo um vetor \hat{a} de atributos $a_{i=1..m}$ mensurados a partir de um elemento pertencente a uma população P de elementos não idênticos.

$$\hat{a} = (a_1, \dots, a_m). \quad (2.1.1)$$

Usando a notação apresentada, um processo de reconhecimento de padrões consiste em se aplicar uma transformação $T\{\}$ a um vetor de padrões \hat{a} de “m” elementos a fim de se extrair um vetor de características \hat{x} com “n” elementos (geralmente $n < m$), como é apresentado na equação (2.1.2). Então, o vetor de características \hat{x} é aplicado a um procedimento de tomada de decisões $M\{\}$ que o associa a um identificador \hat{z} a representar a classe à qual pertence o elemento inicial representado pelo vetor de padrões \hat{a} . De uma forma resumida, um processo de reconhecimento de padrões associa um vetor m-dimensional \hat{a} de atributos a um vetor n-dimensional \hat{z} através da mapeamento $R\{\} = M\{T\{\}\}$, assim como o presente na equação (2.1.4).

$$\hat{x} = T\{\hat{a}\} = (x_1, \dots, x_n) \quad (2.1.2)$$

$$\hat{z} = M\{\hat{x}\} = (z_1, \dots, z_k) \quad (2.1.3)$$

$$\hat{z} = R\{\hat{a}\} \quad (2.1.4)$$

Um diagrama representando as etapas de um processo típico de reconhecimento de padrões é apresentado na Figura 14.

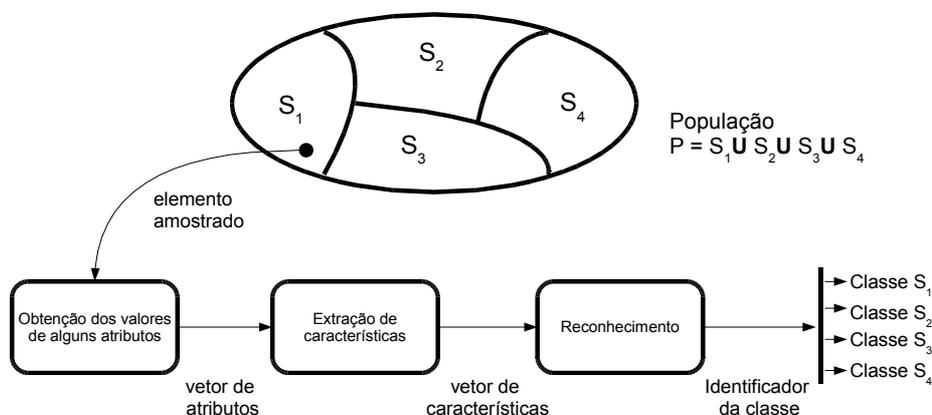


Figura 14 - Procedimentos presentes em um processo de reconhecimento de padrões.

Para o caso em que um elemento da população P é reconhecido como pertencente a uma subpopulação S_i de P que contenha um único elemento, então o processo de reconhecimento passa a ser designado como sendo um processo de identificação. A identificação de um elemento corresponde a levantar o menor conjunto possível de seus atributos que permitam com que seus respectivos vetores de padrões \hat{a} e de características \hat{x} geram um vetor de identificação \hat{z} que o associe a uma classe C_i a conter apenas o próprio elemento representado vetor de padrões inicial. Em outras palavras, o processo de identificação remete ao levantamento das atributos de um elemento que sejam capazes de individualizá-lo perante a população da qual provenha e, assim, o classifique como um elemento pertencente a uma classe unitária.

Por sua vez, o processo de classificação consiste em se agrupar em classes os elementos de uma população que compartilhem de características em comum e assim formem subpopulações que não possam ser mais subdivididas ao se aplicar os mesmos requisitos que as criaram. Nesse contexto, o conceito de identificação já citado pode ser definido como um processo de classificação no qual são aplicados requisitos classificatórios até que todas as classes existentes sejam subdivididas em conjuntos unitários.

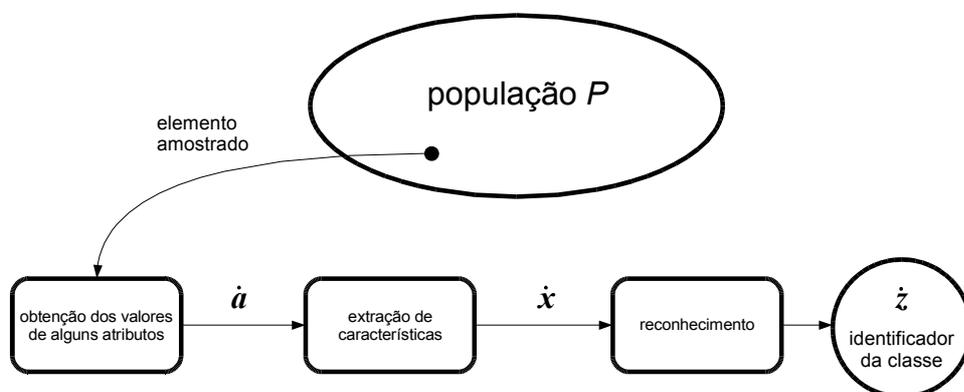


Figura 15 - Procedimentos presentes em um processo de classificação.

Uma outra forma possível de se observar e então diferenciar os conceitos de reconhecimento e classificação é a que faz uso dos conceitos de raciocínio e aprendizagem. O conceito de raciocínio está ligado à capacidade de se obter conclusões perante fatos e valendo-se para tanto de conhecimentos prévios, como o que exatamente ocorre nos processo de reconhecimento de padrões e pode ser elucidado através do exemplo já citado à respeito de profissionais da área da saúde que se utilizam de conhecimentos já adquiridos para se diagnosticar e então concluir sobre possíveis enfermidades em seus pacientes. Assim, quando se fala em reconhecimento de padrões, fala-se em raciocínio. Já o processo de classificação está ligado ao conceito de aprendizagem no sentido de que ambos representam a mudança de comportamento a partir do acúmulo de experiências. Um exemplo para isso é o simples caso em que uma criança aprende o conceito de coisas boas e ruins e passa a utilizá-lo durante sua infância para adicionar suas experiências ou ao grupo de coisas boas ou ao grupo de coisas ruins e, a partir de então, utiliza tais grupos para se concluir que um fato novo, sobre o qual não se tinha uma posição anterior, é, agora, uma boa oportunidade ou algo do qual se prevenir.

Uma maneira mais formal de se ainda observar e então diferenciar as duas áreas do reconhecimento de padrões que são os processo de reconhecimento e de classificação é apresentada pela referência [64]. Segundo essa maneira, a partir de uma população P de elementos não-idênticos representados por seus vetores de características \mathbf{x} e particionada em classes equivalentes C_1, \dots, C_k , o processo de classificação para esta população consiste em se decidir se múltiplos vetores de características pertencem a uma mesma classe ou a classes equivalentes, enquanto o processo de reconhecimento consiste em se concluir se um vetor de características proveniente de um dado elemento de P é, ou não, similar a um ou mais vetores de características que representem um classe.

2.2 Aplicação

A presença de padrões em muitas das áreas de interesse aos esforços de pesquisa e mesmo em muitas situações do cotidiano permite com que diversas linhas de desenvolvimento de sistemas de reconhecimento sejam criadas e mantidas. Duas dessas áreas, a de reconhecimento estatístico de padrões e a linha de reconhecimento sintático de padrões, são consideradas duas das abordagens clássicas do sistemas de reconhecimento e são baseadas, respectivamente, em modelos estatísticos para representar as informações que manipula e para implementar o mecanismo de reconhecimento que utiliza, e em estruturas advindas da interconexão entre características dos objetos que analisa. Exemplos de aplicação da abordagem de reconhecimento estatístico de padrões são o reconhecimento automático de caracteres escritos à mão [65], o reconhecimento de faces e o reconhecimento de linguagem (fala), enquanto alguns exemplos para a abordagem sintática são os reconhecimentos de palavras, frases e sentenças.

Às duas linhas já citadas sobre as abordagens de reconhecimento de padrões podem ser unidas duas outras abordagens para que, assim, seja formado um grupo que represente as linhas de reconhecimento de padrões cujas aplicações são mais encontradas na literatura. Essas outras duas abordagens são a que trata do reconhecimento difuso de padrões e a que trabalha com redes neurais artificiais [64] para se desenvolver mecanismos de mapeamento a serem utilizados para se construir sistemas de reconhecimento de padrões. A área de reconhecimento difuso de padrões tem base na aplicação da teoria de conjuntos difusos [66],

que implementa uma lógica capaz de manipular conjuntos e valores que apresentem incertezas. Já a abordagem do reconhecimento neural de padrões faz parte de uma área maior que representa todas as abordagens que fazem uso do chamado mapeamento associativo. Essa abordagem tem base na aplicação de modelos neurais como estruturas básicas para seus mecanismos de reconhecimento e classificação.

A abordagem de reconhecimento de padrões com base em modelos neurais pertence a uma linha de sistemas de reconhecimento e classificação de padrões na qual são utilizadas as capacidades intrínsecas dos modelos neurais em se aprender, generalizar e se adaptar a condições tanto lineares quanto não-lineares. Essas características atribuem ao modelo neural a condição de aproximadores universais que, desta forma, podem até mesmo emular modelos estatísticos e sintáticos de reconhecimento. Exemplos de aplicação dessa abordagem são os próprios sistemas de reconhecimento facial [67], que também são implementados através da abordagem estatística, e os de reconhecimento de voz [68], dentre muitos outros.

2.3 Operação e Treinamento

Indiferentemente ao emprego de uma abordagem neural, estatística, sintática ou mesmo difusa para se desenvolver os sistemas de reconhecimento de padrões, a utilização desses sistemas está sempre relacionada aos seus procedimentos básicos de operação e de treinamento. O procedimento de operação de um sistema de reconhecimento consiste na utilização prática do mesmo, em que amostras de elementos de uma população fornecidas

como parâmetros a suas entradas resultam na disponibilidade de identificadores das classes às quais pertençam.

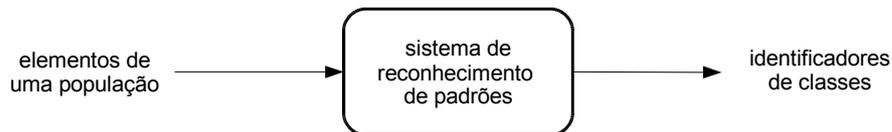
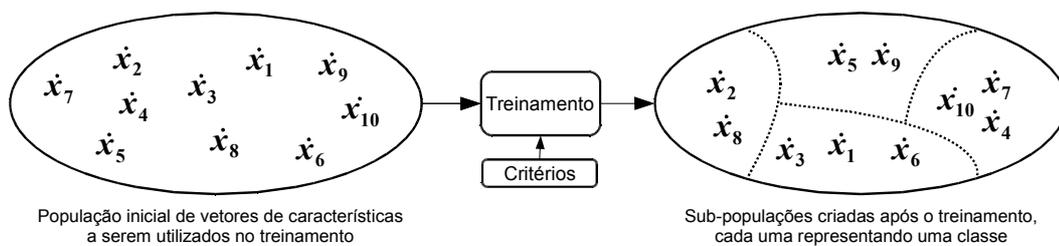


Figura 16 - Sistema de reconhecimento de padrões em seu modo de operação

No entanto e como já mencionado, o procedimento de operação de um sistema de reconhecimento é equiparado a um processo de raciocínio no qual conclusões são construídas mediante a aplicação de regras à uma base de conhecimento previamente adquirida e que, assim, deve fazer parte do próprio sistema de reconhecimento de padrões. A base de conhecimento do qual fazem uso esses sistemas são obtidas em um processo específico chamado de *treinamento*.

O modo de treinamento dos sistema de reconhecimento de padrões nada mais é que um processo de classificação com o qual esses sistemas aprendem a agrupar os elementos de uma população em subpopulações que respeitem certos parâmetros de similaridade, assim como o exemplificado na Figura 17.



População inicial de vetores de características a serem utilizados no treinamento

Sub-populações criadas após o treinamento, cada uma representando uma classe

Figura 17 - Ilustração do processo de treinamento no ponto de vista dos conjuntos de vetores de características envolvidos

Uma outra forma de entender os processos de treinamento e até melhor compreender algumas de suas dificuldades é quando se faz menção aos espaços vetoriais das características \hat{x} . Um vetor de características \hat{x} é, assim como o já apresentado pela equação (2.1.2), um vetor n-dimensional que contém “N” características obtidas a partir de um vetor de atributos \hat{a} . Por espaço vetorial do vetor \hat{x} entende-se, assim, o espaço vetorial euclidiano N-dimensional E^N que o comporta. Se for tomado um espaço vetorial E^2 , então os respectivos vetores de características serão da forma $\hat{x}=(x_1, x_2)$, com x_1 e x_2 sendo escalares a caracterizar o vetor. Um exemplo de espaço vetorial para esse caso seria, então, o plano apresentado na Figura 18 (a), onde os vetores que compartilhem de uma mesma similaridade são representados por um mesmo símbolo gráfico.

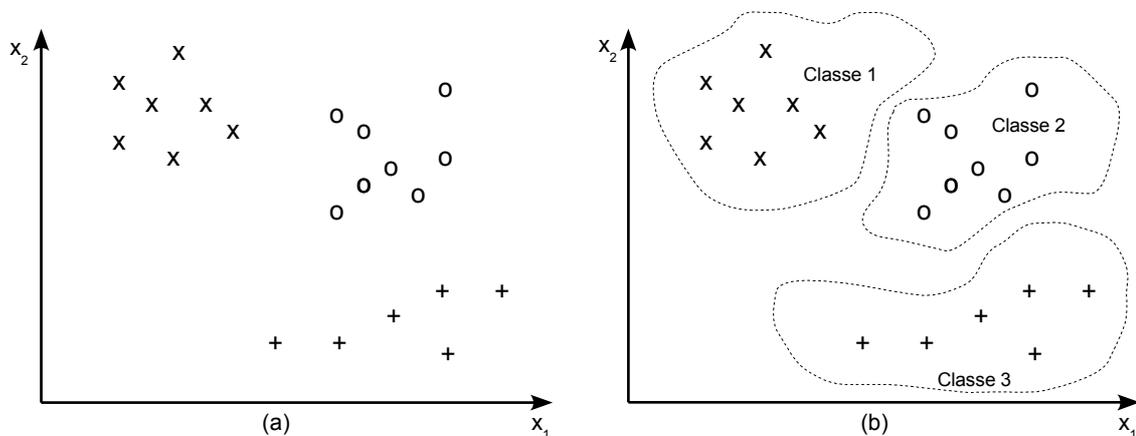


Figura 18 - Plano vetorial comportando uma distribuição de vetores (a) e uma tentativa de classificá-los (b).

Nas condições apresentadas e supondo que o objetivo do processo de treinamento de um sistema de reconhecimento de padrões seja aprender a separar os vetores de características segundo a mesma similaridade que faz com que eles sejam representados por símbolos gráficos iguais ou diferentes, então o processo de treinamento deverá ser capaz de particionar o espaço vetorial presente na Figura 18 (a) de uma forma semelhante ao apresentado na

Figura 18 (b), em que a presença das três classes de atributos é elucidada pela delimitação a ser gerada em um processo de treinamento. Para atender a esse objetivo e então permitir que um sistema de reconhecimento possa corretamente identificar um vetor de atributos como pertencente a uma dentre as três classes existentes, pode ser utilizada tanto uma metodologia de treinamento supervisionado quanto uma metodologia de treinamento não-supervisionado.

A metodologia de treinamento supervisionado consiste em fornecer ao sistema de treinamento vetores de características juntamente a seus já conhecidos identificadores de classes, para o que sistema se adapte automaticamente a cada par formado por um vetor de característica e seu identificador de classe. Já para o caso do treinamento não-supervisionado, que é também conhecido como classificação auto-organizável [65], os sistemas de reconhecimento executam ambos os processos de operação e de treinamento para cada vetor de atributos que analisem, e no caso de um vetor não ser identificado como pertencente em nenhuma das classes já existentes, uma nova classe é criada e associada ao vetor e o funcionamento interno do sistema é atualizado.

Exemplos de aplicações da metodologia de treinamento supervisionado que resolveriam a questão do particionamento do espaço vetorial apresentado na Figura 18 (a) são os sistemas de reconhecimento de padrões baseados nos modelos neurais de *perceptrons* de múltiplas camadas [64]. Esses sistemas podem ser utilizados para se particionar o espaço vetorial em questão assim como é feito na Figura 19, onde as linhas P12, P23 e P32 particionam o espaço vetorial inicial em regiões que podem ser unidas para que se aproxime do particionamento ideal apresentado pela Figura 18 (b). Para esse caso, o trabalho do processo de treinamento seria o de ajustar as linhas L12, L23 e L32 de forma que as regiões resultantes criassem um particionamento capaz de representar as classes 1, 2 e 3 existentes.

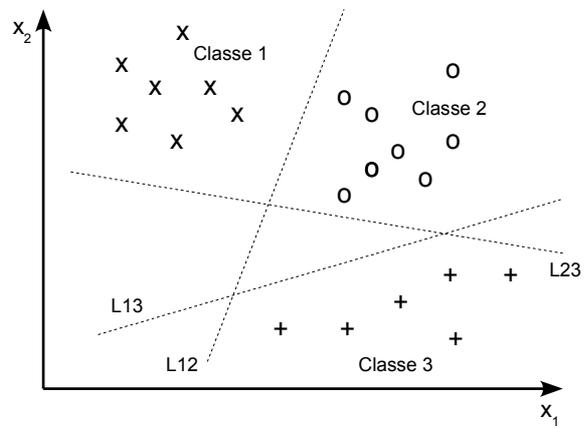


Figura 19 - Particionamento de um espaço vetorial através de linhas que separem regiões.

CAPÍTULO 3

Redes Neurais Artificiais

As redes neurais artificiais consistem em um grupo de técnicas de grande emprego e sucesso dentre todo o ferramental disponível para se processar informações. Esse grupo de técnicas representa ferramentas computacionais baseadas na aplicação de modelos de unidades neurais artificiais simples, que foram inicialmente propostas por *McCulloch e Pitts* em 1943 [69] na tentativa de replicar o funcionamento de neurônios biológicos e que, quando agrupadas em certas estruturas, proporcionam às redes que criam as notáveis habilidades de executar processamento paralelo, de armazenar informações e de se adaptarem ou, em outras palavras, aprenderem. Unidos, esses atributos possibilitam que essas redes artificiais sejam utilizadas tanto em problemas simples e de caráter linear quanto junto a problemas complexos e não-lineares, o que possibilita, assim, com que essas mesmas redes se façam presentes em várias áreas de atuação como a de processamento de linguagem e de imagens, a de controle de processos, a de análise de investimentos, a de mineração de dados e a da medicina, além de

algumas outras áreas dentre as quais a de reconhecimento de padrões representa uma daquelas onde o emprego e o sucesso das redes neurais artificiais são mais notados.

3.1 Neurônios Artificiais

A primeira tentativa de desenvolver um modelo funcional para neurônios biológicos foi apresentada em 1943 pelos pesquisadores *Warren Sturgis McCulloch* e *Walter Pitts* em seu trabalho “*A logical calculus of the ideas immanent in nervous activity*” [69], cuja proposição principal é apresentada na Figura 20 em companhia ao modelo biológico que a inspirou.

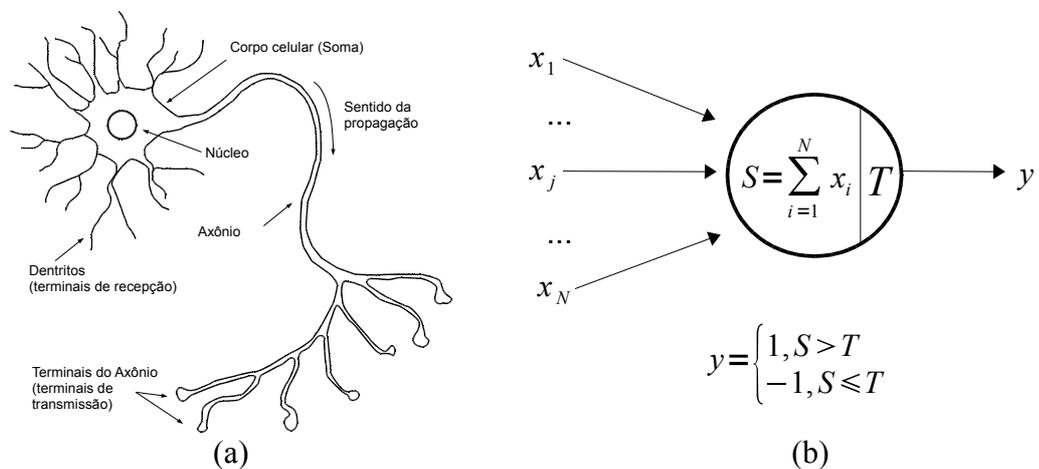


Figura 20 - Neurônio biológico (a) e seu respectivo modelo funcional (b) segundo *McCulloch* e *Pitts*.

Esse primeiro modelo neural busca reproduzir a estrutura funcional de um neurônio biológico através de linhas de entrada x_i a representarem seus terminais de recepção, através de uma unidade de processamento representando a respectiva estrutura somática e também através de uma linha y , que é responsável por transmitir e assim disponibilizar o

resultado do processamento do neurônio como sendo um estado ativo (1) ou inativo (-1), que é dado segundo o resultado da soma S de suas entradas x_i em relação a seu nível de ativação T .

Embora essa primeira tentativa seja uma proposta válida de reproduzir uma estrutura neural em termos funcionais, o trabalho inicial de *McCulloch e Pitts* apresenta algumas limitações, sendo que a principal delas é sua baixa capacidade de aprender, a qual significa, para o caso de um neurônio artificial, a possibilidade de ajustar sua estrutura interna de modo que sua saída apresente um valor pré-estabelecido quando a sua entrada for disponibilizado um dado vetor $\hat{x} = (x_1, \dots, x_N)$. Nesse sentido, toda a capacidade de aprendizagem do modelo de *McCulloch e Pitts* está restrita ao ajuste de seu único parâmetro interno, seu nível de ativação T . Essa limitação, no entanto, é sanada através do *princípio da correlação hebbiana*, a qual diz que a ativação de um neurônio através de uma dada entrada x_j está diretamente relacionada ao peso w_j que essa mesma entrada tem perante o processo de ativação desse neurônio. Somada a essa proposição, as contribuições de *Marvis Minsky e Dean Edmon* [70], de 1951, e de *Rosenblatt, Wightman e Martin* [71], de 1957, resultaram no modelo de neurônio conhecido como *perceptron*.

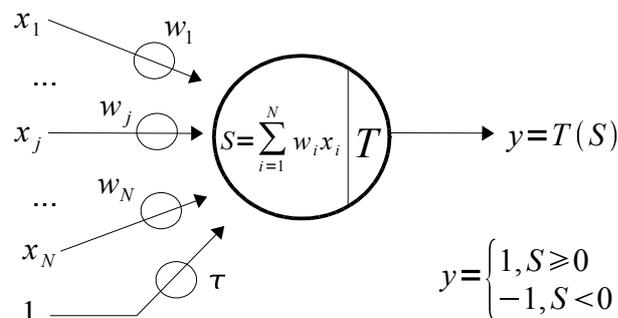


Figura 21 - Modelo neural denominado *perceptron*.

O *perceptron* é um modelo neural que adiciona à proposição inicial de *McCulloch* e *Pitts* uma maior capacidade de aprendizagem. Essa adição é feita através da inclusão dos pesos ajustáveis $w_{i=1..N}$ para as entradas $x_{i=1..N}$ e da representação do nível de ativação também através de um peso ajustável τ . Com esses pesos, a capacidade do neurônio artificial de aprender a reproduzir uma dada saída y quando em suas entradas houver um vetor \mathbf{x} é melhorada de forma significativa. Um outro parâmetro importante a esse modelo é a sua função de ativação T , que passa a condicionar a saída y segundo a aplicação de uma função matemática à diferença entre o somatório S e o nível de ativação τ , sendo que as funções de ativação mais comuns são as unipolares e bipolares apresentadas na Figura 22. Dessa forma, para um *perceptron* com função de ativação bipolar como a apresentada na Figura 22 (b), sua operação passa a ser representada pelas equações (3.1.1), (3.1.2) e (3.1.3), que permitem evidenciar, dentre outras coisas, a capacidade desse modelo neural em particionar o espaço vetorial de seus vetores de entrada.

$$S = \sum_{i=1}^N w_i x_i \quad (3.1.1)$$

$$T(\alpha) = \begin{cases} 1, & \alpha \geq \tau \\ -1, & \alpha < \tau \end{cases} \quad (3.1.2)$$

$$y = T(S) \quad (3.1.3)$$

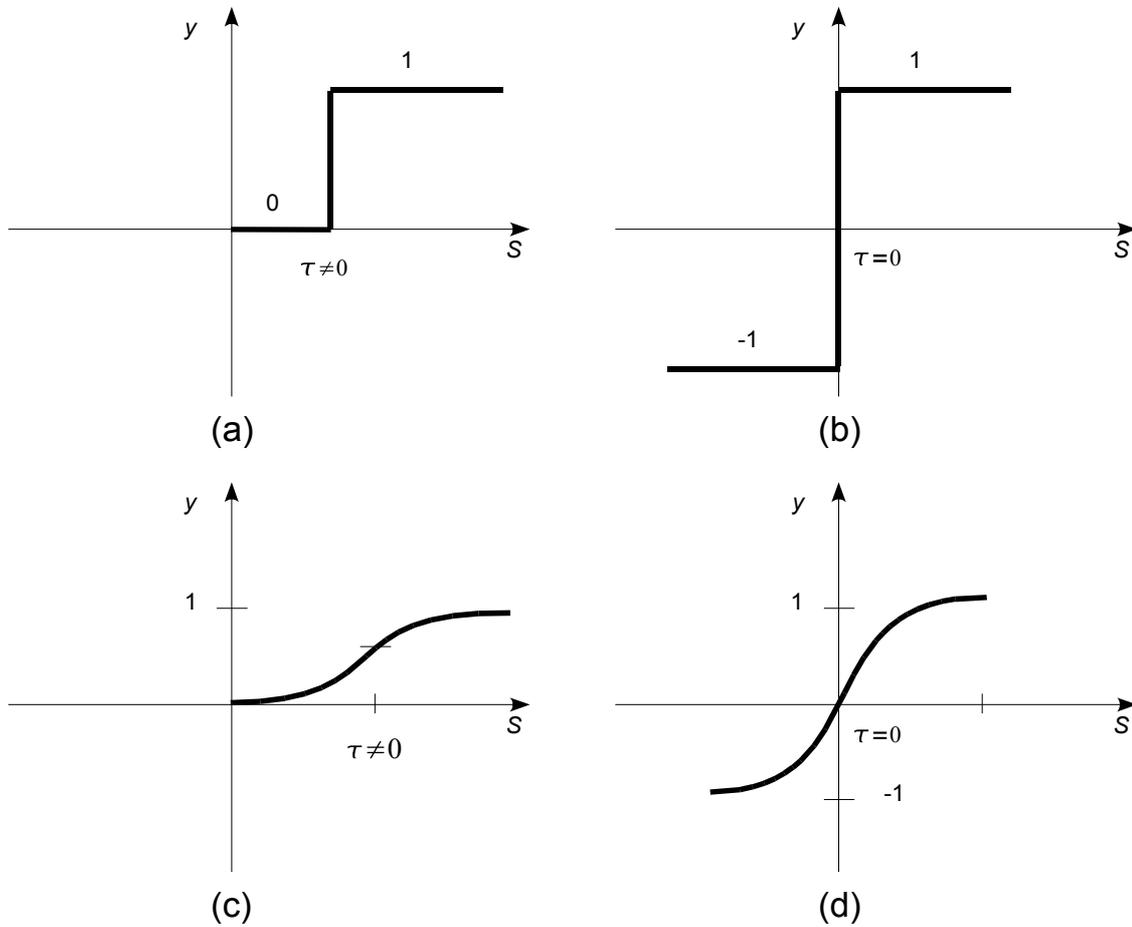


Figura 22 - Exemplos de funções de ativação unipolares (a)(c) e bipolares (b)(d) mais comuns. Fonte: [64].

Um forma de observar a capacidade do *perceptron* de particionar o espaço vetorial de seus vetores de entrada \hat{x} é expandir a equação (3.1.3) a partir das demais relacionadas e igualá-la a 0 (zero). Nesse caso, o que se obtém é a equação analítica (3.1.4), que define um hiperplano P no espaço vetorial dos próprios vetores de entrada. Esse plano particiona o respectivo espaço vetorial em duas regiões distintas P^+ e P^- definidas pelas equações (3.1.5) e (3.1.6). Um exemplo desse tipo de particionamento é apresentado na Figura 23, que retrata a divisão de um espaço vetorial bidimensional efetuada por um *perceptron* com vetores de entrada da forma $\hat{x} = (x_1, x_2)$.

$$w_1 x_1 + w_2 x_2 + \dots + w_N x_N - \tau = 0 \quad (\text{P}) \quad (3.1.4)$$

$$w_1 x_1 + w_2 x_2 + \dots + w_N x_N > \tau \rightarrow \dot{\mathbf{x}} \in \text{P}^+ \quad (3.1.5)$$

$$w_1 x_1 + w_2 x_2 + \dots + w_N x_N < \tau \rightarrow \dot{\mathbf{x}} \in \text{P}^- \quad (3.1.6)$$

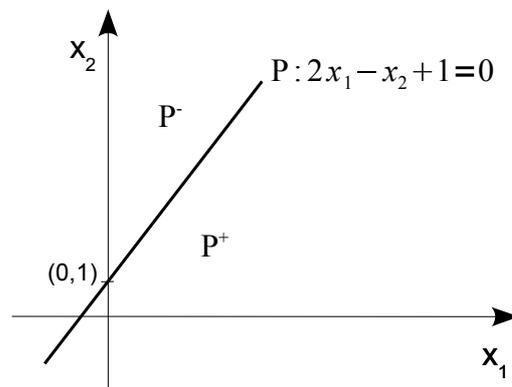


Figura 23 - Exemplo de partição de um espaço vetorial bidimensional efetuada por um *perceptron*.

Essa capacidade dos *perceptrons* em particionar o espaço vetorial de seus vetores de entrada é justamente uma de suas características que os levou a formar estruturas tanto maiores quanto mais capazes de efetuar o particionamento de espaços vetoriais.

3.2 Perceptrons de Múltiplas Camadas

A capacidade do modelo funcional do *perceptron* [71] em particionar o espaço vetorial de seus vetores de entrada é o fundamento para a criação de estruturas maiores denominadas

de *perceptrons de múltiplas camadas*, que são baseadas nos próprios *perceptrons* e têm como finalidade realizar operações complexas com regiões dos espaços vetoriais com as quais trabalhem. Um forma inicial dessas estruturas é representada pela Figura 24, em que uma camada de N entradas é relacionada a uma camada de M *perceptrons* através de pesos w_{nm} . Por sua vez, a camada intermediária de M neurônios produz uma camada de saída com também M resultados. A camada contendo M *perceptrons* é responsável pelo processamento realizado pela estrutura e permite, assim, particionar um mesmo espaço vetorial em uma quantidade máxima de 2^M e mínima de M+1 regiões convexas e distintas. A Figura 25 exemplifica o caso em que a menor quantidade de M+1 regiões particiona um plano vetorial bidimensional em 4 (M+1) regiões distintas através de 3 (M) planos não coincidentes e paralelos criados por uma camada de processamento contendo 3 neurônios.

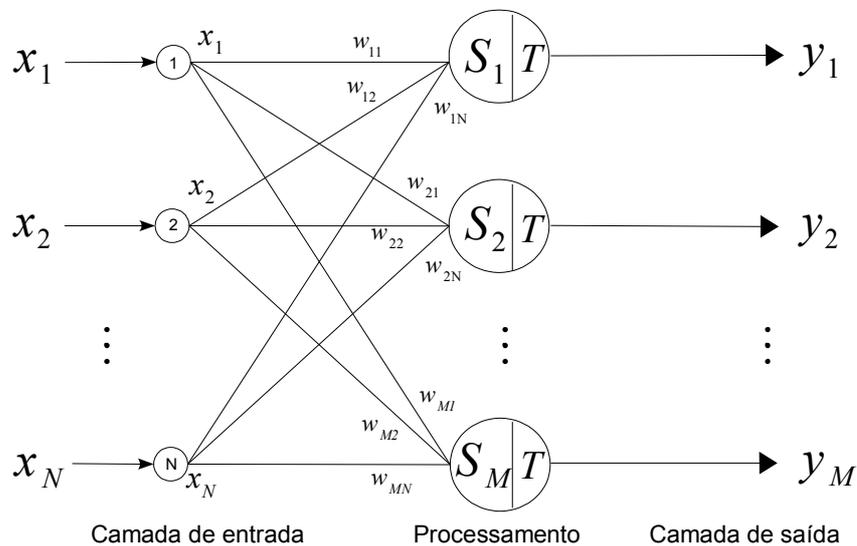


Figura 24 - Camada de M *perceptrons* com N entradas e M saídas.

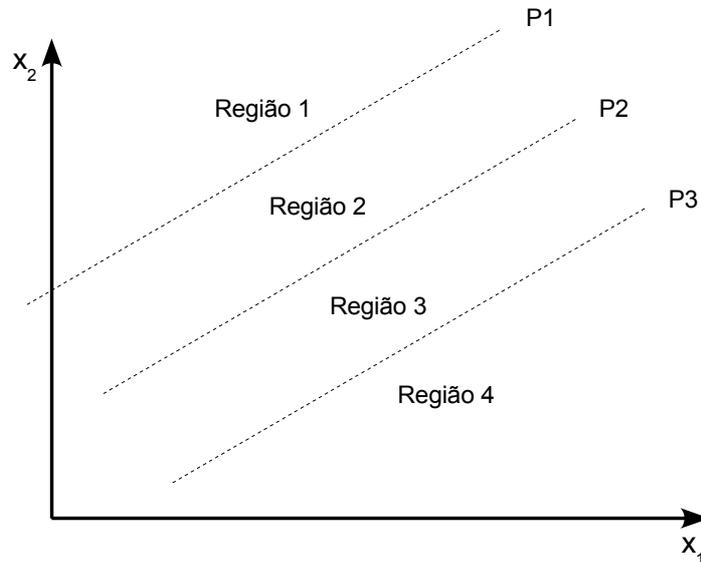


Figura 25 - Divisão de um espaço vetorial bidimensional em quatro regiões devido à presença de 3 planos (linhas) não coincidentes e paralelos.

A capacidade de uma camada de *perceptrons* em particionar um mesmo espaço vetorial em múltiplas regiões convexas pode ser aplicada como um procedimento inicial a resolver os problemas de classificação típicos em sistemas de reconhecimento de padrões, que aqui foram primeiramente exemplificados através da Figura 18 (a) e que, agora, têm um outro exemplo reproduzido de maneira similar pela Figura 26. Assim como o já mencionado, um problema de classificação requer que um espaço vetorial como o apresentado na Figura 26 (a) seja particionado de uma forma similar à apresentada na Figura 26 (b) e que desta forma possibilite com que os vetores $\hat{x} = (x_1, x_2)$, pertencentes a cada uma das classes 1, 2 e 3, sejam separados dos pertencentes às demais. Uma forma de atender a esse requisito é utilizar, para esse caso, uma estrutura similar à estrutura presente na Figura 27, que é derivada da apresentada na Figura 24. Essa estrutura pode ter seus pesos w_{mn} ajustados de forma que sejam criados planos capazes de particionar o espaço vetorial inicial em uma forma similar ao que é feito na Figura 26 (b) pelos planos P12, P13 e P23. Cada um desses planos particiona o espaço vetorial em duas regiões que contemplam, cada uma, os elementos de uma das duas

classes que estejam relacionadas a cada um dos planos. Esse processo permite com que, ao término dos três processos de separação, os elementos de cada uma das classes sejam corretamente separados daqueles pertencentes às outras classes.

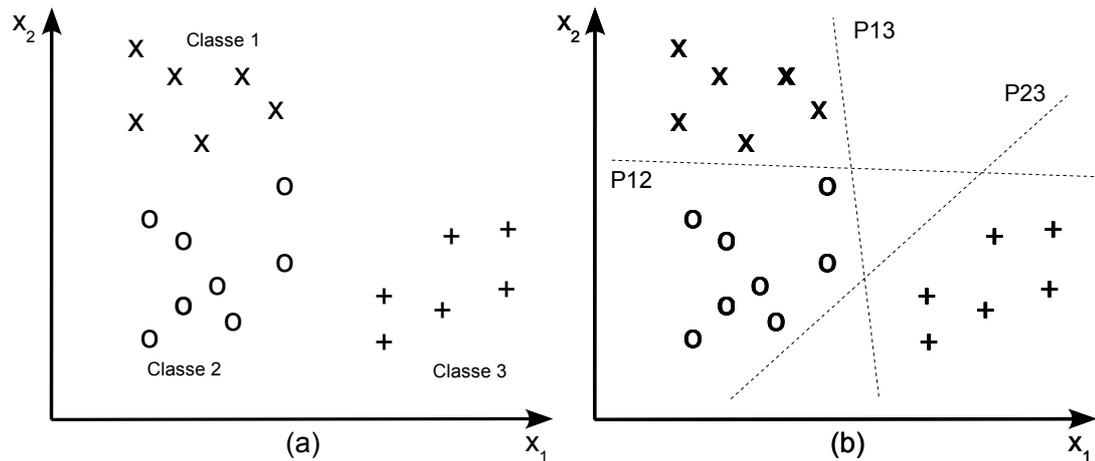


Figura 26 - Distribuição de vetores em um espaço bidimensional (a) e a respectiva tentativa de separá-los (b) através de planos (linhas).

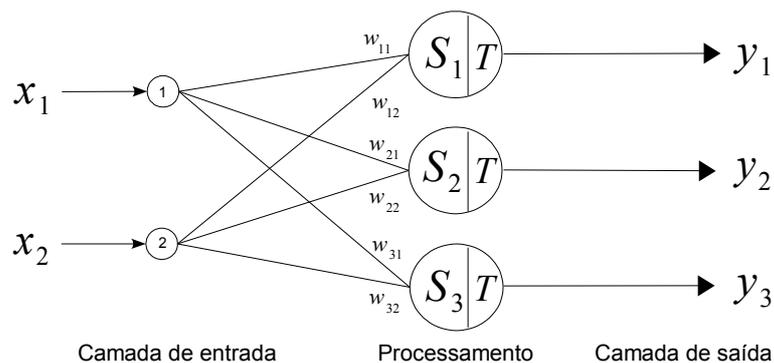


Figura 27 - Estrutura neural contendo 2 elementos na camada de entrada e 3 perceptrons na camada de processamento.

Embora uma solução inicial para o particionamento proposto na Figura 26 seja a estrutura proposta na Figura 27, o espaço vetorial do problema é particionado em 7 regiões distintas, assim como o apresentado na Figura 28, e não em apenas 3, como seria o ideal para

para esse caso. Cada uma dessas regiões é resultado da interseção dos planos criados por cada um dos três *perceptrons* presentes na camada de processamento da estrutura, e uma vez que um único *perceptron* particiona o espaço vetorial em duas regiões distintas, então cada uma das 7 regiões convexas presentes é resultado de uma combinação particular entre cada uma das regiões criadas pelos três *perceptrons* utilizados. Desta forma, para que um vetor pertença à classe 2, por exemplo, ele deve pertencer à região 6, que é criada através de uma combinação específica entre uma das regiões criadas pelo plano P12 ao separar as regiões entre as classes 1 e 2, uma das regiões criadas pelo plano P13 que separa as classes 1 e 3 e uma das regiões criadas pelo plano P23 ao delimitar as regiões entre as classes 2 e 3. Em outras palavras, o resultado da saída y_1 da estrutura utilizada diz à qual das duas regiões criadas pelo respectivo *perceptron* pertence o vetor em questão, enquanto as demais saídas y_2 e y_3 fazem o mesmo, mas levando-se em conta as regiões que respectivamente criam e assim deixam claro que as saídas $y_{i=1..3}$, por si próprias, não permitem com que estruturas similares ao apresentado na Figura 24 forneçam resultados que apontem diretamente à qual das classes pertence um dado vetor disponibilizado a suas camadas de entrada.

O fato das saídas dos *perceptrons* presentes em estruturas como a apresentada na Figura 24 fornecerem resultados que remetem somente às regiões separadas pelos planos que cada um cria, não torna possível que os mesmos, quando em uma única camada de processamento, gerem informações que contemplem resultados que agreguem as condições de todos os neurônios envolvidos na mesma camada.

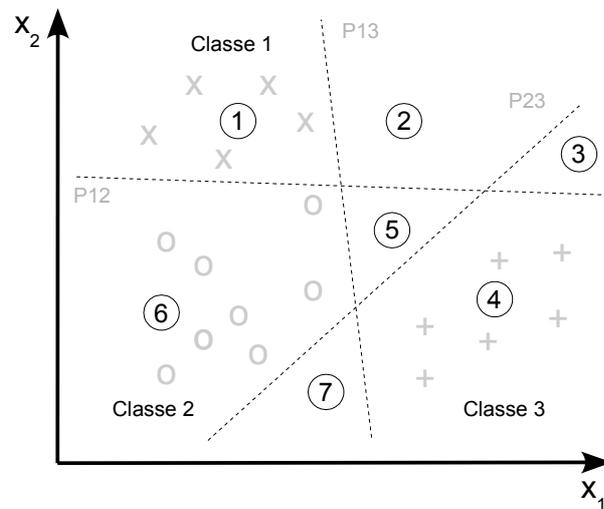


Figura 28 - Regiões resultantes do particionamento efetuado por uma camada de 3 *perceptrons*.

Uma forma de resolver a deficiência citada é adicionar mais camadas de neurônios à estrutura presente na Figura 24 e assim formar os chamados *perceptrons de múltiplas camadas* (PMC). Essas redes permitem expandir a capacidade de processamento apresentada por uma única camada de neurônios e tornam possível com que os resultados que disponibilizem sejam combinações dos resultados obtidos pelos neurônios presentes nas camadas de processamento que antecedem suas saídas, o que significa dizer que para o problema apresentado Figura 26, é possível empregar uma rede neural artificial de *perceptrons* de múltiplas camadas cuja camada de saída forneça como resultado os identificadores das classes às quais pertencem cada um dos vetores disponibilizados a sua camada de entrada.

