



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**ÁTILA GIRÃO DE OLIVEIRA**

**CLASSIFICADORES NEURAIS APLICADOS NA DETECÇÃO DE CURTO-CIRCUITO**  
**ENTRE ESPIRAS ESTATÓRICAS EM MOTORES DE INDUÇÃO TRIFÁSICOS**  
**ACIONADOS POR CONVERSORES DE FREQUÊNCIA**

**FORTALEZA**  
**2014**

ÁTILA GIRÃO DE OLIVEIRA

CLASSIFICADORES NEURAIIS APLICADOS NA DETECÇÃO DE CURTO-  
CIRCUITO ENTRE ESPIRAS ESTATÓRICAS EM MOTORES DE INDUÇÃO  
TRIFÁSICOS ACIONADOS POR CONVERSORES DE FREQUÊNCIA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Ricardo Silva Thé Pontes

Co-orientador: Prof. Dr. Cláudio Marques de Sá Medeiros

**FORTALEZA**

**2014**

Dados Internacionais de Catalogação na Publicação

Universidade Federal do Ceará

Biblioteca de Pós-Graduação em Engenharia - BPGE

---

O45c Oliveira, Átila Girão de.

Classificadores neurais aplicados na detecção de curto-circuito entre espirais estatóricas em motores de indução trifásicos acionados por conversores de frequência / Átila Girão de Oliveira. – 2014.

98 f. : il.color., enc. ; 30 cm.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2014.

Área de Concentração: Eletrônica de Potência e Acionamentos Elétricos.

Orientação: Prof. Dr. Ricardo Silva Thé Pontes.

Coorientação: Prof. Dr. Cláudio Marques de Sá Medeiros.

1. Engenharia elétrica. 2. Conversores de frequência. I. Título.

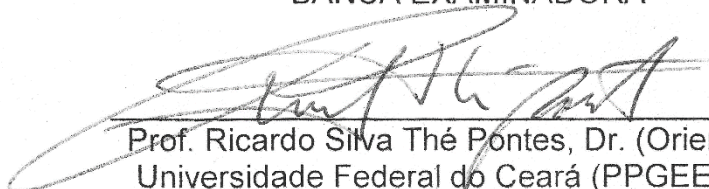
ÁTILA GIRÃO DE OLIVEIRA

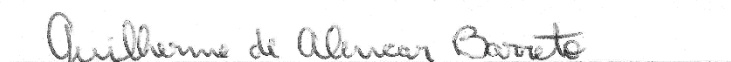
**CLASSIFICADORES NEURAI APLICADOS NA DETECÇÃO DE CURTO-CIRCUITO ENTRE ESPIRAS ESTATÓRICAS EM MOTORES DE INDUÇÃO TRIFÁSICOS ACIONADOS POR CONVERSORES DE FREQUÊNCIA**


Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistema de Energia Elétrica.

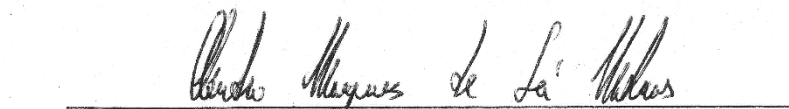
Aprovada em: 23/05/2014


**BANCA EXAMINADORA**

  
Prof. Ricardo Silva Thé Pontes, Dr. (Orientador)  
Universidade Federal do Ceará (PPGEE-UFC)

  
Prof. Guilherme de Alencar Barreto, Dr.  
Universidade Federal do Ceará (DETI-UFC)

  
Prof. Clayton Ricarte da Silva, Dr.  
Instituto Federal de Educação, Ciência e Tecnologia (IFCE)

  
Prof. Dr. Cláudio Marques de Sá Medeiros, Dr.  
Instituto Federal de Educação, Ciência e Tecnologia (IFCE)

  
Prof. Arthur Plínio de Souza Braga, Dr.  
Universidade Federal do Ceará (PPGEE-UFC)

Dedico ao meu filho recém-nascido,  
Lorenzo.

## AGRADECIMENTOS

Pai e Mãe, obrigado por sempre investirem na minha educação, reconheço todo o esforço que fizeram para permitir que eu pudesse seguir estudando. Desejo transformar meu sentimento de gratidão em ações de retribuição e deixá-los, cada dia, um pouco mais orgulhosos. Amo vocês.

Estendo minha gratidão à minha irmã e irmãos, avó, tios e tias, primos e primas, enfim, todos os parentes e “agregados” (a família entenderá), pois todos contribuem para criar um ambiente de cooperação e segurança. Sinto-me acolhido nessa grande família. Obrigado.

Um agradecimento especial deve ser feito a Lorena, que me apoia sempre e está comigo em todo momento. Estarei sempre à disposição, quando precisar. Amo-te.

Estas pessoas participam ou participaram da minha formação como pessoa, e eu não chegaria aqui sem seu suporte.

Agradeço também às pessoas que estiveram comigo nestes 2 anos de mestrado, participando diretamente neste processo.

Ao Prof. Cláudio Sá, obrigado por acreditar no meu potencial, ser meu orientador duas vezes, e trabalhar de maneira excepcional. Agradecido, também, por me ajudar em situações outras fora do trabalho acadêmico. Sou-lhe muito grato.

Minha gratidão ao Prof. Ricardo Thé, também meu orientador, pela grande confiança depositada em mim, e por me guiar sempre pelos melhores caminhos deste percurso.

Obrigado ao Rafael ‘*Thrash*’, ao Eduardo, ao Rodrigo ‘*Mestrim*’ e a todos do Lamotriz, por me ajudarem, ensinando-me sempre algo importante em diferentes momentos. Um agradecimento especial ao Tobias, por apresentar um artigo em meu lugar, no EEMODS.

Agradeço, igualmente, ao pessoal do LPE, no IFCE, com quem também estive compartilhando experiências (e vitaminas de Açaí) por todo este tempo. Um agradecimento especial ao Rennan, por me trazer mais trabalho ao tentar usar o PCA, pela primeira vez, na solução do problema de detecção.

Foi um período muito valioso para mim, e sinto-me feliz em ter vocês comigo. Sinto-me, realmente, sortudo nesta vida. Mais uma vez, obrigado!

*“Nossas virtudes e nossos defeitos são inseparáveis como a força e a matéria. Quando se separam, o homem não existe.”*

Nikola Tesla – (1856-1943).

## RESUMO

Este trabalho deriva da aplicação de redes neurais artificiais para a detecção de curto-circuito entre espiras em motor de indução trifásico, acionado por inversor de frequência. As redes neurais artificiais, do tipo Perceptron Simples e Multicamadas, são usadas para detectar falhas de curto-circuito no bobinamento estatórico de motores de indução trifásicos de forma *off-line*. Para treinamento do Perceptron Multicamadas são usados dois algoritmos distintos: o *error back-propagation*, que figura como o algoritmo clássico na literatura especializada, e o *extreme learning machine*, que é uma alternativa, relativamente recente, ao algoritmo clássico. Este algoritmo é uma opção atraente para o desenvolvimento rápido de classificadores. O banco de dados usado para treinamento e validação das redes é obtido a partir de experimentação laboratorial, portanto composto de dados reais. Os atributos utilizados para a detecção da falha são componentes de frequência do espectro harmônico da corrente estatórica do motor. O critério de escolha destas componentes, a priori, é fundamentado em resultados de investigações prévias da assinatura de corrente e, em segunda instância, é aplicada a técnica de análise de componentes principais. São apresentados os resultados obtidos pelos classificadores projetados, e feitas algumas considerações quanto à utilização destes em aplicação embarcada e em tempo real, que é a principal projeção de futuros trabalhos a partir do atual.

**Palavras-chave:** Curto circuito entre espiras; Motor de Indução Trifásico; Perceptron Multicamadas.



## ABSTRACT

This dissertation reports applications of artificial neural networks to detect stator winding interturn fault of three phase induction motor driven by frequency inverter. The artificial neural networks, like Simple and Multilayer Perceptron, served as off-line classifiers to short-circuit fault condition or healthy condition. In the training of Multilayer Perceptron, two different algorithms are used: the error back-propagation, which is a classic algorithm, and the extreme learning machine, as a relative new alternative for the classic back-propagation. The new one is more worthwhile because of its implementation easiness and higher speed of computation. The database used on the training and validation of the networks is created from an experimental setting, therefore it is composed by true data. The attributes used as failures' indicators are selected from certain frequencies of the spectrum, based on some theories of current signature analysis. In the second instance, the technique of principal components analysis is employed. The results obtained for the designed classifiers are shown, and some considerations are made on their use in real time embedded applications, which is the most important projection for future researches.

**Keywords:** Winding interturn short-circuit; Three Phase Induction Motor; Multilayer Perceptron.

## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| Figura 1. Principais subsistemas no isolamento do estator do motor de indução. ....   | 20 |
| Figura 2. Arquitetura do neurônio artificial da rede Perceptron. ....   | 32 |
| Figura 3. Modelo genérico de rede neural alimentada diretamente com única camada escondida. ...   | 34 |
| Figura 4. Visão geral das etapas de formação do conjunto de dados. ....   | 46 |
| Figura 5. Acoplamento Motor-Carga. ....   | 47 |
| Figura 6. Bobinamento estatórico da fase A do MIT. ....   | 48 |
| Figura 7. Esquema mostrando grupos de bobinas disponíveis após rebobinamento. ....  | 48 |
| Figura 8. Detalhe das derivações externas e bornes. ....  | 48 |
| Figura 9. Esquema de emulação de: (a) alta impedância e (b) baixa impedância. ....  | 49 |
| Figura 10. Sistema de aquisição de dados (A), sensores de efeito Hall (B) e sistema de filtragem e amplificação (C). ....                     | 50 |
| Figura 11. Representação dos conjuntos de dados formados. ....  | 51 |
| Figura 12. Gráfico ampliado de variâncias por componentes aproximadas do espectro. ....   | 66 |
| Figura 13. Gráfico da representatividade da variância total do conjunto de dados versus o número de atributos selecionados pela técnica. .... | 73 |
| Figura 14. Taxa de acerto média por número de neurônios no conjunto de treinamento. ....  | 80 |
| Figura 15. Taxa de acerto média por número de neurônios no conjunto de validação. ....  | 81 |
| Figura 16. Esquema para ensaio do motor na fase A. ....   | 94 |
| Figura 17. Desenho esquemático da placa de filtragem e amplificação. ....   | 98 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1. Medições do valor percentual de cada conjunto de bobinas. ....   | 47 |
| Tabela 2. Resumo de parâmetros da rede MLP usados nos testes para definição de topologia. ....   | 54 |
| Tabela 3. Resultados da rede MLP ao usar 100% dos dados para treinamento. ....   | 55 |
| Tabela 4. Resultados do conjunto de treinamento para teste de generalização da rede MLP. ....  | 55 |
| Tabela 5. Resultados do conjunto de validação para teste de generalização da rede MLP. ....  | 56 |
| Tabela 6. Resultados do conjunto de validação por teste de normalização. ....  | 57 |
| Tabela 7. Resultados no ajuste do termo de momento. ....   | 58 |
| Tabela 8. Resultados no ajuste dos limites da taxa de aprendizagem. ....   | 59 |
| Tabela 9. Divisão do conjunto de dados para treinamento. ....  | 60 |
| Tabela 10. Resultados por ajuste na divisão dos conjuntos de treinamento, teste e validação. ....  | 60 |
| Tabela 11. Taxas de acerto obtidas em relação ao número de épocas utilizadas. ....   | 61 |
| Tabela 12. Resultados da rede ELM ao usar 100% dos dados para treinamento. ....  | 62 |
| Tabela 13. Resultados para teste de generalização da rede ELM de 5 a 115 neurônios. ....   | 63 |
| Tabela 14. Resultados para teste de generalização da rede ELM de 5 a 25 neurônios. ....  | 63 |
| Tabela 15. Resultados após seleção de atributos da rede MLP/ELM. ....  | 64 |
| Tabela 16. Combinações de atributos testadas em busca de melhor acerto de classificação. ....  | 68 |
| Tabela 17. Resultados do Perceptron Simples. ....  | 69 |
| Tabela 18. Resultados globais médios das <i>MLPs</i> . ....  | 70 |
| Tabela 19. Taxas de classificação média por classe. ....   | 70 |
| Tabela 20. Resultado para RNAs específicas. ....   | 71 |
| Tabela 21. Taxa de classificação usando Perceptron Simples com diferentes quantidades de atributos obtidos através do PCA. ....                    | 74 |
| Tabela 22. Matriz de Confusão considerando todas as subclasses. ....   | 76 |
| Tabela 23. Taxa de classificação usando Perceptron Simples, usando 7 rótulos, com diferentes quantidades de atributos obtidos através do PCA. .... | 76 |
| Tabela 24. Quantidade de dados obtidos após adição de ruídos tomando 80% do conjunto de falha como base para o treinamento. ....                   | 77 |
| Tabela 25. Taxas de acerto em relação ao ruído adicionado. ....  | 78 |
| Tabela 26. Taxas de classificação por classe para várias quantidades de neurônios ocultos na rede <i>MLP/ELM</i> . ....                            | 79 |
| Tabela 27. Custo computacional da série de Taylor. ....  | 84 |
| Tabela 28. Comparação de custo computacional entre 4 classificadores selecionados. ....  | 86 |
| Tabela 29. Medições de tensão e valor percentual para cada conjunto de bobinas na 'fase A'. ....   | 95 |
| Tabela 30. Módulo das frequências dados pela equação (B.1) quando $p = 2, f_1 = 60$ Hz. ....   | 96 |

## LISTA DE ABREVIATURAS E SIGLAS

|             |   |
|-------------|---|
| AI          | Alta Impedância.  |
| AT          | Combinação de Atributos.                                |
| BI          | Baixa Impedância.                                       |
| <i>BP</i>   | <i>error Backpropagation.</i>                           |
| CCEEE       | Curto Circuito entre Espiras Estatóricas.               |
| CR          | <i>Classification Rate.</i>                             |
| <i>DWT</i>  | <i>Discrete Wavelet Transform.</i>                      |
| <i>ELM</i>  | <i>Extreme Learning Machine.</i>                        |
| <i>EPVA</i> | <i>Extended Park's Vector Approach.</i>                 |
| <i>FFT</i>  | <i>Fast Fourier Transform.</i>                          |
| <i>LVM</i>  | <i>Levenberg-Marquadt.</i>                              |
| <i>MCSA</i> | <i>Motor Current Signature Analysis.</i>                |
| <i>MLP</i>  | <i>Multi-Layer Perceptron.</i>                          |
| MQ          | Mínimos Quadrados.                                      |
| MQR         | Mínimos Quadrados Recursivos.                           |
| <i>PSD</i>  | <i>Power Spectral Density.</i>                          |
| RF          | Rádio Frequência.                                       |
| RNAs        | Redes Neurais Artificiais.                              |
| <i>SLFN</i> | <i>Single hidden-Layer Feed-forward Neural network.</i> |
| <i>SOM</i>  | <i>Self-Organized Map.</i>                              |
| <i>SVM</i>  | <i>Support Vector Machine.</i>                          |

## LISTA DE SÍMBOLOS

|                        |   |
|------------------------|---|
| $f_{cc}$               | Componentes harmônicas de frequência em função do curto-circuito entre espiras. |
| $k$                    | Ordem das harmônicas temporais.   |
| $n$                    | Ordem das harmônicas espaciais.   |
| $s$                    | Escorregamento.   |
| $p$                    | Número de pares de polos.   |
| $f_1$                  | Frequência fundamental da tensão de alimentação.                                |
| $\mathbf{d}$           | Vetor de saída desejada.  |
| $\varphi(\cdot)$       | Função qualquer que relaciona um espaço de entrada a um espaço de saída.        |
| $\mathbf{x}$           | Vetor de entrada qualquer.  |
| $\hat{\varphi}(\cdot)$ | Função estimada de $\varphi(\cdot)$ .   |
| $\mathbf{y}$           | Vetor de saída dado pela função $\hat{\varphi}(x)$ .                            |
| $t$                    | Índice da iteração.   |
| $\mathbf{x}(t)$        | Vetor de entrada qualquer na iteração $t$ .                                     |
| $x_j(t)$               | Componente qualquer do vetor de entrada $\mathbf{x}(t)$ .                       |
| $q$                    | Quantidade de neurônios.  |
| $\mathbf{d}(t)$        | Vetor saída desejada na iteração $t$ .  |
| $d_i(t)$               | Elemento qualquer do vetor $\mathbf{d}(t)$ .                                    |
| $\mathbf{w}_i(t)$      | Vetor de pesos que conecta a entrada aos neurônios.                             |
| $w_{ij}$               | Peso sináptico que conecta $x_j$ ao $i$ -ésimo neurônio.                        |
| $\theta_i$             | Límiar associado ao $i$ -ésimo neurônio.  |
| $u_i(t)$               | Ativação do $i$ -ésimo neurônio.  |
| $y_i(t)$               | Saída do $i$ -ésimo neurônio na iteração $t$ .                                  |
| $\mathbf{z}$           | Redefinição de $\mathbf{x}$ .   |
| $J[\mathbf{w}]$        | Função objetivo, ou função custo.   |
| $\mathbf{z}_k$         | $k$ -ésimo vetor mal classificado.  |
| $Z$                    | Conjunto de vetores mal classificados.  |
| $\mathbf{w}$           | Vetor de pesos para classificador perceptron envolvendo duas classes.           |
| $w^{novo}$             | Valor $w$ após ajuste.  |

|                     |   |
|---------------------|---|
| $w^{atual}$         | Valor atual de $w$ .  |
| $\Delta w$          | Gradiente de $w$ .  |
| $\eta$              | Taxa ou passo de aprendizagem.  |
| $e(t)$              | Vetor de erro na iteração $t$ .   |
| $e_i$               | Erro do $i$ -ésimo neurônio de saída.   |
| $W$                 | Matriz de parâmetros ou pesos.  |
| $\hat{W}$           | Matriz de parâmetros ou pesos aproximados.  |
| $X$                 | Matriz de todos os vetores de entrada disponíveis.                                    |
| $D$                 | Matriz de todos os vetores de saída desejada.   |
| $y_{ts}$            | Saída dada a partir de um novo vetor de amostra não visto antes.                      |
| $x_{ts}$            | Novo vetor de amostra não visto antes pelo estimador.                                 |
| $Y_{ts}$            | Matriz formada por cada um dos $n$ vetores de saída $y_{ts}$ .                        |
| $X_{ts}$            | Matriz formada por cada um dos $n$ vetores de entrada $x_{ts}$ .                      |
| $\psi$              | Matriz transposta de $X$ .  |
| $n$                 | Número de vetores de amostra.   |
| $\psi(i)$           | $i$ -ésimo elemento tomado dos vetores de amostra.                                    |
| $P_N$               | Definição usada na prova dos mínimos quadrados recursivos.                            |
| $K_N$               | Definição usada na prova dos mínimos quadrados recursivos.                            |
| $\alpha(N)$         | Definição usada na prova dos mínimos quadrados recursivos.                            |
| $P_0$               | Valor inicial de $P_N$ para implementação do método dos mínimos quadrados recursivos. |
| $u_i^{(h)}(t)$      | Ativação do $i$ -ésimo neurônio da camada escondida na iteração $t$ .                 |
| $q$                 | Número de neurônios na camada escondida.  |
| $p$                 | Dimensão do vetor de entrada (excluindo o limiar).                                    |
| $\theta_i^{(h)}(t)$ | Limiar do neurônio escondido $i$ na iteração $t$ .                                    |
| $y_i^{(h)}(t)$      | Saída do $i$ -ésimo neurônio da camada escondida na iteração $t$ .                    |
| $u_i^{(h)}$         | Ativação do $i$ -ésimo neurônio da camada escondida na iteração $t$ .                 |
| $\varphi_i(\cdot)$  | Função de ativação do $i$ -ésimo neurônio da camada escondida na iteração $t$ .       |
| $y_k^{(o)}(t)$      | Saída do $i$ -ésimo neurônio da camada de saída na iteração $t$ .                     |
| $u_k^{(o)}(t)$      | Ativação do $i$ -ésimo neurônio da camada de saída na iteração $t$ .                  |

|                       |   |
|-----------------------|---|
| $\varphi_k(\cdot)$    | Função de ativação do $i$ -ésimo neurônio da camada de saída na iteração $t$ .                        |
| $m_{ki}$              | Peso que conecta o $i$ -ésimo neurônio da camada escondida ao $k$ -ésimo neurônio da camada de saída. |
| $m$                   | Número de neurônios da camada de saída.   |
| $\theta_k^{(o)}(t)$   | Limiar do neurônio de saída $k$ na iteração $t$ .   |
| $e_k^{(o)}(t)$        | Erro do $k$ -ésimo neurônio de saída na iteração $t$ .  |
| $d_k(t)$              | Saída desejada para o $k$ -ésimo neurônio de saída na iteração $t$ .                                  |
| $\delta_k^{(o)}(t)$   | Gradiente local do $k$ -ésimo neurônio de saída na iteração $t$ .                                     |
| $\delta_i^{(h)}(t)$   | Gradiente local do $i$ -ésimo neurônio oculto na iteração $t$ .                                       |
| $\varepsilon_{train}$ | Erro quadrático médio.  |
| $M$                   | Matriz de todos os pesos que conectam a camada escondida à camada de saída.                           |
| $Y^{(h)}$             | Matriz formada por todos os vetores de saída da camada escondida.                                     |
| $N$                   | Número de pontos obtidos na discretização do espectro de frequência.                                  |

## SUMÁRIO

|  |    |
|--|----|
| 1 INTRODUÇÃO.....  | 15 |
| 1.1 Motivação.....   | 17 |
| 1.2 Objetivos .....  | 17 |
| 1.3 Produção Científica .....                                    | 18 |
| 1.4 Estrutura do Texto.....                                      | 18 |
| 2 FALHA DE CURTO-CIRCUITO NO BOBINAMENTO ESTATÓRICO .....        | 20 |
| 2.1 Fatores de causa de avaria nos enrolamentos estatóricos..... | 21 |
| 2.1.1 Envelhecimento Térmico.....                                | 21 |
| 2.1.2 Sobrecargas Térmicas.....                                  | 22 |
| 2.1.3 Esforços Dielétricos .....                                 | 23 |
| 2.1.4 Descargas Parciais .....                                   | 23 |
| 2.1.5 Transitórios na Tensão de Alimentação .....                | 24 |
| 2.1.6 Esforços Mecânicos .....                                   | 24 |
| 2.1.7 Fatores Ambientais .....                                   | 25 |
| 2.1.8 Considerações .....  | 25 |
| 2.2 Estado da Arte em Detecção de Falhas de CCEEE.....           | 26 |
| 3 REDES NEURAIS ARTIFICIAIS E CLASSIFICADORES.....               | 30 |
| 3.1 Perceptron.....  | 31 |
| 3.2 Perceptron Multicamadas.....                                 | 34 |
| 3.3 Algoritmos de Aprendizado.....                               | 35 |
| 3.3.1 Aprendizado do Perceptron .....                            | 35 |
| 3.3.2 Mínimos Quadrados.....                                     | 37 |
| 3.3.3 Mínimos Quadrados Recursivos.....                          | 40 |
| 3.3.4 Back-propagation .....                                     | 40 |
| 3.3.5 Extreme Learning Machine .....                             | 43 |
| 4 ENSAIOS PARA AQUISIÇÃO DE DADOS .....                          | 46 |
| 4.1 Descrição Geral da Bancada .....                             | 46 |
| 4.1.2 Conjunto de acionamento.....                               | 46 |
| 4.1.3 Emulação de Curto-circuito.....                            | 48 |
| 4.1.4 Sistema de Aquisição de Dados.....                         | 49 |
| 4.2 Conjuntos de Dados .....                                     | 50 |
| 5 SELEÇÃO DE TOPOLOGIAS E DE ATRIBUTOS .....                     | 53 |



|   |    |
|---|----|
| 5.1 Definição de Topologias .....   | 53 |
| 5.1.1 Topologia MLP/Back-propagation.....   | 53 |
| 5.1.2 Topologia MLP/ELM .....   | 61 |
| 5.2 Seleção de Atributos .....  | 64 |
| 5.2.1 Análise de Variâncias.....  | 65 |
| 5.2.2 Combinações de Atributos .....  | 66 |
| 6 ANÁLISE DOS RESULTADOS .....  | 69 |
| 6.1 Teste de separabilidade do conjunto de dados .....  | 69 |
| 6.2 Redes MLP aplicadas como classificadores.....   | 70 |
| 6.3 Aplicação do PCA.....   | 72 |
| 6.3.1 Perceptron Simples com extração de características a partir do PCA .....  | 74 |
| 6.4 Geração de novos dados para conjunto normal .....   | 76 |
| 6.4.1 Adição de Ruído.....  | 77 |
| 6.4.1 Algoritmo ELM usando mínimos quadrados recursivos.....  | 79 |
| 6.5 Estimativa de custo computacional .....   | 81 |
| 6.5.1 Custo computacional no PS.....  | 82 |
| 6.5.2 Custo computacional da MLP.....   | 82 |
| 6.5.3 Custo computacional da Seleção de atributos.....  | 85 |
| 6.5.5 Custo computacional nos classificadores.....  | 86 |
| 7 CONCLUSÃO .....   | 88 |
| 7.1 Futuros trabalhos .....   | 89 |
| REFERÊNCIAS .....   | 91 |
| APÊNDICE A – Ensaio para Determinação dos Percentuais de Espiras Acessíveis Através das Derivações Externas Após Rebobinamento do Motor. .... | 94 |
| APÊNDICE B – Análise do Espectro de Frequência Baseada na Teoria de Penman (1994) e Thomson (2001). ....                                      | 96 |
| ANEXO A – Placa de filtragem .....  | 98 |

# 1 INTRODUÇÃO

A máquina de indução trifásica (MIT) é a principal fonte de conversão eletromecânica nas indústrias. Segundo Thomson e Fenger (2001), em uma nação industrializada os motores de indução trifásicos podem demandar, tipicamente, entre 40% e 50% de toda a capacidade elétrica gerada no país.

No entanto, mesmo com a robustez e confiabilidade típicas desta máquina, ela não está isenta de falhas, que podem ocorrer devido a condições no ambiente de instalação, aplicação inadequada e falta de manutenção preventiva. As falhas mais comuns ocorrem nos rolamentos, nos isolamentos do estator ou do rotor (no caso de rotor bobinado), através da abertura de barras ou de rachadura de anéis (no caso de rotor do tipo gaiola de esquilo), e por excentricidade (Nandi; Toliyat; Xiaodong, 2005).

As falhas nas máquinas produzem sintomas, como tensão e corrente de linha desbalanceadas, aumento na pulsação de conjugado, queda no conjugado médio, aumento das perdas e redução na eficiência, e aquecimento excessivo (Nandi e Toliyat, 1999). A identificação dos sintomas no início da falha permite que a manutenção seja realizada antes da ocorrência de maiores prejuízos. Para isto, diversas são as técnicas para diagnosticar falhas. Nandi e Toliyat (1999) citam algumas: monitoramento do fluxo eletromagnético axial, medidas de temperatura, padrões de infravermelho, monitoramento de emissões de rádio frequência (RF), monitoramento de ruído e vibração, análises químicas, medidas de ruído acústico, análise de assinatura de corrente, e análise de modelos aliadas a técnicas de inteligência computacional como as redes neurais.

As falhas podem levar a paradas não programadas nas linhas de produção, o que geralmente resulta em elevados custos. Assim, o monitoramento contínuo e em tempo real, aliado à detecção prematura de falhas, apresenta-se como uma ferramenta poderosa na redução dos custos de manutenção, quando viabiliza o agendamento de ordens de serviços para manutenção preditiva. Em falhas como quebras de barras do rotor, excentricidade, e falhas de rolamento, que demoram um tempo apreciável para evoluir até um estágio mais danoso, o monitoramento constante dá condições para que os serviços de manutenção sejam agendados. No entanto, nem sempre a detecção prematura pode evitar uma parada não programada. Por exemplo, a falha por curto-circuito entre espiras do estator evolui rapidamente.

Thomson e Fenger (2001) fizeram um ensaio destrutivo, em um motor de indução trifásico de baixa tensão, aplicando um curto-circuito entre espiras estatóricas e detectaram que existe um tempo de apenas alguns minutos para a evolução da falha. Neste caso, a detecção prematura da falha permite uma rápida parada na operação, o que evita a geração de arcos elétricos e oferece uma proteção adicional em áreas onde há riscos de explosão. Outra vantagem, em caso de motores de alta potência, é a possibilidade de reparo do motor pela remoção das bobinas sob falha. Em motores de baixa potência, geralmente, não é viável o reparo parcial do bobinamento, ainda assim a detecção prematura do curto-circuito pode evitar danos permanentes no núcleo estatórico, por decorrência de arcos elétricos.

