

TESE DE DOUTORADO Nº 005

**TREINAMENTO DE REDES NEURAS ARTIFICIAIS
BASEADO EM SISTEMAS DE ESTRUTURA VARIÁVEL
COM APLICAÇÕES EM ACIONAMENTOS ELÉTRICOS**

Gustavo Guimarães Parma

DATA DA DEFESA: 30.06.2000

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica

**Treinamento de Redes Neurais
Artificiais Baseado em Sistemas de
Estrutura Variável com Aplicações em
Acionamentos Elétricos**

por

Gustavo Guimarães Parma

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção de título de Doutor em Engenharia Elétrica

Orientador: Benjamim Rodrigues de Menezes

Co-Orientador : Antônio de Pádua Braga

30 de junho de 2000

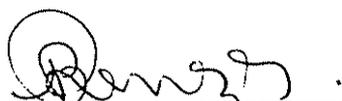
“Treinamento de Redes Neurais Artificiais baseado em Sistemas de Estrutura Variável com Aplicações em Acionamentos Elétricos”

Gustavo Guimarães Parma

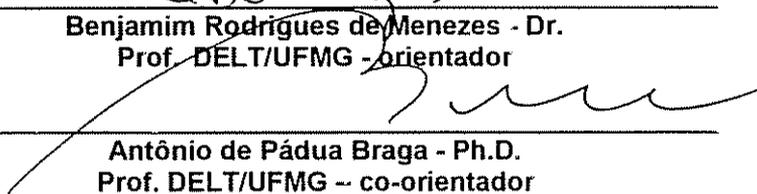
Tese de Doutorado submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 30 de junho de 2000.

Por:



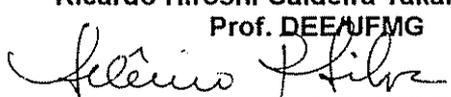
Benjamin Rodrigues de Menezes - Dr.
Prof. DELT/UFMG - orientador



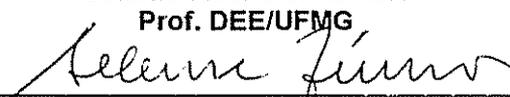
Antônio de Pádua Braga - Ph.D.
Prof. DELT/UFMG - co-orientador



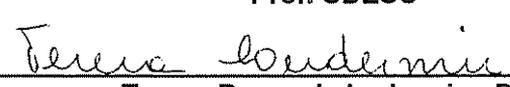
Ricardo Hiroshi Caldeira Takahashi - Dr.
Prof. DEE/UFMG



Selênio Rocha Silva - Dr.
Prof. DEE/UFMG



Seleme Isaac Seleme Jr. - Dr.
Prof. UDESC



Teresa Bernarda Ludermir - Dra.
Profa. CI/UFPe

Agradecimentos

Após uma longa jornada como esta, depois de vários erros cometidos (alguns corrigidos), corre-se o risco de se cometer ao menos mais um, dentre dois grandes erros: o erro da ingratidão, quando não se menciona qualquer nome de pessoas que foram, sem dúvidas, muito importantes neste trabalho ou, então, o erro da falta de memória, quando se menciona alguns nomes, mas, infelizmente, não todos os merecidos. Dentre estes dois erros, cometerei o segundo e já adianto as minhas desculpas àqueles cujos nomes não foram citados mas, sem sombra de dúvida, cujos rostos e contribuições jamais serão esquecidos.

Desta forma, gostaria de agradecer:

À minha família, pais e irmãos, sem a qual eu, sem sombra de dúvida, não apresentaria este texto. Agradeço muito pela paciência e pelos valiosos ensinamentos e, principalmente, aos meus pais, por me ensinarem a gostar tanto de buscar o conhecimento e o aprimoramento, técnico e pessoal, incentivando a minha formação.

Aos meus orientadores, Prof. Benjamim R. Menezes e Prof. Antônio P. Braga, os quais me ensinaram muito mais do que imaginam, não só pelo conhecimento técnico, mas também pelo conhecimento humano.

Aos meus grandes amigos do CPDEE, Cecília, Régis, Marisa, Ernane, Amilton, Rojane, Marco Aurélio, Joaquim, Maria Luisa, Francisco, Aymore e Ana Liddy, dentre outros que, sem dúvida, esqueci de citar. Agradeço por terem auxiliado-me neste trabalho, através de valiosas contribuições técnicas e por terem compartilhado e compreendido etapas difíceis, quando o problema enfrentado não era somente de ordem técnica. Agradeço, de forma especial, à Cecília, pelo companheirismo, paciência, compreensão e carinho durante todo este trabalho.

A todos os professores, funcionários e colegas do CPDEE que contribuíram direta ou indiretamente para a realização deste trabalho. Agradeço a todos os professores por me ensinarem a dignidade e importância da docência.

À FAPEMIG, CAPES e CNPq pelo suporte financeiro.

”Um cuidadoso exame de todas as nossas experiências passadas pode revelar o fato surpreendente de que, tudo que nos aconteceu, foi para nosso bem.”

James Allen

Resumo

Neste trabalho são propostos alguns algoritmos híbridos de treinamento para a topologia *MultiLayer Perceptron*. Os algoritmos são baseados na teoria de controle de Sistemas de Estrutura Variável, mais especificamente Controle por Modos Deslizantes. A aplicação desta teoria de controle possibilita determinar os limites para os parâmetros envolvidos no treinamento, sendo apresentado um estudo matemático detalhado sobre estes limites.

São apresentados resultados comparativos entre os algoritmos propostos e outros encontrados na literatura.

Como aplicação do sistema de treinamento apresentado, é proposto um sistema de acionamento para motor de indução utilizando redes neurais artificiais.

Abstract

Some hybrid algorithms for training MultiLayer Perceptron Artificial Neural networks are proposed in this thesis. These algorithms are developed using a well known control theory called Sliding Mode Control which is based on Variable Structure Systems. This approach makes it possible to mathematically analyze and determine the gains' limits which are needed in the algorithm. In order to verify the proposed algorithms performance in classical artificial neural network problems, they are compared with well known classical algorithms.

Finally, a control system for induction machines, in which artificial neural networks are applied, is proposed.

Sumário

Agradecimentos	i
Epígrafe	iii
Resumo	v
Abstract	vii
Sumário	ix
Lista de Símbolos	1
Lista de Figuras	3
Lista de Tabelas	9
Capítulo 1 - Introdução	13
Capítulo 2 - Redes Neurais Artificiais	17
2.1 - Introdução	17
2.1.1 - Topologia <i>MultiLayer Perceptron</i>	21
2.2 - Algoritmos de Treinamento	22
2.2.1 - Técnicas de Treinamento	24

2.2.1.1 - Treinamento supervisionado	24
2.2.1.2 - Treinamento não-supervisionado	25
2.2.1.3 - Treinamento por reforço	25
2.2.2 - Regra de Treinamento de Widrow-Hoff	25
2.2.3 - Regra Delta	27
2.2.4 - Backpropagation	28
2.2.4.1 - Treinamento em lote	30
2.2.4.2 - Ganho adaptativo	31
2.2.5 - Momentum	32
2.2.6 - RPROP	32
2.2.7 - QUICKPROP	35
2.2.8 - Adaptação da Taxa de Aprendizado pela Mudança de Sinal	39
2.2.9 - SuperSAB	39
2.2.10 - Levenberg-Marquardt	40
2.3 - Conclusões	42
Capítulo 3 - Sistemas de Estrutura Variável	45
3.1 - Introdução	45

3.2 - Sistemas de Segunda Ordem	46
3.3 - Sistemas Amostrados	51
3.4 - Sistemas Híbridos - SEV e RNAs	55
3.5 - Conclusões	56
Capítulo 4 - Algoritmos Propostos	57
4.1 - Treinamento <i>Off-line</i> - 1a. Proposta	62
4.1.1 - Superfície de Deslizamento	62
4.1.2 - Regras de Atualização dos Pesos	63
4.1.3 - Estudo da Convergência	64
4.1.3.1 - Camada de saída	64
4.1.3.2 - Camada escondida	65
4.2 - Treinamento <i>Off-line</i> - 2a. Proposta	67
4.2.1 - Superfície de Deslizamento	68
4.2.2 - Regras de Atualização dos Pesos	69
4.2.3 - Estudo da Convergência	70
4.2.3.1 - Camada de saída	70
4.2.3.2 - Camada escondida	71

4.3 - Treinamento <i>On-line</i> - 1a. proposta	73
4.3.1 - Superfície de Deslizamento	74
4.3.2 - Regras de Atualização dos Pesos	75
4.3.3 - Estudo da Convergência	75
4.3.3.1 - Camada de saída	75
4.3.3.2 - Camada escondida	78
4.4 - Treinamento <i>On-line</i> - 2a. proposta	80
4.4.1 - Regras de Atualização dos Pesos	82
4.4.2 - Estudo da Convergência	83
4.4.2.1 - Camada de saída	83
4.4.2.2 - Camada escondida	84
4.5 - Conclusões	86
Capítulo 5 - Bancada Experimental	89
5.1 - Inversor	89
5.2 - Circuito de Start-up e Chopper	93
5.3 - Circuito de Interface de Comandos	95
5.4 - Circuito de Histerese de Corrente	97

5.5 - Circuito de Aquisição de Corrente e Tensão	100
5.6 - Circuito de Medição Analógica de Correntes e Tensões	100
5.7 - Circuito de Medições Digitais de Tensão e Velocidade	104
5.7.1 - Medição de Velocidade	106
5.7.1.1 - Medição de velocidade por pulso	106
5.7.1.2 - Medição de velocidade por tempo	109
5.7.2 - Medição de Tensão.....	110
5.8 - Circuito de Medição de Fluxo e Conjugado	115
5.8.1 - Medição do fluxo através de bobinas exploradoras	116
5.8.2 - Medição do fluxo através da tensão parcial nas bobinas das fases	117
5.9 - Circuito de Medição do Tacogerador	117
5.10 - Circuito de Interface	118
5.11 - Conjunto Motor de Indução/Motor Corrente Contínua	119
5.11.1 - Torquímetro	121
5.11.2 - Encoder	122
5.11.3 - Medição de fluxo	122
5.12 - Conclusões	122

Capítulo 6 - Resultados de Simulação e Experimentais	125
6.1 - Treinamento de um Único Neurônio	126
6.2 - Aproximação de Funções	128
6.2.1 - Treinamento <i>Off-line</i>	132
6.2.2 - Treinamento <i>On-line</i>	142
6.3 - Classificação de Padrões	145
6.4 - Controle Vetorial Direto	151
6.4.1 - Observador Neural de Fluxo Estatórico	154
6.4.2 - Resultados de Simulação	158
6.4.3 - Controlador Neural de Velocidade	162
6.5 - Conclusões	168
Capítulo 7 - Conclusões e Propostas de Continuidade	169
7.1 - Propostas de Continuidade	172
Referência Bibliográfica	175
Apêndice A - Equacionamento Matemático	181
Apêndice B - Diagramas Esquemáticos da Bancada Experimental	209
Apêndice C - Medição de Fluxo - Bobinas Exploradoras	225

Apêndice D - Medição de Fluxo - Soma das Tensões Parciais	231
Apêndice E - SemiDriver SKHI23	241

Lista de Símbolos

RNA	Rede Neural Artificial
MLP	<i>MultiLayer Perceptron</i>
$T_{h()}$	Entrada h da RNA
$R_{i()}$	Saída linear do neurônio i da camada escondida
$Y_{Hi()}$	Saída do neurônio i da camada escondida, após a aplicação da função de ativação
$V_{j()}$	Saída linear do neurônio j da camada de saída
$Y_{j()}$	Saída do neurônio j da camada de saída, após a aplicação da função de ativação
$Y_{dj()}$	Saída desejada do neurônio j da camada de saída
$f_H(\circ)$	Função de ativação da camada escondida
$f(\circ)$	Função de ativação da camada de saída
$f'_H(\circ)$	derivada da função de ativação da camada escondida em função da saída linear
$f'(\circ)$	derivada da função de ativação da camada de saída em função da saída linear
W	Matriz de pesos da camada de saída
Z	Matriz de pesos da camada escondida
W_{ji}	Peso que conecta a saída i da camada escondida ao neurônio j da camada de saída
Z_{ih}	Peso que conecta a entrada h ao neurônio i da camada escondida
h	Índice utilizado para identificar um entrada da RNA ($h \leq n + 1$)
i	Índice utilizado para identificar um neurônio da camada escondida ($i \leq m + 1$)
j	Índice utilizado para identificar um neurônio da camada de saída ($j \leq p$)
n	Número de entradas da RNA
m	Número de nodos na camada escondida da RNA
p	Número de nodos na camada de saída da RNA
$E_{(k)}$	Somatória do erro médio quadrático da camada de saída no instante k
∇	Função gradiente
$\delta_{.(k)}$	Gradiente local do neurônio (\cdot)
$e_{j(k)}$	erro existente entre a saída desejada da RNA e a saída real para o instante k
∇	Função Gradiente

\vec{v}_s	Vetor tensão de estator
\vec{i}_s	Vetor corrente de estator
$\vec{\lambda}_s$	Vetor fluxo de estator
R_s	Resistência estatórica
L_s	Indutância própria de estator
\vec{i}_r	Vetor corrente de rotor
$\vec{\lambda}_r$	Vetor fluxo de rotor
R_r	Resistência rotórica
L_r	Indutância própria de rotor
L_m	Indutância mútua entre rotor e estator
ω_r	Velocidade angular do rotor (<i>rad/s</i>)
T_e	Conjugado eletromagnético ($N \cdot m$)
T_l	Conjugado de carga ($N \cdot m$)
P	Número de pólos da máquina
J	Momento de inércia

Lista de Figuras

2.1	Nodo MCP. Modelo do neurônio artificial idealizado por McCulloch e Pitts.	18
2.2	Rede <i>MultiLayer Perceptron</i> : Composta de nodos perceptron dispostos em camadas.	22
2.3	Métodos de treinamento baseados no gradiente descendente.	24
2.4	Gráficos das derivadas das funções de ativação. (a) derivada da sigmoide; (b) derivada da tangente hiperbólica.	28
2.5	Função de erro considerada no algoritmo QProp.	35
2.6	Derivada da função de erro considerada no algoritmo QProp.	36
3.1	Superfície de deslizamento utilizada nos sistemas de CMD.	47
3.2	Sistema de controle para um robô, acionado por um motor de indução, utilizando RNAs e SEV.	55
4.1	Diagrama esquemático utilizado para exemplificar como o treinamento de uma RNA pode ser analisado como um problema de controle.	58
5.1	Visão geral do inversor desenvolvido.	89
5.2	Módulo de potência utilizado pelo inversor trifásico.	90
5.3	Placas de medição desenvolvidas.	91
5.4	Circuito lógico do comparador com janela utilizado na histerese de corrente.	98
5.5	Oscilador utilizado para a limitação de frequência da histerese de corrente.	98
5.6	Diagrama esquemático do sistema implementado na EPLD do circuito de	

	medição digital, responsável pela medição de velocidade e multiplexação dos sinais de saída.	107
5.7	Diagrama de tempo do decodificador de quadratura. (a) máquina girando no sentido horário e (b) máquina girando no sentido anti-horário.	108
5.8	Diagrama implementado em uma das EPLDs responsáveis pela medição de tensão (CI Tensão CI-a).	110
5.9	Diagrama implementado em uma das EPLDs responsáveis pela medição de tensão (CI Tensão CI-b).	111
5.10	Diagrama esquemático do circuito necessário para a medição das tensões sobre cada chave que serão posteriormente utilizadas pela EPLD.	113
5.11	Sinais de comando provenientes do sistema de medição. $ch3$ e $\overline{ch3}$ representam os estados das chaves após a inclusão do tempo morto proveniente das placas dos drivers de comando das chaves de potência.	113
5.12	Detalhe da instalação do torquímetro rotacional acoplado entre a máquina de indução e a máquina de corrente contínua.	121
5.13	Detalhe da máquina de indução utilizada na bancada experimental. O conector disponibiliza os sinais de tensão necessários à medição de fluxo.	122
6.1	Superfície de erro para um neurônio treinado para reconhecer os pares $\{-3.0, 0.4\}$ e $\{2.0, 0.8\}$	127
6.2	Evolução dos pesos em cada uma das simulações para o treinamento de um único neurônio.	129
6.3	Evolução do erro em função do número de <i>epochs</i> para a simulação 1.	130
6.4	Evolução do erro em função do número de <i>epochs</i> para a simulação 2.	130
6.5	Evolução do erro em função do número de <i>epochs</i> para a simulação 3.	131
6.6	Evolução do erro em função do número de <i>epochs</i> para a simulação 4.	131
6.7	Segunda função utilizada na comparação dos algoritmos de treinamento <i>off-line</i> . $f_1(x) = -3.2 \cdot x^2 + 3.2 \cdot x$	133

6.8	Função utilizada para a comparação dos algoritmos de treinamento <i>off-line</i> . $f_2(x) = x^4$	134
6.9	Evolução do erro durante o treinamento utilizando a função $f_1(x) = -3.2 \cdot x^2 + 3.2 \cdot x$	139
6.10	Evolução do erro durante o treinamento utilizando a função $f_2(x) = x^4$	140
6.11	Treinamento da RNA utilizando a primeira proposta para treinamento <i>on-line</i> e o algoritmo <i>Backpropagation</i> . Em t=0.45s, os algoritmos de treinamento foram desabilitados.	143
6.12	Curva de erro referente ao treinamento da RNA utilizando a primeira proposta para treinamento <i>on-line</i> e o algoritmo <i>Backpropagation</i> . Em t=0.45s, os algoritmos de treinamento foram desabilitados.	143
6.13	Treinamento da MLP, para a segunda função, utilizando a primeira proposta para treinamento <i>on-line</i>	144
6.14	Conjunto de treinamento utilizado no problema de classificação.	146
6.15	Erro durante o treinamento para o problema de classificação de padrões.....	148
6.16	Ampliação da figura anterior para uma melhor visualização da convergência dos algoritmos utilizados.	149
6.17	Linha de decisão encontrada pelos algoritmos de treinamento utilizados no problema de classificação.	150
6.18	Sistema de controle a ser implementado.	154
6.19	Diagrama esquemático do observador neural de fluxo estatórico implementado.	155
6.20	Fluxo real e observado para uma frequência de alimentação de 60Hz.....	156
6.21	Fluxo real e observado para uma frequência de alimentação de 10Hz.....	156
6.22	Fluxo real e observado para uma frequência de alimentação de 60Hz. Variação de 20%na resistência estatórica em t=0.25s.	157

6.23	Fluxo real e observado para uma frequência de alimentação de 10Hz. Variação de 20% na resistência estatórica em $t=0.25s$	157
6.24	Topologia MLP utilizadas como: (a) controlador de fluxo e (b) controlador de velocidade.	159
6.25	Velocidade de referência ($\pm 100 \text{ radele/s}$) e medida. Partida e reversão de velocidade. Primeira partida, pesos da RNA amostrados de uma distribuição normal de média nula.	160
6.26	Fluxo de estator de eixo direto de referência e fluxo simulado. Partida e reversão de velocidade. Primeira partida da máquina, pesos da RNA amostrados de uma distribuição normal de média nula.	161
6.27	Corrente de estator de eixo direto e eixo em quadratura. Partida e reversão de velocidade. Primeira partida da máquina, pesos da RNA amostrados de uma distribuição normal de média nula.	161
6.28	Velocidade de referência ($\pm 100 \text{ radele/s}$) e medida. Partida e reversão de velocidade. Quinta partida, pesos da RNA provenientes das partidas anteriores.	163
6.29	Fluxo de estator de eixo direto de referência e fluxo simulado. Partida e reversão de velocidade. Quinta partida da máquina, pesos da RNA oriundos das partidas anteriores.	163
6.30	Corrente de estator de eixo direto e eixo em quadratura. Partida e reversão de velocidade. Quinta partida da máquina, pesos da RNA oriundos das partidas anteriores.	164
6.31	Diagrama esquemático do controle vetorial direto do motor de indução utilizando sistema de eixos orientados pelo fluxo de rotor.	165
6.32	MLP utilizada como controlador de velocidade no controle vetorial direto.	166
6.33	Curvas de resposta do controle implementado. Observa-se a curva de velocidade real e de referência (W_r e W_{ref} , respectivamente), conjugado observado e de referência (T_{eobs} e T_{eref} , respectivamente) e corrente de eixo direto (I_{sdo} , responsável pelo fluxo).	166
6.34	Ampliação da curva de velocidade da Figura anterior. Observa-se o pequeno sobre-sinal existente.	167

6.35	Gráfico do conjugado em função da velocidade do motor.	167
B.1	Diagrama de ligações do barramento DC utilizado pela bancada experimental desenvolvida.	210
B.2	Circuito elétrico da placa de interface.	211
B.3	Circuito elétrico dos buffers utilizados pela placa de interface.	212
B.4	Circuito da placa de medições digitais.	213
B.5	Circuito de histerese desenvolvido.	214
B.6	Circuito eletrônico para a medição de fluxo utilizando bobinas pesquisadoras. .	215
B.7	Circuito para a medição de fluxo utilizando tensões parciais.	216
B.8	Circuito para a medição do torque.	217
B.9	Circuito para a medição analógica das tensões e correntes.	218
B.10	Circuito para a medição da tensão aplicada à máquina, utilizando a integração no período de PWM.	219
B.11	Circuito para a transformação das grandezas trifásicas em grandezas do sistema de eixos $\alpha\beta$ fixos no estator.	220
B.12	Circuito para a interface entre as placas de comando dos IGBTs e o inversor. .	221
B.13	Circuito para a aquisição das correntes e tensões na máquina.	222
B.14	Circuito para o controle da tensão do barramento DC.	223
B.15	Circuito para a medição da velocidade utilizando o tacogerador.	224
C.1	Bobinas Pesquisadora: Configuração adotada. Eixo magnético centrado em um dente.	226
C.2	Bobinas pesquisadoras: Distribuição espacial dos eixos $\alpha\beta$ no estator da	

	máquina utilizada. Os números indicados se referem às ranhuras do estator. . .	228
C.3	Implementação prática das bobinas pesquisadoras para o eixo α e β . A numeração refere-se ao número de ranhuras de estator.	230
D.1	Distribuição espacial dos enrolamentos de uma fase do estator.	238
D.2	Diagrama fasorial das tensões induzidas nas bobinas n_1 e n_2	238
D.3	Diagrama fasorial para a diferença das tensões parciais nas bobinas de cada uma das fases.	239
D.4	Sistema de eixos utilizado para a mudança de variáveis.	239
E.1	Disposição dos jumpers e conectores do driver de comando SKHI23.	243
E.2	Diagrama esquemático do semidriver de comando.	245
E.3	Forma de onda de V_{CE} e, função dos parâmetros R_{CE} e C_{CE}	250