A Figura 29 apresenta um exemplo de arquitetura de uma rede neural de *perceptrons de múltiplas camadas*, que também é conhecida como rede neural artificial sem retroalimentação. Essa rede exemplifica o caso de uma rede contendo três camadas, sendo uma de entrada, com S^0 elementos, e duas outras camadas com as respectivas quantidades de S^1 e

S^2 perceptrons. Para essa estrutura, o resultado de processamento v_i^m de um dado neurônio na posição i da camada m é obtido como sendo o resultado da aplicação da função de ativação f^m à diferença entre o respectivo nível de ativação τ_i^m e o somatório dos produtos entre os q valores v^{m-1} da camada anterior com os também q pesos w_{iq}^m . As equações (3.2.2) e (3.4.1) obtêm o valor das saídas v_i^m em função dos demais parâmetros da estrutura neural.

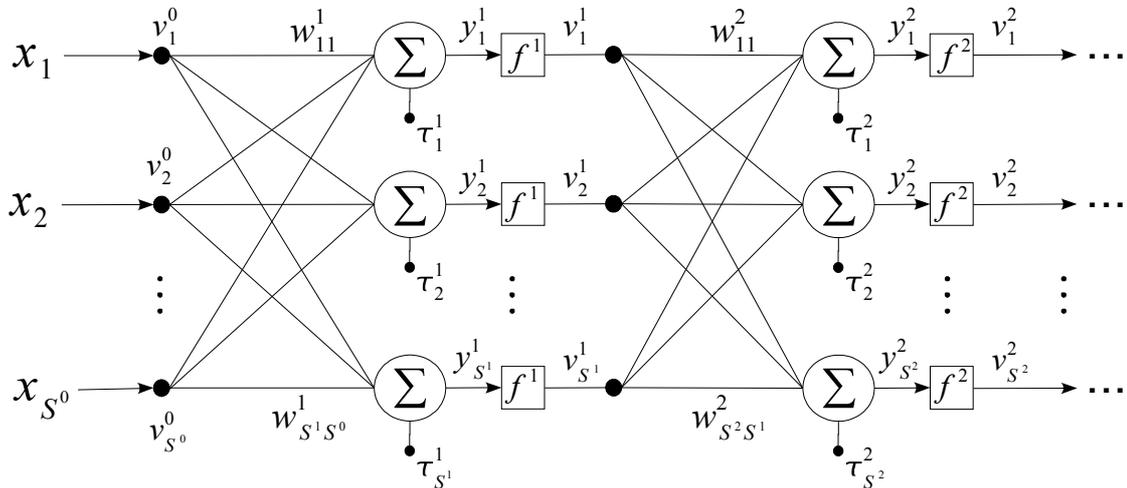


Figura 29 - Arquitetura básica de uma rede de perceptrons de múltiplas camadas contendo uma camada de entrada e duas de processamento.

$$y_i^m = \sum_{q=1}^{S^{m-1}} w_{iq}^m v_q^{m-1} + \tau_i^m \quad (3.2.1)$$

$$v_i^m = f^m(y_i^m) \quad (3.2.2)$$

A rede neural apresentada na Figura 29 contém a estrutura básica das redes neurais de múltiplas camadas. Essas redes podem ainda ser aprimoradas através da adição de mais

camadas intermediárias, que permitem com que suas capacidades de processamento sejam aumentadas e que seus mecanismos de particionamento do espaço vetorial sejam aprimorados até o ponto em que as próprias regiões particionadas por suas camadas iniciais de processamento sejam divididas em outras regiões, o que faz com que também sejam aprimoradas suas próprias capacidades de trabalhar com problemas com elevado grau de não-linearidade.

3.3 Perceptrons como Reconhedores de Padrões

A elevada capacidade das redes neurais em particionar e manipular regiões de espaços vetoriais fazem dessas redes artificiais exímias ferramentas de reconhecimento de padrões. Essas ferramentas, por si próprias, já apresentam tanto a capacidade de mapear vetores de características \hat{x} em vetores \hat{z} que identifiquem as classes as quais pertençam, o que é tido como a atividade de operação dos sistema de reconhecimento, quanto a capacidade de se aprender, com a qual são capazes de ajustar seus pesos e níveis de ativações internos a fim de que certos vetores-alvo \hat{i} possam se reproduzidos por suas saídas \hat{z} a partir de certos vetores de entrada \hat{x}_i , correspondentes, o que é tido como o processo de treinamento dos sistemas de reconhecimento de padrões.

Segundo *Looney* [64], algumas outras características das redes neurais que reforçam sua aptidão para trabalharem como sistemas de reconhecimento de padrões são sua característica não-linear e sua estabilidade, no sentido de que se um vetor de características

\hat{x}_K pertencente a uma classe K é mapeado para um identificador \hat{z}_K dessa classe, na forma $\hat{x}_K \rightarrow \hat{z}_K$, então um vetor \hat{x} próximo ao vetor \hat{x}_K é mapeado para um identificador \hat{z} que é próximo a \hat{z}_K , e mesmo um vetor \hat{x}' que esteja longe de \hat{x}_K pode ser mapeado para um identificador \hat{z}' também próximo a \hat{z}_K .

3.4 Treinamento

Para que uma rede neural possa ser utilizada como um sistema de reconhecimento de padrões, ela precisa passar por um processo de treinamento no qual seus pesos e níveis de ativação devem ser ajustados para que as regiões criadas no espaço vetorial possam ser manipuladas de forma a se gerar uma representação correta das classes de elementos existentes.

Os seguintes passos são atividades básicas de um treinamento de redes neurais utilizadas como sistemas de reconhecimento de padrões:

- (i) Deve-se obter um conjunto de vetores de características \hat{x}_K que contenha pelo menos um vetor pertencente a cada uma das K classes existentes;
- (ii) Para cada um dos vetores \hat{x}_K pertencente à classe K , deve ser relacionado um vetor-alvo \hat{i}_K que identifique a classe à qual pertence o vetor;

- (iii) Deve-se, assim, formar pares $(\dot{\mathbf{x}}_K, \dot{\mathbf{i}}_K)$ a representar um membro da classe K e um respectivo vetor que identifique essa mesma classe;
- (iv) Para cada um dos pares formados, a rede deve ajustar seus pesos w_{iq}^m e seus níveis de ativação τ_n^j até que suas saídas reproduzam corretamente cada um dos vetores $\dot{\mathbf{i}}_K$ quando em suas entradas houver o vetor de características $\dot{\mathbf{x}}_K$ correspondente.

Para que um processo de treinamento possa ocorrer, deve ser estabelecido um índice que seja capaz de mensurar o quão eficaz o processo se encontra e, a partir de então, finalizar o treinamento. Ainda segundo *Looney* [64], redes que passem por processos insuficientes de treinamento não são capazes de corretamente efetuar a associação entre vetores de características e os identificadores das classes às quais pertençam, enquanto que redes que tenham sido treinadas em excesso podem perder suas capacidades de generalizar e, assim, não mais reconhecer vetores de características que sejam de uma mesma classe que um vetor que é reconhecido corretamente mas que sejam ligeiramente diferentes do mesmo.

Uma forma comumente aplicada para se mensurar o quão treinada está uma rede neural artificial é obter o erro de treinamento total dado pelo somatório entre os quadrados das distâncias euclidianas entre cada um dos vetores de saída $\dot{\mathbf{Z}}_K$ e o esperado $\dot{\mathbf{i}}_K$ relacionados a cada um dos Q vetores de características $\dot{\mathbf{x}}_K$ presentes no conjunto de vetores de treinamento. A equação (3.4.1) apresenta essa formulação.

$$E = \sum_{q=1}^Q \|\mathbf{i}_{K(q)} - \mathbf{z}_{K(q)}\|^2 \quad (3.4.1)$$

Do ponto de vista do erro E , a atividade de treinamento de uma rede neural pode ser resumida a:

- (i) Estipular valores iniciais para seus pesos w_{iq}^m e seus níveis de ativação τ_n^j ;
- (ii) Obter o erro E_0 para os valores atuais de seus pesos e níveis de ativação;
- (iii) Atualizar seus pesos e níveis de ativação através de alguma estratégia que busque diminuir o erro E ; Obter novamente o erro;
- (iv) Se o novo erro E atingir algum critério esperado, então finalizar a atividade de treinamento, caso contrário voltar ao passo (iii).

3.4.1 Propagação de Erros

A perspectiva de treinar uma rede neural através de ajustes em seus parâmetros internos a fim de diminuir o erro de representação E possibilita com que vários algoritmos sejam aplicados. Esses algoritmos têm como função básica encontrar o mínimo local ou global da função-erro através da melhor forma possível, que quase sempre se traduz no uso do menor esforço computacional e na obtenção da melhor convergência possíveis. Dentre tais algoritmos, os mais comuns são os que efetuam o chamado treinamento supervisionado por

propagação retrógrada de erros [72], nos quais os erros de representação presentes nas camadas de saídas são propagados através de cada uma das camadas anteriores da rede neural.

Um dos algoritmos mais comuns utilizado pelo treinamento supervisionado através da propagação retrógrada de erros é o baseado no gradiente descendente, que também recebe a designação de método de mínimo passo descendente. Esse método basicamente busca encontrar, para a função erro em análise e a cada iteração, em qual direção a função erro decresce mais rapidamente. Para uma função não-linear qualquer $f(t)$, uma forma de encontrar um de seus mínimos locais através de seu gradiente descendente pode ser feita através de uma aproximação iterativa da forma [64]

$$t(r+1) = t(r) - \eta \frac{df(t)}{dt}(r), \quad (3.4.2)$$

em que o valor do parâmetro t na iteração $r+1$ é obtido em função de seu valor na iteração anterior r , em função da direção do mínimo passo descendente $-df(t)/dt$ e em função da taxa de convergência η , que pode ser adaptada a cada problema em que se aplique. Para o caso do treinamento da rede neural apresentada na Figura 29, ao se escrever o erro de treinamento E como sendo uma função das matrizes \mathbf{W} e \mathbf{T} formadas pelos respectivos pesos w_{iq}^m e níveis de ativação τ_i^m , a direção de seu mínimo passo descendente passa a ser dada pelo divergente $\nabla E(\mathbf{W}, \mathbf{T})$. Isso permite com que sejam estabelecidas equações que atualizem, para cada nova iteração $r+1$, os valores dos parâmetros internos da rede neural através dos valores desses mesmos parâmetros obtidos na iteração anterior r e de uma taxa de “aprendizado” η . As equações seguintes resumem essa abordagem [64].

$$E = E(\mathbf{W}, \mathbf{T}) \quad (3.4.3)$$

$$\nabla E(\mathbf{W}, \mathbf{T}) = \left(\frac{\partial E(\mathbf{W}, \mathbf{T})}{\partial \mathbf{W}}, \frac{\partial E(\mathbf{W}, \mathbf{T})}{\partial \mathbf{T}} \right) \quad (3.4.4)$$

$$w_{iq}^m(r+1) = w_{iq}^m(r) - \eta \left[\frac{\partial E(\mathbf{W}, \mathbf{T})}{\partial w_{iq}^m}(r) \right] \quad (3.4.5)$$

$$\tau_i^m(r+1) = \tau_i^m(r) - \eta \left[\frac{\partial E(\mathbf{W}, \mathbf{T})}{\partial \tau_i^m}(r) \right] \quad (3.4.6)$$

Um segundo método de treinamento também bastante presente na literatura é o chamado método de Newton [72], sendo que através desse método é possível calcular, para uma dada função, qual é a direção com a qual se pode minimizá-la da forma mais rápida. Diferentemente de usar o gradiente de $f(t)$ para obter a variação do parâmetro t a fim de se obter seu mínimo local através de iterações, o método de Newton obtém a variação Δt que minimize $f(t)$ através do gradiente $\nabla f(t)$ e da inversa da matriz Hessiana $\nabla^2 f(t)$, segundo a forma

$$\Delta t = -\nabla f(t) [\nabla^2 f(t)]^{-1}. \quad (3.4.7)$$

Comparado ao método do gradiente descendente, o método de Newton é computacionalmente mais complexo por envolver a inversão de matrizes, mas mesmo assim é mais rápido que o anterior [72].

Um outro método ainda mais eficiente de treinamento de redes neurais artificiais através da metodologia de propagação de erros é o utilizado por *Hagan* e *Menhaj* em seu

trabalho “*Training Feedforward Networks with the Marquardt Algorithm*” [73], que consistem em utilizar o algoritmo de *Levenberg-Marquardt* de minimização de funções. Esse algoritmo utiliza uma construção que permite com que se adapte tanto ao método de Newton quanto ao do gradiente descendente, o que permitiu otimizar significativamente o tempo requerido para se efetuar o treinamento de redes neurais baseadas em *perceptrons* de múltiplas camadas. Nesse sentido, se a função de erro de treinamento E for escrita como um somatório de funções quadráticas como

$$E(\dot{\mathbf{x}}) = \sum_{i=1}^N e_i^2(\dot{\mathbf{x}}), \quad (3.4.8)$$

então o método de Newton, ou método de *Gauss-Newton*, fornece [73]

$$\Delta \dot{\mathbf{x}} = J^T(\dot{\mathbf{x}}) \dot{e}(\dot{\mathbf{x}}) [J^T(\dot{\mathbf{x}}) J(\dot{\mathbf{x}})]^{-1}, \quad (3.4.9)$$

com $J(\dot{\mathbf{x}})$ representando o Jacobiano, segundo

$$J(\dot{\mathbf{x}}) = \begin{bmatrix} \frac{\partial e_1(\dot{\mathbf{x}})}{\partial x_1} & \dots & \frac{\partial e_1(\dot{\mathbf{x}})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_n(\dot{\mathbf{x}})}{\partial x_1} & \dots & \frac{\partial e_n(\dot{\mathbf{x}})}{\partial x_n} \end{bmatrix}. \quad (3.4.10)$$

Já a alteração de *Levenberg-Marquardt* fornece, para o mesmo caso, a seguinte formulação [73]

$$\Delta \dot{\mathbf{x}} = J^T(\dot{\mathbf{x}}) \dot{e}(\dot{\mathbf{x}}) [J^T(\dot{\mathbf{x}}) J(\dot{\mathbf{x}}) + \mu \mathbf{I}]^{-1}. \quad (3.4.11)$$

Com relação ao método de Newton, essa variação basicamente introduz o termo $\mu \mathbf{I}$ formado pelo parâmetro μ e pela matriz identidade \mathbf{I} . Ao ser alterado segundo o valor do erro $E(\dot{\mathbf{x}})$ obtido na última iteração, esse parâmetro tem como função fazer com que o método proposto ou se aproxime do método de mínimo passo descendente ou se aproxime do método de *Gauss-Newton*, o que permite com que o processo de aprendizagem das redes neurais seja aprimorado de forma significativa [73].

3.5 Arquitetura das Redes Neurais

Embora a arquitetura e a eficiência de uma rede neural artificial estejam diretamente ligados ao tipo de rede aplicada ao problema, uma forma geral de se pensar sobre a estrutura a ser utilizada para resolver algum problema de classificação ou de reconhecimento de padrões é a de inicialmente levantar o número de classes representadas pelos vetores de características. Ao se levantar essa quantidade, questionamentos quanto ao número de nós de entrada e de saída da rede também são abordados de uma forma natural.

O número de camadas de neurônios em uma rede de múltiplas camadas é uma questão sobre a qual a literatura afirma que uma quantidade de 2 camadas, sendo uma de

processamento e uma de saída, é suficiente para resolver grande parte dos problemas de mapeamento [65]. Assim, uma estrutura contendo duas camadas de neurônios a executar processamento e uma destinada a ser a camada de saída, seria, até como consequência da primeira afirmação, suficiente para resolver qualquer problema de mapeamento desde que os dados com os quais a rede venha a trabalhar sejam realmente apropriados para expressar as classes envolvidas [64].

Para um conjunto de K classes em um problema de reconhecimento de padrões, o que se costuma encontrar na literatura [64][65][72] é que uma boa estimativa par a quantidade J de nós na camada de saída é se fazer $J=K$ quando se trabalha com uma quantidades baixas de classes, e $J=\log_2(K)$ para quantidades elevadas.

Já quanto ao número de neurônios presentes na camada intermediária de processamento, o que é comumente encontrado na literatura é que, ao se levar em conta que um número M de neurônios na camada intermediária é capaz de formar entre $M+1$ e 2^M regiões convexas no espaço vetorial no qual residem os elementos de características, então uma boa abordagem seria iniciar uma rede neural com $M=K$ elementos em sua camada de processamento e assim esperar que a camada de saída efetue as combinações necessárias para se representar corretamente cada uma das K classes existentes. No caso dessa estrutura inicial não treinar corretamente em algumas poucas tentativas, então a estrutura deve ser modificada através da adição ou da exclusão de neurônios da camada de processamento [64].

CAPÍTULO 4

Transformações de Fourier

As transformações de Fourier compreendem um conjunto de técnicas desenvolvidas por volta de 1804 e apresentadas em 1807 pelo francês *Jean Baptiste Joseph Fourier* em seu trabalho “*Théorie de la propagation de la chaleur dans les solides*” [74], com o qual se fez a colocação inicial de que qualquer função periódica pode ser representada através de uma série de funções seno harmonicamente relacionadas. Após cerca de 150 anos de sua publicação, a teoria envolta à série proposta por Fourier foi devidamente compreendida e compilada junto a toda a teoria das integrais de Fourier. Desde então, o sucesso de seu emprego junto à análise de sinais e demais problemas de cunho matemático e físico faz com que essas proposições sejam consideradas o pilar do processamento de sinais, com aplicações e conceitos que se disseminam em muitas áreas acadêmicas e que são a base de muitas tecnologias que permeiam todo o cotidiano contemporâneo.

4.1 A Transformada de Fourier de Tempo Contínuo

O objetivo de se aplicar uma transformação em um objeto matemático está basicamente ligado à possibilidade de revelar atributos desse objeto que não são facilmente percebidos ao se analisá-lo de uma certa óptica ou, como é comum de se utilizar ao se fazer menção à teoria de espaços lineares, quando esse mesmo objeto está representado em um espaço linear inadequado.

Qualquer ente matemático como funções e afins pode ser representado por combinações lineares entre funções base que formem o próprio espaço linear onde seja representada a função inicial. Uma função polinomial da forma $f(x) = 1 + 2x + 0.75x^2 + x^3$ pode, por exemplo, ser representada por uma combinação linear entre as funções presentes no conjunto $B = \{x^0, x^1, x^2, x^3\}$. Essa representação é definida pela notação

$$f(x) = c_0x^0 + c_1x^1 + c_2x^2 + c_3x^3, \quad (4.1.1)$$

na qual os valores $c_{i=0..3}$ são constantes. Para esse caso, em que combinações lineares entre os componentes do conjunto B podem ser utilizadas para se representar todas as possíveis funções polinomiais com grau máximo 3, diz-se que o conjunto B é uma base do espaço linear dessa classe de funções polinomiais. Assim, pode-se dizer que cada uma das funções linearmente independentes presentes no conjunto B contribui para a formação da função $f(x)$ através de uma constante c_i ou, em outras palavras, que a projeção da função $f(x)$ sobre um polinômio x^i equivale a c_i .

Essa possibilidade de representar uma função $f(t)$ pertencente a um espaço linear $L(\mathbb{R})$ através de sua projeção sobre as funções base que formam esse mesmo espaço naturalmente induz a se pensar em uma transformada como sendo um operador $\mathcal{L}: L(\mathbb{R}) \rightarrow L'(\mathbb{R})$, que seja capaz de gerar a representação da função $f(t)$ no espaço $L'(\mathbb{R})$ através da obtenção da projeção dessa mesma função sobre cada uma das funções-base desse novo subespaço. Dessa forma, uma transformação pode ser equacionada como o produto interno entre a função sobre a qual aplicar a transformação e a função que caracterize essa mesma transformação, assim como o que ocorre na equação

$$\hat{f}(\alpha) = \mathcal{L}\{f(\mu)\} = \int_{-\infty}^{\infty} f(\mu) \overline{g(\mu)} d\mu, \quad (4.1.2)$$

onde o termo $\overline{g(\mu)}$ representa o complexo conjugado de funções que caracterizem a transformação e que, no caso das proposições de *Joseph Fourier*, assumem a forma complexa $e^{j\omega t}$.

A transformada de Fourier de tempo contínuo para funções não-periódicas é uma transformação aplicável à classe de sinais pertencentes ao espaço L^2 de funções de energia finita, que são integráveis e obedecem à equação

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \text{infinito}. \quad (4.1.3)$$

Essa transformação é dada pela integral [48]

$$\hat{f}(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \quad (4.1.4)$$

A idéia básica dessa transformação é mensurar “o quanto” de oscilações na frequência ω há na função $f(t)$, e sua operação basicamente se resume a obter a projeção de $f(t)$ sobre cada uma das funções periódicas $e^{j\omega t}$, o que significa dizer que, se em algum intervalo de tempo, a função $f(t)$ oscilar com uma frequência π , haverá nesse intervalo um deslocamento de fase constante entre essa função e a função de análise $e^{j\pi t}$, o que fará com que a transformada $\hat{f}(\omega)$ receba uma contribuição não nula para a frequência π [47]. No entanto, vale ressaltar que pelo fato das funções de análise terem suporte global, ou seja, serem definidas e não decaírem para todo o intervalo $[-\infty, \infty]$, a transformada de Fourier não fornece a informação sobre qual é o intervalo de tempo no qual a frequência π se apresenta em $f(t)$.

Quando a transformada $\hat{f}(\omega)$ for também integrável, então sua transformação inversa é dada por

$$f(t) = \mathcal{F}^{-1}\{\hat{f}(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{j\omega t} d\omega. \quad (4.1.5)$$

Algumas das principais propriedades da transformada de Fourier e de sua inversa são [47]:

- i. a linearidade, no sentido de que se $f(t) = a f_1(t) + b f_2(t)$, então

$$\hat{f}(\omega) = a \hat{f}_1(\omega) + b \hat{f}_2(\omega);$$

- ii. o deslocamento no tempo, que faz com que a transformada de

$$f_0(t) = f(t - t_0) \text{ resulte em } \hat{f}_0(\omega) = e^{-j\omega t_0} \hat{f}(\omega);$$

- iii. o escalonamento, com $\hat{f}_a(\omega) = \frac{1}{|a|} \hat{f}\left(\frac{\omega}{a}\right)$ para $f_a(t) = f(at)$;
- iv. o deslocamento na frequência, que faz com que se obtenha $f_0(t) = f(t)e^{j\omega_0 t}$ ao se fazer $\hat{f}_0(\omega) = \hat{f}(\omega - \omega_0)$;
- v. o escalonamento na frequência, com o qual se obtém $f_a(t) = \frac{1}{a} f\left(\frac{t}{a}\right)$ para $\hat{f}_0(\omega) = \hat{f}(a\omega)$;
- vi. o *momento*, definido como $M_n = (-i)^{-n} \left. \frac{d^n \hat{f}(\omega)}{d\omega^n} \right|_{\omega=0}$;
- vii. e o *teorema de Parseval*, que diz que $\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |\hat{f}(\omega)|^2 d\omega$, ou, em outras palavras, que a decomposição de um sinal através da transformada de Fourier é capaz de conservar a energia do sinal em sua representação no domínio da frequência.

4.2 A Série de Fourier de Tempo Contínuo

A série de Fourier de tempo contínuo é uma variação da transformada de Fourier de tempo contínuo, sendo destinada a trabalhar com sinais periódicos e também de tempo contínuo.

Para um sinal periódico de frequência fundamental $1/T$, sua série de Fourier é dada por

$$\hat{f}[k] = \mathcal{F}_s\{f(t)\} = \frac{1}{T} \int_{\langle T \rangle} f(t) e^{-jk\omega_0 t} dt \quad (4.2.1)$$

e a respectiva transformada inversa pode ser obtida através da equação

$$f(t) = \mathcal{F}_s^{-1}\{\hat{f}[k]\} = \sum_{k=-\infty}^{\infty} \hat{f}[k] e^{jk\omega_0 t}, \quad (4.2.2)$$

nas quais se tem $\omega_0 = 2\pi/T$.

Um exemplo de aplicação da série de Fourier pode ser o caso onde se queira obter a representação no domínio da frequência para o sinal

$$x(t) = 3 \cos\left(\frac{\pi}{2}t + \frac{\pi}{4}\right).$$

A série de Fourier para esse caso resulta em

$$\hat{x}[k] = \begin{cases} (3/2)e^{-i\pi/4}, & k = -1 \\ (3/2)e^{i\pi/4}, & k = 1 \\ 0, & \text{caso contrário} \end{cases},$$

cujos espectros de amplitude e fase são apresentadas na Figura 30.

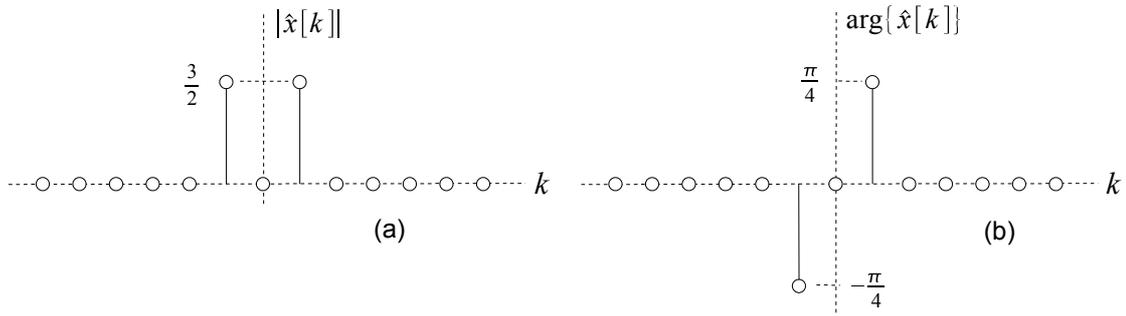


Figura 30 - Espectros de amplitude (a) e fase (b) obtidos a partir da série de Fourier da função $x(t)$.

4.3 A Transformada de Fourier de Tempo Discreto

Assim como a série de Fourier de tempo discreto, a transformada de Fourier de tempo discreto representa uma transformação de Fourier para sinais discretos, sendo que para um sinal discreto e não periódico $f[n]$ de comprimento (número de elementos) N , sua transformada e a respectiva inversa são dadas por

$$\hat{f}(e^{j\Omega}) = \mathcal{F}_d\{f[n]\} = \sum_{n=-\infty}^{\infty} f[n] e^{-j\Omega n} \quad (4.3.1)$$

$$f[n] = \mathcal{F}_d^{-1}\{\hat{f}(e^{j\Omega})\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{f}(e^{j\Omega}) e^{j\Omega n} d\Omega, \quad (4.3.2)$$

com o período de $\hat{f}(e^{j\Omega})$ sendo de 2π .

4.4 A Série de Fourier de Tempo Discreto

Possivelmente como uma das transformações de Fourier mais aplicadas em atividades práticas na área de processamento digital de sinais, a série de Fourier de tempo discreto busca representar um sinal periódico e discreto através de sua representação no domínio da frequência.

Para um sinal periódico e discreto $f[n]$ com um período fundamental N , a sua representação através da série discreta é dada por

$$\hat{f}[k] = \mathcal{F}_{ds}\{f[n]\} = \sum_{n=\langle N \rangle} f[n] e^{-jk\Omega_0 n}, \quad (4.4.1)$$

enquanto sua inversa tem a forma

$$f[n] = \mathcal{F}_{ds}^{-1}\{\hat{f}[k]\} = \frac{1}{N} \sum_{k=\langle N \rangle} \hat{f}[k] e^{jk\Omega_0 n}, \quad (4.4.2)$$

com $\Omega_0 = 2\pi/N$.

4.5 Aplicação

A família de transformações de Fourier tem aplicações que percorrem desde as áreas acadêmicas até as áreas práticas aliadas ao processamento digital de sinais. Nas áreas acadêmicas, a grande parte de desenvolvimentos relacionados costuma fazer uso das versões das transformações de Fourier destinadas ao trabalho junto a sinais contínuos, que são tanto a série quanto a transformada de Fourier para tempo contínuo. Já as aplicações práticas das transformações de Fourier não tem outra escolha senão a de fazerem uso das versões discretas dessas transformações, em especial da série de Fourier de tempo discreto.

A aplicação da série de Fourier de tempo discreto e de sua inversa é considerada como sendo uma das ferramentas que dão suporte a muitas das técnicas e mesmo teorias que envolvem a área de processamento digital de sinais. O conceito básico de aplicação dessas duas transformações está atrelado à possibilidade de se manipular, monitorar ou mesmo representar sinais a partir de seus conteúdos espectrais de amplitude e fase, que são obtidos através da série discreta de Fourier. Um exemplo simples de processamento digital de sinais baseado nessa série pode ser a implementação de um filtro para uma dada frequência, cujo princípio de funcionamento seria o de aplicar a série de Fourier ao sinal a ser filtrado e com as informações dos espectros de fase e amplitude, excluir a frequência a ser filtrada para, através da aplicação da transformada inversa da série discreta, reconstruir e disponibilizar o sinal devidamente filtrado. Mesmo a execução de uma convolução entre dois sinais discretos, um processo notoriamente custoso em termos computacionais, pode ser obtida com maior rapidez e menor esforço computacional quando os espectros dos sinais envolvidos são extraídos, multiplicados no domínio da frequência e depois são utilizados para se reconstruir o resultado da convolução através da transformada inversa da série discreta [75].

Através da série de Fourier para sinais discretos, sinais devidamente amostrados podem ser representados por seus conteúdos harmônicos de amplitude e fase, que são utilizados para se formar os chamados espectros de amplitude e fase. A Figura 31 contém uma representação discreta para o sinal de tempo contínuo $x(t) = \text{sen}(2\pi t - \pi/18) + 0,75\text{sen}(2(2\pi t) + \pi/9)$ e seus respectivos espectros harmônicos. Esses espectros contêm os parâmetros de amplitude e fase de cada uma das formas de onda complexas $e^{j\omega t}$ que são necessárias para se reproduzir o sinal original.

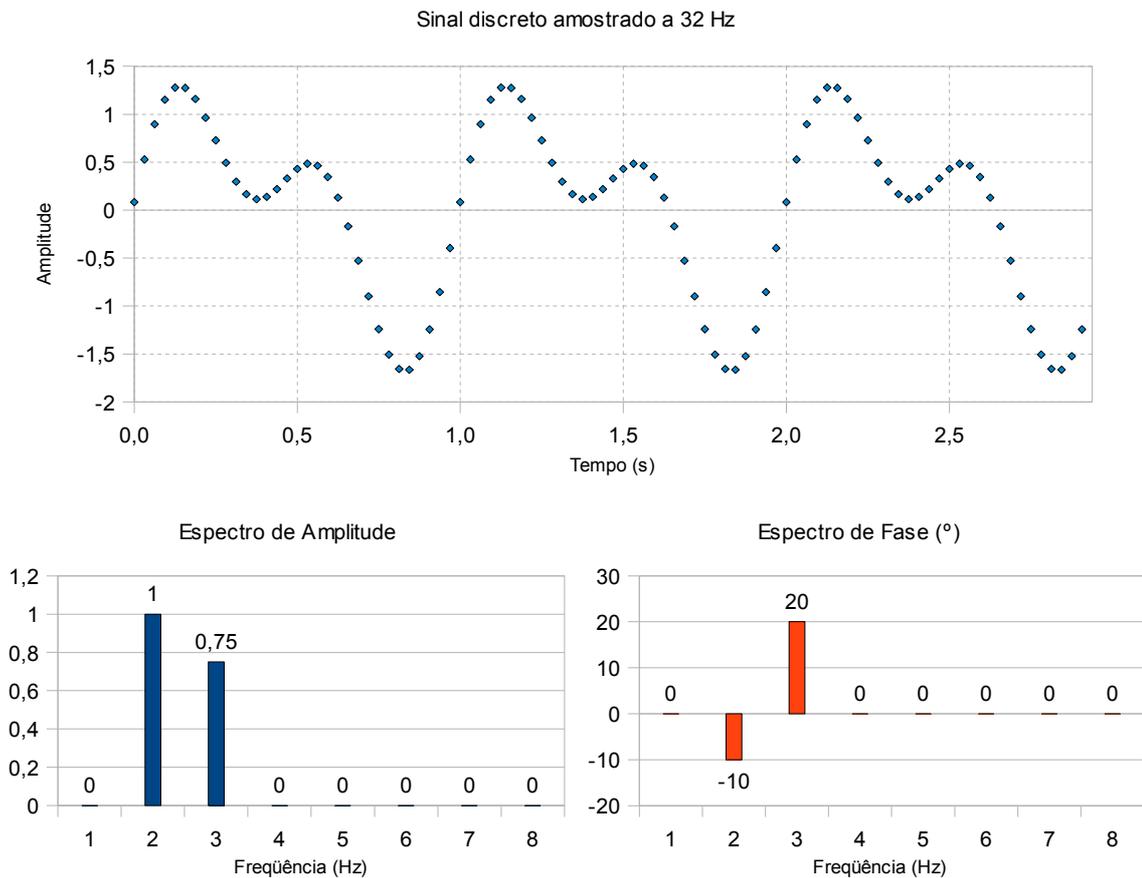


Figura 31 - Sinal discreto com frequência fundamental de 1 Hz e seus respectivos espectros harmônicos de amplitude e fase.

A representação de sinais discretos através de seus espectros harmônicos de amplitude e fase é um recurso muito utilizado ao se trabalhar com técnicas de processamento digital de sinais, mas, no entanto, esse tipo de representação começou a ser adotada principalmente a partir da aparição inicial de um algoritmo denominado de transformada rápida de Fourier [44], que mesmo apresentando limitações devido a problemas inerentes às transformações de Fourier para sinais de tempo discreto, como o espalhamento (vazamento) de frequências e as limitações ao se representar sinais contendo descontinuidades, viabilizou a aplicação da série discreta de Fourier, que até então não era adotada na prática devido a seu elevado custo computacional.

4.5.1 A Transformada Rápida de Fourier

Em atividades práticas, as transformações de Fourier para tempo discreto são implementados a partir de alguma modalidade ou variação do algoritmo conhecido como transformada rápida de Fourier (FFT). Acredita-se que esse algoritmo tenha sido originalmente iniciado por *Gauss*, mas sua primeira aparição formal como uma técnica completa e aplicável é atribuída a *J. W. Cooley* e *J. W. Tukey* devido ao trabalho “*An algorithm for the machine calculation of complex Fourier series*”, de 1965. Esse conjunto de algoritmos opera de forma a recursivamente dividir a série de Fourier em séries menores para que possam ser utilizadas as propriedades de simetria e de periodicidade da senóide complexa $e^{j\omega m}$, que são as funções de análise utilizadas pelas transformações de Fourier.

Para se analisar a estrutura da transformada rápida de Fourier baseada no algoritmo de raiz 2 com decimação no tempo [44], é uma das formas mais simples dessa classe de algoritmos, é tomado um sinal discreto $x[n]$ que contenha o caso especial de $N=2^l$ elementos. Levando-se conta que $\Omega_0=2\pi/N$, a transformada da série discreta de Fourier apresentada na equação (4.4.1) pode ser reescrita, para esse sinal, através da forma

$$\hat{x}[k]=\mathcal{F}_{ds}\{x[n]\}=\sum_{n=\langle N\rangle}x[n]e^{-j2\pi kn/N}, \quad (4.5.1)$$

que, com

$$W_N=e^{\frac{-j2\pi}{N}}, \quad (4.5.2)$$

ainda pode tomar a forma

$$\hat{x}[k]=\sum_{n=0}^{N-1}x[n]W_N^{kn}, \quad \text{para } 0 \leq k < N. \quad (4.5.3)$$

O primeiro passo desse algoritmo é dividir o sinal a ser transformado em duas seqüências, sendo que uma deve conter os elementos de índices pares $x_p[n]=x[2n]$ e a outra deve conter os elementos de índices ímpares $x_i[n]=x[2n+1]$. As transformadas dessas seqüências resultam na equações

$$\hat{x}_p[k]=\sum_{n=0}^{N/2-1}x[2n]W_N^{2kn}, \quad \text{para } 0 \leq k < N/2. \quad (4.5.4)$$

$$\hat{x}_i[k]=\sum_{n=0}^{N/2-1}x[2n+1]W_N^{2kn}, \quad \text{para } 0 \leq k < N/2. \quad (4.5.5)$$

Assim, ao se reescrever o sinal original $x[n]$ em função de suas porções $x_p[n]$ e $x_l[n]$ e ao submetê-lo à transformação presente na equação (4.5.2), o que se obtém é, segundo a referência [44], a dupla de equações

$$\hat{x}[k] = \hat{x}_p[k] + W_N^k \hat{x}_l[k] \quad \text{e} \quad (4.5.6)$$

$$\hat{x}[k + N/2] = \hat{x}_p[k] - W_N^k \hat{x}_l[k], \quad (4.5.7)$$

ambas para $0 \leq k < N/2$.

As equações (4.5.6) e (4.5.7) dizem que a série de Fourier de tempo discreto do sinal $x[n]$ contendo $N = 2^l$ elementos por ser obtida através de duas séries discretas de sinais $x_p[n]$ e $x_l[n]$ que contém $N/2$ elementos cada. Esse pensamento pode ser extrapolado até o ponto em que se obtenha um sinal $x^*[n]$ de apenas 2 elementos e cuja série discreta de Fourier possa ser facilmente obtida e, então, utilizada para se obter a transformada de série de 4 elementos que a originou e, assim, repetir o procedimento de forma recursiva até que se obtenha a série discreta do sinal original $x[n]$.

Como exemplo de aplicação da transformada rápida com raiz 2 e decimação no tempo, para um sinal discreto de 8 elementos $x[n] = [x_0, \dots, x_7]$, o primeiro passo a ser adotado seria dividir sucessivamente o sinal original em sinais de comprimentos menores e que contenham ou os índices pares ou os índices ímpares do sinal anterior. O resultado desse procedimento é, para o caso do sinal $x[n]$ de exemplo, apresentado na Figura 32 e resulta em um sinal da forma $x'[n] = [x_0, x_4, x_2, x_6, x_1, x_5, x_3, x_7]$.

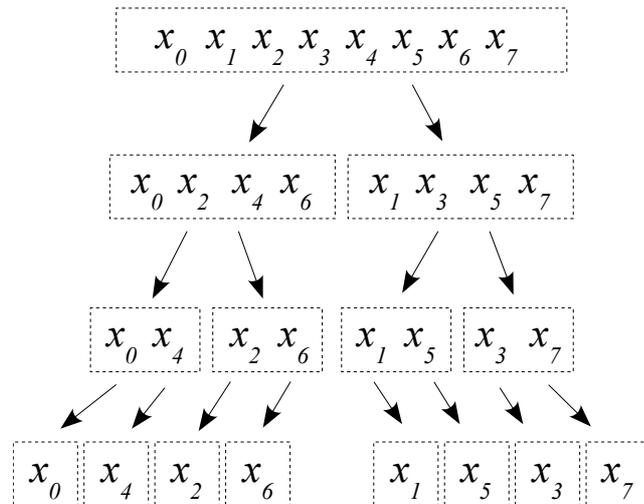


Figura 32 - Decomposição de um sinal de 8 elementos em 8 sinais de 1 elemento cada.

Como passo intermediário de aplicação, são encontrados os espectros de frequências de cada um dos 8 sinais no domínio no tempo (1 ponto cada), mas como a transformação de um sinal com um único elemento corresponde a ele mesmo, ao se obter os 8 sinais com 1 elemento cada já se obtém os 8 espectros desejados. O passo final de aplicação consiste em combinar os N espectros de 1 frequência cada em um único espectro de N frequências. Assim, no primeiro estágio, 8 espectros de 1 frequência geram 4 espectros de 2 frequências, no segundo estágio, 4 espectros com duas frequências cada geram 2 espectros com 4 frequências cada, sendo que ao último estágio cabe a geração de 1 espectro de 8 frequências a partir de 2 espectros de 4 frequências. Para esses últimos passos é utilizada uma estrutura chamada de *estrutura em borboleta*, que é apresentada na Figura 33 e na qual dois números complexos dão origem a outros dois números complexos. Essa estrutura é o elemento básico de operação da transformada rápida de Fourier para o caso do algoritmo de raiz 2 com decimação no tempo.

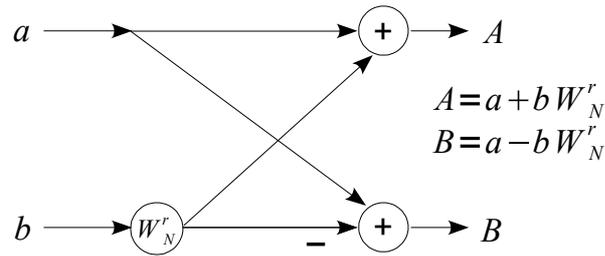


Figura 33 - Estrutura em borboleta utilizada pela transformada rápida de Fourier.

De uma forma geral, a aplicação da transformação rápida de Fourier através do algoritmo de raiz 2 com decimação no tempo consiste em decompor sucessivamente o sinal discreto a ser transformado e a se aplicar também sucessivas transformações para se obter a série discreta de Fourier para cada um dos estágios de decomposição realizados, o que se resume a dizer que séries discretas de Fourier são inicialmente obtidas e depois passam a ser combinadas em resultados de 2, 4, 8, ..., 2^m pontos, até que se obtenha o resultado final de aplicação da transformada. A Figura 34 apresentam as atividades envolvidas ao se aplicar a transformada de Fourier exemplificada.