No caso de acionamentos de MIT através de conversores de frequência, notadamente em aplicações com rotação variável (Bezesky e Kreitzer, 2003), a detecção prematura de falhas, por curto-circuito, seguida de uma interrupção no fornecimento de energia para o motor pode evitar a ocorrência de surtos de corrente e estresse térmico nas chaves eletrônicas, conferindo, assim, um caráter redundante à proteção contra curto-circuito.

A constante evolução de técnicas de inteligência computacional, aliada à crescente capacidade computacional de processadores de uso industrial, tem criado condições objetivas para a implementação dos detectores de falhas embarcados nos próprios conversores de frequência, o que é uma grande vantagem, por aproveitar o dispositivo eletrônico, já existente, para criar um detector de falhas como uma proteção adicional. Neste sentido, percebe-se o interesse de pesquisadores através da publicação de trabalhos recentes, (Kowalski e Wolkiewicz, 2009, Hyun *et al.*, 2010, Das; Purkait e Chakravorti, 2012, Coelho e Medeiros, 2013).

Nesta dissertação, são testados classificadores neurais Perceptron Simples (PS) e Perceptron Multi-Camadas (*Multi-layer Perceptron – MLP*) na detecção de falhas, por curto-circuito, entre espiras estatóricas (CCEEE) de um motor de indução trifásico acionado por um conversor de frequência. As redes *MLP* são testadas com dois algoritmos de treinamento: retro-propagação do erro (*Back-propagation – BP*) e Máquina de Aprendizado Extremo (*Extreme Learning Machine – ELM*).

A seleção de atributos para os classificadores é baseada no princípio da análise de assinatura de correntes, e define atributos em função de teorias relacionadas às falhas de CCEEE em motores de indução.

A aquisição de dados é realizada em uma bancada de testes que inclui um MIT rebobinado para permitir aplicação de vários níveis de CCEEE, um conversor de frequência para acionamento, e um freio de Foucault para a aplicação de carga ao motor. Através do sistema de aquisição de dados, faz-se a coleta de sinais de corrente das fases do motor e forma-se um banco de dados com as diversas condições dos ensaios.

## 1.1 Motivação

A falha de CCEEE destaca-se por representar 30 a 40% de todas as falhas que ocorrem em MIT (Nandi *et al.*, 2005). Detectá-la no seu início pode trazer uma série de vantagens, especialmente através do uso de uma técnica não invasiva. Por evoluir rapidamente, o monitoramento constante, em tempo real, é necessário para detectar a falha no seu início e amenizar suas consequências. Com o uso cada vez maior de conversores de frequência no acionamento das máquinas, surge a possibilidade de se criar um sistema que possa ser embarcado no processador de conversores comerciais, aproveitando a arquitetura eletrônica já existente, e evitando custos na criação de outro dispositivo para detecção. A detecção, através do princípio da análise de assinatura de correntes, permite aproveitarem-se os sensores, já presentes no conversor, na detecção do CCEEE (e, posteriormente, de outras falhas também). Desta forma, cria-se um sistema de detecção não invasivo, integrado ao dispositivo de acionamento do MIT.

A motivação desta pesquisa é o desenvolvimento de classificadores que possam ser, em futuros trabalhos, embarcados no conversor de frequência, durante o acionamento de MIT.

## 1.2 Objetivos

Abaixo segue uma descrição dos objetivos dessa dissertação:

- a) Detectar falhas de curto-circuito entre espiras, em motor de indução trifásico acionado por conversor de frequência, pela aplicação de redes

neurais, utilizando dados experimentais aquiridos a partir das correntes de linha;

- b) Investigar as potencialidades e limitações das redes utilizadas na solução do problema.

### 1.3 Produção Científica

Ao longo do desenvolvimento desta dissertação, os seguintes artigos científicos foram produzidos.

OLIVEIRA, A. G.; MEDEIROS, C. M. S.; PONTES, R. S. T. **Stator Winding Interturns Short Circuit Fault Detection in a Three Phase Induction Motor Driven by Frequency Converter Using Neural Networks**. Energy Efficiency in Motor Driven Systems 2013 Conference, EEMODS 2013. Rio de Janeiro – RJ.

OLIVEIRA, A. G.; PONTES, R. S. T. MEDEIROS, C. M. S.; **Neural Network used to Stator Winding Inter-turns Short-Circuit Fault Detection in an Induction Motor Driven By Frequency Converter**. BRICS-CCI & CBIC 2013, 1<sup>st</sup> Countries Congress on Computacional Intelligence and 11<sup>th</sup> Brazilian Congress on Computacional Intelligence. Porto de Galinhas – PE.

OLIVEIRA, A. G.; BESSA, R. COELHO, D., MEDEIROS, C. M. S.; **Redes Neurais utilizadas na Detecção de Falhas de curto-circuito entre espiras de um motor de indução trifásico**. XI Simpósio Brasileiro de Automação Inteligente e XI Conferência Brasileira de Dinâmica, Controle e Aplicações, SBAI - DINCON 2013. Fortaleza – CE.

### 1.4 Estrutura do Texto

O restante do trabalho se desenvolve como descrito a seguir.

No capítulo 2, são abordados detalhes sobre isolamento do motor e as causas de suas avarias mais comuns. Em seguida, é descrito o estado da arte relativo à detecção de falhas em motores de indução, especialmente em relação às falhas de curto-circuito entre espiras e métodos baseados em inteligência computacional.

No capítulo 3, são descritos cada um dos classificadores utilizados na dissertação e seus algoritmos, principalmente o algoritmo de retro-propagação do erro, amplamente conhecido como *backpropagation*, e o algoritmo *Extreme Learning Machine (ELM)*, que é usado na comparação de resultados e nos testes para seleção de atributos.

No capítulo 4, é descrita a bancada de testes, os equipamentos e o processo de coleta dos dados. No capítulo 5, é descrito o processo de seleção de atributos para as RNA. No capítulo 6, são detalhados e comparados os resultados

obtidos na classificação das redes treinadas pelos dois diferentes algoritmos. Em seguida, as conclusões do trabalho são apresentadas, no capítulo 7.

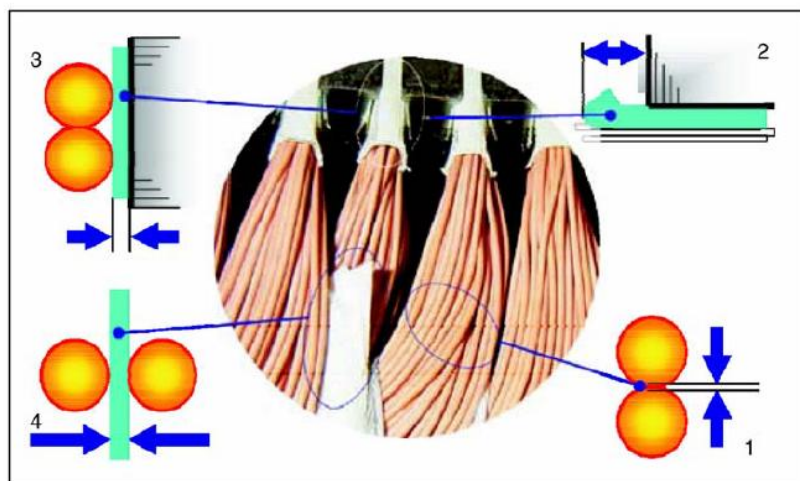
## 2 FALHA DE CURTO-CIRCUITO NO BOBINAMENTO ESTATÓRICO

Sistemas de isolamento em motores são submetidos a diversos tipos de esforços que podem causar curto-circuito. De acordo com Nandi *et al.* (2005), falhas de isolamento representam de 30% a 40% do total de falhas relatadas nos motores de indução. Devido ao uso de conversores de frequência no acionamento de motores elétricos, o estresse gerado chega a ser dez vezes maior do que o de máquinas acionadas diretamente pela rede (Kaufhold, 2002). Tratando-se, especificamente, de falhas associadas ao bobinamento estatórico, a operação com frequências de chaveamento típicas de 10 kHz sujeita o isolamento a esforços extras devidos a grandes variações da tensão em curtos intervalos de tempo.

O isolamento elétrico, em motores de indução, é constituído por subsistemas de isolamento, cada um deles submetido a esforços causados pela tensão entre as espiras das bobinas, pelos picos de tensão ocorridos e pela diferença de potencial entre as espiras e a terra. Os principais sistemas de isolamento, conforme Cruz (2004), podem ser visualizados na Figura 1, e são listados a seguir:

- 1) Isolamento entre camadas de condutores da mesma bobina;
- 2) Blindagem na zona das testas das bobinas;
- 3) Isolamento entre os condutores e a ranhura onde estão alojados;
- 4) Isolamento entre fases distintas.

Figura 1. Principais subsistemas no isolamento do estator do motor de indução.



Fonte: Kaufhold, 2002.

Outro fator que aumenta o estresse no isolamento entre espiras, em motores de baixa tensão, é a falta de cuidado na montagem das bobinas e na acomodação das mesmas nas ranhuras. Desta forma, a primeira espira pode encontrar-se adjacente à última espira, por exemplo, criando, assim, uma maior diferença de potencial entre elas. Além disso, os condutores de seção circular, típicos destes motores, deixam pequenos espaços vazios entre si, e os campos elétricos que surgem podem causar descargas parciais. Esta situação agrava-se com o uso de conversores de frequência. Segundo Cruz (2004), nesta situação, o campo elétrico existente nesses pequenos espaços pode atingir valores da ordem dos 100 kV/mm, provocando ionização do ar e consequentes descargas elétricas.

A seguir são apresentadas as maiores causas de avarias nos subsistemas de isolamento em enrolamentos estatóricos.

## **2.1 Fatores de causa de avaria nos enrolamentos estatóricos**

Há diversos fatores que interagem para a ocorrência de falha no isolamento. Entretanto, para uma melhor análise das possíveis causas envolvidas, dividem-se tais fatores como de natureza térmica, elétrica, mecânica e ambiental. Apesar de muitos efeitos serem correlacionados, esta divisão permite relevar os aspectos individuais mais importantes. Nos itens seguintes, são descritos os fatores mais relevantes de causa de falha no isolamento, para motores de baixa tensão, segundo Cruz (2004).

### **2.1.1. Envelhecimento Térmico**

A expectativa de vida de motores de indução é avaliada conforme o Instituto Americano de Engenheiros Eletricistas 510 (AIEE Committe, 1959) para efeitos como períodos de temperatura excessiva, vibração, exposição à umidade, e sobretensão. Quanto ao efeito da temperatura no sistema de isolamento dos enrolamentos, chegou-se a resultados de redução de 40% no tempo de vida útil do motor, para um acréscimo de apenas 5°C na temperatura nominal de operação, para a classe de isolamento A.

A partir deste exemplo, pode-se perceber a importância da temperatura de funcionamento dos enrolamentos, para a durabilidade da máquina. Uma vez que o sistema de isolamento perde sua integridade física, ele não resistirá normalmente aos



esforços dielétricos e mecânicos, tornando o sistema mais vulnerável a todos os outros mecanismos de fadiga.

### **2.1.2 Sobrecargas Térmicas**

Cruz (2004) cita os diversos fatores mais significativos que podem levar o motor a operar sob condição de sobrecarga térmica:

- a) Variações na tensão do motor - de acordo com Cruz (2004) *apud* NEMA-MG (2011), motores são projetados para operar, satisfatoriamente, com variação de  $\pm 10\%$  da tensão nominal. Operar fora desta faixa poderia reduzir drasticamente a vida útil do motor, devido ao sobreaquecimento causado.
- b) Desequilíbrios na alimentação - pequenos desequilíbrios nas tensões de fase podem causar um aumento considerável de temperatura nos enrolamentos. Segundo o estudo de causas de falhas em motor de indução, feito por Bonnet e Soukup (1992), um desequilíbrio de 3,5% no sistema de tensões de alimentação do motor resulta, de forma aproximada, em um aumento de 25% na temperatura do enrolamento onde circula o maior valor de corrente. Mesmo desconsiderando outros efeitos, como desequilíbrio magnético e elevação do conteúdo harmônico na rede, percebe-se a importância de um sistema de alimentação equilibrado.
- c) Partida repetitiva - a corrente de partida do motor pode alcançar 5 a 8 vezes a corrente nominal. Se o motor for sujeito a diversas partidas em curto período de tempo, a temperatura do enrolamento vai se elevar rapidamente. Para acionar cargas com este tipo de característica deve-se ter cuidado na especificação do motor. Ademais, o calor gerado devido às partidas repetitivas causa dilatação térmica da isolação, o que pode tornar os materiais isolantes quebradiços em médio prazo. O projetista deve garantir que os materiais isolantes sejam flexíveis o suficiente para não se tornarem quebradiços, porém rígidos o suficiente para suportarem os esforços térmicos causados pela intensa ação eletromagnética a que são submetidos, especialmente durante a partida.

- d) Sobrecarga no eixo - além do fator de serviço presente nas máquinas elétricas, é costume se dimensionar o motor de forma que a potência da carga mecânica acoplada ao eixo seja 85% do valor nominal. Esta prática garante um aumento na margem de segurança em relação às sobrecargas que, ainda assim podem ocorrer. É estimado que a temperatura nos enrolamentos se eleva proporcionalmente ao quadrado da corrente de carga, além de causar esforços ao acoplamento mecânico do motor. Conclui-se que este é um fator bastante significativo na vida útil do motor.
- e) Ventilação obstruída - o calor gerado no motor é dissipado por condução, convecção e radiação. Qualquer empecilho ao fluxo de ar através ou ao redor do motor causará uma diminuição da dissipação do calor, acarretando um aumento na temperatura de operação.
- f) Temperatura ambiente - é importante que o motor esteja instalado em local arejado. Se estiver confinado em espaço fechado, ou com pouco fluxo de ar, o calor gerado pela máquina vai aquecer o ambiente ao redor, dificultando a dissipação de calor.

### **2.1.3 Esforços Dielétricos**

Dentre os diversos subsistemas de isolamento no motor, aquele com menor capacidade de isolamento é que determina, segundo a norma ABNT NBR IEC 60085:2012, a classe de isolamento empregada no motor. Alguns motores possuem um isolamento especial para suportar picos de tensão causados pelo uso de inversores, porém, se os esforços dielétricos, causados pela diferença de potencial entre materiais isolantes, forem excessivos eles podem gerar uma falha de curto-circuito entre espiras de uma mesma bobina, entre espiras de fases distintas, e entre espiras e a ranhura onde estão alojadas.

### **2.1.4. Descargas Parciais**

As descargas parciais são descargas intermitentes locais, resultantes da ionização dos gases, no meio, quando campos elétricos locais excedem valores críticos no sistema de isolamento. Existem três tipos de descargas parciais: a) descargas internas, que ocorrem nas cavidades entre as espiras; b) descargas na

superfície dos enrolamentos; c) descargas em zonas geométricas críticas com pontas ou curvas acentuadas. Os fatores que influenciam as descargas parciais são: umidade, temperatura, geometria dos materiais, espaços vazios, frequência da tensão de alimentação, espessura do dielétrico, e esforços mecânicos.

Elas podem tornar-se um sério problema, especialmente em bobinas operando com tensões nominais acima de 3 kV. No entanto, mesmo em motores de baixa tensão, quando são alimentados por conversores de frequência, há igual surgimento de descargas parciais, devido a grandes picos de tensão em curtos intervalos de tempo.

Como consequência destas descargas, há sobreaquecimento das zonas adjacentes, erosão, ou reações químicas diversas, resultando numa degradação do isolamento dos enrolamentos.

### **2.1.5. Transitórios na Tensão de Alimentação**

Variações de curta duração na tensão de alimentação são causas de redução na vida útil dos enrolamentos, ou mesmo de ocorrência de curto-circuito (entre espiras ou entre espira e núcleo). Os causadores de transitórios na tensão de alimentação são muitos, como por exemplo: a) curtos circuitos no barramento de alimentação; b) atuação de fusíveis limitadores de corrente; c) manobras em bancos capacitivos; d) descargas atmosféricas. O uso massivo de conversores de frequência no acionamento de máquinas de indução é uma causa de transitórios que merece destaque. As altas frequências de chaveamento fazem com que picos sejam aplicados em períodos da ordem de micro e nano segundos, o que aumenta, aproximadamente, 10 vezes os esforços no sistema de isolamento, se comparados a um motor alimentado diretamente pela rede elétrica (Kaufhold, 2002).

### **2.1.6 Esforços Mecânicos**

Há vários tipos de esforços mecânicos, que podem afetar o isolamento do motor e causar curto-circuito, dos quais se destacam:

- a) Movimento das bobinas - A corrente que circula pelas espiras produz um campo magnético que cria uma força magnética nos enrolamentos. Esta força é máxima durante os ciclos de partida e causam vibração com o dobro da frequência de alimentação (Bonnet e Soukup, 1992),

tanto na direção tangencial quanto na radial, principalmente durante as partidas, o que pode provocar a movimentação das bobinas e causar danos ao seu isolamento. Este efeito agrava-se, aumentando a probabilidade de enfraquecimento do sistema, para maiores velocidades de rotação, maiores comprimentos de bobina, maior frequência de partida e menor tempo de aceleração.

- b) Atrito no rotor - o contato do rotor com o estator, seja por falha no rolamento, por desbalanceamento no eixo ou por desalinhamento entre rotor e estator, pode degradar, continuamente, o isolamento. A severidade da degradação depende do nível de contato, da frequência com que ocorre o contato, e se o mesmo ocorre apenas durante as partidas ou em regime permanente.
- c) Causas variadas - outras situações podem ser encontradas capazes de causar uma falha no isolamento, tais como, peças que atingem o estator, lâminas soltas do rotor, porcas, parafusos ou partículas externas que entram pela ventilação, e falhas no rotor que causam sobreaquecimento nos enrolamentos do estator.

### **2.1.7 Fatores Ambientais**

O ambiente em que se encontra o motor pode ser uma fonte de agentes contaminantes, como poeira, umidade, ferrugem e produtos químicos, por exemplo. Estes agentes contribuem para maior degradação do sistema de isolamento, especialmente a umidade, que pode facilitar a circulação de correntes na superfície das espiras, em caso de um início de curto-circuito entre espiras.

### **2.1.8 Considerações**

O processo de falha inicia-se, normalmente, como um curto de alta impedância (da ordem de  $k\Omega$ ) entre espiras da mesma fase, entre fases, ou entre fase e terra (Natarajan, 1989). A corrente de falha pode atingir duas vezes a corrente de rotor bloqueado, o que causa um alto aquecimento localizado e faz com que a falha se espalhe rapidamente. Se a falha for detectada no início é possível reutilizar o motor após reparo adequado, mas se a falha evoluir é provável que o dano causado ao núcleo estatórico, devido à ocorrência de arcos elétricos, seja irreparável (Thomson,

2001). Com o intuito de minorar os danos, diversos métodos para detecção de falhas por curto-circuito entre espiras têm sido desenvolvidos no âmbito acadêmico, como é apresentado no item 2.2.

## 2.2 Estado da Arte em Detecção de Falhas de CCEEE

Diferentes métodos de detecção de curto-circuito entre espiras têm sido usados, por muitos pesquisadores, buscando melhorar sua eficiência e torná-los menos invasivos. O método de análise de assinatura de corrente consiste em se detectar falhas usando determinadas componentes do espectro de frequência da corrente, como característica de falha. Considerando este método, Joksimovic e Penman (2000) mostram que não existem novas componentes no espectro de frequências devido a falhas de isolamento, e que apenas ocorre um aumento nas componentes já existentes. Stavrou, Sedding e Penman (1999) procuraram, no espectro de frequências de corrente, a variação nas frequências em função do número de polos, ranhuras e escorregamento, ou seja, em função das características construtivas e operacionais específicas.

Penman, Sedding e Lloyd (1994) desenvolveram a seguinte equação,

$$f_{cc} = \{k \pm n(1 - s)/p\}f_1, \quad (2.1)$$

para obter componentes harmônicas ( $f_{cc}$ ) no fluxo de dispersão axial de motores relacionadas a falhas de curto-circuito entre espiras do estator. Nesta equação,

$f_{cc}$ : são as componentes em função do curto-circuito entre espiras;

$k = 1, 3, 5...$ , é a ordem das harmônicas temporais;

$n = 1, 2, 3...$ , é a ordem das harmônicas espaciais;

$s$  é o escorregamento;

$p$  é o número de pares de polos;

$f_1$  é a frequência fundamental da tensão de alimentação.

É importante salientar que algumas das frequências, obtidas por decorrência da aplicação da equação (2.1), também podem estar relacionadas a outros tipos de falhas no motor ou na própria fonte de alimentação. Por exemplo, de acordo com Das *et al.* (2012), a tensão de alimentação desbalanceada pode produzir uma assinatura de corrente que é, aparentemente, idêntica à assinatura causada pelo

CCEEE. Os autores propõem um método para separar os dois padrões. Este método é baseado na Aproximação por Vetor de Park Estendido (*Extended Park's Vector Approach - EPVA*) combinado com ferramentas de processamento de sinais, como Transformada Rápida de Fourier (*Fast Fourier Transform - FFT*), Transformação Wavelet Discreta (*Discrete Wavelet Transform – DWT*) e Densidade Espectral de Potência (*Power Spectral Density – PSD*), para fazer a diferenciação.

Para tentar encontrar assinaturas no espectro de frequência exclusivas do CCEEE, Thomson e Fenger (2001) tomaram como base as frequências das correntes estatísticas sugeridas pela equação (2.1), e realizaram análises experimentais em motores de baixa potência, onde aplicaram curto-circuito entre espiras. As componentes encontradas como função unicamente do curto-circuito ocorrem, simultaneamente, quando  $k = 1$  e  $n = 3$ , e quando  $k = 1$  e  $n = 5$ . Para um motor a vazio ( $s \cong 0$ ), com dois pares de polos, estas componentes são, aproximadamente,  $2,5f_1$  e  $3,5f_1$ .

Ainda utilizando análise da assinatura de corrente do motor, Gazzana, Pereira e Fernandes (2010) implementaram um sistema para detectar e diagnosticar, prematuramente, quebras de barras no rotor, excentricidade no eixo e falhas por CCEEE, em motores de indução. Para estas últimas, a equação (2.1) é utilizada com  $k = 1$  e  $n = 7$ , e o método de Welch é utilizado como estimador da densidade espectral de potência. A escolha de uma componente espacial de alta ordem no espectro se dá pelo fato de que componentes de baixa ordem são afetadas por excentricidade no eixo, assim como por curto-circuito.

Outra forma de detecção de falhas de CCEEE é apresentada por Ballal *et al.* (2006), que empregam a teoria dos componentes simétricos para detecção de falhas de curto-circuito entre espiras. A técnica consiste em usar uma expressão para separar as correntes de sequência positiva, de sequência negativa, e de sequência zero. Um gráfico em que as componentes de sequência positiva e negativa descrevem trajetórias circulares de direção oposta de giro é analisado. Em seguida a detecção é feita através de uma medida de deformação causada nos gráficos, quando ocorre a falha. Já Xu, Li e Sun (2004) definem como característica para detecção de CCEEE a impedância aparente de sequência negativa. Eles elaboraram um método efetivo, baseado em simulações, e realizaram testes experimentais, através dos quais perceberam que ocorre uma oscilação no valor de impedância em função do tempo,

que pode, entretanto, ser corrigida usando-se um filtro passa-baixa, o que garante a confiabilidade da técnica.

Tratando-se de métodos computacionais, Hyun *et al.* (2010) criaram modelos neurais usando as correntes estatóricas como variáveis para simular o estado de um motor de indução sem falhas, de um com falha no isolamento e de outro com falha no rolamento. Os motores, bem como seus modelos, são excitados com as mesmas entradas, e a saída real é, constantemente, comparada com as saídas dos modelos neurais. Assim, uma rede Bayesiana avalia os resíduos dos modelos e detecta falha tanto no isolamento, quanto no rolamento.

Bouzid *et al.* (2008) utilizaram uma rede neural para localizar a fase onde ocorre o curto. Foram escolhidas, como características de falha, os ângulos de fase entre as correntes e tensões das três fases. A detecção foi feita através de uma rede *Multi-Layer Perceptron (MLP)* com 3 neurônios de saída, cada um representando uma fase. Se um neurônio está ativo, então a fase representada por este está em curto. Os autores validaram o método usando dois motores de indução, e constataram que, uma vez que uma Rede Neural Artificial (RNA) for treinada para um motor, ela pode ser usada para outras máquinas de mesmas características.

Das *et al.* (2010) processaram os sinais de corrente de linha através de uma transformação de Park, seguida por uma transformação Wavelet contínua, e utilizaram uma Máquina de Vetores de Suporte (*Support Vector Machine - SVM*) para classificar motores com falha a partir das características extraídas. Dos 18 casos de teste utilizados para predição, um total de 16 casos de falha foram corretamente identificados por uma SVM apropriadamente configurada.

Dentre todos os possíveis métodos de detecção de falhas, a assinatura de corrente tem um grande potencial, visto que este não é invasivo, não requer a instalação de sensores na máquina, não precisa ser adaptado para áreas com risco de explosão, apresenta alta capacidade para monitoramento remoto, pode ser aplicado a qualquer máquina sem restrição de potência, apresenta sensibilidade para falhas mecânicas da máquina, falhas elétricas no estator e problemas de alimentação, dentre outras vantagens (Thorsen; Dalva, 1997). Além destas verificadas em motores alimentados por conversores de frequência, há a possibilidade de se embarcar o sistema de detecção no próprio conversor, especialmente se técnicas de inteligência computacional são utilizadas.

Nesta dissertação é apresentada uma proposta de utilização de redes neurais artificiais, do tipo Perceptron (simples e multicamadas) (Haykin,1999), para classificar curto-circuito entre espiras do estator. A Transformada de Fourier é usada para se obter o espectro de frequência dos dados coletados, e a equação (2.1), desenvolvida por Penman e explorada por Thompson, é utilizada como referência para a escolha das componentes de frequência que caracterizam as falhas de CCEEE. Em seguida, através da análise de variâncias e de experimentações, selecionam-se novas características como atributos de entrada das redes.

Posteriormente, no capítulo 3, os conceitos básicos necessários à compressão das redes neurais artificiais, usadas nesta dissertação, são revisados de forma sucinta.



### 3 REDES NEURAIAS ARTIFICIAIS E CLASSIFICADORES

As redes neurais artificiais (RNA) podem ser projetadas para uso nas mais diversas aplicações, como classificação, reconhecimento de padrões, otimização, controle e aproximação de funções (Engelbrecht, 2007). Neste trabalho, são projetados classificadores através de RNA, também chamados de classificadores neurais, já que possuem, segundo Haykin (1999, p. 24-26), uma vasta lista de atrativos para seu uso, tais como: a) terem capacidade de tratar problemas não lineares, mesmo complexos modelos multivariáveis; b) serem um modelo não paramétrico, aprendendo com base nos dados do problema, sem que seja necessário profundo conhecimento teórico do processo; c) serem potencialmente adaptativos, podendo ser treinados constantemente para se adaptar a novas condições; d) em classificação de padrões, terem capacidade de gerar resposta comprobatória, que permita avaliar a confiabilidade da classificação e rejeitar padrões ambíguos; e) serem tolerantes a falhas nas implementações em *hardware*, já que possuem informação distribuída entre suas conexões, ou seja, falha em uma conexão poderá reduzir o desempenho da rede, mas não irá causar discrepâncias extremas nos resultados; f) possuírem uniformidade nos projetos e análises, como nas notações e na unidade básica (neurônio), o que permite compartilhar teorias e algoritmos de aprendizado para diferentes aplicações.

Apesar da extensa lista de atrativos das RNA, seu uso deve se restringir a problemas para os quais ferramentas de menor complexidade e custo computacional não sejam suficientes para atingir resultados apreciáveis. Mesmo entre redes capazes de mapear classes não linearmente separáveis, como a MLP, o custo computacional envolvido pode variar bastante em função da topologia selecionada. Por isso busca-se projetar uma rede que equilibre o custo computacional com os requisitos estabelecidos. Diferenças em relação ao tempo de processamento podem parecer irrelevantes em protótipos desenvolvidos em programas de alto nível, como o MATLAB® e OCTAVE®, que são executados em computadores com alta capacidade de processamento. No entanto, se há intenção de embarcar a rede neural em um sistema eletrônico, no qual há maior limitação de memória e capacidade de processamento, é necessário reduzir o esforço computacional envolvido. Como a rede Perceptron Simples (PS) é um classificador para problemas linearmente separáveis,

ele ou outro classificador linear não neural, pode ser usado, inicialmente, para atestar a separabilidade do conjunto de dados disponível. Se o desempenho de um classificador linear for considerado suficiente para a aplicação, então o uso de modelos neurais complexos é impertinente, pois apenas dificultaria o projeto e poderia aumentar o custo computacional envolvido.