Lista de Tabelas

5.1	Sinais utilizados pela placa de Startup e Chopper disponíveis no conector DB9 .	94
5.2	Sinais de comando para os gates dos IGBTs. Conector P24.....	96
5.3	Sinais de controle utilizados pelo circuito. Conector P23.....	96
5.4	Sinais de comando para os gates dos IGBTs. Conector P10.....	99
5.5	Sinais de entrada para o circuito de histerese. Conector P9.	99
5.6	Pinagem do conector P19 que disponibiliza os sinais das medições das correntes e tensões de fase.	101
5.7	Pinagem do conector P20 que disponibiliza os sinais das medições necessárias ao circuito de medição digital.	101
5.8	Pinagem do conector P22 responsável por fornecer os valores de tensão e corrente provenientes do circuito de aquisição de corrente e tensão.....	103
5.9	Pinagem do conector P21 responsável disponibilizar para a placa de interface as grandezas analógicas medidas.	103
5.10	Pinagem do conector P4 responsável por conectar a placa de medição digital ao encoder incremental.	104
5.11	Pinagem do conector P5 responsável por conectar a placa de medição digital à placa de aquisição de corrente e tensão.	105
5.12	Pinagem do conector P3 utilizado para conectar a placa de medição digital placa de interface.	105
5.13	Sinais de controle da placa de medição digital e a respectiva grandeza disponibilizada na saída do circuito.	106

5.14	Tabela Verdade dos estados possíveis assumidos pelas chaves $ch3$ e $/ch3$ e consequente saída dos amplificadores operacionais	112
5.15	Pinagem do conector P7 responsável por conectar a placa de medição de torque e fluxo ao torquímetro.	115
5.16	Pinagem do conector P6 responsável por conectar a placa de medição de torque e fluxo a máquina de indução.	116
5.17	Pinagem do conector P8 responsável por conectar a placa de medição de torque e fluxo a placa de interface.	117
5.18	Pinagem do conector P2 responsável pela alimentação do circuito do tacogerador e por disonibilizar o sinal de corrente de saída do circuito.	118
5.19	Pinagem do conector P14 utilizado para conectar a placa de aquisição de dados à placa de interface.	119
5.20	Parâmetros da máquina de indução utilizada	120
5.21	Parâmetros da máquina de corrente contínua utilizada	120
6.1	Condições iniciais para o par $(bias, peso)$ utilizado no treinamento de um único neurônio.	127
6.2	Ganhos dos treinamentos utilizados por cada um dos algoritmos utilizados.	128
6.3	Número de $epochs$ necessárias ao treinamento de um único neurônio. Limite de erro estipulado em 10^{-4}	129
6.4	Parâmetros dos algoritmos de treinamentos utilizados para a função f_1	136
6.5	Parâmetros dos algoritmos de treinamentos utilizados para a função f_2	136
6.6	Número de $epochs$ necessárias ao treinamento da RNA para as função apresentadas. Limite de erro igual a $2.5 \cdot 10^{-3}$ e número máximo de $epochs$ igual a 5000.	137
6.7	Tempo de processamento (em segundos) necessário a cada algoritmo para o treinamento da RNA, considerando cada uma das funções e os limites impostos.	138

6.8	Numeração utilizada nas figuras para a identificação das curvas.	142
6.9	Parâmetros do treinamento para cada um dos algoritmos utilizados no problema de classificação	147
6.10	Classificações erradas para o conjunto de treinamento.	148
6.11	Erro de classificação para o conjunto de validação.	148
6.12	Identificação das curvas mostradas do observador neural de fluxo estatístico. ...	156
E.1	Valores típicos do driver de comando.	242
E.2	Conector de entrada - 14 pinos, modelo DIN 41651	244
E.3	Conector de saída dos sinais de comando para o IGBT, modelo MOLEX 41791.	246
E.4	Jumpers utilizados na placa do driver de comando.	247
E.5	Componentes de ajuste utilizados na placa do driver de comando.	248
E.6	Valores típicos para o disparo ou corte do IGBT.	249
E.7	Tabela Verdade para o sinal de erro em nível alto.	249
E.8	Variação do tempo morto em função do resistor R_{TD}	251
E.9	Valores característicos dos componentes para IGBT 1200V	252
E.10	Valores característicos dos componentes para IGBT 1700V	253

Capítulo 1

Introdução

Redes Neurais Artificiais (RNAs) são ferramentas poderosas que podem ser utilizadas em vários tipos de problemas como, por exemplo: problemas de classificação, de aproximação de funções, de interpolação, ou, ainda, de predição, dentre outras aplicações. Devido à sua ampla gama de utilização, RNAs possibilitam uma interdisciplinariedade acentuada, na qual utiliza-se a RNA como uma ferramenta para solucionar um problema específico de uma determinada área ou disciplina. Portanto, encontram-se exemplos de aplicação de RNAs em áreas diversas e distintas como, por exemplo, economia, saúde, agricultura ou engenharia, entre outras.

A utilização de RNAs, principalmente no que concerne à topologia (maneira pela qual os neurônios estão conectados) *MultiLayer perceptron* (MLP), enfrenta, principalmente, duas dificuldades. Inicialmente, a escolha da dimensão da rede ainda é feita de forma empírica, existindo alguns algoritmos que efetuam modificações na estrutura da rede no decorrer do treinamento. Em segundo lugar, o tempo gasto durante o treinamento, normalmente realizado *off-line*, é uma função do algoritmo e dos parâmetros de treinamento escolhidos, não existindo, na grande maioria das vezes, um estudo matemático sobre a convergência do algoritmo ou de seus parâmetros. Além disso, a implementação de um determinado algoritmo de treinamento não é, muitas vezes, trivial, sendo necessárias modificações em função do problema em questão.

Neste trabalho são propostos alguns algoritmos híbridos de treinamento. É utilizada uma teoria de controle, conhecida por Sistemas de Estrutura Variável (SEV) [Itkis, 1976] e, mais especificadamente, Controle por Modos Deslizantes (CMD)[Utkin, 1978], para o treinamento de MLPs. Sistemas de CMD são equivalentes a sistemas de alto ganho e que apresentam robustez frente a variações paramétricas. Optou-se por utilizar esta técnica de controle, além de ser uma teoria bem consolidada, para possibilitar o controle

do ajuste dos parâmetros da RNA e possibilitar um estudo analítico dos ganhos envolvidos no treinamento.

Encontram-se, na literatura técnica, trabalhos envolvendo RNAs e sistemas de controle. Nos trabalhos encontrados, a RNA participa do controle, sendo responsável direta ou indireta por alguma etapa do controle como, por exemplo, a observação de alguma variável necessária. Não foi encontrada nenhuma aplicação direta de sistemas de controle no treinamento da RNA, bem como não se encontrou nenhum algoritmo genérico de treinamento utilizando a metodologia de SEV. Objetiva-se, portanto, desenvolver algoritmos de treinamento que possuam as características de robustez e velocidade de convergência, inerentes a sistemas de CMD.

Com a metodologia adotada, o problema de treinamento de MLP é tratado e solucionado como um problema de controle. Desta forma, a teoria de CMD é aplicada diretamente no treinamento da MLP. A RNA é, portanto, tratada como um sistema a ser controlado e não como uma parte do sistema de controle. A metodologia possibilita, ainda, um estudo matemático aprofundado sobre a convergência e os limites dos parâmetros envolvidos no treinamento. Como mencionado anteriormente, para os algoritmos de treinamento normalmente utilizados, não existe um estudo matemático sobre os limites nos quais os parâmetros envolvidos no treinamento devem permanecer.

São propostos algoritmos de treinamento *off-line* e *on-line*. Os algoritmos de treinamento *off-line* são comparados com algoritmos bem divulgados e conhecidos na área de RNAs. Os algoritmos para treinamento *on-line* são utilizados no desenvolvimento de um sistema de acionamento para um motor de indução. Objetiva-se, portanto, desenvolver um sistema de controle que seja o mais independente possível dos parâmetros do motor, sendo capaz de se adaptar à máquina que se deseja controlar.

Pode-se listar os seguintes pontos principais de contribuição deste trabalho:

1. Desenvolvimento e análise matemática de dois algoritmos de treinamento *off-line*, sendo o principal deles a segunda proposta de treinamento (seção 4.2);
 2. Desenvolvimento e análise matemática de dois algoritmos de treinamento *on-line*,
-

com enfoque à primeira proposta de treinamento (seção 4.3);

3. Desenvolvimento de uma bancada experimental com recursos de *hardware* capazes de possibilitar a implementação e comparação de diversas técnicas de acionamento de máquinas de indução;
4. Apresentação de uma técnica de acionamento elétrico utilizando redes neurais artificiais.

Como consequência do trabalho desenvolvido, foram publicados dois artigos em revistas internacionais ([Parma et al., 1998b, Parma et al., 1999b]), um artigo em congresso internacional ([Parma et al., 1999c]) e seis artigos em congressos nacionais ([Parma et al., 1997a, Parma et al., 1998a, Parma et al., 1998c, Parma et al., 1999a, Parma et al., 1999c, Neves et al., 1999]).

Este trabalho é dividido em oito partes, a saber:

1. Primeiro capítulo: É feita uma abordagem genérica dos objetivos deste trabalho, mostrando a relevância e originalidade do assunto tratado;
 2. Segundo capítulo: Faz-se uma abordagem dos conceitos básicos de RNAs. São mostradas as principais topologias de RNAs. É feito um estudo dos principais algoritmos de treinamento para MLP.
 3. Terceiro capítulo: É feito um estudo introdutório aos conceitos de SEV e CMD. A metodologia a ser aplicada nos algoritmos de treinamento é mostrada através de um exemplo abordando sistemas contínuos e sistemas discretos.
 4. Quarto capítulo: Os algoritmos de treinamento são propostos e analisados matematicamente. São propostos dois algoritmos para treinamento *off-line* e dois para treinamento *on-line*.
 5. Quinto capítulo: São mostrados os detalhes da bancada experimental na qual será implementado o sistema de controle do motor de indução utilizando MLPs.
 6. Sexto capítulo: Utilizam-se RNAs em problemas de aproximação de funções e os resultados da comparação entre os algoritmos de treinamento *off-line* propostos e existentes são mostrados. São mostrados resultados da aplicação do trei-
-

namento *on-line* proposto em problemas de aproximação de funções temporais. Apresentam-se resultados de simulação e experimentais do sistema de controle proposto. São apresentadas as conclusões e propostas de continuidade do trabalho.

7. Sétimo capítulo: As principais conclusões deste trabalho são apresentadas neste capítulo, bem como as propostas para trabalhos futuros.
 8. Apêndices: Para propiciar um melhor entendimento dos assuntos abordados neste trabalho e facilitar a leitura deste texto, optou-se por colocar, em separado do texto principal, alguns itens relevantes aos tópicos abordados nesta tese.
-

Capítulo 2

Redes Neurais Artificiais

Neste capítulo será apresentada uma breve introdução ao histórico de redes Neurais Artificiais (RNAs) e serão abordados os principais algoritmos para o treinamento da topologia *MultiLayer Perceptron* (MLP).

2.1 Introdução

A origem do estudo de RNAs advém da tentativa de modelagem do neurônio biológico e suas interconexões. Entretanto, se o estudo do neurônio biológico e do cortex cerebral humano tem avançado constantemente, o estudo de redes neurais artificiais não possui mais o seu vínculo original com o neurônio biológico e, agora, uma RNA é vista como uma ferramenta de trabalho que pode ser aplicada em diversas situações, independentemente de sua origem biológica.

O estudo de RNAs iniciou o seu desenvolvimento com o modelo matemático do neurônio biológico proposto por McCulloch e Pitts, em 1943 [McCulloch & Pitts, 1943]. O modelo idealizado pelos pesquisadores implementa, basicamente, uma função de limiar (Equação 2.1) aplicada a um somatório ponderado pelos pesos (Equação 2.2). A ativação do neurônio ocorre quando o somatório das entradas, ponderado pelos pesos, ultrapassa o valor de limiar (θ). Desta forma, implementou-se matematicamente o modelo simplificado das sinapses (através da ponderação das entradas pelos pesos W_n) e do potencial de ação do neurônio (através da função de limiar).

O neurônio de McCulloch e Pitts, ou nodo MCP como é conhecido, é mostrado na Figura 2.1.

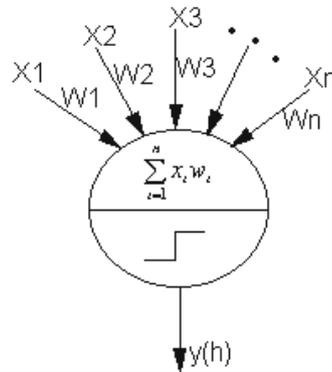


Figura 2.1- Nodo MCP. Modelo do neurônio artificial idealizado por McCulloch e Pitts.

$$y(h) = \begin{cases} 1 & \text{se } h \geq \theta \\ 0 & \text{se } h < \theta \end{cases} \quad (2.1)$$

$$h = \sum_{l=1}^n X_l \cdot W_l \quad (2.2)$$

Devido à função de limiar, utilizada principalmente quando o neurônio executa funções Booleanas, este modelo é, muitas vezes, conhecido como portas de limiar (*Threshold gates*) [H.Hassoun, 1995].

Analisando as Equações 2.1 e 2.2, pode-se escrever que o neurônio será ativado quando:

$$\sum_{l=1}^n X_l \cdot W_l \geq \theta; \text{ ou,}$$

$$\sum_{l=1}^n [X_l \cdot W_l] - \theta \geq 0;$$

Observa-se que a inequação acima é a expressão de um hiperplano no espaço definido pelas variáveis X_l e θ (termo de polarização ou *bias*). No caso de duas variáveis

(X_1 e X_2) pode-se escrever:

$$X_1 \cdot W_1 + X_2 \cdot W_2 - \theta \geq 0, \text{ ou,}$$

$$X_1 \geq -\frac{W_2}{W_1} \cdot X_2 + \frac{\theta}{W_1}.$$

Ou seja, a expressão de um semiplano de X_1 em função de X_2 , cuja inclinação é dada por $-\frac{W_2}{W_1}$ e *off-set* de $\frac{\theta}{W_1}$.

Conclui-se, portanto, que um único nodo MCP é capaz de discriminar somente classes que sejam linearmente separáveis, não podendo ser aplicado a uma gama mais ampla de problemas. Em casos simples, como o “ou-exclusivo”, não é possível encontrar um conjunto de parâmetros (pesos W) tais que o nodo implemente a função desejada. Este fato foi explorado no livro publicado por Minsky e Papert em 1969 [Minsky & Papert, 1969], mostrando as deficiências e limitações do nodo MCP.

Anteriormente à publicação de Minsky e Papert, em 1949 Donald Hebb [Hebb, 1949] descreveu a primeira teoria neuro-fisiológica para a modificação de sinapses em neurônios animais. Esta teoria é muito utilizada em algoritmos de treinamento de RNAs, dando origem à chamada **Regra de Hebb**.

Bernard Widrow e Marcian Hoff [Widrow & Hoff, 1960, Widrow, 1962] apresentaram uma regra de treinamento para o seu neurônio, chamado ADALINE (ADActive LINEar combiner), na qual o ajuste dos pesos é uma função do erro entre o valor desejado para a saída e o valor atual fornecido pelo elemento. Essa regra ficou conhecida como “Regra de Treinamento de *Widrow-Hoff*”, a qual evoluiu para a “Regra Delta”, e, posteriormente, para o “*Backpropagation*” [Rumelhart et al., 1986].

Em 1958 [Rosenblatt, 1958], Rosenblatt explora uma topologia de RNA, a qual ele denomina *perceptron*, composta de duas camadas. Em seu perceptron, os pesos da primeira camada são escolhidos aleatoriamente e mantidos fixos durante todo o treinamento; os pesos da segunda camada são, então, ajustados de forma a possibilitar o reconhecimento de imagens. Com seu trabalho Rosenblatt instigou vários pesquisadores

a desenvolverem diversos trabalhos pois ficara provado a potencialidade de RNAs em problemas complexos, como é o caso de reconhecimento de imagens. Este grande número de novos trabalhos desenvolvidos com RNAs perdurou até a publicação do trabalho de Minsky e Papert, em 1969, desacreditando a potencialidade das RNAs.

Hopfield, em 1982 [Hopfield, 1982], introduziu a arquitetura recorrente, originalmente idealizada para a utilização como memórias associativas. Em seu trabalho ele descreve uma função de energia para a rede, demonstrando que esta função é monotonicamente decrescente e fazendo uma associação com fenômenos físicos. Utilizando teorias da Física, diversos pesquisadores da Física foram atraídos para a área de RNAs, propiciando um novo desenvolvimento da área.

Em trabalho publicado em 1989 [Cybenko, 1989], Cybenko prova, baseado no teorema de Hahn-Banach [Luenberger, 1969], a universalidade na aproximação de funções contínuas por RNAs *feedforward* com uma camada escondida, utilizando função de ativação sigmoidal, e uma camada de saída, utilizando a saída linear do neurônio.

No estudo realizado por Bishop, em 1995 [Bishop, 1995], foi explorada a relação apenas linear do número de parâmetros a serem ajustados na RNA em função do número de variáveis da função que se deseja aproximar. Sabe-se que em, por exemplo, uma aproximação polinomial, o número de parâmetros (ou coeficientes do polinômio) a serem ajustados é uma relação exponencial com o número de variáveis da função. Os trabalhos de Bishop e Cybenko abriram um novo campo para a aplicação de RNAs, sendo, agora, utilizadas como aproximadores de funções, das quais, muitas vezes, se conhece apenas alguns vetores de dados, mas não uma expressão analítica. Nestes casos a RNA pode ser utilizada, durante o treinamento, para extrair dos dados a função e, após treinada, para interpolar ou extrapolar o conjunto de treinamento.

Em 1974, Paul Werbos [Werbos, 1974], em seu trabalho "*Backpropagation through time*", explora um algoritmo de treinamento para redes recorrentes utilizando o método do gradiente descendente. Rumelhart, Hinton e Willians [Rumelhart et al., 1986], em 1986, apresentam o algoritmo *Backpropagation*, possibilitando o treinamento de redes *feedforward* com várias camadas, comumente chamadas de *MultiLayer Perceptron*

(MLP).

Na atualidade, o nodo MCP é utilizado com um dos três seguintes tipos de funções de ativação:

1. Função de limiar: como originalmente proposto, Equação 2.1;
2. Função linear: sendo, desta forma, idêntico ao neurônio utilizado no ADALINE, no qual a Equação 2.1 é alterada para: $y(h) = h$;
3. Função não linear: utiliza-se, normalmente, uma das duas funções abaixo descritas em lugar da Equação 2.1:
 - (a) Sigmoides: $y(h) = \frac{1}{1 + \exp(-h)}$;
 - (b) Tangente hiperbólica: $y(h) = \tanh(h)$.

2.1.1 Topologia *MultiLayer Perceptron*

Entende-se por topologias as diversas maneiras pela qual os neurônios podem ser conectados e, por algoritmos ou regras de treinamento a metodologia lógica pela qual os parâmetros da RNA são ajustados. Apenas a topologia RNA *MultiLayer Perceptron* (MLP) será abordada nesta seção pois as demais topologias de redes, também importantes na área de RNAs, estão fora do escopo deste trabalho.

A MLP é formada pela combinação de nodos MCP, dispostos em camadas conectadas através dos pesos dos neurônios (Figura 2.2). Uma MLP que possua uma camada escondida e uma camada de saída é dita de duas camadas, apesar de existir algumas controvérsias na literatura. Portanto, uma RNA MLP com três camadas possui, além das entradas, duas camadas escondidas e uma camada de saída. O sentido de propagação do sinal durante a utilização da rede é somente das entradas para a saída, caracterizando uma RNA *feedforward*. Durante o processo de treinamento, pode existir uma etapa de *Backpropagation*, na qual o sinal de erro é propagado das saídas em direção às entradas. As RNAs MLP, com mais de uma camada e que utilizem função de ativação não linear, ao menos nos nodos da camada escondida, são ditas aproximadores universais

de funções contínuas [Cybenko, 1989].

Nas seções subseqüentes são apresentados os principais algoritmos de treinamento encontrados na literatura para a topologia MLP.