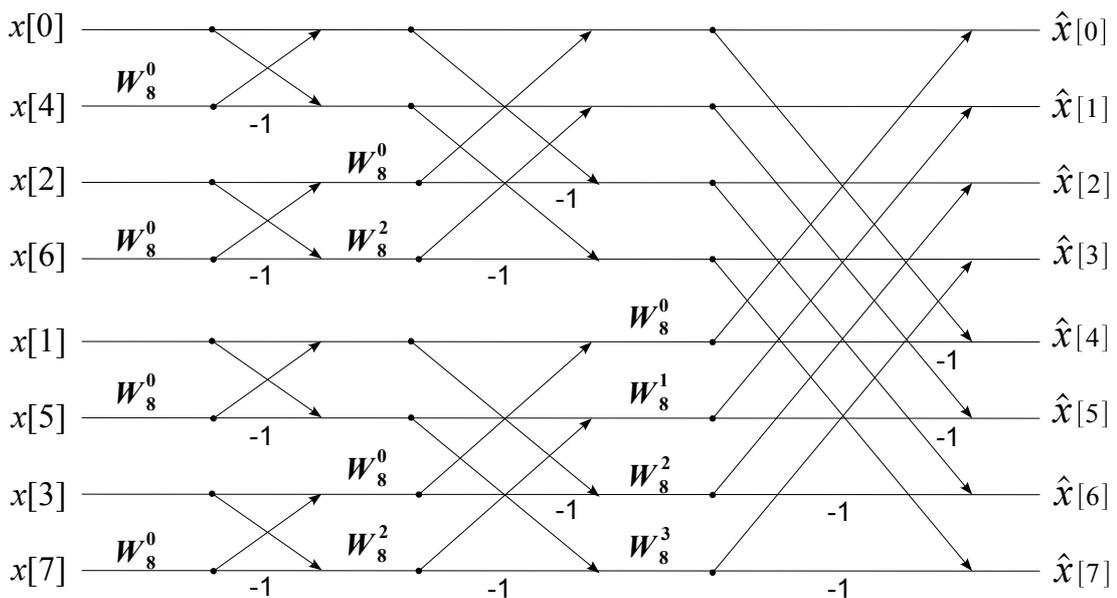


Figura 34 - Aplicação de uma transformada rápida de Fourier a um sinal contendo 8 elementos.

Um dos principais resultados relacionados à aplicação da transformada rápida de Fourier é o fato de que a essa classe de algoritmos é creditada a capacidade de aprimorar substancialmente a obtenção da série de Fourier para sinais de tempo discreto. Em termos de esforços computacionais, essa classe de técnicas pode obter a série de Fourier para um sinal com N elementos através de algoritmos de ordem $O(N \log_2(N))$, em oposição à ordem $O(N^2)$ da aplicação direta dessa mesma transformação. Como um exemplo, a transformação de um sinal contendo 1024 elementos através da aplicação direta da série discreta requer uma quantidade de operações que é cerca de 120 vezes a quantidade requerida pela transformação desse mesmo sinal através de certas transformadas rápidas de Fourier.

4.6 Limitações da Representação por Séries de Fourier de Tempo Discreto

A representação prática de sinais através da série de Fourier de tempo discreto apresenta limitações que são relacionadas tanto à formulação da transformada de Fourier quanto às limitações corriqueiras a qualquer tentativa de representar sinais contínuos em ambientes computacionais. Dentre estas limitações, as mais importantes são o reconhecimento de falsas frequências, o vazamento de frequências e os problemas inerentes à representação de sinais contendo descontinuidades.

4.6.1 Reconhecimento de Falsas Frequências

O problema de reconhecimento de falsas frequências não é questão própria da série de Fourier de tempo discreto, mas sim um problema relacionado a processos impróprios de amostragem de sinais contínuos, que geram os sinais digitais com os quais a série discreta trabalha e que assim costuma se propagar nos resultados de aplicação dessa série. Na literatura, esse fenômeno recebe a designação de *aliasing* devido ao falso entendimento que ocasiona acerca das reais frequências presentes em um sinal.

De uma forma geral, a ocorrência desse fenômeno de falso reconhecimento de frequências faz com que as frequências de um sinal que sejam superiores à metade da frequência com a qual esse sinal tenha sido amostrado (digitalizado), sejam erroneamente projetadas nos espectros de amplitude e fase obtidos através da série de Fourier desse sinal. Como um exemplo, se um sinal amostrado a uma frequência de 10 Hz contiver uma componente com frequência de 6 Hz, então essa componente será erroneamente projetada para dentro do espectros de amplitude e fase desse sinal como sendo uma componente de frequência de 4 Hz. Uma explicação formal que se vale da própria transformada de Fourier e justifica a ocorrência desse fenômeno é apresentada na referência [48]. Essa mesma referência também apresenta uma formulação simples, o teorema de *Nyquist*, com a qual evitar esse tipo de problema. Segundo esse teorema, uma frequência mínima f_s com a qual um sinal deve ser amostrado precisa ser superior ao dobro da maior frequência f_c presente nesse sinal, segundo

$$f_s \geq 2f_c. \quad (4.6.1)$$

O desenvolvimento da justificativa para esse fenômeno feito pela referência [48] também permite desenvolver a relação existente entre a frequência aparente f_a que é “entendida” pelo processo de amostragem e as frequências mínima de amostragem f_s e real f presente em um sinal. Essa relação é apresentada na Figura 35.

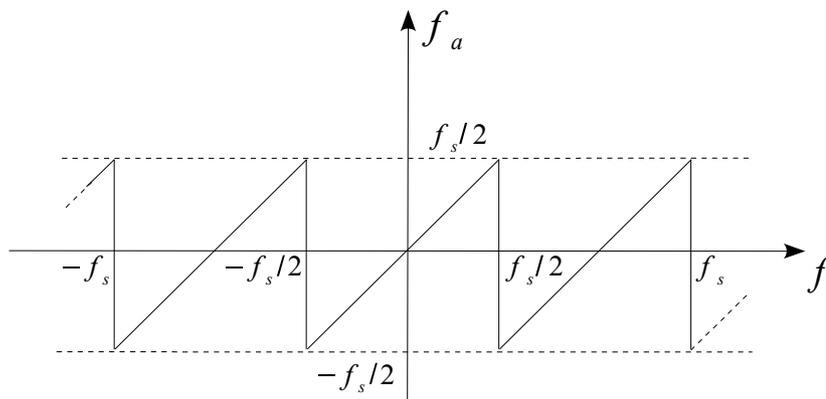


Figura 35 - Relação entre as frequências aparente, de amostragem e real em um sinal amostrado.

4.6.2 Vazamento de Frequências

A aplicação da série de Fourier a um sinal discreto $x[n]$ contendo N elementos gera um conjunto $\hat{x}[k]$ de também N elementos. Essa quantidade de N elementos é associada à frequência de amostragem f_s para produzir a chamada resolução em frequência Δf , que é dada pela formulação

$$\Delta f = \frac{f_s}{N}. \quad (4.6.2)$$

A resolução em frequência da série de Fourier de tempo discreto estipula quais são as frequências que podem ser reconhecidas pela aplicação dessa série discreta, sendo que cada uma das componentes presentes no sinal $\hat{x}[k]$ tem uma frequência que é dada por um múltiplo inteiro de Δf que esteja entre o intervalo $[-N/2, N/2[$. Assim, se fosse utilizada uma frequência de 10 Hz para se amostrar um sinal contendo 10 elementos, a resolução de frequência obtida seria de 1 Hz, o que faria com que as frequências passíveis de serem reconhecidas pela série de Fourier estivessem entre -5 Hz e 4 Hz, incluindo 0 Hz.

O problema de espalhamento de frequências ocorre, portanto, quando a resolução em frequência relacionada a uma aplicação da série discreta de Fourier não for um múltiplo inteiro de pelo menos uma das frequências presentes no sinal discreto do qual é tomada a série. Nessa ocasião, as frequências não múltiplas, chamadas de inter-harmônicas, são espalhadas nas raias espectrais mais próximas de cada uma das frequências inter-harmônicas existentes, o que ocasiona um resultado de transformação que não condiz com a realidade do sinal.

Um exemplo dessa consequência pode ser o caso em que a uma forma de onda que contenha inicialmente frequências múltiplas inteiras da resolução em frequência seja adicionada uma frequência inter-harmônica, que acaba por ser espalhada nas raias de frequência próximas ao valor real de sua frequência. A Figura 36 apresenta os espectros harmônicos de amplitude de uma forma de onda amostrada de forma a se obter uma resolução em frequência de 15 Hz e que, no primeiro espectro apresentado (a), é formada por duas componentes de amplitude unitárias de frequências de 60 Hz e 180 Hz. Já para o segundo caso (b), à onda inicial foi acrescida uma componente de amplitude também unitária mas com uma frequência inter-harmônica de 110 Hz. Essa frequência, ao não ser uma múltipla

inteira de 15 Hz, se espalha nas faixas espectrais próximas a 110 Hz e assim produz informações no espectro que erroneamente apontam para a existência de componentes no sinal que não estão realmente presentes, além de também influenciar na detecção das amplitudes das frequências de 60 Hz e 180 Hz que realmente compõem o sinal transformado.

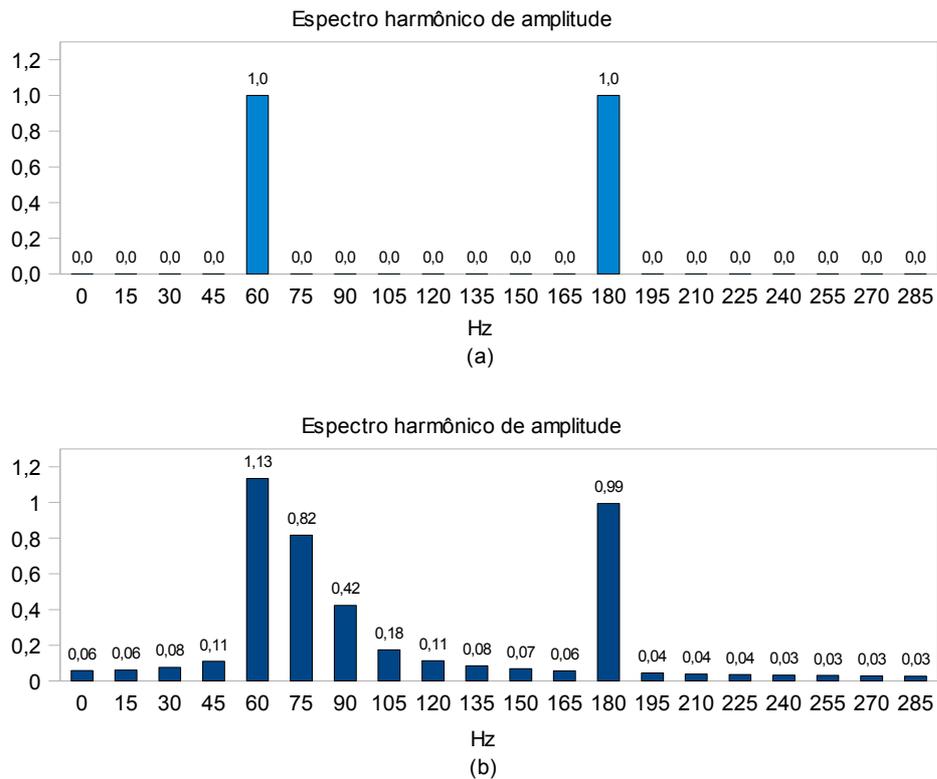


Figura 36 - Espectro de amplitude de uma onda contendo as frequências de 60 Hz e 180 Hz (a) e o resultado (b) do acréscimo de uma componente frequência inter-harmônica de 110 Hz à onda inicial.

O fato da presença de componentes inter-harmônicos provocarem a ocorrência de espalhamento espectral que agregam informações inverídicas aos resultados de aplicação da série de Fourier de tempo discreto é uma questão importante para muitas áreas que utilizam as transformações de Fourier. Muitos ramos dos sistemas elétricos de potência fazem uso de decomposições harmônica para obter informações de sinais de corrente e tensão discretizados,

sendo que grande parte dos equipamentos elétricos e eletrônicos são tratados como fontes de distorções puramente harmônicas, cujas frequências são múltiplas inteiras da frequência da tensão da rede de alimentação, desprezando ou mesmo sendo inviável considerar a presença de inter-harmônicas. Exemplos desses casos são os sistemas de reconhecimento de cargas elétricas não-lineares que são baseados na análise das formas de onda de corrente através de decomposição através da série discreta de Fourier. Essas propostas [28]-[34] buscam relacionar cada classe de cargas elétricas a uma certa distribuição harmônica das amplitudes de suas formas de onda de corrente e, assim, estabelecem que certos valores de amplitudes para algumas das frequências múltiplas de 60 Hz caracterizam essa ou aquela classe. Desta forma, a presença de componentes inter-harmônicas nos sinais de corrente que analisam podem, entre outras coisas, direcionar a reconhecimentos errôneos ou mesmo a um incapacidade de reconhecimento.

4.6.3 Presença de Descontinuidades

A presença de descontinuidades em sinais não é fato incomum, ainda mais quando esses sinais estão relacionados às formas de onda de corrente de cargas elétricas não-lineares [45]. No entanto, a aplicação desses sinais às transformações discretas de Fourier revelam uma importante deficiência existente ao se abordar essas técnicas de representação.

Segundo [46], a soma parcial presente nas séries de Fourier para tempo discreto, que aqui são apresentada pela equações (4.4.1) e (4.4.2), são erros de aproximação para com o sinal periódico original. Para um sinal $x_r[t]$ reconstruído a partir dos coeficientes $\hat{x}[k]$

obtidos através da série de Fourier do sinal $x[n]$ de $2N$ elementos, a reconstrução é feita, ainda segundo [46], pela equação (4.6.3), em que o termo multiplicativo a $x(\tau)$ é um termo oscilatório nos pontos de continuidades de $x[n]$ que não desaparece mesmo ao se fazer $N \rightarrow \infty$.

$$x_r(t) = \frac{1}{T} \int_{-T/2}^{T/2} x(\tau) \frac{\text{sen}(N+1/2)(t-\tau)\omega_0}{\text{sen}\frac{1}{2}(t-\tau)\omega_0} d\tau \quad (4.6.3)$$

Um exemplo prático dessa limitação pode ser observado quando a forma de onda presente na curva (a) da Figura 37, amostrada com uma resolução em frequência de 60 Hz e que é uma forma de onda típica de uma lâmpada incandescente com *dimer* [45], é submetida à série de Fourier de tempo discreto e, então, novamente reconstruída através de suas 16, 32 e 64 primeiras harmônicas e depois através de suas 128 harmônicas totais obtidas da série discreta. Os resultados são apresentados nas curvas (b)-(e) da mesma Figura 37 e permitem evidenciar que a presença de descontinuidades provocam oscilações que podem comprometer a representação obtida para o sinal.

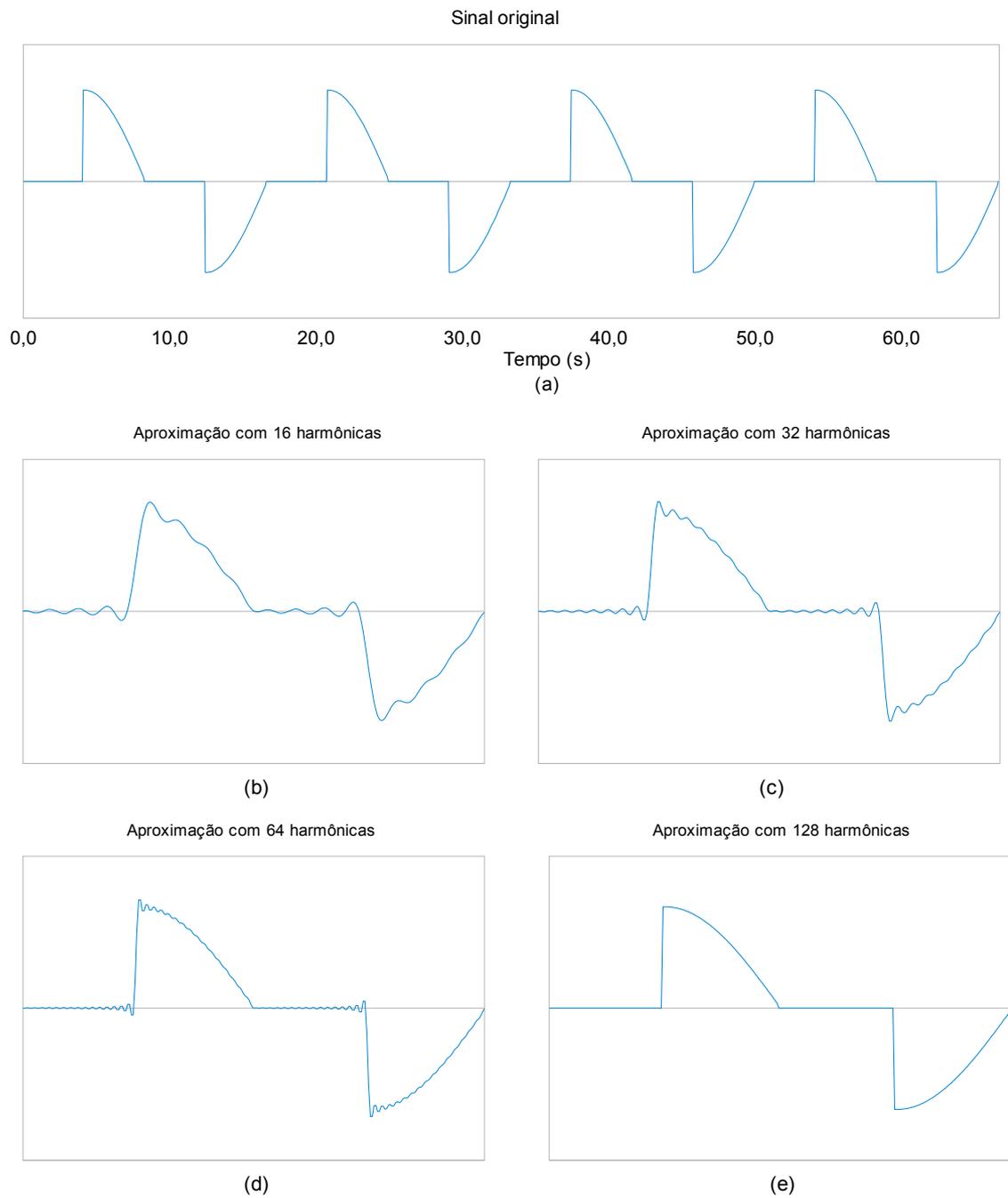


Figura 37 - Sinal (a) e suas aproximações (b)-(e) obtidas através da série de Fourier de tempo discreto.

A presença de oscilações nas reconstruções apresentadas na Figura 37 recebe a designação de *fenômeno de Gibbs* [46]. Uma consequência direta desse fenômeno é a necessidade de se utilizar uma quantidade elevada de coeficientes da série de Fourier para

representar corretamente um sinal. Essa circunstância pode agregar limitações a certas áreas na qual a representação pela série discreta seja utilizada, como é o caso de áreas ligadas aos sistemas elétricos de potência, onde os estudos sobre harmônicas corriqueiramente utilizam representações contendo até a 31ª harmônica. Ao se comparar as curvas (a) e (c), pode-se perceber que a representação de forma de onda com descontinuidades até o 31º índice harmônico é notoriamente incompleta e não contém toda a energia presente no sinal discreto da qual é derivada. Em sistemas de reconhecimento de cargas elétricas não-lineares baseados na decomposição harmônica das formas de onda de corrente, em especial naqueles que também utilizem técnicas de reconhecimento de padrões como redes neurais artificiais, existe um comprometimento em se utilizar o menor número possível de características das formas de onda (índices harmônicos) de modo a facilitar os mecanismos de reconhecimento e assim viabilizar a operação desses sistemas. Esse fato leva a casos [43][44] em que são utilizadas pequenas quantidades de índices harmônicos para se representar as cargas elétricas e, assim, à utilização de representações incompletas que podem, entre outras coisas, comprometer a eficiência desses mesmos sistemas.

CAPÍTULO 5

Transformações Wavelet

A utilização de transformações matemáticas sempre foi um dos pontos básicos ao desenvolvimento e ao emprego de técnicas de processamento de sinais mais avançadas. Essa estreita relação ocorre principalmente pelo fato de que certas transformações matemáticas permitem com que sinais pertencentes a certas classes possam ser analisados através de uma óptica diferente daquela com a qual naturalmente se apresentam. Dessa forma, o simples fato de se transformar um sinal e assim o mapear para um subespaço adequado e diferente do inicial equivalha, por exemplo, a se passar a observar um objeto através de um novo ângulo de visão com o qual possam ser extraídas características que até então eram desconhecidas e que facilitem a análise que se queira empregar ao sinal ou objeto transformado. Um bom exemplo para esse tipo relação estreita é o caso da utilização das transformações de Fourier [44], pois se por um lado tais transformações possibilitaram com que sinais ou funções pudessem ser analisados segundo a óptica de suas componentes de frequências e assim

permitiram com que inúmeras técnicas de processamento fossem desenvolvidas, por outro lado essas mesmas transformações contêm limitações [46][47] que as caracterizam como sendo um ângulo de visão inadequado para se trabalhar com a análise de algumas classes de sinais, em especial as que contemplem descontinuidades ou transitórios, que são melhor representados através da proposta oferecida pelas chamadas *transformações wavelet* [46].

As transformações *wavelet* representam uma classe de transformações matemáticas inicialmente desenvolvidas em meados da década que compreende os anos de 1980 e 1990, como um fruto de trabalhos de processamento de sinais sísmicos que requereram resultados mais expressivos que os disponibilizados pelas transformações de Fourier [47]. Essas transformações formam um conjunto de operadores matemáticos que propõem efetuar a representação de sinais e funções através de suas projeções em funções de suporte local, em oposição ao suporte global das funções $e^{j\omega t}$ que são avaliadas em todo o intervalo $]-\infty, \infty[$ pelas transformações de Fourier. Esse suporte local, aliado a desenvolvimentos posteriores como à análise em multirresolução e às famílias de funções de decomposição desenvolvidas especificamente para as transformações *wavelets* [48], permite com que sejam geradas representações de sinais através de vários níveis de resolução (aproximação) e, dentre outras coisas, com que sejam realizadas a extração e a compactação de características de sinais de uma forma mais eficiente que a executada pelas técnicas da proposta de Fourier quando da presença de descontinuidades ou transitórios nos sinais em análise. Assim, essas e outras características fazem com que as transformações *wavelet* se façam presentes em várias áreas de aplicação como a de processamento de sinais e de imagens, a de diagnósticos médicos, a de problemas de valores limite, a de processamento de sinais geofísicos, a de análises estatísticas e, em especial, a de reconhecimento de padrões.

5.1 A Transformada Wavelet Contínua

A base das transformações *wavelet* são funções designadas pelo termo *wavelet* e que devem respeitar as características de serem pertencentes ao subespaço L^2 , conterem média zero, serem normalizadas, centralizadas em $t=0$ e oscilatórias. As equações (5.1.1) a (5.1.3) formulam essas três primeiras características.

$$\int_{-\infty}^{\infty} \|\psi(t)\|^2 dt < \infty \quad (5.1.1)$$

$$\hat{\psi}(\omega=0) = \int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (5.1.2)$$

$$\|\psi\| = 1 \quad (5.1.3)$$

Ao serem deslocadas no tempo através de uma quantidade b e ao serem dilatadas ou comprimidas através de um parâmetro de escala a , as funções *wavelet* geram famílias de funções $\psi_{a,b}$ que representam pequenas ondas, e daí o termo *wavelets*, que possuem as mesmas características básicas das funções de origem, inclusive a norma unitária. A equação (5.1.4) define essas famílias de pequenas ondas.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \text{ com } a \neq 0 \quad (5.1.4)$$

Assim, para uma função $f(t)$ também pertencente ao subespaço L^2 de funções quadraticamente integráveis e de energia finita, sua transformada *wavelet* contínua é obtida pela formulação

$$f(a, b) = W\{f(t)\} = \int_{-\infty}^{\infty} f(t) \overline{\psi_{a,b}(t)} dt, \quad (5.1.5)$$

na qual o termo $\overline{\psi_{a,b}}$ representa o conjugado complexo de $\psi_{a,b}$.

Por essa formulação, a transformada *wavelet* mensura a variação de $f(t)$ cujo tamanho seja proporcional à escala a e que esteja nas vizinhanças de b , o que equivale a dizer que essa transformada obtém a correlação cruzada entre a função $f(t)$ e a função de análise $\psi_{a,b}$ e assim mensura o grau de similaridade entre elas para uma dada posição (b, a) no plano tempo-escala.

A Figura 38 apresenta variações na escala e no deslocamento para o caso de uma função *wavelet* da família *Daubechies* [48], exemplificando a idéia básica por trás da manipulação feita com a função $\psi_{a,b}$ a fim de que sejam obtidas as correlações entre suas variações e a função $f(t)$ para cada um dos pontos (b, a) possíveis.

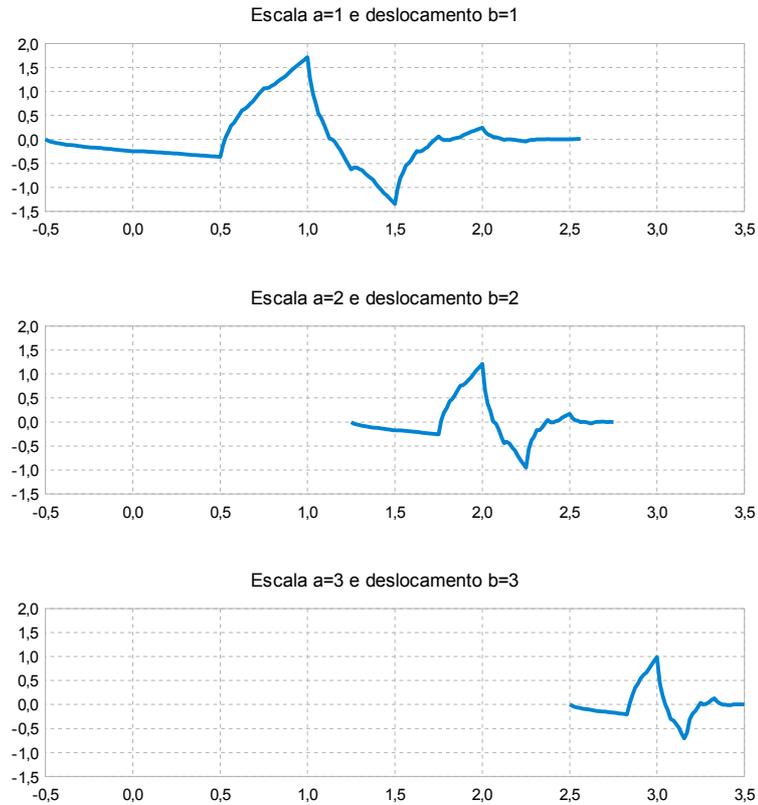


Figura 38 - Funções *wavelets* da família *Daubechies 2* com diferentes deslocamentos e escalas.

As variações que podem ser feitas com os valores dos parâmetros de escala e deslocamento da função de decomposição $\psi_{a,b}$ possibilitam com que a equação (5.1.5) efetue uma análise de $f(t)$ através de todo o plano tempo-escala. Essa mesma análise é também feita de forma indireta para o plano tempo-freqüência, pois o plano tempo-escala pode ser aproximado ao plano tempo-freqüência através de uma conversão adequada entre a escala a e uma freqüência f representativa. Uma forma de observar essa possibilidade é verificar que variações no parâmetro de escala dilatam ou contraem a função $\psi_{a,b}$, aumentando ou diminuindo seu suporte temporal, e como a operação de sua transformada mensura o de grau de sua similaridade com relação à porção de $f(t)$ centrada em b , suas

expansões permitem identificar frequências menores enquanto suas contrações permitem identificar frequências maiores. Dessa forma, pode ser estabelecida uma relação de proporção inversa entre a escala de dilatação de $\psi_{a,b}$ e a frequência que é indiretamente analisada pela equação (5.1.5).

$$f \approx \frac{1}{a} \quad (5.1.6)$$

Uma forma de visualizar a operação da transformada *wavelet* no plano tempo-frequência é apresentada pela referência [46], que desenvolve a janela tempo-frequência em função dos parâmetros de escala a , de deslocamento b e das variações no suporte de temporal Δ_ψ e na seção positiva do espectro de frequências Δ_{ψ^+} , chegando-se à conclusão de que o produto da janela de tempo-frequência é constante e pode ser obtido pela equação (5.1.7). Isso permite representar a operação da transformada *wavelet* como sendo uma varredura no plano tempo-frequência executada por uma janela de área constante na qual um aumento de escala diminui o suporte no plano de frequências e um aumento no plano de frequências diminui o suporte temporal.

$$2 a \Delta_\psi \frac{2}{a} \Delta_{\psi^+} = 4 \Delta_\psi \Delta_{\psi^+} = \text{constante} \quad (5.1.7)$$

A Figura 39 exemplifica três janelas de varredura no plano tempo-frequência. Cada uma delas está relacionada a um ponto (b, a) específico no plano tempo-escala e permite que seja exemplificado a relação existente entre o suporte temporal, a escala e o suporte no domínio da frequência que são aplicados ao se obter a transformação *wavelet* contínua de um

signal. A frequência ω^* e o tempo t^* presentes nessa mesma figura são os centros do suporte $\Delta_{\hat{\psi}^+}$ no plano de frequências e do suporte temporal Δ_{ψ} , respectivamente.

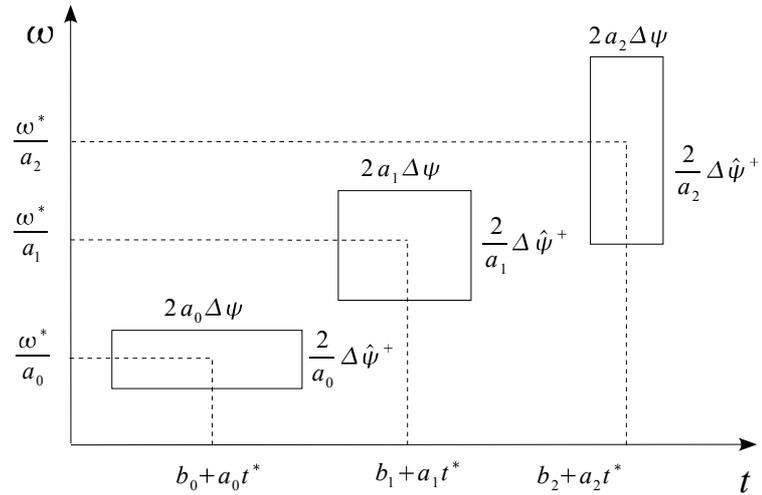


Figura 39 - Janelas de tempo-frequência utilizadas pela transformada *wavelet*.
 Fonte: [46].

Já quanto à transformada *wavelet* inversa, as formulações seguintes apresentam-na.

$$f(t) = W^{-1}\{f(a, b)\} = \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} f(a, b) \psi_{a,b}(t) db da \tag{5.1.8}$$

$$C_{\psi} = \int_0^{\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty \tag{5.1.9}$$

5.2 A Transformada Wavelet Discreta

A transformada *wavelet* discreta é obtida ao se discretizar os parâmetros de escala a e de deslocamento no tempo b presentes na equação (5.1.5). Valores de discretização que são costumeiramente utilizados para esses parâmetros são $a=2^{-s}$ e $b=k2^{-s}$, devido a facilidades que a utilização de potências de 2 agregam ao cálculo da transformada discreta. Levando-se em conta essa discretização e o fato de que as funções contínuas $f(t)$ e $\psi(t)$ devem ser representadas por suas versões discretas, a transformada *wavelet* contínua pode ser reescrita, segundo [46], como

$$f[2^{-s}, k2^{-s}] = W_d\{f[n]\} = \sum_n f[n]\psi[n2^s - k]. \quad (5.2.1)$$

Diferentemente da transformada *wavelet* contínua que pode gerar informações redundantes [46], a escolha dos valores $a=2^{-s}$ e $b=k2^{-s}$ como discretização da escala e do deslocamento, respectivamente, permite com que a transformada discreta mantenha a mínima quantidade possível de informações do sinal original que permitam com que seja efetuada sua reconstrução sem que haja agregação de erros [46]. Em aplicações práticas, a transformada *wavelet* discreta é utilizada para se implementar algoritmos de decomposição de sinais segundo a proposta de sua contra-parte contínua, dentre os quais alguns chegam até mesmo a aprimorar e ampliar a proposta inicial das transformações *wavelet*, como o que ocorre com a teoria da análise em multirresolução [48].

5.3 Análise em Multirresolução

Desenvolvida por *Mallat e Meyer* [48], a análise em multirresolução busca representar sinais através de suas decomposições em vários níveis graduais de aproximação, o que permite, entre outras coisas, que sinais complexos sejam decompostos em sinais mais simples que contenham características específicas do sinal do qual se originam e, assim, possam ser analisados individualmente. Essa técnica de análise é inspirada na aplicação da transformada *wavelet* discreta e na teoria de espaços lineares e suas respectivas bases de formação, com as quais podem ser criadas representações de um sinal em subespaços que são capazes de gradativamente aumentar ou diminuir o nível de detalhes que se queira empregar ao se analisar um sinal original.

Em termos de uma notação matemática formal, um sinal discreto $f[n]$ pode ser representado pela decomposição em multirresolução através do conjunto de sinais

$$\left[a_{M+N}, \{ d_n \}_{M < n \leq M+N} \right], \quad (5.3.1)$$

em que o sinal a_M representa o sinal original em sua maior resolução e sinal a_{M+N} , em sua aproximação de menor resolução $M+N$ desejada, enquanto que os sinais $\{ d_n \}$ contêm os detalhes de cada uma das N aproximações. Esses sinais podem ser utilizados para se reconstruir uma aproximação $a_{j \in [M, M+N]}$ de maior resolução através da aproximação e dos detalhes de menor resolução a_{j+1} e d_{j+1} , segundo

$$a_j \Leftrightarrow a_{j+1} + d_{j+1}. \quad (5.3.2)$$

5.3.1 Espaços de Multirresolução

A idéia por trás dos espaços de multirresolução é a de poder utilizar um conjunto $\{A_s\}$ de subespaços de L^2 em que cada um destes possa abrigar funções e sinais cujas representações nesses mesmos subespaços contenham níveis de detalhes que sejam majorados ou suprimidos à medida que suas representações sejam projetadas em subespaços adjacentes aos atuais de sua decomposição.

Para que um conjunto $\{A_s\}$ de subespaços possa gerar uma representação em multirresolução, as condições (5.3.3) a (5.3.7) [48] devem ser preenchidas.

$$\forall (j, k) \in \mathbb{Z}^2, x(t) \in A_j \Leftrightarrow x(t - 2^j k) \in A_j \quad (5.3.3)$$

$$\forall j \in \mathbb{Z}, A_{j+1} \subset A_j \quad (5.3.4)$$

$$\forall j \in \mathbb{Z}, x(t) \in A_j \Leftrightarrow x(t/2) \in A_{j+1} \quad (5.3.5)$$

$$A_{-\infty} \dots \cap A_{j-1} \cap A_j \cap A_{j+1} \cap \dots A_{\infty} = \{0\} \quad (5.3.6)$$

$$A_{-\infty} \dots \cup A_{j-1} \cup A_j \cup A_{j+1} \cup \dots A_{\infty} = L^2 \quad (5.3.7)$$

A condição (5.3.3) significa dizer que A_j é invariável a qualquer translação proporcional a 2^j , enquanto a condição (5.3.4) permite salientar que um representação através de um subespaço A_j contém todas as informações necessárias para se obter uma

aproximação dessa mesma representação em um subespaço A_{j+1} de menor resolução. A condição (5.3.5) ressalta que a dilatação por 2 de uma função pertencente ao subespaço A_j garante que a função resultante represente uma aproximação dessa mesma função inicial em um subespaço de menor resolução A_{j+1} . Já as condições (5.3.6) e (5.3.7) descrevem, respectivamente, que à medida que uma função é sucessivamente projetada em subespaços de menor resolução, seus detalhes são suprimidos até que se chegue a ponto de serem completamente descartados, mas já à medida que são utilizadas resoluções maiores para se representar um sinal, a aproximação converge para o próprio sinal.

Além das condições citadas, para que o conjunto $\{A_s\}$ possa gerar representações em multirresolução, deve haver uma função base $\phi(t)$, denominada de função escala, que seja capaz de gerar cada um dos subespaços A_j . Uma formulação para essa função base é, segundo [46], a equação

$$\phi(t) = \sum_k g_a[k] \phi(at - k), \quad (5.3.8)$$

onde o parâmetro a representa a dilatação e a conseqüente mudança da resolução necessária para formar, junto aos coeficientes $g_a[k]$, as demais funções presentes em cada subespaço que contenha uma resolução de interesse.

Cada um dos subespaços A_j gerados por $\phi(t)$ representam níveis de resoluções que podem gerar subespaços A_{j+1} de menores resoluções. Isso condiz com a condição (5.3.4) e deixa claro que existe um outro subespaço W_{j+1} que seja o complemento de A_{j+1} em A_j e que ao ser combinado com o próprio subespaço A_{j+1} , permite formar o conjunto de maior

resolução A_j . Dessa forma, as relações possíveis entre esses conjuntos complementares são resumidas pelas equações (5.3.9) e (5.3.10) e exibidas pela Figura 40.

$$A_j \cap W_j = 0, \quad j \in \mathbb{Z} \quad (5.3.9)$$

$$A_j = A_{j+1} + W_{j+1} \quad (5.3.10)$$

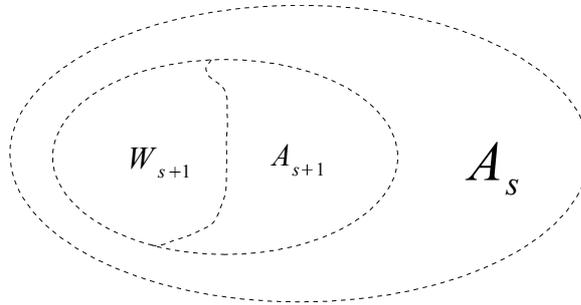


Figura 40 - Relação entre os subespaços de maior e menor resoluções

Os subespaços W_j são chamados de subespaços de funções *wavelet* e são gerados por funções base $\psi(t) \in L^2$. Assim, uma função $x_s(t)$ pertencente ao subespaço A_s e uma função $y_s(t)$ pertencente ao subespaço W_s podem ser escritas como combinações das respectivas funções $\phi(t)$ e $\psi(t)$. Ainda, se o subespaço W_j for ortogonal ao subespaço A_j , ou seja, $A_j \perp W_j$, diz-se que a representação em multirresolução efetuada é ortogonal.

$$x_s(t) \in A_s \rightarrow x_s(t) = \sum_k a_{k,s} \phi(2^s t - k) \quad (5.3.11)$$

$$y_s(t) \in W_s \rightarrow y_s(t) = \sum_k w_{k,s} \psi(2^s t - k) \quad (5.3.12)$$

Para uma representação efetuada a partir de um subespaço A_M , a relação do subespaço de funções escala e funções *wavelet* pode ser expandida segundo as equações (5.3.13) a (5.3.15).

$$A_M = A_{M+1} + W_{M+1}, \quad (5.3.13)$$

$$A_M = A_{M+2} + W_{M+2} + W_{M+1} \text{ e} \quad (5.3.14)$$

$$A_M = A_{M+N} + \sum_{n=1}^N W_{M+n}. \quad (5.3.15)$$

A equação (5.3.15) permite concluir que para uma representação em multirresolução iniciada através da aproximação A_M e que utilize N resoluções diferentes, os componentes dessa representação são obtidos pela projeção da função em análise em um subespaço de resolução A_{M+N} , chamado de subespaço de aproximação na resolução $M+N$, e pela projeção da mesma função em análise através de N subespaços a partir do subespaço W_M , que são chamados de subespaços de detalhes ou subespaços *wavelets*. As equações (5.3.13) a (5.3.15) também permitem arquitetar esse mesmo processo de decomposições como sendo uma atividade recursiva em que cada uma das aproximações da função em análise são projetadas sobre subespaços de resolução imediatamente menor que a do subespaço no qual foi construída a aproximação a ser projetada. Para essa visão, somente as porções de aproximação são recursivamente projetadas, enquanto as porções de detalhes são mantidas inalteradas e então são agrupadas para se chegar na proposição presente na equação (5.3.15).

A Figura 41 apresenta uma estrutura decomposição utilizada para se efetuar uma representação em multirresolução formada por subespaços $\{A_s\}$ e $\{W_s\}$ e iniciada a partir da resolução conferida ao subespaço de aproximação A_M .

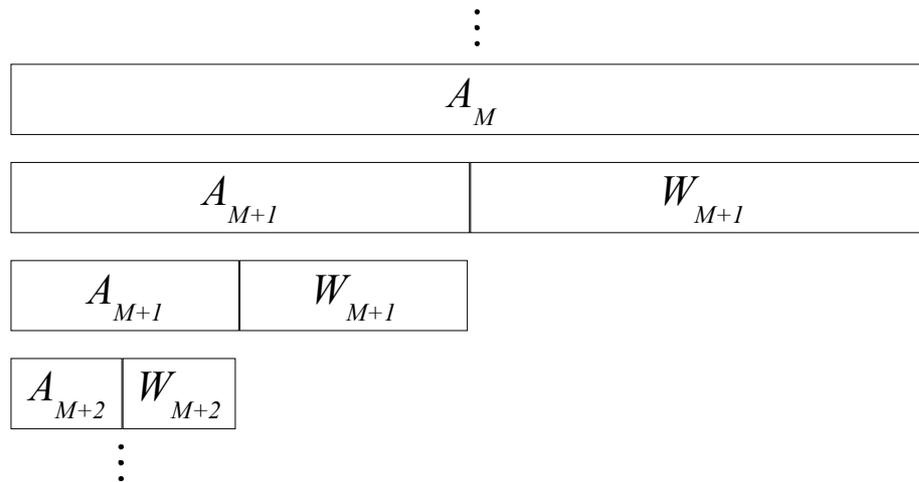


Figura 41 - Análise em multirresolução efetuada através dos subespaços A_s e W_s .