Além de serem usados para testes iniciais nos conjuntos de dados e na solução de problemas com conjuntos de dados linearmente separáveis, a forma de treinamento dos classificadores lineares ajuda na compreensão dos algoritmos das redes multicamadas utilizadas nesta dissertação. No item 3.1 é feita uma descrição da rede Perceptron, que constitui a base para arranjos neurais de múltiplas camadas. No item 3.2 é abordada, brevemente, a técnica dos Mínimos Quadrados, que faz o ajuste dos pesos sinápticos no algoritmo Máquina de Aprendizado Extremo (*Extreme Learning Machine – ELM*). O método dos Mínimos Quadrados Recursivos, apresentado no item 3.3, constitui uma alternativa no treinamento para o algoritmo *ELM* em que não é necessária a inversão de matrizes. No item 3.4, a rede MLP é abordada, assim como os algoritmos de treinamento de retropropagação do erro e *ELM*.

### 3.1 Perceptron

Barreto (2007) apresenta a teoria sobre o Perceptron. Primeiramente, assume-se que há uma função matemática  $\varphi(\cdot)$  que relaciona um vetor de entrada qualquer,  $x \in \mathbb{R}^{p+1}$ , com um vetor de saída,  $d \in \mathbb{R}^q$ . Ou,

$$\mathbf{d} = \varphi(\mathbf{x}). \quad (3.1)$$

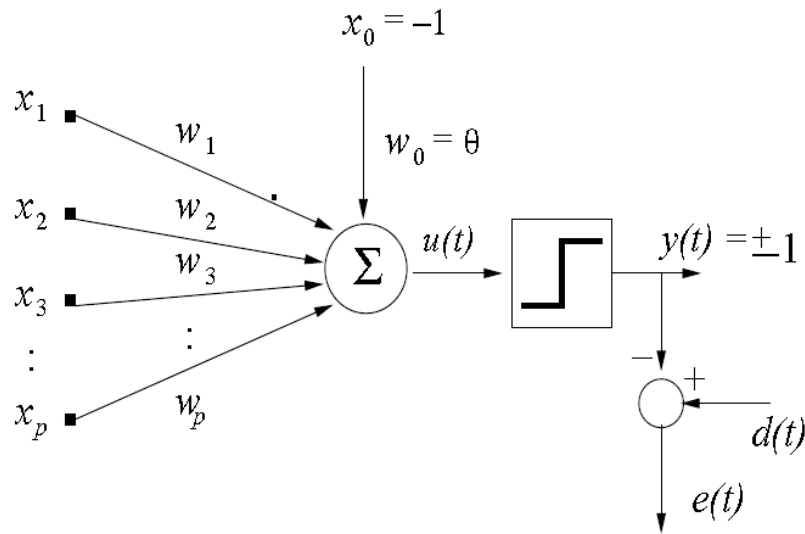
Assume-se que  $\varphi(\cdot)$  é totalmente desconhecida, mas que pode representar um problema de classificação de padrões. O Perceptron deve aprender esta relação com base na apresentação de várias amostras do espaço de entrada em conjunto com a saída esperada para a respectiva amostra, ou seja, pares de entrada-saída. Este tipo de aprendizado, através da apresentação de pares de entrada-saída, é chamado de supervisionado.

Supondo um conjunto finito de  $n$  pares de entrada-saída, a rede neural implementará um mapeamento aproximado,  $\hat{\varphi}(\cdot)$ , tal que

$$\mathbf{y} = \hat{\varphi}(\mathbf{x}) \quad (3.2)$$

em que  $y$ , espera-se, seja muito próximo do valor desejado  $d$ .

Figura 2. Arquitetura do neurônio artificial da rede Perceptron.



Fonte: (Barreto, 2007)

As definições de  $n$ ,  $x$ ,  $\hat{\phi}(\cdot)$ ,  $y$  e  $d$  são compartilhadas por todas as RNA com aprendizado supervisionado.

Na Figura 2, é exposta a arquitetura da primeira rede neural artificial reconhecida, composta de um único neurônio artificial, a rede Perceptron Simples. O vetor de entradas do Perceptron pode ser definido como

$$\mathbf{x}(t) = \begin{pmatrix} x_0(t) \\ x_1(t) \\ \vdots \\ x_j(t) \\ \vdots \\ x_p(t) \end{pmatrix} = \begin{pmatrix} -1 \\ x_1(t) \\ \vdots \\ x_j(t) \\ \vdots \\ x_p(t) \end{pmatrix} \quad (3.3)$$

em que  $x_j(t)$  denota a  $j$ -ésima componente do vetor de entrada  $\mathbf{x}(t)$ , e  $t$  indica o instante de apresentação deste vetor à rede.

Para um classificador com um único neurônio, a saída desejada está em um espaço binário  $\mathbb{R}^2$ . No entanto, a saída desejada para uma rede Perceptron com mais neurônios pode ser representada, genericamente, por um vetor de  $q$  componentes, ou seja,

$$\mathbf{d}(t) = \begin{pmatrix} d_1(t) \\ \vdots \\ d_i(t) \\ \vdots \\ d_q(t) \end{pmatrix} \quad (3.4)$$

em que  $d_i(t)$  denota a saída desejada para o  $i$ -ésimo neurônio. Em geral a rede possui  $q$  neurônios de saída, sendo a ativação de cada neurônio associada a uma classe.

O vetor de pesos associado ao  $i$ -ésimo neurônio é representado como

$$\mathbf{w}_i(t) = \begin{pmatrix} w_{i0}(t) \\ w_{i1}(t) \\ \vdots \\ w_{ij}(t) \\ \vdots \\ w_{ip}(t) \end{pmatrix} = \begin{pmatrix} \theta_i(t) \\ w_{i1}(t) \\ \vdots \\ w_{ij}(t) \\ \vdots \\ w_{ip}(t) \end{pmatrix} \quad (3.5)$$

em que  $w_{ij}$  é o peso sináptico que conecta a entrada  $j$  ao  $i$ -ésimo neurônio e  $\theta_i$  é o limiar (*bias*) associado ao  $i$ -ésimo neurônio.

É importante observar que cada neurônio de uma rede Perceptron possui seu próprio vetor de pesos  $w_i$ , de tal forma que uma rede com  $q$  neurônios terá  $p \times q$  pesos sinápticos  $w_{ij}$  e  $q$  limiares  $\theta_i$ , resultando em um total de  $(p + 1) \times q$  parâmetros ajustáveis. O ajuste destes parâmetros é feito através da Regra de Aprendizagem do Perceptron.

O funcionamento e o aprendizado do Perceptron dão-se a partir da apresentação de um vetor de entrada  $x$ , na iteração  $t$ , e do cálculo da ativação  $u_i(t)$  do  $i$ -ésimo neurônio de saída. A ativação é calculada da seguinte forma

$$u_i(t) = \sum_{j=1}^p w_{ij}(t)x_j(t) - \theta_i(t) \quad (3.6)$$

$$u_i(t) = \sum_{j=1}^p w_{ij}(t)x_j(t) + w_{i0}(t)x_0(t) \quad (3.7)$$

$$u_i(t) = \sum_{j=0}^p w_{ij}(t)x_j(t) \quad (3.8)$$

$$u_i(t) = \mathbf{w}_i^T(t) \cdot \mathbf{x}(t) \quad (3.9)$$

em que  $x_0(t) = -1$  e  $w_{i0}(t) = \theta_i$ . Pode-se notar que a ativação do neurônio no instante  $t$  é simplesmente o produto escalar do vetor de entrada  $\mathbf{x}(t)$  com o vetor de pesos  $\mathbf{w}_i(t)$  do  $i$ -ésimo neurônio, que pode ser entendido como uma medida da proximidade entre vetores.

A saída do Perceptron é função da medida de proximidade entre o vetor de entrada  $\mathbf{x}$  e o vetor de pesos  $\mathbf{w}_i$  no instante  $t$ , e pode ser dada pela aplicação da função sinal

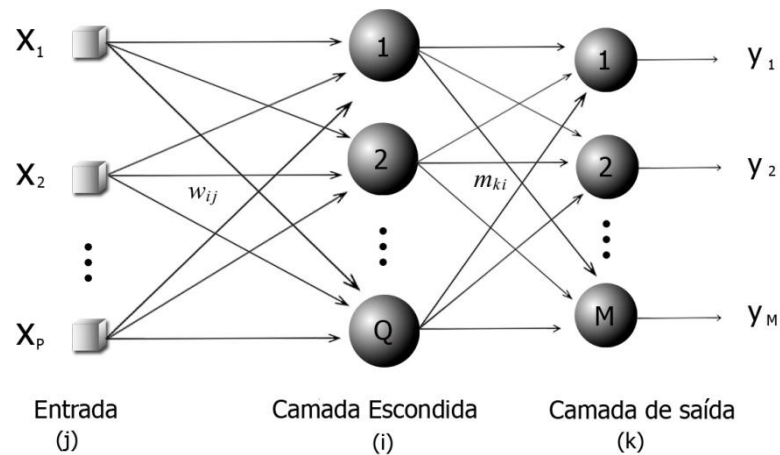
$$y_i(t) = \text{sinal}(u_i(t)) = \begin{cases} +1, & \text{se } u_i(t) \geq 0 \\ -1, & \text{se } u_i(t) < 0 \end{cases} \quad (3.10)$$

A saída estimada  $y_i(t)$  é comparada à saída esperada  $d_i(t)$ . Caso sejam diferentes, ou seja, caso haja classificação errada, então a Regra de Aprendizado do Perceptron deve corrigir os pesos da rede. No item 3.3.1 são mostrados os detalhes deste processo.

### 3.2. Perceptron Multicamadas

Foi provado que uma RNA MLP, com uma camada intermediária, pode aproximar qualquer função contínua, desde que tenha neurônios suficientes (Hornik *et al.*, 1989). Em problemas de classificação elas são recomendadas quando há uma relação não linear desconhecida entre um conjunto de dados de entrada e um conjunto de saída, mesmo em complexos problemas multivariáveis. Elas são capazes de aprender tal relação através da apresentação dos dados e então generalizar o conhecimento e classificar novos dados.

**Figura 3. Modelo genérico de rede neural alimentada diretamente com única camada escondida.**



**Fonte: Autor.**

Na Figura 3, é mostrada uma arquitetura genérica de uma Rede Neural Alimentada Diretamente de Única Camada Escondida, ou *Single Hidden-Layer Feedforward Neural Network (SLFN)*. No início estão os dados de entrada da rede ( $x$ ), que são completamente conectados à camada escondida pelos pesos  $w_{ij}$  de forma semelhante às conexões sinápticas dos neurônios biológicos. A função da camada escondida é realizar uma transformação no espaço original dos dados para um novo

espaço onde as classes sejam linearmente separáveis. A camada escondida é, então, completamente conectada à camada de saída, através dos pesos  $m_{ki}$ . Na camada de saída é aplicada outra função de ativação e, a partir dos valores do vetor de saída obtido, a classificação é feita.

A MLP treinada pelo algoritmo *Back-propagation* (MLP/BP) é, provavelmente, o mais estudado e clássico modelo neural, especialmente em aplicações de classificação, mas mesmo com o avanço no estudo e técnicas relacionadas a esta ferramenta, um usuário logo percebe as dificuldades em encontrar uma arquitetura satisfatória para aplicações do mundo real. Há uma grande quantidade de variáveis que precisam ser ajustadas (taxa de aprendizado, termo de momento, critério de parada, divisão do conjunto de dados, normalização dos dados de entrada, dentre outros) por métodos heurísticos ou, mais comumente, por tentativa e erro. Ademais, o tempo necessário para o treinamento da rede através do algoritmo *Back-propagation*, normalmente, é excessivo. Isto levou muitos pesquisadores a buscarem novos algoritmos de treinamento.

Um novo algoritmo para treinamento de *SLFN* chamado *Extreme Learning Machine*, ou Máquina de Aprendizado Extremo foi apresentado, à comunidade científica, por Huang *et al.* (2004), e se tornou alvo de muitos estudos (Gaitang e Ping, 2010, Nan e Wang, 2010, Miche *et al.*, 2010, Soria-Olivas *et al.*, 2011, Yang; Yaonan e Xiaofang, 2012, Huang *et al.*, 2012). A grande vantagem do algoritmo ELM é a velocidade extrema no treinamento da rede. Além desta, pode-se acrescentar a maior facilidade ao se projetar uma rede, praticamente sem a necessidade de ajuste de variáveis, como taxa de aprendizado, critério de parada e número de épocas.

### **3.3 Algoritmos de Aprendizado**

A seguir são descritos os principais algoritmos que são usados para treinamento das redes neurais utilizadas nesta dissertação.

#### **3.3.1 Aprendizado do Perceptron**

A regra de aprendizado do Perceptron é baseada na minimização do erro de classificação dos vetores de entrada (Webb, 2002). Para isso, pode-se usar a função-objetivo,

$$J[w] = \sum_{z_k \in Z} (-w^T z_k) \quad (3.11)$$

em que  $z_k$  denota o  $k$ -ésimo vetor de entrada classificado incorretamente, e  $Z$  é o conjunto do vetores classificados incorretamente.

A partir da equação 3.11, utiliza-se um método iterativo, baseado na derivada primeira da função, para se encontrar uma solução ótima. Tal regra, de ajuste recursivo, é chamada de método do gradiente descendente. Assim

$$w^{novo} = w^{atual} + \Delta w \quad (3.12)$$

$$w^{novo} = w^{atual} - \eta \frac{\partial J[w]}{\partial w} \quad (3.13)$$

em que  $w^{atual}$  corresponde ao valor atual de  $w$ , enquanto  $w^{novo}$  denota o valor após o ajuste. A constante  $0 < \eta \ll 1$  é chamada de taxa de aprendizagem.

A derivada é dada por

$$\frac{\partial J[w]}{\partial w} = \sum_{z_k \in Z} (-z_k), \quad (3.14)$$

que nada mais é do que a soma dos vetores mal classificados. Substituindo-se (3.13) em (3.14), chega-se a

$$w^{novo} = w^{atual} + \eta \sum_{z_k \in Z} (z_k) \quad (3.15)$$

A regra de aprendizagem mostrada atualiza os pesos usando todos os vetores mal classificados de uma única vez, por isso é chamado de treinamento em lote. Muitas vezes, é mais conveniente atualizarem-se os pesos logo que ocorre um erro de classificação, o que caracteriza um treinamento iterativo. Neste caso, a regra pode ser escrita como

$$w(t+1) = w(t) + \eta z(t) \quad (3.16)$$

em que  $t$  denota a iteração de apresentação do vetor de entrada  $z(t)$ .

É possível definir a regra (3.16) em função dos vetores de entrada  $x(t)$  substituindo-se  $z(t)$  pelo produto dos sinais de erro  $e(t)$  pelos vetores de entrada  $x(t)$ . A regra de aprendizado é, então, reescrita como

$$w(t+1) = w(t) + \eta e(t)x(t) \quad (3.17)$$

em que  $e(t) = d(t) - y(t)$  corresponde ao erro de classificação do vetor de entrada  $x(t)$ . Caso haja classificação correta, o erro será nulo e não haverá ajuste no vetor de pesos, ou seja, o ajuste só ocorre para vetores mal classificados.

Expandindo a regra para o caso em que há  $q$  neurônios, o ajuste do vetor de pesos do  $i$ -ésimo neurônio é dado por

$$w_i(t + 1) = w_i(t) + \eta e_i(t)x(t) \quad (3.18)$$

em que  $e_i(t) = d_i(t) - y_i(t)$  corresponde ao erro de classificação do  $i$ -ésimo neurônio.

Considerando cada elemento do vetor de entradas e do vetor de pesos separadamente, a regra pode ser escrita como

$$w_{ij}(t + 1) = w_{ij}(t) + \eta e_i(t)x_j(t), i = 1, \dots, q \quad j = 0, 1, \dots, p \quad (3.19)$$

Então, após a apresentação das  $n$  amostras disponíveis, os pesos da rede são ajustados. No entanto, se o número de amostras não for suficiente para classificação satisfatória, os dados são novamente apresentados à rede. Cada apresentação das  $n$  amostras de treinamento é chamada de *época* e é feita de forma aleatória para que a busca da solução ótima, no espaço de pesos, seja, aproximadamente, estocástica (Medeiros e Barreto, 2013).

Uma maneira simples de avaliar a convergência do algoritmo é através do erro quadrático médio ( $\varepsilon_{train}$ )

$$\varepsilon_{train} = \frac{1}{2N} \sum_{t=1}^N \sum_{i=1}^q [d_i(t) - y_i(t)]^2, \quad (3.20)$$

calculado após cada época. Se o valor obtido está abaixo de um patamar especificado, então a convergência é atingida. A avaliação da capacidade de generalização é obtida por um conjunto de validação, que contém os exemplos não apresentados antes à rede.

### 3.3.2 Mínimos Quadrados

O método conhecido como Mínimos Quadrados (MQ) é uma técnica de otimização que procura encontrar a curva com o melhor ajuste para um conjunto de dados, tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados (Datta, 1995). Portanto, é possível criar um classificador linear a partir da técnica dos mínimos quadrados, uma vez que ela pode ser usada para criar uma fronteira de separação, ou decisão, entre dados associados a diferentes classes, de forma semelhante à rede Perceptron.



A seguir, é mostrado como o método é derivado de maneira menos formal, do ponto de vista matemático. O objetivo é facilitar o entendimento da rede ELM, além do seu uso como classificador.

Dado o sistema linear

$$d = Wx \quad (3.21)$$

em que  $x$  denota um vetor de entrada qualquer,  $x \in \mathbb{R}^{p+1}$ , e  $d$  um vetor de saída esperada ou desejada,  $d \in \mathbb{R}^q$ . O vetor de entrada  $x$  e o vetor de saída  $d$  estão relacionados por uma função linear, em que a matriz  $W$ , cuja dimensão  $\dim[W] = q \times (p + 1)$ , é a matriz de parâmetros ajustáveis que se deseja estimar a partir de valores conhecidos dos pares de entrada-saída  $x$  e  $d$ .

Tais definições são semelhantes às definições feitas na descrição da rede Perceptron e, de forma semelhante, é obtido um mapeamento aproximado, tal que

$$y = \widehat{W}x \quad (3.22)$$

em que  $\widehat{W}$  é uma matriz de transformação que produz uma saída  $y$  próxima da saída desejada  $d$ . O vetor de erros para o par entrada-saída é definido como

$$e = d - y \quad (3.23)$$

Para se estimar os elementos da matriz  $\widehat{W}$ , define-se uma matriz  $X$  composta por todos os vetores de amostra disponíveis, e  $D$  uma matriz composta por todos os vetores de saída esperada para os respectivos valores de  $X$

$$X = \begin{pmatrix} x_{01} & x_{02} & & x_{0n} \\ x_{11} & x_{12} & & x_{1n} \\ \vdots & \vdots & & \vdots \\ x_{j1} & x_{j2} & \cdots & x_{jn} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & & x_{pn} \end{pmatrix} \quad (3.24)$$

$$D = \begin{pmatrix} d_{11} & d_{12} & & d_{1n} \\ \vdots & \vdots & & \vdots \\ d_{i1} & d_{i2} & \cdots & d_{in} \\ \vdots & \vdots & & \vdots \\ d_{q1} & d_{q2} & & d_{qn} \end{pmatrix} \quad (3.25)$$

A dimensão de  $X$  é  $\dim[X] = (p + 1) \times n$  e a dimensão de  $\dim[D] = q \times n$ . Desta forma, através dos valores conhecidos dos pares de entrada-saída  $X$  e  $D$ , constrói-se o mapeamento linear, como mostrado em (3.22), na forma matricial

$$D = \widehat{W}X \quad (3.26)$$

Tem-se o interesse de calcular  $\widehat{W}$ . Se a matriz  $X$  fosse uma matriz quadrada e inversível, bastaria multiplicar  $X^{-1}$  à direita, em ambos os lados da equação. Se a matriz  $X$  não é quadrada, o artifício usado para se obter  $\widehat{W}$  é, primeiro, multiplicar ambos os lados pela transposta de  $X$ ,  $X^T$

$$DX^T = \widehat{W}XX^T. \quad (3.27)$$

Após a multiplicação por  $X^T$ , o termo  $XX^T$  possui dimensão  $\dim[XX^T] = (p + 1) \times (p + 1)$ . Ou seja, forma uma matriz quadrada, sendo, então, possível se obter  $\widehat{W}$  fazendo

$$DX^T(XX^T)^{-1} = \widehat{W}XX^T(XX^T)^{-1}, \quad (3.28)$$

$$\widehat{W} = DX^T(XX^T)^{-1}. \quad (3.29)$$

Desta forma, obtém-se a matriz estimada de parâmetros. Pode-se perceber que a qualidade da estimativa depende diretamente da inversibilidade de  $XX^T$ , que se torna mais difícil com o crescimento de  $n$ , ou seja, se as dimensões de  $XX^T$  forem grandes. É importante salientar que a implementação computacional da equação (3.29) deve ser feita usando-se decomposição em valores singulares ou outro método numérico. A implementação direta conduz a pesos com altos valores. Isto pode levar a instabilidade numérica.

Após o cálculo de  $\widehat{W}$ , o desempenho do classificador com um novo vetor de amostra  $x_{ts}$  pode ser testado fazendo-se

$$y_{ts} = \widehat{W}x_{ts}, \quad (3.30)$$

e verificando-se, em seguida, a proximidade do valor obtido com o valor esperado, para tal amostra. Igualmente, um conjunto  $X_{ts}$  composto por várias novas amostras pode ser testado de uma vez, usando-se notação matricial

$$Y_{ts} = \widehat{W}X_{ts} \quad (3.31)$$

Diferentemente do Perceptron, este método não é iterativo, ou seja, todos os parâmetros ajustáveis são redefinidos de uma vez. Além disso, tal ajuste é dado pela inversão de uma matriz e, portanto, a qualidade da aproximação está diretamente ligada ao processo usado para inversão da matriz.

Uma maneira de se realizar o método dos mínimos quadrados sem necessidade de uma inversão de matrizes, e, conseqüentemente, se evitar inversão

de possíveis matrizes mal condicionadas, é através dos Mínimos Quadrados Recursivos (MQR).

### 3.3.3 Mínimos Quadrados Recursivos

O método dos mínimos quadrados recursivos tem a vantagem de dispensar a inversão de matriz para estimação dos parâmetros do modelo.

Aguirre (2007) prova, a partir da equação (3.29) que, para os MQR, a expressão de atualização iterativa dos parâmetros do modelo é dada por

$$\widehat{W}_n = \left[ \sum_{i=1}^n y(i)\psi^T(i-1) \right] \left[ \sum_{i=1}^n \psi^T(i-1)\psi(i-1) \right]^{-1} \quad (3.32)$$

em que se mantêm as definições anteriores, e

$$\psi = [\psi(0)|\psi(1)| \dots |\psi(n)]^T = \begin{pmatrix} x_{01} & x_{02} & & x_{0n} \\ x_{11} & x_{12} & & x_{1n} \\ \vdots & \vdots & & \vdots \\ x_{j1} & x_{j2} & \dots & x_{jn} \\ \vdots & \vdots & & \vdots \\ x_{p1} & x_{p2} & & x_{pn} \end{pmatrix}, \quad (3.33)$$

ou seja,  $\psi(i)$  representa o  $i$ -ésimo elemento tomado dos  $n$  vetores de amostra.

O ajuste de  $\widehat{W}$  dá-se pelo cálculo recursivo

$$K_n = \frac{P_{n-1}\psi(n-1)}{1 + \psi^T(n-1)P_{n-1}\psi(n-1)}, \quad (3.34)$$

$$\widehat{W}_n = \widehat{W}_{n-1} + K_n[y(n) - \psi^T(n-1)\widehat{W}_{n-1}]; \quad (3.35)$$

$$P_n = P_{n-1} - K_n\psi^T(n-1)P_{n-1}. \quad (3.36)$$

Como primeiro passo, deve-se inicializar a matriz  $P_0$  de  $\dim[P_0] = nxn$ , como uma matriz identidade multiplicada por uma constante de valor alto. O valor de  $P_0$ , e dos demais parâmetros, é ajustado, continuamente, à medida que as iterações evoluem.

A desvantagem deste método é a necessidade da inicialização de  $P_0$ , que afeta diretamente o desempenho do estimador.

### 3.3.4 Back-propagation

Esta subseção descreve, brevemente, o algoritmo de treinamento mais comum das redes MLP. Uma versão detalhada pode ser encontrada em Engelbrecht

(2007). O algoritmo de aprendizagem requer dois passos de computação: um passo no sentido direto, ou seja, da entrada para camada de saída, e um passo no sentido reverso, ou seja, da camada de saída para a entrada. Durante o passo direto, os pesos sinápticos permanecem inalterados, enquanto as ativações e saídas são computadas neurônio por neurônio. Na iteração  $t$ , a ativação de um neurônio da camada escondida é calculada por

$$u_i^{(h)}(t) = \sum_{j=0}^p w_{ij}(t)x_j(t), \quad i = 1, \dots, q \quad (3.37)$$

em que  $w_{ij}$  é o peso que conecta a entrada  $j$  ao neurônio escondido  $i$ ,  $q$  ( $2 \leq q < \infty$ ) é o número de neurônios escondidos, e  $p$  é a dimensão do vetor de entrada (excluindo o limiar). O sobrescrito  $(h)$  vem de *hidden*. Para simplificar a notação, define-se  $x_0(t) = -1$  e  $w_{i0} = \theta_i^{(h)}(t)$ , em que  $\theta_i^{(h)}(t)$  é o limiar da camada escondida  $i$ . A saída do neurônio  $i$  é, então, definida como

$$y_i^{(h)}(t) = \varphi_i \left[ u_i^{(h)}(t) \right] = \varphi_i \left[ \sum_{j=0}^p w_{ij}(t)x_j(t) \right] \quad (3.38)$$

em que  $\varphi_i(\cdot)$  é, normalmente, uma função sigmoideal. De forma similar, os valores de saída dos neurônios da camada de saída são dados por

$$y_k^{(o)}(t) = \varphi_k \left[ u_k^{(o)}(t) \right] = \varphi_k \left[ \sum_{i=0}^q w_{ki}(t)x_i(t) \right] \quad (3.39)$$

onde  $w_{ki}$  é o peso que conecta o neurônio escondido  $i$  ao neurônio de saída  $k$  ( $k = 1, \dots, m$ ), e  $m \geq 1$  é o número de neurônios de saída. O sobrescrito  $(o)$  vem de *out*. Ainda por simplificação, foi definido  $y_0(t) = -1$  e  $w_{k0} = \theta_k^{(o)}(t)$ , onde  $\theta_k^{(o)}(t)$  é o limiar do neurônio de saída  $k$ .

O passo reverso começa na camada de saída pela propagação dos sinais de erro em direção à camada escondida. Para isto, primeiramente é calculado o valor de erro  $e_k^{(o)}(t)$  gerado por cada neurônio de saída na iteração  $t$

$$e_k^{(o)}(t) = d_k(t) - y_k^{(o)}(t), \quad k = 1, \dots, m; \quad (3.40)$$

em que  $d_k(t)$  é o valor desejado de saída para o neurônio de saída  $k$ . O sinal de erro  $e_k(t)$  deve ser multiplicado pela derivada  $\phi'_k[u_k^{(o)}(t)] = \partial\phi_k/\partial u_k^{(o)}$  antes de ser propagado para trás. Isto gera o chamado gradiente local do neurônio de saída  $k$

$$\delta_k^{(o)}(t) = \phi'_k[u_k^{(o)}(t)] e_k^{(o)}(t) \quad (3.41)$$

De forma similar, o gradiente local  $\delta_i^{(h)}(t)$  do neurônio escondido  $i$  é, então, calculado como

$$\delta_i^{(h)}(t) = \phi'_i[u_i^{(h)}(t)] \sum_{k=1}^m m_{ki}(t) \delta_k^{(o)}(t) = \phi'_i[u_i^{(h)}(t)] e_i^{(h)}(t), \quad i = 0, \dots, q \quad (3.42)$$

em que o termo  $e_i^{(h)}(t)$  é o sinal de erro retro-propagado, para o neurônio escondido  $i$ . Finalmente, os pesos sinápticos dos neurônios são ajustados de acordo com a regra

$$m_{ki}(t+1) = m_{ki}(t) + \eta \delta_k^{(o)}(t) y_i^{(h)}(t), \quad i = 0, \dots, q, \quad (3.43)$$

em que  $0 < \eta \ll 1$  é a taxa de aprendizagem. Os pesos dos neurônios escondidos são ajustados através de uma regra de aprendizado semelhante

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i^{(h)}(t) x_j(t), \quad j = 0, \dots, p. \quad (3.44)$$

Algumas vezes, a convergência pode ser acelerada adicionando-se um termo ou fator de momento na regra de atualização dos pesos, resultando

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i^{(h)}(t) x_j(t) + \alpha (w_{ij}(t) - w_{ij}(t-1)), \quad j = 0, \dots, p. \quad (3.45)$$

e também,

$$m_{ki}(t+1) = m_{ki}(t) + \eta \delta_k^{(o)}(t) y_i^{(h)}(t) + \alpha (m_{ki}(t) - m_{ki}(t-1)), \quad i = 0, \dots, q, \quad (3.46)$$

em que,  $\alpha$  é o termo ou fator de momento que determina o efeito da modificação do peso passado na direção do espaço de pesos.