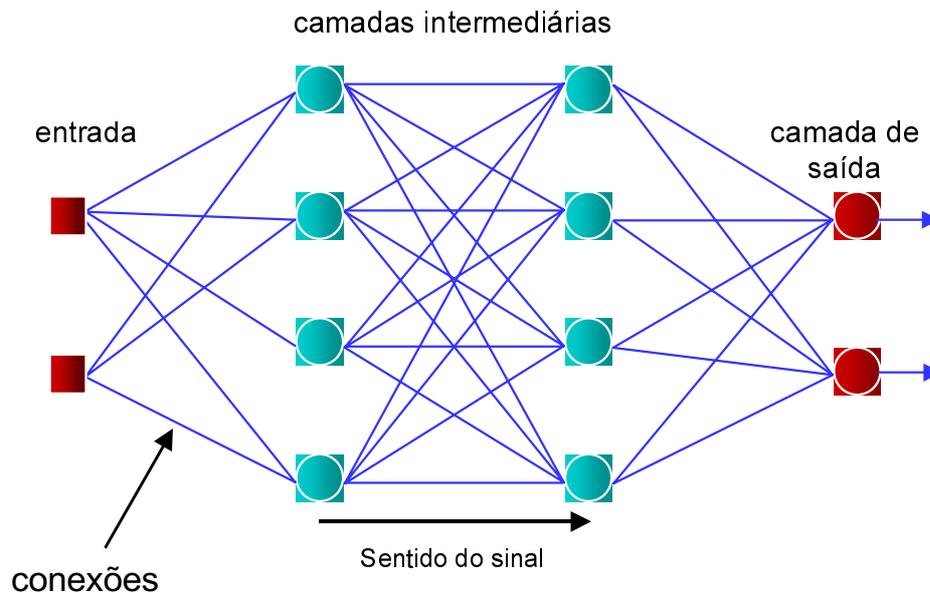


Figura 2.2- Rede *MultiLayer Perceptron*: Composta de nodos perceptron dispostos em camadas.

2.2 Algoritmos de Treinamento

Uma RNA pode possuir várias entradas, caracterizando um vetor de entradas e, da mesma forma, várias saídas, correlatas às entradas, caracterizando um vetor de saída. A união de todos os vetores de dados aplicados às entradas com os respectivos vetores de saídas, se estes forem conhecidos, forma o conjunto de treinamento, ou conjunto de dados. Durante o treinamento, a apresentação de todo o conjunto de dados recebe o nome de “*epoch*” e a contabilização do tempo gasto durante um treinamento pode ser feito em números de *epochs*.

Entende-se por **treinamento** o ajuste dos parâmetros da RNA (ajustes na função de ativação ou, mais usualmente, ajuste dos pesos), de tal forma que a RNA atinja determinados critérios desejados de avaliação da sua performance.

Normalmente são utilizados dois limites para finalizar o treinamento *off-line* de uma RNA:

1. Número máximo de *epochs*;
2. Limite de erro: Estipula-se um limiar para o erro existente entre o vetor de saída da RNA e o vetor de saída desejada (treinamento supervisionado), para toda a *epoch*.

Caracteriza-se por treinamento *off-line* aquele realizado fora da operação normal da RNA e no qual tem-se o conhecimento de todo o conjunto de treinamento ou da resposta esperada da RNA. O treinamento *on-line* é realizado durante a operação normal da RNA e não se conhece, a priori, todo o conjunto de treinamento.

O treinamento pode ser realizado ainda de duas maneiras básicas:

1. Treinamento em lote: a atualização dos pesos é feita apenas uma vez por *epoch*, ocorrendo após a apresentação de todo o conjunto de treinamento;
2. Treinamento por vetor: faz-se a atualização dos pesos a cada vetor entrada-saída apresentado, ou seja, em uma *epoch* serão realizados tantas atualizações nos pesos quanto for o número de vetores do conjunto de treinamento.

Os algoritmos de treinamento são definidos em função da topologia da RNA utilizada. Neste trabalho serão apresentados alguns dos algoritmos mais difundidos para o treinamento de MLP, os quais podem ser divididos em duas classes principais [H.Hassoun, 1995]:

1. Algoritmos baseados no método do gradiente descendente: a atualização dos pesos é feita em direção contrária ao gradiente do erro em função dos pesos. Ou seja, sendo $\mathbf{W}_{(k)}$ a matriz de pesos no instante k ,

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} - \eta \cdot \nabla E_{(k)} \quad (2.3)$$

onde $\eta > 0$ é a taxa de aprendizagem.

Esta classe pode ser, por sua vez, dividida em duas outras categorias, as quais estão exemplificadas na Figura 2.3:

- (a) Métodos de busca local: minimizam o gradiente instantâneo, podendo ficar presos em mínimos locais, Figura 2.3;
 - (b) Métodos de busca global: minimizam a função de erro total, buscando sempre o mínimo global. Geralmente são mais complexos;
2. Algoritmos de correção de erro: inicialmente propostos para o treinamento de apenas um neurônio.

Como pode ser observado na Figura 2.3, a qual mostra a curva de erro para um neurônio com apenas um peso, os métodos de busca local do gradiente podem ficar presos em mínimos locais, enquanto que os métodos globais tendem a convergir, sempre, para o mínimo global da função.

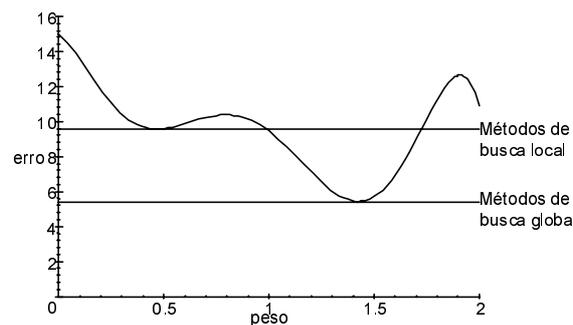


Figura 2.3- Métodos de treinamento baseados no gradiente descendente.

2.2.1 Técnicas de Treinamento

Os algoritmos de treinamento podem ser divididos em, basicamente, três técnicas de treinamento. Estas técnicas se referem à maneira com a qual o treinamento é conduzido.

2.2.1.1 Treinamento supervisionado

Caracterizado pelo conhecimento prévio de todo o conjunto de entrada e saída desejada. Ou seja, sabe-se, a priori, qual deve ser a saída da RNA para um determinado vetor de

entrada. O ajuste dos pesos é, geralmente, realizado em função do erro existente entre a saída desejada (ou esperada) e a saída real da RNA.

São exemplos de treinamento supervisionado: Regra Delta [Zurada, 1992], *Back-propagation* [Rumelhart et al., 1986], QuickProp [Fahlman, 1988b, Fahlman, 1988a], Rprop [Riedmiller & Braun, 1993] e outros.

2.2.1.2 Treinamento não-supervisionado

Caracteriza-se pela ausência do conhecimento da saída desejada da RNA para um determinado conjunto de treinamento [Braga et al., 2000]. Este tipo de treinamento detecta características estatísticas nos dados, criando representações internas e típicas à topologia utilizada.

A regra de Hebb e a técnica de *Winner-Takes-All* [Kohonen, 1982] são exemplos deste tipo de treinamento.

2.2.1.3 Treinamento por reforço

O treinamento por reforço [H.Hassoun, 1995] é um estágio intermediário entre o treinamento supervisionado e o treinamento não supervisionado. É caracterizado pelo não conhecimento da exata saída desejada da rede, mas pelo conhecimento da execução de uma determinada tarefa pela rede como, por exemplo, manter um pêndulo invertido sempre na posição vertical, independente de qual seja a ação da rede sobre esse pêndulo.

2.2.2 Regra de Treinamento de Widrow-Hoff

A chamada "Regra de treinamento de *Widrow-Hoff*" foi formalmente apresentada em 1960 [Widrow & Hoff, 1960] para o treinamento do neurônio denominado ADALINE. O ajuste é realizado em função do erro entre o valor desejado e o valor fornecido pela saída. Posteriormente foi mostrado que esta regra converge para o mínimo erro médio quadrático

de saída, sendo uma regra baseada no método de gradiente descendente.

Sendo a função de custo definida pela Equação A.7, o método de gradiente descendente é expresso como:

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} - \mu \cdot \nabla E_{(k)} \quad (2.4)$$

onde, $\nabla E_{(k)} = \frac{\partial E_{(k)}}{\partial \mathbf{W}}$ e, para apenas um elemento, $E_{(k)} = \frac{1}{2} (Y_{d(k)} - Y_{(k)})^2$, onde $Y_{(k)} =$

$\sum_{l=1}^{m+1} (W_{l(k)} \cdot X_{l(k)})$, portanto, utilizando a regra da cadeia:

$$\nabla E_{(k)} = -1 \cdot (Y_{d(k)} - Y_{(k)}) \cdot \frac{\partial Y_{(k)}}{\partial \mathbf{W}}$$

Definindo-se $e_{(k)} = Y_{d(k)} - Y_{(k)}$, tem-se:

$$\nabla E_{(k)} = -e_{(k)} \cdot \frac{\partial Y_{(k)}}{\partial \mathbf{W}}$$

Considerando que, para o Adaline, utiliza-se a saída linear do neurônio, ou seja, $\frac{\partial Y_{(k)}}{\partial \mathbf{W}} = X_{(k)}$, a regra de atualização dos pesos pode, portanto, ser expressa por:

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} + \mu \cdot e_{(k)} \cdot \mathbf{X}_{(k)} \quad (2.5)$$

onde:

\mathbf{W} : vetor de pesos;

e : erro entre a saída desejada (Y_d) para o elemento e a saída real do elemento (Y)

\mathbf{X} : vetor de entrada;

μ : ganho do treinamento, $\mu > 0$.

Ou, para sua versão de treinamento em lote, como foi originalmente proposta, tem-se que $E_{(k)} = \frac{1}{2} \sum_{v=1}^q \left(Y_{d(k)}^v - Y_{(k)}^v \right)^2$ e, prosseguindo-se de maneira análoga:

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} + \mu \cdot \sum_{v=1}^q \left[e_{(k)}^v \cdot \mathbf{X}_{(k)}^v \right] \quad (2.6)$$

onde o sobrescrito v refere-se ao vetor de treinamento apresentado do conjunto de treinamento que contém q elementos.

2.2.3 Regra Delta

A Regra Delta é a generalização da Regra de Treinamento de *Widrow-Hoff*, sendo utilizada para neurônios cuja função de ativação é não-linear e diferenciável. Neste caso,

$$Y_{(k)} = f(V_{(k)}), \text{ onde } V_{(k)} = \sum_{l=1}^{m+1} (W_{l(k)} \cdot X_{l(k)}), \text{ portanto,}$$

$$\frac{\partial Y_{(k)}}{\partial \mathbf{W}} = \frac{\partial f(V_{(k)})}{\partial V_{j(k)}} \frac{\partial V_{j(k)}}{\partial \mathbf{W}}, \text{ logo:}$$

$$\frac{\partial Y_{(k)}}{\partial \mathbf{W}} = f'(V_{(k)}) \cdot X_{(k)}$$

Desta forma, o gradiente do erro é expresso como (considerando a função de erro definida na Equação A.7):

$$\nabla E_{(k)} = -1 \cdot (Y_{d(k)} - Y_{(k)}) \cdot \frac{\partial Y_{(k)}}{\partial \mathbf{W}}$$

$$\nabla E_{(k)} = -1 \cdot (Y_{d(k)} - Y_{(k)}) \cdot f'(V_{(k)}) \cdot X_{(k)}$$

e a regra de atualização dos pesos pode ser escrita da seguinte maneira, considerando-se $e_{(k)} = Y_{d(k)} - Y_{(k)}$:

$$\mathbf{W}_{(k+1)} = \mathbf{W}_{(k)} + \mu \cdot e_{(k)} \cdot f'(V_{(k)}) \cdot X_{(k)} \quad (2.7)$$

Analisando essa expressão fica claro que, para o Adaline, onde a função de ativação do neurônio é uma função linear, ou seja, $f(V_{(k)}) = V_{(k)}$ e, conseqüentemente, $f'(V_{(k)}) = 1$, a Regra Delta reduz-se à Regra de Treinamento de *Widrow-Hoff*.

Considerando as funções de ativação comumente utilizadas nos nodos MCP com função de ativação não linear (sigmoide e tangente hiperbólica), pode-se observar (Figura 2.4) que suas derivadas são nulas quando a saída linear ($V_{(k)}$) possui valores altos ($|V_{(k)}| > 3$, para a tangente hiperbólica e $|V_{(k)}| > 6$ para a sigmoide). Esta característica é uma desvantagem da Regra Delta pois, nestes casos, a alteração dos pesos será, praticamente, nula e o neurônio não será capaz de apresentar um comportamento satisfatório, não assimilando a relação entre a entrada-saída, ou seja, não aprendendo.

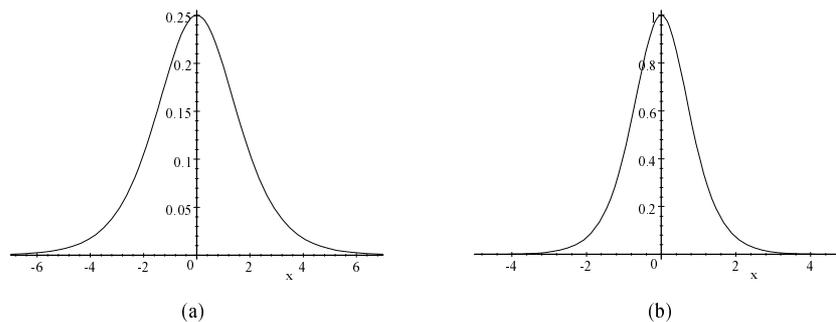


Figura 2.4- Gráficos das derivadas das funções de ativação. (a) derivada da sigmoide; (b) derivada da tangente hiperbólica.

2.2.4 Backpropagation

O algoritmo *Backpropagation* foi apresentado, para o treinamento de MLP, por Rumelhart, Hinton e Williams [Rumelhart et al., 1986] em 1986, baseando-se no método de gradiente descendente e caracterizando-se como um método de gradiente local.

Desta forma, a atualização dos pesos da RNA é feita obedecendo a Equação 2.3. Considerando-se uma rede de duas camadas para a camada de saída, tem-se que:

$$W_{ji(k+1)} = W_{ji(k)} - \eta \cdot \nabla E_{(k)}$$

Utilizando-se a Equação A.8:

$$W_{ji(k+1)} = W_{ji(k)} + \eta \cdot e_{(k)} \cdot f' (V_j(k)) \cdot Y_{Hi(k)}$$

Ou, a partir da Equação A.9, tem-se:

$$W_{ji(k+1)} = W_{ji(k)} + \eta \cdot \delta_{j(k)} \cdot Y_{Hi(k)} \quad (2.8)$$

Para a camada escondida, considerando-se a Equação 2.3:

$$Z_{ih(k+1)} = Z_{ih(k)} - \eta \cdot \nabla E_{(k)}$$

Utilizando-se a Equação A.11:

$$Z_{ih(k+1)} = Z_{ih(k)} + \eta \cdot T_{h(k)} \cdot f'_H (R_i(k)) \cdot \sum_{j=1}^p [\delta_{j(k)} \cdot W_{ji(k)}]$$

Ou, a partir da Equação A.12:

$$Z_{ih(k+1)} = Z_{ih(k)} + \eta \cdot \delta_{i(k)} \cdot T_{h(k)} \quad (2.9)$$

Observa-se que a Equação 2.7 é similar à Equação 2.8, quando se tem apenas um neurônio. Desta forma, o problema relacionado à derivada da função de ativação dos neurônios de saída que ocorre na Regra Delta, também aparecerá no algoritmo *Backpropagation*.

Analisando-se a Equação 2.9, observa-se a existência do termo da derivada da função de ativação dos nodos da camada escondida, o que provocará o problema semelhante ao existente na camada de saída (e à Regra Delta).

Comparando-se a Equação 2.8 com a Equação 2.9, pode-se utilizar a seguinte regra generalizada de atualização dos pesos para uma rede com um número x de camadas:

$$W_{ji}^x = W_{ji}^{x(k)} + \eta \cdot \delta_{j(k)}^x \cdot Y_{i(k)}^x$$

onde:

W_{ji}^x : peso que conecta a entrada i (ou saída do neurônio da camada anterior) ao neurônio j da camada x ;

$\delta_{j(k)}^x$: gradiente local do neurônio j da camada x ;

$Y_{i(k)}^x$: Entrada i (ou saída do neurônio da camada anterior) do peso W_{ji}^x da camada x .

O algoritmo Backpropagation sofre de algumas fortes deficiências:

1. Considerando o fato de ser um algoritmo baseado no método de gradiente descendente, se a superfície de erro possuir uma inclinação pequena, o valor do gradiente será, conseqüentemente, pequeno e o sistema de treinamento caminhará lentamente nesta superfície. Se a superfície de erro contiver regiões planas, o treinamento da RNA será interrompido;
2. Por ser um método de gradiente local, o treinamento da RNA poderá ficar preso em um mínimo local, podendo não atingir um determinado critério de erro não por deficiência na estrutura da rede, mas sim pela existência deste mínimo local;
3. Não existe uma metodologia bem definida para a determinação das taxas de treinamento (η);
4. Devido à existência das derivadas das funções de ativação nas expressões de ajuste dos pesos, um determinado neurônio pode permanecer sem o ajuste em seus pesos se sua saída linear atingir valores elevados.

Apesar das características mencionadas anteriormente, o algoritmo *backpropagation* é amplamente utilizado no treinamento de RNAs MLP. Desde sua publicação, vários novos algoritmos são propostos na tentativa de aumentar a eficiência do treinamento; entretanto, grande parte dos novos algoritmos se mantém fiel à proposta do *Backpropagation*.

2.2.4.1 Treinamento em lote

De maneira similar à Regra de Treinamento de *Widrow-Hoff*, pode-se determinar uma regra de atualização dos pesos para o algoritmo *Backpropagation* utilizando treinamento em lote. Para tal, define-se a seguinte função de custo:

$$E_{(k)} = \frac{1}{2} \sum_{v=1}^q \sum_{j=1}^p \left(e_{j(k)}^v \right)^2 \quad (2.10)$$

ou, em sua forma vetorial:

$$E_{(k)} = \frac{1}{2} \sum_{q=1}^Q \mathbf{e}_{q(k)}^T \cdot \mathbf{e}_{q(k)} \quad (2.11)$$

onde o sobrescrito v refere-se ao v -ésimo vetor de treinamento apresentado do conjunto de treinamento que contém Q elementos, $e_{j(k)}^v = Y_{dj(k)}^v - Y_{j(k)}^v$ e o subscrito j refere-se ao neurônio j da camada de saída que contém p neurônios.

Procedendo de maneira similar à utilizada na seção anterior, determinam-se as regras de ajuste dos pesos para a camada de saída e escondida como, respectivamente:

$$W_{ji(k+1)} = W_{ji(k)} + \eta \cdot \sum_{v=1}^q e_{j(k)}^v \cdot f' \left(V_{j(k)}^v \right) \cdot Y_{Hi(k)}^v \quad (2.12)$$

$$Z_{ih(k+1)} = Z_{ih(k)} + \eta \cdot \sum_{v=1}^q \left\{ T_{h(k)}^v \cdot f_H' \left(R_{i(k)} \right) \cdot \sum_{j=1}^p \left[e_{j(k)}^v \cdot f' \left(V_{j(k)}^v \right) \cdot W_{ji(k)}^v \right] \right\} \quad (2.13)$$

2.2.4.2 Ganho adaptativo

Algumas das propostas para melhorar a eficiência do algoritmo *Backpropagation* padrão são baseadas na escolha adequada da taxa de aprendizado (η). Uma escolha apropriada da taxa de aprendizado permitiria evitar mínimos locais ou, então, uma convergência mais rápida para um mínimo.

No trabalho desenvolvido por Schiffmann, Joost e Werner [Schiffmann et al., 1992], é feita uma comparação entre diferentes técnicas de adaptação da taxa de treinamento.

Podem ser citados vários trabalhos que utilizam taxa de treinamento adaptativo, tais como os trabalhos de Schmidhuber em 1989 [Schmidhuber, 1989], Darken e Moody em 1990 [Darken & Moody, 1991], Moreira e Fiesler em 1995 [Moreira & Fiesler, 1995], dentre vários outros.

2.2.5 Momentum

Para aumentar a velocidade de convergência do algoritmo *Backpropagation* padrão, principalmente em regiões planas da superfície de erro, introduziu-se um termo de ajuste nas regras de atualização dos pesos, as quais passam a ser, para a camada de saída:

$$W_{ji(k+1)} = W_{ji(k)} + \eta \cdot \delta_{j(k)} \cdot Y_{Hi(k)} + \mu \cdot W_{ji(k-1)} \quad (2.14)$$

e para a camada escondida:

$$Z_{ih(k+1)} = Z_{ih(k)} + \eta \cdot T_{h(k)} \cdot \delta_{i(k)} + \mu \cdot Z_{ih(k-1)} \quad (2.15)$$

onde:

$\delta_{j(k)}$ e $\delta_{i(k)}$ são definidos pelas Equações A.9 e A.12, respectivamente;

μ é chamado de taxa, ou termo, de momento ($\mu > 0$).

A inclusão do termo de momento [Bertsekas & Tsitsiklis, 1996] introduz uma característica de aceleração ao treinamento. Entretanto, esse termo pode provocar oscilações em uma superfície de erro que possua muitos mínimos. Da mesma forma que o algoritmo *Backpropagation*, não existe uma metodologia para a correta escolha do valor do termo de momento.

2.2.6 RPROP

Riedmiller e Braun [Riedmiller & Braun, 1993], apresentaram um algoritmo para o treinamento em lote de RNAs MLP baseado no sinal do gradiente, e não no valor do seu módulo, como acontece com o algoritmo *Backpropagation* padrão. Dessa forma, espera-se evitar problemas relacionados com superfícies muito planas, cujo módulo do gradiente é praticamente nulo. O algoritmo recebeu o nome de RPROP (Resilient PROPagation) e pode ser sumarizado, para o ajuste do peso W_{ji} da camada de saída, por:

$$\Delta_{ji(k)} = \begin{cases} \eta^+ \cdot \Delta_{ji(k-1)} & \text{se } \frac{\partial E_{(k-1)}}{\partial W_{ji}} \cdot \frac{\partial E_{(k)}}{\partial W_{ji}} > 0 \\ \eta^- \cdot \Delta_{ji(k-1)} & \text{se } \frac{\partial E_{(k-1)}}{\partial W_{ji}} \cdot \frac{\partial E_{(k)}}{\partial W_{ji}} < 0 \\ \Delta_{ji(k-1)} & \text{caso contrario} \end{cases} \quad (2.16)$$

$$\Delta W_{ji(k)} = \begin{cases} -\Delta_{ji(k)} & \text{se } \frac{\partial E_{(k)}}{\partial W_{ji}} > 0 \\ \Delta_{ji(k)} & \text{se } \frac{\partial E_{(k)}}{\partial W_{ji}} < 0 \\ 0 & \text{caso contrario} \end{cases} \quad (2.17)$$

$$W_{ji(k+1)} = W_{ji(k)} + \Delta W_{ji(k)} \quad (2.18)$$

Onde:

$$\begin{cases} \eta^+ > 1 \\ 0 < \eta^- < 1 \end{cases}$$

Para os pesos das camadas escondidas o algoritmo é o mesmo, mudando apenas no cálculo do gradiente que é sempre relativo aos pesos da camada na qual se faz os ajustes.