Na Figura 42 é exemplificada a aplicação de uma decomposição em multirresolução até o terceiro nível de resolução. Nessa figura, é possível notar que à medida que o sinal original é decomposto através dos subespaços A_1 , A_2 e A_3 , sua representação se aproxima de uma forma de onda mais simples, que para este caso é uma onda senoidal. Também à medida que essas decomposições em subespaços de menor resolução são efetuadas, são gerados sinais de detalhes pertencentes aos respectivos subespaços W_1 , W_2 e W_3 . Uma outra forma de compreender o processo de decomposição em multirresolução, ainda utilizando Figura 42, é partir do sinais de menor resolução e então retroceder até o sinal original. Por exemplo, a onda presente na aproximação A_3 é praticamente senoidal, mas se a ela for adicionado o sinal presente no subespaço W_3 , o que se obtém é uma forma de onda similar à aproximação em A_2 , que se trata de um sinal senoidal acrescido de uma

descontinuidade. Assim, à aproximação A_2 é acrescentado o sinal de detalhes em W_2 e, então, é obtida a onda presente no subespaço A_1 , que se trata de uma forma senoidal com ruídos e uma descontinuidade. Ao fim, se à onda presente em A_1 forem acrescentadas as oscilações presentes no subespaço W_1 , então o sinal original, contendo ruídos e uma descontinuidade em seu segundo pico mínimo, é recriado.

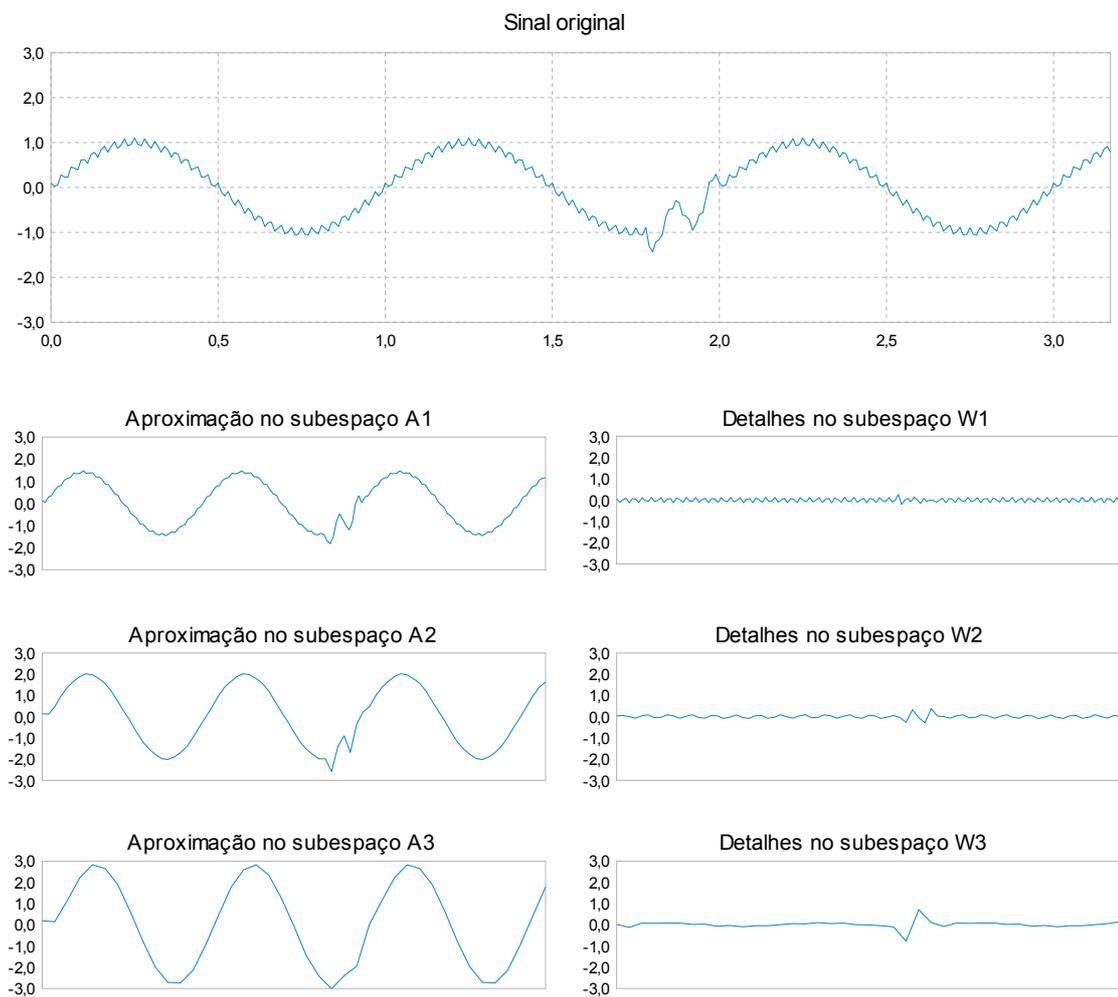


Figura 42 - Exemplo de uma representação em multirresolução de 3 níveis.

5.4 Aplicação

A aplicação prática das transformações *wavelets* é feita através de sua representante discreta, a transformada *wavelet* discreta definida pela equação (5.2.1). Por conseguinte, os resultados obtidos através da operação dessa transformada discreta estão sujeitos a limitações tanto de ordem de discretização, uma vez que os ambientes computacionais agregam limitações a representações de sinais, quanto herdadas da própria proposta e formulação da transformada *wavelet* contínua. Dentre essas limitações, as mais pertinentes são a quantidade de momentos nulos de uma função *wavelet* e a questão da influência que um deslocamento no tempo em um sinal provoca nos resultados de sua decomposição através de certas transformadas *wavelet*.

5.4.1 Momentos Nulos

O número de momentos nulos de uma função *wavelet* $\psi(t)$ equivale ao número de zeros de sua transformada no domínio da frequência, $\hat{\psi}(\omega)$, na frequência $\omega=0$ [48]. Isso significa que se $\psi(t)$ tem p momentos nulos, então a igualdade presente na equação (5.3.12) tem que ser satisfeita.

$$\int_{-\infty}^{+\infty} t^k \psi(t) dt = 0 \quad \text{para } 0 \leq k < p. \quad (5.4.1)$$

A quantidade de momentos nulos de uma função *wavelet* é importante, pois, segundo [76], para uma função analisada pela transformada *wavelet* contínua, suas seções que puderem ser modeladas por polinômios de ordem menor que a ordem de momentos nulos da função *wavelet* não contribuirão para o resultado da transformação, enquanto que a transformada se concentrará nas regiões nas quais hajam mudanças de uma seção polinomial para outras, ou seja, nas discontinuidades da função analisada. Nesse sentido, quanto maior a quantidade de momentos nulos em uma função *wavelet* de análise, melhor essa função será para representar sinais complexos.

A presença de momentos nulos em um família de funções *wavelets*, ao lado de algumas outras características, consiste em um dos principais parâmetros de escolha da família a ser utilizada em função da complexidade dos sinais a serem representados pelas transformações.

5.4.2 Variação ao Deslocamento

Em suas construções mais simples, as ortogonais, as transformações *wavelet* discretas são notoriamente variantes a deslocamentos no sinal que analisam. Essa propriedade pode ser observada ao se fazer

$$f_m = f(t - t_m) \quad (5.4.2)$$

para a equação (5.2.1) de definição da transformada *wavelet* discreta, que resulta, segundo [46], em

$$f_m[2^{-s}, k2^{-s}] = W_d\{f_m[n]\} = f[(k - m2^s)2^{-s}, 2^{-s}]. \quad (5.4.3)$$

A equação (5.4.3) permite concluir que ao se deslocar um sinal no tempo, a consequência em sua representação através da transformada *wavelet* discreta não equivale ao simples deslocamento pelo qual o sinal passou.

5.4.3 Algoritmo de Banco de Filtros

Bancos de filtros são estruturas comumente alocadas em atividades de processamento de sinais para realizar atividades de filtragem e mesmo a decomposição de sinais. Esses bancos são estruturas hierárquicas similares à estrutura utilizada para se obter a representação de sinais em espaços de multirresolução e, por isso, são empregues para tal fim.

Para se adaptar uma análise em multirresolução à estrutura utilizada por bancos de filtros, o primeiro passo é utilizar as relações de decomposição presentes nas equações (5.3.13) a (5.3.15) para se obter as condições

$$x_{s+1}(t) \in A_{s+1} \rightarrow x_{s+1}(t) = \sum_k a_{k,s+1} \phi(t), \quad (5.4.4)$$

$$x_s(t) \in A_s \rightarrow x_s(t) = \sum_k a_{k,s} \phi(t) \text{ e} \quad (5.4.5)$$

$$y_s(t) \in W_s \rightarrow y_s(t) = \sum_k \omega_{k,s} \psi(t). \quad (5.4.6)$$

Ao se efetuar certas manipulações [46] com a função escala presente na equação (5.3.2), as equações (108) a (109), unidas à equação (5.3.8), fornecem

$$a_{k,s+1} = \sum_l h_0[2k-l] a_{l,s}, \quad (5.4.7)$$

$$w_{k,s+1} = \sum_l h_1[2k-l] w_{l,s}. \quad (5.4.8)$$

As equações (5.4.7) e (5.4.8) permitem, então, com que os coeficientes $\{a_{k,s}\}$ e $\{w_{k,s}\}$, que formam a representação de um sinal nos subespaços A_s e W_s , sejam obtidos através dos coeficientes $\{a_{k,s-1}\}$ e $\{w_{k,s-1}\}$ da representação desse mesmo sinal nos subespaços de maior resolução A_{s-1} e W_{s-1} . Os coeficientes $\{h_0[k]\}$ são relacionados à função escala da família de funções *wavelets* utilizada pela decomposição e representam os coeficientes de um filtro passa-baixas. Já os coeficientes $\{h_1[k]\}$ estão relacionados a um filtro passa-altas e são derivados da função *wavelet* que também pertence à família de *wavelets* utilizada na decomposição. Esses coeficientes formam filtros chamadas de filtros de decomposição.

Ainda em termos das equações (5.4.7) e (5.4.8), essas equações representam os processos de convolução e de decimação do respectivo resultado. A convolução é realizada entre os coeficientes dos filtros e os coeficientes da representação de maior resolução, enquanto as decimações, que para esse caso significa descartar um a cada dois elementos, são

aplicadas aos resultados das convoluções efetuadas. A Figura 43 apresenta a estrutura gráfica das atividades relacionadas a essa decomposição.

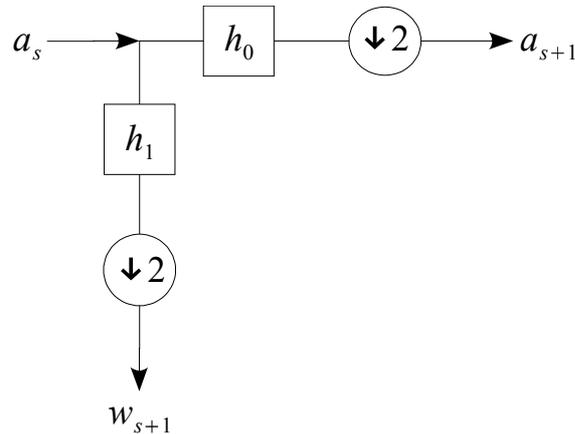


Figura 43 - Estrutura básica da decomposição realizada pela análise em multirresolução efetuada através de bancos de filtros.

Levando-se em conta a estrutura presente na Figura 43, na qual é representada a decomposição em um simples nível de resolução, decomposições que requeiram N níveis de resolução deverão utilizar N estruturas similares à presente na Figura 43 e dispostas em forma hierárquica, sendo que cada aproximação resultante de um estrutura básica de decomposição seja encaminhada à próxima estrutura básica de decomposição existente.

O processo de reconstrução de sinais decompostos através de uma representação em multirresolução é similar ao da decomposição, bastando com que decimações sejam substituídas por interpolações, com as quais amostras com valores 0 são incluídas a cada dois elementos de um vetor, e os filtros de decomposição sejam substituídos por filtros de reconstrução dados pelos coeficientes $\{g_0[k]\}$ e $\{g_1[k]\}$. A Figura 44 contém a representação gráfica desse procedimento.

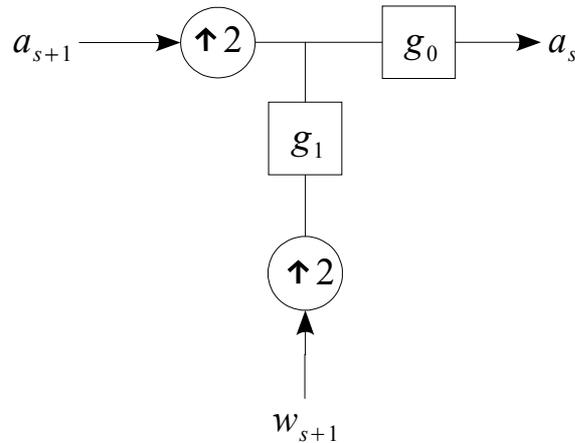


Figura 44 - Estrutura básica de reconstrução realizada pela representação em multirresolução efetuada através de bancos de filtros.

5.4.4 Famílias de Funções *Wavelets*

O fato da proposta das transformações *wavelets* serem baseadas na representação de sinais através da correlação destes com as chamadas funções *wavelets* e de que as propriedades dessas funções poderem influenciar nas representações obtidas, assim como é exposto na seção 5.4.1, possibilitou com que esforços fossem direcionados ao desenvolvimentos das chamadas famílias de funções *wavelets*.

Um família de funções *wavelets* corresponde a um ou mais pares formados por uma função escala e uma função *wavelet* relacionadas. Cada uma das famílias existentes [48] gera os subespaços de representação de uma maneira em que as decomposições de sinais nesses subespaços contenham características que são melhor evidenciadas ou mitigadas, dependendo das características da própria família utilizada.

Dependendo da quantidade de momentos nulos e de algumas outras características, a tabela seguinte, composta a partir de informações presentes na referência [76], apresenta as famílias de *wavelets* ortogonais mais comuns.

Tabela 9 - Principais famílias de *wavelets* ortogonais e suas respectivas características

Família	Momentos Nulos	Características
Haar	1	Suporte compacto, função escala simétrica e função <i>wavelet</i> anti-simétrica.
Shannon (sinc)	Infinitos	Funções escala e <i>wavelet</i> simétricas e suporte infinito.
Meyer	Infinitos	Funções escala e <i>wavelet</i> simétricas, limitada em banda, suporte infinito mas com rápido decaimento e diferenciável infinitamente.
Daubechies N	N	Suporte compacto, não há simetria para $n > 1$ e o tamanho dos filtros associados é $2N$.
Symlets N	N	Suporte compacto, o tamanho dos filtros associados é $2N$, a função escala é de fase linear.
Coiflets N	A função <i>wavelet</i> contém $2N$, enquanto a escala, $2N-1$.	Suporte compacto, o tamanho dos filtros associados é $6N$ e é quase simétrica.

CAPÍTULO 6

Modelo Proposto

Este trabalho propõe uma metodologia de reconhecimento de cargas elétricas não-lineares monofásicas através dos coeficientes obtidos pela decomposição *wavelet* em multirresolução de suas formas de onda de corrente em estado estacionário. Esse modelo é baseado no fato de que as cargas elétricas não-lineares criam distorções em suas formas de onda de corrente que podem individualizar cada um das classes de cargas elétricas não-lineares existente [45]. Outra consideração tomada ao se estruturar a proposição desse modelo é a possibilidade de que as transformações *wavelet*, ao terem sido desenvolvidas para se trabalhar com descontinuidades e transitórios, que são comuns às formas de corrente de cargas não-lineares [45], e por terem uma considerável capacidade [46] de compactar as características dos sinais que analisam, possam ser utilizadas como mecanismos de extrações das características que podem ser suficientes para se reconhecer as classes às quais pertençam cada um das formas de onda de corrente analisadas pela metodologia proposta. De uma

forma geral, o modelo é baseado no fato de que se cada um das classes de cargas elétricas não-lineares possui distorções em suas formas de onda de corrente que são diferentes das distorções das demais classes, e se as transformações *wavelet* são capazes de efetuar a correta decomposição dessas formas de onda, então, um sistema de reconhecimento de padrões que utilize redes neurais artificiais para classificar os espaços vetoriais das características extraídas dessas formas de onda de corrente pode ser, assim, capaz de reconhecer a qual classe de cargas elétricas não-lineares pertence uma carga elétrica cuja forma de onda de corrente seja analisada. Uma representação gráfica da idéia proposta é apresentada na Figura 45, enquanto a estrutura do sistema que faz parte dessa proposta, na Figura 46.

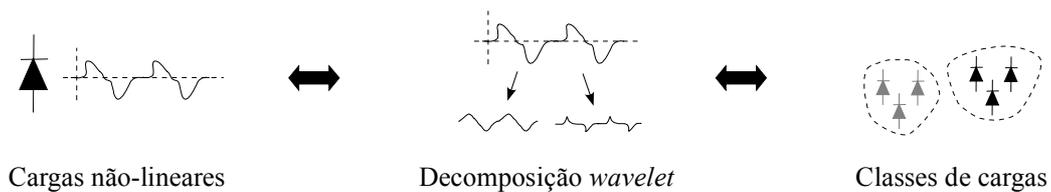


Figura 45 - Representação do modelo proposto, que busca relacionar os coeficientes da decomposição *wavelet* de uma forma de onda de corrente à classe de carga elétrica não-linear correspondente.

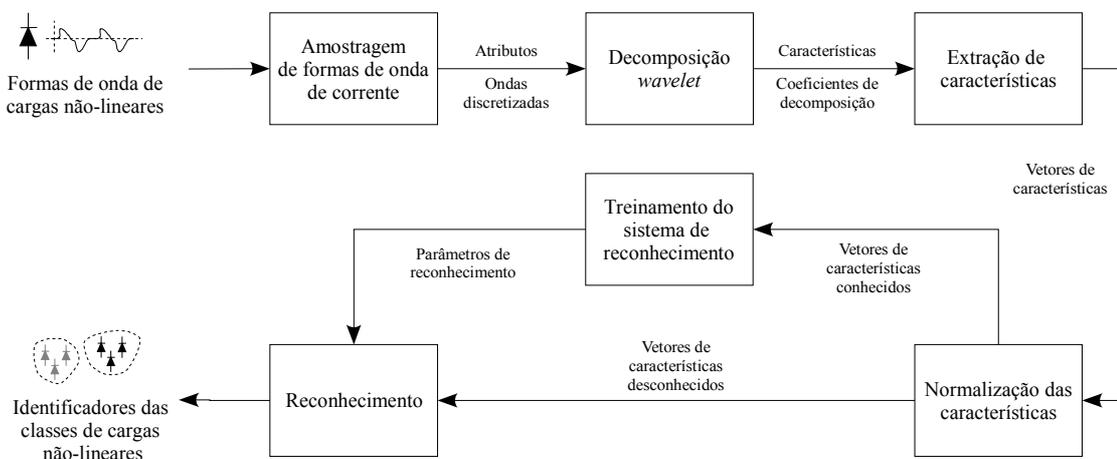


Figura 46 - Diagrama em blocos do sistema proposto.

6.1 Amostragem

O processo de amostragem tem como objetivo digitalizar as formas de onda de corrente que serão analisadas pelo sistema proposto. Esse processo leva em consideração as resoluções necessárias para se representar corretamente tanto as magnitudes das formas de onda quanto as frequências que elas contenham.

Os parâmetros escolhidos para o processo de amostragem do modelo são uma resolução de 16 *bits* por amostra coletada e uma frequência de coleta de 15.360 Hz. Essa frequência foi escolhida para que sejam gerados 256 pontos a cada ciclo da forma de onda fundamental presente nos sistemas elétricos brasileiros, que é de 60 Hz. Com essa frequência de amostragem, pode ser feita a discretização correta de sinais até a frequência de 7.680 Hz, o que corresponde à 128ª onda harmônica .

Os resultados desse processo são formas de onda de corrente em tempo discreto que são tratadas como vetores de atributos para o processo de decomposição subsequente.

6.2 Decomposição

O modelo utiliza um processo de decomposição *wavelet* em multirresolução para representar e então inicialmente extrair as características presentes nos vetores de atributos

provenientes do processo de amostragem. Esse processo gera um conjunto de formas de onda que representam a onda de corrente original através de um sinal de aproximação e de um conjunto de sinais de detalhes cujas resoluções dependem do nível de resolução adotado.

Os parâmetros relacionados ao nível de decomposição e à família de funções *wavelets* foram obtidos de forma prática, ao decorrer da aplicação do modelo proposto.

6.3 Extração de Características

Para completar a extração de características iniciada pelo processo de decomposição *wavelet*, foi escolhido representar cada uma das formas de onda precedentes através de suas energias.

Tanto para o sinal de aproximação quanto para o sinais de detalhes, suas energias são obtidas através da equação

$$e = \sum_{j=1}^n x_j^2, \quad (6.3.1)$$

onde a energia e do sinal de componentes $\{x_j\}$, que pode ser tanto o sinal de aproximação quanto os sinais de detalhes, é obtida como sendo o resultado da soma dos quadrados de cada um de seus n componentes.

Cada uma das energias associadas à representação *wavelet* foi adotada como um elemento do vetor de características, formando um vetor com a forma

$$\hat{\mathbf{v}} = [a_{M+N}, d_M, d_{M+1}, \dots, d_{M+N}]. \quad (6.3.2)$$

6.4 Normalização de Características

A forma de normalização adotada para o vetor de características consiste em obter o percentual que a energia de cada sinal de representação tem em relação à energia do sinal original. A equação (6.3.2) exemplifica essa relação.

$$\bar{e}_i = \frac{e_i}{e_{\text{sinal}}}. \quad (6.4.1)$$

Assim, o vetor de características, apresentado na equação (6.3.1), toma a forma

$$\bar{\mathbf{v}} = [\bar{a}_{M+N}, \bar{d}_M, \bar{d}_{M+1}, \dots, \bar{d}_{M+N}]. \quad (6.4.2)$$

Essa forma de normalização foi escolhida devido ao fato da decomposição *wavelet* em multirresolução poder conservar, nos sinais de aproximação e de detalhes, a energia do sinal original [48].

6.5 Reconhecimento e Treinamento

Para que o sistema proposto possa reconhecer e classificar as cargas elétricas não-lineares analisadas, foi adotada uma rede neural artificial de múltiplas camadas como mecanismos de reconhecimento e classificação.

Em modo de reconhecimento, essa rede recebe os vetores de características normalizados e então busca associá-los a um vetor que identifique a classe às quais os vetores pertençam.

Já em modo de treinamento, a rede neural recebe um conjunto de vetores de características e os respectivos vetores de identificação de classe para, então, classificar seu espaço de vetores de características de modo a ajustar seus pesos e níveis internos, que são utilizados como parâmetros da rede neural em seu modo de operação.

Quanto à estrutura adotada para a rede neural artificial, o número de nós na camada de entrada e de neurônios nas camadas de processamento e de saídas, bem como o próprio número de camadas de processamento, foram estipulados ao decorrer da aplicação do modelo proposto.

CAPÍTULO 7

Implementação do Modelo

Para que o modelo proposto de reconhecimento de cargas elétricas não-lineares pudesse ser aplicado, foram implementadas e validadas cada uma das técnicas relacionadas. A primeira atividade de implementação gerou o projeto e o protótipo de um equipamento específico de aquisição de dados que pudesse atender aos requisitos de amostragem e resolução utilizados pelo modelo. As implementações e os testes que se seguiram foram a de uma rede neural artificial de múltiplas camadas, em companhia a dois algoritmos de treinamento, a de uma transformada rápida de Fourier e a de uma decomposição *wavelet* em multirresolução.

Todos os algoritmos foram desenvolvidos por meio da utilização da linguagem de programação C++ para ambientes gráficos. Já o programa embarcado desenvolvido para o equipamento de aquisição de dados foi implementado através de linguagem C específica para microcontroladores.

7.1 Sistema de Aquisição de Dados

Para que pudessem ser amostradas formas de onda de corrente segundo as especificações necessárias (vide a seção 6.1), foi desenvolvido um equipamento próprio de aquisição de dados.

Esse equipamento é composto por sensores de efeito *hall* como transdutores para as formas de onda de corrente e tensão, em união a filtros *anti-aliasing*, a um conversor de sinais analógico-digital, a uma unidade de microprocessamento e a um circuito de comunicação serial, com o qual o equipamento se comunica com sua interface de operação. As principais características do equipamento desenvolvido são apresentadas na Tabela 10.

Tabela 10 - Principais especificações do equipamento de aquisição de dados

Item	Especificação
Frequência de amostragem	15.360 Hz
Resolução por amostra	16 bits
Canais de entrada	2, sendo um para corrente e um para tensão
Limite do canal de tensão	+/- 250 V (pico)
Limite do canal de corrente	+/- 10 A (pico)
Comunicação	Serial, no padrão RS232
Sincronismo de amostragem	Passagem por zero da onda de tensão
Ciclos de amostragem	4 ciclos (onda de 60 Hz), que equivalem a 1024 pontos
Filtros analógicos	Arquitetura <i>Chebyshev</i> e <i>Salenkey</i> , com frequência de corte em 7.680 Hz

A operação do equipamento desenvolvido é controlada através da interface de operação apresentada na Figura 47, que permite com que sejam requisitadas a coleta de formas de onda de corrente e tensão bem como o armazenamento delas em arquivos a serem utilizados nas atividades de processamento digital de sinais presentes no modelo proposto.

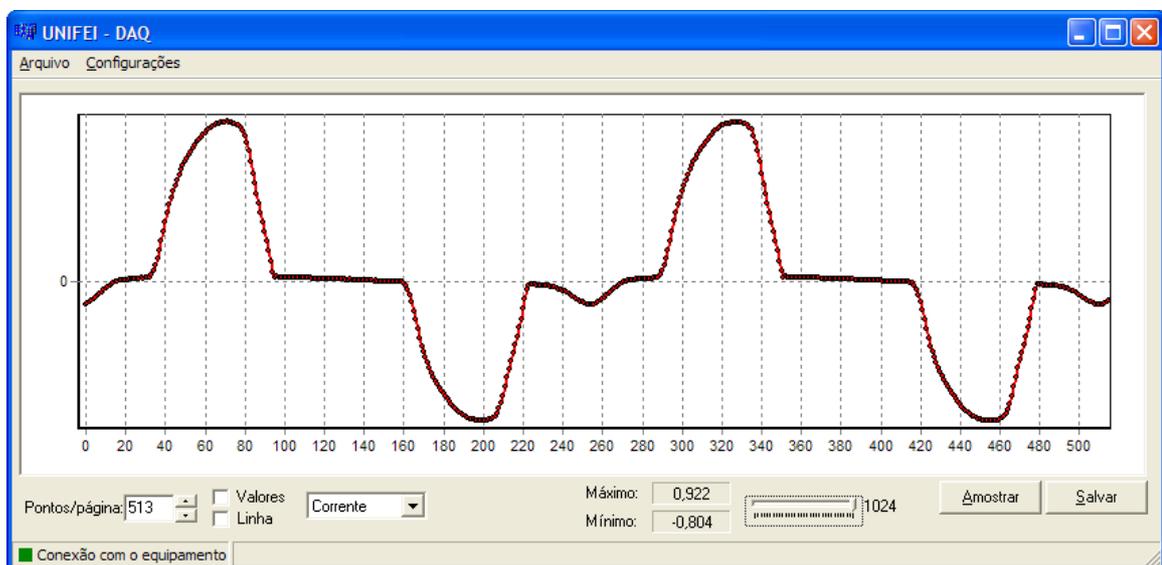


Figura 47 - Interface de operação do equipamento de aquisição de dados desenvolvido.

A Figura 48 exibe uma representação em diagrama de blocos do equipamento desenvolvido. Os nomes entre parênteses presentes nessa figura especificam os principais componentes elétricos e eletrônicos utilizados para realizar a operação especificada por cada um dos blocos presentes.

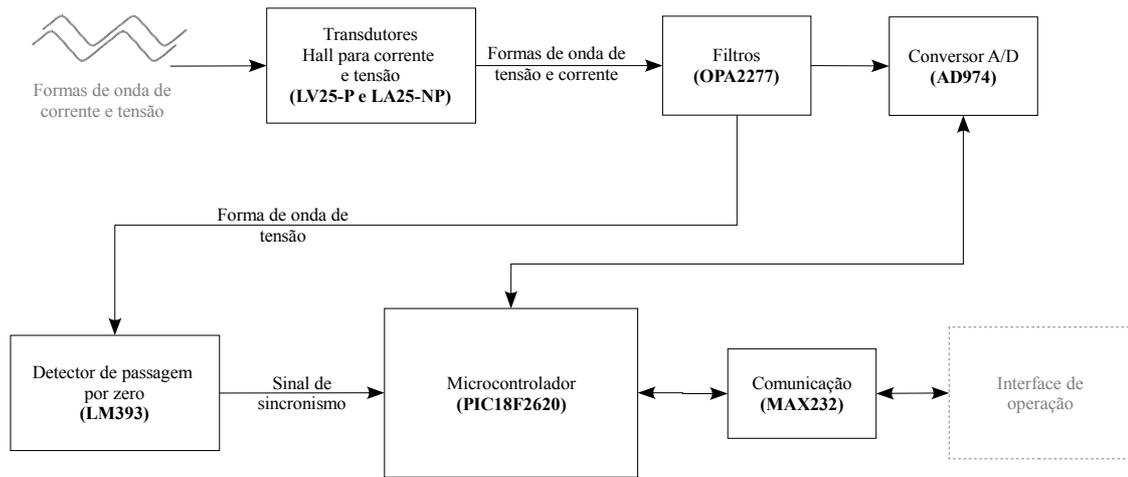


Figura 48 - Diagrama em blocos do equipamento de aquisição de dados desenvolvido.

O apêndice A apresenta os esquemas elétricos relacionados a cada um dos blocos presentes no diagrama da Figura 48.

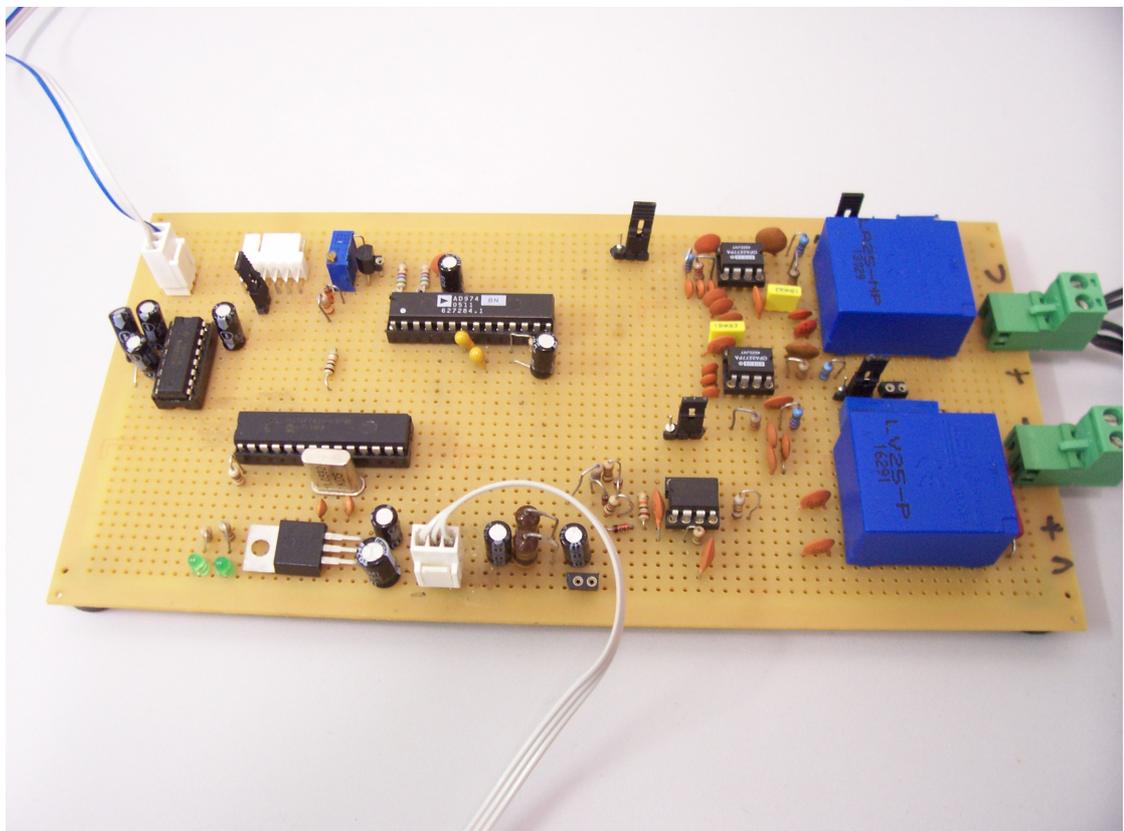


Figura 49 - Protótipo do equipamento de aquisição de dados desenvolvido.

7.2 Rede Neural Artificial de Múltiplas Camadas

A implementação de um rede neural artificial de múltiplas camadas foi realizada em três etapas distintas, sendo que inicialmente foi implementado seu modo de operação e depois dois algoritmos de treinamento, o de mínimo passo descendente e o de *Levenberg-Marquart*, ambos apresentados na seção 3.4.1.

Tanto o modo de operação quanto o de treinamento, que pode utilizar um dos dois algoritmos citados, foram implementados utilizando-se notação matricial, o que permitiu com que a estrutura da rede neural possua um número de camadas e de elementos em cada camada configuráveis e teoricamente irrestritos. Essas implementações foram baseadas nas implementações sugeridas pela referência [72].

Para que a rede pudesse ser treinada e então operasse, foi desenvolvido um arquivo de configuração em modo texto no qual são definidos seus parâmetros como o algoritmo de treinamento a utilizar, os fatores de aprendizado e afins e os vetores-alvo e de entrada utilizados para se treinar a rede. Um exemplo de arquivo de configuração é apresentado no apêndice B.

A Figura 50 apresenta a interface de operação e treinamento criada para se trabalhar com a rede neural artificial implementada.

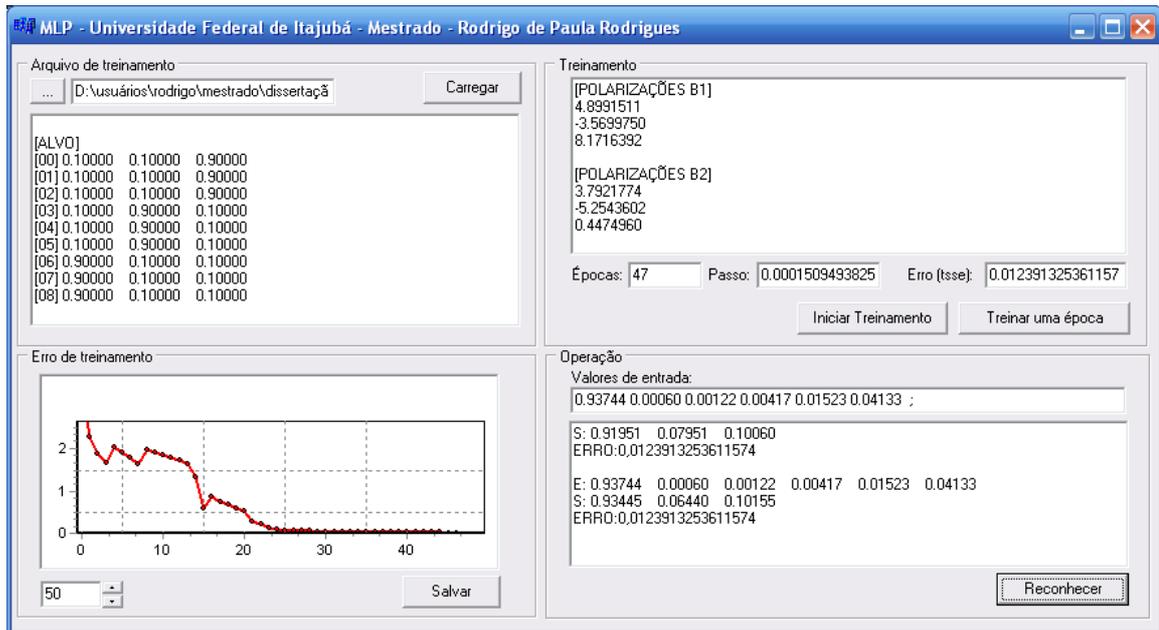


Figura 50 - Interface desenvolvida para se manipular a rede neural implementada.

7.2.1 Testes

Para validar as implementações dos modos de operação e de treinamento, foi escolhido reproduzir a operação da função matemática ou-exclusivo. Para dois valores de entrada, essa função matemática se comporta assim como o apresentado na Tabela 11. Essa função foi escolhida por ser um caso clássico de particionamento e manipulação do espaço vetorial [64]. Assim, foi escolhida um arquitetura de rede neural de múltiplas camadas com 2 nós de entrada, dois *perceptrons* intermediários e um *perceptron* de saída.

Tabela 11 - Função matemática “ou-exclusivo” para duas variáveis.

x	y	resultado (z)
0	0	0
0	1	1
1	0	1
1	1	0

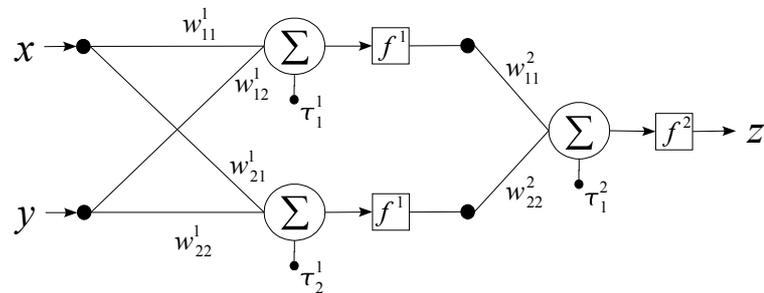


Figura 51 - Rede neural de múltiplas camadas utilizada para reproduzir a operação da função matemática ou-exclusivo para duas variáveis.

Para realizar os testes, foram gerados, aleatoriamente, dois conjuntos de pesos e níveis de ativação. Cada um desses conjuntos de pesos foi utilizado para realizar um treinamento tanto com o algoritmo de mínimo passo descendente quanto com o algoritmo de *Levenberg-Marquardt*. Foram, então, levantados os erros de treinamento obtidos em função do número de épocas necessárias para se chegar a um erro próximo a 0,005. A variação do erro obtida para cada treinamento é apresentada na Figura 52, para o algoritmo de mínimo passo descendente, e na Figura 53, para o de *Levenberg-Marquardt*.

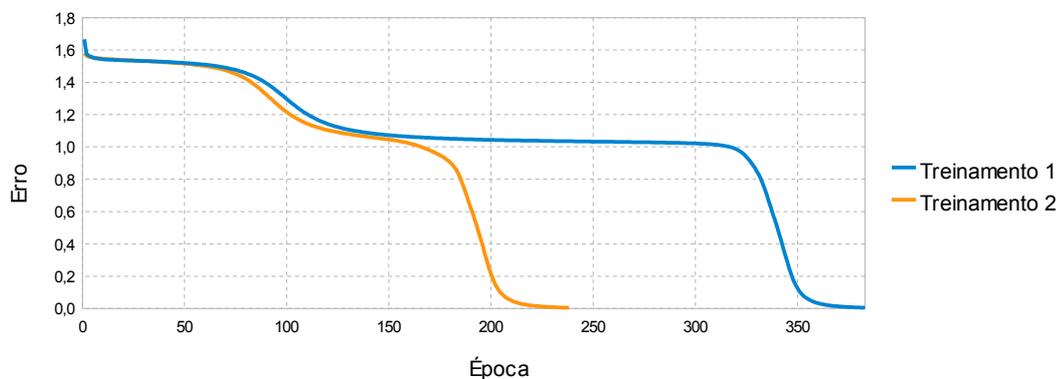


Figura 52 - Erros de treinamento obtidos em dois processos de treinamento específicos onde se utilizou o algoritmo de mínimo passo descendente.

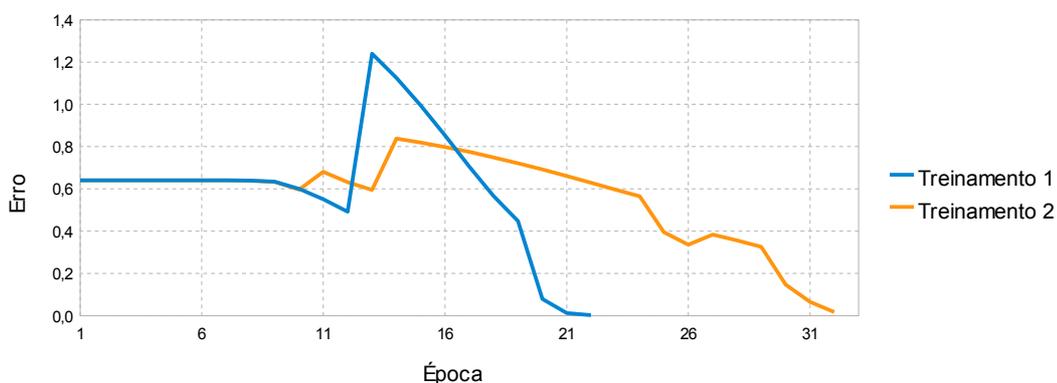


Figura 53 - Erros de treinamento obtidos em dois processos de treinamento específicos onde se utilizou o algoritmo de *Levenberg-Marquardt*.

Pelos resultados apresentados por essas figuras, e embora esses sejam testes específicos que não representam todas as possibilidades existentes, há um indicativo de que o algoritmo de *Levenberg-Marquardt* é mais rápido que o outro implementado, uma vez que o número de épocas de treinamento para essa implementação foi visivelmente inferior ao de mínimo passo descendente. Esse comportamento já era esperado, tendo-se em vista os resultados apresentados na referência [73], que apresenta comparações entre algoritmos de treinamento e chega à conclusão que o de *Levenberg-Marquardt* é um dos melhores, pois além de não necessitar de otimizações relacionadas a momentos de aprendizado, à busca de valores iniciais para os pesos, dentre outros [64], é capaz de convergir de maneira rápida para a solução. Alguns outros testes com o algoritmo em questão são apresentados no apêndice B.