A avaliação da convergência do algoritmo pode ser feita através do erro quadrático médio ( $\varepsilon_{train}$ ), que para o *back-propagation* é escrito como

$$\varepsilon_{train} = \frac{1}{2n} \sum_{t=1}^n \sum_{k=1}^m [d_k(t) - y_k^{(o)}(t)]^2. \quad (3.47)$$

Assim como no algoritmo de treinamento do Perceptron Simples, o erro quadrático médio é calculado após cada época. Se o valor obtido está abaixo de um patamar especificado, então a convergência é atingida. A avaliação da capacidade de

generalização é obtida por um conjunto de validação, que contém os exemplos não apresentados antes a rede.

Outra forma de avaliar o desempenho do classificador durante o treinamento, porém levando em conta a capacidade de generalização, é através da parada prematura.

#### 3.3.4.1 Parada prematura

Para realizar a parada prematura é necessário separar um conjunto de testes com amostras diferentes das usadas para o treinamento da rede, então, após cada época, o conjunto de testes é apresentado completamente à rede e o erro quadrático médio calculado. O treinamento prossegue enquanto os erros quadráticos médios de treinamento e de testes estiverem decrescendo. Quando o erro no conjunto de testes começa a se elevar, significa que a rede está perdendo sua capacidade de generalização, ou seja, está sofrendo sobre-ajuste, ou overfitting, ao conjunto de treinamento, o que não é desejado. Portanto, os pesos escolhidos para a rede são aqueles que ocorrem imediatamente antes do sobre-ajuste.

#### 3.3.5 Extreme Learning Machine

O algoritmo ELM foi proposto, por Huang *et al.* (2004), como uma opção atrativa para ser usada para treinamento de *SLFN*, em substituição aos métodos clássicos baseados no gradiente descendente, seja de primeira ordem, como o *BP*, ou de segunda ordem, como o Levenberg-Marquadt (LVM) (Moré, 1978). Os autores provam que o algoritmo por eles proposto pode, comumente, treinar qualquer conjunto de dados milhares de vezes mais rápido do que o LVM, que, por sua vez, já é muito mais veloz do que o BP. A seguir é mostrada uma versão do algoritmo ELM.

Entrada e saída da camada escondida podem ser expressas em uma notação vetor-matriz respectivamente como (3.47) e (3.48),

$$\mathbf{u}(t) = W\mathbf{x}(t), \quad (3.48)$$

$$\mathbf{y}^{(h)}(t) = \boldsymbol{\varphi}_i(u_i(t)) = \boldsymbol{\varphi}_i(W\mathbf{x}(t)), \quad (3.49)$$

em que  $W$  é uma matriz de pesos arbitrários que conecta a entrada com a camada escondida.

$$W = \begin{pmatrix} w_{11}(t) & w_{12}(t) & \dots & w_{1p}(t) \\ \vdots & \vdots & \dots & \vdots \\ w_{i2}(t) & w_{i2}(t) & \dots & w_{ip}(t) \\ \vdots & \vdots & \dots & \vdots \\ w_{q1}(t) & w_{q2}(t) & \dots & w_{qp}(t) \end{pmatrix}_{q \times p} \quad (3.50)$$

A função  $\varphi_i(\cdot)$  é aplicada a cada um dos  $q$  componentes do vetor  $\mathbf{u}(t)$ .

$$\varphi_i(\mathbf{u}_i(t)) = \begin{pmatrix} \varphi(u_1(t)) \\ \vdots \\ \varphi(u_i(t)) \\ \vdots \\ \varphi(u_q(t)) \end{pmatrix}_{q \times 1} \quad (3.51)$$

O vetor  $\mathbf{y}^{(h)}(t)$  é calculado para cada amostra do conjunto de dados, e organizada em uma matriz  $Y^{(h)}$  com  $q$  (número de neurônios escondidos) linhas e  $n$  (número de vetores de amostra) colunas.

$$Y^{(h)} = \begin{pmatrix} \varphi(u_1(1)) & \varphi(u_1(2)) & \dots & \varphi(u_1(n)) \\ \vdots & \vdots & \dots & \vdots \\ \varphi(u_i(1)) & \varphi(u_i(2)) & \dots & \varphi(u_i(n)) \\ \vdots & \vdots & \dots & \vdots \\ \varphi(u_q(1)) & \varphi(u_q(2)) & \dots & \varphi(u_q(n)) \end{pmatrix}_{q \times n} \quad (3.52)$$

Esta matriz é usada para calcular os pesos que conectam a camada escondida à camada de saída.

Para cada vetor de entrada  $x(t)$ ,  $t = 1, \dots, n$ , existe um vetor de saída desejada  $d(t)$ . Os  $n$  vetores de saída desejada podem ser organizados em uma matriz com  $m$  (número de neurônios de saída) linhas e  $n$  colunas.

$$D = [\mathbf{d}(1) \mid \mathbf{d}(2) \mid \dots \mid \mathbf{d}(n)]_{m \times n} \quad (3.53)$$

O cálculo da matriz de pesos  $M$ , que conecta a camada escondida à camada de saída, pode ser considerado como o cálculo de um mapeamento linear entre a saída da camada escondida e a camada de saída. Ou seja, busca-se encontrar a matriz  $M$  que melhor represente a transformação dos vetores de entrada  $x(t)$  em seus correspondentes vetores de saída desejada  $d(t)$ ,

$$\mathbf{d}(t) = M\mathbf{y}^{(h)}(t). \quad (3.54)$$

Isto pode ser feito através do método dos mínimos quadrados, também conhecido como método da pseudo-inversa. A expressão é dada por

$$M = DY^{(h)T} (Y^{(h)}Y^{(h)T})^{-1} \quad (3.55)$$

Assim como foi descrito para o método dos MQ, o desempenho do classificador pode ser testado apresentando-se, à rede treinada, um novo vetor  $x_{ts}$ , ou um conjunto de vetores  $X_{ts}$ , não apresentado à rede durante o treinamento.

Como apresentado neste capítulo, os classificadores descritos são baseados em aprendizado supervisionado. Para isto, é necessário que amostras de cada classe definida sejam coletadas e rotuladas. O conjunto de um vetor de entrada, acompanhado de um rótulo, é chamado de par entrada-saída. A aquisição destes dados é feita de forma experimental, como é detalhado no capítulo 4, que inclui a descrição dos equipamentos e convenções usadas.



## 4 ENSAIOS PARA AQUISIÇÃO DE DADOS

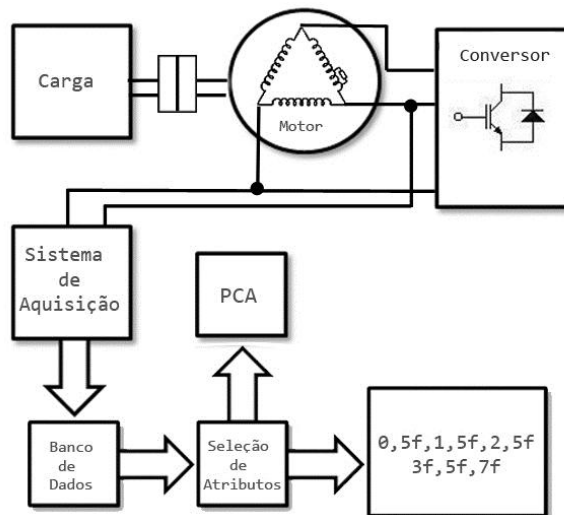
Este capítulo descreve, em detalhes, a metodologia e os equipamentos usados na montagem da bancada de ensaios, para aquisição dos dados que são usados no treinamento e validação das redes neurais.

### 4.1 Descrição Geral da Bancada

A coleta dos dados é realizada através de uma bancada composta por um conversor de frequência, um motor de indução trifásico conectado em delta e preparado para emulação de falhas de curto-circuito em diversos níveis, e um freio magnético para aplicação de carga. Um sistema de aquisição de dados coleta os sinais das correntes de linha, que são armazenados para composição de um banco de dados, do qual são selecionados, por dois diferentes métodos, os atributos que serão usados no classificador.

Na Figura 4, tem-se uma visão esquemática geral das etapas de formação do conjunto de dados. Nas seções seguintes, cada uma das etapas é descrita em detalhes.

**Figura 4. Visão geral das etapas de formação do conjunto de dados.**

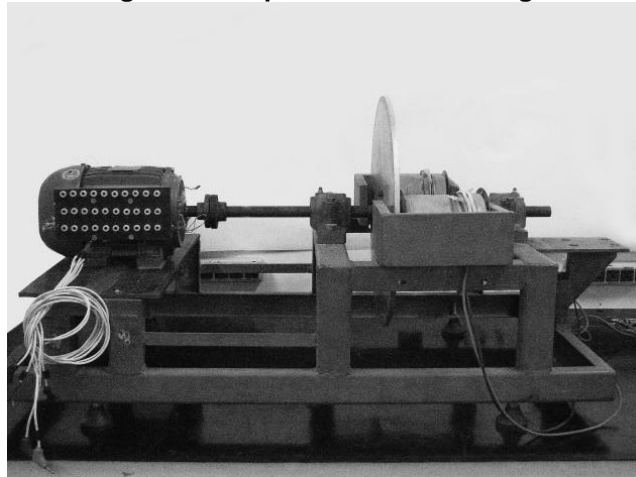


Fonte: Autor desta dissertação.

#### 4.1.2 Conjunto de acionamento

O conjunto de acionamento é composto por um conversor WEG CFW-09, o Motor de Indução Trifásico (MIT) rebobinado e um freio de Foucault, construído em laboratório. A Figura 5 ilustra o acoplamento motor-carga.

**Figura 5. Acoplamento Motor-Carga.**



**Fonte: Autor desta dissertação.**

O motor utilizado é do tipo gaiola de esquilo de 0,75 kW (1,0 CV), fabricado pela WEG, com velocidade nominal de 1720 rpm, rendimento de 79,5%, fator de potência 0.82, tensões 220/380 V e correntes 3,02/1,75 A.

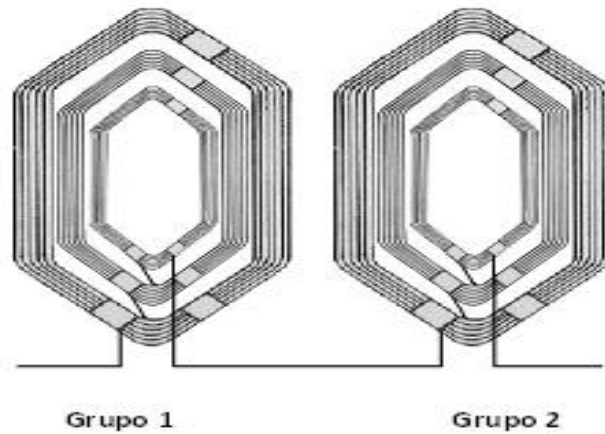
Para utilização na bancada, foi feito um rebobinamento do motor por uma empresa especializada. O enrolamento original da máquina possuía dois grupos de três bobinas concêntricas, cada uma com 58 espiras, como é representado na Figura 6. Após o rebobinamento, derivações do Grupo 1 (vide Figura 6), de cada uma das três fases do bobinamento estático, foram deixadas externas à carcaça. As derivações de duas fases foram soldadas a uma placa de bornes. Na Figura 7, os bornes são representados por B1, B2...B9, para uma fase do motor. Na Figura 8, há uma foto com o detalhe das derivações externas das duas fases soldadas aos bornes, e das derivações da terceira fase. Como o motor foi rebobinado, realizou-se um ensaio para determinar o percentual de espiras de cada um dos conjuntos (B1-B2, B3-B4, etc.) em relação ao total. Na Tabela 1, são mostrados os valores encontrados. No Apêndice A, estão os detalhes do ensaio feito para determinação desses percentuais.

**Tabela 1. Medições do valor percentual de cada conjunto de bobinas.**

| <b>Conjunto</b> | <b>Percentual Fase 1</b> | <b>Percentual Fase 2</b> |
|-----------------|--------------------------|--------------------------|
| <b>B1-B2</b>    | 0,63%                    | 0,45%                    |
| <b>B3-B4</b>    | 0,78%                    | 1,06%                    |
| <b>B5-B6</b>    | 3,40%                    | 3,93%                    |
| <b>B7-B8</b>    | 4,45%                    | 5,33%                    |
| <b>Grupo 2</b>  | 89,53%                   | 89,32%                   |

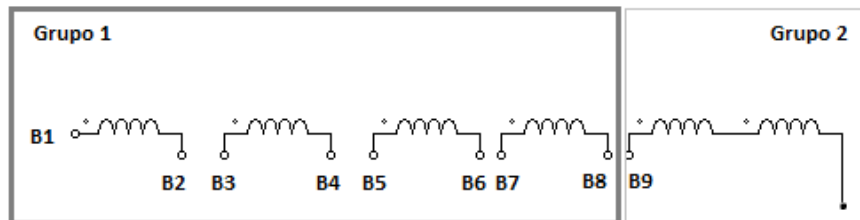
**Fonte: Autor desta dissertação.**

Figura 6. Bobinamento estátórico da fase A do MIT.



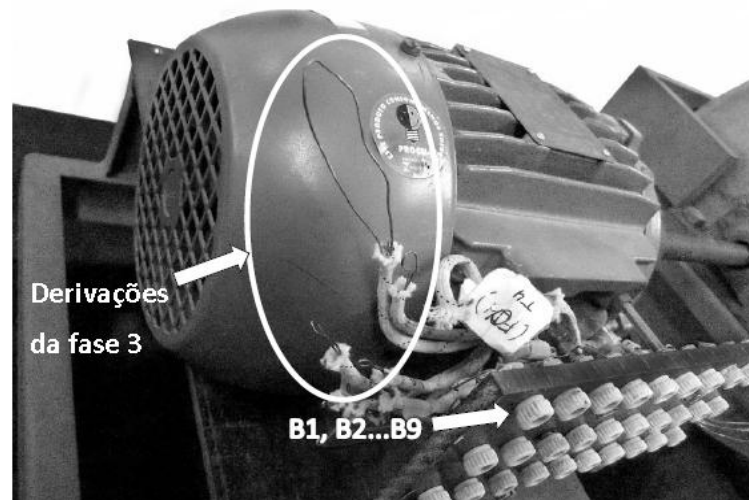
Fonte: Medeiros, 2008, modificada.

Figura 7. Esquema mostrando grupos de bobinas disponíveis após rebobinamento.



Fonte: Autor desta dissertação.

Figura 8. Detalhe das derivações externas e bornes.



Fonte: Autor desta dissertação.

#### 4.1.3 Emulação de Curto-circuito

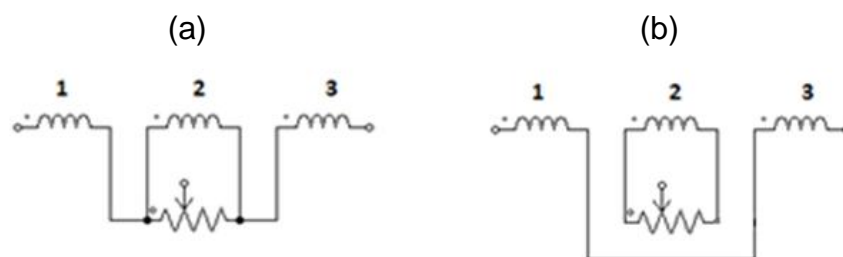
Três percentuais de curto-circuito são emulados a partir dos conjuntos de bobinas disponíveis na placa de bornes, onde estão soldadas as derivações externas:

1,41%, 4,81% e 9,26%. Respectivamente, os três percentuais são conseguidos pela aplicação de curto-circuito entre B1-B4, B1-B6 e B1-B8 (vide Figura 7 e Tabela 1).

Os percentuais de curto-circuito são emulados em uma das fases com diferentes níveis de severidade. No primeiro nível, aqui denominado curto-circuito de alta impedância (AI), a falha é emulada pela conexão de um resistor em paralelo com as espiras, de tal forma que ele desvie parte da corrente daquela seção de espiras (Figura 9.a). No segundo nível, aqui denominado curto-circuito de baixa impedância (BI), as espiras sob falha são destacadas do bobinamento e o resistor é conectado, em série, às mesmas, para a limitação da corrente de curto-circuito ao nível da corrente nominal (Figura 9.b). No restante do texto, os níveis de falha de alta impedância e baixa impedância podem ser associados aos Algarismos de 1 a 3 para representar a extensão da falha em relação aos percentuais crescentes (1,41%, 4,81% e 9,26%) do número total de espiras por bobinamento, sob condição de falha. Assim, AI1 representa um curto-circuito de alta impedância em 1,41% das espiras (condição pouco severa), e BI3 representa um curto-circuito de baixa impedância em 9,26% das espiras (o mais severo dentre todos emulados).

É importante salientar que, em ambos os casos, a corrente no bobinamento em curto-circuito é limitada à corrente nominal, portanto, não há, efetivamente, um curto-circuito de baixa impedância ocorrendo. Se isto ocorresse, levaria a valores de correntes muito elevados, que danificariam, rapidamente, as bobinas, e tornariam o ensaio destrutivo.

**Figura 9. Esquema de emulação de: (a) alta impedância e (b) baixa impedância.**



Fonte: Autor desta dissertação

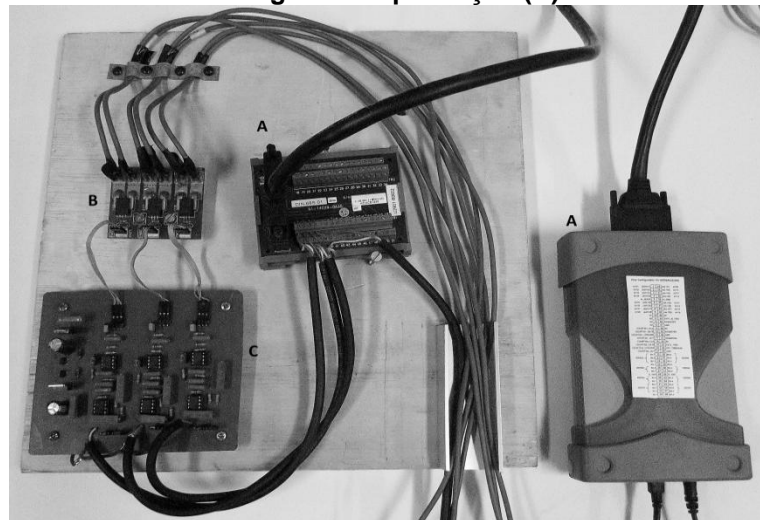
#### **4.1.4 Sistema de Aquisição de Dados**

Medições de corrente são realizadas com taxa de amostragem de 10 kHz, durante 10 segundos, nas três fases do motor, através de sensores de efeito Hall. Os sinais de corrente são filtrados, analogicamente, com filtro passa-baixa de segunda

ordem com frequência de corte de 1 kHz, e amplificados para adequar os níveis dos sinais à entrada do sistema de aquisição U2352 da Agilent, que possui 16 canais para aquisição, com resolução de 16 bits. O sistema de aquisição é conectado a um computador, no qual o *software* da Agilent pode ser usado para ajustar parâmetros como taxa e tempo de amostragem, faixa de tensão, tipo de disparo, e tipo de medição. Na Figura 10, é mostrada a foto do sistema de aquisição de dados, assim como da placa com os sensores de corrente e da placa de filtragem e amplificação.

O sistema de aquisição é configurado com taxa de amostragem de 10 kHz e tempo de amostragem de 10 segundos, o que leva a coleta de 100.000 amostras dos sinais de corrente em cada fase do motor. A faixa de tensão que representa os valores de corrente está entre -5 V e +5 V. O tipo de disparo *single shot* (disparo único) garante a captura dos dados apenas no tempo de amostragem especificado. O tipo de medição diferencial usa dois canais do sistema de aquisição para fazer a medição, dando como resultado a diferença entre os sinais medidos, e é usado para diminuir a influência de ruídos.

**Figura 10. Sistema de aquisição de dados (A), sensores de efeito Hall (B) e sistema de filtragem e amplificação (C).**



**Fonte: Autor desta dissertação**

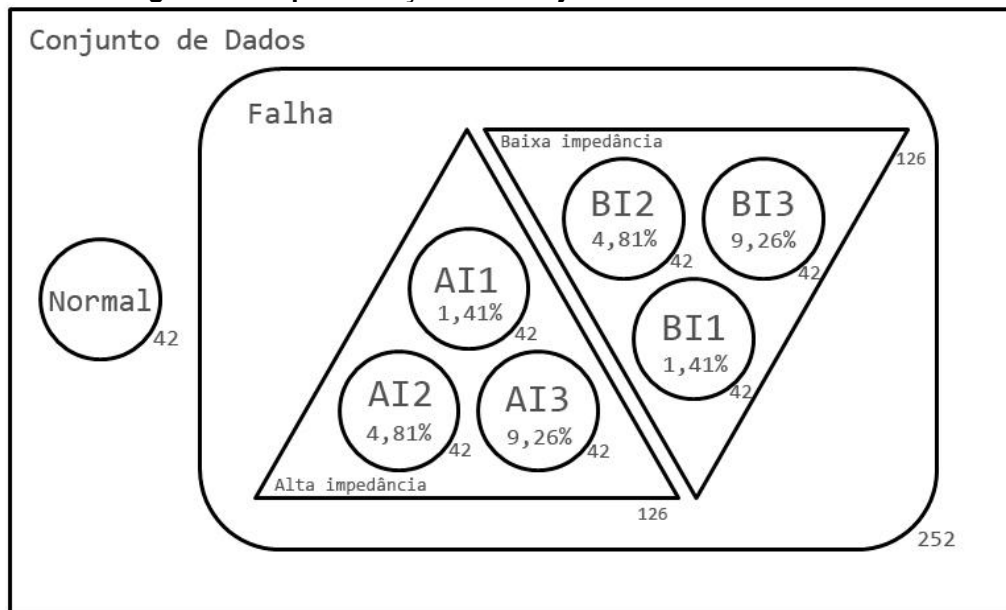
No anexo A, é mostrado o desenho esquemático da placa de filtragem.

## **4.2. Conjuntos de Dados**

O conjunto de dados base para o treinamento dos classificadores contempla, além das condições operacionais normais, diversas combinações de condições operacionais do motor sob falha. Ou seja, há dados representando

operação normal a vazio, com 50% de carga e 100% de carga para os níveis de frequência de 30 Hz, 35 Hz, 40 Hz, 45 Hz, 50 Hz, 55 Hz e 60 Hz, aplicados pelo conversor de frequência. No caso do motor sob falha de curto-circuito entre espiras, todas as condições operacionais citadas são aplicadas ao motor sob os níveis AI1, AI2, AI3, BI1, BI2 e BI3 de curto-circuito. A Figura 11 ilustra, graficamente, a composição do conjunto de dados. Nesta representação, os números associados a cada subconjunto representam a quantidade de amostras que o mesmo contém.

**Figura 11. Representação dos conjuntos de dados formados.**



**Fonte: Autor desta dissertação.**

Observando a Figura 11, pode-se perceber que o conjunto normal e cada subconjunto de falha são compostos por 42 amostras, referentes às três cargas vezes as sete frequências vezes as duas fases. O conjunto de falhas completo é composto por 252 amostras, e o conjunto de dados total por 294 amostras. Como os sinais das correntes de linha são coletados a uma taxa de amostragem de 10 kHz, durante um tempo de 10 segundos, cada vetor de corrente amostrada contém 100.000 pontos, porém para que seja feita uma análise de assinatura de corrente, é preciso que cada um destes sinais seja transformado para o domínio da frequência, em que é feita a seleção de atributos. Para se obter o espectro de frequências destes sinais, aplica-se a Transformada de Fourier, a qual retorna um espectro de frequência na faixa entre 0

Hz e 5 kHz, com a resolução de 0,1 Hz. Devido à filtragem analógica, as componentes de frequência de interesse situam-se dentro da faixa de 0 Hz a 500 Hz.<sup>1</sup>

A assinatura de corrente deve estar presente em cada um desses espectros, portanto, com base na equação de Penman (1994) e na análise estatística dos dados, são selecionadas as frequências a serem usadas como atributos de entrada das redes neurais. O capítulo 5, a seguir, contém os detalhes da metodologia usada para seleção de atributos.

---

<sup>1</sup> Mais precisamente, a faixa de 0 a 500 Hz vai de 0 a 499 Hz, enquanto que a faixa de 0 a 5 kHz vai de 0 a 4999 Hz.

## 5 SELEÇÃO DE TOPOLOGIAS E DE ATRIBUTOS

No projeto de classificadores neurais, a definição de uma topologia adequada requer uma análise minuciosa dos parâmetros envolvidos. Além disso, a escolha de atributos é essencial para um desempenho satisfatório da rede projetada.

Nesta dissertação, são definidas, primeiramente, as topologias da rede, utilizando-se atributos advindos do espectro de frequência, e escolhidos a partir da equação de Penman (1994). Após a escolha das topologias, novos testes são feitos, partindo-se dos atributos iniciais, até se chegar a um conjunto final reduzido de atributos.

### 5.1 Definição de Topologias

A definição de topologias das redes neurais é feita em duas etapas. A primeira considera apenas a capacidade da rede de mapear o conjunto de dados de entrada em relação às saídas desejadas. Desta etapa, algumas possíveis topologias são pré-selecionadas. A segunda etapa consiste em testar a capacidade de generalização das topologias pré-selecionadas para, finalmente, se escolher uma topologia definitiva.

A seguir, mostra-se o procedimento utilizado para definir as topologias empregadas nas redes MLP/*Backpropagation* e MLP/ELM. Todas as implementações foram feitas usando-se a plataforma MATLAB®.

#### 5.1.1 Topologia MLP/*Back-propagation*

Para se escolher a topologia desta rede, alguns valores são arbitrados. Primeiramente, é escolhida uma rede de uma única camada oculta, com taxa de aprendizagem única para a camada oculta e a de saída, variando-a, exponencialmente, de um valor inicial até um valor mínimo. É usado um termo de momento para atualização de todos os pesos, e os dados são normalizados na faixa entre -1 e +1. A função de ativação dos neurônios ocultos e de saída é a tangente hiperbólica. Na Tabela 2, são resumidas todas estas escolhas.

A rede é treinada usando-se todo o conjunto de dados durante 1000 épocas. Em testes preliminares, havia sido notado que eram necessárias 5000 épocas para se atingir taxas de classificação muito próximas de 100%. No entanto, usaram-



se apenas 1000 épocas nos testes para definição da topologia, visando-se reduzir o tempo gasto nos treinamentos. Isto pôde ser feito sem grandes prejuízos, pois no teste de generalização, em que é implementada a parada prematura do treinamento da rede, quase 80% dos treinamentos são interrompidos em até 1000 épocas.