Como foi dito, o módulo da variação dos pesos (Δ_{ji}) não é mais função do gradiente. Ele é ajustado de maneira adaptativa, seguindo a seguinte lógica:

Se em dois ajustes sucessivos o gradiente não alterou seu sinal, significando que o ajuste dos pesos não passou por nenhum mínimo, o passo de ajuste, ou seja, o módulo da variação dos pesos, pode ser aumentado por um fator η^+ , diminuindo o tempo de treinamento e evitando regiões planas da superfície de erro.

Caso ocorra alguma variação de sinal do gradiente em dois ajustes sucessivos, significando a passagem por um mínimo, o módulo de variação dos pesos é diminuído por um fator η^- , na tentativa de se encontrar o mínimo em questão.

Uma vez determinado o valor do módulo do ajuste dos pesos, é necessário determinar a direção para a execução do ajuste. Lembrando que o gradiente é a taxa de variação do erro em função da variação do peso, um gradiente positivo significa que, aumentando-se o peso, o erro aumentará. Já um gradiente negativo significa que, aumentando-se o peso, o erro diminuirá. Desta forma, se o gradiente do erro for positivo, o peso deverá ser diminuído e, portanto, $\Delta W_{ji} = -\Delta_{ji}$. Se o gradiente do erro for negativo, o peso deverá ser aumentado, logo, $\Delta W_{ji} = +\Delta_{ji}$. Se o gradiente for nulo, não é realizada nenhuma alteração no valor do peso.

Para a execução do algoritmo de RPROP é necessário, inicialmente, determinar o valor de η^+ e η^- . Em seu trabalho, Riedmiller e Braun utilizam $\eta^+ = 1.2$ e $\eta^- = 0.5$. Entretanto não existe uma metodologia para a determinação destes valores. A escolha inadequada pode provocar oscilações, ou um salto sobre um mínimo, impossibilitando o treinamento da RNA.

É necessário também, como acontece com qualquer outro algoritmo de treinamento, inicializar os pesos da RNA. Neste caso o algoritmo se apresenta robusto, não dependente da escolha dos pesos iniciais.

Na implementação prática são estabelecidos dois limites, máximo e mínimo, para o módulo da variação dos pesos ($\Delta_{ji(k)}$) para que seja evitado uma saturação dos nodos da RNA ($\Delta_{ji(k-1)} \rightarrow \infty$) ou que o processo de adaptação seja interrompido (se $\Delta_{ji(k-1)} = 0$).

Pela explicação fornecida e análise do algoritmo, fica claro que este procedimento

não impede que a RNA fique presa em mínimos locais, não sendo, portanto, um algoritmo de mínimo global. Não existe, também, uma formulação matemática bem definida para o estudo de convergência do algoritmo. O vínculo existente entre este algoritmo e o *Backpropagation* padrão reside no cálculo do gradiente do erro em relação aos pesos a serem ajustados, o qual é determinado pelas expressões 2.8 e 2.9 para a camada de saída e camada escondida, respectivamente.

2.2.7 QUICKPROP

Scott Fahlman apresentou em 1988 [Fahlman, 1988b] um algoritmo, o qual ele denominou de QUICKPROP (QUICK PROPagation) e conhecido por QProp, baseado no conhecimento da superfície de erro.

Fahlman, em seu trabalho, faz duas considerações:

1. A superfície de erro do conjunto de treinamento é uma parábola em função do peso, como mostra a Figura 2.5;
2. A parábola não é afetada pelo ajuste dos demais pesos existentes na RNA.

O ajuste do peso é feito de tal forma que, em um passo de cálculo, considerando o estado do peso anterior, a função de erro é minimizada. Para isso considera-se, como foi dito, a seguinte função de erro:

$$E_{(k)} = a \cdot W_{(k)}^2 + b \cdot W_{(k)} + c \quad (2.19)$$

onde $a > 0$, b e c são constantes definidas em função do erro.

Considerando a função de erro definida na Equação 2.19, cujo gráfico está mostrado na Figura 2.5, pode-se determinar, com relação a última atualização do peso ($W_{(k-1)}$), o próximo passo do peso ($\Delta W_{(k)}$) de forma que o mínimo da parábola seja atingido.

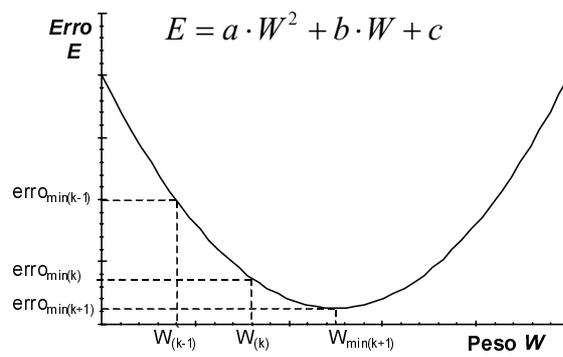


Figura 2.5- Função de erro considerada no algoritmo QProp.

Derivando a Equação 2.19, com relação ao peso, tem-se:

$$\frac{\partial E}{\partial W}(k) = 2 \cdot a \cdot W_{(k)} + b \quad (2.20)$$

cujo gráfico está mostrado na Figura 2.6

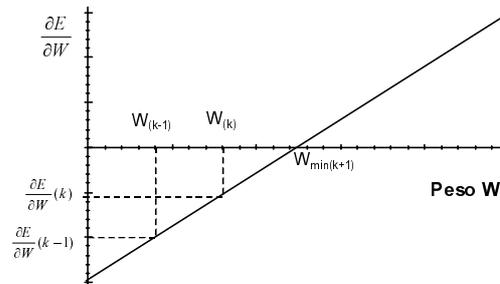


Figura 2.6- Derivada da função de erro considerada no algoritmo QProp.

Analisando a Equação 2.20, observa-se que a inclinação da reta é fornecida pelo termo $2 \cdot a$, o qual é, agora, uma variável de interesse. Pode-se determiná-lo utilizando o peso no último passo de cálculo, ou seja,

$$2 \cdot a = \frac{\frac{\partial E_{(k)}}{\partial W} - \frac{\partial E_{(k-1)}}{\partial W}}{\Delta W_{(k-1)}} \quad (2.21)$$

A Equação 2.21 será utilizada posteriormente.

O valor do peso que irá minimizar a função de erro pode, então, ser determinado em função da derivada do erro, ou seja,

$$\frac{\partial E}{\partial W} = 0 \Rightarrow W_{(K+1)} = W_{\min(K+1)}, \text{ logo,}$$

$$2 \cdot a \cdot W_{\min(k+1)} + b = 0$$

$$W_{\min(K+1)} = -\frac{b}{2 \cdot a} \quad (2.22)$$

Escrevendo a equação para o ajuste do peso, tem-se:

$$W_{(K+1)} = W_{(K)} + \Delta W_{(K)}$$

$$\Delta W_{(K)} = W_{(K+1)} - W_{(K)}$$

Considerando que o objetivo do algoritmo QUICKPROP é encontrar o mínimo da função de erro em apenas um passo de cálculo, ou seja, $W_{(K+1)} = W_{\min(K+1)}$, pode-se escrever,

$$\Delta W_{(K)} = W_{\min(K+1)} - W_{(K)}$$

Reescrevendo a Equação 2.20, colocando o peso em função da derivada do erro, e utilizando a Equação 2.22,

$$\Delta W_{(K)} = -\frac{b}{2 \cdot a} - \frac{\frac{\partial E_{(k)}}{\partial W} - b}{2 \cdot a}$$

$$\Delta W_{(K)} = \frac{-\frac{\partial E_{(k)}}{\partial W}}{2 \cdot a}$$

Utilizando a Equação 2.21,

$$\Delta W_{(K)} = \frac{\frac{\partial E_{(k)}}{\partial W}}{\frac{\partial E_{(k-1)}}{\partial W} - \frac{\partial E_{(k)}}{\partial W}} \cdot \Delta W_{(k-1)} \quad (2.23)$$

a qual é a regra de atualização dos pesos utilizada no algoritmo QUICKPROP.

Analisando a Equação 2.23, observa-se que o ajuste do peso é feito de maneira recursiva, ou seja, a taxa de variação do peso para o instante atual é função da taxa de variação do peso que foi utilizada no passo anterior. Entretanto, não existe uma taxa de aprendizado que necessite ser ajustada.

O algoritmo QUICKPROP apresenta a grande vantagem de ser mais rápido do que o algoritmo *Backpropagation* padrão, existindo, porém, algumas peculiaridades que necessitam ser explicadas:

1. Por ser um algoritmo recursivo, podem acontecer casos nos quais ou o passo de cálculo é muito grande ou muito pequeno, podendo o treinamento ficar paralisado. Em seu trabalho, Fahlman propôs algumas metodologias para a reinicialização do algoritmo. Uma de suas propostas é acrescentar um termo similar ao algoritmo *Backpropagation* na atualização dos pesos, sendo necessário, agora, determinar um valor para a taxa de treinamento;
2. Podem ocorrer oscilações em torno do mínimo da função;
3. Observa-se uma diferença entre o gradiente do erro no instante atual e no instante anterior, no denominador da Equação 2.23. Estando o sistema em uma área muito plana da superfície de erro, podem ocorrer problemas numéricos, levando o neurônio à saturação e a um comportamento não desejado da rede. Na implementação prática é feita uma saturação do máximo fator de crescimento do peso, chamado, por Fahlman, de *Maximum Growth Factor*, o qual não possui uma metodologia matemática para sua determinação;
4. O algoritmo não garante que será atingido um mínimo global, podendo ficar preso em mínimos locais;
5. Em seu trabalho, Fahlman utiliza uma modificação da função de ativação dos neurônios, acrescentando um *off-set* de forma que o valor de $\frac{\partial E(k)}{\partial W}$ nunca seja zero e evitando, desta forma, que valores elevados da saída linear da RNA impossibilitem o treinamento, como acontece com o algoritmo *Backpropagation* padrão.

Apesar de, na teoria, ser um algoritmo de fácil implementação e bastante eficiente, sua implementação prática não é trivial, devido às considerações mencionadas acima. A determinação do gradiente do erro é feita de maneira idêntica à desenvolvida no algoritmo *Backpropagation*.

2.2.8 Adaptação da Taxa de Aprendizado pela Mudança de Sinal

Silva e Almeida [Silva & Almeida, 1990] utilizam, no treinamento da RNA, uma taxa de aprendizado (η) para cada conexão, sendo que esta taxa é adaptada em função do sinal

do gradiente do erro no instante atual e no instante anterior. O algoritmo pode ser descrito, para um peso W_{ji} , como:

$$\eta_{ij(k)} = \begin{cases} \eta_{ij(k-1)} \cdot u & \text{se } \frac{\partial E(k)}{\partial W} \cdot \frac{\partial E(k-1)}{\partial W} \geq 0 \\ \eta_{ij(k-1)} \cdot d & \text{caso contrario} \end{cases} \quad (2.24)$$

$$W_{ji(k+1)} = -\eta_{ij(k)} \cdot \frac{\partial E(k)}{\partial W_{ij}} + \mu \cdot W_{ji(k-1)} \quad (2.25)$$

onde,

$$u \simeq \frac{1}{d} \quad (2.26)$$

Como pode ser observado em seu trabalho, Silva e Almeida utilizam um termo similar ao Momentum (seção 2.2.5). Eles utilizam ainda, em sua proposta, uma estratégia de *backtracking*, a qual volta um passo de cálculo, em toda a rede, se houver uma aumento do erro total.

2.2.9 SuperSAB

O algoritmo proposto por Tollenaere em 1990 [Tollenaere, 1990] é similar ao mostrado na seção anterior, proposto por Silva e Almeida. Entretanto, em sua proposta, Tollenaere não faz as atualizações nos pesos que provocaram alteração no sinal do gradiente. Seu algoritmo, para um peso W_{ji} , pode ser sumarizado como:

$$\eta_{ij(k)} = \begin{cases} \eta_{ij(k-1)} \cdot u & \text{se } \frac{\partial E(k)}{\partial W} \cdot \frac{\partial E(k-1)}{\partial W} \geq 0 \\ \eta_{ij(k-1)} \cdot d & \text{caso contrario} \end{cases} \quad (2.27)$$

$$W_{ji(k+1)} = \begin{cases} -\eta_{ij(k)} \cdot \frac{\partial E(k)}{\partial W_{ij}} + \mu \cdot W_{ji(k-1)} & \text{se } \frac{\partial E(k)}{\partial W} \cdot \frac{\partial E(k-1)}{\partial W} \geq 0 \\ W_{ji(k)} & \text{caso contrario} \end{cases} \quad (2.28)$$

2.2.10 Levenberg-Marquardt

Martin Hagan e Mohammad Menhaj, em 1994, apresentaram um algoritmo que utiliza mínimos quadráticos não-lineares para o treinamento em lote de RNAs MLP [Hagan & Menhaj, 1994]. Em seu trabalho, Hagan e Menhaj afirmam que, para RNAs muito grandes, acima de algumas centenas de parâmetros, a memória requerida impossibilita a implementação do sistema proposto. É importante ressaltar que essa afirmativa foi baseada nos computadores de 1994.

O algoritmo de Levenberg-Marquardt [Marquardt, 1963] é uma aproximação do método de Newton, possibilitando minimizar uma função em relação a um determinado vetor de parâmetros.

Seja a seguinte função que se deseja minimizar com respeito ao vetor de parâmetros \mathbf{x} , dada pelo seguinte soma de quadrados:

$$V(\mathbf{x}) = \sum_{i=1}^N e_{i(\mathbf{x})}^2 \quad (2.29)$$

Para minimizar essa função, com relação ao vetor \mathbf{x} , tem-se que:

$$\Delta \mathbf{x} = - [\nabla^2 V(\mathbf{x})]^{-1} \cdot \nabla V(\mathbf{x}) \quad (2.30)$$

onde,

$\nabla^2 V(\mathbf{x})$ é a matriz Hessiana de $V(\mathbf{x})$;

$\nabla V(\mathbf{x})$ é o gradiente da função $V(\mathbf{x})$.

Pode-se mostrar que, utilizando o método de Gauss-Newton ($\sum_{i=1}^N [e_{i(\mathbf{x})} \cdot \nabla^2 e_{i(\mathbf{x})}] \approx 0$):

$$\nabla V(\mathbf{x}) = J_{(\mathbf{x})}^T \cdot \mathbf{e}(\mathbf{x}) \quad (2.31)$$

$$\nabla^2 V(\mathbf{w}) = J_{(\mathbf{x})}^T \cdot J_{(\mathbf{x})} \quad (2.32)$$

onde J é a matriz jacobiana, ou seja,

$$J_{(\mathbf{x})} = \begin{bmatrix} \frac{\partial e_{1(\mathbf{x})}}{\partial W_1} & \frac{\partial e_{1(\mathbf{x})}}{\partial W_2} & \cdots & \frac{\partial e_{1(\mathbf{x})}}{\partial W_N} \\ \frac{\partial e_{2(\mathbf{x})}}{\partial W_1} & \frac{\partial e_{2(\mathbf{x})}}{\partial W_2} & \cdots & \frac{\partial e_{2(\mathbf{x})}}{\partial W_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{q(\mathbf{x})}}{\partial W_1} & \frac{\partial e_{q(\mathbf{x})}}{\partial W_2} & \cdots & \frac{\partial e_{q(\mathbf{x})}}{\partial W_N} \end{bmatrix} \quad (2.33)$$

logo,

$$\Delta \mathbf{x} = - \left[J_{(\mathbf{x})}^T \cdot J_{(\mathbf{x})} \right]^{-1} \cdot J_{(\mathbf{x})}^T \cdot \mathbf{e}_{(\mathbf{x})} \quad (2.34)$$

O método de Levenberg-Marquardt introduz uma pequena modificação no método de Gauss-Newton e a Equação 2.34 é alterada, tornando-se:

$$\Delta \mathbf{x} = - \left[J_{(\mathbf{x})}^T \cdot J_{(\mathbf{x})} + \mu \cdot I \right]^{-1} \cdot J_{(\mathbf{x})}^T \cdot \mathbf{e}_{(\mathbf{x})} \quad (2.35)$$

O parâmetro μ é multiplicado por um fator β sempre que o ajuste nos pesos provocar um aumento na função de erro e dividido por β caso contrário. Se o fator μ for muito grande, o algoritmo proposto torna-se um método de gradiente descendente, ou um método de Gauss-Newton, se o fator μ for muito pequeno.

A principal questão neste método é como determinar a matriz jacobiana ($J_{(\mathbf{x})}$). Para problemas de treinamento de RNAs, pode-se determiná-la com algumas modificações no algoritmo *Backpropagation* padrão para treinamento em lote.

Observa-se que a Equação 2.29 é igual à Equação 2.11 se considerarmos o vetor \mathbf{x} igual a:

$$\mathbf{x} = \left[W_{1,1}^1 \quad W_{1,2}^1 \quad \cdots \quad W_{S_1,R_1}^1 \quad W_{1,1}^2 \quad \cdots \quad W_{S_2,R_2}^2 \quad \cdots \quad W_{1,1}^M \quad \cdots \quad W_{S_M,R_M}^M \right] \quad (2.36)$$

onde W_{S_i,R_i}^i é o elemento S, R da i -ésima matriz de pesos de uma RNA contendo M camadas. Incluem-se, nesta matriz de pesos, os termos de polarização.

O índice N do somatório (Equação 2.29) é igual à $N = Q * M$, onde Q é o número

total de padrões que serão utilizados no treinamento.

Desta forma, pode-se determinar a matriz Jacobiana (Equação 2.33) alterando-se o algoritmo *Backpropagation* de tal forma que o vetor de gradiente local da camada de saída seja dado por:

$$\delta_M = -F'_{M(V_M)} \quad (2.37)$$

onde:

V_M é a saída linear da camada de saída;

$F'_{M(V_M)}$ é uma matriz diagonal cujos elementos são as derivadas das funções de ativação da camada de saída calculadas no valor da saída linear. Cada coluna desta matriz será propagada de volta na rede produzindo uma linha da matriz jacobiana.

Outros algoritmos que utilizam técnicas de otimização também são encontrados na literatura. Uma das opções é a utilização do *Backpropagation* com o método de Polak-Ribiere ([Kramer & S.-Vincentelli, 1989]). Neste algoritmo as operações em cada iteração são menos intensivas do que no algoritmo de Levenberg-Marquardt apesar deste algoritmo necessitar de um maior número de *epochs* para o completo treinamento da rede. Outros algoritmos utilizam, ainda, alguma forma de limitação de memória em métodos quasi-newton ([Battiti & Massuli, 1990]) pois, nestes métodos, o esforço computacional é proporcional ao quadrado da dimensão da RNA ([Hagan & Menhaj, 1994]).

2.3 Conclusões

Neste capítulo foi feita uma breve abordagem ao histórico do desenvolvimento da área de RNAs e foram apresentados os principais algoritmos da topologia MLP encontrados na literatura. Dentre os algoritmos apresentados, os algoritmos *Backpropagation*, *Momentum*, *QProp*, *Rprop*, *Levenberg-Marquadt* e dois algoritmos com taxa de aprendizado variável serão utilizados como base para a comparação com os algoritmos que serão propostos no Capítulo 4. A comparação dos algoritmos de treinamento *off-line* será realizada em problemas clássicos da área de RNAs, aproximação de funções e classificação.

Dentre os sete algoritmos implementados, além dos propostos, os algoritmos Rprop e Levenberg-Marquadt serão os principais a serem comparados, por apresentarem os melhores resultados dentre os algoritmos clássicos, nos problemas analisados.

Foram analisadas, metodologicamente, as principais características e deficiências de cada um dos algoritmos apresentados, sendo que, pode-se considerar como a principal deficiência a não existência de uma análise matemática tanto dos ganhos envolvidos nos treinamentos quanto da convergência do algoritmo.

Capítulo 3

Sistemas de Estrutura Variável

Neste capítulo serão introduzidos os conceitos básicos da Teoria de Sistemas de Estrutura Variável (SEV) e Controle por Modos Deslizantes (CMD). Será feita a abordagem de um sistema de controle para uma planta genérica de segunda ordem, considerando sistema contínuo e sistema amostrado. Serão abordados, também, alguns algoritmos híbridos, encontrados na literatura, utilizando SEV e RNAs.

3.1 Introdução

Os sistemas de controle usualmente utilizados (proporcional e/ou integral e/ou derivativo, controle por modelo de referência e outros) mantêm, em sua maioria, fixa a estrutura da planta a ser controlada, podendo considerar variações paramétricas ou, ainda, considerando o sistema linear ou não. Já um sistema de controle, em malha fechada, baseado em SEV, promove mudanças na estrutura da planta a ser controlada. A capacidade de alterar a estrutura de uma planta possibilita ao controle resolver um dilema típico, existente nos demais sistemas de controle: o compromisso entre estabilidade, robustez e velocidade de convergência. Em princípio, podem existir leis de controle distintas que possibilitam a mudança na estrutura de uma planta.