7.3 Transformada Rápida de Fourier

Foi utilizado o algoritmo de raiz 2 com decimação no tempo para implementar a transformada rápida de Fourier utilizada por este trabalho. Esse algoritmo é explicado na seção 4.5.1 e seu código-fonte é apresentado no apêndice C. Um fato importante desse algoritmo é sua limitação quanto a trabalhar somente com sinais discretos cujos tamanhos sejam dados por potências de 2.

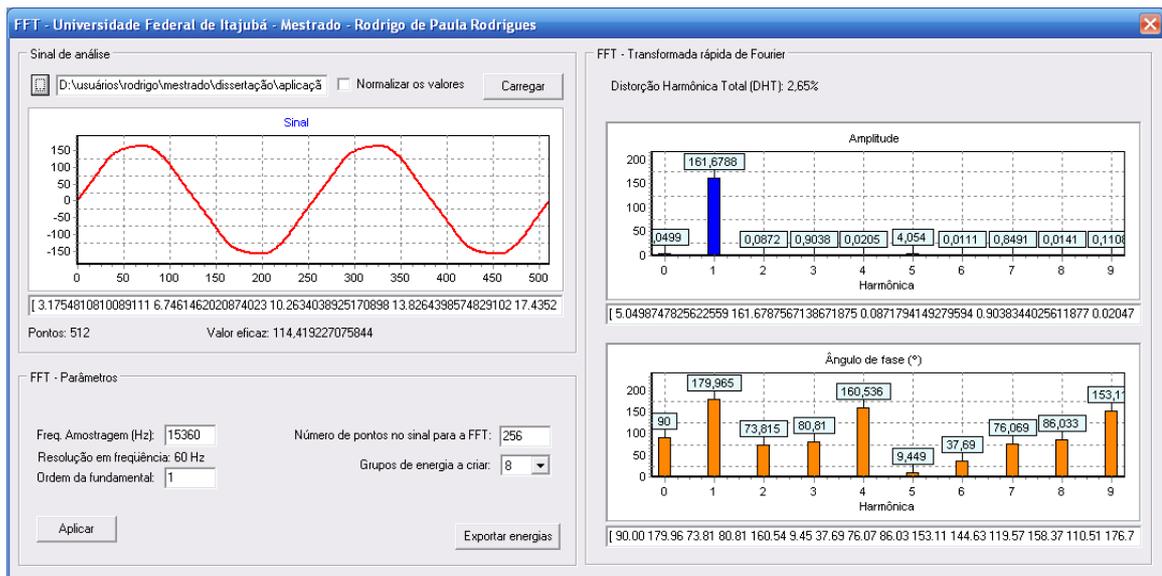
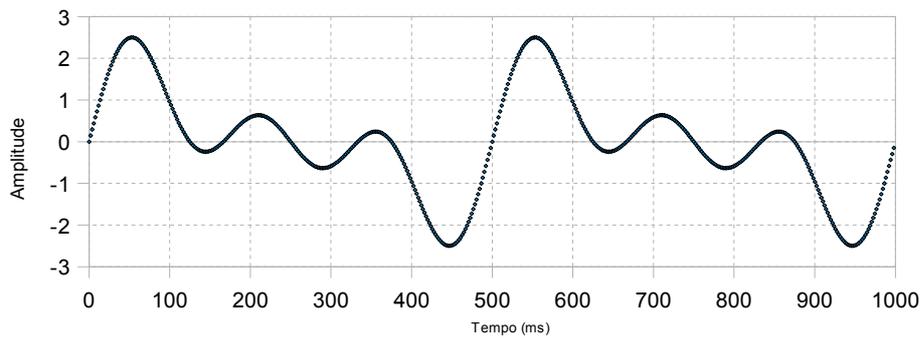


Figura 54 - Interface de operação da transformada rápida de Fourier implementada para o modelo.

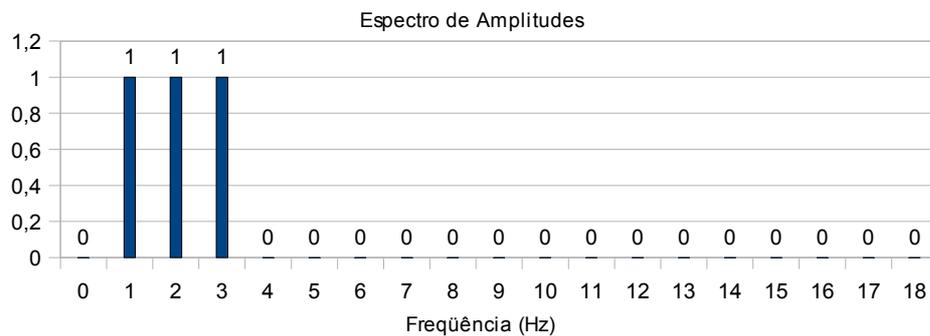
7.3.1 Testes

Os testes de validação da implementação da transformada rápida de Fourier foram simples, tendo sido geradas formas de onda contendo frequências conhecidas que foram então submetidas à decomposição implementada.

A Figura 55 apresenta um dos testes realizados. Nesse caso, uma onda contendo componentes com amplitudes unitárias e frequências de 1, 2 e 3 Hz, foi amostrada com uma frequência de 512 Hz durante 1 s. O espectro presente na Figura 55 (b) ajuda a validar a implementação realizada.



(a)



(b)

Figura 55 - Forma de onda contendo as frequências de 1, 2 e 3 Hz e seu espectro de amplitudes obtido através da implementação da transformada rápida de Fourier.

7.4 Transformada Wavelet Discreta

A implementação da decomposição de sinais em uma representação de multirresolução através da transformada *wavelet* discreta foi feita tendo-se como base o algoritmo de bancos de filtros apresentado na seção 5.4.3, que se resume em basicamente aplicar, recursivamente,

as equações (5.4.7) e (5.4.8) a fim de decompor o sinal original em um conjunto de sinais que descrevam sua aproximação em um nível N e seus detalhes em cada nível de resolução anterior e igual ao nível N adotado. O ponto inicial para essa implementação foi considerar que o sinal a ser decomposto equivale a sua aproximação em um nível gerado pelo subespaço A_0 , para, então, obter sua aproximação e seus detalhes em subespaços de menores resoluções.

Ao se analisar as equações (5.4.7) e (5.4.8), pode-se concluir que os procedimentos a serem realizados pela transformada *wavelet* discreta são um par de convoluções e um par de decimação que representam o descarte de um a cada dois elementos do resultado da convolução. Nesse sentido, o que o algoritmo implementado realmente faz é convoluir cada uma das aproximações do sinal inicial, incluindo este como sendo uma aproximação no subespaço A_0 - como já foi dito, e decimar cada um dos resultados. Assim, ao final de N processos de composição do sinal original, são obtidos uma aproximação do sinal no subespaço de resolução A_N e os respectivos sinais de detalhes em cada um dos subespaços $\{W_j\}_{1 < j \leq N}$.

Para que as equações citadas pudessem ser utilizadas, os coeficientes dos filtros de cada uma das famílias disponibilizadas pela implementação tiveram que ser levantados, o que foi feito utilizando funções do pacote de ferramentas sobre *wavelets* presente na versão *R2007a* do aplicativo *Matlab*. Esses coeficientes de decomposição foram listados em arquivos específicos para cada família de *wavelet* com a qual a implementação pretendeu trabalhar. No apêndice D são apresentados alguns desses coeficientes utilizados.

Uma outra consideração importante a respeito da implementação efetuada da transformada *wavelet* é uma questão que remonta às limitações inerentes à representação de sinais em ambientes computacionais, a limitação em tamanho dos sinais discretizados. Essa limitação acarreta em imperfeições ao início e ao fim do sinal resultante de uma convolução discreta [43]. Essas imperfeições, que podem ser geradas pela convolução discreta presente no algoritmo da transformada *wavelet* implementado, podem agregar erros de representação aos sinais de aproximação e de detalhes obtidos através da decomposição em multirresolução. Uma forma de atenuar esses problemas no algoritmo implementado foi ampliar o número de amostras no sinal de entrada. Para tanto, foram implementadas duas técnicas simples: a ampliação do sinal de entrada mediante a inclusão de zeros ao início e ao final do sinal original, e a ampliação através de simetrização das extremidades do sinal. A Figura 56 exemplifica ambas as abordagens adotadas.

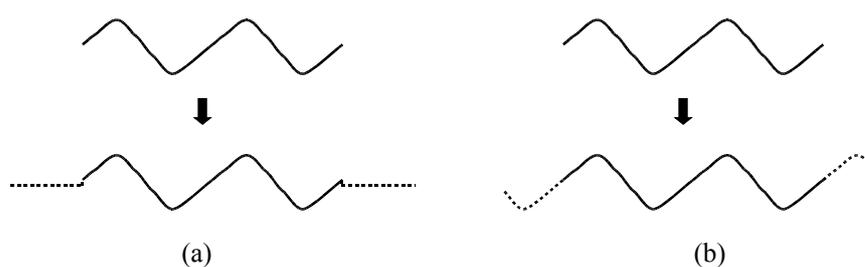


Figura 56 - Ampliação de formas de onda através do preenchimento das extremidades com valores 0 (a) e com valores simétricos (b).

A interface de aplicação da decomposição em multirresolução implementada é apresentada na Figura 57 e o código-fonte dessa mesma implementação, no apêndice D.

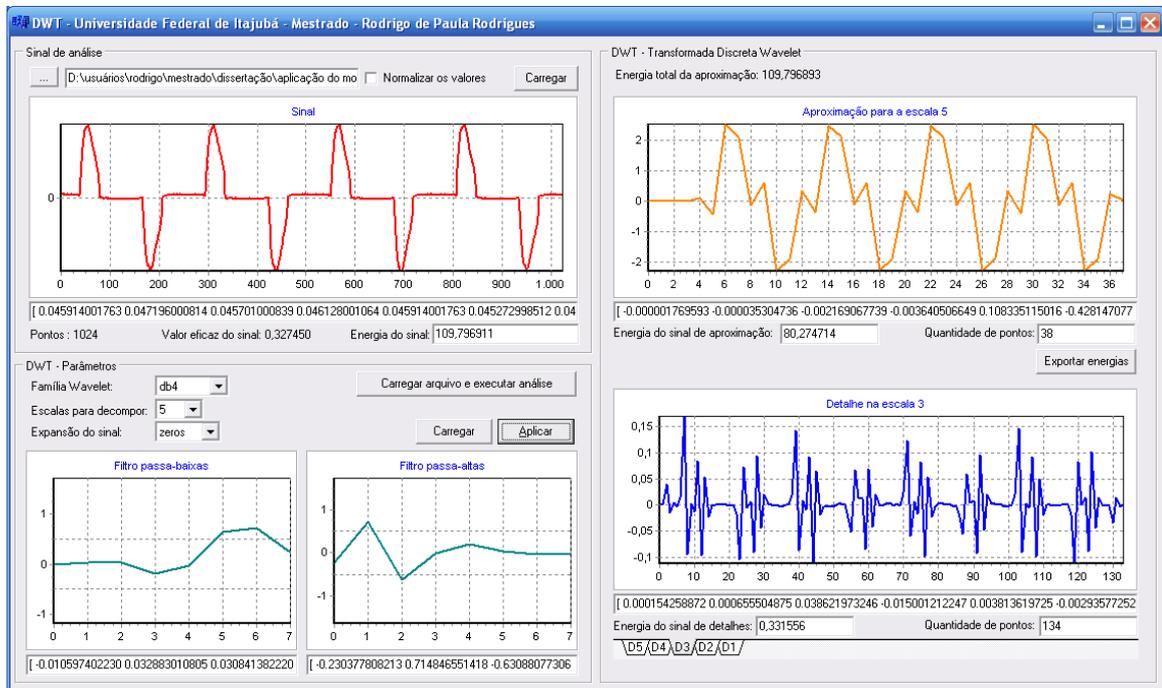


Figura 57 - Interface de operação da decomposição em multirresolução implementada para o modelo.

7.4.1 Testes

De uma forma geral, os testes realizados com a implementação da transformada *wavelet* discreta tiveram o objetivo de validar o desenvolvimento e foram compostos pela geração de formas de onda e suas respectivas decomposições. Como parâmetro de validação para a implementação, foi utilizado o fato de que a energia contida no sinal original ser preservado quando esse sinal é decomposto através de uma transformação *wavelet*, o que já foi mencionado na seção 6.4.

A Figura 58 apresenta uma das formas de onda utilizadas para desenvolver os testes citados. Essa forma de onda contém três componentes com amplitude unitária e cada qual com uma das frequências de 1 Hz, 2 Hz ou 3 Hz.

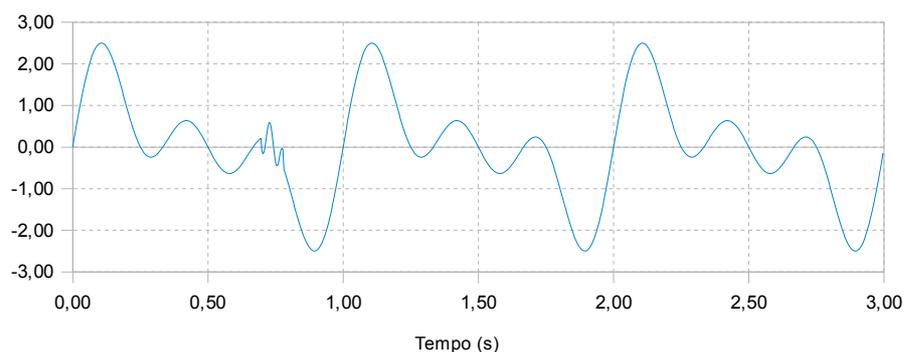


Figura 58 - Forma de onda utilizada para se realizar um dos testes de validação da transformada *wavelet* implementada.

Uma das decomposições da forma de onda presente na Figura 58 foi realizada até o quinto nível de resolução e se utilizando os coeficientes dos filtros de decomposição pertencentes à família de *wavelets* Haar [76]. Um resultado importante dessa decomposição foi o fato de que a energia presente no sinal original, obtida através da equação (6.3.1) e que é da ordem de 1.153 unidades, foi preservada na decomposição e, assim, distribuída entre o sinal de aproximação e os 5 sinais de detalhes. Essa decomposição utilizou o método de adaptação do sinal de entrada através da adição de extremidades com valores zero.

A Figura 57 apresenta as formas de onda obtidas através da decomposição citada e permite evidenciar que o sinal de aproximação obtido é mesmo uma simplificação do sinal presente na Figura 58.

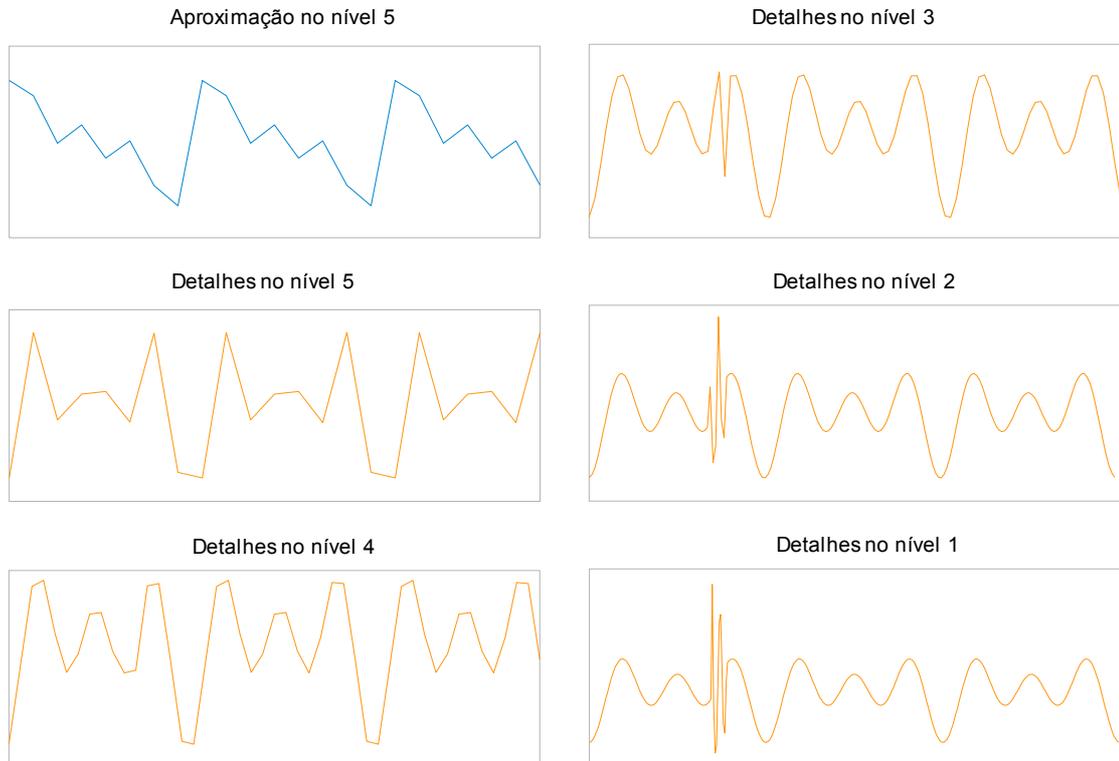


Figura 59 - Aproximação em 5 níveis da forma de onda presente na Figura 58 e obtida através da transformada discreta *wavelet* implementada

A forma de onda presente na Figura 58 também foi decomposta através de outras famílias de funções *wavelets* como a Daubechies 2 e 4 e também através de diferentes níveis de resolução, tendo-se verificado que a energia do sinal original foi preservada em todas as decomposições efetuadas.

CAPÍTULO 8

Aplicação do Modelo

Para que o modelo fosse aplicado, foram desenvolvidas as atividades de:

- levantamento de formas de onda de corrente pertencentes a algumas cargas elétricas monofásicas não-lineares;
- análise das formas de onda levantadas;
- teste do modelo proposto quanto ao reconhecimento individual de cargas elétricas através da análise de formas de onda de corrente produzidas por uma única classe de equipamentos;
- teste do modelo proposto quanto ao reconhecimento das cargas elétricas associadas a formas de onda de corrente produzidas por um conjunto de equipamentos;

- realização de um teste comparativo entre o modelo proposto e um sistema similar de reconhecimento baseado na decomposição das formas de onda de corrente através da transformada discreta de Fourier.

8.1 Levantamento de Formas de Onda de Corrente

O levantamento de formas de onda de corrente de algumas cargas elétricas foi efetuado por meio do equipamento de aquisição de dados desenvolvido. Para um total de 8 classes de equipamentos, foram levantadas suas ondas de corrente através de 1024 amostras coletadas com a frequência de 15.360 Hz, que resultaram em 256 amostras por ciclo de 60 Hz.

A Tabela 12 apresenta as classes de equipamentos cujas formas de onda de corrente foram levantadas, enquanto a Figura 60 apresenta exemplos de formas de onda de corrente que foram coletadas para cada uma das classes analisadas.

Tabela 12 - Equipamentos cujas formas de onda de corrente foram levantadas.

Equipamentos analisados	
Aparelho de som	Osciloscópio analógico
Aparelho de telefone	Osciloscópio digital
Gerador de funções	Televisor de cristal líquido
Lâmpada fluorescente compacta	Televisor de raios catódicos

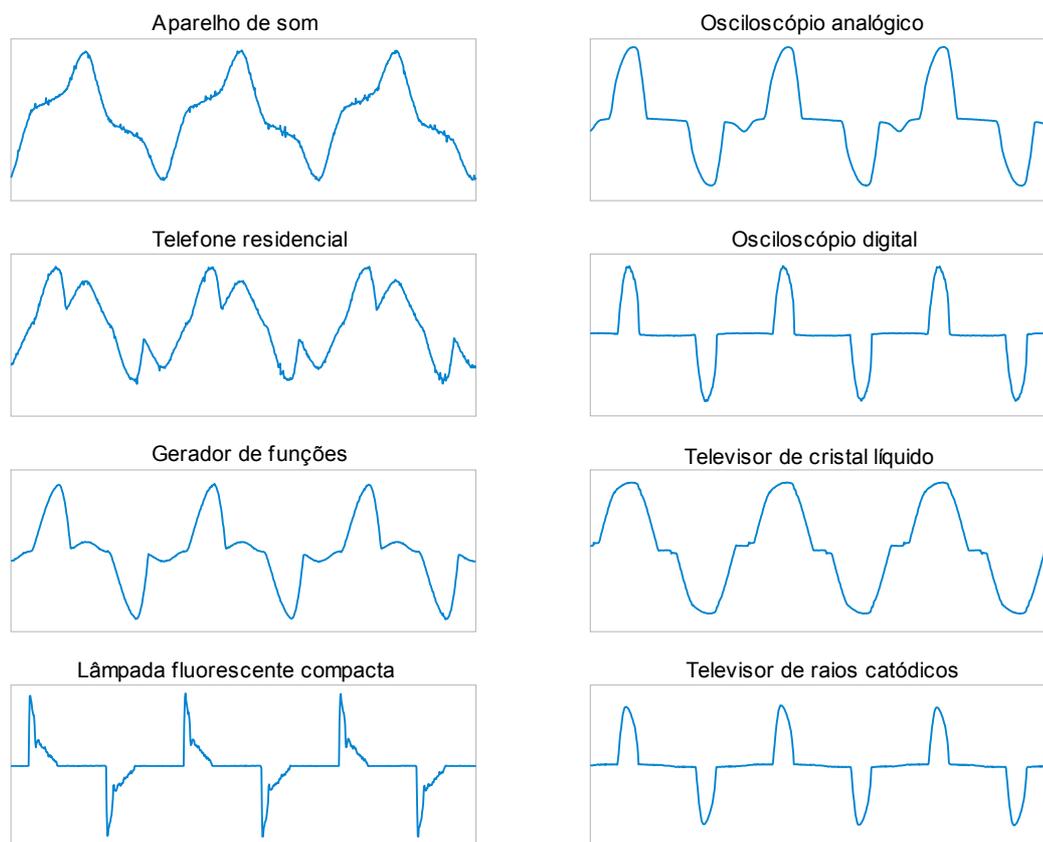


Figura 60 - Exemplos de formas de ondas de corrente dos aparelhos analisados.

Através da Figura 60 pôde-se notar que as formas de onda de corrente relacionadas à classes analisadas são essencialmente periódicas e com várias discontinuidades, o que já era esperado por se tratarem de cargas elétricas não-lineares [45]. Ainda pela mesma figura, pode ser observado que as formas de onda de corrente relacionadas ao televisor de raios catódicos e ao osciloscópio digital são semelhantes, o que provavelmente ocorre devido ao fato de que esses equipamentos possuem um mesmo tipo de circuito de conversão de corrente alternada para corrente contínua. Embora não tenham sido apresentadas, as formas de onda levantadas para monitores de raios catódicos, para monitores de cristal líquido e mesmo para computadores pessoais são semelhantes à forma de onda do osciloscópio digital analisado.

8.2 Análise das Formas de Onda

Para que as formas de onda coletadas fossem manipuladas pelo sistema de reconhecimento proposto, cada uma delas foi decomposta pela transformada *wavelet* discreta em multirresolução. Como as família de funções *wavelet* têm propriedades específicas e assim podem gerar representações diferentes de um mesmo sinal, foram utilizadas as famílias de funções *wavelet* Haar e Daubechies 4, com as quais cada sinal de corrente coletado foi decomposto em 5 e 7 níveis de resolução, o que proporcionou a obtenção de um sinal de aproximação e mais 5 ou 7 sinais de detalhes, dependendo do nível de resolução adotado.

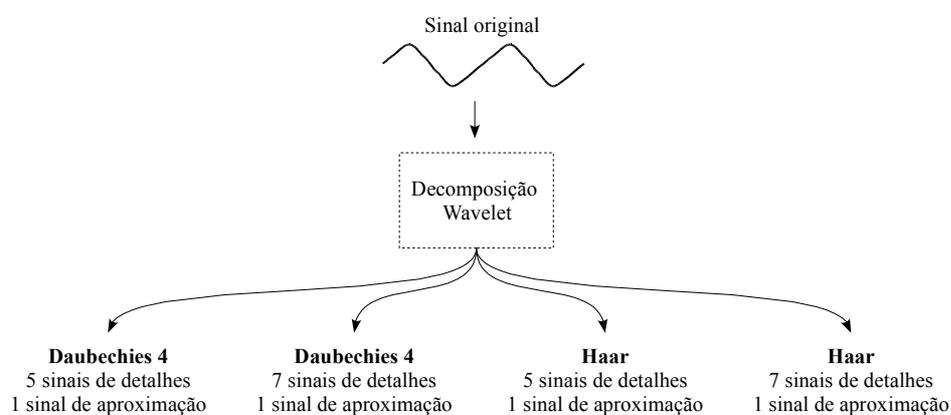


Figura 61 - Decomposições *wavelet* em multirresolução aplicadas a cada forma de onda de corrente coletada.

Através da decomposição *wavelet* dos sinais de corrente, foram obtidas as energias e os vetores de características definidos pelas equações (6.4.1) e (6.4.2), respectivamente. Por exemplo, para o caso da forma de onda presente na Figura 60 relacionada a um osciloscópio analógico, seus vetores de características obtidos são apresentados na Tabela 13.

Tabela 13 - Vetores de características obtidos a partir da forma de onda de corrente de um osciloscópio analógico.

Família	Níveis	A5/7	D1	D2	D3	D4	D5	D6	D7
Daub. 4	5	0.91533	0.00006	0.00006	0.00082	0.01192	0.07180	-	-
	7	0.35413	0.00006	0.00006	0.00082	0.01192	0.07180	0.25417	0.30704
Haar	5	0.81568	0.00068	0.00253	0.01063	0.04876	0.12172	-	-
	7	0.42314	0.00068	0.00253	0.01063	0.04876	0.12172	0.38675	0.00579

Os vetores presentes na Tabela 13 permitem levantar algumas peculiaridades da decomposição e da representação adotadas para os vetores de características do modelo proposto. A primeira delas diz respeito à preservação da energia do sinal original em suas formas de onda de aproximação e de detalhes, pois se forem somados cada um dos sinais que definem as 4 representações obtidas, se obtém, para cada um deles, o valor 1, o que significa dizer que a energia do sinal original é distribuída ao longo dos sinais obtidos através de sua decomposição e assim o sinal original é representado de forma completa. Outra questão de relevância que pode ser levantada através de uma segunda análise da Tabela 13 é o fato de que à medida que o sinal de corrente é decomposto através de mais níveis de detalhes, a energia contida em seu sinal de aproximação é também decomposta nos sinais de detalhes que são agregados à representação. Essa questão é muito importante para a metodologia proposta por este trabalho, pois se duas formas de ondas pertencentes a classes diferentes de equipamentos forem ligeiramente semelhantes em termos de seus vetores de características, esses vetores podem ser modificados de forma a serem formados por mais níveis de detalhes até que sejam encontrados detalhes que possibilitem diferenciar cada uma das ondas. Uma terceira questão interessante é o fato de que pelas formas de onda de corrente analisadas se tratarem de formas de onda com frequência fundamental de 60Hz, a distribuição de energia de suas decomposições em multirresolução tendem a serem concentradas nos sinais de aproximação,

pois esse sinal representa, até um certo nível de decomposição, a frequência fundamental da forma de onda analisada. Nesse sentido, uma maneira interessante de se definir esses vetores de características é pensar neles como se fossem uma distribuição de energia que contempla a frequência fundamental através do sinal de aproximação e as demais energias como faixas de frequências que são representadas pelos sinais de detalhes.

8.3 Reconhecimento de Cargas Individuais

Um teste inicial para o modelo proposto foi validar sua capacidade de identificar cargas elétricas individuais através das formas de onda de corrente que individualmente produzem. Esse teste teve como objetivo principal verificar se a extração de características efetuada pela decomposição *wavelet* em multirresolução é capaz de levantar detalhes que pudessem individualizar cada uma das classes de cargas elétricas monofásicas não-lineares analisadas.

O primeiro passo para se realizar esse teste inicial foi escolher três classes de cargas elétricas distintas para então estudar as formas de onda relacionadas. Assim, foram escolhidas as classes que representam lâmpadas fluorescentes compactas, aparelhos de televisão e aparelhos de som. Como passo seguinte, foram coletadas formas de onda de corrente de conjuntos de elementos que representaram cada uma das classes escolhidas. A Tabela 14 detalha tais conjuntos.

Tabela 14 - Conjuntos de formas de onda de corrente obtidos para o teste inicial do modelo proposto

Classe	Quantidade	Especificações
Lâmpada fluorescente compacta	10	De 11 W a 25 W e de diferentes fabricantes.
Aparelho de som	10	De 5W a 10W e de diferentes fabricantes e modelos.
Televisor	9	Tanto de raios catódicos quanto de cristal líquido, 12 a 29 polegadas e de diferentes fabricantes.

Dos 29 exemplares presentes nos três conjuntos levantados, 9 unidades foram utilizadas para treinamento, três de cada classe, e 20 para validação. Ao total foram feitos 5 testes que utilizaram os mesmos parâmetros para a rede neural, como passos de treinamento e funções de ativação. As alterações entre os testes foram na estrutura da rede e no uso dos vetores de características, que foram gerados por 5 ou 7 níveis de detalhes através da família de funções *wavelets* Haar e Daubechies 4. O algoritmo utilizado para o treinamento foi o de *Levenberg-Marquardt*, que foi implementado especialmente para o modelo proposto e teve resultados melhores que o de mínimo passo descendente. Cada um das classes de cargas elétricas presentes nos testes foram codificadas como vetores de identificação de classes segundo a relação apresentada na Tabela 15.

Tabela 15 - Codificação dos vetores-alvo para o teste de reconhecimento individual

Classe	Saída 1	Saída 2	Saída 3
Lâmpada fluorescente	0.1	0.1	0.9
Aparelho de som	0.1	0.9	0.1
Televisor	0.9	0.1	0.1

Os resultados dos 5 testes são listados na Tabela 16, enquanto cada um dos arquivos de configuração utilizados durante esses testes são listados no apêndice E.

Tabela 16 - Resultados dos testes de identificação individual de cargas elétricas monofásicas não-lineares através do modelo proposto

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
Família wavelet	Haar	Haar	Haar	Daubechies 4	Daubechies 4
Níveis de decomposição	5	5	7	5	7
Número de camadas da rede	3	3	3	3	3
Estrutura da rede	6-3-3	6-12-3	8-6-3	6-3-3	8-6-3
Épocas de treinamento	52	42	30	40	25
Erro de treinamento	0,000983	0,000917	0,000813	0,000791	0,000626
Vetores de validação	20	20	20	20	20
Reconhecimentos corretos	19	20	20	20	20
Falsos reconhecimentos	1	0	0	0	0
Eficiência do sistema	95 %	100 %	100 %	100 %	100 %

Como se pode observar nos dados presentes na Tabela 16, todos os cinco testes efetuados com o sistema obtiveram pelo menos 95 % de reconhecimentos corretos. Esses resultados significam que para as três classes de cargas elétricas analisadas, cada um das decomposições *wavelet* em multirresolução foi capaz de extrair características que puderam individualizar as formas de onda de corrente analisadas. Uma vez que tais formas de onda são diferentes de classe para classe, como pode ser observado nas curvas de corrente de um aparelho de som, de um televisor e de uma lâmpada fluorescente compacta, que são apresentadas na Figura 60, esse resultado sugere que se as formas de onda de corrente produzidas por uma associação de cargas elétricas não-lineares forem diferentes das formas de onda individuais dos equipamentos presentes na associação, mesmo que ligeiramente, então uma decomposição em multirresolução dessas formas de onda pode resultar em sinais de

representação que expressem tais diferenças, da mesma maneira como foram expressas as diferenças existentes entre as formas de onda de corrente produzidas pelos equipamentos utilizados nesses cinco testes realizados.

8.4 Reconhecimento de Associações de Cargas Elétricas

Na prática, os ramais de distribuição de energia elétrica hospedam formas de onda de corrente que são produzidas por associações entre várias cargas elétricas. Nessa condição, para que a idéia básica do sistema proposto possa ser aplicada junto a ramais de distribuição ou mesmo em outras condições reais nas quais sejam analisadas formas de onda de corrente originadas por associações de cargas elétricas, foi necessário verificar se a decomposição em multirresolução é capaz de gerar coeficientes que individualizem tanto as cargas elétricas individuais quanto suas associações, para que, então, o mecanismo de reconhecimento proposto possa ser aplicado em ocasiões mais próximas das condições reais que são encontradas nos sistemas de distribuição de energia elétrica.

Para realizar um teste com o intuito de verificar a possibilidade citada, foram selecionadas três outras cargas elétricas não-lineares cujas formas de onda de corrente foram coletadas. Essas cargas elétricas foram um monitor de cristal líquido de 19" e de 70 W de consumo, uma lâmpada fluorescente compacta de 20 W e um osciloscópio analógico de 50 W. As formas de onda desses equipamentos são rerepresentadas na Figura 62, em companhia das formas de onda de corrente produzidas por todas as suas possíveis associações.

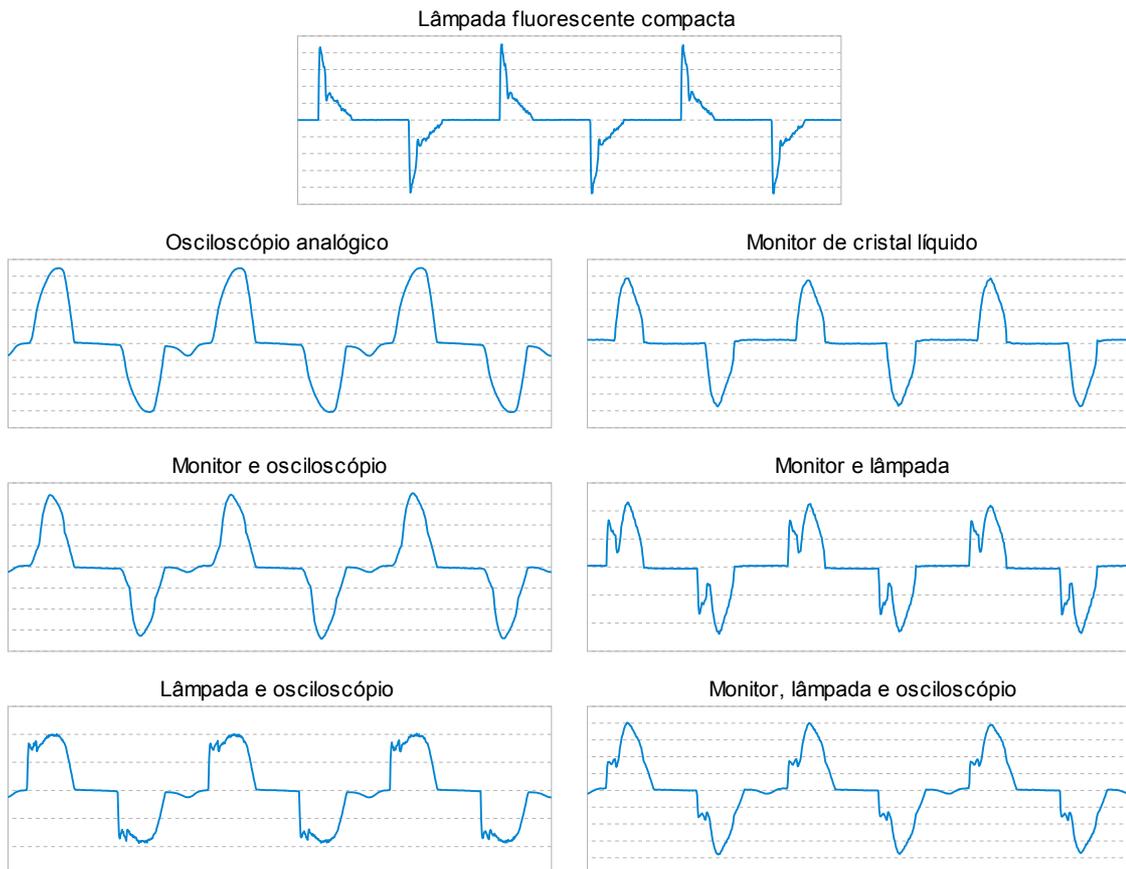


Figura 62 - Formas de onda de corrente originadas por associações entre um osciloscópio analógico, um monitor de cristal líquido e uma lâmpada fluorescente compacta.

Como se pode observar na Figura 62, cada uma das associações entre as três cargas iniciais produziu uma forma de onda que mantém informações das formas de onda dos equipamentos que a geraram, e mesmo a associação dos três equipamentos resulta em uma forma de onda de corrente que contém peculiaridades de cada um dos equipamentos associados. Esse fato reforçou a possibilidade de que o sistema proposto fosse capaz de identificar cada um dos equipamentos presentes em cada uma das associações analisadas.

Para testar essa possível capacidade do sistema proposto, cada uma das curvas presentes na Figura 62 foi decomposta em 5 níveis através da família de *wavelets* Haar. Os

resultados dessa decomposição foram utilizados para se gerar vetores de características para uma rede neural artificial treinada pelo algoritmo de *Levenberg-Marquard* e contendo 6 nós de entrada, 7 *perceptrons* na camada de processamento e 3 *perceptrons* na camada de saída. Cada uma das associações e mesmo os casos das presenças individuais das cargas elétricas foi considerada como sendo uma classe possível para os vetores de características, o que resultou em um total de 7 classes. Os vetores de características dessas classes são apresentados na Tabela 17, juntamente com os respectivos valores eficazes das ondas de corrente dos quais foram obtidos.

Tabela 17 - Vetores de características obtidos para as associações de cargas elétricas analisadas.

Equipamentos	A5	D1	D2	D3	D4	D5	Valor eficaz
Lâmpada	0,55257	0,00991	0,00798	0,15350	0,14989	0,12616	0,23 A
Osciloscópio	0,81568	0,00068	0,00253	0,01063	0,04876	0,12172	0,49 A
Monitor	0,52133	0,00219	0,00745	0,02130	0,03331	0,41443	0,36 A
Osciloscópio + Lâmpada	0,87390	0,00171	0,00187	0,03053	0,03338	0,05861	0,58 A
Monitor + Lâmpada	0,72988	0,00421	0,00903	0,03808	0,04389	0,17492	0,49 A
Monitor + Osciloscópio	0,71365	0,00082	0,00322	0,00982	0,03008	0,24242	0,77 A
Todos os três	0,0407	0,00099	0,00273	0,01806	0,02162	0,15253	0,89 A

Como era esperado, a Tabela 17 permite constatar que cada um dos vetores de características obtidos é diferente dos demais, o que condiz com o fato de cada um deles estar associado a uma forma de onda diferente das demais, assim como pode ser notado através da Figura 62.

Após serem associados a identificadores únicos para suas classes, os vetores de características foram utilizados para se treinar a rede neural artificial proposta para o teste em questão. Os identificadores das 7 classes presentes nesse treinamento são exibidos na Tabela

18, que também traz os respectivos resultados de saída da rede neural após ser treinada e ao receber em sua entrada cada um dos próprios vetores de treinamento que foram utilizados.

Tabela 18 - Vetores de identificação de classe e os vetores de saída da rede para cada um dos vetores de características relacionados aos vetores de identificação apresentados.

Equipamentos	Identificadores de classes			Resultados da rede neural		
	Saída 1	Saída 2	Saída 3	Saída 1	Saída 2	Saída 3
Lâmpada	0,100	0,100	0,900	0,099	0,000	0,900
Osciloscópio	0,100	0,900	0,100	0,099	0,900	0,100
Monitor	0,100	0,900	0,900	0,099	0,899	0,900
Osciloscópio + Lâmpada	0,900	0,100	0,100	0,900	0,100	0,100
Monitor + Lâmpada	0,900	0,100	0,900	0,899	0,100	0,900
Monitor + Osciloscópio	0,900	0,900	0,100	0,899	0,900	0,100
Todos os três	0,900	0,900	0,900	0,899	0,900	0,900

Se forem considerados os intervalos de 0,0 a 0,3 para representar o valor 0,1, e de 0,7 a 1,0 como sendo um intervalo aceitável dentro do qual os valores de saída da rede neural sejam considerados como sendo 0,9, então os resultados apresentados pela Tabela 18 deixam claro que a rede proposta foi corretamente treinada, uma vez que os vetores de saída da rede neural para cada um dos vetores de características está muito próximo dos vetores-alvo que foram estipulados para identificar as respectivas classes dos vetores de características utilizados no treinamento. Dessa forma, pode-se concluir que para o caso dos equipamentos utilizados para se realizar o teste em questão, o sistema proposto é capaz de reconhecer a classe dos equipamentos presentes em cada uma das formas de onda resultantes das associações possíveis entre as cargas estudadas. Esse resultado sugere que qualquer associação entre cargas não-lineares de classes distintas possam ser reconhecidas pelo sistema proposto, desde que as formas de onda das classes envolvidas sejam diferentes umas das outras.

Uma forma de estender o teste realizado e então tentar reforçar a conclusão obtida foi adicionar um quarto equipamento aos três já analisados e obter todas as formas de onda das 15 associações possíveis entre os elementos do novo grupo formado para, então, tentar treinar o sistema com as novas associações obtidas. O equipamento selecionado para compor o novo grupo de testes foi o gerador de funções cuja forma de onda é também apresentada na Figura 60. Para esse novo teste foi utilizada a decomposição através da família Haar em 7 níveis de detalhes, o que conferiu à rede utiliza 8 nós de entrada (7 detalhes e 1 aproximação). Foram escolhidos 30 *perceptrons* para a camada intermediária e 4 para a camada de saída. Os quatro nós na camada de saída são devido às 15 variações necessárias para se representar as 15 classes originadas pelas associações entre as quatro cargas elétricas. Assim como o apresentado na Tabela 17, a Tabela 19 contém os vetores de características das classes obtidas.

Tabela 19 - Vetores de características obtidos para as associações entre um monitor (M), uma lâmpada fluorescente compacta (L), um gerador de funções (G) e um osciloscópio analógico (O).