**Tabela 2. Resumo de parâmetros da rede MLP usados nos testes para definição de topologia.**

| Parâmetro               | Valor   |
|-------------------------|---|
| Rede                    | 16-5-1 , <i>feedforward</i> , <i>totalmente conectada</i> .   |
| Atributos               | 0,5f; 1; 1,5f; 2f; 2,5f; 3f; 3,5f; 4f; 4,5f; 5f; 5,5f; 6f; 6,5; 7f; 7,5f; 8f.   |
| $N_{ep}$ máximo         | 1000  |
| Critério de Parada      | Parada prematura  |
| $\phi^h$ e $\phi^o$     | Tangente Hiperbólica  |
| $\eta^h$ e $\eta^o$     | Dado pela função: $\eta = \eta_f + \eta_i \left( \frac{\eta_f}{\eta_i} \right)^{Nep/\tau}$<br>Onde:<br>$\eta$ : taxa de aprendizagem<br>$\eta_i = 0,25$<br>$\eta_f = 0,005$<br>$Nep$ : Valor corrente da época<br>$\tau$ : Constante = Número de épocas/3 |
| $\alpha^h$ e $\alpha^o$ | 0,6   |
| Normalização            | Normalização entre -1 e +1  |
| Rotulação               | Com falha: -0,98<br>Sem falha: +0,98  |

**Fonte: Autor desta dissertação.**

Cada treinamento é repetido 50 vezes. Este procedimento é repetido variando-se, de 3 a 20, a quantidade de neurônios na camada oculta. A taxa de acerto média por classes é analisada para se decidir em qual topologia há melhor aprendizado do conjunto de dados. Nas tabelas seguintes,  $N_h$  refere-se ao número de neurônios na camada escondida,  $CR$  à taxa de classificação, e  $\sigma$  ao desvio padrão.

Como pode ser visto na Tabela 3, os melhores acertos estão entre 5 a 9 neurônios. Portanto, estas cinco topologias são usadas no teste de generalização para que seja feita a escolha mais adequada.

#### 5.1.1.1 Generalização da MLP/Back-propagation

Para a rede MLP/BP, são verificados os resultados de generalização para redes de 5 a 9 neurônios na camada oculta. Para isto, o conjunto de dados normal é dividido em 70% para apresentação da rede, 20% para teste de parada prematura, e 10% para validação. Do conjunto de falhas, é escolhida, aleatoriamente, igual

quantidade de amostras em relação ao conjunto normal. As demais amostras do conjunto de falhas são acrescentadas ao conjunto de validação.

**Tabela 3. Resultados da rede MLP ao usar 100% dos dados para treinamento.**

| $N_h$    | $CR_{Normal}$ | $CR_{Falha}$ |
|----------|---------------|--------------|
| 3        | 95,89         | 93,10        |
| 4        | 96,35         | 95,10        |
| <b>5</b> | <b>97,57</b>  | <b>96,28</b> |
| <b>6</b> | <b>97,60</b>  | <b>97,03</b> |
| <b>7</b> | <b>97,53</b>  | <b>95,85</b> |
| <b>8</b> | <b>97,89</b>  | <b>95,50</b> |
| <b>9</b> | <b>96,57</b>  | <b>95,67</b> |
| 10       | 96,82         | 94,50        |
| 11       | 96,21         | 94,39        |
| 12       | 95,39         | 94,32        |
| 13       | 96,53         | 94,14        |
| 14       | 94,42         | 93,78        |
| 15       | 94,28         | 93,78        |
| 16       | 94,03         | 93,96        |
| 17       | 94,00         | 93,75        |
| 18       | 94,17         | 92,46        |
| 19       | 94,42         | 92,32        |
| 20       | 90,92         | 91,42        |

Fonte: Autor desta dissertação.

Na Tabela 5, são destacados em negrito os dois maiores acertos no conjunto de validação em cada classe. O mesmo é feito no conjunto de treinamento (Tabela 4) e, em ambos, os melhores resultados de classificação, do conjunto normal, ocorrem para 6 e 9 neurônios na camada oculta, enquanto para o conjunto de falha, os dois melhores acertos ocorrem para 5 e 7 neurônios. A diferença de resultados, porém, não é grande o suficiente para que se possa garantir que uma destas topologias seja melhor do que a outra. Opta-se pela rede com 5 neurônios na camada oculta, por ela possuir menos pesos para serem ajustados e, conseqüentemente, menor custo computacional.

**Tabela 4. Resultados do conjunto de treinamento para teste de generalização da rede MLP.**

| Resultados de Treino |               |              |                   |                  |
|----------------------|---------------|--------------|-------------------|------------------|
| $N_h$                | $CR_{Normal}$ | $CR_{Falha}$ | $\sigma_{Normal}$ | $\sigma_{Falha}$ |
| 5                    | 67,48         | <b>79,23</b> | 25,14             | 18,83            |
| <b>6</b>             | <b>77,33</b>  | 71,89        | 17,16             | 18,39            |
| 7                    | 70,25         | <b>78,35</b> | 23,44             | 17,25            |
| 8                    | 71,43         | 76,25        | 25,93             | 20,84            |
| <b>9</b>             | <b>74,20</b>  | 72,61        | 20,82             | 22,05            |

Fonte: Autor desta dissertação.

Pode ser observado que as taxas de acerto obtidas estão em torno de 60% no conjunto de validação (Tabela 5). No entanto, nesta etapa, o interesse é se fazer uma análise de sensibilidade, ou seja, verificar tendências de comportamento, para que decisões, no projeto das redes, sejam tomadas. Decisões com respeito a quantidade de neurônios na camada escondida, taxa de aprendizado, termo de momento, forma de normalização dos dados de entrada, número máximo de épocas de treinamento, atributos mais relevantes, dentre outras. Espera-se que, ao final de todos os ajustes, se obtenham melhores taxas de acerto.

**Tabela 5. Resultados do conjunto de validação para teste de generalização da rede MLP.**

| <b>Resultados de Validação</b> |                            |                           |                           |                          |
|--------------------------------|----------------------------|---------------------------|---------------------------|--------------------------|
| <b>N<sub>h</sub></b>           | <b>CR<sub>Normal</sub></b> | <b>CR<sub>Falha</sub></b> | <b>σ<sub>Normal</sub></b> | <b>σ<sub>Falha</sub></b> |
| 5                              | 52,00                      | <b>65,14</b>              | 28,49                     | 16,41                    |
| 6                              | <b>67,66</b>               | 57,45                     | <b>21,66</b>              | <b>13,64</b>             |
| 7                              | 57,66                      | <b>62,45</b>              | 30,34                     | 17,09                    |
| 8                              | 58,00                      | 61,39                     | 30,16                     | 19,45                    |
| 9                              | <b>61,66</b>               | 59,05                     | 26,98                     | 18,28                    |

**Fonte: Autor desta dissertação.**

A seguir, é descrito como os parâmetros, além do número de neurônios na camada escondida, são ajustados ou escolhidos para a versão final das redes.

#### 5.1.1.2 Definição da normalização dos dados

Sabe-se que, em redes neurais, não existe um método único de ajuste de parâmetros que garanta o melhor resultado para todos os problemas. Por isso, são testadas várias formas de normalização dos dados, baseadas em determinadas conjecturas:

- a) Normalização dos dados entre -1 e +1. Esta normalização é feita pela equação

$$x_n = 2 \cdot \left( \frac{x - x_{min}}{x_{max} - x_{min}} \right) - 1, \quad (5.1)$$

em que  $x_n$  é o atributo de entrada normalizado,  $x$  o atributo antes da normalização,  $x_{min}$  é o menor valor de amostra encontrado para aquele atributo e  $x_{max}$  é o maior valor de amostra encontrado para aquele atributo. Após a normalização, os dados de entrada ficam nos limites da função de ativação tangente hiperbólica.

- b) Normalização usando-se remoção de média e divisão pelo desvio padrão antes de se ajustar os dados entre -1 e +1. A remoção da média e

divisão pelo desvio padrão visa diminuir a diferença na ordem de grandeza dos dados de entrada.

c) Normalização dos conjuntos de dados divididos pelo valor da fundamental. A componente fundamental possui ordem de grandeza superior às demais, e é sensível aos efeitos da variação na carga. Remove-se esta componente dos atributos, para utilizá-la como forma de normalização, dividindo-a pelas demais componentes. Com isso, busca-se eliminar a influência da carga nos conjuntos de dados.

d) Normalização dos conjuntos de dados divididos pelo valor da fundamental, usando-se remoção da média e divisão pelo desvio padrão. Este teste combina a hipótese de divisão pela fundamental, juntamente com a remoção da média e divisão pelo desvio padrão.

e) Normalização usando-se logaritmo antes de se ajustar entre -1 e +1. Quando há grandes discrepâncias, é comum analisar o espectro de frequências em escala logarítmica, por isto este teste é feito.

f) Semelhante à normalização do item anterior, porém nesta acrescenta-se o atributo 1f à entrada, pois em escala logarítmica, a diferença nas ordens de grandeza é reduzida.

g) Normalização dos conjuntos de dados divididos pelo valor da fundamental e, depois, com normalização usando-se logaritmo antes de se ajustar entre -1 e +1.

h) Semelhante à normalização do item anterior, porém nesta acrescenta-se o atributo 1f à entrada, Já que, em escala logarítmica, a diferença de amplitudes é reduzida, tenta-se reincluir a amplitude da frequência fundamental como atributo da rede.

**Tabela 6. Resultados do conjunto de validação por teste de normalização.**

| Teste    | Resultado de Treinamento |              |                 |                  | Resultados de Validação |              |                 |                  |
|----------|--------------------------|--------------|-----------------|------------------|-------------------------|--------------|-----------------|------------------|
|          | $CR_{Norm}$              | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Norm}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| A        | 75,43                    | 66,92        | 16,58           | 19,4             | 65,33                   | 55,47        | 24,00           | 16,51            |
| <b>B</b> | <b>72,66</b>             | <b>71,23</b> | <b>16,10</b>    | <b>14,29</b>     | <b>62,33</b>            | <b>60,90</b> | <b>24,23</b>    | <b>14,22</b>     |
| C        | 65,17                    | 74,20        | 19,82           | 17,64            | 58,66                   | 63,70        | 25,91           | 16,48            |
| D        | 63,79                    | 74,30        | 17,38           | 16,42            | 53,66                   | 64,89        | 25,03           | 15,02            |
| E        | 77,17                    | 68,87        | 21,88           | 24,32            | 69,33                   | 55,91        | 27,01           | 20,04            |
| F        | 74,82                    | 71,64        | 27,02           | 19,13            | 56,66                   | 60,10        | 31,04           | 18,33            |
| G        | 74,20                    | 66,71        | 17,30           | 16,44            | 58,33                   | 62,85        | 21,09           | 14,25            |
| H        | 74,15                    | 72,35        | 19,02           | 15,65            | 61,33                   | 59,52        | 25,74           | 14,00            |

Analisando a Tabela 6, é escolhido, como processo final de normalização, o apresentado no item b, ou seja, remoção da média e divisão pelo desvio padrão e, em seguida, ajuste dos dados entre -1 e +1. O critério foi a obtenção de um acerto equilibrado para cada classe com maior taxa de acerto e menor desvio padrão.

### 5.1.1.3 Ajuste do Termo de Momento

Para ajustar o termo de momento, usa-se a condição de normalização definida anteriormente, e varia-se o termo de momento, mantendo-se os demais parâmetros fixos. Os resultados são mostrados na Tabela 7. Os dois melhores acertos e as duas menores variâncias encontradas estão em negrito. Após se observar, cuidadosamente, a tabela, escolheu-se o valor de 0,8 para o termo de momento, usando-se o mesmo critério utilizado na escolha do processo de normalização, ou seja, maior taxa de acerto desde que haja equilíbrio entre as classes.

Os demais testes são feitos usando-se este termo de momento.

**Tabela 7. Resultados no ajuste do termo de momento.**

|                  | Resultados de Treinamento |              |                 |                  | Resultados de Validação |              |                 |                  |
|------------------|---------------------------|--------------|-----------------|------------------|-------------------------|--------------|-----------------|------------------|
|                  | $CR_{Norm}$               | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Norm}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| mom = 0,0        | 67,48                     | 68,30        | 23,33           | 21,03            | 61,33                   | 60,76        | 29,82           | 19,12            |
| mom = 0,1        | 71,89                     | 66,92        | 16,96           | 18,44            | 64,66                   | 58,73        | 28,09           | 16,05            |
| mom = 0,2        | 71,33                     | 64,05        | <b>16,72</b>    | 21,20            | 63,33                   | 54,21        | <b>24,74</b>    | 18,32            |
| mom = 0,3        | 67,58                     | 69,07        | 25,61           | 18,95            | 62,33                   | 60,48        | 31,37           | 18,62            |
| mom = 0,4        | 69,94                     | 67,38        | 20,14           | 19,9             | 63,00                   | 58,36        | 28,22           | 18,36            |
| mom = 0,5        | <b>74,97</b>              | 64,25        | <b>15,57</b>    | <b>18,54</b>     | 65,00                   | 55,52        | 26,35           | <b>13,70</b>     |
| mom = 0,6        | 69,02                     | <b>72,15</b> | 22,20           | <b>17,07</b>     | 61,66                   | <b>61,67</b> | 24,80           | <b>15,03</b>     |
| mom = 0,7        | <b>77,58</b>              | 66,00        | 18,57           | 20,16            | <b>72,00</b>            | 56,40        | <b>22,94</b>    | 16,11            |
| <b>mom = 0,8</b> | 73,28                     | <b>71,43</b> | 24,02           | 22,58            | <b>68,00</b>            | <b>62,03</b> | 27,52           | 18,79            |
| mom = 0,9        | 59,33                     | 47,53        | 47,89           | 45,86            | 59,66                   | 46,67        | 47,98           | 45,54            |

Fonte: Autor desta dissertação.

### 5.1.2.3 Ajuste da Taxa de Aprendizagem

O próximo ajuste é realizado variando-se os limites da taxa de aprendizagem exponencial. A Tabela 8 resume todos os testes realizados para o ajuste.

A constante  $\tau$  é mantida fixa e igual a

$$\tau = N_{ep}/3, \quad (5.2)$$

em que  $N_{ep}$  é o número de épocas. O teste se inicia com valores arbitrários para o limite superior e o inferior da taxa de aprendizagem, representados por  $\eta_i$  e  $\eta_f$ , respectivamente, na Tabela 8.

A tabela é montada a partir de uma varredura em que os limites superior e inferior da taxa de aprendizado são alterados em busca de melhora nas taxas de acerto, principalmente em relação ao conjunto de falhas, por possuir mais amostras.

**Tabela 8. Resultados no ajuste dos limites da taxa de aprendizagem.**

| Limites                               | Resultados de Treinamento |              |                 |                  | Resultados de Validação |              |                 |                  |
|---------------------------------------|---------------------------|--------------|-----------------|------------------|-------------------------|--------------|-----------------|------------------|
|                                       | $CR_{Norm}$               | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Norm}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| $\eta_i = 0,05$<br>$\eta_f = 0,001$   | 68,30                     | 69,43        | 20,91           | 19,77            | 63,33                   | 59,28        | 27,35           | 18,05            |
| $\eta_i = 0,05$<br>$\eta_f = 0,0005$  | 65,74                     | 65,48        | 21,60           | 20,03            | 58,66                   | 58,69        | 27,81           | 18,97            |
| $\eta_i = 0,05$<br>$\eta_f = 0,002$   | 70,71                     | 69,53        | 14,60           | 15,33            | 64,33                   | 59,68        | 21,56           | 13,04            |
| $\eta_i = 0,05$<br>$\eta_f = 0,004$   | 72,20                     | 71,74        | 19,01           | 18,74            | <b>68,66</b>            | 61,63        | 23,24           | 16,64            |
| $\eta_i = 0,05$<br>$\eta_f = 0,006$   | 71,74                     | 69,28        | 15,73           | 22,32            | 65,33                   | 60,22        | 24,93           | 17,42            |
| $\eta_i = 0,025$<br>$\eta_f = 0,004$  | 72,82                     | 73,28        | 19,86           | 16,76            | 63,66                   | 63,91        | 25,57           | 15,71            |
| $\eta_i = 0,0125$<br>$\eta_f = 0,004$ | <b>74,51</b>              | <b>73,64</b> | 15,62           | 18,95            | 66,00                   | <b>63,97</b> | 24,27           | 16,52            |
| $\eta_i = 0,009$<br>$\eta_f = 0,004$  | <b>74,41</b>              | <b>73,94</b> | 15,44           | 11,71            | <b>69,00</b>            | <b>63,30</b> | 23,81           | 10,48            |
| $\eta_i = 0,006$<br>$\eta_f = 0,004$  | 72,71                     | 73,43        | 18,23           | 17,66            | 65,66                   | 62,72        | 22,18           | 15,26            |

Fonte: Autor desta dissertação.

Após todos os testes, os limites escolhidos para taxa de aprendizagem variável são 0,009 e 0,004. Estes valores fazem a rede alcançar os melhores resultados equilibrados, tanto para conjunto de validação como para conjunto de treinamento.

#### 5.1.2.4 Ajuste da Divisão do Conjunto de Treino/Teste e Validação

Até então, todos os testes foram realizados com 70% dos dados normais sendo usados para treinamento, e 20% para o teste de parada prematura. Do conjunto de falhas, é escolhida, aleatoriamente, igual quantidade de amostras em relação ao conjunto normal. Os 10% restantes do conjunto normal são usados para validação, juntamente com todos os dados do conjunto de falha que não são usados para treino/teste. Para esclarecer, a Tabela 9 mostra tanto os percentuais como os valores

absolutos de cada classe e do total de vetores de amostras usados para treinamento, teste e validação.

Pode-se observar na Tabela 9, que o equilíbrio na quantidade de dados, do conjunto normal e do conjunto de falha, é equilibrado no treinamento e teste, porém o conjunto de validação possui uma quantidade muito maior de dados de falha.

**Tabela 9. Divisão do conjunto de dados para treinamento.**

|               | TR  |     | TS  |     | VAL |     | Total |     |
|---------------|-----|-----|-----|-----|-----|-----|-------|-----|
|               | %   | Qtd | %   | Qtd | %   | Qtd | %     | Qtd |
| <b>Normal</b> | 70% | 29  | 20% | 8   | 10% | 5   | 100%  | 42  |
| <b>Falha</b>  | 12% | 29  | 3%  | 8   | 85% | 215 | 100%  | 252 |
| <b>Total</b>  | 20% | 58  | 5%  | 16  | 75% | 220 | 100%  | 294 |

Fonte: Autor desta dissertação.

A divisão percentual, mostrada na Tabela 10, refere-se ao percentual para classe Normal, que é usado como base para a divisão dos conjuntos de dados. Analisando-se esta tabela, observa-se que a única combinação que parece trazer alguma melhora é usando-se 80% para treino, 10% para teste e 10% para validação, pois a taxa de acerto na classificação do conjunto de falha apresenta uma melhora significativa, porém o acerto no conjunto normal apresenta uma queda proporcional. Além disso, os desvios padrão nos acertos de ambas as classes aumentou, portanto a divisão que vinha sendo usada é mantida após estes testes, ou seja, 70% do conjunto de dados normal para treino, 20% para teste e 10% para validação.

**Tabela 10. Resultados por ajuste na divisão dos conjuntos de treinamento, teste e validação.**

| Divisão (TR,TS, VAL) | Resultados de Treinamento |              |                 |                  | Resultados de Validação |              |                 |                  |
|----------------------|---------------------------|--------------|-----------------|------------------|-------------------------|--------------|-----------------|------------------|
|                      | $CR_{Norm}$               | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Norm}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| 70%, 20%,10%         | 72,82                     | 75,07        | 18,16           | 14,87            | 67,00                   | 63,83        | 24,85           | 13,44            |
| 80%, 10%,10%         | 62,26                     | 79,02        | 25,13           | 16,78            | 61,20                   | 69,97        | 30,88           | 16,68            |
| 70%, 10%,20%         | 62,92                     | 81,38        | 23,48           | 15,26            | 58,36                   | 71,00        | 24,63           | 15,99            |
| 70%, 15%,15%         | 67,69                     | 76,35        | 27,97           | 16,17            | 58,50                   | 66,58        | 29,28           | 16,36            |
| 60%, 30%,10%         | 69,87                     | 74,00        | 20,50           | 16,01            | 60,80                   | 63,46        | 26,17           | 15,05            |
| 60%, 20%,20%         | 67,27                     | 72,176       | 22,29           | 16,87            | 60,00                   | 63,69        | 23,52           | 15,75            |
| 50%, 40%,10%         | 67,50                     | 79,35        | 21,78           | 12,66            | 58,40                   | 68,25        | 27,65           | 11,93            |
| 50%, 30%,20%         | 72,64                     | 72,71        | 16,85           | 15,98            | 62,36                   | 59,75        | 20,03           | 15,52            |

Fonte: Autor desta dissertação.

### 5.1.2.5 Ajuste do Número de Épocas

Uma vez que todos os ajustes são feitos nos parâmetros da rede MLP, resta verificar a influência do número de épocas no resultado final. Todos os testes anteriores foram feitos usando 1000 épocas, no entanto, já havia sido observado que um número maior de épocas é necessário para que a rede aprendesse bem os dados de treinamento. O número de 1000 épocas fora escolhido porque, aproximadamente, 80% das paradas prematuras ocorrem antes de 1000 épocas. No entanto, os 20% de paradas que ocorrem após as 1000 épocas podem ser importantes para melhorar o desempenho da rede neural, principalmente porque as taxas de acerto estão muito baixas até então.

Analisando a Tabela 11, é possível se perceber que, quanto maior é o número de épocas, maiores são os acertos da rede. Entretanto, o tempo de treinamento também aumenta consideravelmente, o que faz com que o valor final tenha que ser limitado. Os resultados usados para comparação com a rede MLP/ELM são aqueles obtidos através de 5000 épocas de treinamento.

**Tabela 11. Taxas de acerto obtidas em relação ao número de épocas utilizadas.**

| Resultados de Treinamento |              |              |                 |                  | Resultados de Validação |              |                 |                  |
|---------------------------|--------------|--------------|-----------------|------------------|-------------------------|--------------|-----------------|------------------|
| $N_{ep}$                  | $CR_{Norm}$  | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Norm}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| 1000                      | 72,87        | 74,87        | 17,29           | 18,57            | 67,66                   | 63,49        | 26,38           | 15,88            |
| 2000                      | 78,35        | 78,30        | 13,46           | 14,47            | 68,66                   | 64,67        | 20,38           | 12,65            |
| 3000                      | 81,28        | 75,28        | 15,54           | 17,42            | 72,66                   | 61,61        | 21,23           | 14,12            |
| 4000                      | 80,30        | 77,79        | 14,01           | 14,19            | 73,0                    | 64,31        | 22,30           | 11,58            |
| <b>5000</b>               | <b>80,25</b> | <b>78,66</b> | <b>15,62</b>    | <b>17,33</b>     | <b>73,33</b>            | <b>64,53</b> | <b>19,34</b>    | <b>14,32</b>     |

Fonte: Autor desta dissertação.

### 5.1.2 Topologia MLP/ELM

Para definir a topologia da rede MLP/ELM, praticamente não é necessário arbitrar qualquer valor. Deve-se apenas variar a quantidade de neurônios na camada oculta, e verificar com quantos neurônios ocorre o melhor resultado. Portanto, para cada quantidade de neurônios, a rede é treinada 50 vezes. Assim como na rede MLP/BP, todo o conjunto de dados é usado neste teste.

Devido à alta dimensionalidade, característica da rede ELM, a quantidade de neurônios foi variada de 5 a 145. Na Tabela 12, é mostrada a taxa média de acerto por classe a cada aumento de 10 neurônios.



É possível observar um crescimento na taxa média de acerto à medida que a quantidade de neurônios na camada oculta aumenta. A partir de 115 neurônios, o acerto passa a ser de 100% para ambas as classes. Isto significa que a rede decorou completamente os dados apresentados, em relação aos rótulos impostos. Claramente, não há como selecionar uma topologia para rede MLP/ELM desta forma, pois é preciso levar em consideração a generalização, como é mostrado no item 5.1.2.1, a seguir. É interessante observar que um teste tão exaustivo só se torna viável devido à grande velocidade de treinamento da rede MLP/ELM.

**Tabela 12. Resultados da rede ELM ao se usar 100% dos dados para treinamento.**

| $N_h$ | $CR_{Normal}$ | $CR_{Falha}$ |
|-------|---------------|--------------|
| 5     | 55,03         | 62,21        |
| 15    | 71,53         | 71,71        |
| 25    | 79,00         | 75,10        |
| 35    | 83,42         | 79,75        |
| 45    | 86,75         | 84,07        |
| 55    | 88,67         | 86,85        |
| 65    | 91,75         | 91,14        |
| 75    | 94,50         | 94,32        |
| 85    | 97,42         | 96,92        |
| 95    | 99,17         | 98,82        |
| 105   | 99,89         | 99,92        |
| 115   | 100,00        | 100,00       |
| 125   | 100,00        | 100,00       |
| 135   | 100,00        | 100,00       |
| 145   | 100,00        | 100,00       |

Fonte: Autor desta dissertação.

#### 5.1.2.1 Generalização da MLP/ELM

O conjunto de dados é dividido em conjunto de treinamento e de validação. Novamente, como a quantidade de dados do conjunto normal é menor do que a quantidade de dados do conjunto de falha, a primeira foi usada como valor de base. Desta forma, para treinamento, são escolhidos aleatoriamente 90% dos dados do conjunto normal e uma quantidade equivalente, em valor absoluto, de dados do conjunto de falha. Todos os demais dados são usados no conjunto de validação.

Os resultados para o conjunto de validação são mostrados na Tabela 13. O teste é feito variando-se a rede de 5 a 115 neurônios, pois com 115 é mostrado que a rede aprende totalmente o conjunto de dados. Os resultados são mostrados, de 10 em 10 neurônios, na tabela.

Tabela 13. Resultados para teste de generalização da rede ELM de 5 a 115 neurônios.

| Resultado de Validação (generalização) |               |              |
|--|---------------|--------------|
| $N_h$                                  | $CR_{Normal}$ | $CR_{Falha}$ |
| 5                                      | <b>48,40</b>  | <b>61,28</b> |
| 15                                     | <b>64,40</b>  | <b>60,47</b> |
| 25                                     | <b>63,20</b>  | <b>61,09</b> |
| 35                                     | 61,20         | 58,5         |
| 45                                     | 58,80         | 58,04        |
| 55                                     | 57,20         | 57,26        |
| 65                                     | 66,80         | 56,87        |
| 75                                     | 57,60         | 55,82        |
| 85                                     | 56,00         | 54,60        |
| 95                                     | 52,40         | 51,19        |
| 105                                    | 53,20         | 51,61        |
| 115                                    | 47,60         | 51,71        |

Fonte: Autor desta dissertação.

Tabela 14. Resultados para teste de generalização da rede ELM de 5 a 25 neurônios.

| Resultado de Validação (generalização) |               |              |
|--|---------------|--------------|
| $N_h$                                  | $CR_{Normal}$ | $CR_{Falha}$ |
| 5                                      | 48,40         | 61,28        |
| 6                                      | 56,00         | 58,56        |
| 7                                      | 59,20         | 57,77        |
| 8                                      | 51,20         | 59,45        |
| 9                                      | 59,20         | 59,35        |
| 10                                     | 55,60         | 59,92        |
| 11                                     | 59,20         | 60,67        |
| 12                                     | 62,40         | 59,93        |
| 13                                     | 58,00         | 61,65        |
| 14                                     | 61,60         | 62,20        |
| 15                                     | 64,40         | 60,47        |
| 16                                     | <b>61,60</b>  | <b>63,35</b> |
| 17                                     | <b>58,80</b>  | <b>63,76</b> |
| 18                                     | <b>66,40</b>  | <b>62,98</b> |
| 19                                     | <b>68,80</b>  | <b>62,92</b> |
| 20                                     | <b>69,20</b>  | <b>62,40</b> |
| 21                                     | <b>69,20</b>  | <b>62,96</b> |
| 22                                     | 60,40         | 62,57        |
| 23                                     | 63,60         | 62,07        |
| 24                                     | 60,00         | 61,00        |
| 25                                     | 63,20         | 61,09        |

Fonte: Autor desta dissertação.

Analisando-se a Tabela 13, é possível se perceber que as melhores capacidades de generalização estão entre 5 e 25 neurônios. Portanto, faz-se uma análise minuciosa dentro desta faixa. O resultado é apresentado na Tabela 14.

Examinando-se a Tabela 14, observa-se que as topologias que possuem melhores capacidades de generalização estão entre 16 e 21 neurônios. As topologias

com 19 e 21 neurônios, na camada oculta, parecem as mais adequadas, por conseguirem tanto os melhores acertos no conjunto normal, quanto um acerto comparável aos melhores, no conjunto com falhas. Como os resultados são muito próximos, a melhor topologia é a de 19 neurônios, por alcançar melhores acertos do que as precedentes, e possuir acertos comparáveis às posteriores.