Em Controle por Modos Deslizantes, a estrutura da planta é alterada em função dos estados do sistema. Em princípio, esta alteração ocorre a uma frequência infinita sendo provocada pelo chaveamento da variável de controle que obedece a uma lei de controle pré-definida pelo controlador. Desta forma, definem-se hiperplanos de chaveamento sobre o espaço de estados e a estrutura da planta é, então, chaveada, sempre que o ponto representativo (PR) do sistema atravessar um destes hiperplanos.

3.2 Sistemas de Segunda Ordem

Seja um sistema monovariável de segunda ordem, com parâmetros invariantes no tempo, descrito pela seguinte equação diferencial e para o qual deseja-se determinar um sistema de CMD:

$$\ddot{x} + a_2 \cdot \dot{x} + a_1 \cdot x = b \cdot u \quad (3.1)$$

onde u é a entrada de controle e o ponto sobre a variável denota derivada temporal $\left(\dot{x} = \frac{dx}{dt}\right)$.

Considerando:

$$\begin{cases} x_1 = x & \Rightarrow & \dot{x}_1 = x_2 \\ x_2 = \dot{x} & \Rightarrow & \dot{x}_2 = -a_2 \cdot x_2 - a_1 \cdot x_1 + b \cdot u \end{cases}$$

onde u , x , x_1 e x_2 são variáveis em t .

Portanto, o sistema descrito pela Equação 3.1 pode ser descrito pelo seguinte conjunto de equações diferenciais:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -a_2 \cdot x_2 - a_1 \cdot x_1 + b \cdot u \end{cases} \quad (3.2)$$

onde, x_1 é o erro existente entre a saída da planta e a referência ($x_1 = e(t)$) e x_2 é a derivada do erro ($x_2 = \frac{de(t)}{dt}$).

A superfície de deslizamento normalmente utilizada é, então, definida como uma combinação linear entre o erro e sua derivada, ou seja:

$$S = C \cdot x_1 + x_2 \triangleq 0 \quad (3.3)$$

onde C é uma constante positiva ($C > 0$). A Figura 3.1 mostra o gráfico da Equação 3.3 no plano de fase definido pelas variáveis x_1 e x_2 .

O sinal de controle u é, então, definido em função da superfície de deslizamento e

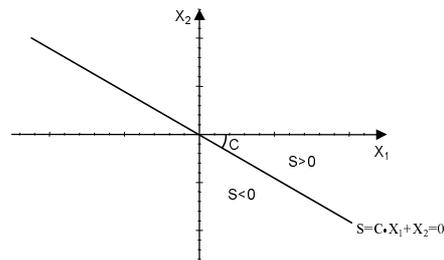


Figura 3.1- Superfície de deslizamento utilizada nos sistemas de CMD.

da variável x_1 :

$$u = -\alpha \cdot |x_1| \cdot \text{sgn}(S) \tag{3.4}$$

onde: $\alpha > 0$ e

$$\text{sgn}(S) = \begin{cases} 1 & \text{se } S > 0 \\ -1 & \text{se } S < 0 \end{cases} .$$

A condição de existência de modos deslizantes [Utkin, 1978] é satisfeita quando:

$$S \cdot \dot{S} < 0 \tag{3.5}$$

Estando em regime deslizante, a condição de existência de modos deslizantes garante a estabilidade do sistema. Considere a seguinte equação candidata de Lyapunov, $v = \frac{1}{2}S^2$. Observa-se que a equação candidata é, além de ser uma função dos estados do sistema, definida positiva e sua derivada determinada por $\dot{v} = S \cdot \dot{S}$, a qual tem que ser, necessariamente, definida negativa. Desta forma, a análise da estabilidade do sistema, considerando o critério de Lyapunov, resulta na Equação 3.5.

Derivando-se a Equação 3.3, tem-se:

$$\dot{S} = C \cdot \dot{x}_1 + \dot{x}_2$$

utilizando as Equações 3.2 e 3.4,

$$\dot{S} = C \cdot x_2 - a_2 \cdot x_2 - a_1 \cdot x_1 - b \cdot \alpha \cdot |x_1| \cdot \text{sgn}(S)$$

$$\dot{S} = (C - a_2) \cdot x_2 - a_1 \cdot x_1 - b \cdot \alpha \cdot |x_1| \cdot \text{sgn}(S)$$

Considerando o PR sobre a superfície de deslizamento, ou seja, $S = 0$, tem-se, a partir da Equação 3.3, $x_2 = -C \cdot x_1$. Substituindo-se esta equação na expressão acima,

$$\dot{S} = (a_2 \cdot C - a_1 - C^2) \cdot x_1 - b \cdot \alpha \cdot |x_1| \cdot \text{sgn}(S) \quad (3.6)$$

Para analisar a Equação 3.6, considerando a Equação 3.5, é necessário considerar quatro condições:

- **Condição 1:** $S > 0 \Rightarrow \dot{S} < 0; x_1 > 0;$
- **Condição 2:** $S > 0 \Rightarrow \dot{S} < 0; x_1 < 0;$
- **Condição 3:** $S < 0 \Rightarrow \dot{S} > 0; x_1 > 0;$
- **Condição 4:** $S < 0 \Rightarrow \dot{S} > 0; x_1 < 0.$

1. **Condição 1:**

Assumindo $S > 0 \Rightarrow \dot{S} < 0; x_1 > 0$, a partir da Equação 3.6, tem-se:

$$(a_2 \cdot C - a_1 - C^2) \cdot |x_1| - b \cdot \alpha \cdot |x_1| < 0$$

$$(a_2 \cdot C - a_1 - C^2 - b \cdot \alpha) \cdot |x_1| < 0$$

$$a_2 \cdot C - a_1 - C^2 - b \cdot \alpha < 0$$

$$\therefore \alpha > \frac{1}{b} (a_2 \cdot C - a_1 - C^2)$$

2. **Condição 2:**

Assumindo $S > 0 \Rightarrow \dot{S} < 0; x_1 < 0$, a partir da Equação 3.6, tem-se:

$$- (a_2 \cdot C - a_1 - C^2) \cdot |x_1| - b \cdot \alpha \cdot |x_1| < 0$$

$$- (a_2 \cdot C - a_1 - C^2 + b \cdot \alpha) \cdot |x_1| < 0$$

$$(a_2 \cdot C - a_1 - C^2 + b \cdot \alpha) \cdot |x_1| > 0$$

$$a_2 \cdot C - a_1 - C^2 + b \cdot \alpha > 0$$

$$\therefore \alpha > -\frac{1}{b} (a_2 \cdot C - a_1 - C^2)$$

3. **Condição 3:**

Assumindo $S < 0 \Rightarrow \dot{S} > 0; x_1 > 0$, a partir da Equação 3.6, tem-se:

$$(a_2 \cdot C - a_1 - C^2) \cdot |x_1| + b \cdot \alpha \cdot |x_1| > 0$$

$$(a_2 \cdot C - a_1 - C^2 + b \cdot \alpha) \cdot |x_1| > 0$$

$$a_2 \cdot C - a_1 - C^2 + b \cdot \alpha > 0$$

$$\therefore \alpha > -\frac{1}{b} (a_2 \cdot C - a_1 - C^2)$$

4. **Condição 4:**

Assumindo $S < 0 \Rightarrow \dot{S} > 0$; $x_1 < 0$, a partir da Equação 3.6, tem-se:

$$-(a_2 \cdot C - a_1 - C^2) \cdot |x_1| + b \cdot \alpha \cdot |x_1| > 0$$

$$-(a_2 \cdot C - a_1 - C^2 - b \cdot \alpha) \cdot |x_1| > 0$$

$$(a_2 \cdot C - a_1 - C^2 - b \cdot \alpha) \cdot |x_1| < 0$$

$$a_2 \cdot C - a_1 - C^2 - b \cdot \alpha < 0$$

$$\therefore \alpha > \frac{1}{b} (a_2 \cdot C - a_1 - C^2)$$

Pode-se sumarizar as quatro condições acima estudadas em:

$$\alpha > \frac{1}{b} |a_2 \cdot C - a_1 - C^2| \quad (3.7)$$

Considerando que a condição descrita pela Equação 3.7 é satisfeita e assumindo que, em regime deslizante (PR sobre a superfície de deslizamento), $S \equiv 0$, o movimento do PR pode ser descrito, a partir da Equação 3.3, por:

$$S = C \cdot x_1 + x_2 \equiv 0$$

$$x_2 = -C \cdot x_1$$

onde $x_2 = \frac{dx_1(t)}{dt}$, logo:

$$\frac{dx_1}{dt} = -C \cdot x_1, \text{ resolvendo esta equação diferencial tem-se:}$$

$$x_1(t) = x_1(t_0) \cdot \exp(-C \cdot (t - t_0))$$

Entretanto, conforme mencionado anteriormente, x_1 é o erro existente entre a referência e a saída da planta, logo:

$$e(t) = e(t_0) \cdot \exp(-C \cdot (t - t_0)) \quad (3.8)$$

onde $e(t_0)$ é o erro quando o sistema entra em regime deslizante.

Uma vez atingida a superfície de deslizamento, o PR permanecerá em regime deslizante se a Equação 3.7 for satisfeita. A partir deste instante, como pode ser observado pela Equação 3.8, para uma planta de segunda ordem (Equação 3.1 ou Equação 3.2), o erro possuirá uma dinâmica de primeira ordem com taxa de decrescimento igual à inclinação da superfície de deslizamento (variável C).

É necessário, agora, garantir que o PR, a partir de qualquer condição inicial, irá atingir a superfície de deslizamento. De acordo com o teorema 2.1.1 de Itkis [Itkis, 1976], o PR do sistema irá atingir a superfície de deslizamento se a equação característica do sistema com realimentação negativa não possuir raízes reais não-negativas.

A equação característica do sistema descrito pela Equação 3.1, considerando a realimentação negativa ($u = -\alpha \cdot x_1$), é dada por:

$\lambda^2 + a_2 \cdot \lambda + (a_1 + b \cdot \alpha)$, com as raízes determinadas por:

$$\lambda_{1,2} = \frac{-a_2 \pm \sqrt{a_2^2 - 4 \cdot (a_1 + b \cdot \alpha)}}{2}$$

Para garantir que o PR atinja a superfície de deslizamento é suficiente que as raízes da equação características sejam complexas, ou seja,

$$a_2^2 - 4 \cdot (a_1 + b \cdot \alpha) < 0$$

$$b \cdot \alpha > \frac{a_2^2}{4} - a_1$$

Portanto, quanto maior o ganho do controlador ($b \cdot \alpha$), maior a garantia de que o PR irá atingir a superfície de deslizamento. Além deste fato, maior a garantia do regime de deslizamento (Equação 3.7), ficando o sistema menos dependente dos parâmetros da planta.

Considerando-se um sistema variante no tempo, a garantia de existência de

modos deslizantes, definida pela inequação 3.7, pode ser reescrita como:

$$\alpha > \max \left\{ \frac{1}{b} |a_2(t) \cdot C - a_1(t) - C^2| \right\} \quad (3.9)$$

onde a função $\max \{x(t)\}$ retorna o máximo valor $x(t)$ para $t \in [0, \infty]$. Desta forma não há prejuízo nas análises feitas anteriormente.

3.3 Sistemas Amostrados

No desenvolvimento da teoria de SEV e CMD em tempo contínuo supõe-se que a planta a ser controlada será chaveada com frequência infinita, possibilitando ao PR deslizar sobre a superfície de deslizamento. Na implementação em tempo discreto, deve-se levar em conta que o sistema será chaveado com uma frequência máxima limitada pela frequência de amostragem. Desta forma o regime de deslizamento não será ideal e o PR poderá sair da superfície de chaveamento.

Sarpturk, em 1987[Sarpturk et al., 1987], define a Equação 3.10 como sendo condição necessária tanto para o regime de deslizamento quanto para a convergência do PR para a superfície. Entretanto, esta equação não garante que o PR atingirá a superfície, podendo existir uma convergência assintótica à superfície de deslizamento.

$$|S_{(k+1)}| < |S_{(k)}| \quad (3.10)$$

Seja o sistema discreto descrito pelo seguinte conjunto de equações:

$$\begin{cases} x_{1(k+1)} = x_{2(k)} \\ x_{2(k+1)} = -a_2 \cdot x_{2(k)} - a_1 \cdot x_{1(k)} + b \cdot u_{(k)} \end{cases} \quad (3.11)$$

e considere, ainda, que:

$$u_{(k)} = -\alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)}) \quad (3.12)$$

onde a_1 , a_2 e b são, agora, os parâmetros do sistema em tempo discreto, $\alpha > 0$ e x_1 , x_2 e u são, a partir deste momento, variáveis em k .

Deseja-se, agora, determinar os limites para o ganho α de forma a manter o regime de deslizamento sobre a superfície definida pela Equação 3.13.

$$S_{(k)} = C \cdot x_{1(k)} + x_{2(k)} \quad (3.13)$$

considerando o instante $k + 1$, na Equação 3.13,

$$S_{(k+1)} = C \cdot x_{1(k+1)} + x_{2(k+1)}$$

Substituindo-se a Equação 3.11 e 3.12 em 3.13, tem-se:

$$S_{(k+1)} = C \cdot x_{2(k)} - a_2 \cdot x_{2(k)} - a_1 \cdot x_{1(k)} - b \cdot \alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)})$$

$$S_{(k+1)} = (C - a_2) \cdot x_{2(k)} - a_1 \cdot x_{1(k)} - b \cdot \alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)})$$

A partir da Equação 3.13, tem-se: $x_{2(k)} = S_{(k)} - C \cdot x_{1(k)}$. Substituindo essa equação na anterior, obtém-se:

$$S_{(k+1)} = (C - a_2) \cdot (S_{(k)} - C \cdot x_{1(k)}) - a_1 \cdot x_{1(k)} - b \cdot \alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)})$$

$$S_{(k+1)} = (C - a_2) \cdot S_{(k)} - C \cdot x_{1(k)} \cdot (C - a_2) - a_1 \cdot x_{1(k)} - b \cdot \alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)})$$

$$S_{(k+1)} = (C - a_2) \cdot S_{(k)} - [C^2 - a_2 \cdot C + a_1] \cdot x_{1(k)} - b \cdot \alpha \cdot |x_{1(k)}| \cdot \text{sgn}(S_{(k)}) \quad (3.14)$$

A partir deste ponto é necessário considerar quatro condições:

- Condição 1: $S_{(k)} > 0; x_{1(k)} > 0;$
- Condição 2: $S_{(k)} > 0; x_{1(k)} < 0;$
- Condição 3: $S_{(k)} < 0; x_{1(k)} > 0;$
- Condição 4: $S_{(k)} < 0; x_{1(k)} < 0.$

1. Condição 1:

Admitindo $S_{(k)} > 0$ e $x_{1(k)} > 0$ na Equação 3.14:

$$S_{(k+1)} = (C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}|$$

Considerando a condição de convergência definida pela Equação 3.10, tem-se que: $|S_{(k+1)}| - |S_{(k)}| < 0$, logo,

- (a) Assumindo $S_{(k+1)} > 0$ (o ponto representativo não cruzou a superfície de deslizamento), tem-se:

$$\begin{aligned} (C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| &< 0 \\ (C - a_2 - 1) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| &< 0 \\ (C - a_2 - 1) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| &< b \cdot \alpha \cdot |x_{1(k)}| \end{aligned}$$

$$\alpha > \frac{(C - a_2 - 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} - \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.15)$$

- (b) Assumindo $S_{(k+1)} < 0$ (o ponto representativo cruzou a superfície de deslizamento), tem-se:

$$\begin{aligned} -(C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| &< 0 \\ -(C - a_2 + 1) \cdot |S_{(k)}| + [C \cdot (C - a_2) + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| &< 0 \\ b \cdot \alpha \cdot |x_{1(k)}| &< (C - a_2 + 1) \cdot |S_{(k)}| - [C \cdot (C - a_2) + a_1] \cdot |x_{1(k)}| \end{aligned}$$

$$\alpha < \frac{(C - a_2 + 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} - \frac{1}{b} [C \cdot (C - a_2) + a_1] \quad (3.16)$$

2. Condição 2:

Admitindo $S_{(k)} > 0$ e $x_{1(k)} < 0$ na Equação 3.14:

$$S_{(k+1)} = (C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}|$$

- (a) Assumindo $S_{(k+1)} > 0$ e a Equação 3.10, tem-se:

$$(C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$

$$(C - a_2 - 1) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$(C - a_2 - 1) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| < b \cdot \alpha \cdot |x_{1(k)}|$$

$$\alpha > \frac{(C - a_2 - 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} + \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.17)$$

- (b) Assumindo $S_{(k+1)} < 0$, tem-se:

$$-(C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$

$$-(C - a_2 + 1) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$b \cdot \alpha \cdot |x_{1(k)}| < (C - a_2 + 1) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}|$$

$$\alpha < \frac{(C - a_2 + 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} + \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.18)$$

3. Condição 3:

Admitindo $S_{(k)} < 0$ e $x_{1(k)} > 0$ na Equação 3.14:

$$S_{(k+1)} = -(C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}|$$

- (a) Assumindo $S_{(k+1)} > 0$ (o PR atravessou a superfície de deslizamento), tem-se:
- $$-(C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$
- $$-(C - a_2 + 1) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$\alpha < \frac{(C - a_2 + 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} + \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.19)$$

- (b) Assumindo $S_{(k+1)} < 0$ (o PR não cruzou a superfície de deslizamento), tem-se:
- $$(C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$
- $$(C - a_2 - 1) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$\alpha > \frac{(C - a_2 - 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} + \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.20)$$

4. Condição 4:

Admitindo $S_{(k)} < 0$ e $x_{1(k)} < 0$ na Equação 3.14:

$$S_{(k+1)} = -(C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}|$$

- (a) Assumindo $S_{(k+1)} > 0$ e a inequação $|S_{(k+1)}| - |S_{(k)}| < 0$, tem-se:
- $$-(C - a_2) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$
- $$-(C - a_2 + 1) \cdot |S_{(k)}| + [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| + b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$\alpha < \frac{(C - a_2 + 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} - \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.21)$$

- (b) Assumindo $S_{(k+1)} < 0$, tem-se:
- $$(C - a_2) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| - |S_{(k)}| < 0$$
- $$(C - a_2 - 1) \cdot |S_{(k)}| - [C^2 - a_2 \cdot C + a_1] \cdot |x_{1(k)}| - b \cdot \alpha \cdot |x_{1(k)}| < 0$$

$$\alpha > \frac{(C - a_2 - 1) \cdot |S_{(k)}|}{b \cdot |x_{1(k)}|} - \frac{1}{b} [C^2 - a_2 \cdot C + a_1] \quad (3.22)$$

As condições para o ganho α podem ser sumarizadas em:

$$\frac{(C - a_2 - 1) \cdot |S(k)|}{b \cdot |x_1(k)|} + \frac{1}{b} |a_2 \cdot C - a_1 - C^2| < \alpha < \frac{(C - a_2 + 1) \cdot |S(k)|}{b \cdot |x_1(k)|} + \frac{1}{b} |a_2 \cdot C - a_1 - C^2| \quad (3.23)$$

Observa-se que a condição descrita pela Equação 3.23 é mais restrita do que a condição para sistemas contínuos, descrita pela Equação 3.7 pois, agora, existe um limite mínimo e máximo dentro do qual o ganho do sistema deve permanecer.

3.4 Sistemas Híbridos - SEV e RNAs

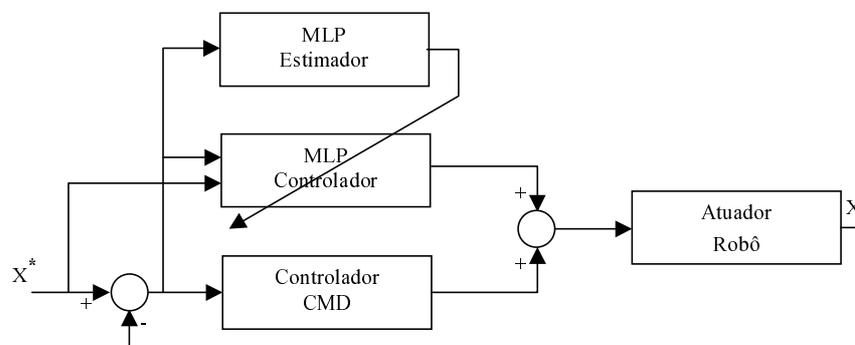


Figura 3.2- Sistema de controle para um robô, acionado por um motor de indução, utilizando RNAs e SEV.

Miran Rodic, Asif Sabanovic e outros [Sabanovic et al., 1996, Rodic et al., 1996] apresentaram, em 1996, um sistema de controle para um robô, acionado por motor de indução, utilizando SEV e RNAs. Em seu trabalho, são utilizadas duas MLP, como mostrado na Figura 3.2. A primeira MLP é utilizada como estimador do sistema a ser controlado, sendo necessária na determinação da superfície de deslizamento e utilizada no treinamento *on-line* da segunda MLP. A segunda MLP é responsável por providenciar a parte do sinal de controle devido às incertezas e não-linearidades da planta. O sinal de controle oriundo do controlador utilizando CMD é, então, acrescido do sinal proveniente da segunda MLP e utilizado no acionamento do motor de indução utilizado no robô. Na Figura 3.2, X^* é o vetor de estados desejado para o sistema.

O treinamento da segunda MLP é realizado utilizando o algoritmo *Backpropagation*

e, devido à presença do controlador utilizando CMD, para a camada de saída da rede, o ajuste dos pesos é determinado como uma função da superfície de deslizamento. Para a camada escondida o treinamento é realizado com o algoritmo *Backpropagation* padrão (Equação 2.9).

Li-Chen Fu, em 1996 [Fu, 1996], utiliza uma MLP como estimador *on-line* de um sistema monovariável genérico para a implementação de um sistema de controle. O estimador é necessário para o desenvolvimento do sistema de controle adaptativo por modelo de referência, baseado em CMD, utilizado na planta. O sistema de treinamento da MLP é, então, derivado diretamente do sistema de controle por modos deslizantes.