Equip.	A7	D1	D2	D3	D4	D5	D6	D7	Valor efic.
L	0,24371	0,00991	0,00798	0,15350	0,14989	0,12616	0,13698	0,17188	0,23 A
O	0,42314	0,00068	0,00253	0,01063	0,04876	0,12172	0,38675	0,00579	0,49 A
M	0,24491	0,00219	0,00745	0,02130	0,03331	0,41443	0,25191	0,02451	0,36 A
L+O	0,49591	0,00171	0,00187	0,03053	0,03338	0,05861	0,36852	0,00947	0,58 A
M+L	0,34708	0,00421	0,00903	0,03808	0,04389	0,17492	0,28707	0,09573	0,49 A
M+O	0,36639	0,00082	0,00322	0,00982	0,03008	0,24242	0,34654	0,00072	0,77 A
M+L+O	0,42997	0,00099	0,00273	0,01806	0,02162	0,15253	0,35328	0,02082	0,89 A
G	0,40938	0,00072	0,00289	0,00860	0,02906	0,06207	0,37302	0,11426	0,12 A
G+L	0,44753	0,01366	0,02421	0,06995	0,08093	0,05102	0,28300	0,02970	0,28 A
G+O	0,43597	0,00061	0,00238	0,00983	0,04275	0,09719	0,39374	0,01753	0,56 A
G+O+L	0,50461	0,00269	0,00367	0,01923	0,02693	0,05113	0,39150	0,00024	0,68 A
G+M+O	0,39038	0,00078	0,00301	0,00945	0,02744	0,20668	0,36205	0,00020	0,86 A
G+M+O+L	0,45056	0,00092	0,00229	0,01483	0,02151	0,14262	0,36050	0,00677	0,96 A
M+G	0,31200	0,00178	0,00666	0,01502	0,02629	0,33795	0,29794	0,00236	0,45 A
M+L+G	0,40630	0,00215	0,00483	0,03289	0,03249	0,15456	0,32399	0,04279	0,57 A

Os identificadores de classe dos 15 vetores de características são apresentados na Tabela 20, onde também são listados os resultados de saída da rede neural após ser treinada e ao receber em sua entrada cada um dos vetores de treinamento.

Tabela 20 - Vetores de identificação de classe e os vetores de saída da rede para cada um dos 15 vetores de características relacionados às associações entre um monitor (M), uma lâmpada fluorescente compacta (L), um gerador de funções (G) e um osciloscópio (O).

Equipamentos	Identificadores de classes				Resultados da rede neural			
	Saída 1	Saída 2	Saída 3	Saída 4	Saída 1	Saída 2	Saída 3	Saída 4
L	0,100	0,100	0,100	0,900	0,000	0,192	0,097	0,896
O	0,100	0,100	0,900	0,100	0,110	0,127	0,896	0,037
M	0,100	0,100	0,900	0,900	0,097	0,102	0,899	0,899
L+O	0,100	0,900	0,100	0,100	0,100	0,878	0,098	0,114
M+L	0,100	0,900	0,100	0,900	0,100	0,896	0,095	0,905
M+O	0,100	0,900	0,900	0,100	0,102	0,890	0,895	0,078
M+L+O	0,100	0,900	0,900	0,900	0,127	0,995	0,905	0,867
G	0,900	0,100	0,100	0,100	1,000	0,003	0,101	0,095
G+L	0,900	0,100	0,100	0,900	0,902	0,102	0,000	0,900
G+O	0,900	0,100	0,900	0,100	0,893	0,034	0,998	0,125
G+O+L	0,900	0,100	0,900	0,900	1,000	0,108	0,895	0,887
G+M+O	0,900	0,900	0,100	0,100	0,888	0,871	0,104	0,120
G+M+O+L	0,900	0,900	0,100	0,900	0,879	0,988	0,098	0,922
M+G	0,900	0,900	0,900	0,100	0,892	0,915	0,908	0,105
M+L+G	0,900	0,900	0,900	0,900	0,895	0,965	0,897	0,989

Se também forem considerados os intervalos de 0,0 a 0,3 e de 0,7 a 1,0 para representar os valores 0,1 e 0,9, respectivamente, então a capacidade do sistema proposto em identificar a presença de cargas elétricas monofásicas não-lineares em sinais de corrente oriundas das associações entre essas cargas é reiterada, e assim se pode concluir que o sistema proposto é capaz de reconhecer tanto a presença individual de cargas elétricas monofásicas não-lineares quanto a presença de associações entre elas.

8.5 Comparação com um Sistema Baseado na Transformada de Fourier

Com o objetivo de comparar a eficiência do sistema de reconhecimento proposto com um sistema baseado na extração de características das formas de onda de corrente através da transformada de Fourier, foram desenvolvidos dois outros testes de reconhecimento. A referência para esses novos testes foi o teste 5 presente na Tabela 16, com o qual foi testado o reconhecimento individual de lâmpadas fluorescente compactas, de aparelhos televisores e de aparelhos de som, cujas formas de onda foram decompostas em representações de 7 níveis de detalhes pela família de funções Daubechies 4. Cada um destes novos testes utilizou a mesma implementação da rede neural artificial e o mesmo algoritmo de treinamento de *Levenberg-Marquardt* empregues no teste de referência, sendo que as únicas diferenças entre estes e aquele foram a quantidade de nós de entrada da rede e os valores dos vetores de características, uma vez que estes foram obtidos através da transformada rápida de Fourier. Até mesmo a ordem dos vetores de características e os valores dos vetores de identificação das classes foram preservados nos testes complementares.

Para se obter o vetor de características através da decomposição via transformada de Fourier, foi decomposto 1 ciclo de cada uma das formas de onda analisadas, o que representou um total de 256 pontos por forma de onda e uma resolução em frequência de 60 Hz, que é a mesma utilizada por grande parte das decomposições executadas por estudos relacionados aos sistemas de distribuição de energia elétrica. Com essas características, cada decomposição gerou um espectro com 128 raias de frequência, o que possibilitou analisar, através de um

passo de 60 Hz, as frequências entre 0 Hz e 7.620 Hz. Para o primeiro teste, o espectro de 128 frequências foi dividido em 8 grupos de 16 frequências cada, enquanto para o segundo foram utilizados 16 grupos de 8 raias cada.



Figura 63- Divisão de um espectro de 128 frequências em N grupos de mesmo tamanho e que não se sobrepõem.

Uma vez que o teorema de *Parseval*, apresentado na seção 4.1, garante que a energia de um sinal é preservada na representação obtida pela decomposição de Fourier, para cada um dos grupos criados foi levantada sua energia associada como sendo a soma dos quadrados da amplitude de cada um das raias presentes no grupo. Por fim, para cada grupo existente foi obtido o percentual de sua energia perante a energia total da representação, a qual é dada pela soma das energias presentes em cada um dos grupos criados para um dado teste. Dessa forma, para cada um dos testes foram gerados vetores de características que contiveram um número de elementos igual ao número de grupos criados e onde cada valor desses elementos foi obtido como sendo o percentual que a energia do respectivo grupo representou perante a energia total da decomposição. Como resultado do procedimento adotado, cada vetor de características foi obtido como sendo uma distribuição da energia do sinal original ao longo dos grupos de frequências criados, o que é exatamente a mesma idéia utilizada para o sistema de reconhecimento baseado na decomposição através da transformada *wavelet*, mas com a diferença de que para tal sistema, a distribuição da energia da decomposição é tomada em seus sinais de aproximação e de detalhes. A equação (8.5.1) apresenta o cálculo da energia de um grupo g_i em função das amplitudes das frequências $\hat{f}[j]$ que o mesmo contempla,

enquanto a equação (8.5.2) exibe a forma como é obtida a contribuição \bar{g}_i que a energia de cada grupo g_i exerce perante a energia total da decomposição. Já a equação (8.5.3) apresenta a forma geral dos vetores de características obtidos.

$$g_i = \sum_{j=a}^b |\hat{f}[j]|^2 \quad (8.5.1)$$

$$\bar{g}_i = \frac{g_i}{g_{total}} \quad (8.5.2)$$

$$\bar{\mathbf{v}} = [\bar{g}_1, \bar{g}_2, \dots, \bar{g}_N] \quad (8.5.3)$$

Os resultados de cada um dos dois testes efetuados são apresentados na Tabela 21, que também reinterpreta os resultados do teste de referência.

Tabela 21 - Resultados dos testes de reconhecimento baseados na extração de características através da transformada rápida de Fourier.

	Referência	Teste 1	Teste 2
Tipo de decomposição	Daubechies 4	Fourier	Fourier
Níveis de decomposição	7 níveis	8 grupos	16 grupos
Número de camadas da rede	3	3	3
Estrutura da rede	8-6-3	8-6-3	16-6-3
Épocas de treinamento	25	600	300
Erro de treinamento	0,000626	0,955030	0,002731
Vetores de validação	20	20	20
Vetores de treinamento	9	9	9
Reconhecimentos corretos	20	8	18
Falsos reconhecimentos	0	0	0
Resposta indefinidas	0	12	2
Eficiência do sistema	100 %	40,00%	90,00%

Como pode ser observado pelos resultados apresentados na Tabela 21, os testes realizados com a extração de características pela transformada rápida de Fourier obtiveram resultados inferiores aos inerentes ao teste que fez uso da decomposição *wavelet* em multirresolução. Enquanto o teste de referência obteve 100 % de reconhecimentos corretos, mesmo tendo sido utilizadas apenas 25 épocas de treinamento, o teste utilizando 8 grupos de 16 frequências cada obteve um percentual de reconhecimentos corretos que nem mesmo chegou a 50 %, o que é ainda bem inferior aos 90 % de acertos obtidos pelo mesmo sistema com 16 grupos de 8 frequências cada e 300 épocas de treinamento. Dessa forma, o que se pode concluir dos resultados obtidos com os testes 1 e 2 é que a rede neural artificial foi treinada com relativa dificuldade ao se utilizar os vetores de características obtidos a partir da transformada de Fourier, pois enquanto o teste de referência utilizou apenas 25 épocas de treinamento, mesmo uma quantidade de 600 épocas de treinamento, o que é 24 vezes a quantidade de épocas utilizada no teste de referência, não foi sequer suficiente para treinar a rede que teve a mesma estrutura da rede utilizada no teste de referência mas que utilizou vetores de características formados pela distribuição de energia do sinal original em 8 grupos de frequências, em oposição ao vetor de características obtidos pela decomposição *wavelet* em multirresolução. Vale ressaltar que os resultados do teste 1 demonstram que a rede neural, mesmo com 600 épocas de treinamento, nem chegou a ser treinada corretamente, pois foram obtidas 12 resposta indefinidas perante um total de 20 tentativas.

A Tabela 22 traz alguns vetores de características obtidos tanto pela decomposição em multirresolução quanto pela distribuição de energias em 8 grupos do espectro de amplitudes dos sinais analisados.

Tabela 22 - Vetores de características obtidos tanto através da transformada de Fourier quanto da decomposição em multirresolução *wavelet* para uma lâmpada fluorescente (L), um televisor de 17 polegadas (T) e um aparelho de som (S).

Equipamentos		Elementos dos vetores de características							
		1	2	3	4	5	6	7	8
<i>Wavelet</i> Daubechies 4, com 7 níveis	L	0,38222	0,00927	0,04127	0,06953	0,17567	0,06518	0,24130	0,01556
	T	0,26510	0,00021	0,00093	0,00517	0,02978	0,34956	0,19440	0,15485
	S	0,20913	0,00016	0,00009	0,00094	0,01004	0,02954	0,17145	0,57865
Fourier, com 8 grupos	L	0,91294	0,05371	0,02280	0,00514	0,00339	0,00117	0,00045	0,00039
	T	0,99661	0,00268	0,00040	0,00016	0,00006	0,00003	0,00003	0,00003
	S	0,99952	0,00034	0,00004	0,00004	0,00002	0,00001	0,00001	0,00002

Ao se analisar a Tabela 22, é possível notar que devido à forma como foram definidos, os vetores de características obtidos ao se dividir o espectro de amplitudes em 8 grupos geraram distribuições de energia concentradas na faixa de frequências entre as raias espectrais de 0 Hz e 900 Hz, o que praticamente dificultou, de uma certa forma, individualizar cada uma das classes às quais pertencem os equipamentos analisados. Essa situação pode ser praticamente estendida às demais classes de cargas não-lineares, pois grande parte dos sinais de corrente dessas cargas contém frequências cujas amplitudes mais elevadas estão próximas da faixa compreendida pelas primeiras 17 frequências harmônicas [45].

Uma forma possível de tentar aprimorar o modelo de reconhecimento baseado na transformada de Fourier e entender o problema levantado é pensar que ao se elevar o número de grupos com os quais dividir um mesmo espectro de amplitudes, a quantidade de elementos em cada grupo pode ser diminuída até que se obtenha um número de grupos equivalente ao número de raias no espectro e, assim, cada grupo conterá somente uma raia espectral. Essa condição deixa claro que sistema baseado na transformação de Fourier teria sua eficiência

umentada, mas, no entanto, algumas considerações devem ser levantadas, pois embora um número cada vez maior de grupos possa melhor representar a distribuição de energia do sinal original ao longo de suas frequências, esse mesmo acréscimo de elementos nos vetores de características pode agregar complexidades no processo de treinamento da rede neural [64], levando-se, assim, a um comprometimento que deve haver entre o número de características e o quão bem cada uma delas representa os detalhes que podem individualizar os sinais decompostos. Uma segunda consideração importante quanto ao uso da transformada de Fourier como método de extração de características para sistemas de reconhecimento de cargas elétricas não-lineares é a questão das limitações existentes na representação de sinais que essa transformação pode gerar. Essas limitações foram levantadas na seção 4.6, sendo que algumas delas, como o problema do vazamento de frequências e a existência de descontinuidades nos sinais de corrente das cargas elétricas não-lineares, podem, por exemplo, gerar resultados nos espectros de amplitudes que possibilitem, entre outras coisas, com que os vetores de características baseados nesses espectros contenham informações que não sejam necessariamente pertencentes à carga elétrica analisada. No caso da presença de ondas inter-harmônicas nos sinais de corrente analisados, o vazamento de frequências resultante pode fazer com que o sinal que contenha tal vazamento espectral resulte em um espectro de amplitudes que se assemelhe a um outro espectro que possa ser pertencente a uma segunda classe de sinais (ou cargas elétricas). Embora uma solução para esse caso possa ser utilizar uma resolução em frequência de 1 Hz ou mesmo agregar informações do espectro de fases aos vetores de características da classe, ambas essas abordagens certamente agregam mais elementos ao vetor de característica e, assim, novamente se cai no comprometimento que deve haver entre o número de características a utilizar nos vetores de características e o quão

bem cada uma dessas características concentram detalhes que podem individualizar os sinais que representem.

É certo que possam existir técnicas de reconhecimento de cargas elétricas monofásicas não-lineares nas quais os vetores de características sejam provenientes dos espectros de amplitude e fases e que mesmo nessas condições, tais técnicas possam apresentar resultados melhores que os obtidos para os testes de comparação efetuados. No entanto, frente aos dados presentes na Tabela 22 e às limitações da transformada de Fourier para representar discontinuidades e frequências inter-harmônicas, pode-se também concluir que para o caso dos sistemas de reconhecimento em que são levantadas as distribuições de energia dos sinais de corrente de cargas elétricas não-lineares, a proposta de um sistema baseado na decomposição *wavelet* em multirresolução se configura como uma opção que possibilita obter desempenhos superiores aos obtidos por sistemas similares que utilizem a transformada de Fourier como mecanismo de extração de características.

CAPÍTULO 9

Conclusão

A contínua agregação de novas tecnologias e aparatos ao cotidiano tem feito com que os sistemas de distribuição de energia elétrica hospedem uma quantidade cada vez maior de dispositivos baseados em circuitos eletrônicos das mais variadas formas e responsáveis pela execução das mais variadas atividades. A presença desses dispositivos, que algumas literaturas denominam de cargas elétricas não-lineares, tem como consequência marcante a injeção de distorções harmônicas que, ao serem responsabilizadas pela degradação de uma relevante parcela da potência elétrica disponibilizadas pelos ramais de distribuição, direcionaram o tema deste trabalho para junto dos estudos da presença dessas cargas nos sistemas de distribuição aos quais são conectadas.

Dessa forma, o objetivo deste trabalho foi o de propor uma metodologia de reconhecimento da presença de cargas elétricas monofásicas não-lineares em seções dos sistemas de distribuição. Essa proposta foi baseada na presunção de que a decomposição

wavelet em multirresolução, à qual a literatura atribui as notáveis capacidades de compactar e de representar, de forma completa, as características de sinais que contenham descontinuidades, o que é o caso das formas de onda de corrente geradas por cargas elétricas não-lineares, pudesse ser utilizada para se extrair características que individualizem as formas de onda de corrente de cada uma das classes dessas cargas elétricas. À transformação *wavelet* foi, então, adicionada uma rede neural artificial de múltiplas camadas para que se pudesse classificar e reconhecer os vetores de características obtidos a partir da decomposição em multirresolução dos sinais de corrente analisados. Esses vetores foram concebidos como sendo uma distribuição de energia do sinal original ao longo dos sinais de aproximação e de detalhes obtidos pela decomposição utilizada.

Para que o sistema pudesse ser testado e então possivelmente validado como uma proposta viável, foram levantadas e implementadas cada uma das técnicas relacionadas à transformação *wavelet* discreta, à transformação de Fourier, às redes neurais artificiais de múltiplas camadas e a um protótipo de equipamento de aquisição de dados, que foi utilizado para se amostrar os sinais de corrente através de uma taxa específica de 15.360 Hz, 16 *bits* por dado amostrado e de forma a sincronizar a coleta dos sinais de corrente amostrados com a passagem por zero do sinal de tensão que alimentou as cargas elétricas analisadas. Cada uma dessas implementações foi testada através de procedimentos específicos que acabaram por validá-las.

Para que as capacidades do sistema proposto fossem analisadas, foram realizados dois testes específicos, sendo que o primeiro teve como objetivo verificar a capacidade do sistema em reconhecer cargas individuais, enquanto o segundo buscou testar a capacidade do mesmo sistema em reconhecer formas de onda de corrente geradas por associações de cargas elétricas

não-lineares, o que se configura como uma situação mais próxima da situação real presente nos ramais de distribuição de energia elétrica.

Os resultados obtidos tanto para o primeiro quanto para o segundo testes foram satisfatórios e expressaram uma eficiência da metodologia que, para o caso das formas de onda analisadas, foi de pelo menos 95% e assim permitiu concluir que tanto o reconhecimento da presença individual de cargas elétricas monofásicas não-lineares quanto a de associações das mesmas é possível de ser efetuada pela metodologia proposta.

A fim de comparar o desempenho do sistema com um baseado na transformação de Fourier como mecanismo de extração de características das formas de onda de corrente, foi realizado um conjunto adicional de testes com os quais a idéia base e a estrutura do sistema proposto foram reproduzidas, sendo que a única alteração realizada foi a troca dos vetores de características iniciais por vetores originados da aplicação da transformada rápida de Fourier. Os resultados desses testes foram bem enfáticos, pois enquanto o teste de referência realizado com a decomposição em multirresolução obteve um percentual de reconhecimento da ordem de 100%, o teste de comparação inicialmente realizado com a mesma estrutura da rede neural e com as mesmas formas de onda do teste de referência, obteve um percentual que nem mesmo atingiu 50% de reconhecimentos corretos, o que sugeriu que a estrutura inicial desses testes fosse modificada de forma a comportar um vetor de características com o dobro de elementos dos vetores inicialmente utilizados e, assim, se obteve o percentual de reconhecimentos de 90 %, que ainda foi inferior ao de 100% obtido pelo teste de referência que utilizou vetores de características com a metade do número de elementos utilizados pelo teste complementar. Esses resultados permitiram concluir que para a metodologia proposta por este trabalho, um sistema de reconhecimento de cargas monofásicas não-lineares que faça

uso de um mecanismo de extração de características baseado na decomposição em multirresolução pode obter resultados mais expressivos que os provenientes do uso da transformada discreta de Fourier para o mesmo fim.

De uma forma geral, os resultados obtidos permitiram concluir que a metodologia proposta para se reconhecer a presença de cargas elétricas monofásicas não-lineares é viável e relativamente melhor que uma metodologia similar que utilize a transformada rápida de Fourier, podendo, então, ser aplicada em situações práticas.

9.1 Propostas de Trabalhos Futuros

Uma idéia de trabalho futuro que decorre com certa naturalidade da metodologia proposta por este trabalho pode ser a de se levantar um base de dados que contemple informações sobre as cargas elétricas não-lineares que se acredita serem preponderantes em determinadas áreas dos sistema de distribuição de energia elétrica. Essa base de dados poderia ser, então, utilizada para se aplicar o metodologia de forma prática.

Uma segunda idéia de trabalho futuro pode ser a de se estudar qual a implicação que a presença de cargas elétricas lineares agrega aos resultados da proposta apresentada. Essa questão se deve ao fato de que grande parte das cargas elétricas conectadas em ramais dos sistemas de distribuição consumirem potências elétricas maiores que as consumidas pelas

cargas não-lineares. Nessas condições, as formas de onda de corrente resultantes das associações possíveis entre cargas lineares e não-lineares podem possuir um formato predominantemente linear, uma vez que a corrente consumida pelas cargas elétricas lineares como refrigeradores, chuveiros e afins costuma conter uma amplitude maior que as formas de onda de corrente relacionadas a alguns equipamentos não-lineares como aparelhos televisores, aparelhos de som, dentre outros.

Uma terceira idéia de trabalho futuro pode ser a de implementar um sistema completo para se aplicar a metodologia proposta na prática. Esse possível sistema poderia ser compostos por dispositivos eletrônicos específicos para se coletar formas de onda de corrente em ramais de distribuição de energia elétrica e por um aplicativo de computador que fosse capaz de concentrar as informações coletadas e então inferisse sobre as cargas presentes em cada uma das formas de onda analisadas. Esse proposta poderia, então, ser utilizada para se levantar todo o perfil de cargas elétricas não-lineares em uma determinada região de interesse às ações de planejamento e manutenção dos sistemas de distribuição de energia elétrica.

Uma quarta vertente poderia ser a de estender, à classe de cargas trifásicas, a idéia apresentada pela metodologia proposta por estes trabalho.

Referências Bibliográficas

- [1] GUPTA, S.M. GEORG SIMON OHM AND OHM'S LAW. IEEE Transactions on Education, vol. 23, nº , p. 156-162, ago. 1980.
- [2] PROUT, H. G. A LIFE OF GEORGE WESTINGHOUSE. 1ª. ed. New York: Beard Books, 2001.
- [3] TESLA, N. MY INVENTIONS : THE AUTOBIOGRAPHY OF NIKOLA TESLA. 1ª. ed. Radford: Filiquarian Publishing, 2006.
- [4] AMPÈRE ET L'HISTOIRE DE L'ÉLECTRICITÉ. [S.L.]: Centre de Recherche en Histoire des Sciences et des Techniques . 2005. Disponível em <<http://www.ampere.cnrs.fr/>>. Acesso em: 12 de Setembro de 2008.
- [5] ELECTRICITY/HEAT STATISTICS. [S.L.]: IEA - International Energy Agency . 2008. Disponível em <<http://www.iea.org/Textbase/stats/prodresult.asp?PRODUCT=Electricity/Heat>>. Acesso em: 11 de Outubro de 2008.
- [6] BRASIL. Empresa de Pesquisa Energética BOLETIM ANUAL DE CONJUNTURA ECONÔMICA REGIONAL E SETORIAL (REF.: 2006). Rio de Janeiro. 2008. Disponível em <<http://www.epe.gov.br/Lists/Mercado/DispForm.aspx?ID=9>>. Acesso em: 23 de Setembro de 2008.
- [7] ELECTRIC POWER SYSTEMS RESEARCH. College Station: Elsevier, 1978 - . ISSN: 0378-7796.
- [8] IEEE POWER & ENERGY MAGAZINE. New Jersey: Institute of Electrical and Electronics Engineers, 2006 - . ISSN: 1540-7977.
- [9] ELECTRIC POWER RESEARCH INSTITUTE. [S.L.]: Electric Power Research Institute . 2008. Disponível em <<http://www.epri.com/>>. Acesso em: 02 de Julho de 2008.
- [10] BOLLEN, M.H.J. WHAT IS POWER QUALITY?. Electric Power Systems Research, vol. 66, nº 1, p. 5-14, jul. 2003.
- [11] BRASIL. Agência Nacional de Energia Elétrica. RESOLUÇÃO ANEEL Nº024, DE 27 DE JANEIRO DE 2000. Estabelece as disposições relativas à continuidade da distribuição de energia elétrica às unidades consumidoras. Diário Oficial da União, Brasília, 25 jan. 2006, Seção 1, p. 54, v. 143, nº 18.
- [12] BRASIL. Agência Nacional de Energia Elétrica. RESOLUÇÃO ANEEL Nº505, DE 26 NOVEMBRO DE 2001. Estabelece de forma atualizada e consolidada, as disposições relativas à conformidade dos níveis e tensão de energia elétrica em regime permanente. Diário Oficial da União, Brasília, 16 de jan. 2004, Seção 1, p. 43, v. 141, nº 11.
- [13] IEEE Standard 519-1992, RECOMMENDED PRACTICES AND REQUIREMENTS FOR HARMONIC CONTROL IN ELECTRIC POWERSYSTEMS, New York: Institute of Electrical and Electronics Engineers, oct. 1993.

- [14] GUTMANN, R.J.; CAMNITZ, L.H. DETERMINATION OF HARMONIC LEVELS IN NONLINEAR NETWORKS BY COMPUTER SIMULATION. *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-24, n° 1, p. 50-52, feb. 1982.
- [15] CAVALLINI, A. et al EVALUATION OF HARMONIC LEVELS IN ELECTRICAL NETWORKS BY STATISTICAL INDEXES. *IEEE Transactions on Industry Applications*, vol. 30, n° 4, p. 1116-1125, jul. 1994.
- [16] WAGNER, V.E. et al EFFECTS OF HARMONICS ON EQUIPMENT. *IEEE Transactions on Power Delivery*, vol. 8, n° 2, p. 672-680, apr. 1993.
- [17] WANG, Y.; O'CONNELL, R.M.; BROWNFIELD G. MODELING AND PREDICTION OF DISTRIBUTION SYSTEM VOLTAGE DISTORTION CAUSED BY NONLINEAR RESIDENTIAL LOADS. *IEEE Transactions on Power Delivery*, vol. 16, n° 4, p. 744-751, oct. 2001.
- [18] ANASTASIADOU, D.; ANTANAKOPOULOS, T. MEASUREMENTS-BASED METHOD FOR IMPEDANCE CHARACTERIZATION OF RESIDENTIAL LOADS. *In: IEEE Instrumentation and Measurement Technology Conference*, 21, 2004. **Proceedings...** Como: 2004, p. 669-674.
- [19] HO, C.M.; LEE, W.K.; HUNG, Y.S. SIGNATURE REPRESENTATION OF UNDERGROUND CABLES AND ITS APPLICATIONS TO CABLE FAULT DIAGNOSIS. *In: IEEE International Conference on Advances in Power System Control, Operation and Management*, 2, 1993. **Proceedings...** : 1993, p. 861-865.
- [20] LAM, H.Y.; FUNG, G.S.; LEE, W.K. A NOVEL METHOD TO CONSTRUCT TAXONOMY OF ELECTRICAL APPLIANCES BASED ON LOAD SIGNATURES. *IEEE Transactions on Consumer Electronics*, vol. 53, n° 2, p. 653-660, may 2007.
- [21] HART, G.W. NONINTRUSIVE APPLIANCE LOAD MONITORING. *Proceedings of the IEEE*, vol. 80, n° 22, p. 1870-1891, dec. 1992.
- [22] COLE, A.I.; ALBICKI, A. ALGORITHM FOR NONINTRUSIVE IDENTIFICATION OF RESIDENTIAL APPLIANCES. *In: IEEE International Symposium on Circuits and Systems*, 1998. **Proceedings...** Monterey: 1998, p. 338-341.
- [23] CALABRESE, M; DI LECCE, V.; PIURI, V. ANN RESIDENTIAL LOAD CLASSIFIER FOR INTELLIGENT DSM SYSTEM. *In: IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2007. **Proceedings...** Ostuni: 2007, p. 33-38.
- [24] SULTANEM, F. USING APPLIANCE SIGNATURES FOR MONITORING RESIDENTIAL LOADS AT METER PANEL LEVEL. , vol. 6, n° 4, p. 1380-1385, oct. 1991.
- [25] VERDU, S.B. et al CHARACTERIZATION AND IDENTIFICATION OF ELECTRICAL CUSTOMERS THROUGH THE USE OF SELF-ORGANIZING MAPS AND DAILY LOAD PARAMETERS. *In: IEEE Power Systems Conference and Exposition*, 2004. **Proceedings...** New York: , p. 899-906.
- [26] BARANSKI, M.; VOSS, J. DETECTING PATTERNS OF APPLIANCES FROM TOTAL LOAD DATA USING A DYNAMIC PROGRAMMING APPROACH. *In: IEEE International Conference on Data Mining*, 4, 2004. **Proceedings...** Brighton: 2004, p. 327-330.
- [27] WALKER, C.F.; POKOSKI, J.L. RESIDENTIAL LOAD SHAPE MODELLING BASED ON CUSTOMER BEHAVIOR. *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-104, n° 7, p. 1703-1711, jul. 1985.

- [28] UMEH, K.C.; MOHAMED, A.; MOHAMED, R. DETERMINING HARMONIC CHARACTERISTICS OF TYPICAL SINGLE PHASE NONLINEAR LOADS. *In: Student Conference on Research and Development, 2003. Proceedings...* Putrajaya: 2003, p. 413-419.
- [29] UMEH, K.C.; MOHAMED, A.; MOHAMED, R. COMPARING THE HARMONIC CHARACTERISTICS OF TYPICAL SINGLE-PHASE NONLINEAR LOADS. *In: National Power Engineering Conference, 2003. Proceedings...* Bangi: 2003, p. 383-387.
- [30] UMEH, K.C. et al CHARACTERIZING NONLINEAR LOAD HARMONICS USING FRACTAL ANALYSIS. *In: IEEE International Symposium on Circuits and Systems, 2004. Proceedings...* Vancouver: 2004, p. V-932-935.
- [31] UMEH, K.; MOHAMED, A. INTELLIGENT SYSTEM FOR IDENTIFICATION OF HARMONICS ORIGINATING FROM SINGLE PHASE NONLINEAR LOADS. *In: IEEE SoutheastCon, 2005. Proceedings...* Fort Lauderdale: 2005, p. 137-142.
- [32] UMEH, K.C; MOHAMED, A. A RULE-BASED EXPERT SYSTEM FOR HARMONIC LOAD RECOGNITION. *In: National Power and Energy Conference, 2004. Proceedings...* Kuala Lumpur: 2004, p. 19-24.
- [33] SRINIVASAN, D.; NG, W.S; LIEW, A.C. NEURAL-NETWORK-BASED SIGNATURE RECOGNITION FOR HARMONIC SOURCE IDENTIFICATION. *IEEE Transactions on Power Delivery*, vol. 21, n° 1, p. 398-405, jan. 2006.
- [34] SHARMA, V.K.; MOINUDDIN DOJA, M.N.; IBRAHEEM KHAN, M.A. POWER QUALITY ASSESSMENT AND HARMONIC COMPARISON OF TYPICAL NONLINEAR ELECTRONIC LOADS. *In: IEEE International Conference on Industrial Technology, 2000. Proceedings...* Goa: 2000, p. 729-734.
- [35] FAURI, M. HARMONIC MODELLING OF NON-LINEAR LOAD BY MEANS OF CROSSED FREQUENCY ADMITTANCE MATRIX. *IEEE Transactions on Power Systems*, vol. 12, n° 4, p. 1632-1638, nov. 1997.
- [36] LEEB, S.B.; KIRTKEY, J.L.J. A MULTISCALE TRANSIENT EVENT DETECTOR FOR NONINTRUSIVE LOADMONITORING. *In: International Conference on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings...* Maui: nov. 1993, p. 354-359.
- [37] YANG, H.; CHANG, H.; LIN, C. DESIGN A NEURAL NETWORK FOR FEATURES SELECTION IN NON-INTRUSIVE MONITORING OF INDUSTRIAL ELECTRICAL LOADS. *In: International Conference on Computer Supported Cooperative Work in Design, 11, 2007. Proceedings...* Melbourne: apr. 2007, p. 1022-1027.
- [38] MARQUES, D.Z. et al . *In: Transmission and Distribution Conference and Exposition: Latin America, 2004. Proceedings...* São Paulo: nov. 1994, p. 908-913.
- [39] TSEKOURAS, G.J.; HATZIARGYRIOU, N.D.; DIALYNAS, E.N. TWO-STAGE PATTERN RECOGNITION OF LOAD CURVES FOR CLASSIFICATION OF ELECTRICITY CUSTOMERS. *IEEE Transactions on Power Systems*, vol. 22, n° 3, p. 1120-1128, aug. 2007.
- [40] ORTEGA, J.M.M. et al STATE ESTIMATION APPROACH TO HARMONIC POLLUTING LOAD CHARACTERIZATION IN DISTRIBUTION SYSTEMS. , vol. 20, n° 2, p. 765-772, may 2005.

- [41] LAUGHMAN, C. et al POWER SIGNATURE ANALYSIS. IEEE Power and Energy Magazine, vol. 1, n° 2, p. 56-63, mar. 2003.
- [42] IEEE Task Force on Harmonics Modeling and Simulation MODELING AND SIMULATION OF THE PROPAGATION OF HARMONICS IN ELETRIC POWER NETWORKS – PART I: CONCEPTS, MODELS, AND SIMULATION TECHNIQUES. IEEE Transactions on Power Delivery, vol. 11, n° 1, p. 452-465, jan. 1996.
- [43] OPPENHEIM, A.V.; SCHAFER, R.W.; BUCK, J.R. DISCRETE-TIME SIGNAL PROCESSING. 2ª. ed. New Jersey: Prentice Hall, jan. 1999.
- [44] COCHRAN, W.T et al WHAT IS THE FAST FOURIER TRANSFORM?. Proceedings of the IEEE, vol. 55, n° 10, p. 1664-1674, oct. 1967.
- [45] PIRES, I.A. CARACTERIZAÇÃO DE HARMÔNICOS CAUSADOS POR EQUIPAMENTOS ELETRO-ELETRÔNICOS RESIDENCIAIS E COMERCIAIS NO SISTEMA DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA. Dissertação de mestrado em Engenharia Elétrica - Universidade Federal de Minas Gerais, Belo Horizonte, 2006.
- [46] GOSWAMI, J.C.; CHAN, A.K. FUNDAMENTALS OF WAVELETS: THEORY, ALGORITHMS, AND APPLICATIONS. 1ª. ed. New York: Wiley, 1999.
- [47] STARK; H.G. WAVELETS AND SIGNAL PROCESSING: AN APPLICATION-BASED INTRODUCTION. 1ª. ed. Leipzig: Springer, 2005.
- [48] MALLAT, S. A WAVELET TOUR OF SIGNAL PROCESSING. 2ª. ed. San Diego: Academic Press, 1999.
- [49] CHAN, W.L.; SO, A.T.P.; LAI, L.L. HARMONICS LOAD SIGNATURE RECOGNITION BY WAVELETS TRANSFORMS. *In: IEEE International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings...* London: 2000, p. 666 - 671.
- [50] ROBERTS, G.; SEDRA, A. SPICE. 2ª. ed. New York: Oxford University Press, 1999.
- [51] SEDRA, A.S.; SMITH, K.C. MICROELETRÔNICA. 4ª. ed. São Paulo: Makron Books, 2000.
- [52] KUSKO, A.; THOMPSON, M.T. POWER QUALITY IN ELECTRICAL SYSTEMS. 1ª. ed. New York: McGraw-Hill, 2007.
- [53] TESTA, A. et al INTERHARMONICS: THEORY AND MODELING. IEEE Transactions on Power Delivery, vol. 22, n° 4, p. 2335-2348, oct. 2007.
- [54] GUNTHER, E.W. INTERHARMONICS IN POWER SYSTEMS. *In: IEEE Power Engineering Society Summer Meeting, 2001. Proceedings...* Vancouver: 2001, p. 813-817.
- [55] GAMA, P. H. R. P.; OLIVEIRA, A. CONSERVAÇÃO DE ENERGIA E SUA RELAÇÃO COM A QUALIDADE DE ENERGIA ELÉTRICA. *In: XV SNPTEE – Seminário Nacional de Produção e Transmissão de Energia Elétrica. Anais...* Foz do Iguaçu: 1999, p. ?.
- [56] JUN, Z.; MASWOOD, A.I TYPICAL INDUSTRIAL LV SYSTEM FOR HARMONIC PROPAGATION STUDY UNDERBALANCED AND UNBALANCED CONDITIONS USING PSPICE MODELS. *In: IEEE International Conference on Power Electronics and Drive Systems, 1999 . Proceedings...* Hong Kong: 1999, p. 1147-1152.

- [57] RIBEIRO, P.F. et all, TUTORIAL ON HARMONICS MODELING AND SIMULATION. *In: IEEE Power Engineering Society Winter Meeting, 1998. Anais...* Tampa: 1998. CD-ROM.
- [58] MARTINEZ, J.A. POWER QUALITY ANALYSIS USING ELECTROMAGNETIC TRANSIENTS PROGRAMS. *In: International Conference on Harmonics And Quality of Power, 8^a. Proceedings...* Athens: 1998, p. 590-597.
- [59] VARADAN, S.; MAKRAN, E.B. GENERALIZED LOAD MODELING IN THE PRESENCE OF HARMONICS ANDDISTORTION. *In: Southeastern Symposium on System Theory, 25. Proceedings...* Tuscaloosa: 1993, p. 124-128.
- [60] SOLIMAN, S. A. A SIMPLE AND NOVEL TECHNIQUE FOR LINEAR AND NONLINEAR LOAD MODELLING IN THE TIME DOMAIN. *International journal of power & energy systems*, vol. , n^o 2, p. 127-131, 1997.
- [61] WILLIAMS, S.M.; BROWFIELD, G.T.; DUFFUS, J.W. HARMONIC PROPAGATION ON AN ELECTRIC DISTRIBUTION SYSTEM: FIELDMEASUREMENTS COMPARED WITH COMPUTER SIMULATION. *IEEE Transactions on Power Delivery* , vol. 8, n^o 2, p. 547-552, apr.1993.
- [62] BROWNE, N.; PERERA, S.; RIBEIRO, P.F. HARMONIC LEVELS AND TELEVISION EVENTS. *In: IEEE Power Engineering Society General Meeting, 2007. Proceedings...* Tampa: 2007, p. 1-6.
- [63] XU, W.; LIU, Y. A METHOD FOR DETERMINING CUSTOMER AND UTILITY HARMONIC CONTRIBUTIONS AT THE POINT OF COMMON COUPLING. *IEEE Transactions on Power Delivery*, vol. 15, n^o 2, p. 804-811, apr. 2000.
- [64] LOONEY, C.G. PATTERN RECOGNITION USING NEURAL NETWORKS. 1^a. ed. New York: Oxford, 1997.
- [65] BISHOP, C.M. NEURAL NETWORKS FOR PATTERN RECOGNITION. 1^a. ed. New York: Oxford, 1995.
- [66] KLIR, G.J.; YUAN, B. FUZZY SETS AND FUZZY LOGIC: THEORY AND APPLICATIONS . 1^a. ed. New Jersey: Prentice Hall, 1995.
- [67] YANG, M.; KRIEGMAN, D.J., AHUJA, N. DETECTING FACES IN IMAGES: A SURVEY. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n^o 1, p. 34-58, jan. 2002.
- [68] YUK, D.; FLANAGAN, J. SPEECH RECOGNITION USING NEURAL NETWORKS. *In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. Proceedings...* Phoenix: 1999, p. 157-160.
- [69] MCCULLOCH, W.S.; PITTS, W. A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY. , vol. 1, n^o 1, p. 115-133, 1943.
- [70] MURPHY, M.L. THE QUEST FOR SYNTHETIC INTELLIGENCE. *In: NASA/Soc. Computer Simulation. Proceedings...* Auburn: 1990, p. 16-43.
- [71] ROSENBLATT, F. THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. *Psychological Review*, vol. 65, n^o , p. 386-408, 1958.
- [72] HAGAN, M. T.; DEMUTH, H.B.; BEALE, M.H. NEURAL NETWORK DESIGN. 1^a. ed. Boston: PWS Publishing Company, 1995.

[73] HAGAN, M.T; MENHAJ, M.B TRAINING FEEDFORWARD NETWORKS WITH THE MARQUARDT ALGORITHM . IEEE Transactions on Neural Networks, vol. 5, nº 6, p. 989-993, 1994.

[74] JOSEPH FOURIER. [S.L]: Académie des sciences . 2005. Disponível em <http://www.academie-sciences.fr/membres/in_memoriam/Fourier/Fourier_oeuvre.htm>. Acesso em: 23 de Setembro de 2008.

[75] HAYKIN, S.; VAN VEEN, B. SIGNALS AND SYSTEMS. 2ª. ed. New York: Wiley, 2002.

[76] STRANG, G.; NGUYEN, t. WAVELETS AND FILTER BANKS. 2ª. ed. Wellesley : Wellesley College, 1996.

[77] RODRIGUES, R.P; TORRES, G.L.; SILVA, L.E.B SISTEMA DE OBSERVABILIDADE REMOTO PARA SISTEMAS DE DISTRIBUIÇÃO. *In: 7th Latin-American Congress on Electricity, Generation, and Transmission. Proceedings...* Valparaiso, Chile: 2007, p. .

Apêndice A

Circuitos do Equipamento de Aquisição de Dados

As figuras seguintes apresentam os circuitos do equipamento de aquisição de dados desenvolvido.