### 5.1.2.2. Ajustes na ELM

Para se manter um padrão de comparação, todos os ajustes em relação à normalização, feitos para rede MLP/BP, são mantidos para rede MLP/ELM.

Durante a escolha da topologia, os resultados com o uso de 16 a 21 neurônios, na camada oculta, são muito próximos. Portanto, após a seleção definitiva de atributos, que é mostrada no item 5.2, é feito um novo teste para melhorar o critério de escolha do número de neurônios na camada escondida, na MLP/ELM.

Na Tabela 15, são mostrados os resultados da MLP/ELM após a seleção final de atributos (mostrada no item 5.2). Os dois melhores resultados em cada coluna estão em negrito. A quantidade de neurônios escolhida é 20, pois faz a rede alcançar bons resultados e há um maior equilíbrio entre acerto no conjunto de falha e no normal.

**Tabela 15. Resultados após seleção de atributos da rede MLP/ELM.**

| $N_h$ | Resultados de Validação |              |                 |                  | Resultados de Treinamento |              |                 |                  |
|-------|-------------------------|--------------|-----------------|------------------|---------------------------|--------------|-----------------|------------------|
|       | $CR_{Normal}$           | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ | $CR_{Normal}$             | $CR_{Falha}$ | $\sigma_{Norm}$ | $\sigma_{Falha}$ |
| 16    | 72,80                   | 64,52        | <b>20,88</b>    | 4,60             | 81,88                     | 75,33        | 6,88            | 7,63             |
| 17    | 68,40                   | 64,54        | 21,41           | 5,06             | 81,64                     | 76,50        | 6,64            | 5,89             |
| 18    | 70,00                   | 64,24        | 24,32           | 4,68             | 83,20                     | 76,07        | <b>4,78</b>     | <b>5,17</b>      |
| 19    | <b>76,40</b>            | 63,85        | 22,01           | <b>4,56</b>      | 83,48                     | 74,62        | 6,54            | 5,88             |
| 20    | <b>74,80</b>            | <b>65,02</b> | <b>20,12</b>    | <b>4,17</b>      | <b>82,64</b>              | <b>77,56</b> | 5,99            | 6,16             |
| 21    | 70,80                   | <b>65,86</b> | 21,45           | 4,67             | <b>83,76</b>              | <b>77,64</b> | <b>5,80</b>     | <b>4,39</b>      |

Fonte: Autor desta dissertação.

## 5.2 Seleção de Atributos

Uma vez definidas as topologias das redes MLP, tanto treinada pelo algoritmo *back-propagation* (MLP/BP) como pelo algoritmo ELM (MLP/ELM), usam-se tais topologias para fazer alguns testes mais detalhados, que permitam uma melhor escolha dos atributos das redes neurais.

A partir da análise do espectro de frequências das correntes de linha, em trabalhos apresentados por Penman (1994) e Thomson (2001), são analisadas, exaustivamente, quais as harmônicas mais relevantes para detecção de cada falha,

em motores de indução. No Apêndice B, estão os detalhes sobre as análises feitas por Penman (1994) e Thomson (2001), para falha de curto-circuito entre bobinas estatóricas do MIT, que são usadas como base para a seleção de atributos de entrada das redes neurais, nesta dissertação.

Para permitir uma rápida conferência dos leitores, pode-se adiantar que os atributos selecionados, como entrada para as redes, são as frequências aproximadas:  $0,5f_1$ ;  $1,5f_1$ ;  $2,5f_1$ ;  $3f_1$ ;  $5f_1$ ;  $7f_1$ .

Em resumo, os procedimentos desta seleção foram feitos usando-se a rede *MLP/ELM*, com a topologia definida anteriormente, para se obterem resultados de classificação com diferentes combinações de atributos extraídos do espectro de frequência, procedimento este detalhado no item 5.2.2. O algoritmo *ELM* foi escolhido devido à sua velocidade de treinamento. A partir dos resultados obtidos com cada combinação de atributos, decisões foram sendo tomadas até se chegar à combinação final.

A primeira combinação de atributos (AT1) foi formada por todas as harmônicas aproximadas, obtidas a partir da equação de Penman (1994), como pode ser conferido no Apêndice B.

A segunda combinação de atributos (AT2) foi obtida a partir de uma análise de variâncias de cada atributo obtido pela equação de Penman (1994), como é detalhado a seguir no item 5.2.1.

Sequencialmente, então, algumas componentes foram incluídas ou retiradas para formação de novas combinações de atributos. Estas combinações foram apresentadas à rede e decisões de manter ou não as novas combinações foram feitas a partir da taxa de classificação da rede neural. Os detalhes deste processo estão mostrados no item 5.2.2.

### **5.2.1 Análise de Variâncias**

As 16 componentes de frequência aproximadas obtidas com base na equação de Penman (1994) são:  $0,5f$ ;  $1f$ ;  $1,5f$ ;  $2f$ ;  $2,5f$ ;  $3f$ ;  $3,5f$ ;  $4f$ ;  $4,5f$ ;  $5f$ ;  $5,5$ ;  $6f$ ;  $6,5f$ ;  $7f$ ;  $7,5f$ ; e  $8f$ .

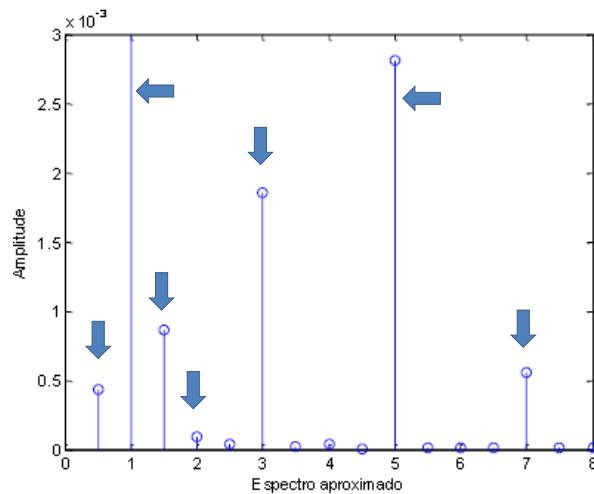
Para análise de variâncias, é montada uma matriz com 16 colunas referentes às 16 componentes acima citadas. Cada linha contém as componentes de

frequência para cada condição de ensaio, perfazendo um total de 294 linhas. Na Figura 12, é mostrado o gráfico da variância pelo espectro aproximado.

Pela Figura 12, percebe-se que há maior variância em algumas harmônicas do espectro, que passam a formar a segunda combinação de atributos a ser testada na rede neural:  $0,5f_1$ ;  $1f_1$ ;  $1,5f_1$ ;  $2f_1$ ;  $3f_1$ ;  $5f_1$  e  $7f_1$ . É importante observar que a ordem de grandeza da fundamental é, dezenas de vezes, superior à das demais e, por isso, ela não aparece por completo no gráfico ampliado.

Além da análise de variâncias, para garantir que atributos selecionados tenham uma boa representatividade da falha de curto-circuito entre espiras, outras combinações são formadas e testadas, na seção 5.2.2.

**Figura 12. Gráfico ampliado de variâncias por componentes aproximadas do espectro.**



**Fonte: Autor desta dissertação.**

### **5.2.2. Combinações de Atributos**

A primeira combinação de atributos, AT1, é formada pelas 16 harmônicas obtidas a partir da equação de Penman, e a segunda combinação, AT2, é feita a partir da primeira, após uma análise de variâncias. Muitas outras combinações são possíveis, por isso, algumas delas foram testadas na rede a partir de algumas hipóteses levantadas. Cada teste é repetido 100 vezes para se obterem resultados médios. Na Tabela 16, são mostradas as combinações testadas, as hipóteses levantadas, o diagnóstico do resultado de classificação da rede, taxas de acerto da classe normal, e falha no conjunto de validação que levam ao diagnóstico. Este é

dado, prioritariamente, com base no aumento da taxa de acerto para ambas as classes. Quando ocorre uma melhora nos resultados, a nova combinação é mantida como base para os próximos testes.

Ao reduzir, com base na análise de variâncias, a quantidade de harmônicas usadas na detecção de falha, a classificação relativa à classe normal melhora, porém, a relativa à classe com falha piora. Isto significa que, provavelmente, alguns atributos importantes para a classificação da falha são retirados no processo, ainda assim é dado um diagnóstico de melhora pela grande quantidade de atributos que foram retirados. A combinação de atributos testada em seguida, AT3, exclui a frequência fundamental. A retirada da componente fundamental melhora, significativamente, o acerto na classificação do motor em estado normal e, praticamente, mantém a taxa de classificação do motor sob falha.

A tentativa seguinte, em AT4, consiste em adicionar os múltiplos de frequência  $2,5f_1$  e  $3,5f_1$ . Segundo o estudo de Thomson (2001), para um motor de 4 polos, estes seriam os múltiplos característicos da falha. De fato, após a inclusão dos novos atributos, há melhora no acerto para a classe do motor com falha, apesar de haver redução do acerto do motor normal. Entretanto, como dois atributos são adicionados, decide-se retirar o múltiplo  $3,5f_1$  para verificar sua relevância, o que gera a combinação AT5. Após a retirada de  $3,5f_1$ , o resultado melhora novamente.

Na tentativa de se reduzir ainda mais o vetor de atributos, retira-se, sequencialmente, um a um, os harmônicos restantes, formando-se as combinações de AT6 a AT11, porém apenas quando se retira  $2f_1$ , em AT9, há uma melhora nos resultados. Observando-se a Figura 12, percebe-se que a variância deste atributo, nos conjuntos de dados, é baixa.

Em AT3, também há boa classificação do conjunto normal, no entanto, uma melhora no conjunto de falha torna-se mais relevante, devido à quantidade maior de amostras neste conjunto, fazendo com que a escolha final dos atributos de entrada das redes, seja para as componentes harmônicas da combinação AT9:  $0,5f_1; 1,5f_1; 2,5f_1; 3f_1; 5f_1$  e  $7f_1$ .

**Tabela 16. Combinações de atributos testadas em busca de melhor acerto de classificação.**

| Combi-<br>nação | Componentes  | Hipótese  | Diagnóstico     |              |              |
|-----------------|--|---|-----------------|--------------|--------------|
|                 |  |   | do<br>Resultado | $CR_{Norm}$  | $CR_{Falha}$ |
| AT1             | $0,5f_1; 1f_1; 1,5f_1$<br>... $8f_1$ .                             | Equação de Penman.  | Base            | <b>62,80</b> | <b>62,69</b> |
| AT2             | $0,5f_1; 1f_1; 1,5f_1;$<br>$2f_1; 3f_1; 5f_1; 7f_1$                | Análise de variâncias.  | Melhora         | <b>72,2</b>  | <b>58,26</b> |
| AT3             | $0,5f_1; 1,5f_1; 2f_1;$<br>$3f_1; 5f_1; 7f_1$                      | A frequência fundamental é retirada por possuir ordem de grandeza muito diferente das demais, podendo dificultar o ajuste de pesos. | Melhora         | <b>78,4</b>  | <b>57,32</b> |
| AT4             | $0,5f_1; 1,5f_1; 2f_1;$<br>$2,5f_1; 3f_1; 3,5f_1;$<br>$5f_1; 7f_1$ | Componentes selecionados com base na teoria de Thomson (2001) como indicativos de falha.  | Melhora         | <b>72,40</b> | <b>61,80</b> |
| AT5             | $0,5f_1; 1,5f_1; 2f_1;$<br>$2,5f_1; 3f_1; 5f_1;$<br>$7f_1$         | Retira-se $3,5f_1$ para verificar sua importância, já que sua ordem de grandeza é muito pequena.                                    | Melhora         | <b>73,40</b> | <b>62,01</b> |
| AT6             | $0,5f_1; 1,5f_1; 2f_1;$<br>$2,5f_1; 3f_1; 5f_1$                    | Retira-se arbitrariamente $7f_1$  | Piora           | 69,40        | 63,63        |
| AT7             | $0,5f_1; 1,5f_1; 2f_1;$<br>$2,5f_1; 3f_1; 7f_1$                    | Retira-se arbitrariamente $5f_1$  | Piora           | 72,8         | 61,51        |
| AT8             | $0,5f_1; 1,5f_1; 2f_1;$<br>$2,5f_1; 5f_1; 7f_1$                    | Retira-se arbitrariamente $3f_1$  | Piora           | 70,60        | 59,32        |
| AT9             | $0,5f_1; 1,5f_1; 2,5f_1;$<br>$3f_1; 5f_1; 7f_1$                    | Retira-se arbitrariamente $2f_1$  | Melhora         | <b>76,60</b> | <b>63,06</b> |
| AT10            | $0,5f_1; 2,5f_1;$<br>$3f_1; 5f_1; 7f_1$                            | Retira-se arbitrariamente $1,5f_1$  | Piora           | 69,28        | 59,5         |
| AT11            | $1,5f_1; 2,5f_1;$<br>$3f_1; 5f_1; 7f_1$                            | Retira-se arbitrariamente $0,5f_1$  | Piora           | 66,00        | 59,02        |

Fonte: Autor desta dissertação.

## 6 ANÁLISE DOS RESULTADOS

Neste capítulo, são apresentados os métodos e resultados obtidos com os classificadores neurais. A ordem em que são apresentados segue a cronologia do desenvolvimento da dissertação, com o intuito de esclarecer o porquê das hipóteses levantadas e testadas, à medida que os resultados são obtidos e analisados. O desenvolvimento dessas hipóteses, por vezes, conduz a resultados secundários, porém relevantes.

### 6.1 Teste de separabilidade do conjunto de dados

O classificador linear Perceptron Simples (PS), com um neurônio, é aplicado, na tentativa de solucionar e atestar a separabilidade do conjunto de dados. Os demais parâmetros da rede (taxa de aprendizagem, número de épocas, termo de momento e atributos) são reproduzidos da rede *MLP/BP*.

Na Tabela 17, são ilustrados os resultados médios alcançados após 50 treinamentos do Perceptron Simples. Nesta e nas demais tabelas,  $N_w$  representa o número de pesos da topologia, considerando-se os atributos mais o bias,  $CR$  representa a taxa de classificação correta (*Classification Rate*), e  $\sigma$  representa o desvio padrão. Os subscritos  $TR$ ,  $TS$  e  $VAL$  referem-se aos conjuntos de treinamento, teste e validação, respectivamente.

**Tabela 17. Resultados do Perceptron Simples.**

| ANN | $N_w$ | $CR_{TR}$ | $\sigma_{TR}$ | $CR_{VAL}$ | $\sigma_{VAL}$ |
|-----|-------|-----------|---------------|------------|----------------|
| PS  | 7     | 60,1      | 20,6          | 50,5       | 19,2           |

Fonte: Autor desta dissertação.

Como se pode notar pela tabela, o PS não apresenta capacidade de classificar satisfatoriamente o conjunto de dados ( $CR_{TR} = 60,1\%$ ), o que sugere que o problema é não linearmente separável.

A partir da premissa de se ter um conjunto de dados não linearmente separável, as redes Perceptron Multicamadas, com topologia e parâmetros selecionados a partir dos testes apresentados no capítulo 5, são aplicadas na detecção de falha, como mostrado na seção seguinte.



## 6.2 Redes MLP aplicadas como classificadores

Como já mencionado, são usadas redes MLP treinadas por dois diferentes algoritmos. Na Tabela 18, são apresentadas as taxas de acerto médias globais, ou seja, considerando-se ambas as classes obtidas com cada rede.

**Tabela 18. Resultados globais médios das MLP.**

| ANN         | $N_W$ | $CR_{TR}$ | $\sigma_{TR}$ | $CR_{TS}$ | $\sigma_{TS}$ | $CR_{VAL}$ | $\sigma_{VAL}$ |
|-------------|-------|-----------|---------------|-----------|---------------|------------|----------------|
| MLP/<br>BP  | 41    | 78,0      | 8,8           | 74,9      | 11,0          | 64,9       | 11,3           |
| MLP/<br>ELM | 161   | 82,5      | 3,7           | -         | -             | 65,2       | 4,8            |

Fonte: Autor desta dissertação.

Os resultados modestos da MLP/BP e MLP/ELM na classificação do conjunto de validação dão uma indicação de que o problema em mãos é uma tarefa difícil, mesmo usando-se classificadores não lineares. É importante salientar que, mesmo apresentando taxas de reconhecimento próximas, a rede MLP/BP leva alguma vantagem, sobre a MLP/ELM, pelo número reduzido de pesos, favorecendo a implementação embarcada do classificador.

Devido à observação de resultados globais semelhantes para ambas as redes, os resultados médios de classificação obtidos para cada classe separadamente são apresentados na Tabela 19, visando-se elucidar possíveis divergências no desempenho das redes. Nela, pode-se observar consistência dos classificadores, evidenciada pelo equilíbrio das taxas de acerto por classe, ou seja, na classificação da classe normal e da falha, há uma diferença pequena entre taxas de acerto individuais (10% no conjunto de treinamento e 5% nos conjuntos de teste e de validação). Os valores apresentados são o percentual em relação ao total da classe considerada.

**Tabela 19. Taxas de classificação média por classe.**

| MLP | Classe | $CR_{TR}$ | $CR_{TS}$ | $CM_{VAL}$ |
|-----|--------|-----------|-----------|------------|
| BP  | Normal | 80,21     | 77,45     | 69,33      |
|     | Falha  | 75,69     | 72,36     | 64,83      |
| ELM | Normal | 87,84     | -         | 75,60      |
|     | Falha  | 77,00     | -         | 64,99      |

Fonte: Autor desta dissertação.

Dentre os 50 classificadores projetados para gerar os resultados médios, apresentados nas Tabela 18 e na Tabela 19, são selecionados, para cada um dos algoritmos de treinamento BP e ELM, o classificador que apresenta, simultaneamente, alta taxa de acerto global nos conjuntos de treinamento e de teste, equilíbrio entre as taxas de acerto por classe, e que não apresenta erro de classificação, nos dados relativos à operação normal, no conjunto de validação. Estes critérios foram utilizados com o intuito de se aplicar, em trabalhos futuros, os classificadores em monitoramento real, com reduzida ocorrência de falsos negativos para operação normal. As redes ainda são submetidas ao algoritmo de poda, denominado CAPE (Medeiros e Barreto, 2013), para se removerem redundâncias danosas, e se reduzir o tempo de processamento e recursos de memória, em aplicação embarcada e em tempo real. Os resultados são apresentados na Tabela 20.

**Tabela 20. Resultado para RNA específicas**

| <b>MLP</b>          | <b><math>N_w</math></b> | <b><math>CR_{TR}</math></b> | <b><math>CR_{TS}</math></b> | <b><math>CR_{VAL}</math></b> | <b>Classe</b> | <b><math>CR_{TR}</math></b> | <b><math>CR_{TS}</math></b> | <b><math>CR_{VAL}</math></b> |
|---------------------|-------------------------|-----------------------------|-----------------------------|------------------------------|---------------|-----------------------------|-----------------------------|------------------------------|
| <b>BP</b>           | 41                      | 89,7                        | 81,8                        | 68,5                         | Normal        | 94,9                        | 81,8                        | 100                          |
|                     |                         |                             |                             |                              | Falha         | 84,7                        | 81,8                        | 67,9                         |
| <b>BP/<br/>CAPE</b> | 34                      | 87,1                        | 81,8                        | 70,2                         | Normal        | 94,9                        | 81,8                        | 100                          |
|                     |                         |                             |                             |                              | Falha         | 79,5                        | 81,8                        | 69,3                         |
| <b>ELM</b>          | 161                     | 84,1                        | -                           | 63,8                         | Normal        | 90,2                        | -                           | 100                          |
|                     |                         |                             |                             |                              | Falha         | 78,0                        | -                           | 63,9                         |

**Fonte: Autor desta dissertação.**

O algoritmo de poda não consegue remover pesos na rede treinada pelo algoritmo *ELM*, mas promove uma redução de 17% no número de pesos ( $N_w$ ) do classificador treinado com o algoritmo *BP*, além de apresentar uma melhora na taxa de acerto no conjunto de validação,  $CR_{VAL}$ , de 68,5% para 70,2%. Na coluna da extrema direita da tabela, pode-se ver que o acerto para classe normal, no conjunto de validação, é de 100% para todas as redes. Para se ressaltar a importância disto é preciso lembrar: primeiramente, que o objetivo futuro destes classificadores é que componham um sistema de detecção contínua e em tempo real; e, em segundo lugar, que se o curto-circuito de alta impedância não for detectado em primeira instância, ainda poderá ser detectado à medida que evolui, pois os dados que compõem o conjunto de treinamento das redes incluem amostras do curto-circuito de baixa impedância, que são mais facilmente detectados. Por outro lado, a ocorrência de um

falso positivo, para o conjunto de falha, seria problemática, uma vez que a detecção de um curto-circuito, em um sistema em condição normal de operação, acarretaria um desligamento desnecessário do motor.

Outra observação importante, sobre a tabela, pode ser feita tomando-se como exemplo a taxa de acerto global no conjunto de validação, para a rede *MLP/BP* (68,5%), e a taxa de acerto no conjunto de falha, para a mesma rede (67,9%). Estes números mostram que a contribuição para a taxa de acerto global vem, quase que inteiramente, do conjunto de falha. O motivo para tal é bem discutido no capítulo 4, no qual se mostra que o número de amostras coletadas para a condição de falha do motor é 6 vezes maior do que o coletado para a condição normal. Como, no treinamento da rede, se busca equilíbrio entre a quantidade de dados normais e de falha, enquanto o primeiro é treinado usando-se 70% do total de dados de sua classe, o outro é treinado com apenas 12%. Em números absolutos, no treinamento, são usados 29 dos 42 dados referentes à classe normal, e 29 dos 252 dados referentes à falha. A consequência disto, é que a quantidade de dados de validação, para o conjunto de falha, torna-se desproporcional, sendo ela a maior responsável pela definição da taxa global de acerto. No entanto, o equilíbrio no treinamento é necessário, para que a rede não fique tendenciosa.

Para efeito de comparação, Coelho e Medeiros (2013) propõem um classificador neural, usando Mapa Auto-Organizável, treinado com o mesmo conjunto de dados. O classificador apresenta taxa de acerto global de 87,5%, porém, com acerto de 52% para dados referentes à operação normal, e 94,5% para a condição de falha, o que demonstra uma clara tendência da rede em classificar os dados como falha, além de que, provavelmente, tem uma alta chance de apontar falsos positivos para o conjunto de falha.

Em busca de melhorar as taxas de acerto obtidas aqui, opta-se pela aplicação da técnica da Análise de Componentes Principais (*Principal Components Analysis - PCA*) (Haykin, 1999) sobre o conjunto de dados original.

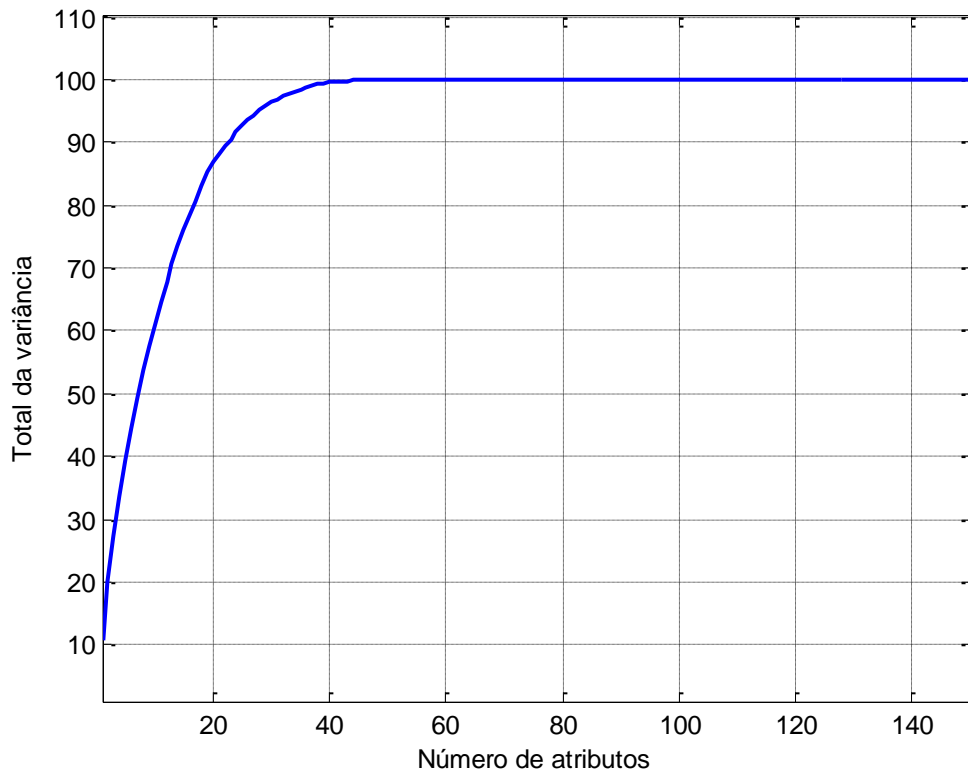
### **6.3 Aplicação do PCA**

Uma dificuldade, comum em problemas de reconhecimento de padrões, é a extração de características. Esta, por sua vez, consiste na definição de um espaço de características a partir do espaço de dados original (Haykin, 1999). A técnica da

análise das componentes principais permite, através de uma transformação linear, se projetar os vetores do espaço de dados original, que contém 5000 atributos (referentes ao espectro de frequência discretizado), em um espaço de características que pode ser reduzido. Isto pode ser feito escolhendo-se os vetores que representam a maior parte da variância dos dados.

No primeiro teste, aplica-se o *PCA* a todo o conjunto de dados, sem discriminação de conjunto de treinamento, teste ou validação, com o objetivo de se descobrir quantos atributos são necessários para representar 99,9999% da variância total do conjunto. Na Figura 13, há um gráfico da representatividade da variância total do conjunto de dados versus o número de atributos selecionados pela técnica.

**Figura 13. Gráfico da representatividade da variância total do conjunto de dados versus o número de atributos selecionados pela técnica.**



**Fonte: Autor desta dissertação.**

Observando-se o gráfico da Figura 13, nota-se que a partir de pouco mais de 40 atributos, não é possível notar qualquer mudança no total da variância. Entretanto, com base dos dados usados para se gerar o gráfico, percebe-se que, a partir de 85 atributos, atingem-se 99,99% da variância total dos dados, havendo

diferenças apenas da terceira casa decimal em diante. A partir de 124 atributos, atingem-se 99,9999%, havendo diferenças depois da quinta casa decimal.

Diferentemente deste teste, em que as quantidades de atributos são obtidas usando-se todo o conjunto de dados, durante a aplicação nos classificadores, a técnica de *PCA* é usada apenas no conjunto de treinamento e, portanto, gera quantidades diferentes, dependendo dos vetores que são sorteados para tal conjunto. Portanto, para se encontrar uma quantidade de atributos mais adequada, quando houver separação entre conjuntos, são feitas dez seleções aleatórias do conjunto de treinamento, sendo obtida uma curva média semelhante à da Figura 13.

Percebe-se que, quando os dados são separados em 80% treinamento - 20% validação (seguindo o padrão de equilíbrio descrito na seção 5.2.1.4), 99 atributos são necessários para se alcançar 99,9999% da variância. Esta quantidade de atributos é tomada como base para se verificarem as taxas de classificação alcançadas na aplicação de uma rede Perceptron Simples.

### 6.3.1 Perceptron Simples com extração de características a partir do PCA

Como a aplicação da técnica do *PCA* envolve a geração de um conjunto de atributos diferentes, é aplicado novamente um Perceptron Simples para atestar a separabilidade dos dados. Os resultados médios obtidos, para a taxa de acerto de cada classe, para diferentes quantidades de atributos, são mostrados na Tabela 21.

**Tabela 21. Taxa de classificação usando Perceptron Simples com diferentes quantidades de atributos obtidos através do PCA.**

| Qtd Atributos | $N_W$ | Var Med (%) | $CR_{TR}$ (%) |       | $CR_{VAL}$ (%) |       |
|---------------|-------|-------------|---------------|-------|----------------|-------|
|               |       |             | Normal        | Falha | Normal         | Falha |
| 99            | 694   | 99,9999     | 100           | 100   | 94,62          | 75,27 |
| 72            | 505   | 99,99       | 100           | 100   | 99,23          | 73,93 |
| 45            | 316   | 99,90       | 96,80         | 91,80 | 94,62          | 67,20 |

Fonte: Autor desta dissertação.