3.5 Conclusões

Apresentou-se, neste capítulo, os principais conceitos da teoria de SEV e CMD, utilizando-se, para isto, um sistema de segunda ordem. A análise feita mostra que, considerando-se os limites os quais os ganhos do sistema deve obedecer, um sistema de segunda ordem, utilizando CMD, terá o comportamento de um sistema de primeira ordem no que se refere ao comportamento do erro.

Apesar da topologia MLP não possuir uma dinâmica em sua operação normal (após treinada), durante a etapa do treinamento da rede o algoritmo de treinamento impõe uma dinâmica ao ajuste dos pesos. Esta dinâmica é uma função dos parâmetros da rede e dos parâmetros do algoritmo de treinamento. Desta forma, pode-se utilizar a teoria de CMD para o controle da dinâmica dos pesos, apesar de que esta dinâmica não será diretamente refletida à dinâmica do erro de saída da rede, por se tratar de um sistema com forte acoplamento entre as variáveis de controle.

Os conceitos e metodologias abordados neste capítulo serão utilizados no capítulo subsequente, quando os algoritmos de treinamento da topologia MLP são propostos.

Capítulo 4

Algoritmos Propostos

Neste capítulo é feito o estudo matemático dos algoritmos propostos. São propostos quatro algoritmos: dois para treinamento *off-line* e dois para treinamento *on-line*. A convergência dos algoritmos para o treinamento *off-line* é baseada na análise de sistemas amostrados, enquanto que, para o treinamento *on-line*, a análise é feita considerando tanto sistema contínuo quanto amostrado. Todos os algoritmos são analisados segundo a teoria de sistemas de estrutura variável, mais precisamente, controle por modos deslizantes.

Com a aplicação da teoria de SEV, o problema de treinamento de RNAs passa a ser um problema de controle e as ferramentas de estudo, já bem definidas nesta área de conhecimento, podem ser aplicadas. A Figura 4.1 mostra, em forma de diagrama de blocos, como o problema de treinamento de RNAs pode ser entendido como um problema de controle.

Desta forma, a RNA é a planta a ser controlada e o sistema de controle é o algoritmo de treinamento, o qual atua sobre os pesos da rede para se atingir o sinal de referência, ou seja, a saída desejada da RNA. Assim, utilizando-se a teoria de SEV, pode-se realizar um estudo mais completo da convergência do sistema, definindo-se os ganhos a serem utilizados durante o treinamento.

Teorema 4.1

Seja a superfície definida por $S_{(k)} = -A \cdot X_{(k)} \cdot \text{sgn}(S_{(k)}) + Y_{(k)}$, tal que em regime de deslizamento $S_{(k)} \triangleq 0$. $S_{(k)} : \mathfrak{R}^2 \rightarrow \mathfrak{R}, \{X_{(k)}, Y_{(k)}, A\} \in \mathfrak{R}$, tal que $X_{(k)} > 0$ e $A > 0$ e $\text{sgn}(S_{(k)}) = \begin{cases} +1 & , S_{(k)} \geq 0 \\ -1 & , S_{(k)} < 0 \end{cases}$. Então, um ponto representativo (PR) do sistema, no

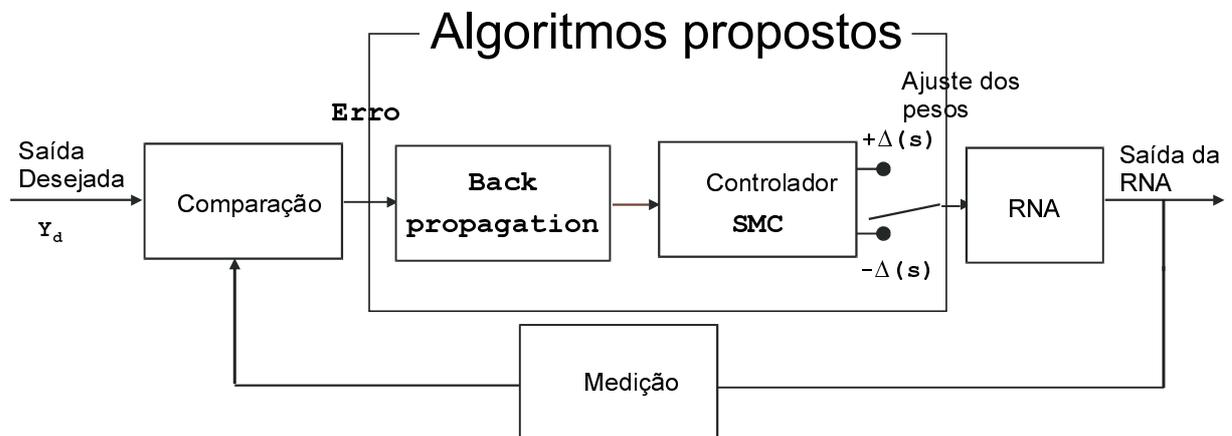


Figura 4.1- Diagrama esquemático utilizado para exemplificar como o treinamento de uma RNA pode ser analisado como um problema de controle.

espaço de estados bidimensional, irá convergir à superfície de deslizamento se:

$$A < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.1)$$

Prova:

Seja a equação:

$$S_{(k)} = -A \cdot X_{(k)} \cdot \text{sgn}(S_{(k)}) + Y_{(k)} \quad (4.2)$$

Analisando a equação de $S_{(k)}$, definida em 4.2, e considerando A como a variável de interesse, pode-se escrever que:

$$|S_{(k)}| \cdot \text{sgn}(S_{(k)}) = -|A \cdot X_{(k)}| \cdot \text{sgn}(S_{(k)}) + Y_{(k)}$$

$$[|S_{(k)}| + |A \cdot X_{(k)}|] \cdot \text{sgn}(S_{(k)}) = Y_{(k)}$$

$$[|S_{(k)}| + |A \cdot X_{(k)}|] \cdot \text{sgn}(S_{(k)}) = |Y_{(k)}| \cdot \text{sgn}(Y_{(k)})$$

$$\therefore \text{sgn}(S_{(k)}) = \text{sgn}(Y_{(k)})$$

portanto,

$$S_{(k)} = [-A \cdot X_{(k)} + |Y_{(k)}|] \cdot \text{sgn}(S_{(k)}) \quad (4.3)$$

A equação acima somente será válida se $A < \frac{|Y_{(k)}|}{X_{(k)}}$, pois o termo entre colchetes tem que ser, necessariamente, maior do que zero para a validação do sinal de $S_{(k)}$, ou seja:

$$|S_{(k)}| \cdot \text{sgn}(S_{(k)}) = [-A \cdot X_{(k)} + |Y_{(k)}|] \cdot \text{sgn}(S_{(k)})$$

$$[-A \cdot X_{(k)} + |Y_{(k)}|] > 0$$

$$-A \cdot X_{(k)} + |Y_{(k)}| > 0$$

$|Y_{(k)}| > A \cdot X_{(k)}$
 donde se conclui que:

$$A < \frac{|Y_{(k)}|}{X_{(k)}} \quad (4.4)$$

A expressão acima estabelece, portanto, uma primeira condição para o valor de A .

A convergência do PR à superfície de deslizamento pode ser analisada considerando-se a convergência da série

$$Z = \sum_{k=1}^{\infty} S_{(k)} \quad (4.5)$$

Uma vez garantida a convergência desta série, garante-se que o PR irá convergir, ao menos assintoticamente, para superfície $S_{(k)} = 0$.

Utilizando-se o princípio de convergência de Cauchy [Kreyszig, 1993]: a série $\{Z_1 + Z_2 + \dots +\}$ converge se e somente se, para um dado valor $\epsilon > 0$, puder ser encontrado um valor de N tal que $|Z_{n+1} + Z_{n+2} + \dots + Z_{n+p}| < \epsilon$ para todo $n > N$ e $p = 1, 2, \dots$. A série será convergente se ela for absolutamente convergente, ou seja, se a série

$$\widehat{Z} = \sum_{k=1}^{\infty} |S_{(k)}| \quad (4.6)$$

for convergente. Para o estudo da convergência da série \widehat{Z} (Equação 4.6), será realizado o teste da razão, ou seja, a série Z é absolutamente convergente se $\left| \frac{S_{(k)}}{S_{(k-1)}} \right| \leq Q < 1$ [Butkov, 1978].

Logo, considerando-se a Equação 4.3 e a condição acima mencionada:

$$\left| \frac{[-A \cdot X_{(k)} + |Y_{(k)}|] \cdot \text{sgn}(S_{(k)})}{[-A \cdot X_{(k-1)} + |Y_{(k-1)}|] \cdot \text{sgn}(S_{(k-1)})} \right| < 1, \text{ portanto,}$$

$\left| \frac{-A \cdot X_{(k)} + |Y_{(k)}|}{-A \cdot X_{(k-1)} + |Y_{(k-1)}|} \right| < 1$. Considerando-se que a condição dada pela Equação 4.4 é satisfeita:

$$-A \cdot X_{(k)} + |Y_{(k)}| < -A \cdot X_{(k-1)} + |Y_{(k-1)}|$$

Logo:

$$A < \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \quad (4.7)$$

Portanto, se a inequação 4.7 for atendida, a série dada pela Equação 4.6 é convergente e, conseqüentemente, a série dada pela Equação 4.5 é absolutamente convergente.

■

Entretanto, considerando o fato que, necessariamente $A > 0$ (condição *sine qua non*), é importante analisar o sinal de A para que a condição 4.7 possa, também, ser atendida.

Multiplicando os termos da desigualdade por $1/A$, tem-se:

$$1 < \frac{|Y_{(k-1)}| - |Y_{(k)}|}{A \cdot (X_{(k-1)} - X_{(k)})}$$

$$1 < \frac{|Y_{(k-1)}| - |Y_{(k)}|}{A \cdot X_{(k-1)} - A \cdot X_{(k)}}$$

Para a análise da expressão acima pode-se reescrever a condição 4.4, da seguinte forma:

$$A < \frac{|Y_{(k)}|}{X_{(k)}}$$

$|Y_{(k)}| > A \cdot X_{(k)}$, ou seja, existe um valor B , tal que $0 \leq B < 1$, e :

$$A \cdot X_{(k)} = B \cdot |Y_{(k)}|$$

logo, utilizando este resultado, tem-se:

$$1 < \frac{|Y_{(k-1)}| - |Y_{(k)}|}{B \cdot |Y_{(k-1)}| - B \cdot |Y_{(k)}|}$$

$$1 < \frac{|Y_{(k-1)}| - |Y_{(k)}|}{B \cdot (|Y_{(k-1)}| - |Y_{(k)}|)}$$

$$1 < \frac{1}{B} \therefore B < 1, \text{ condição esta já estabelecida anteriormente.}$$

Desta forma, pode-se afirmar que a condição 4.7 será sempre maior do que zero.

4.1 Treinamento *Off-line* - 1a. Proposta

No treinamento da RNA MLP com o algoritmo padrão de *Backpropagation*, a direção do ajuste dos pesos é feito considerando-se apenas o sinal do gradiente da função de erro para o peso em questão. Dessa forma, não existe nenhum controle sobre a trajetória do PR da saída da RNA no gráfico de erro. A inclusão de um termo que controle essa trajetória pode significar a escolha de um caminho mais apropriado para o ajuste dos pesos. A primeira proposta para o treinamento *off-line* da RNA [Parma et al., 1998a] baseia-se nessa consideração e introduz o termo de correção oriundo da teoria de sistemas de estrutura variável.

4.1.1 Superfície de Deslizamento

A superfície de deslizamento proposta é uma função do gradiente do erro da saída e de sua taxa de variação, podendo ser escrita como:

- Para a camada de saída:

$$S_{ji(k)} = C \cdot X_{1ji(k)} + X_{2ji(k)} \quad (4.8)$$

onde,

$$X_{1ji(k)} = \frac{\partial E_{(k)}}{\partial W_{ji(k)}} \quad (4.9)$$

$$X_{2ji(k)} = X_{1ji(k)} - X_{1ji(k-1)} \quad (4.10)$$

- Para a camada escondida:

$$S_{ih(k)} = C_H \cdot X_{1ih(k)} + X_{2ih(k)} \quad (4.11)$$

onde,

$$X_{1ih(k)} = \frac{\partial E(k)}{\partial Z_{ih(k)}} \quad (4.12)$$

$$X_{2ih(k)} = X_{1ih(k)} - X_{1ih(k-1)} \quad (4.13)$$

4.1.2 Regras de Atualização dos Pesos

Com base nas equações que definem as superfícies de deslizamento, as seguintes regras de atualização dos pesos são, então, definidas.

- Camada de saída:

$$\Delta W_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \quad (4.14)$$

- Camada Escondida:

$$\Delta Z_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \quad (4.15)$$

Como se observa pelas Equações 4.14 e 4.15, a direção de atualização dos pesos não é fornecida pelo gradiente do erro mas pelas superfícies de deslizamento, definidas pelas Equações 4.8 e 4.11. Os resultados obtidos em simulação mostram que essa nova proposta pode trazer ganhos significativos no treinamento da RNA, quando comparado com o algoritmo padrão de *Backpropagation*.

4.1.3 Estudo da Convergência

Uma vez definida a superfície de deslizamento, e as regras de atualização dos pesos, é necessário fazer o estudo da convergência do PR do sistema no plano determinado pelas variáveis que a definem. Este estudo está melhor detalhado na seção A.4.

4.1.3.1 Camada de saída

Substituindo a Equação 4.9 e A.8 em 4.10 e utilizando a Equação A.5:

$$X_{2ji(k)} = -e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot Y_{Hi(k-1)}$$

Considerando-se a definição A.6 para as variáveis $f'(V_{j(k-1)})$, $Y_{Hi(k-1)}$ e $Y_{j(k)}$

$$X_{2ji(k)} = [-\Delta Y_{dj(k)} + \Delta Y_{j(k)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Utilizando-se a Equação A.15 e 4.14 na expressão acima, obtém-se:

$$X_{2ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Substituindo a Equação 4.9 e a expressão acima determinada em 4.8:

$$S_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \text{sgn}(S_{ji(k)}) - [C \cdot e_{j(k)} + \Delta Y_{dj(k)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Definindo:

$$X_{(k)} = |X_{1ji(k)}| \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \quad (4.16)$$

$$Y_{(k)} = -[C \cdot e_{j(k)} + \Delta Y_{dj(k)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}] \quad (4.17)$$

Logo, pode-se reescrever a equação de $S_{ji(k)}$ obtida anteriormente como:

$$S_{ji(k)} = -\alpha \cdot X_{(k)} \cdot \text{sgn}(S_{ji(k)}) + Y_{(k)}$$

onde $X_{(k)} > 0$ e $\alpha > 0$.

Desta forma, pode-se utilizar a condição 4.1 definida no Teorema (4.1), para se obter o limite da variável α , o qual deve ser obedecido para que a convergência do sistema seja garantida.

$$\alpha < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.18)$$

onde $X_{(k)}$ e $Y_{(k)}$ estão definidos pelas Equações 4.16 e 4.17.

4.1.3.2 Camada escondida

A análise do estudo da convergência para a camada escondida é realizada de forma semelhante à utilizada para a camada de saída.

Desta forma, substituindo-se as Equações 4.12 e A.11 em 4.13 tem-se:

$$X_{2ih(k)} = -f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + f'_H(R_{i(k-1)}) \cdot T_{h(k-1)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Utilizando-se a definição A.6 para as variáveis $f'_H(R_{i(k-1)})$, $T_{h(k-1)}$, e :

$$X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{- [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}]$$

$$\sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

novamente, considerando a definição A.6 para as variáveis $f'(V_{j(k-1)})$ e $W_{ji(k-1)}$:

$$\begin{aligned} X_{2ih(k)} = & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-Y_{dj(k)} + Y_{j(k)} + \\ & Y_{dj(k-1)} - Y_{j(k-1)})] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot \\ & W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot \\ & f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

Substituindo a Equação A.16 e 4.15 na equação anterior e considerando-se a Equação A.9,

$$\begin{aligned} X_{2ih(k)} = & -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + \\ & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \\ & f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot \\ & T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}] \end{aligned}$$

Substituindo a expressão acima determinada e a Equação 4.12 na Equação 4.11, obtém-se:

$$\begin{aligned} S_{ih(k)} = & -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + \\ & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot \\ & (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \\ & \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}] \end{aligned}$$

Definindo:

$$X_{(k)} = |X_{1ih(k)}| \cdot [f'_H (R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f' (V_{j(k)}) \cdot W_{ji(k)}]^2 \quad (4.19)$$

$$Y_{(k)} = f'_H (R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \left\{ [-\Delta Y_{dj(k)} \cdot f' (V_{j(k)}) \cdot W_{ji(k)}] \right. \\ \left. - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f' (V_{j(k)}) \cdot \Delta W_{ji(k)} - f' (V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f' (V_{j(k)}) \cdot W_{ji(k)})] \right\} \\ + [\Delta f'_H (R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H (R_{i(k)}) \cdot T_{h(k)} - f'_H (R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}] \quad (4.20)$$

então, a equação para $S_{ih(k)}$ pode ser re-escrita como:

$$S_{ih(k)} = -\beta \cdot X_{(k)} \cdot \text{sgn} (S_{ih(k)}) + Y_{(k)}$$

onde $X_{(k)} > 0$ e $\beta > 0$

Considerando a condição 4.1 definida no Teorema (4.1), pode-se, agora, estabelecer o limite para a variável β , de forma a garantir a convergência do sistema.

$$\beta < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.21)$$

onde as variáveis $X_{(k)}$ e $Y_{(k)}$ estão definidas pelas Equações 4.19 e 4.20.

Desta forma, a convergência do treinamento da RNA MLP estará garantida para as superfícies de deslizamento definidas pelas Equações 4.8 e 4.11 se os limites para os ganhos de treinamento definidos em 4.18 e 4.21 forem observados.

4.2 Treinamento *Off-line* - 2a. Proposta

Na proposta anteriormente apresentada, é introduzida uma superfície de deslizamento com o objetivo de possibilitar um controle na trajetória dos pesos durante o treinamento.

Entretanto, o módulo do ajuste continua sendo da mesma ordem de grandeza do ajuste realizado no algoritmo padrão de *Backpropagation*. Sabe-se, porém, que o módulo do ajuste baseado apenas no gradiente local pode acarretar um grande tempo para a convergência do treinamento. Para entender este fato, basta imaginar uma superfície de treinamento muito plana. Neste caso, o algoritmo de treinamento ficará muito tempo percorrendo essa superfície em pequenos passos pois o módulo do gradiente do erro será pequeno.

Deste modo, introduziu-se uma variação na determinação da superfície de deslizamento e, conseqüentemente, nas regras de atualização dos pesos. Esta alteração proporciona uma aceleração ao treinamento da RNA.

4.2.1 Superfície de Deslizamento

A superfície de deslizamento proposta é, agora, não somente uma função do gradiente do erro da saída e de sua taxa de variação, mas sim do gradiente do erro de saída e de seu somatório durante uma *epoch*. Pode-se escrever a superfície de deslizamento da seguinte maneira:

- Para a camada de saída:

$$S_{ji(k)} = C \cdot X_{1ji(k)} + X_{2ji(k)} \quad (4.22)$$

onde,

$$X_{1ji(k)} = \sum_{l=1}^k \frac{\partial E_{(l)}}{\partial W_{ji(l)}} \quad (4.23)$$

$$X_{2ji(k)} = \frac{\partial E_{(k)}}{\partial W_{ji(k)}} \quad (4.24)$$

$$X_{1ji(k)} = 0, \forall k < 1$$

- Para a camada escondida:

$$S_{ih(k)} = C_H \cdot X_{1ih(k)} + X_{2ih(k)} \quad (4.25)$$

onde,

$$X_{1ih(k)} = \sum_{l=1}^k \frac{\partial E(l)}{\partial Z_{ih(l)}} \quad (4.26)$$

$$X_{2ih(k)} = \frac{\partial E(k)}{\partial Z_{ih(k)}} \quad (4.27)$$

$X_{2ih(k)} = 0, \forall k < 1$, onde k representa o vetor apresentado dentro de uma *epoch*.

4.2.2 Regras de Atualização dos Pesos

Com base nas equações que definem as superfícies de deslizamento, as seguintes regras de atualização dos pesos são, então, definidas:

- Camada de saída:

$$\Delta W_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \quad (4.28)$$

- Camada escondida:

$$\Delta Z_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot f'_H(R_{i(k)}) \quad (4.29)$$

Como pode ser observado, através das Equações 4.23, 4.26, 4.28 e 4.29, o módulo de atualização dos pesos é uma função do somatório de todos os gradientes, desde o início da apresentação da *epoch* até o instante presente. Desta forma, a interferência destrutiva presente no algoritmo tradicional de *Backpropagation* (interferência existente devido à apresentação consecutiva de vetores dentro da *epoch*) é minimizada, pois o algoritmo de treinamento mantém uma memória de todas as atualizações realizadas até o momento atual. Da mesma forma, para compor a superfície de deslizamento, o gradiente

do erro é levado em consideração para a determinação do sentido de atualização dos pesos, possibilitando um controle da trajetória dos pesos na superfície de erro.

Para evitar a saturação prematura dos nodos da camada escondida, que ocorreria quando o valor de $|X_{1ih(k)}|$ fosse elevado, introduziu-se o termo $f'_H(R_{i(k)})$ não presente na Equação 4.15. São dois os motivos principais que acarretariam em um elevado valor de $|X_{1ih(k)}|$:

1. elevado erro entre a saída desejada e a saída real da rede, nas primeiras *epochs*, devido à inicialização randômica dos parâmetros da rede;
2. elevado valor do somatório dos gradientes do erro, em um treinamento com muitos vetores de treinamento, nas primeiras *epochs*.

Tais fatos são mais marcantes nas camadas escondidas pelo fato do gradiente do erro, para um nodo da camada escondida, ser uma função do somatório dos gradientes locais do erro da camada de saída multiplicado pelos pesos que conectam as camadas.