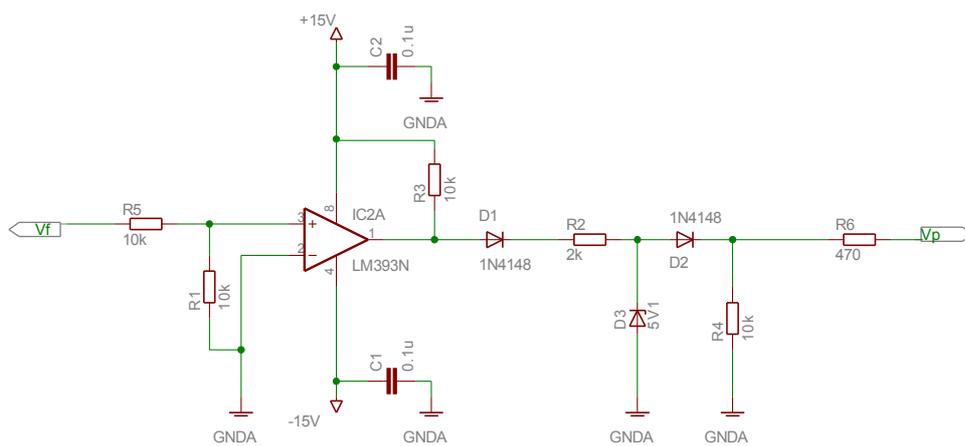


Figura 64 - Circuito de detecção de passagem por zero.

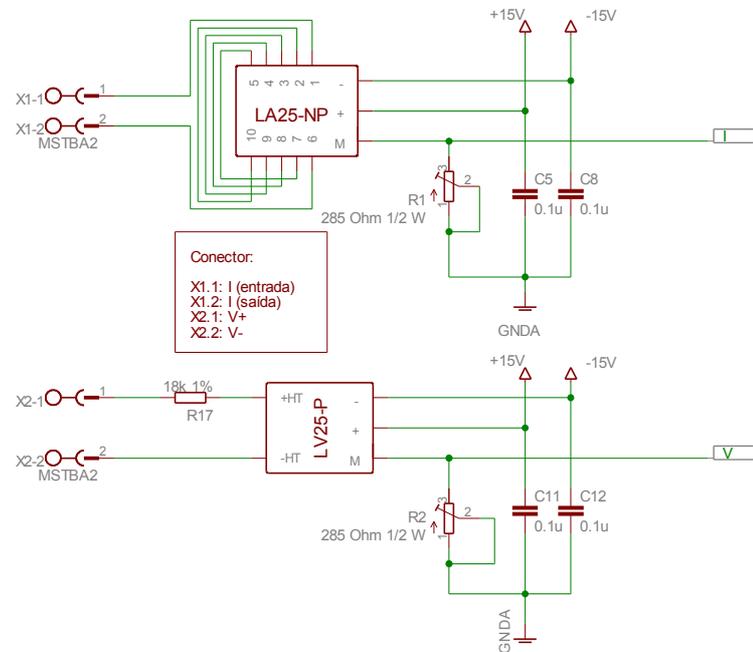


Figura 65 - Circuito de transdução dos sinais de tensão e corrente.

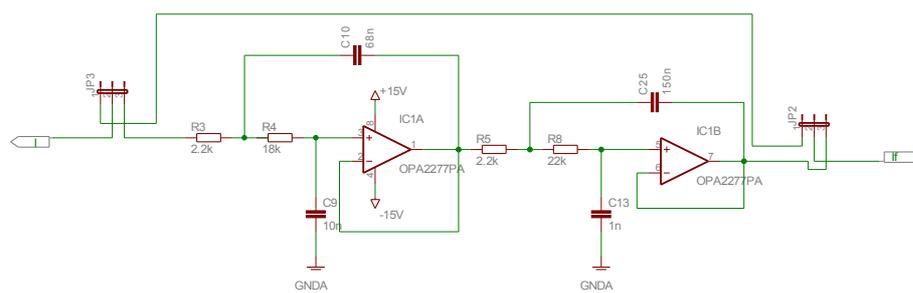


Figura 66 - Circuito de filtragem dos sinais de tensão e corrente.

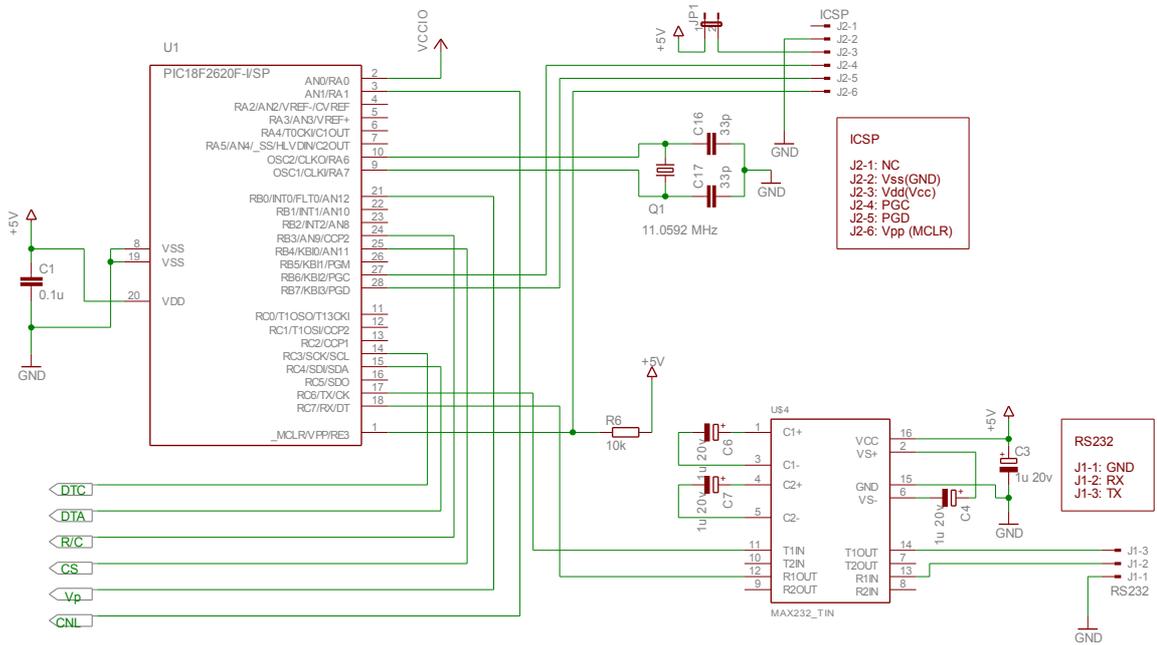


Figura 67 - Circuito de operação.

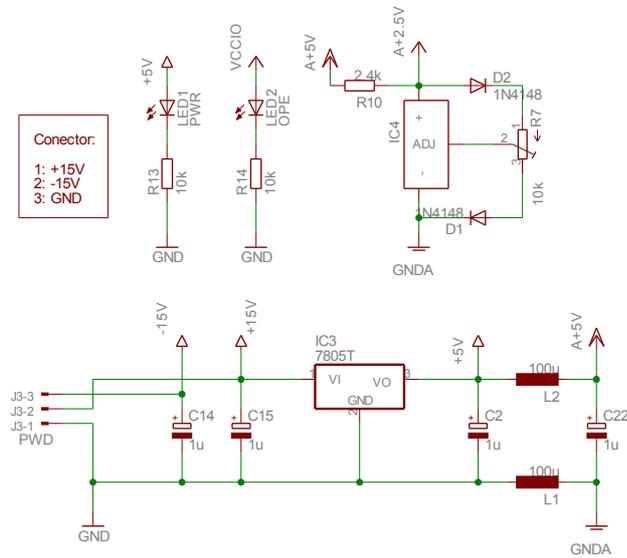


Figura 68 - Circuito de alimentação.

Código-fonte do Aplicativo Embarcado

Listagem 1 - Código-fonte do aplicativo embarcado

```

/*-----*
 * Inclusões
 *-----*/
#include "daq.h"
//#include "adc.h"

//#include "adc.c"
#ZERO_RAM

// Tipos de dados
#ifndef word
typedef unsigned int16 word;
#endif

#ifndef byte
typedef unsigned int8 byte;
#endif

/*-----*
 * Definições
 *-----*/

// Modos da interface SPI
#define SPI_MODE_0_0 0x4000
#define SPI_MODE_0_1 0x0000
#define SPI_MODE_1_0 0x0010
#define SPI_MODE_1_1 0x4010

// Pinos de E/S
#define LED_OPE PIN_A0
#define ADC_RC PIN_B3
#define ADC_CS PIN_B4

// Registros do timer 0
#define TMR0L 0xFD6
#define TMR0H 0xFD7
#define T0CON 0xFD5
#define INTCON 0xFF2
#define TMR3L 0xFB2
#define TMR3H 0xFB3
#define T3CON 0xFB1
#define TMR2 0xFCC
#define PR2 0xFCB
#define T2CON 0xFCA

// Registros de E/S
#define LATB 0xF8A
#define PINO_ADC_RC 3
#define PINO_ADC_CS 4
#define LATA 0xF89
#define PINO_ADC_CANAL 1

#define delay(clock=11059200,RESTART_WDT, Slow)
#define spi(FORCE_HW, BITS=8, stream=SPI_STREAM)
#define rs232(baud=38400,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PC,ERRORS)

/*-----*
 * Variáveis globais
 *-----*/
byte gbtLED; // Controle do led de sinalização de operação
char gacMsg[5]; // Buffer para recepção de mensagens
byte gbtTamMsg; // Tamanho da mensagem recebida
byte gbtProcMsg; // Sinalização para indicar que uma mensagem foi recebida
word gwQuantidade; // Quantidade de amostras a ser obtida
word gwAmostras; // Quantidade de amostras obtidas
byte gabtAmostras[2048]; // Vetor de amostras
byte *gpbtAmostra; // Ponteiro para o buffer de amostras
byte gbtValor; // Valor coletado
byte gbtCanal; // Canal a ser utilizado
byte gbtEnviarAmostras; // Sinalização para o envio das amostras coletadas

```

```

float      gfValor;
byte       vbtValor1, vbtValor2;

#priority timer0, ext

/*-----*
 * vISR_DETECTOR_ZERO
 *-----*
 *
 * Finalidade:
 * Manipulador da interrupção externa, utilizada para se
 * detectar a passagem por zero do sinal de tensão.
 *-----*/
#int_ext HIGH
void vISR_DETECTOR_ZERO(void)
{
    // Desativar a interrupção externa
    // utilizada para se detectar a passagem
    // por zero da onda de tensão
    disable_interrupts(INT_EXT);

#asm

    // Obter a primeira amostra e
    // ajustar o timer para se obter as demais
    // amostras
    bcf     TOCON, 7
    movlw  122
    movwf  TMR0L
    bsf    TOCON, 7

    // Requisitar uma conversão
    bcf    LATB, PINO_ADC_RC
    nop

    // Inibir a conversão e colocar o ADC em modo de leitura
    bsf    LATB, PINO_ADC_RC
    nop

#endasm

    // Recuperar o resultado da conversão e obter o byte MSB
    gpbtAmostra[0] = spi_read(0xFF);

    // Obter o byte LSB
    gpbtAmostra[1] = spi_read(0xFF);

    gpbtAmostra += 2;
    gwQuantidade--;
}

/*-----*
 * vISR_AMOSTRAGEM
 *-----*
 *
 * Finalidade:
 * Manipulador da interrupção do timer0, que é utilizado
 * para se obter amostras
 *-----*/
#int_timer0 HIGH
void vISR_AMOSTRAGEM(void)
{
    // Obter a próxima amostra e ajustar o timer para obter
    // a próxima

#asm

    // Para uma freq. de amostragem de 15360 Hz (256*60)
    // e com um cristal de 11059200 Hz como clock da CPU,
    // tem-se a relação (11059200/4)/(256*60) = 180 entre os períodos.
    //
    // Desta forma, tem-se 180 ciclos do clock interno
    // para equivaler o tempo de 1/15360.
    //
    // No entanto, para se ajustar o clock para o novo estouro
    // equivalente a 1/15360, são utilizados os comandos bcf, bsf,
    // movwf e movlw, que necessitam de 1+1+1+1=4 ciclos.
    //

```

```

// Como o ajuste de estouro do timer perante os 180 ciclos
// deve ser feito através do valor 256-180 = 76, os 4 ciclos
// utilizados para se reconfigurar o timer para a próxima amostra
// devem ser levados em conta, e, desta forma, o valor de ajuste
// torna-se 76 + 4 = 80
//
// Ainda, devido à instrução GOTO que toma 6 ciclos e que é utilizada
// pelo manipulador de instruções para se executar o manipulador da
// interrupção do timer, o valor correto é de 86

// Desligar o timer
bcf    TOCON, 7

// Ajustar o intervalo de estouro
movlw  126    // 15360 Hz = 256 * 60 Hz com HIGH na interrupção
movwf  TMR0L

// Reiniciar o timer
bsf    TOCON, 7

// Requisitar uma conversão
bcf    LATB, PINO_ADC_RC
nop

// Inibir a conversão e colocar o ADC em modo de leitura
bsf    LATB, PINO_ADC_RC
nop

#endasm

// Recuperar o resultado da conversão e obter o byte MSB
gpbtAmostra[0] = spi_read(0xFF);

// Obter o byte LSB
gpbtAmostra[1] = spi_read(0xFF);
gpbtAmostra += 2;

// Validar a continuação do processo de amostragem
if ( gwQuantidade == 0 )
{
#asm
    // Finalizar o timer
    bcf    TOCON, 7

    // Inabilitar o ADC
    bsf    LATB, PINO_ADC_CS
#endasm

    gbtEnviarAmostras = 1;
}

gwQuantidade--;
}

/*-----*
 * Laço de repetição do firmware *
 *-----*/

/*-----*
 * main *
 *-----*
 *
 * Finalidade: *
 * Laço de repetição da aplicação. *
 *
 * Todas as funcionalidades da placa de aquisição são *
 * são disparadas por interrupções. Tais interrupções *
 * sinalizam a necessidade de tratamento dos eventos que, *
 * pelo laço de repetição, tem suas rotinas chamadas. *
 *
 *-----*/
void main()
{
    char    vcChr;

    // Configuração dos periféricos

    // Não utilizar os ADCs

```

```

setup_adc_ports(NO_ANALOGS|VSS_VDD);
setup_adc(ADC_OFF|ADC_TAD_MUL_0);

// Desativar o watchdog
setup_wdt(WDT_OFF);

// Configuração dos timers
setup_timer_0(RTCC_OFF|RTCC_DIV_4);
setup_timer_1(T1_DISABLED|T1_DIV_BY_1|T1_CLK_OUT);
setup_timer_2(T2_DISABLED,0,1);
setup_timer_3(T3_DISABLED|T3_DIV_BY_1);

// Desativar os comparadores
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);

// Configurar as entradas/saídas digitais
set_tris_b( 0b11100001 );
set_tris_a( 0b11001100 );
set_tris_c( 0b11111101 );

// Iniciar o barramento SPI
setup_spi(SPI_MASTER|SPI_MODE_0_0|SPI_CLK_DIV_4|0x8000);
output_high(LED_OPE);

// Iniciar variáveis de estado
gbtLED = 0;
gbtTamMsg = 0;
gbtProcMsg = 0;
gbtEnviarAmostras = 0;

// Configurar as interrupções
ext_int_edge( 0, L_TO_H );
disable_interrupts(GLOBAL);
disable_interrupts(INT_EXT);
disable_interrupts(INT_EXT1);
disable_interrupts(INT_EXT2);
enable_interrupts(GLOBAL);

// Configurar o timer 0 como:
//
// - inabilitado
// - 8 bits
// - clock interno (CPU)
// - sem preescalar
T0CON = 0b01001000;

// Inicialização do conversor AD
#asm

bsf    LATB, PINO_ADC_CS
bsf    LATB, PINO_ADC_RC
bcf    LATA, PINO_ADC_CANAL

#endasm

//
// Laço de repetição do software embarcado
//
//
// Todas as funcionalidades da placa
// de aquisição são disparadas por interrupções.
// Tais interrupções sinalizam a necessidade de
// tratamento dos eventos que, pelo laço de repetição,
// tem suas rotinas chamadas.
//
do
{

// Enviar as amostras ao PC, quando aplicável
if ( gbtEnviarAmostras == 1 )
{
gbtEnviarAmostras = 0;
gwAmostras = gwAmostras << 1;

for ( gwQuantidade = 0; gwQuantidade < gwAmostras; gwQuantidade++ )

```

```

    {
        putc( gabtAmostras[gwQuantidade] );
    }

    delay_ms(100);

#asm
    bcf      LATA, PINO_ADC_CANAL
#endasm
}
// Processar comandos vindos do computador
else if ( kbhit() )
{
    // Receber comandos com a estrutura [ABC]
    vcChr = getc();
    gacMsg[ gbtTamMsg ] = vcChr;
    gbtTamMsg++;
    gbtProcMsg = 0;

    if ( vcChr == '[' )
    {
        gbtTamMsg = 1;
        gacMsg[ 0 ] = '[';
    }
    else if ( gbtTamMsg == 5)
    {
        gbtProcMsg = (vcChr == ']');
        gbtTamMsg = 0;
    }
}
// Receber e processar mensagens recebidas através do dispositivo serial
if ( gbtProcMsg == 1)
{
    gbtProcMsg = 0;

    // Comando para se amostrar
    if ( gacMsg[1] == 'A' )
    {
        // Obter a quantidade de valores a amostrar
        gbtCanal = gacMsg[2] >> 7;
        gwQuantidade = gacMsg[2] & 0x0F;
        gwQuantidade = (gwQuantidade << 8);
        gwQuantidade += gacMsg[3];
        gwAmostras = gwQuantidade;
        gpbtAmostra = &gabtAmostras[0];

        if ( gwQuantidade > 1024 )
        {
            continue;
        }

        // Selecionar o canal no AD
        if ( gbtCanal == 0)
        {
#asm
            bcf      LATA, PINO_ADC_CANAL
#endasm
        }
        else
        {
#asm
            bsf      LATA, PINO_ADC_CANAL
#endasm
        }

        delay_ms(1000);

#asm
        // Habilitar a comunicação com o conversor AD
        bcf      LATB, PINO_ADC_CS
        bsf      LATB, PINO_ADC_RC
        nop

        // Preparar o timer de amostragem para
        // iniciar o processo de coleta de amostras
        bsf      INTCON, 5
#endasm

        // Habilitar a interrupção
        // externa no pino B0, que

```

```
        // é utilizada para se detectar
        // a passagem pelo valor zero
        // da forma de onda de tensão.
        clear_interrupt(INT_EXT);
        enable_interrupts(INT_EXT);

    }
}

// Controle do LED de sinalização
gbtLED++;

if ( gbtLED == 0 )
{
    output_low(LED_OPE);
}
else if ( gbtLED == 128 )
{
    output_high(LED_OPE);
}
else if ( gbtLED == 255 )
{
    // Sinalizar o estado de operação do equipamento
    fprintf(PC, "[OPE]");
}

    delay_ms (10);
}
while( true );
}
```

Apêndice B

Arquivo de Configuração da Rede Neural

Através do arquivo de configuração da rede neural artificial implementada é possível parametrizar a estrutura da rede, escolhendo seu número de camadas bem como o número de *perceptrons* em cada camada, o tipo de função de ativação para cada camada, o algoritmo de treinamento, o número de épocas e o erro de treinamento a serem utilizados como parâmetros de finalização do treinamento, o número de pares formados por vetores-alvo e vetores de entrada a serem utilizados no treinamento da rede, além da especificação de parâmetros de treinamento como passo, fator e a taxa de aprendizado, bem como os próprios vetores-alvo e os vetores de entrada utilizados no treinamento.

Um exemplo de arquivo de configuração é apresentado na Listagem 2. Esse arquivo define uma rede neural composta por 2 nós de entrada, 2 *perceptrons* na camada intermediária e 1 *perceptron* na cada de saída. Para cada uma das camadas foi escolhido utilizar uma função de ativação unipolar, tendo sido escolhido para treinamento o algoritmo de *Levenberg-Marquardt* e sendo disponibilizados 4 pares formados por vetores de entrada, chamados de amostras no arquivo, e vetores-alvo.

Mais explicações com relação aos demais parâmetros do arquivo de configuração são apresentados na própria listagem que o exemplifica.

Listagem 2 - Arquivo de configuração

```

;
; UNIFEI - Universidade Federal de Itajubá
; IEE - Instituto de Engenharia Elétrica
; Mestrado em Engenharia Elétrica
;
; Rodrigo de Paula Rodrigues
;
; Modelo de arquivo de dados de treinamento de uma MLP
;
;
;
; O arquivo deve sempre conter as sessões [REDE], [TREINAMENTO],
; [AMOSTRA] e [ALVO], que devem ser apresentadas nesta mesma
; seqüência.
;
;
;
; IMPORTANTE: Cada linha do arquivo não pode conter mais que 1024 caracteres
;             de comprimento.
;
;
;
; Arquitetura da MLP
;
[REDE]
CAMADAS=3             ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=2 2 1   ; nós em cada camada. Separar com espaço " ".
                    ; Pode conter até 4 camadas, incluindo a camada de entrada.
                    ; No exemplo:
                    ;     - 2 nós de entrada
                    ;     - 2 neurônios na camada intermediária
                    ;     - 1 neurônios na camada de saída

ATIVACAO=0 0         ; funções de ativação. Separar com espaço " ".
                    ; Defini as funções de ativação a partir da segunda camada,
                    ; podendo ser: 0-unipolar, 1-linear
                    ; No exemplo:
                    ;     - função sigmoidal na camada de entrada;
                    ;     - função linear na camada de saída.

;
; Configuração do treinamento
;
; O treinamento é controlado pelo número de épocas,
; pelo erro mínimo ou por ambos.
;
; Se somente número de épocas for fornecido (diferente de 0), então
; o treinamento é executado até a época fornecida.
; Se somente o erro mínimo for fornecido (diferente de 0), então
; o treinamento é executado até que se alcance o erro mínimo, ou
; até a quantidade de 200 épocas.
; Caso ambos sejam preenchidos, ambos são utilizados.
;
;
;
;
[TREINAMENTO]
ALGORITMO=1          ; algoritmo de treinamento:
                    ;     0-steepest descent; 1-
                    ;     levenberg-marquardt
                    ;     quantidade de épocas a treinar
EPOCAS=10            ; erro mínimo a ser atingido. Usar ponto decimal '.'
ERRO=0.12345        ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
AMOSTRAS=4          ; passo para o algoritmo de Levenber-marquardt
PASSO=0.01          ; fator para o algoritmo de Levenber-marquardt
FATOR=1.2           ; taxas de aprendizado. Separar com espaço " ".
APRENDIZADO=3 2    ; Estipula as taxas de aprendizado para as camadas.

```

```

; Deve conter o mesmo número de argumentos fornecidos
; para o parâmetro "ATIVACAO"
; No exemplo:
;       - A camada intermediária tem taxa de aprendizado 0.4
;       - A camada de saída tem taxa de aprendizado de 0.1

;
; Sessões de dados
;
;
;
; Amostras de entrada (Q)
;
;
; - Deve ser utilizado o separador '.' para casas decimais;
; - Os valores devem ser separados por um único espaço em branco (' ');
; - Cada vetor deve conter 'ENTRADA' (N) componentes;
; - Deve haver pelo menos 'AMOSTRAS' vetores.
; - Somente a quantidade "AMOSTRAS" de amostras é lida, sendo as demais, caso haja, desprezadas
;

[AMOSTRA]
0.1 0.1
0.1 0.9
0.9 0.1
0.9 0.9

;
; Vetores-alvo (T)
;
;
; - Deve ser utilizado o separador '.' para casas decimais;
; - Os valores devem ser separados por um único espaço em branco (' ');
; - Cada vetor deve conter 'SAIDA' (J) componentes;
; - Deve possuir a mesma quantidade de vetores apresentada na seção ENTRADA;
; - Os vetores alvo e de entreva na posição X formam o par (Qx,Tx).
; - Somente a quantidade "AMOSTRAS" de alvos é lida, sendo as demais, caso haja, desprezadas
;

[ALVO]
0.1
0.9
0.9
0.1

```

Testes de Operação e Treinamento

Em companhia ao teste da rede neural relatado na seção 7.2.1, alguns outros testes de operação e treinamento foram efetuados. Esses testes tiveram como objetivo principal validar a implementação efetuada e assim testar a convergência dos algoritmos utilizados para se treinar a rede neural.

De uma forma geral, os testes foram efetuados utilizando:

- i. Uma rede neural com estrutura 3-3-2-1 (três nós de entrada, três *perceptrons* de processamento na primeira camada, e 1 perceptron de saída), com treinamento através do algoritmo de mínimo passo descendente;
- ii. A estrutura 3-2-2-1, mas com algoritmo de treinamento de *Levenberg-Marquardt*;
- iii. Uma rede neural com estrutura 3-4-1, com treinamento via algoritmo de mínimo passo descendente;
- iv. Uma rede neural com estrutura 2-2-2-1, com treinamento através de ambos os algoritmos implementados para o treinamento mas com uma quantidade de vetores de treinamento inferior ao número de equações geradas pela estrutura utilizada, o que sugere um problema de convergência da solução.

Exemplos da variação do erro de treinamento em função do número de épocas de treinamento são uns para cada um dos testes são apresentados através da Figura 69 até a Figura 70, tendo sido levantadas algumas observações como:

- i. De uma forma geral, sempre quando a estrutura da rede testada conteve uma quantidade de *perceptrons* apropriada pra o número de vetores de treinamento e de classes a serem classificadas, houve convergência no treinamento;
- ii. Para o caso dos treinamentos efetuados pelo algoritmo de *Levenberg-Marquadt*, as descontinuidades observados em seus erros de treinamento são devido à mudança de abordagem efetuada pelo algoritmo, com a qual o algoritmo ou opera como método de *Gauss-Newton* ou como o método de mínimo passo descendente [72];

- iii. O algoritmo de *Levenberg-Marquardt* obteve melhor desempenho que o de mínimo passo descendente, pois para todos os testes efetuados, sua convergência requereu, em comparação ao último, quantidades menores de épocas de treinamento;
- iv. No caso em que foi utilizada uma quantidade inapropriada de vetores de treinamento em comparação à quantidade de *perceptrons* na estrutura da rede [64], houvesse uma significativa dificuldade de convergência do algoritmo de mínimo passo descendente e, para alguns casos testados, até do algoritmo de *Levenberg-Marquardt*.

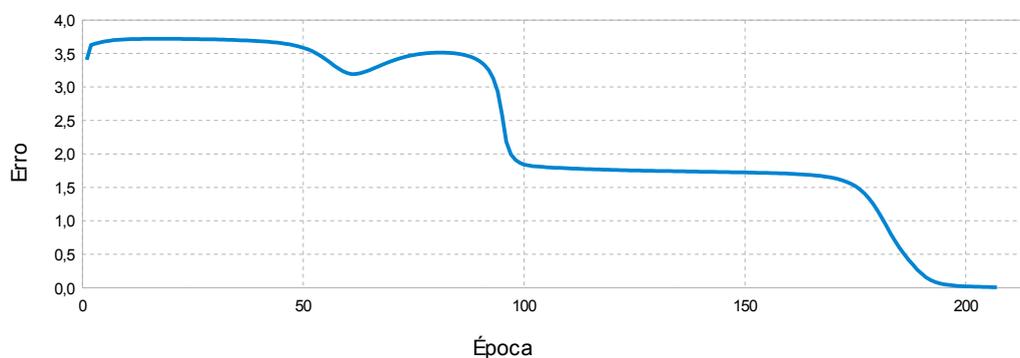


Figura 69 - Exemplo da variação do erro de treinamento obtido pelo algoritmo de mínimo passo descendente para uma rede com estrutura 3-3-2-1.

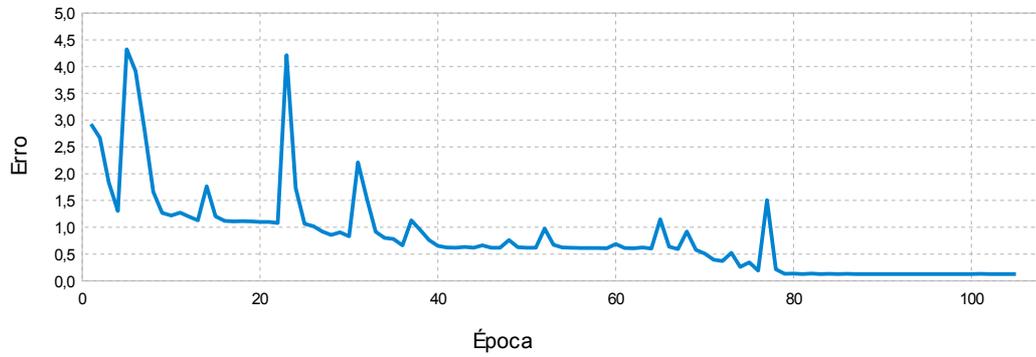


Figura 70 - Exemplo de variação do erro de treinamento obtido pelo algoritmo de *Levenberg-Marquadt* ao treinar uma rede com estrutura 3-2-2-1.

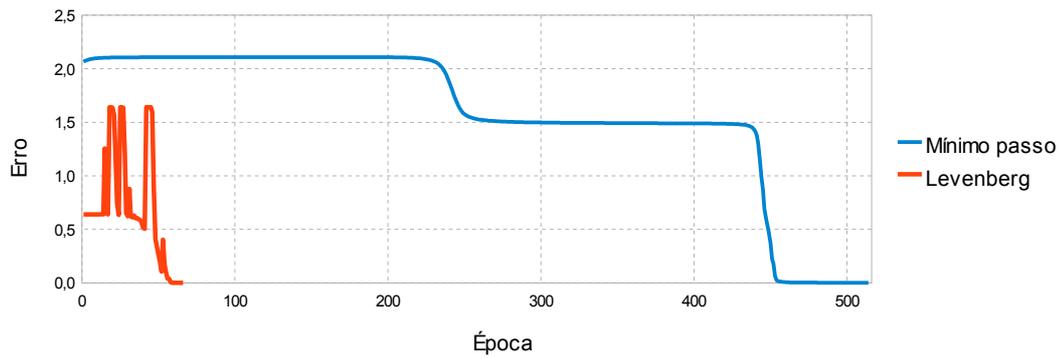


Figura 71 - Exemplo de variações do erro de treinamento para uma rede com estrutura 2-2-2-1.

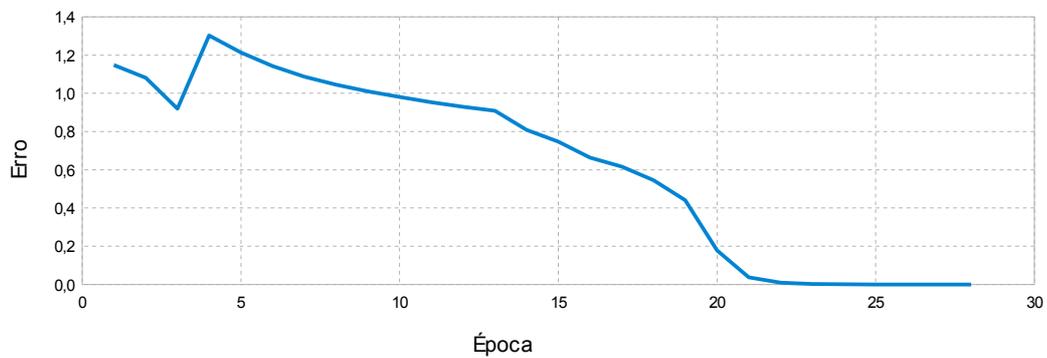


Figura 72 - Exemplo de variação do erro de treinamento obtido pelo algoritmo de *Levenberg-Marquadt* ao treinar uma rede com estrutura 3-4-1.

Código-fonte

Foram desenvolvidos os códigos-fonte para todas as necessidades relacionadas à implementação da rede neural em modo de treinamento e operação, desde as operações relacionadas à manipulação de matrizes, até as operações de interpretação do arquivo de configuração e de implementação da interface de manipulação da rede neural. No entanto, como essas implementações são acessórias à implementação da rede neural propriamente dita, somente os códigos-fonte relacionados aos modos de operação e de treinamento da rede artificial são listados.

Listagem 3 - Código-fonte da rede neural artificial

```

#define K_MLP_ERRO                0x00
#define K_MLP_OK                  0x01
#define K_MLP_ERRO_SESSAO_REDE   K_MLP_OK + 1    // Erro com a seção REDE
#define K_MLP_ERRO_SESSAO_TREINAMENTO K_MLP_OK + 2 // Erro com a seção TREINAMENTO
#define K_MLP_ERRO_SESSAO_AMOSTRA K_MLP_OK + 3    // Erro com a seção AMOSTRA
#define K_MLP_ERRO_SESSAO_ALVO   K_MLP_OK + 4    // Erro com a seção ALVO
#define K_MLP_CAMADAS            0x04            // Número máximo de camadas na MLP

//
// Estrutura stMatriz
//

typedef struct
{
    WORD    wLinhas,
           wColunas;

    double  **pdDados;
} stMatriz;

//
// Dados de uma MLP
//

typedef struct
{
    WORD    wCamadas;                // Quantidade de camadas na MLP, incluindo a
                                     // camada de entrada, podendo assumir até 4 camadas

    WORD    awCamada[K_MLP_CAMADAS]; // Quantidade de nós em cada camada

    // Peso das camadas intermediária e de saída
    stMatriz stW[K_MLP_CAMADAS-1]; // Pesos das camadas ("weights")
                                     // Têm a forma
                                     // NEURÔNIO NA RESPECTIVA CAMADA x NEURÔNIO NA CAMADA ANTERIOR

    stMatriz stB[K_MLP_CAMADAS-1]; // Polarização das camadas ("biases")
                                     // Têm a forma
                                     // NEURÔNIO NA RESPECTIVA CAMADA x 1

    // Função de ativação

```

```

    BYTE    abtAtivacao[K_MLP_CAMADAS-1]; // Função de ativação em cada camada (0-unipolar, 1-
bipolar)
} stMLPRede;

//
// Treino de uma MLP
//
typedef struct
{
    stMLPRede    stMLP;           // Arquitetura da rede

    BYTE        btAlgoritmo;     // Treinamento: 0-steepest descent; 1-levenberg-marquardt
    double      dErroMinimo;     // Erro mínimo para o treinamento
    WORD        wEpocas;        // Número de épocas a treinar
    WORD        wEpocaAtual;    // Número de treinamento atual
    WORD        wAmostras;      // Quantidade (Q) de pares (Amostar,Alvo) para treinamento

    // Vetores de amostra e de alvo

    stMatriz    *pstAmostra;     // Vetores de características Q[AMOSTRAS][N]
    stMatriz    *pstAlvo;       // Vetores-alvo T[AMOSTRAS][J]

    // Taxas de aprendizado para o algoritmo de mínimo passo descendente

    double      adAprendizado[K_MLP_CAMADAS-1]; // Taxas de aprendizado para as camadas

    // Parâmetros para o algoritmo de Levenberg-Marquardt

    double      dPasso;
    double      dFator;

    // Erro total

    double      dErroTotal;     // erro total (sum-squared error) para a última época detreinamento
} stMLPTreinamento;

/*-----*
 * wMLPOperar                                     *
 *-----*
 *
 * Finalidade:                                     *
 * Operar a rede MLP e gerar, para o vetor de entrada, *
 * o respectivo vetor de saída.                   *
 *
 * Parâmetros:                                     *
 * p_stRede: entrada.                             *
 * Ponteiro para a estrutura de dados da rede MLP. *
 *
 * p_dQ: entrada.                                 *
 * Ponteiro para o vetor de entrada, que deve conter o *
 * mesmo número de elementos presentes na camada de *
 * entrada da rede MLP.                          *
 *
 * p_dY: saída.                                   *
 * Ponteiro para o vetor a receber os dados de saída *
 * da camada intermediária da MLP.               *
 *
 * p_dJ: saída.                                   *
 * Ponteiro para o vetor a receber os dados de saída *
 * da operação da MLP sobre os dados de entrada. *
 *
 * Retorno:                                       *
 * Não há.                                       *
 *-----*/
WORD wMLPOperar( stMLPRede    *p_stRede,
                stMatriz    *p_stEntrada,
                stMatriz    *p_stSaida)
{
    WORD        vwCamada;

    //
    // Validar os parâmetros
    // de entrada

```

```

//
if ( ( p_stRede->wCamadas < (K_MLP_CAMADAS-1) ) ||
      ( p_stRede->wCamadas > (K_MLP_CAMADAS) ) )
{
    return K_MLP_ERRO;
}

if ( ( p_stRede->awCamada[ 0 ] != p_stEntrada->wLinhas ) ||
      ( p_stEntrada->wColunas != 1 ) )
{
    return K_MLP_ERRO;
}

if ( ( p_stRede->awCamada[ p_stRede->wCamadas - 1 ] != p_stSaida->wLinhas ) ||
      ( p_stSaida->wColunas != 1 ) )
{
    return K_MLP_ERRO;
}

vMLPOperar( p_stRede,
            p_stEntrada,
            p_stSaida,
            false,
            NULL );

return K_MLP_OK;
}

/*-----*
 * vMLPTreinarEpocaSteepestDescent                                     *
 *-----*
 *
 * Finalidade:                                                         *
 *   Treinar a rede por uma época.                                     *
 *
 * Parâmetros:                                                         *
 *   p_stTreinamento: entrada/saída.                                  *
 *   Ponteiro para a estrutura que contém as definições               *
 *   da rede neural e da seção de treinamento.                       *
 *
 *
 * Retorno:                                                            *
 *   K_MLP_OK: sucesso.                                               *
 *   K_MLP_ERRO: erro.                                               *
 *
 *-----*/
void vMLPTreinarEpocaSteepestDescent( stMLPTreinamento *p_stTreinamento )
{
    stMatriz    *pstA;
    WORD        vwCamada,
               vwAmostra,
               vwIndice;
    double       vdErroAmostra,
               vdErroTotal;

    //
    // Criar as matrizes A, as quais
    // receberão os valores de saída de cada
    // uma das camadas da rede MLP
    //

    pstA = new stMatriz[ p_stTreinamento->stMLP.wCamadas ];

    for ( vwCamada = 0; vwCamada < p_stTreinamento->stMLP.wCamadas ; vwCamada++ )
    {
        //
        // A matriz pstA[0] é utilizada como um ponteiro
        // para a amostra a ser processada e, portanto,
        // não é requisitada memória para a mesma.
        //

        pstA[ vwCamada ].pdDados = NULL;

        wMATALocar( &pstA[ vwCamada ],
                   p_stTreinamento->stMLP.awCamada[ vwCamada ],
                   1 );
    }
}

```

```

}

//
// Ajustar os pesos W e as polarizações B
// para cada uma das amostras da seção de
// treinamento.
//

vdErroTotal = 0.0;

for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    //
    // Preencher a matriz de entrada com os dados
    // da respectiva amostra
    //

    for ( vwIndice = 0; vwIndice < p_stTreinamento->stMLP.awCamada[0]; vwIndice++ )
        pstA[ 0 ].pdDados[vwIndice][0] = p_stTreinamento->pstAmostra[vwAmostra].pdDados[vwIndice]
[0];

    //
    // Processar cada um das amostras da época
    // de treinamento
    //

    for( vwCamada = 1; vwCamada < p_stTreinamento->stMLP.wCamadas ; vwCamada++ )
    {

        //
        // Efetuar as operações  $A^m = f(W^m * A^{(m-1)} + B^m)$ 
        //

        vMLPOperar( &p_stTreinamento->stMLP,
                    &pstA[ 0 ],
                    &pstA[ p_stTreinamento->stMLP.wCamadas - 1 ],
                    true,
                    pstA );

        //
        // Calcular o erro total
        //

        for( vwIndice = 0; vwIndice < p_stTreinamento->stMLP.awCamada[p_stTreinamento-
>stMLP.wCamadas - 1] ; vwIndice++ )
        {
            vdErroTotal += pow((p_stTreinamento->pstAlvo[vwAmostra].pdDados[vwIndice][0] -
                pstA[ p_stTreinamento->stMLP.wCamadas - 1 ].pdDados[vwIndice][0]), 2);
        }

        //
        // Ajustar os pesos e as polarizações
        // através do algoritmo "steepest descent"
        //

        vMLPSteepestDescent( p_stTreinamento,
                             pstA,
                             vwAmostra );
    }

}

//
// Liberar recursos utilizados
//

for ( vwCamada = 0; vwCamada < p_stTreinamento->stMLP.wCamadas; vwCamada++ )
{
    //
    // A matriz pstA[0] é utilizada como um ponteiro
    // para a amostra a ser processada e, portanto,

```

```

        // não foi requisitada memória para a mesma.
        //
        vMATLiberar( &pstA[ vwCamada ] );
    }

    delete [] pstA;

    //
    // Atualizar o número de épocas de
    // treinamento já efetuadas
    //

    p_stTreinamento->wEpocaAtual = p_stTreinamento->wEpocaAtual + 1;
    p_stTreinamento->dErroTotal = vdErroTotal;
}

/*-----*
 * vMLPTreinarEpocaLevenbergMarquardt *
 *-----*
 * *
 * Finalidade: *
 * Treinar uma época utilizando o algoritmo de *
 * Levenberg-Marquardt *
 * *
 * Parâmetros: *
 * p_stTreinamento : entrada/saída. *
 * Ponteiro para os dados de treinamento da rede *
 * neural. *
 * *
 * Retorno: *
 * Não há. *
 * *
 *-----*/
void vMLPTreinarEpocaLevenbergMarquardt( stMLPTreinamento *p_stTreinamento )
{
    stMatriz    **pstS,**pstA, vstParametros, vstErroEpoca, vstErroAmostra,
                vstJacobiano, vstF, vstOperacao, vstTransposta;

    double      vdErroEpoca, vdNovoErroEpoca;

    WORD        vwAmostra, vwCamada, vwLinha, vwColuna, vwParametros, vwUltimaCamada,
                vwParametro, vwIndice, vwH, vwQ, vwI, vwJ;

    bool        vbPeso;

    //
    // Preparar variáveis de auxílio
    //
    vwUltimaCamada = p_stTreinamento->stMLP.wCamadas - 1;

    //
    // Criar as matrizes auxiliares
    //
    vMATIniciar( &vstErroEpoca );
    vMATIniciar( &vstJacobiano );
    vMATIniciar( &vstParametros );
    vMATIniciar( &vstErroAmostra );

    vwParametros = 0;
    for( vwCamada = 1; vwCamada < p_stTreinamento->stMLP.wCamadas; vwCamada++ )
    {
        vwParametros += p_stTreinamento->stMLP.awCamada[vwCamada] *
            ( p_stTreinamento->stMLP.awCamada[vwCamada-1] + 1 );
    }

    wMATALocar( &vstParametros,
                vwParametros,
                1 );

    wMATALocar( &vstErroAmostra,
                p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ],
                1 );

    wMATALocar( &vstErroEpoca,
                p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ] *
                p_stTreinamento->wAmostras,
                1 );

    wMATALocar( &vstJacobiano,