Observando os resultados, percebe-se que, no novo espaço de atributos, o problema é linearmente separável, uma vez que 100% dos dados são corretamente classificados no conjunto de treinamento, tanto com 99 atributos, quanto com 72 atributos. Observa-se, ainda, uma considerável melhora nas taxas de classificação médias no conjunto de validação, em relação às aquelas apresentadas na Tabela 19. Entretanto, não se pode perder de vista a necessidade de se projetar cada vetor de

atributos, do espaço original, no espaço de baixa dimensionalidade, o que cria dificuldades para uma implementação embarcada e em tempo real.

No conjunto de validação, é notável uma divergência entre as taxas de acerto da classe relativa às condições de falha e às da normal. Deve-se lembrar que o conjunto de dados contém uma quantidade de 252 amostras referentes à falha, e de apenas 42 para a condição normal, e que o equilíbrio na quantidade de dados de treinamento faz com que muitas amostras de falha não participem desta fase, sendo usadas apenas para validação. Numericamente, se 34 vetores referentes à condição normal são usados para o treinamento da rede, apenas 34 referentes à falha são usados (divididos igualmente entre as classes), compondo os demais o conjunto de validação.

Uma solução para tentar se aproveitar melhor o conjunto de dados disponível é usar um novo modo de rotulação dos dados, durante o treinamento. Neste modo, o treinamento é feito com cada subclasse tendo seu próprio rótulo (valor de saída desejada). Para isto, a rede precisa de 7 neurônios, um para indicar a classe normal e outros seis para as subclasses (AI1, AI2, AI3, BI1, BI2, BI3). Apresentam-se à rede, durante o treinamento, vetores de todas as subclasses, aproveitando todo o conjunto de dados. Numericamente, se 34 vetores normais são apresentados à rede, também são apresentados 34 da subclasse AI1, 34 da subclasse AI2 e assim por diante. No entanto, é importante salientar que todas as subclasses, apesar de possuírem rótulos diferentes, são classificadas apenas como falha. Em outras palavras, os erros ocorridos entre diferentes subtipos de falhas são desconsiderados.

A seguir, na Tabela 22, é mostrada uma matriz de confusão, considerando todas as subclasses. Pode-se perceber que a quantidade de erros cometidos entre as subclasses é relativamente alta. Entretanto, ao se desconsiderarem estes erros, obtêm-se altas taxas de acerto, como pode ser comprovado na Tabela 23.

Na Tabela 23, são mostradas as quantidades de atributos e a variância média atingida usando-se tais quantidades, assim como as taxas de acerto por classe, obtidas para os conjuntos de treinamento e de validação. É importante observar que a maior quantidade de rótulos, considerados no conjunto de dados, aumenta a quantidade de atributos necessária para se atingirem as variâncias apresentadas na Tabela 21.

A apresentação dos resultados, nas Tabela 21 e Tabela 23, é importante para se demonstrar que, à medida que se usam mais atributos, melhora-se a representatividade da variância do conjunto de dados, e conseguem-se maiores taxas de acerto; em contrapartida, pode-se reduzir, significativamente, o número de atributos para reduzir o esforço computacional, desde que se aceite uma diminuição nas taxas de acerto.

**Tabela 22. Matriz de Confusão considerando-se todas as subclasses.**

|        | Normal (%) | AI1 (%) | AI2 (%) | AI3 (%) | BI1 (%) | BI2 (%) | BI3 (%) |
|--------|------------|---------|---------|---------|---------|---------|---------|
| Normal | 95,38      | 0       | 4,62    | 0       | 0       | 0       | 0       |
| AI1    | 0          | 64,62   | 0       | 6,15    | 27,6    | 1,54    | 0       |
| AI2    | 3,08       | 0       | 73,85   | 0       | 4,62    | 18,45   | 0       |
| AI3    | 4,62       | 0       | 0       | 93,84   | 0       | 0       | 1,54    |
| BI1    | 0          | 35,38   | 0       | 0       | 64,62   | 0       | 0       |
| BI2    | 0          | 3,08    | 20      | 0       | 0       | 76,92   | 0       |
| BI3    | 0          | 0       | 3,08    | 4,62    | 3,08    | 9,23    | 80      |

Fonte: Autor desta dissertação.

**Tabela 23. Taxa de classificação usando-se Perceptron Simples, usando-se 7 rótulos, com diferentes quantidades de atributos obtidos através do PCA.**

| Qtd Atributos | $N_W$ | Var Med (%) | $CR_{TR}$ (%) |       | $CR_{VAL}$ (%) |       |
|---------------|-------|-------------|---------------|-------|----------------|-------|
|               |       |             | Normal        | Falha | Normal         | Falha |
| 346           | 2423  | 99,9999     | 100           | 100   | 93,08          | 98,46 |
| 207           | 1450  | 99,99       | 100           | 100   | 95,38          | 99,10 |
| 60            | 421   | 99,90       | 89,80         | 97,20 | 80,00          | 95,38 |

Fonte: Autor desta dissertação.

Pode-se observar ainda, pela Tabela 23, que as taxas de classificação de cada classe estão mais próximas entre si, o que leva à hipótese de que a utilização de todo o conjunto de dados pode ser a causa desta melhora.

Para confirmar tal hipótese, decide-se equilibrar as quantidades de dados de cada classe, através da geração de novos dados para o conjunto normal. Isto é feito através da adição de ruído branco ao conjunto original, como explicado na seção 6.4, a seguir.

#### 6.4 Geração de novos dados para conjunto normal

Na Figura 11, na seção 4.2, fica clara a diferença entre a quantidade de dados coletados para o conjunto normal (42 dados) e a quantidade de dados para falha (252 dados).

Deseja-se criar novos dados, para o conjunto normal, a partir da adição de ruído ao conjunto original. No entanto, os dados criados devem participar apenas do conjunto de treinamento, para que os conjuntos de teste e validação contenham somente dados reais, coletados na bancada de testes.

O método usado para a divisão do conjunto de dados em treinamento e validação, portanto, deve ser mudado de forma que seja feita com base no conjunto de falha. Exemplificando, 80% (ou a porcentagem desejada) dos vetores de cada classe são usados para composição do conjunto de treinamento. Como o conjunto de falhas possui mais dados, esta forma de divisão deixa menos dados representativos da classe normal. É nesta fase que a quantidade de dados normais é aumentada, acrescentando novos dados, gerados a partir da inclusão de ruído nos próprios dados normais selecionados. Desta forma, fica garantido que os conjuntos de teste e validação possuam apenas dados reais. Na Tabela 24, são mostradas as quantidades obtidas para cada conjunto de dados, ao se tomar 80% como porcentagem de base para o conjunto de falha.

O conjunto normal possui 34 atributos do conjunto original, que se referem a 80% da quantidade total desta classe e, além disso, foram acrescentados 168 dados ruidosos de forma a totalizar 202 dados, ou seja, a quantidade referente a 80% do total de dados de falha. O conjunto de validação é composto apenas por dados reais coletados na bancada, portanto, possui menor quantidade de dados para o conjunto normal. Não foi considerado o conjunto de teste neste caso, pois não foi detectada necessidade de parada prematura.

**Tabela 24. Quantidade de dados obtidos após adição de ruídos, tomando-se 80% do conjunto de falha como base para o treinamento.**

| Classe        | Quantidade |    |     | Total |
|---------------|------------|----|-----|-------|
|               | TR         | TS | VAL |       |
| <b>Normal</b> | 34+168     | -  | 8   | 210   |
| <b>Falha</b>  | 202        | -  | 50  | 252   |
| <b>Total</b>  | 404        | -  | 58  | 462   |

Fonte: Autor desta dissertação.

#### **6.4.1 Adição de Ruído**

O ruído foi gerado a partir de uma função randômica uniforme (*rand*), que retorna valores entre dois limites pré-estabelecidos. O teste foi feito a partir de redes *MLP/BP* e *MLP/ELM*.



Na Tabela 25, são mostradas as taxas de acerto obtidas com a rede *MLP/BP* e *MLP/ELM*, em relação ao nível de ruído adicionado. Os conjuntos de treinamento e validação foram divididos conforme a exemplificação anterior.

**Tabela 25. Taxas de acerto em relação ao ruído adicionado.**

|                    | <i>MLP/BP</i> |              |              |              | <i>MLP/ELM</i> |              |              |              |
|--------------------|---------------|--------------|--------------|--------------|----------------|--------------|--------------|--------------|
|                    | $CR_{TR}$     |              | $CR_{VAL}$   |              | $CR_{TR}$      |              | $CR_{VAL}$   |              |
| Nível de ruído     | Normal        | Falha        | Normal       | Falha        | Normal         | Falha        | Normal       | Falha        |
| <b>0</b>           | 94,74         | 85,42        | 95,08        | 80,58        | 85,54          | 63,13        | 84,87        | 66,06        |
| <b>0 – 0,00001</b> | <b>94,74</b>  | <b>86,05</b> | <b>95,38</b> | <b>81,79</b> | <b>85,08</b>   | <b>65,26</b> | <b>85,38</b> | <b>66,05</b> |
| <b>0 – 0,0001</b>  | 94,46         | 86,11        | 94,46        | 81,21        | 84,69          | 66,29        | 84,24        | 67,20        |
| <b>0 – 0,001</b>   | 91,38         | 87,71        | 75,38        | 85,37        | 61,85          | 75,29        | 85,72        | 77,84        |
| <b>0 - 0,01</b>    | 86,82         | 98,78        | 41,85        | 94,11        | 8,92           | 96,08        | 78,44        | 96,30        |
| <b>0 - 0,1</b>     | 83,54         | 99,75        | 18,77        | 99,32        | 0              | 100          | 78,85        | 100          |

Fonte: Autor desta dissertação.

Como pode ser percebido na Tabela 25, mesmo quando o nível de ruído adicionado foi zero, ou seja, houve apenas repetição dos dados originais na composição do novo conjunto, ocorreu considerável melhora nas taxas de acerto em relação às redes *MLP*, em que havia discrepância entre as quantidades de dados de cada classe (Tabela 19). Porém, como está destacado na segunda linha, o nível mínimo de ruído (quinta casa decimal) adicionado gerou o melhor resultado de classificação na rede neural.

No lado direito da Tabela 25, são apresentados os resultados obtidos para a rede *MLP/ELM* com 20 neurônios na camada oculta. É perceptível que o desempenho também foi melhor para o menor nível de ruído testado, no entanto, tornou-se muito inferior ao obtido pela rede *MLP/BP*, fato que não ocorria nos testes precedentes. Portanto, usando-se a quantidade de ruído mencionada, testou-se a rede *MLP/ELM* para outras quantidades de neurônios na camada oculta. Os resultados obtidos são apresentados na Tabela 26.

Na Tabela 26, pode-se ver que o uso crescente de neurônios na camada escondida ( $N_h$ ) favorece a classificação até 150 neurônios; para 170 e 190 neurônios, fica visível uma diminuição drástica nos acertos, tanto para conjunto de treinamento quanto para de validação. Para explicar o ocorrido, é preciso lembrar que o algoritmo de treinamento *ELM* envolve a inversão de uma matriz, para se obterem os pesos da rede. Esta matriz possui dimensão  $q \times q$ , onde  $q$  é o número de neurônios na camada escondida. Portanto, o aumento na quantidade de neurônios pode tornar pouco apurado o cálculo da inversa, devido ao grande aumento de dimensionalidade. Isto

pode ser avaliado através do índice de condicionamento  $R_{cond}$  da matriz, que é definido pela norma da matriz multiplicado pela norma de sua inversa. Um valor elevado obtido indica um mau condicionamento para a solução.

**Tabela 26. Taxas de classificação por classe, para várias quantidades de neurônios ocultos na rede MLP/ELM**

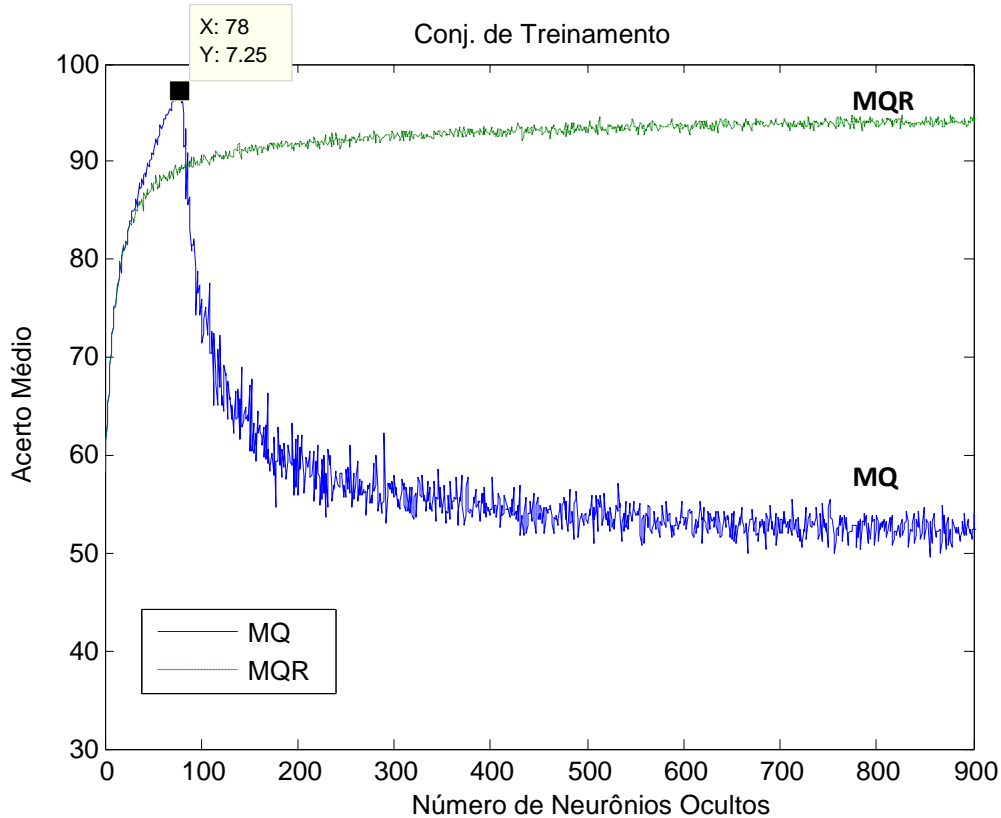
| MLP/ELM |           |       |            |       |            |
|---------|-----------|-------|------------|-------|------------|
| $N_h$   | $CM_{TR}$ |       | $CM_{VAL}$ |       | $R_{cond}$ |
|         | Normal    | Falha | Normal     | Falha |            |
| 20      | 85,38     | 66,05 | 85,08      | 65,26 | 2,3745e+09 |
| 50      | 88,89     | 75,87 | 88,31      | 72,13 | 2,3983e+11 |
| 100     | 90,47     | 82,70 | 89,54      | 74,08 | 3,1701e+15 |
| 120     | 91,56     | 84,64 | 89,69      | 73,84 | 3,7184e+16 |
| 150     | 94,01     | 86,15 | 94,77      | 72,26 | 6,5677e+17 |
| 170     | 86,77     | 83,15 | 84,92      | 66,74 | 6,9663e+18 |
| 190     | 67,61     | 73,19 | 69,08      | 61,16 | 1,7901e+19 |

Fonte: Autor desta dissertação.

#### 6.4.2 Algoritmo ELM usando mínimos quadrados recursivos

Como discutido no capítulo 3, os mínimos quadrados recursivos oferecem uma solução alternativa onde não é necessário se realizar um cálculo de inversão de matriz. Uma comparação entre o algoritmo *ELM* convencional e o *ELM* treinado pelo método recursivo é apresentada através do gráfico da Figura 14. O gráfico mostra a taxa de acerto de duas redes *MLP/ELM*, uma treinada pelos MQ em batelada e outra pelos MQR. No gráfico da Figura 14, é mostrado o acerto médio de 50 treinamentos, com vetores sorteados aleatoriamente, porém, usando-se o mesmo conjunto de dados, para ambas, versus o número de neurônios ocultos. A condição utilizada para gerar o gráfico foi aquela em que não foram inseridos dados ruidosos no conjunto de dados.

**Figura 14. Taxa de acerto média por número de neurônios no conjunto de treinamento.**



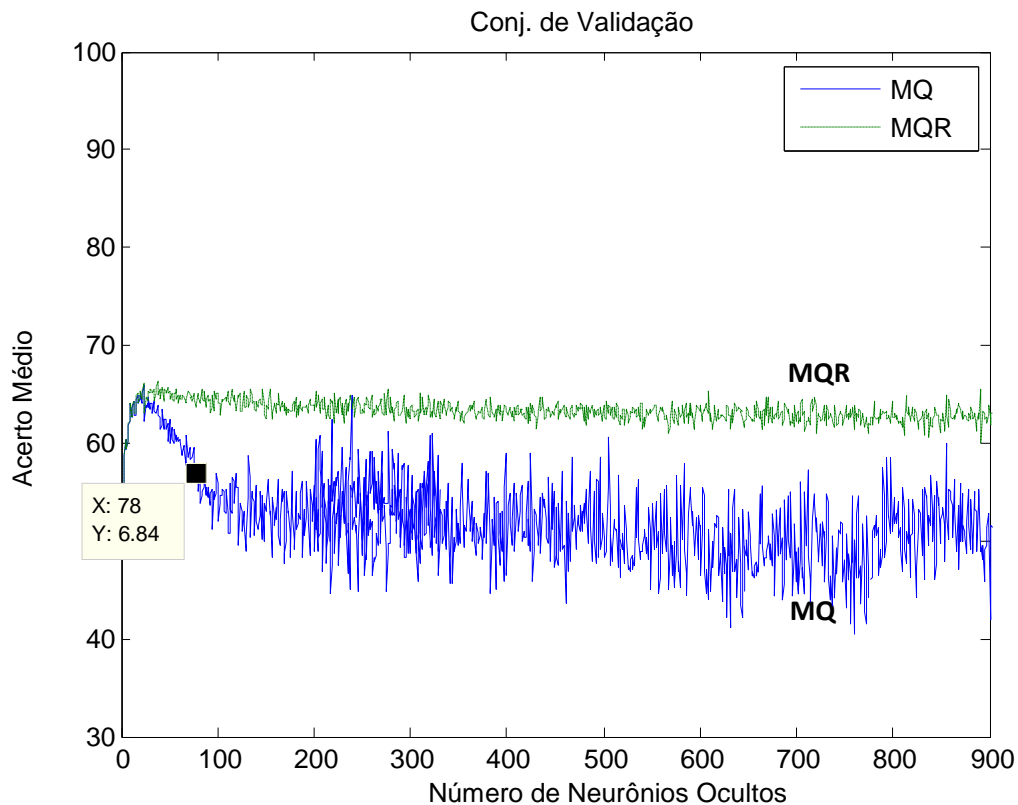
**Fonte: Autor desta dissertação.**

Observando este gráfico, percebe-se que os algoritmos causaram desempenho semelhante na rede neural até, aproximadamente, 50 neurônios. Daquele ponto em diante o desempenho da *MLP/ELM* com MQ continua a crescer vertiginosamente, enquanto o da *MLP/ELM* com os MQR, apesar manter um pequeno aumento no desempenho, já começa um processo de saturação. Próximo a 80 neurônios, o desempenho da *MLP/ELM* com os MQ inicia uma queda íngreme, enquanto o outro segue ainda um lento crescimento. Em, aproximadamente, 100 neurônios os gráficos se cruzam; o desempenho da *MLP/ELM* com MQR se mantém, enquanto o da *MLP/ELM* com os MQ continua a cair até se estabilizar com taxas próximas de 50%, que significam que a rede não está sendo mais capaz de diferenciar as classes do problema, já que esta taxa indica uma classificação praticamente aleatória.

No gráfico da Figura 15, podem-se comparar os resultados obtidos com o conjunto de validação. Percebe-se que, no ponto marcado nos gráficos, em 78 neurônios, a taxa de acerto é máxima no conjunto de treinamento para o *MLP/ELM*

com os MQ, mas no conjunto de validação a taxa está bem abaixo da obtida com a *MLP/ELM* com os MQR. A conclusão é que, à medida que a quantidade de neurônios se torna maior, os mínimos quadrados recursivos possuem melhor capacidade de generalização e deixam a rede mais estável. As desvantagens disto são a necessidade de se ajustar um novo parâmetro,  $P$ , como explicado no capítulo 3, seção 3.3.3, e a diminuição na velocidade de computação do algoritmo de treinamento.

**Figura 15. Taxa de acerto média por número de neurônios no conjunto de validação.**



Fonte: Autor desta dissertação.

## 6.5 Estimativa de custo computacional

O conhecimento de custo computacional torna-se especialmente importante para sistemas de detecção em tempo real embarcados em microprocessadores. Neste tipo de aplicação, é necessária uma resposta rápida do sistema, para que possa ser considerada detecção em “tempo real”, mas se tem uma maior limitação na velocidade de processamento. Para implementação em conversores de frequência comerciais, a redução do custo envolvido se torna ainda

mais importante, pois se deseja adicionar a função de detecção de falhas sem alterar a arquitetura eletrônica vigente.

Para fazer uma estimativa preliminar do custo computacional relacionado aos principais classificadores envolvidos, é feito um levantamento da quantidade de operações matemáticas necessárias à computação dos classificadores usados, incluindo-se a etapa de seleção de atributos.

### 6.5.1 Custo computacional no PS

Pela equação (3.6), pode-se concluir que são necessárias  $p$  multiplicações e  $P$  somas algébricas (incluindo o *bias*) para se obter a entrada líquida,  $u_i(t)$ , de cada um dos  $m$  neurônios de saída do Perceptron. Portanto, a quantidade total de somas e produtos necessários à computação desta rede é, respectivamente,

$$N_S^{PS} = p \cdot m, \quad (6.1)$$

$$N_P^{PS} = p \cdot m, \quad (6.2)$$

em que  $N_S$  é o número de somas algébricas e  $N_P$  é o número de produtos;  $p$  e  $m$ , já definidos no capítulo 3, são a quantidade de atributos de entrada e a quantidade de neurônios de saída, respectivamente. O sobrescrito *PS* faz referencia ao Perceptron Simples.

Cada uma das entradas líquidas é submetida a uma função de ativação para gerar a saída da rede. No caso do OS, esta função pode ser uma função sinal. Computacionalmente, a função sinal é implementada através de dois testes condicionais, como está mostrado na equação (3.10). O número de funções de ativação,  $N_A$ , necessárias é igual ao número de neurônios de saída.

$$N_A = m. \quad (6.3)$$

### 6.5.2 Custo computacional da MLP

Na *MLP* com uma camada oculta, pode-se observar, pela equação (3.68), que são necessárias  $p$  multiplicações e  $p$  somas algébricas (incluindo o *bias*) para se obter a entrada líquida,  $u_i^{(h)}(t)$ , de cada um dos  $q$  neurônios da camada escondida. Portanto, o número de somas e produtos necessários à computação da primeira etapa da rede é, respectivamente,

$$N_{SH}^{MLP} = p \cdot q, \quad (6.4)$$

$$N_{PH}^{MLP} = p \cdot q, \quad (6.5)$$

em que  $N_{SH}$  é o número de somas algébricas na camada escondida e  $N_{PH}$  o número de produtos na camada escondida;  $p$  e  $q$ , já definidos no capítulo 3, são a quantidade de atributos de entrada e a quantidade de neurônios na camada escondida, respectivamente. O sobrescrito  $MLP$  faz referência à rede  $MLP$ .

Na camada escondida o processo se repete, com a diferença que há  $m$  neurônios de saída e  $q$  parâmetros de entrada. Assim, o número de somas e multiplicações necessárias para computar essa etapa é dado, respectivamente, por

$$N_{SO}^{MLP} = q \cdot m, \quad (6.6)$$

$$N_{PO}^{MLP} = q \cdot m, \quad (6.7)$$

em que  $N_{SO}$  é o número de somas algébricas na camada de saída e  $N_{PO}$  é o número de produtos na camada de saída;  $m$ , já definido no capítulo 3, é a quantidade de neurônios na camada de saída.

O número total de somas algébricas realizadas nas duas etapas da  $MLP$ ,  $N_S^{MLP}$ , é dado por

$$N_S^{MLP} = q \cdot (p + m), \quad (6.8)$$

e o número total de produtos,  $N_P^{MLP}$ , é dado por

$$N_P^{MLP} = q \cdot (p + m). \quad (6.9)$$

Cada uma das entradas líquidas, tanto na camada escondida quanto na camada de saída, é submetida a uma função de ativação. Na camada escondida e na de saída, as funções de ativação usadas são todas tangentes hiperbólicas. Portanto, a quantidade de tangentes hiperbólicas,  $N_{TH}^{MLP}$ , necessárias é igual à quantidade de neurônios na rede  $MLP$ .

$$N_{TH}^{MLP} = q + m. \quad (6.10)$$

A implementação prática de uma função tangente hiperbólica envolve diversas operações matemáticas de soma e produto, pois é computada com base na série de Taylor. A quantidade exata depende da quantidade de termos na série, como é explicado a seguir.

#### 6.5.2.1 Custo computacional da Tangente Hiperbólica

A função tangente hiperbólica é dada por

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6.11)$$

Para implementação prática, usa-se uma versão rápida dada por

$$\tanh(x) = 1 - \frac{2}{e^{2x} + 1} \quad (6.12)$$

A exponencial no denominador é calculada através da série de Taylor,

$$e^x = (1 + x) + \left(\frac{x^2}{2!}\right) + \left(\frac{x^3}{3!}\right) + \left(\frac{x^4}{4!}\right) \dots \quad (6.13)$$

Na computação da série, os termos constantes são calculados antes da implementação para reduzir o custo computacional e os produtos de  $x$  realizados são armazenados e reaproveitados no cálculo do próximo termo.

Uma função tangente hiperbólica presente na biblioteca matemática de um compilador usa 25 termos da série de Taylor para fazer tal aproximação. A partir do truncamento dos termos da série é possível reduzir o custo computacional envolvido. Na Tabela 27,  $N_{Taylor}$ , é a quantidade de termos usados para a computação da série de Taylor;  $N_S$ , é o número de somas computadas;  $N_P$  é o número de produtos computados; e  $erro_{m\acute{a}x}$  é o maior erro cometido na aproximação.

**Tabela 27. Custo computacional da série de Taylor.**

| $N_{Taylor}$ | $N_S$ | $N_P$ | $erro_{m\acute{a}x}$ |
|--------------|-------|-------|----------------------|
| 24           | 24    | 46    | 5,54e-14             |
| 12           | 12    | 22    | 9,67e-06             |
| 8            | 8     | 14    | 7,70e-04             |
| 6            | 6     | 10    | 4,40e-03             |
| 4            | 4     | 6     | 2,13e-02             |
| 2            | 2     | 2     | 1,16e-01             |

**Fonte: Autor desta dissertação.**

Nesta tabela, é importante notar a redução no custo computacional (número de somas e produtos) em função do truncamento da série de Taylor e o erro máximo cometido em cada caso. Para 12 termos, o custo computacional foi reduzido em mais da metade, e o erro cometido está na sexta casa decimal. Para oito termos, o erro ocorre na quarta casa decimal, e assim por diante, conforme a tabela. Em uma rede neural, em que a quantidade de tangentes hiperbólicas computadas é igual ao número de neurônios da rede, o truncamento da série de Taylor pode se tornar uma ferramenta fundamental à redução de custo computacional, desde que o erro cometido seja aceitável aos requisitos do projeto.

### 6.5.3 Custo computacional da Seleção de atributos

Esta etapa também envolve um número de operações consideráveis e não pode ser menosprezada. Neste trabalho, duas formas de seleção de atributos são feitas, a seleção de múltiplos da frequência de acionamento e a *PCA*.

#### 6.5.3.1 Seleção por múltiplos da frequência de acionamento

Como detalhado no Apêndice B, devido ao escorregamento, não é possível armazenar o valor exato dado pelos múltiplos  $0,5f_1$ ;  $1,5f_1$ ;  $2,5f_1$ ;  $3f_1$ ;  $5f_1$ ;  $7f_1$ . Não à toa, estes são chamados de múltiplos aproximados da frequência de acionamento.

Para encontrar os valores usados na rede, é feita uma varredura de  $\pm 2,7$  Hz ao redor de cada um dos múltiplos aproximados da frequência de acionamento, e selecionado aquele com maior valor de amplitude.

Como a resolução é de 0,1 Hz, a varredura feita para cada espectro aproximado é de 54 pontos. Em cada um destes pontos, deve ser feito um teste condicional para saber se aquele é o maior valor. Se for, então uma atribuição deve ser feita. Considerando-se cada teste condicional e atribuição como uma operação equivalente a uma soma algébrica, pode-se estimar o custo computacional desta etapa.