4.2.3 Estudo da Convergência

De maneira semelhante àquela mostrada na seção 4.1, é necessário proceder com o estudo da convergência do treinamento para que sejam conhecidos os limites para os ganhos α e β . Este estudo está melhor detalhado na seção A.5.

4.2.3.1 Camada de saída

Substituindo-se as Equações 4.23 e 4.24 em 4.22 e utilizando-se a Equação A.8, tem-se:

$$S_{ji(k)} = (C + 1) \cdot [-e_{(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

considerando a Equação A.5 e a definição A.6 para a variável $Y_{j(k)}$,

$$S_{ji(k)} = (C + 1) \cdot \Delta Y_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} - (C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

usando a Equação A.15 e 4.28:

$$S_{ji(k)} = -\alpha \cdot \text{sgn}(S_{ji(k)}) \cdot (C + 1) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \left| \sum_{l=1}^k \frac{\partial E(l)}{\partial W_{ji(l)}} \right| - (C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

definindo:

$$X_{(k)} = (C + 1) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \left| \sum_{l=1}^k \frac{\partial E(l)}{\partial W_{ji(l)}} \right| \quad (4.30)$$

$$Y_{(k)} = -(C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}} \quad (4.31)$$

Pode-se re-escrever a equação de $S_{ji(k)}$ da seguinte forma:

$$S_{ji(k)} = -\alpha \cdot \text{sgn}(S_{ji(k)}) \cdot X_{(k)} + Y_{(k)}$$

onde $X_{(k)} > 0$ e $\alpha > 0$.

Novamente considerando a condição 4.1, definida no Teorema (4.1), a convergência do sistema estará garantida se:

$$\alpha < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.32)$$

4.2.3.2 Camada escondida

O estudo da convergência prossegue de maneira análoga à utilizada para a camada de

saída, assim sendo, substituindo a Equação 4.26 e 4.27 em 4.25 e considerando-se a Equação A.11:

$$S_{ih(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial Z_{ih(l)}}$$

considerando, agora, as Equações A.5 e A.6 para a variável $Y_{j(k)}$:

$$S_{ih(k)} = (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [\Delta Y_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial Z_{ih(l)}}$$

Substituindo-se a Equação A.16, e posteriormente a Equação 4.29, na expressão acima, tem-se:

$$S_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \operatorname{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot (C_H + 1) \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 - (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial Z_{ih(l)}}$$

Definindo:

$$X_{(k)} = |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot (C_H + 1) \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 \quad (4.33)$$

$$Y_{(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial Z_{ih(l)}} \quad (4.34)$$

Então, pode-se reescrever $S_{ih(k)}$ da seguinte maneira:

$$S_{ih(k)} = -\beta \cdot \operatorname{sgn}(S_{ih(k)}) \cdot X_{(k)} + Y_{(k)}$$

É necessário, neste ponto, uma explicação: mesmo existindo o fator $f'_H(R_{i(k)})$ na expressão de $X_{(k)}$ (Equação 4.33), o qual poderia, a princípio, ser negativo, o valor de $X_{(k)}$ será sempre positivo se considerarmos as funções de ativação dos nodos da RNA crescentes, como é de costume. Neste caso, $f'_H(R_{i(k)})$ é sempre positivo e pode-se afirmar que $X_{(k)} > 0$ podendo, pois, utilizar a condição 4.1, mostrada no Teorema (4.1), a qual garante a convergência do sistema se:

$$\beta < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.35)$$

Desta forma, a convergência do treinamento da RNA MLP será garantida para as superfícies de deslizamento definidas em 4.22 e 4.25, considerando as regras de atualização dos pesos definidas em 4.28 e 4.29, se os ganhos α e β obedecerem os limites determinados em 4.32 e 4.35.

4.3 Treinamento *On-line* - 1a. proposta

Diferentemente do treinamento *off-line*, o treinamento *on-line* possui características específicas que devem ser consideradas durante a análise de um algoritmo que se proponha a lidar nesse campo de treinamento. A sua utilização possibilita um contínuo ajuste dos parâmetros da RNA às necessidades do sistema na qual ela está inserida, sendo, desta maneira, bastante atraente em sistemas de controle.

O treinamento *on-line* possui, dentre outras, as seguintes características:

1. não se possui, a priori, os vetores de dados com os quais a RNA será treinada;
2. os vetores de treinamento não são, na sua maioria, armazenados;
3. o fator tempo tem que ser considerado como elemento seqüenciador dos vetores de treinamento, os quais eram, no treinamento *off-line*, apresentados de forma randômica;

Deve-se levar em consideração, no treinamento *on-line*, o efeito destrutivo sobre o prévio treinamento da RNA durante a apresentação contínua de um mesmo vetor de treinamento, fato que pode ocorrer durante uma operação em regime do sistema. O algoritmo de treinamento deve ser pequeno e eficiente, possibilitando um treinamento rápido e sem alto custo computacional.

Com a utilização da teoria de sistemas de estrutura variável [Utkin, 1978], espera-se atingir as metas citadas anteriormente, ou seja, um algoritmo robusto e eficiente para o treinamento *on-line* de RNA [Parma et al., 1998b][Parma et al., 1997a].

4.3.1 Superfície de Deslizamento

- Para a camada de saída:

$$S_{ji(t)} = C \cdot X_{1ji(t)} + X_{2ji(t)} \quad (4.36)$$

onde,

$$X_{1ji(t)} = e_{j(t)} \quad (4.37)$$

$$X_{2ji(t)} = \frac{\partial X_{1ji(t)}}{\partial t} \quad (4.38)$$

- Para a camada escondida:

$$S_{ih(t)} = C_H \cdot X_{1ih(t)} + X_{2ih(t)} \quad (4.39)$$

onde,

$$X_{1ih(t)} = \frac{1}{2} \cdot \sum_{j=1}^p [e_{j(t)}]^2 \quad (4.40)$$

$$X_{2ih(t)} = \frac{\partial X_{1ih(t)}}{\partial t} \quad (4.41)$$

4.3.2 Regras de Atualização dos Pesos

Com base nas equações que definem as superfícies de deslizamento, as seguintes regras de atualização dos pesos são, então, definidas.

- Camada de saída:

$$\dot{W}_{ji(t)} = \frac{\alpha \cdot |X_{1ji(t)}| \cdot \text{sgn}(S_{ji(t)}) \cdot Y_{Hi(t)}}{f'(V_{j(k)})} \quad (4.42)$$

- Camada Escondida:

$$\dot{Z}_{ih(t)} = \frac{\beta \cdot |X_{1ih(t)}| \cdot \text{sgn}(S_{ih(t)}) \cdot T_{h(k)}}{f'_H(R_{i(k)}) \cdot \sum_{j=1}^p [(Y_{dj(t)} - Y_{j(t)}) \cdot f'(V_{j(t)}) \cdot W_{ji(t)}]} \quad (4.43)$$

4.3.3 Estudo da Convergência

De maneira semelhante à análise desenvolvida para os algoritmos de treinamento *off-line*, é necessário estudar os limites dos ganhos envolvidos no treinamento, de forma a garantir a convergência e estabilidade do sistema proposto. Por se tratar de treinamento *on-line*, no qual a variável tempo está presente, a condição para a convergência a ser analisada é descrita pela Equação 4.44[Utkin, 1978], a qual é uma consequência direta da análise de estabilidade de Lyapunov. O estudo dos limites dos ganhos envolvidos no treinamento está melhor detalhado na seção A.6.

$$S \cdot \dot{S} < 0 \quad (4.44)$$

4.3.3.1 Camada de saída

Substituindo-se a Equação 4.37 em 4.38, efetuando-se a derivada e utilizando uma expressão análoga à Equação A.15, para o tempo contínuo, $\frac{\partial Y_{j(t)}}{\partial t} = f'(V_{j(k)}) \cdot Y_{Hi(k)} \cdot$

$\dot{W}_{ji(k)}$, em conjunto com a Equação 4.42, para a determinação de $X_{2ji(t)}$ obtém-se:

$$X_{2ji(t)} = \frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \text{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \quad (4.45)$$

Substituindo a Equação 4.45 em 4.36 ,

$$S_{ji(t)} = C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \text{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2$$

$$\text{sgn}\left(C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t}\right) = \text{sgn}(S_{ji(t)})$$

$$S_{ji(t)} = \left[\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right| - \alpha \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right] \cdot \text{sgn}(S_{ji(t)}) \quad (4.46)$$

Logo, para que a expressão acima mencionada seja válida:

$$\alpha < \frac{\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right|}{|X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2} \quad (4.47)$$

Derivando a Equação 4.45,

$$\frac{\partial X_{2ji(t)}}{\partial t} = \frac{\partial}{\partial t} \left[\frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \text{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right]$$

$$\frac{\partial X_{2ji(t)}}{\partial t} = \frac{\partial^2 Y_{dj(t)}}{\partial t^2} - \alpha \cdot \text{sgn}(S_{ji(t)}) \cdot [Y_{Hi(t)}]^2 \cdot \left[\frac{\partial |X_{1ji(t)}|}{\partial t} + 2 \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \right] \quad (4.48)$$

Derivando a Equação 4.36 e utilizando a Equação 4.38, para que se possa prosseguir na análise da condição 4.44,

$$\frac{\partial S_{ji(t)}}{\partial t} = C \cdot X_{2ji(t)} + \frac{\partial X_{2ji(t)}}{\partial t}$$

Substituindo as Equações 4.45 e 4.48 na expressão acima,

$$\dot{S}_{ji(t)} = -\alpha \cdot \text{sgn}(S_{ji(t)}) \cdot [Y_{Hi(t)}]^2 \cdot \left\{ 2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right\} + \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t}$$

Considerando a Equação 4.44, na convergência do sistema, $\text{sgn}(S_{ji(t)}) = -\text{sgn}(\dot{S}_{ji(t)})$, desta forma pode-se escrever:

$$\dot{S}_{ji(t)} = \left\{ \alpha \cdot [Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right] + \left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right| \right\} \cdot \text{sgn}(\dot{S}_{ji(t)})$$

Analisando a expressão acima, tem-se, necessariamente, um segundo limite para o ganho α

$$\alpha > - \frac{\left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right|}{[Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right]} \quad (4.49)$$

Portanto, para a convergência do sistema é necessário que o ganho α atenda às condições impostas pelas Equações 4.47 e 4.49, logo,

$$\begin{aligned} & - \frac{\left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right|}{[Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right]} < \alpha \\ & < \frac{\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right|}{|X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2} \end{aligned} \quad (4.50)$$

4.3.3.2 Camada escondida

Substituindo a Equação 4.40 em 4.41 e derivando,

$$X_{2ih(t)} = \sum_{j=1}^p e_{j(t)} \cdot \left[\frac{\partial Y_{dj(t)}}{\partial t} - \frac{\partial Y_{j(t)}}{\partial t} \right]$$

Considerando a expressão análoga à Equação A.16 para tempo contínuo,

$$\frac{\partial Y_{j(k)}}{\partial t} = f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t}$$

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ e_{j(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t} \cdot \sum_{j=1}^p \left\{ e_{j(t)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)} \right\}$$

Utilizando as Equações 4.43 e 4.37

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \quad (4.51)$$

Utilizando a expressão acima e a Equação 4.40 na expressão 4.39 e Substituindo a Equação 4.37,

$$S_{ih(t)} = \frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

Prosseguindo de maneira análoga à utilizada durante a análise da convergência para a camada de saída tem-se:

$$\operatorname{sgn}(S_{ih(t)}) = \operatorname{sgn} \left(\frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} \right)$$

$$S_{ih(t)} = \left\{ \left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right| - \beta \cdot |X_{1ih(t)}| \cdot [T_{h(k)}]^2 \right\} \cdot \operatorname{sgn}(S_{ih(t)})$$

Conseqüentemente,

$$\left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right| - \beta \cdot |X_{1ih(t)}| \cdot [T_{h(k)}]^2 > 0$$

$$\beta < \frac{\left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right|}{|X_{1ih(t)}| \cdot [T_{h(k)}]^2} \quad (4.52)$$

Derivando a expressão 4.39,

$$\dot{S}_{ih(t)} = C_H \cdot X_{2ih(t)} + \frac{\partial}{\partial t} X_{2ih(t)}$$

Para a determinação da parcela $\frac{\partial}{\partial t} X_{2ih(t)}$, a expressão 4.51 é derivada, obtendo:

$$\frac{\partial X_{2ih(t)}}{\partial t} = \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} - \beta \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \left[2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right]$$

Substituindo a expressão acima determinada, juntamente com a Equação 4.51 na expressão de $\dot{S}_{ih(t)}$,

$$\dot{S}_{ih(t)} = -\beta \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\}$$

Considerando a Equação 4.44, na convergência do sistema, necessariamente, $\operatorname{sgn}(S_{ih(t)}) = -\operatorname{sgn}(\dot{S}_{ih(t)})$, conseqüentemente,

$$\operatorname{sgn}(\dot{S}_{ih(t)}) = \operatorname{sgn} \left(\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right)$$

$$\dot{S}_{ih(t)} = \left\{ \beta \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right\} \cdot \text{sgn} \left(\dot{S}_{ih(t)} \right)$$

e, conseqüentemente,

$$\beta > - \frac{\left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right|}{[T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right]} \quad (4.53)$$

Desta forma, para garantir a convergência do sistema, o ganho β tem que atender, necessariamente as condições impostas pelas Equações 4.52 e 4.53, ou seja:

$$\begin{aligned} & \left| \frac{\frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right]}{|X_{1ih(t)}| \cdot [T_{h(k)}]^2} \right| > \beta \\ & > - \frac{\left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right|}{[T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right]} \quad (4.54) \end{aligned}$$

Pode-se dizer, então, que o treinamento irá convergir se as inequações 4.50 e 4.54 forem obedecidas.

4.4 Treinamento *On-line* - 2a. proposta

A segunda proposta para treinamento *on-line* é, basicamente, a versão *on-line*, considerando a variável tempo, para a primeira proposta de treinamento *off-line* (seção 4.1). Por se tratar da implementação em sistemas amostrados, o algoritmo foi desenvolvido considerando o período de amostragem T , sendo que a discretização das derivadas em

tempo contínuo foi realizada utilizando a regra de Euler.

Observou-se que, na primeira proposta de treinamento *on-line*, a presença da derivada das funções de ativação no denominador das equações de atualização dos pesos (Equação 4.42 e 4.43) pode, em casos de saturação do neurônio, provocar problemas de instabilidade numérica ao algoritmo. Este problema pode ser facilmente contornado evitando-se a saturação do neurônio ou, então, impondo um limite máximo para a atualização dos pesos. A saturação de um determinado neurônio, principalmente na camada escondida, pode evidenciar uma não adequação da estrutura da RNA ao problema a ela proposto, ou seja, um número pequeno de neurônios na camada escondida.

Optou-se, então, por estudar um algoritmo de treinamento *on-line* que não apresentasse este inconveniente. O algoritmo proposto foi utilizado no desenvolvimento de um observador neural de fluxo estatístico de um motor de indução [Parma et al., 1998c].

Desta forma, baseando-se na primeira proposta de treinamento *off-line*, as seguintes superfícies de erro são escolhidas:

- Para a camada de saída:

$$S_{ji(k)} = C \cdot X_{1ji(k)} + X_{2ji(k)} \quad (4.55)$$

onde,

$$X_{1ji(k)} = \frac{\partial E(k)}{\partial W_{ji(k)}} \quad (4.56)$$

$$X_{2ji(k)} = \frac{X_{1ji(k)} - X_{1ji(k-1)}}{T} \quad (4.57)$$

- Para a camada escondida:

$$S_{ih(k)} = C_H \cdot X_{1ih(k)} + X_{2ih(k)} \quad (4.58)$$

onde,

$$X_{1ih(k)} = \frac{\partial E(k)}{\partial Z_{ih(k)}} \quad (4.59)$$

$$X_{2ih(k)} = \frac{X_{1ih(k)} - X_{1ih(k-1)}}{T} \quad (4.60)$$

4.4.1 Regras de Atualização dos Pesos

Com base nas equações que definem as superfícies de deslizamento, as seguintes regras de atualização dos pesos são, então, definidas.

- Camada de saída:

$$\Delta W_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \quad (4.61)$$

- Camada Escondida:

$$\Delta Z_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \quad (4.62)$$

A atualização dos pesos é, então, calculada considerando-se, também, a regra de Euler para a integração, ou seja,

- Camada de saída:

$$W_{ji(k+1)} = W_{ji(k)} + T \cdot \Delta W_{ji(k)} \quad (4.63)$$

- Camada Escondida:

$$Z_{ih(k+1)} = Z_{ih(k)} + T \cdot \Delta Z_{ih(k)} \quad (4.64)$$

Como pode-se observar, não existe nenhum termo no denominador das Equações 4.61 e 4.62, evitando-se, desta forma, os problemas numéricos mencionados anteriormente. De forma similar à seção 4.1, a direção de atualização dos pesos não é fornecida pelo gradiente do erro mas pela superfície de deslizamento, definidas pelas Equações 4.55 e 4.58.

4.4.2 Estudo da Convergência

O estudo da convergência do algoritmo é realizado para a camada de saída e camada escondida, independentemente, considerando-se a condição de convergência dada por $|S_{(k)}| < |S_{(k-1)}|$, a qual é equivalente à condição descrita pela Equação 3.10. De forma semelhante ao estudo da convergência dos algoritmos previamente descritos, o estudo dos limites dos ganhos envolvidos no treinamento deste algoritmo está melhor detalhado na seção A.7.

4.4.2.1 Camada de saída

Substituindo as Equações 4.56 e A.8 em 4.57 e considerando-se a definição A.6:

$$T \cdot X_{2ji(k)} = [-Y_{dj(k)} + Y_{dj(k-1)} + Y_{j(k)} - Y_{j(k-1)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Utilizando a definição A.6 para a variável $Y_{j(k)}$, e as Equações A.15 e 4.61

$$T \cdot X_{2ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Substituindo a Equação 4.56 e a expressão acima determinada em 4.55:

$$S_{ji(k)} = -\frac{1}{T} \cdot \alpha \cdot |X_{1ji(k)}| \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \text{sgn}(S_{ji(k)}) - \left[C \cdot e_{j(k)} + \frac{\Delta Y_{dj(k)}}{T} \right] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + \frac{1}{T} \cdot e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Definindo:

$$X_{(k)} = \frac{|X_{1ji(k)}| \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2}{T} \quad (4.65)$$

$$Y_{(k)} = - \left[C \cdot e_{j(k)} + \frac{\Delta Y_{dj(k)}}{T} \right] \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + \frac{1}{T} \cdot e_{j(k-1)} \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}] \quad (4.66)$$

Logo, pode-se reescrever a equação de $S_{ji(k)}$ obtida anteriormente como:

$$S_{ji(k)} = -\alpha \cdot X_{(k)} \cdot \operatorname{sgn} (S_{ji(k)}) + Y_{(k)}$$

onde $X_{(k)} > 0$ e $\alpha > 0$.

Desta forma, pode-se utilizar a condição 4.1 definida no Teorema (4.1), para se obter o limite da variável α , o qual deve ser obedecido para que a convergência do sistema seja garantida.

$$\alpha < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.67)$$

onde $X_{(k)}$ e $Y_{(k)}$ estão definidos pelas Equações 4.65 e 4.66.

4.4.2.2 Camada escondida

A análise do estudo da convergência para a camada escondida é realizada de forma semelhante à utilizada para a camada de saída.

Desta forma, substituindo-se as Equações 4.59 e A.11 em 4.60 tem-se:

$$T \cdot X_{2ih(k)} = -f'_H (R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k)}) \cdot f' (V_{j(k)}) \cdot W_{ji(k)}] + f'_H (R_{i(k-1)}) \cdot T_{h(k-1)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f' (V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Utilizando-se a definição A.14 para as variáveis $f'_H (R_{i(k-1)})$ e $T_{h(k-1)}$:

$$T \cdot X_{2ih(k)} = f'_H (R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{- [e_{j(k)} \cdot f' (V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot$$

$$f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

novamente, considerando a definição A.14 para as variáveis $f'(V_{j(k-1)})$ e $W_{ji(k-1)}$ e a definição dada pela Equação A.6:

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-\Delta Y_{dj(k)} + \Delta Y_{j(k)})] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Substituindo a Equação A.16 e 4.62 na equação acima,

$$T \cdot X_{2ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Substituindo a expressão acima determinada e a Equação 4.12 na Equação 4.11 e considerando-se a Equação A.9, obtém-se:

$$S_{ih(k)} = -\frac{1}{T} \cdot \beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \left\{ \frac{1}{T} \cdot [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + \frac{1}{T} \cdot [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \right\} + \frac{1}{T} \cdot [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}]$$

Definindo:

$$X_{(k)} = \frac{1}{T} \cdot |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 \quad (4.68)$$

$$\begin{aligned} Y_{(k)} = & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \left\{ \frac{1}{T} \cdot [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] \right. \\ & - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + \frac{1}{T} \cdot [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} \\ & - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \left. \right\} + \frac{1}{T} \cdot [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} \\ & - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}] \end{aligned} \quad (4.69)$$

então, a equação para $S_{ih(k)}$ pode ser re-escrita como:

$$S_{ih(k)} = -\beta \cdot X_{(k)} \cdot \text{sgn}(S_{ih(k)}) + Y_{(k)}$$

onde $X_{(k)} > 0$ e $\beta > 0$

Considerando a condição 4.1 definida no Teorema (4.1), pode-se, agora, estabelecer o limite para a variável β , de forma a garantir a convergência do sistema.

$$\beta < \min \left\{ \frac{|Y_{(k)}|}{X_{(k)}}, \frac{|Y_{(k-1)}| - |Y_{(k)}|}{X_{(k-1)} - X_{(k)}} \right\} \quad (4.70)$$

onde as variáveis $X_{(k)}$ e $Y_{(k)}$ estão definidas pelas Equações 4.68 e 4.69.