```

```

        p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ] *
        p_stTreinamento->wAmostras ,
        vwParametros );

//
// Criar as matrizes para armazenar
// a saída de cada um das camadas
// da rede, para cada uma das amostras.
// Esta matriz é da forma Q x M
//

pstA = new stMatriz*[ p_stTreinamento->wAmostras ];

for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    pstA[ vwAmostra ] = new stMatriz[ p_stTreinamento->stMLP.wCamadas ];

    for( vwCamada = 0; vwCamada < p_stTreinamento->stMLP.wCamadas; vwCamada++ )
    {
        vMATIniciar( &pstA[ vwAmostra ][ vwCamada ] );
        wMATALocar( &pstA[ vwAmostra ][ vwCamada ],
                    p_stTreinamento->stMLP.awCamada[ vwCamada ],
                    1 );
    }

//
// A primeira camada deve receber
// os dados de entrada da amostra
//

    for ( vwLinha = 0; vwLinha < pstA[ vwAmostra ][ 0 ].wLinhas; vwLinha++ )
    {
        pstA[ vwAmostra ][ 0 ].pdDados[ vwLinha ][ 0 ] =
            p_stTreinamento->pstAmostra[vwAmostra].pdDados[ vwLinha ][ 0 ];
    }

}

//
// Criar as matrizes de sensibilidade
// para cada uma das camadas e para
// cada uma das amostras
//
// Esta matriz é da forma (M-1)xQ,
// onde M designa o número total
// de camadas na rede MLP, incluindo
// a camada de entrada.
//
//
// Cada uma das matriz "m" é da forma
//
//  $(S^m)x(s^{(m+1)})$ ,
//
// enquanto a última camada "M"
// é da forma
//
//  $(S^M)x(S^M)$ 
//

pstS = new stMatriz*[ ( p_stTreinamento->stMLP.wCamadas - 1 ) ];

for( vwCamada = 0; vwCamada < ( p_stTreinamento->stMLP.wCamadas-1); vwCamada++ )
{
    pstS[ vwCamada ] = new stMatriz[ p_stTreinamento->wAmostras ];
}

//
// Última camada
//

vwCamada = vwUltimaCamada-1;
for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    vMATIniciar( &pstS[ vwCamada ][ vwAmostra ] );
    wMATALocar( &pstS[ vwCamada ][ vwAmostra ],
                p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ],
                p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ] );
}

//
// Demais camadas
//

```

```

vwCamada = vwUltimaCamada-1;
while( (vwCamada-- > 0 ) )
{
    for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
    {
        vMATIniciar( &pstS[ vwCamada ][ vwAmostra ] );
        wMATALocar( &pstS[ vwCamada ][ vwAmostra ],
                    p_stTreinamento->stMLP.awCamada[ vwCamada + 1 ],
                    p_stTreinamento->stMLP.awCamada[ vwCamada + 2 ] );
    }
}

//
// Aplicar todas as entradas à rede
// e obter o erro para cada uma delas.
//

vdErroEpoca = 0.0;
for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    //
    // Processar a amostra corrente
    // e armazenar os valores de saída
    // de cada uma das camadas da MLP.
    //

    vMLPOperar( &p_stTreinamento->stMLP,
                &pstA[ vwAmostra ][ 0 ],
                &pstA[ vwAmostra ][ vwUltimaCamada ],
                true,
                pstA[ vwAmostra ] );

    //
    // Calcular o erro da amostra segundo
    //
    //  $E_q = t_q - A_q$ 
    //
    // e preencher a matriz de erros da
    // época de treinamento.
    //

    wMATSubtrair( &p_stTreinamento->pstAlvo[vwAmostra],
                  &pstA[ vwAmostra ][ vwUltimaCamada ],
                  &vstErroAmostra );

    for( vwCamada = 0; vwCamada < pstA[ vwAmostra ][ vwUltimaCamada ].wLinhas; vwCamada++ )
    {
        // Calcular o erro total da época
        vdErroEpoca += pow( vstErroAmostra.pdDados[vwCamada][0], 2 );

        // Preencher a matriz de erros da época
        vstErroEpoca.pdDados[ vwAmostra*vstErroAmostra.wLinhas + vwCamada ][0] =
            vstErroAmostra.pdDados[vwCamada][0];
    }
}

//
// Obter as matrizes de sensibilidade
//
//
// Obter a matriz de sensibilidade para a última camada
// e para cada uma das amostras processadas, através
// da fórmula
//
//  $S^M = -F^M(n^M)$ 
//

for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    for ( vwLinha = 0; vwLinha < pstS[ vwUltimaCamada - 1 ][ vwAmostra ].wLinhas; vwLinha++ )
    {
        pstS[vwUltimaCamada - 1][ vwAmostra ].pdDados[ vwLinha ][ vwLinha ] =
            (-1)*dMLPDerivadaAtivacao(&p_stTreinamento->stMLP,
                                       (BYTE)(vwUltimaCamada-1),
                                       pstA[vwAmostra][vwUltimaCamada].pdDados[vwLinha][0] );
    }
}

```

```

}

//
// A partir das sensibilidades da última
// camada, obter as sensibilidades
// das demais camadas para cada um das
// amostras aplicadas à rede
//

vwCamada = vwUltimaCamada-1;
while( (vwCamada-- > 0) )
{
    //
    // Criar a matriz F e as matrizes
    // de auxílio para a camada corrente
    //

    wMATALocar( &vstF,
                p_stTreinamento->stMLP.awCamada[ vwCamada+1 ],
                p_stTreinamento->stMLP.awCamada[ vwCamada+1 ] );

    wMATALocar( &vstTransposta,
                p_stTreinamento->stMLP.stW[ vwCamada + 1 ].wColunas,
                p_stTreinamento->stMLP.stW[ vwCamada + 1 ].wLinhas );

    wMATALocar( &vstOperacao,
                vstF.wLinhas,
                vstTransposta.wColunas );

    //
    // Para a camada corrente,
    // processar a sensibilidade
    // para cada uma das amostras
    //

    for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
    {

        //
        // Preencher a matriz F
        // com os dados da camada
        // e amostra correntes
        //

        for ( vwLinha = 0; vwLinha < vstF.wLinhas; vwLinha++ )
        {
            vstF.pdDados[ vwLinha ][ vwLinha ] =
                dMLPDerivadaAtivacao(&p_stTreinamento->stMLP,
                                     (BYTE)vwCamada,
                                     pstA[vwAmostra][vwCamada+1].pdDados[vwLinha][0] );
        }
        //
        // Efetuar
        //
        //  $S^m = F^m * (W^{(m+1)})^T S^{(m+1)}$ 
        //

        // Obter a transposta de  $W^{(m+1)}$ 
        wMATTranspor( &p_stTreinamento->stMLP.stW[ vwCamada + 1 ],
                     &vstTransposta );

        // Efetuar  $F^m * (W^{(m+1)})^T$ 
        wMATMultiplicar( &vstF,
                        &vstTransposta,
                        &vstOperacao );

        // Efetuar  $S^m = F^m * (W^{(m+1)})^T * S^{(m+1)}$ 
        wMATMultiplicar( &vstOperacao,
                        &pstS[vwCamada+1][vwAmostra],
                        &pstS[vwCamada][vwAmostra] );

    }

    // Liberar as matrizes de auxílio
    vMATLiberar( &vstTransposta );
    vMATLiberar( &vstOperacao );
    vMATLiberar( &vstF );

}

//
// Obter a matriz Jacobiana

```

```

// a partir das matrizes pstA
// e pstS
//
//
// A matriz jacobiana é da forma:
// - Linha: cada linha corresponde a um erro;
// - Coluna: cada coluna corresponde a um parâmetro.
//

vwCamada = 1;
vwIndice = 0;

for( vwParametro = 0; vwParametro < vstJacobiano.wColunas; vwParametro++ )
{
    if ( vwIndice == ( p_stTreinamento->stMLP.awCamada[ vwCamada ] *
        ( p_stTreinamento->stMLP.awCamada[ vwCamada - 1 ] + 1 ) ) )
    {
        vwCamada++;
        vwIndice = 0;
    }

    // Verificar se trata-se de um peso ou de uma polarização
    vbPeso = ( vwIndice < ( p_stTreinamento->stMLP.awCamada[ vwCamada ] *
        p_stTreinamento->stMLP.awCamada[ vwCamada - 1 ] ) );

    if ( vbPeso == true )
    {
        // Acertar os pesos i e j
        vwI = vwIndice / p_stTreinamento->stMLP.awCamada[ vwCamada - 1 ];
        vwJ = vwIndice % ( p_stTreinamento->stMLP.awCamada[ vwCamada - 1 ] );
    }
    else
    {
        // Neste caso o índice vwI está relacionado a uma
        // polarização, e não a um peso
        vwI = vwIndice - ( p_stTreinamento->stMLP.awCamada[ vwCamada ] *
            p_stTreinamento->stMLP.awCamada[ vwCamada - 1 ] );
    }

    for( vwH = 0; vwH < vstJacobiano.wLinhas; vwH++ )
    {
        // Acertar os pesos k e q, os quais estão relacionados com o erro
        vwQ = vwH / p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ];

        vstJacobiano.pdDados[ vwH ][ vwParametro ] =
            pstS[ vwCamada - 1 ][ vwQ ].pdDados[ vwI ][ vwH %
                p_stTreinamento->stMLP.awCamada[ vwUltimaCamada ] ];

        if ( vbPeso == true )
        {
            vstJacobiano.pdDados[ vwH ][ vwParametro ] *=
                pstA[ vwQ ][ vwCamada - 1 ].pdDados[ vwJ ][ 0 ];
        }
    }

    vwIndice++;
}

//
// Preencher a matriz de parâmetros
//

vwParametro = 0;
for( vwCamada = 0; vwCamada < ( p_stTreinamento->stMLP.wCamadas - 1 ); vwCamada++ )
{
    for ( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stW[ vwCamada ].wLinhas ; vwLinha++ )
    {
        for ( vwColuna = 0; vwColuna < p_stTreinamento->stMLP.stW[ vwCamada ].wColunas; vwColuna++ )
        {
            vstParametros.pdDados[ vwParametro ][ 0 ] =
                p_stTreinamento->stMLP.stW[ vwCamada ].pdDados[ vwLinha ][ vwColuna ];
            vwParametro++;
        }
    }

    for ( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stB[ vwCamada ].wLinhas ; vwLinha++ )
    {
        vstParametros.pdDados[ vwParametro ][ 0 ] =
            p_stTreinamento->stMLP.stB[ vwCamada ].pdDados[ vwLinha ][ 0 ];
    }
}

```

```

        vwParametro++;
    }
}

//
// Obter o delta para os parâmetros,
// que é dado por:
//
//  $-\{[(J^T)J + uI]^{-1}\} * (J^T) * V$ ,
//
// no qual  $J$  denota a matriz jacobiana,  $I$ , a identidade,
// e  $V$ , a matriz de erros.
//

//
// Efetuar
//
//  $(J^T)J$ 
//

// Obter a transposta de  $J$ 

wMATALocar( &vstTransposta,
            vstJacobiano.wColunas,
            vstJacobiano.wLinhas );

wMATTranspor( &vstJacobiano,
              &vstTransposta );

// Efetuar  $(J^T)J$ 

wMATALocar( &vstOperacao,
            vstTransposta.wLinhas,
            vstJacobiano.wColunas );

wMATMultiplicar( &vstTransposta,
                 &vstJacobiano,
                 &vstOperacao );

//  $(J^T)J + uI$ 

for( vwLinha = 0; vwLinha < vstOperacao.wLinhas; vwLinha++ )
{
    vstOperacao.pdDados[vwLinha][vwLinha] += p_stTreinamento->dPasso;
}

//
// Obter a inversa de  $[(J^T)J + uI]$ 
//

wMATALocar( &vstF,
            vstOperacao.wLinhas,
            vstOperacao.wColunas );

wMATInverter( &vstOperacao,
              &vstF );

vMATLiberar( &vstOperacao );

//  $\{[(J^T)J + uI]^{-1}\} * (J^T)$ 

wMATALocar( &vstOperacao,
            vstF.wLinhas,
            vstTransposta.wColunas );

wMATMultiplicar( &vstF,
                 &vstTransposta,
                 &vstOperacao );

vMATLiberar( &vstF );

// Obter  $dX = -\{[(J^T)J + uI]^{-1}\} * (J^T) * V$ 

wMATALocar( &vstF,
            vstOperacao.wLinhas,
            vstErroEpoca.wColunas );

wMATMultiplicar( &vstOperacao,

```

```

        &vstErroEpoca,
        &vstF );

wMATMultEscalar( -1,
                &vstF,
                &vstF );

//
// Salvar os parâmetros atuais
//

vMATLiberar( &vstOperacao );
wMATAlocar( &vstOperacao,
            vstParametros.wLinhas,
            vstParametros.wColunas );

for ( vwParametro = 0; vwParametro < vstParametros.wLinhas; vwParametro++ )
{
    vstOperacao.pdDados[ vwParametro ] [ 0 ] =
        vstParametros.pdDados[ vwParametro ] [ 0 ];
}

// Obter x = x + dX

wMATAdicionar(&vstParametros,
              &vstF,
              &vstParametros );

//
// Obter novamente o erro total das
// amostras com os novos valores dos
// pesos e das polarizações
//

//
// Atualizar os parâmetros
//

vwParametro = 0;
for( vwCamada = 0; vwCamada < (p_stTreinamento->stMLP.wCamadas - 1); vwCamada++ )
{
    for ( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stW[vwCamada].wLinhas ; vwLinha++ )
    {
        for ( vwColuna = 0; vwColuna < p_stTreinamento->stMLP.stW[vwCamada].wColunas; vwColuna++ )
        {
            p_stTreinamento->stMLP.stW[vwCamada].pdDados[vwLinha][vwColuna] =
                vstParametros.pdDados[vwParametro][0];
            vwParametro++;
        }
    }

    for ( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stB[vwCamada].wLinhas ; vwLinha++ )
    {
        p_stTreinamento->stMLP.stB[vwCamada].pdDados[vwLinha][0] =
            vstParametros.pdDados[vwParametro][0];

        vwParametro++;
    }
}

//
// Aplicar novamente todas as entradas à rede
// e obter o erro para cada uma delas.
//

vdNovoErroEpoca = 0.0;
for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
{
    //
    // Processar a amostra corrente
    // e armazenar os valores de saída
    // de cada uma das camadas da MLP.
    //

    vMLPOperar( &p_stTreinamento->stMLP,

```

```

        &pstA[ vwAmostra ][ 0 ],
        &pstA[ vwAmostra ][ vwUltimaCamada ],
        true,
        pstA[ vwAmostra ] );

    //
    // Calcular o erro da amostra segundo
    //
    //  $E_q = t_q - A_q$ 
    //
    //

    wMATSubtrair( &p_stTreinamento->pstAlvo[vwAmostra],
                 &pstA[ vwAmostra ][ vwUltimaCamada ],
                 &vstErroAmostra );

    for( vwCamada = 0; vwCamada < pstA[ vwAmostra ][ vwUltimaCamada ].wLinhas; vwCamada++ )
    {
        // Calcular o erro total da época
        vdNovoErroEpoca += pow( vstErroAmostra.pdDados[vwCamada][0], 2);
    }

    //
    // Modificar o passo segundo
    // o aumento ou diminuição do
    // erro total da amostra com
    // relação aos novos parâmetros
    //

    if ( vdNovoErroEpoca <= vdErroEpoca )
    {
        p_stTreinamento->dPasso /= p_stTreinamento->dFator;
    }
    else
    {
        p_stTreinamento->dPasso *= (p_stTreinamento->dFator);

        //
        // Desprezar os parâmetros atuais
        // e recuperar os parâmetros anteriores
        //

        vwParametro = 0;
        for( vwCamada = 0; vwCamada < (p_stTreinamento->stMLP.wCamadas - 1); vwCamada++ )
        {
            for( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stW[vwCamada].wLinhas ; vwLinha++ )
            {
                vwColuna++ )
                {
                    for( vwColuna = 0; vwColuna < p_stTreinamento->stMLP.stW[vwCamada].wColunas;
                        {
                            p_stTreinamento->stMLP.stW[vwCamada].pdDados[vwLinha][vwColuna] =
                                vstOperacao.pdDados[vwParametro][0];
                            vwParametro++;
                        }
                    }
                }
            }
            for( vwLinha = 0; vwLinha < p_stTreinamento->stMLP.stB[vwCamada].wLinhas ; vwLinha++ )
            {
                p_stTreinamento->stMLP.stB[vwCamada].pdDados[vwLinha][0] =
                    vstOperacao.pdDados[vwParametro][0];

                vwParametro++;
            }
        }
    }

    //
    // Liberar recursos
    //

    vMATLiberar( &vstF );
    vMATLiberar( &vstOperacao );
    vMATLiberar( &vstTransposta );

    // Matriz pstA

    for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )

```

```

{
    for( vwCamada = 0; vwCamada < p_stTreinamento->stMLP.wCamadas; vwCamada++ )
    {
        vMATLiberar( &pstA[ vwAmostra ][ vwCamada ] );
    }

    delete [] pstA[ vwAmostra ];
}

delete [] pstA;
// Matriz pstS

for( vwCamada = 0; vwCamada < (p_stTreinamento->stMLP.wCamadas -1 ); vwCamada++ )
{
    for( vwAmostra = 0; vwAmostra < p_stTreinamento->wAmostras; vwAmostra++ )
    {
        vMATLiberar( &pstS[ vwCamada ][ vwAmostra ] );
    }

    delete [] pstS[ vwCamada ];
}

delete [] pstS;

//
// Atualizar o erro e o número
// de épocas já treinadas
//
p_stTreinamento->wEpocaAtual++;
p_stTreinamento->dErroTotal = vdNovoErroEpoca;

vMATLiberar( &vstErroEpoca );
vMATLiberar( &vstJacobiano );
vMATLiberar( &vstParametros );
vMATLiberar( &vstErroAmostra );
}

```

Apêndice C

Código-fonte da Transformada Rápida de Fourier

A listagem seguinte contém o código-fonte da função de decomposição através da transformada rápida de Fourier. Os demais códigos-fonte relacionados à interface de operação da transformada aplicada não é listado.

Listagem 4 - Código-fonte da transformada rápida de Fourier

```

/*-----*
* vFFT                                         *
*-----*
*
* Finalidade:
*   Calcular a transformada rápida de fourier para o sinal *
*   de entrada.
*
*   O algoritmo utilizado é o RAD2 e apenas a parte
*   positiva do espectro é retornada.
*
*   Os ângulos de fase são resultantes são em relação a
*   uma função seno.
*
* Parâmetros:
*   p_fSinal: entrada.
*   Ponteiro para o sinal de entrada.
*
*   v_wTamSinal: entrada.
*   Número de amostras presentes no sinal.
*
*   p_fAmplitude: saída.
*   Amplitudes.
*
*   p_fFase: saída.
*   Fases.
*
*   p_wTam: saída.
*   Tamanho dos vetores de amplitudes e fases.
*
* Retorno:
*   Não há.

```

```

*-----*
void vFFT( float *p_fSinal,
          word v_wTamSinal,
          float **p_fAmplitude,
          float **p_fFase,
          word *p_wTam )
{
    long double vldValor;

    word vbtN2,
        vbtEtapa,
        vbtTroca,
        vbtSalto,
        vbtRN,
        vbtIndBase,
        vbtEtapas;

    word vwIndice;

    long double *pldImg,
                *pldReal,
                vldReal,
                vldImg,
                vldWReal,
                vldWImg;

    double vdAngulo;

    pldImg = new long double[ v_wTamSinal << 1 ];
    pldReal = new long double[ v_wTamSinal << 1 ];

    for ( vwIndice = 0; vwIndice < v_wTamSinal; vwIndice ++ )
    {
        pldImg[vwIndice] = 0;
        pldReal[vwIndice] = (long double)p_fSinal[vwIndice];
    }

    //
    // Obter o número de etapas a serem efetuadas
    //

    vbtN2 = (v_wTamSinal >> 1);
    vbtEtapas = 0;
    vbtRN = vbtN2;
    do
    {
        vbtEtapas++;
    }while( vbtRN >= 1 );

    //
    // Aplicar a reversão de bits
    //

    for ( vbtEtapa = 0; vbtEtapa < v_wTamSinal; vbtEtapa++ )
    {
        vbtTroca = 0;
        vbtRN = 0;

        for( vwIndice = 0; vwIndice < vbtEtapas; vwIndice++ )
        {
            vbtTroca = vbtTroca << 1;
            vbtTroca += ((vbtEtapa >> vbtRN)&(0x01));
            vbtRN++;
        }

        if ( vbtTroca < vbtEtapa )
            continue;

        vldValor = pldReal[ vbtEtapa ]*2.0/v_wTamSinal;
        pldReal[ vbtEtapa ] = pldReal[ vbtTroca ]*2.0/v_wTamSinal;
        pldReal[ vbtTroca ] = vldValor;
    }

    vbtSalto = 1;
    vbtRN = 2;
}

```

```

for ( vbtEtapa = 0; vbtEtapa < vbtEtapas; vbtEtapa++ )
{
    vbtIndBase = 0;
    for ( vwIndice = 0; vwIndice < vbtN2; vwIndice++ )
    {
        //
        // Calcular W(r,N)
        //

        vdAngulo = -2.0*(double)M_PI*(double)(vbtIndBase % vbtSalto)/(double)vbtRN;
        vldWReal = cos( (double)vdAngulo );
        vldWImg = sin( (double)vdAngulo );

        //
        // vCCalc = W(vbtEtapa,8/(4,2,1) ) * vComplexo[ vbtIndBase + vbtSalto ];
        //

        vldReal = (pldReal[ vbtIndBase + vbtSalto ]*vldWReal) -
            (pldImg[ vbtIndBase + vbtSalto ]*vldWImg);

        vldImg = (pldReal[ vbtIndBase + vbtSalto ]*vldWImg) +
            (pldImg[ vbtIndBase + vbtSalto ]*vldWReal);

        //
        // vComplexo[ vbtIndBase + vbtSalto ] = vComplexo[ vbtIndBase ] - vCCalc;
        //

        pldReal[ vbtIndBase + vbtSalto ] = pldReal[ vbtIndBase ] - vldReal;
        pldImg [ vbtIndBase + vbtSalto ] = pldImg [ vbtIndBase ] - vldImg;

        //
        // vComplexo[ vbtIndBase ] = vComplexo[ vbtIndBase ] + vCCalc;
        //

        pldReal[ vbtIndBase ] = pldReal[ vbtIndBase ] + vldReal;
        pldImg [ vbtIndBase ] = pldImg [ vbtIndBase ] + vldImg;

        vbtIndBase++;
        if ( ( vbtIndBase % vbtSalto ) == 0 )
            vbtIndBase += vbtSalto;
    }

    vbtRN = vbtRN << 1;
    vbtSalto = vbtSalto << 1;
}

//
// Preencher os vetores de retorno
//

*p_wTam = (word)ceil( (float)v_wTamSinal / 2 );
*p_fAmplitude = new float[ *p_wTam ];
*p_fFase = new float[ *p_wTam ];

for ( vwIndice = 0; vwIndice < *p_wTam; vwIndice++ )
{
    (*p_fAmplitude)[ vwIndice ] =
        sqrtl( powl(pldReal[ vwIndice ],2) + powl(pldImg[ vwIndice ],2) );

    if ( (*p_fAmplitude)[ vwIndice ] < 0.001 )
    {
        (*p_fFase)[ vwIndice ] = 0;
        continue;
    }

    if ( pldReal[ vwIndice ] == 0 )
    {
        vldReal = 0;
    }
    else
    {
        vldReal =
            atanl( pldImg[ vwIndice ]/pldReal[ vwIndice ] );

        (*p_fFase)[ vwIndice ] = (float)vldReal*((long double)180.00/(long double)M_PI);
    }
}
//

```

```
        // A decomposição está em relação a uma função cosseno.  
        // Para transferi-la com relação a uma função seno, basta adicionar 90°  
        //  
        (*p_fFase)[ vwIndice ] += 90;  
    }  
}  
  
delete [] pldImg;  
delete [] pldReal;  
}
```

Apêndice D

Coeficientes dos Filtros de Decomposição

Os coeficientes de alguns dos filtros de decomposição utilizados pela implementação são apresentados na Tabela 23.

Tabela 23 - Coeficientes dos filtros de decomposição de algumas famílias de funções *wavelets*.

Família	$h_0[]$	$h_1[]$
Haar	0.7071067811865476 0.7071067811865476	-0.7071067811865476 0.7071067811865476
Daubechies 2	-0.1294095225509215 0.2241438680418574 0.8365163037374690 0.4829629131446903	-0.4829629131446903 0.8365163037374690 -0.2241438680418574 -0.1294095225509215
Daubechies 4	-0.0105974017849973 0.0328830116669829 0.0308413818359870 -0.1870348117188811 -0.0279837694169838 0.6308807679295904 0.7148465705525415 0.2303778133088552	-0.2303778133088552 0.7148465705525415 -0.6308807679295904 -0.0279837694169838 0.1870348117188811 0.0308413818359870 -0.0328830116669829 -0.0105974017849973
Coyfilets 1	-0.0156557281354645 -0.0727326195128539 0.3848648468642029 0.8525720202122554 0.3378976624578092 -0.0727326195128539	0.0727326195128539 0.3378976624578092 -0.8525720202122554 0.3848648468642029 0.0727326195128539 -0.0156557281354645

Código-fonte da Transformada Wavelet Discreta

O código-fonte da transformada *wavelet* discreta é apresentada na Listagem 5. Os demais códigos-fonte são omitidos por não conterem informações importantes para esta dissertação.

Listagem 5 - Código-fonte da transformada *wavelet* discreta

```

/*-----*/
* vDWT *
*-----*/
*
* Finalidade: *
*   Calcular a transformada Wavelet discreta para decompor *
*   o sinal de entrada em um sinal de aproximação (A) e em *
*   um sinal de detalhes (D). *
* *
*   Os seguintes passos são efetuados: *
*   1 - O sinal de entrada é expandido para que haja *
*       simetria com relação aos seus início e fim, a fim *
*       de minimizar os efeitos das bordas do sinal; *
* *
*   2 - Aplica-se, na seqüência, uma etapa do algoritmo de *
*       de decomposição piramidal através de filtros *
*       espelhos conjugados. *
* *
* Parâmetros: *
* p_fSinal: entrada. *
*   Ponteiro para o sinal de entrada. *
* *
* v_wTamSinal: entrada. *
*   Número de amostras presentes no sinal. *
* *
* p_fFiltroLo: entrada. *
*   Ponteiro para o filtro passa-baixas. *
* *
* p_fFiltroHi: entrada. *
*   Ponteiro para o filtro passa-altas. *
* *
* v_wTamFiltro: entrada. *
*   Número elementos presentes nos filtros. *
* *
* p_fA: saída. *
*   Ponteiro para o sinal de aproximação. *
* *
* p_wTamA: saída. *
*   Ponteiro para variável a receber o número de *
*   elementos do sinal de aproximação. *
* *
* p_fD: saída. *
*   Ponteiro para o sinal de detalhes. *
* *
* p_wTamD: saída. *
*   Ponteiro para variável a receber o número de *
*   elementos do sinal de detalhes. *
* *
* v_bExpZero: saída. *
*   Flag de expansão do sinal. *
* *
* Retorno: *
*   Não há. *
* *
*-----*/
void vDWT( float *p_fSinal,
           word v_wTamSinal,
           float *p_fFiltroLo,
           float *p_fFiltroHi,
           word v_wTamFiltro,

```

```

        float **p_fA,
        word *p_wTamA,
        float **p_fD,
        word *p_wTamD,
        bool v_bExpZero )
{
    word    vwIndice,
           vwIndice2,
           vwTamSinal;

    float   *pfSinal,
           *pfConvolucao;

    //
    // Validar os parâmetros de entrada
    //

    *p_wTamA = 0;
    *p_wTamD = 0;

    if ( ( v_wTamFiltro == 0 ) || ( v_wTamSinal == 0 ) )
    {
        return;
    }

    //
    // Expandir o sinal de entrada para que haja
    // simetria com relação a seu início e a seu
    // fim.
    //

    vwTamSinal = (v_wTamFiltro<<1) + v_wTamSinal;

    pfSinal = new float[ vwTamSinal ];

    for ( vwIndice = 1; vwIndice <= v_wTamFiltro; vwIndice++ )
    {
        if ( v_bExpZero == true )
        {
            pfSinal[v_wTamFiltro-vwIndice] = 0;
            pfSinal[v_wTamSinal+v_wTamFiltro+vwIndice-1] = 0;
            continue;
        }

        pfSinal[v_wTamFiltro-vwIndice] = p_fSinal[vwIndice];
        pfSinal[v_wTamSinal+v_wTamFiltro+vwIndice-1] = p_fSinal[v_wTamSinal-vwIndice-1];
    }

    memcpy( (void *)&pfSinal[v_wTamFiltro],
           (void *)&p_fSinal[0],
           v_wTamSinal*sizeof( float ) );

    //
    // Obter os sinais de aproximação e de detalhe
    //

    //
    // Convoluir o sinal expandido com o filtro
    // passa-baixas, a fim de obter o sinal de aproximação
    //
    // No entanto, essa convolução despreza "v_wTamFiltro-1" pontos
    // ao início e ao fim, visto que tais pontos são desnecessários
    // por conterem o problema das bordas (transitórios).
    //
    // Também aplicar uma amostragem a cada 2 amostras (downsampling).
    //

    *p_fA = new float[ vwTamSinal + v_wTamFiltro - 1 ];
    *p_fD = new float[ vwTamSinal + v_wTamFiltro - 1 ];

    for ( vwIndice = (v_wTamFiltro+1); vwIndice < ( vwTamSinal - 1 ); vwIndice += 2 )
    {
        (*p_fA)[ *p_wTamA ] = 0;
        (*p_fD)[ *p_wTamA ] = 0;

        for( vwIndice2 = 0; vwIndice2 < v_wTamFiltro; vwIndice2++ )
        {

```

```
        if ( vwIndice >= vwIndice2 )
        {
            (*p_fA)[ *p_wTamA ] += p_fFiltroLo[ vwIndice2 ] * pfSinal[ vwIndice - vwIndice2 ];
            (*p_fD)[ *p_wTamA ] += p_fFiltroHi[ vwIndice2 ] * pfSinal[ vwIndice - vwIndice2 ];
        }
        *p_wTamA = *p_wTamA + 1;
    }

    *p_wTamD = *p_wTamA;

    //
    // Liberar o sinal expandido
    //

    delete pfSinal;
}
```

Apêndice E

Arquivos de Configuração

Exemplos dos arquivos de configuração utilizados para se testar cada um das redes neurais utilizadas durante os testes do modelo são apresentados nas listagem que se seguem.

Listagem 6 - Configuração base utilizada pelos testes 1 e 2 citados na seção 8.3.

```
[REDE]
CAMADAS=3 ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=6 3 3

ATIVACAO=0 0 ; 0-unipolar, 1-linear

[TREINAMENTO]
ALGORITMO=1 ; algoritmo de levenberg-marquardt
EPOCAS=10 ; quantidade de épocas a treinar
ERRO=0.12345 ; erro mínimo a ser atingido. Usar ponto decimal '.'
AMOSTRAS=9 ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
PASSO=0.01 ; passo para o algoritmo de Levenber-marquardt
FATOR=1.2 ; fator para o algoritmo de Levenber-marquardt
APRENDIZADO=3 2 ; taxas de aprendizado. Separar com espaço " ".

[AMOSTRA]
0.51918 0.00235 0.01023 0.02030 0.05467 0.39327 ; TV CRT 12"
0.50603 0.00201 0.00732 0.01411 0.03340 0.43713 ; TV CRT 15"
0.49188 0.00200 0.00651 0.02214 0.05155 0.42593 ; TV CRT 21"
0.74301 0.00784 0.00867 0.00799 0.10222 0.13027 ; LFC 11W
0.61140 0.01044 0.08158 0.05110 0.12408 0.12140 ; LFC 20W
0.53905 0.01533 0.01646 0.07857 0.09614 0.25445 ; LFC 25W
0.88453 0.00061 0.00202 0.00722 0.02886 0.07676 ; SOM 1 modo cd
0.91325 0.00040 0.00116 0.00444 0.01714 0.06361 ; SOM 3
0.91080 0.00068 0.00119 0.00414 0.01674 0.06646 ; SOM 2 modo fita

[ALVO]
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.9 0.1 0.1 ;SOM
```

```
0.9 0.1 0.1 ;SOM
0.9 0.1 0.1 ;SOM
```

Listagem 7 - Configuração utilizada pelo teste 5 citado na seção 8.3.

```
[REDE]
CAMADAS=3 ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=8 6 3

ATIVACAO=0 0 ; 0-unipolar, 1-linear

[TREINAMENTO]
ALGORITMO=1 ; algoritmo de levenberg-marquardt
EPOCAS=10 ; quantidade de épocas a treinar
ERRO=0.12345 ; erro mínimo a ser atingido. Usar ponto decimal '.'
AMOSTRAS=9 ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
PASSO=0.01 ; passo para o algoritmo de Levenber-marquardt
FATOR=1.2 ; fator para o algoritmo de Levenber-marquardt
APRENDIZADO=3 2 ; taxas de aprendizado. Separar com espaço " ".

[AMOSTRA]
0.18869 0.00012 0.00068 0.01210 0.02139 0.28403 0.25297 0.24001 ; TV CRT 12"
0.31965 0.00019 0.00088 0.00479 0.02705 0.30510 0.23860 0.10373 ; TV CRT 20"
0.26510 0.00021 0.00093 0.00517 0.02978 0.34956 0.19440 0.15485 ; TV CRT 21"
0.38153 0.01122 0.00869 0.14488 0.08799 0.06670 0.27941 0.01958 ; LFC 11W
0.38222 0.00927 0.04127 0.06953 0.17567 0.06518 0.24130 0.01556 ; LFC 20W
0.53152 0.01028 0.00842 0.01741 0.04824 0.26070 0.09506 0.02837 ; LFC 25W
0.20913 0.00016 0.00009 0.00094 0.01004 0.02954 0.17145 0.57865 ; SOM 1 em modo cd
0.29570 0.00042 0.00017 0.00162 0.00188 0.04580 0.05548 0.59892 ; SOM 3
0.24195 0.00119 0.00049 0.00265 0.00175 0.03342 0.12249 0.59606 ; SOM 2 em modo fita

[ALVO]
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.9 0.1 0.1 ;SOM
0.9 0.1 0.1 ;SOM
0.9 0.1 0.1 ;SOM
```

Listagem 8 - Configuração utilizada pelo primeiro teste citado na seção 8.4.

```
[REDE]
CAMADAS=3 ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=8 7 3

ATIVACAO=0 0 ; 0-unipolar, 1-linear

[TREINAMENTO]
ALGORITMO=1 ; algoritmo de levenberg-marquardt
EPOCAS=10 ; quantidade de épocas a treinar
ERRO=0.12345 ; erro mínimo a ser atingido. Usar ponto decimal '.'
AMOSTRAS=9 ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
PASSO=0.01 ; passo para o algoritmo de Levenber-marquardt
FATOR=1.2 ; fator para o algoritmo de Levenber-marquardt
APRENDIZADO=3 2 ; taxas de aprendizado. Separar com espaço " ".

[AMOSTRA]
0.24371 0.00991 0.00798 0.15350 0.14989 0.12616 0.13698 0.17188 ; L
0.42314 0.00068 0.00253 0.01063 0.04876 0.12172 0.38675 0.00579 ; O
0.24491 0.00219 0.00745 0.02130 0.03331 0.41443 0.25191 0.02451 ; M
0.49591 0.00171 0.00187 0.03053 0.03338 0.05861 0.36852 0.00947 ; LO
0.34708 0.00421 0.00903 0.03808 0.04389 0.17492 0.28707 0.09573 ; ML
0.36639 0.00082 0.00322 0.00982 0.03008 0.24242 0.34654 0.00072 ; MO
0.42997 0.00099 0.00273 0.01806 0.02162 0.15253 0.35328 0.02082 ; MLO

[ALVO]
0.1 0.1 0.9 ;L
0.1 0.9 0.1 ;O
0.1 0.9 0.9 ;M
```

```

0.9 0.1 0.1 ;LO
0.9 0.1 0.9 ;ML
0.9 0.9 0.1 ;MO
0.9 0.9 0.9 ;MLO

```

Listagem 9 - Configuração utilizada pelo segundo teste citado na seção 8.4.

```

[REDE]
CAMADAS=3 ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=8 30 4

ATIVACAO=0 0 ; 0-unipolar, 1-linear

[TREINAMENTO]
ALGORITMO=1 ; algoritmo de levenberg-marquardt
EPOCAS=10 ; quantidade de épocas a treinar
ERRO=0.12345 ; erro mínimo a ser atingido. Usar ponto decimal '.'
AMOSTRAS=9 ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
PASSO=0.01 ; passo para o algoritmo de Levenber-marquardt
FATOR=1.2 ; fator para o algoritmo de Levenber-marquardt
APRENDIZADO=3 2 ; taxas de aprendizado. Separar com espaço " ".

[AMOSTRA]
0.24371 0.00991 0.00798 0.15350 0.14989 0.12616 0.13698 0.17188 ; L
0.42314 0.00068 0.00253 0.01063 0.04876 0.12172 0.38675 0.00579 ; O
0.24491 0.00219 0.00745 0.02130 0.03331 0.41443 0.25191 0.02451 ; M
0.49591 0.00171 0.00187 0.03053 0.03338 0.05861 0.36852 0.00947 ; LO
0.34708 0.00421 0.00903 0.03808 0.04389 0.17492 0.28707 0.09573 ; ML
0.36639 0.00082 0.00322 0.00982 0.03008 0.24242 0.34654 0.00072 ; MO
0.42997 0.00099 0.00273 0.01806 0.02162 0.15253 0.35328 0.02082 ; MLO
0.40938 0.00072 0.00289 0.00860 0.02906 0.06207 0.37302 0.11426 ; G
0.44753 0.01366 0.02421 0.06995 0.08093 0.05102 0.28300 0.02970 ; GL
0.43597 0.00061 0.00238 0.00983 0.04275 0.09719 0.39374 0.01753 ; GO
0.50461 0.00269 0.00367 0.01923 0.02693 0.05113 0.39150 0.00024 ; GOL
0.39038 0.00078 0.00301 0.00945 0.02744 0.20668 0.36205 0.00020 ; GMO
0.45056 0.00092 0.00229 0.01483 0.02151 0.14262 0.36050 0.00677 ; GMOL
0.31200 0.00178 0.00666 0.01502 0.02629 0.33795 0.29794 0.00236 ; MG
0.40630 0.00215 0.00483 0.03289 0.03249 0.15456 0.32399 0.04279 ; MLG

[ALVO]
0.1 0.1 0.1 0.9 ;L
0.1 0.1 0.9 0.1 ;O
0.1 0.1 0.9 0.9 ;M
0.1 0.9 0.1 0.1 ;LO
0.1 0.9 0.1 0.9 ;ML
0.1 0.9 0.9 0.1 ;MO
0.1 0.9 0.9 0.9 ;MLO
0.9 0.1 0.1 0.1 ;G
0.9 0.1 0.1 0.9 ;GL
0.9 0.1 0.9 0.1 ;GO
0.9 0.1 0.9 0.9 ;GOL
0.9 0.9 0.1 0.1 ;GMO
0.9 0.9 0.1 0.9 ;GMOL
0.9 0.9 0.9 0.1 ;MG
0.9 0.9 0.9 0.9 ;MLG

```

Listagem 10 - Configuração base utilizada pelos testes 1 e 2 citados na seção 8.5.

```

[REDE]
CAMADAS=3 ; número de camadas na rede, incluída a entrada (4, no máximo)
ARQUITETURA=8 6 3

ATIVACAO=0 0 ; 0-unipolar, 1-linear

[TREINAMENTO]
ALGORITMO=1 ; algoritmo de levenberg-marquardt
EPOCAS=10 ; quantidade de épocas a treinar

```

```

ERRO=0.12345                ; erro mínimo a ser atingido. Usar ponto decimal '.'
AMOSTRAS=9                  ; quantidade de pares (Q,T) em cada uma das sessões ENTRADA e ALVO
PASSO=0.01                  ; passo para o algoritmo de Levenber-marquardt
FATOR=1.2                    ; fator para o algoritmo de Levenber-marquardt
APRENDIZADO=3 2             ; taxas de aprendizado. Separar com espaço " ".

[AMOSTRA]
0.993931 0.005406 0.000554 0.000080 0.000021 0.000004 0.000002 0.000003 ; TV CRT 12"
0.995157 0.003992 0.000597 0.000121 0.000069 0.000038 0.000017 0.000008 ; TV CRT 20"
0.996611 0.002675 0.000401 0.000160 0.000063 0.000031 0.000027 0.000031 ; TV CRT 21"
0.908114 0.070293 0.012507 0.004102 0.002734 0.000983 0.000807 0.000461 ; LFC 11W
0.912938 0.053711 0.022798 0.005143 0.003393 0.001172 0.000454 0.000392 ; LFC 20W
0.953163 0.023343 0.008173 0.003500 0.005443 0.004429 0.000869 0.001081 ; LFC 25W
0.999523 0.000337 0.000042 0.000035 0.000020 0.000014 0.000013 0.000016 ; SOM 1 em modo cd
0.999686 0.000194 0.000027 0.000022 0.000017 0.000023 0.000017 0.000013 ; SOM 3
0.999442 0.000092 0.000092 0.000088 0.000068 0.000077 0.000088 0.000054 ; SOM 2 em modo fita

[ALVO]
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.1 0.9 ;TV
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.1 0.9 0.1 ;LFC
0.9 0.1 0.1 ;SOM
0.9 0.1 0.1 ;SOM
0.9 0.1 0.1 ;SOM

```