A quantidade máxima de operações, neste método de seleção por múltiplos da frequência,  $N_{MF}$ , será

$$N_{MF} = 54 \text{ testes condicionais} \cdot 54 \text{ atribuições} \cdot 6 \text{ espectros} = 17.496 . \quad (6.14)$$

#### 6.5.3.2 Seleção através do *PCA*

Com a utilização do *PCA*, chega-se a uma matriz de transformação que deve ser aplicada ao espectro de frequências completo, para se obter o espaço de características, ou atributos, usados na entrada da rede.

Esta etapa envolve a multiplicação do espectro de frequência, de dimensão  $dim = 1 \times 5000$  (espectro de frequência de 0 a 500 Hz, com resolução de 0,1 Hz), por uma matriz de transformação de dimensões  $dim = 5000 \times p$ , em que  $p$  é a quantidade de atributos definidos para entrada da rede.

Este processo é semelhante à computação de uma multiplicação de matrizes, portanto, o número de somas algébricas,  $N_S^{PCA}$ , necessárias é

$$N_S^{PCA} = 1 \cdot p \cdot (5000 - 1), \quad (6.15)$$



$$N_S^{PCA} = 4999 \cdot p, \quad (6.16)$$

e o número de multiplicações,  $N_P^{PCA}$ , é

$$N_P^{PCA} = 5000 \cdot p \quad (6.17)$$

Caso a resolução seja diferente de 0.1 Hz, as expressões genéricas para quantidade de somas e produtos, respectivamente, são dadas por

$$N_S^{PCA} = p \cdot (N - 1), \quad (6.18)$$

$$N_P^{PCA} = p \cdot N, \quad (6.19)$$

em que  $N$  é o número de pontos obtidos na discretização do espectro de frequência.

### 6.5.5 Custo computacional nos classificadores

Nesta seção, as equações mostradas no item 6.5 são usadas para se calcular a quantidade de operações realizadas em 4 classificadores selecionados. Estes resultados são mostrados na Tabela 28, que contém a topologia de cada um deles, as taxas de classificação que cada um alcançou e o número de operações estimadas para a computação de cada um deles, incluindo a etapa de seleção dos atributos no espectro de Fourier.

**Tabela 28. Comparação de custo computacional entre 4 classificadores selecionados.**

| Classificador | Topologia | $CR_{VAL}$ | $N_S$ | $N_P$ | $N_{TH}$ | $N_A$ | $N_{MF}$ | $N_S^{PCA}$ | $N_P^{PCA}$ |
|---------------|-----------|------------|-------|-------|----------|-------|----------|-------------|-------------|
| <b>MLP_1</b>  | 6-5-1     | 80%        | 35    | 35    | 6        | -     | 17.496   | -           | -           |
| <b>MLP_2</b>  | 6-120-1   | 73%        | 840   | 840   | 121      | -     | 17.496   | -           | -           |
| <b>PS_1</b>   | 99-1      | 75%        | 99    | 99    | -        | 1     | -        | 494.901     | 495.000     |
| <b>PS_2</b>   | 207-7     | 98%        | 1449  | 1449  | -        | 7     | -        | 1.034.793   | 1.035.000   |

Fonte: Autor desta dissertação.

Na primeira coluna da Tabela 28, são mostrados os tipos de redes neurais usados nos classificadores. Na coluna seguinte, é mostrada a topologia, no formato  $p-q-m$ , para as *MLP*, e  $p-m$  para os Perceptrons. Na terceira coluna, é mostrado um valor aproximado para a taxa de classificação média no conjunto de validação de cada classificador, representado por  $CR_{VAL}$ . Na coluna  $N_S$ , é mostrada a quantidade de somas necessárias aos classificadores, e na coluna  $N_P$ , a quantidade de produtos. Na coluna  $N_{TH}$ , é indicada a quantidade de tangentes hiperbólicas usadas nos classificadores, e na coluna  $N_A$ , a quantidade funções de ativação do tipo sinal. A coluna  $N_{MF}$  refere-se à quantidade de operações equivalentes às somas algébricas que são necessárias, no processo de seleção de atributos por múltiplos da frequência. As duas últimas colunas referem-se à quantidade de somas e de produtos,

respectivamente, usadas para transformar o espectro de frequência em um vetor de atributos de entrada do classificador.

Na Tabela 28, é interessante observar que o processo de seleção de atributos é responsável por grande parte do custo computacional dos classificadores, especialmente a seleção de atributos através do *PCA*. Isto pode ser observado claramente no classificador *PS\_2*, em que as quantidades de somas e de produtos na seleção de atributos são mais de mil vezes maiores do que as somas e produtos na computação da RNA. Uma maneira de se reduzir o custo computacional desta etapa do processo seria diminuir a resolução na discretização dos espectros de frequência.

Também podemos perceber que, como a computação de uma tangente hiperbólica envolve o uso da série de Taylor, que para 25 termos, necessita 25 somas e 48 produtos, então para a rede *MLP\_2*, que possui 121 neurônios, a computação de todas as tangentes hiperbólicas envolve 3025 somas e 5808 produtos. Ou seja, há um alto custo computacional associado à computação das funções de ativação dos vários neurônios da rede.

Pode-se perceber que a topologia que alcança as melhores taxas de classificação, *PS\_2*, é também a que envolve o maior custo computacional. Este resultado demonstra que a escolha de uma rede neural, baseada apenas na taxa de classificação, pode ser inadequada em aplicações para as quais a redução do custo computacional seja imprescindível. A *MLP\_1* alia boa taxa de classificação e menor custo computacional, podendo ser uma escolha mais viável na implementação em microprocessador.

## 7 CONCLUSÃO

A partir de técnicas computacionais, foi feita a classificação dos dados em normal ou falha, a partir de dados experimentais. Inicialmente, os classificadores específicos, treinados pelo BP/CAPE e pela ELM, foram capazes de classificar, corretamente, cerca de 70% e 64%, respectivamente, dos dados do conjunto de validação, com acerto de 100% para classe normal. Concluiu-se que busca por atributos, através de modelos matemáticos, gerou um espaço de dados não-linearmente separável, que justifica o uso de redes multicamadas, mas tornou difícil a tarefa de classificação do conjunto de dados em falha e normal.

Com a inclusão de novos dados ao conjunto de treinamento, através da adição de ruído branco, foi possível elevar as taxas de acerto para rede *MLP/BP*, chegando-se a mais de 80%, em média, no conjunto de validação. Este resultado mostra-se interessante, uma vez que a rede *MLP/BP* possui uma topologia reduzida, com 5 neurônios na camada oculta, sendo uma opção atrativa para a implementação prática. Mostra, também, que os resultados iniciais foram prejudicados pela grande diferença na quantidade de dados entre cada classe. A preocupação com a perspectiva de errar a classificação, em 20% das vezes, é minimizada, pois o sistema, funcionando em tempo real, tenderá a perceber a falha à medida que ela evolui. A preocupação é maior no caso de falsos positivos de falha, que poderiam interromper, indevidamente, a operação do motor. Pensando nisto, buscou-se sempre o maior acerto para o conjunto normal. No entanto, é evidente que, apenas após serem feitos testes com o sistema em tempo real, é que se poderá chegar a conclusões mais definitivas.

A adição de ruído também gerou uma melhora nas taxas de classificação com a rede *MLP/ELM*, mas somente quando o número de neurônios ocultos foi aumentado para 100. Ainda assim, a rede treinada pelo BP obteve melhor resultado, além de usar uma quantidade muito menor de neurônios na camada oculta. Isto mostra que, apesar da rapidez de projeto e de treinamento do algoritmo *ELM*, a quantidade de neurônios necessária pode significar um maior custo para aplicação em sistema embarcado. Vale lembrar que variações do algoritmo *ELM* foram e continuam a ser propostos, e que as conclusões aqui obtidas não se estendem a todos eles.

Foi visto que o desempenho do algoritmo *ELM* convencional tende a cair quando a quantidade de neurônios na camada oculta é elevada, devido à necessidade de inversão de matriz na computação do algoritmo. Por isso, os mínimos quadrados recursivos são uma opção para se contornar esta necessidade. O uso dos MQR torna o treinamento mais estável para redes com grande quantidade de neurônios ocultos. Como desvantagem, verifica-se a necessidade de ajustar um parâmetro a mais na rede e a redução na velocidade de treinamento.

Buscando taxas de classificação ainda melhores para o problema, foi usada a técnica de *PCA*. Em conjunto com esta técnica, foi usado um novo método de rotulação e classificação, para se tentar aproveitar melhor os dados disponíveis.

Dessa forma, utilizando-se uma rede Perceptron Simples, conseguiram-se taxas de acerto médias superiores a 98%, para o conjunto de validação, o que leva à conclusão de que a técnica de *PCA* gerou um espaço de características linearmente separável, e tornou a separação mais precisa.

No entanto, para ser criterioso nesta avaliação, não se pode perder de vista que a necessidade de mudança do espaço original de atributos, com 5000 pontos referentes ao espectro de frequência, para um novo espaço através de uma matriz de transformação obtida com o *PCA*, implica em requisitos superiores de memória e processamento em uma implementação prática, em um sistema eletrônico embarcado. (*Frase muito longa; procure dividir em duas*)

Por fim, conclui-se que a solução do problema de detecção de CEEE em MIT é possível, mas deve-se manter em vista a necessidade de implementação prática futura antes de se decidir sobre a melhor topologia. Possivelmente, a rede *MLP/BP* treinada, com auxílio da adição de ruídos, seja a melhor opção, se a rede for implementada em um microprocessador. Apesar da taxa média de acerto com esta rede ter sido de 80%, ela possui a menor topologia dentre as testadas e, conseqüentemente, envolve o menor esforço computacional.

## 7.1 Futuros trabalhos

A bancada de testes e os dados coletados dão margem para novos trabalhos, dos quais alguns já estão em desenvolvimento.

- **Detecção de curto-circuito, usando novos sintomas** - Além de medições de correntes de linha, a aquisição feita na bancada de testes também

coletou dados de vibração e do fluxo axial. A detecção de falha, através destes sintomas, são temas de trabalhos em desenvolvimento. A comparação entre resultados, obtidos com tais técnicas, também é uma projeção para novas pesquisas.

- **Resultados de simulação** - É previsto o uso de um modelo matemático do motor de indução trifásico, para o qual seja possível a aplicação de curto-circuito entre espiras. A análise dos dados obtidos com simulação traz novas perspectivas ao trabalho:
  - Comparar os dados coletados da simulação e das medições na bancada;
  - Treinar classificadores com dados de simulação e testar com dados mensurados na bancada.
  - Aumentar a quantidade de amostras para a classe normal, usando dados de simulação.
- **Aplicação de novas técnicas** - O conjunto de dados formado pode, ainda, ser explorado pelo uso de novas técnicas computacionais para análise e processamento dos dados, como, por exemplo, Máquinas de Vetor de Suporte, para classificação, e transformada Wavelet, para extração de características.
- **Aplicação em conversor de frequência** - A principal perspectiva, para futuros trabalhos, é a aplicação dos diversos tipos de classificadores desenvolvidos, embarcados em conversor de frequência, para que a detecção seja feita em tempo real. A comparação dos classificadores, aplicados desta forma, é importante para evidenciar quais características são mais relevantes na escolha de um classificador específico. Um trabalho deste tipo permite, ainda, analisar a quantidade de memória ocupada pelo classificador, o tempo de resposta, e a ocorrência de falsos positivos, entre outros aspectos práticos.

## REFERÊNCIAS

- ABNT NBR IEC. **Isolação elétrica — Avaliação térmica e designação**. ABNT/CB. [S.l.]. 2012.
- AGUIRRE, L. A. **Introdução à identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3ª. ed. Belo Horizonte: UFMG, 2007. ISBN 9788570415844.
- AIEE COMMITTEE. **The life expectancy of class a random-wound motor insulation as determined by AIEE standard no. 510 test procedure**. American Institute of Electrical Engineers, Part II: Applications and Industry, Transactions of the, v. 78, n. 4, p. 224,228, Sept. 1959. ISSN doi: 10.1109/TAI.1959.6371563.
- BALLAL, M. S. et al. **Detection of inter-turn short-circuit fault in induction motor using theory of instantaneous symmetrical components**. IEEE International Conference on Industrial technology. [S.l.]: [s.n.]. 2006. p. 460-464.
- BARRETO, G. A. **Perceptron Simples**. Universidade Federal do Ceará. Fortaleza. 2007.
- BEZESKY, D. M.; KREITZER, S. **Selecting ASD systems**. Industry Applications Magazine, IEEE, v. 9, p. 39-49, July-Aug. 2003.
- BONNETT, A. H.; SOUKUP, G. **Causes and analysis of stator and rotor failures in three-phase squirrel-cage induction motors**. IEEE Transaction on Industry Applications, v. 28, n. 4, p. 921-937, 1992.
- BOUZID, M. et al. **An effective neural approach for the automatic location of stator interturn faults in induction motor**. IEEE Transactions on Industrial Electronics, v. 55, n. 12, p. 4277-4289, December 2008.
- COELHO, D.; MEDEIROS, C. M. S. **Short circuit incipient fault detection and supervision in a three-phase induction motor with a SOM-based Algorithm**. Book of Advances in Self-Organizing Maps, v. 198, p. 315-323, January 2013. ISSN 978-3-642-35229-4.
- CRUZ, S. M. A. D. **Diagnóstico e Análise de Avarias nos Enrolamentos Estatóricos de Motores de Indução Trifásicos Através da Aplicação do Método dos Referenciais Múltiplos**. Tese, Doutorado. Universidade de Coimbra. Coimbra. 2004.
- DAS, S. et al. **Wavelet aided SVM classifier for stator interturn fault monitoring in induction motors**. IEEE Power and Energy Society General Meeting. [S.l.]: [s.n.]. 2010. p. 1-6.
- DAS, S.; PURKAIT, P.; CHAKRAVORTI, S. **Separating induction Motor Current Signature for stator winding faults from that due to supply voltage unbalances**. Power and Energy in NERIST (ICPEN), 1st International Conference on. [S.l.]: [s.n.]. 2012. p. 1,6, 28-29.
- DATTA, B. N. **Numerical linear algebra and applications**. USA: Brooks/Cole Publishing Company, 1995.
- ENGELBRECHT, A. P. **Introduction to computacional intelligence, in computacional intelligence: An introduction**. 2ª. ed. Chichester: John Wiley & Sons, 2007.

- GAITANG, W.; PING, L. **Dynamic Adaboost Ensemble Extreme Learning Machine**. International Conference on Advanced Computer Theory and Engineering (ICACTE). [S.l.]: IEEE. 2010. p. V3-54-58.
- GAZZANA, D. D. S.; PEREIRA, L. A.; FERNANDES, D. **A system for incipient fault detection and fault diagnosis based on MCSA**. IEEE Transmission and Distribution Conference and Exposition. [S.l.]: [s.n.]. 2010. p. 1-6.
- Huang, G. et al. **Extreme Learning Machine for Regression and Multiclass Classification**. IEEE Transactions on Systems, Man, and Cybernetics -Part B: Cybernetics, v. 42, n. 2, p. 513-529, April 2012. ISSN 1083-4419.
- HAYKIN, S. S. **Neural Networks: a comprehensive foundation**. 2<sup>a</sup>. ed. [S.l.]: Prentice Hall International, 1999.
- HORNIK, K.; STINCHCOMBE, M.; WHITE, H. **Multilayer feedforward networks are universal approximators**. Neural Networks, 1989. 359-366.
- HUANG, G.; QIN-YU, Z.; CHEE-KHEONG, S. **Extreme learning machine: a new learning system of feedforward neural networks**. IEEE International Joint Conference. [S.l.]: [s.n.]. 2004. p. 985-990.
- HYUN, C. C. et al. **Fault detection and isolation of induction motors using recurrent neural networks and dynamic bayesian modeling**. IEEE Transactions on Control Systems Technology, v. 18, n. 2, p. 430-437, March 2010.
- JOKSIMOVIC, G. M.; PENMAN, J. **The detection of inter-turn short circuits in the stator windings of operating motors**. IEEE Transactions on Industrial Electronics, v. 47, n. 5, p. 1078-1084, October 2000.
- KAUFHOLD, M. et al. **Interface phenomena in stator winding insulation**. IEEE Electrical Insulation Magazine, v. 18, n. 2, p. 27-36, March/April 2002.
- KOWALSKI, C. T.; WOLKIEWICZ, M. **Converter-fed induction motor diagnosis using instantaneous electromagnetic torque and power signals**. EUROCON 2009, IEEE. [S.l.]: [s.n.]. 2009. p. 811,816.
- KOWALSKI, C. T.; WOLKIEWICZ, M. **Stator faults diagnosis of the converter-fed induction motor using symmetrical components and neural networks**. Power Electronics and Applications, 13th European Conference on. EPE '09. [S.l.]: [s.n.]. 2009. p. 1,6,8-10.
- MEDEIROS, C. M. D. S. **Controle Digital**. Instituto Federal de Educação, Ciência e Tecnologia do Ceará. Fortaleza. 2008.
- MEDEIROS, C. M. S.; BARRETO, G. A. **A novel weight pruning method for MLP classifiers based on the MAXCORE principle**. Neural Computing and Applications, v. 22, n. 1, p. 71-84, 01 jan. 2013. ISSN 0941-0643.
- MICHE, Y. et al. **OP-ELM: Optimally Pruned Extreme Learning Machine**. IEEE Transactions on Neural Networks, v. 21, n. 1, p. 158-162, January 2010. ISSN 1045-9227.

MORÉ, Jorje J. **The Levenberg-Marquardt algorithm: Implementation and theory**. Numerical Analysis: Lecture Notes in Mathematics. Springer Berlin Heidelberg, v. 630, p. 105-116, 1978. ISBN 978-3-540-08538-6.

NAN, L.; WANG, H. **Ensemble Based Extreme Learning Machine**. IEEE Signal Processing Letters, v. 17, n. 8, p. 754-757, August 2010. ISSN 1070-9908.

NANDI, S.; TOLIYAT, H. A.; XIAODONG, L. **Condition signature analysis to detect induction motor faults**. Energy Conversion, IEEE Transactions on, v. 20, n. no.4, p. 719-729, December 2005.

NANDI, W. T.; TOLIYAT, H. A. **Condition monitoring and fault diagnosis of electrical motors - a review**. Industry Applications Conference. Thirty-Fourth IAS Annual Meeting Conference Record of the 1999 IEEE. [S.l.]: [s.n.]. 1999. p. 197-204.

NATARANJAN, R. **Failure identification of induction motors by sensing unbalanced stator currents**. IEEE Transactions on Energy Conversion, v. 18, n. 2, p. 585-590, December 1989.

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. **Motor and Generators**. NEMA. [S.l.], p. 12-16. 2011.

PENMAN, J. et al. **Detection and location of interturn short circuits in the stator winding of operating motors**. Energy Conversion, IEEE Transactions on, v. IX, p. 652-658, December 1994. ISSN 10.1109/60.368345.

SORIA-OLIVAS, E. et al. **BELM: Bayesian Extreme Learning Machine**. IEEE Transactions on Neural Networks, v. 22, n. 3, p. 505-509, March 2011. ISSN 1045-9227.

STAVROU, A.; SEDDING, H.; PENMAN, J. **Current monitoring for detecting inter-turn short circuits in induction motors**. International Conference on Electric Machines and Drives. [S.l.]: [s.n.]. 1999. p. 345-347.

THOMSON, W. T.; FENGER, M. **Current signature analysis to detect induction motor faults**. Industry Applications Magazine, v. 7, p. 26-34, Jul/Aug 2001.

THORSEN, O.; DALVA, M. **Condition monitoring methods, failure identification and analysis for high voltage motors in petrochemical industry**. Eighth International Conference on Electrical Machines and Drives. [S.l.]: [s.n.]. 1997. p. 109-113.

WEBB, A. R. **Statistical Pattern Recognition**. 2<sup>a</sup>. ed. Malvern: John Wiley & Sons, 2002.

XU, B.-Q.; LI, H.-M.; SUN, L.-L. **Detection of Stator Winding inter-turn short circuit fault in induction motors**. Power System Technology. [S.l.]: [s.n.]. 2004. p. 1005-1009.

Yimin, Y.; Yaonan, W.; Xiaofang, Y. **Bidirectional Extreme Learning Machine for Regression Problem and Its Learning Effectiveness**. IEEE Transactions on Neural Networks, v. 23, n. 9, p. 1498-1505, September 2012. ISSN 2162-237X.

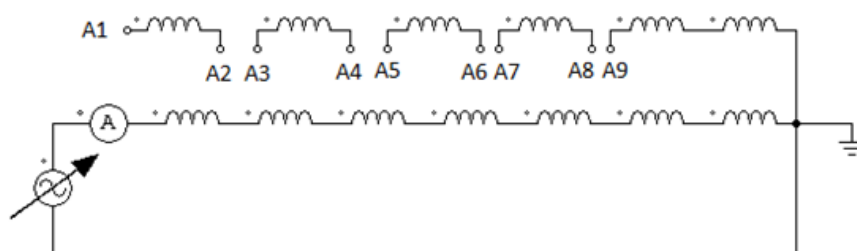


## APÊNDICE A – Ensaio para Determinação dos Percentuais de Espiras Acessíveis através das Derivações Externas após Rebobinamento do Motor.

O motor foi rebobinado de forma a deixar derivações do enrolamento externas à carcaça. Foram dadas, à empresa responsável pelo rebobinamento, especificações sobre a quantidade de espiras desejadas, em cada conjunto de espiras acessíveis externamente. Este ensaio foi feito, portanto, para verificar se as especificações foram corretamente satisfeitas.

Do ensaio obtiveram-se as medidas percentuais dos conjuntos disponíveis, em relação ao número total de espiras das bobinas, em cada fase. Na Figura 16, é mostrado o esquema do ensaio realizado na ‘fase A’ do motor, que foi a fase em que os curtos-circuitos foram aplicados durante a coleta de dados.

**Figura 16. Esquema para ensaio do motor na fase A.**



**Fonte: Autor desta dissertação.**

O ensaio consistiu em se aplicar tensão alternada em uma das fases do motor, através de um variador de tensão, de forma a se manter a corrente limitada ao valor nominal. Em seguida, foi medida a tensão induzida em cada conjunto de espiras no enrolamento da fase A. Na Figura 16, os conjuntos são A1–A2, A3–A4, A5–A6, A7–A8, A9–Terra, porém este último conjunto é na verdade o Grupo 2 de enrolamentos do motor nesta fase (vide Figura 6, p. 47).

Mediu-se, também, a tensão induzida total na fase A. Calculou-se o percentual da tensão de cada conjunto em relação ao valor total. Como a relação de transformação da tensão induzida é proporcional à relação de espiras, tomou-se este valor como o percentual de espiras de cada conjunto.

Na Tabela 29, são mostrados os valores medidos para cada conjunto, e o os valores percentuais em relação à tensão induzida total. Além dos resultados

mostrados na tabela, convém saber que a tensão de alimentação medida foi de 24,97 V, e a corrente medida foi de 1,0 A. Observa-se que a soma dos percentuais de todos os conjuntos de bobinas não chega, exatamente, a 100%, possivelmente, devido aos erros de medição.

**Tabela 29. Medições de tensão e valor percentual para cada conjunto de bobinas na 'fase A'.**

| <b>Conjunto</b> | <b>Tensão Medida (V)</b> | <b>Percentual</b> |
|-----------------|--------------------------|-------------------|
| <b>A1-A2</b>    | 0,024                    | 0,63%             |
| <b>A3-A4</b>    | 0,030                    | 0,78%             |
| <b>A5-A6</b>    | 0,13                     | 3,40%             |
| <b>A7-A8</b>    | 0,17                     | 4,45%             |
| <b>Grupo 2</b>  | 3,42                     | 89,53%            |
| <b>Total</b>    | 3,82                     | 100%              |

**Fonte: Autor desta dissertação.**

Os equipamentos utilizados para o ensaio foram:

- Multímetro digital Minipa ET-2110 - usado para medir as tensões.
- Amperímetro analógico: GANZ HDA-2 - escala de 1 A.
- Motor de Indução WEG, Trifásico do Tipo Gaiola de Esquilo. Dados de placa - 0,75 kW, 220/380 V, 3,02/1,75 A,  $\cos \phi = 0,82$ .

## APÊNDICE B – Análise do Espectro de Frequência Baseada na Teoria de Penman (1994) e Thomson (2001).

Thomson (2001) fez uma análise experimental em motores de indução de baixa tensão, com curto-circuito entre espiras no estator, para verificar alterações na amplitude dos espectros das correntes estatóricas, que caracterizassem o curto-circuito entre espiras, e que não fossem função de nenhum outro tipo de condição no motor, como desbalanceamento das fases, desalinhamento do eixo, quebra de barras rotóricas, etc.

A análise feita no presente trabalho baseou-se nos estudos anteriores de Penman *et al.* (1994), que havia deduzido, matematicamente, a seguinte expressão, para descobrir quais eram os componentes harmônicos, na forma de onda, do fluxo do entreferro, que são função do curto-circuito entre espiras do estator.

$$f_{cc} = \{k \pm n(1 - s)/p\}f_1, \quad (\text{B.1})$$

onde

$f_{cc}$  = componentes que são função do curto-circuito entre espiras

$k = 1, 3, 5, \dots$ ,  $n = 1, 2, 3, \dots$ ,  $s$  = escorregamento,  $p$  = pares de pólos

$f_1$  = frequência fundamental

Para exemplificação, usar-se-á o motor ensaiado acionado por conversor de frequência a 60 Hz, operando a vazio, ou seja, quando o escorregamento pode ser aproximado para zero. Para este caso, usando a expressão (B.1) encontram-se as frequências dependentes do curto, como mostrado na Tabela 30.

**Tabela 30. Módulo das frequências dados pela equação (B.1) quando  $p = 2, f_1 = 60$  Hz.**

| Parâmetros     | $f_{cc1}$ (Hz) | $f_{cc2}$ (Hz) |
|----------------|----------------|----------------|
| <b>k=1,n=1</b> | 90             | 30             |
| <b>k=1,n=2</b> | 60             | 0              |
| <b>k=1,n=3</b> | 150            | 30             |
| <b>k=1,n=4</b> | 180            | 60             |
| <b>k=1,n=5</b> | 210            | 90             |
| <b>k=1,n=6</b> | 240            | 120            |
| <b>k=1,n=7</b> | 270            | 150            |

Fonte: Autor desta dissertação.

Em termos de múltiplos da frequência de acionamento usada, os valores encontrados pela equação, colocados em ordem crescente, são:  $0,5f_1$ ;  $1,0f_1$ ;  $1,5f_1$ ;  $2,0f_1$ ;  $2,5f_1$ ;  $3,0f_1$ ;  $3,5f_1$ ;  $4,0f_1$ ;  $4,5f_1$ . Se a tabela for expandida para valores de  $n$  maiores que 7, os múltiplos continuarão a crescer a cada  $0,5f_1$ .

Considerando-se que o limite de banda de frequência, devido ao filtro de 1 kHz, é de 500 Hz, então o maior múltiplo efetivo nesta progressão é de  $8f_1$ . No entanto, das experimentações de Thomson (2001), conclui-se que nem todas estas frequências são assertivas para o problema do curto-circuito entre espiras, pois também podem ser geradas por outras causas. Destes estudos, foi descoberto que as frequências que são únicas para caracterização da falha de curto-circuito entre espiras, na expressão (C.1), aparecem em  $f_{cc1}$  quando  $k = 1, n = 3$  e  $k = 1, n = 5$ . Neste caso, 150 Hz e 210 Hz, ou em termos de múltiplos da frequência fundamental  $2,5f_1$  e  $3,5f_1$ .

Para esta análise, considerou-se escorregamento igual zero, o que não ocorre na prática. O escorregamento nominal, para a máquina utilizada neste trabalho, é de 0,044 ( $s = (1800 - 1720)/1800$ ). Em termos de frequência, este escorregamento equivale a 2,7 Hz. Para cobrir tal variação nas frequências, os valores de amplitude, considerados como possíveis atributos para os classificadores neurais, são obtidos fazendo-se uma varredura de  $\pm 2,7$  Hz em torno do valor obtido, quando o escorregamento é igual a zero, e selecionando-se o valor de maior amplitude. Por isso, os valores de  $0,5f_1; 1,0f_1; 1,5f_1, \dots 8f_1$  são chamados de múltiplos aproximados dos espectros.

## ANEXO A – Placa de filtragem

Na Figura 17, é mostrado um desenho esquemático referente à filtragem do sinal de corrente.

Figura 17. Desenho esquemático da placa de filtragem e amplificação.