Desta forma, a convergência do treinamento da RNA MLP estará garantida para as superfícies de deslizamento definidas pelas Equações 4.55 e 4.58 se os limites para os ganhos de treinamento definidos em 4.67 e 4.70 forem observados e se não existirem perturbações, em relação à dinâmica de segunda ordem proposta, que forem acima do limiar de robustez.

4.5 Conclusões

Neste capítulo foram apresentadas quatro propostas de treinamento para MLPs. Duas

propostas para treinamento *off-line* e duas propostas para treinamento *on-line*.

Foi apresentada e utilizada uma metodologia para o desenvolvimento e análise dos algoritmos propostos utilizando a teoria de Sistemas de Estrutura Variável. Esta teoria, utilizada em sistemas de controle, possibilita analisar matematicamente os algoritmos de treinamento, possibilitando determinar os limites dos ganhos envolvidos nos algoritmos. Esta análise matemática é uma importante contribuição para a área de RNAs pois os ganhos necessários pelos algoritmos clássicos de treinamento são, em sua maioria, determinados empiricamente, não existindo uma análise matemática de seus limites.

A performance dos algoritmos propostos para treinamento *off-line* será, no capítulo 6, comparada à performance dos algoritmos clássicos encontrados na literatura em problemas clássicos de RNAs: aproximação de funções e classificação. Os algoritmos propostos para treinamento *on-line* serão utilizados, no capítulo 6, para o desenvolvimento de uma proposta de acionamento para motores de indução.

Capítulo 5

Bancada Experimental

Durante o trabalho desta tese, desenvolveu-se uma bancada experimental, composta de um conjunto motor de indução/motor de corrente contínua e um inversor trifásico utilizando IGBTs. Nesta bancada foram implementados sistemas de medição redundantes, possibilitando a comparação entre diversas técnicas de medição de grandezas necessárias ao controle de motores de indução. Esta bancada é, por si só, uma contribuição deste trabalho no estudo comparativo entre diversas técnicas de medição propostas na literatura.

Neste capítulo será feita uma descrição dos elementos envolvidos nesta bancada.

5.1 Inversor

Os circuitos referentes ao inversor estão mostrados no Apêndice B. A Figura 5.1 mostra a visão geral do inversor desenvolvido.

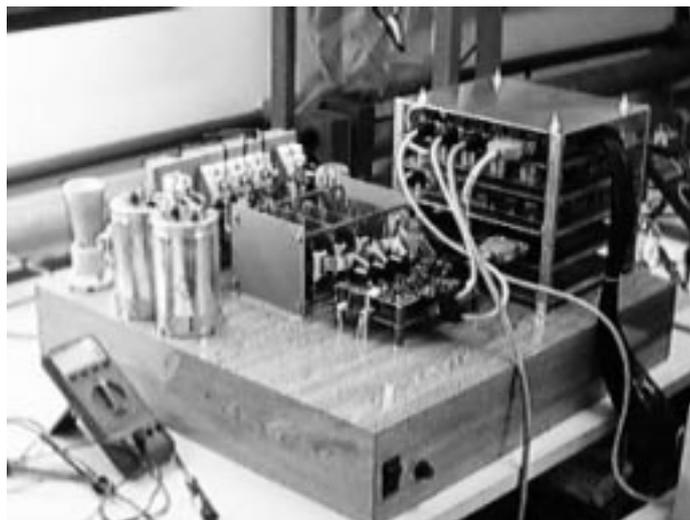


Figura 5.1- Visão geral do inversor desenvolvido.

Desenvolveu-se um inversor trifásico não regenerativo, ou seja, utilizando, na entrada de potência, um retificador trifásico a diodos (Figura B.1). O inversor é composto de três módulos contendo, cada um, dois IGBTs modelo SKM 40GB, mais um módulo contendo apenas um IGBT, modelo SKM 40GA, todos os módulos da *Semikron Conductors*. Optou-se por não desenvolver as placas de comando dos IGBTs, tendo sido adquiridas três placas modelo SKHI23 (Apêndice E) e uma modelo SKHI10, para os IGBTs do inversor e do chopper, respectivamente. Todas as placas de comando foram desenvolvidas, também, pela *Semikron*. A Figura 5.2 mostra a parte de potência do inversor.



Figura 5.2- Módulo de potência utilizado pelo inversor trifásico.

Para o desenvolvimento do sistema de acionamento a ser implementado, tornou-se necessário o desenvolvimento de circuitos analógicos para a interface entre o sistema a ser controlado e o microcomputador no qual será implementado o controle. Após diversas versões, optou-se por desenvolver placas separadas, dedicadas a cada uma das funções a serem implementadas (Figura 5.3), e todas conectadas a uma única placa, responsável pela interface entre o inversor e a placa de aquisição de dados.

O inversor desenvolvido possui as seguintes medições:

1. Medição de tensão
 - (a) medição analógica através de filtros passa-baixa;

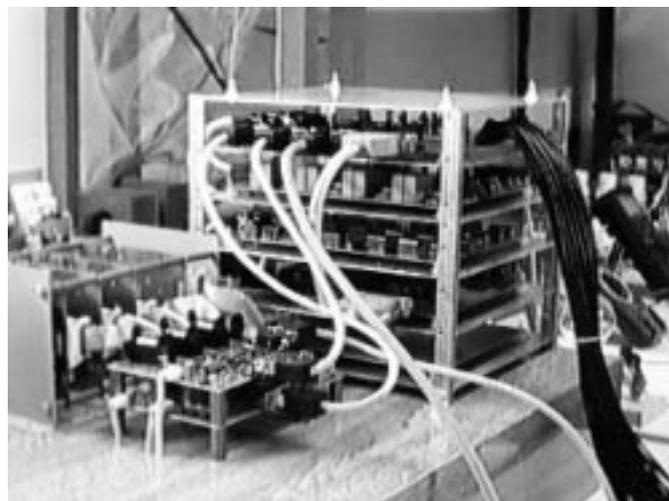


Figura 5.3- Placas de medição desenvolvidas.

-
- (b) medição analógica pela integração dos pulsos de PWM ([Chin et al., 1998]);
 - (c) medição digital
2. Medição de corrente
 - (a) medição através de filtros passa-baixa;
 - (b) medição direta das correntes (sem filtragem);
 3. Medição de velocidade
 - (a) medição analógica através do tacogerador;
 - (b) medição digital através do encoder;
 - (I) medição por pulso;
 - (II) medição por tempo;
 4. Medição de fluxo
 - (a) medição através de bobinas exploradoras;
 - (b) medição através da tensão parcial nas bobinas das fases;
 5. Medição de conjugado
 - (a) medição através de torquímetro rotacional

São disponibilizadas tanto as grandezas trifásicas quanto as grandezas convertidas para o sistema $\alpha\beta$, fixo no estator, para as medições analógicas de tensão e correntes.

O acionamento dos IGBTs pode ser realizado de duas formas distintas:

1. Sinal de controle proveniente de uma placa qualquer, como, por exemplo, uma placa dedicada à geração de PWM;
2. sinal de controle proveniente de um circuito de histerese existente no próprio inversor.

As placas desenvolvidas implementam, basicamente, as seguintes funções:

1. Circuito de Start-up e chopper (Figura B.14);
 2. Circuito de interface de comandos (Figura B.12);
 3. Circuito de histerese de corrente (Figura B.5);
 4. Circuito de aquisição de corrente e tensão (Figura B.13);
-

5. Circuito de medição analógica de correntes e tensões (Figuras B.9, B.10 e B.11);
6. Circuito de medições digitais de tensão e velocidade (Figura B.4);
7. Circuito de medição de fluxo e conjugado (Figuras B.6, B.7 e B.8);
8. Circuito de medição do tacogerador (Figura B.15);
9. Circuito de interface (Figuras B.2 e B.3);

As placas foram desenvolvidas utilizando-se tanto tecnologia CMOS quanto TTL, além de amplificadores operacionais para as medições analógicas. Desta forma, para o correto funcionamento do sistema, é necessária uma alimentação de $\pm 15V$ e $\pm 5V$.

Considerando-se as máquinas utilizadas no Laboratório de Conversão e Controle de Energia do Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica (UFMG), os circuitos de medição de corrente e tensão foram projetados considerando-se a máquina de corrente alternada ligada em triângulo e disponibilizando na saída dos circuitos as grandezas por fase.

Os ganhos dos circuitos de condicionamento dos sinais analógicos foram calculados considerando-se a máxima amplitude dos sinais para a amostragem, pela placa de aquisição de dados, igual a $\pm 5V$.

5.2 Circuito de Start-up e Chopper

Este circuito é responsável pelos comandos de ligar e desligar o inversor bem como pelo gerenciamento do chopper de frenagem, responsável pela limitação da tensão do barramento (Figura B.14).

O inversor pode ser ligado/desligado através de duas chaves *dip-switch* ($S1$ e $S2$). Com o comando da chave de ligar ($S1$), o relé de comando do capacitor é acionado e o barramento DC começa a se carregar. Quando o valor da tensão do barramento atinge o valor de referência ajustado pelo potenciômetro R352, o relé de comando do contator é acionado e o contator é energizado. Com isto, o resistor de carga do capacitor do

barramento é curto-circuitado possibilitando o carregamento final do barramento DC. Desta forma, a corrente de carga do acionamento irá fluir pelo contator e não pelo resistor de carga do capacitor.

Ao se acionar a chave de desligar (S2), o contator é desenergizado, isolando o retificador trifásico do barramento DC. Ao mesmo tempo, para possibilitar o completo descarregamento do barramento DC, o chopper é acionado, colocando-se uma carga resistiva em paralelo com o capacitor do barramento. Enquanto o inversor permanecer desligado, o chopper estará acionado. O chopper também será acionado se a tensão do barramento DC ultrapassar um valor de referência ajustado pelo potenciômetro R354.

O chopper é comandado pela placa de comando modelo SKH110 da *Semikron* e o circuito desenvolvido possui duas sinalizações visuais, feitas pelos leds D9 e D10. O primeiro led é acionado caso seja identificado algum erro na placa de comando do chopper e o segundo é acionado toda vez que o chopper for acionado. Caso exista algum erro na placa de comando do chopper pode ser realizado um reset desta placa através da chave S3.

A placa desenvolvida possui um conector DB9, utilizado para conectá-la à placa do circuito de interface. Através deste conector pode-se, também, comandar o inversor em substituição às chaves *dip-switch*. Através deste conector também é enviado um sinal indicando se o inversor está acionado ou não e se existe um erro na placa de comando do chopper. Além disto, é transmitido, também, o sinal do nível da tensão do barramento. A Tabela 5.1 mostra a pinagem do conector DB9 utilizado, bem como os níveis de tensão de cada sinal e os sinais lógicos necessários ao comando do inversor.

Tabela 5.1: Sinais utilizados pela placa de Startup e Chopper disponíveis no conector DB9

Pino	sinal	Nível de tensão	Sinal lógico
1	Tensão do barramento	0V a +5V	
2, 6 e 7	GND		
3	Reset da placa de comando do Chopper	+15V	pulso positivo
4	Controle para desligar o inversor	+15V	pulso positivo
5	Controle para ligar o inversor	+15V	pulso positivo
8	Sinalização de inversor ligado	+15V	nível alto
9	Sinalização de erro na placa de comando do chopper	+15V	nível alto

5.3 Circuito de Interface de Comandos

Este circuito é responsável pelo gerenciamento dos sinais de erro provenientes dos drivers dos IGBTs principais do inversor (Figura B.12).

As placas de comando dos IGBTs (SKHI23 da *Semikron*) possuem, basicamente, duas proteções. Caso uma das proteções seja ativada, o driver é desabilitado gerando um sinal de erro. Para que a placa volte a funcionar normalmente é necessário aplicar um sinal de reset.

Como são utilizadas três placas de comando independentes, foi necessário desenvolver um circuito no qual, caso uma das placas de comando apresente um sinal de erro, todos os sinais de comando, que eram aplicados aos *gates* dos IGBTs, são desabilitados. Uma outra função básica do circuito desenvolvido é a capacidade de se habilitar ou desabilitar os sinais que são aplicados aos *gates* dos IGBTs, mesmo em funcionamento normal das placas de comando. Desta forma, para o perfeito funcionamento do inversor é necessário manter o sinal de habilitação sempre aplicado. Esta proteção é importante para que se evite um chaveamento indesejado dos IGBTs já que a placa responsável pela geração dos sinais de *gate* é separada da placa de aquisição de dados.

Portanto, caso exista algum sinal de erro em uma das placas de comando dos IGBTs, será gerado um sinal de erro que será transmitido à placa de interface. Também é realizada uma indicação visual através do led D24. O reset das placas de comando pode ser efetuado através da chave S4 ou através de um sinal de controle proveniente de um dos conectores DB9 existentes na placa desenvolvida.

Esta placa possui dois conectores DB9. O conector P24 é responsável pela entrada dos sinais de comando dos *gates*, proveniente da placa de PWM, além do sinal de TRG necessário pela placa de medições digitais. O conector P23 é responsável pela conexão da placa desenvolvida com a placa de interface. As Tabelas 5.2 e 5.3 mostram as pinagens dos conectores, bem como o nível de tensão utilizado em cada um dos sinais.

Tabela 5.2: Sinais de comando para os gates dos IGBTs. Conector P24.

Pino	Sinal
1	Comando IGBT A
2	Comando IGBT \bar{A}
3	Comando IGBT B
4	Comando IGBT \bar{B}
5	Comando IGBT C
6	Comando IGBT \bar{C}
7	GND
8	TRG1
9	não conectado

Tabela 5.3: Sinais de controle utilizados pelo circuito. Conector P23.

Pino	Sinal	Nível de tensão	Sinal lógico
1	Sinalização de erro nas placas de comando dos IGBTs	+15V	nível alto
2	Controle de habilitação dos sinais de comando	+15V	nível alto
3	Controle de reset das placas de comando dos IGBTs	+15V	pulso positivo
4	TRG	+15V	
5	não conectado		
6,7,8 e 9	GND		

5.4 Circuito de Histerese de Corrente

Este circuito é responsável por gerar os sinais de comando para os drivers dos IGBTs principais utilizando um controlador de histerese de corrente com limitação da frequência de chaveamento (Figura B.5).

Este circuito gera, diretamente a partir de três correntes de referência e dos três valores de correntes medidas, os sinais de comando para os *gates* dos IGBTs.

O sinal de erro entre as correntes de referência e as correntes reais medidas são aplicados a três comparadores de janela (Figura 5.4) cuja janela é ajustada através da seguinte equação:

$$\Delta V_{in} = \frac{2 \cdot V_{cc} \cdot R_2}{R_1 + R_2} \quad (5.1)$$

O sinal de saída dos comparadores é, então, aplicado às entradas de três flip-flops tipo D, cujo clock é determinado pelo oscilador (Figura 5.5). Desta forma, a maior frequência do sinal de chaveamento é definida pelo oscilador, através da equação:

$$f = \frac{1}{C_1 \cdot R_1 \cdot \ln\left(\frac{2 \cdot R_2 + R_3}{R_3}\right)} \quad (5.2)$$

Desta forma, pode-se acionar o inversor utilizando, também, a histerese de corrente. A diferença deste circuito de histerese de corrente dos circuitos convencionais reside no fato de que, neste circuito, a máxima frequência de chaveamento é limitada.

Esta placa possui dois conectores DB9. Um responsável pela saída dos sinais de comando a serem aplicados nos *gates* dos IGBTs (conector P10) e o segundo responsável pelos sinais de entrada das correntes de referência e correntes medidas. As Tabelas 5.4 e 5.5 mostram as pinagens e o nível de tensão dos sinais de cada um dos conectores.

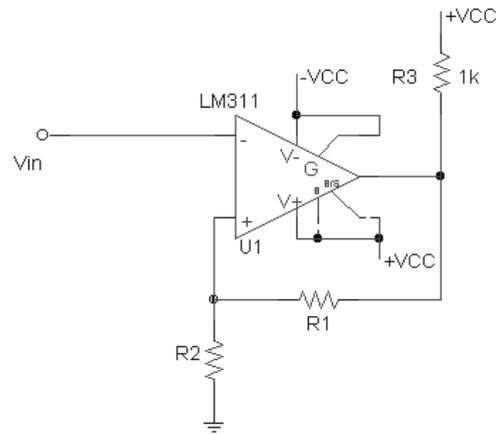


Figura 5.4- Circuito lógico do comparador com janela utilizado na histerese de corrente.

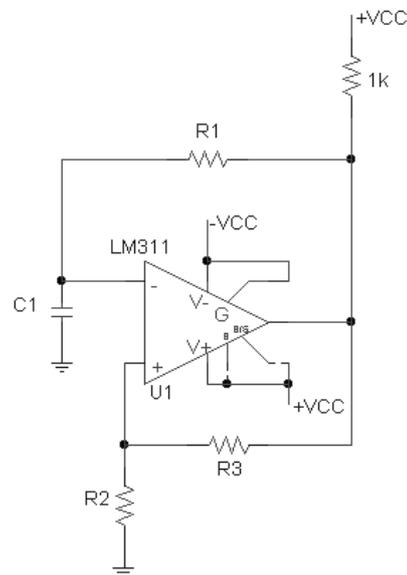


Figura 5.5- Oscilador utilizado para a limitação de frequência da histerese de corrente.

Tabela 5.4: Sinais de comando para os gates dos IGBTs. Conector P10.

Pino	Sinal	Nível de tensão
1	Comando IGBT A	Coletor aberto
2	Comando IGBT \bar{A}	Coletor aberto
3	Comando IGBT B	Coletor aberto
4	Comando IGBT \bar{B}	Coletor aberto
5	Comando IGBT C	Coletor aberto
6	Comando IGBT \bar{C}	Coletor aberto
7	GND	
8	não conectado	
9	não conectado	

Tabela 5.5: Sinais de entrada para o circuito de histerese. Conector P9.

Pino	Sinal	Nível de tensão
1	Corrente de referência da fase ab	$\pm 5V$
2	Corrente de referência da fase ca	$\pm 5V$
3	Corrente medida da fase ab	$\pm 5V$
4	Corrente medida da fase ca	$\pm 5V$
5	GND	
6	Corrente de referência da fase bc	$\pm 5V$
7	não conectado	
8	Corrente medida da fase bc	$\pm 5V$
9	GND	

5.5 Circuito de Aquisição de Corrente e Tensão

Este circuito é responsável pela medição das tensões de fase, tensões do barramento utilizadas pelo circuito de medição digital de tensão e das correntes de linha, com a posterior transformação para grandezas de fase (Figura B.13).

O circuito de medição de corrente utiliza três sensores de efeito Hall para a medição das correntes de linha. Estas correntes são, então, aplicadas a um circuito para a transformação das grandezas de linha em grandezas de fase. As correntes de fase são, finalmente disponibilizadas através do conector P19.

A medição das tensões de fase é realizada utilizando-se amplificadores operacionais no modo diferencial para se evitar a utilização de transformadores, os quais iriam incluir uma defasagem entre o sinal medido e o sinal real, uma vez que a tensão medida é a tensão aplicada aos terminais da máquina, usualmente uma tensão na forma de PWM. Desta forma, a saída do circuito de medição de tensão é composta pelas tensões de fase aplicadas aos terminais da máquina. Estas tensões são, então, disponibilizadas no conector P19.

Um outro sistema de medição, necessário ao circuito de medição digital de tensão é implementado neste mesmo circuito. Desta forma, é disponibilizada ao circuito de medição digital o sinal de tensão da diferença entre os terminais do barramento DC e as tensões terminais da máquina. Estes sinais são disponibilizados através do conector P20.

As Tabelas 5.6 e 5.7 mostram a pinagem dos conectores DB9 utilizados nesta placa, bem como o nível de tensão de cada sinal.

5.6 Circuito de Medição Analógica de Correntes e Tensões

Responsável pelas medições analógicas das tensões e correntes de fase (Figura B.9, e B.11).

Tabela 5.6: Pinagem do conector P19 que disponibiliza os sinais das medições das correntes e tensões de fase.

Pino	Sinal	Nível de tensão
1,2 e 6	GND	
3	Corrente medida na fase <i>ca</i>	$\pm 5V$
4	Corrente medida na fase <i>bc</i>	$\pm 5V$
5	Corrente medida na fase <i>ab</i>	$\pm 5V$
7	Tensão medida na fase <i>ca</i>	$\pm 5V$
8	Tensão medida na fase <i>bc</i>	$\pm 5V$
9	Tensão medida na fase <i>ab</i>	$\pm 5V$

Tabela 5.7: Pinagem do conector P20 que disponibiliza os sinais das medições necessárias ao circuito de medição digital.

Pino	Sinal	Nível de tensão
1	Diferença de tensão entre a fase <i>a</i> e o terminal negativo do barramento DC	$\pm 5V$
2	Diferença de tensão entre a fase <i>c</i> e o terminal negativo do barramento DC	$\pm 5V$
3	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>a</i>	$\pm 5V$
4	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>c</i>	$\pm 5V$
5,7 e 9	GND	
6	Diferença de tensão entre a fase <i>b</i> e o terminal negativo do barramento DC	$\pm 5V$
8	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>b</i>	$\pm 5V$

Foram implementadas duas formas distintas para a medição da tensão analógica. A primeira é a aplicação direta de filtros Butterworth passa-baixa de segunda ordem. Os filtros foram sintonizados em, aproximadamente, $600Hz$ (Figura B.9). Como se sabe, a defasagem decorrente da aplicação de filtros é uma função da frequência do sinal a ser medido. Desta forma, para acionamentos a frequência variável, a compensação da defasagem imposta pelo filtro não é uma tarefa trivial e o sistema de controle tem que ser capaz de compensar esta distorção introduzida pela presença do filtro.

Uma segunda forma de medição das tensões de fase envolve a integração da tensão em um período de PWM ([Chin et al., 1998, Neves et al., 1999]). Desta forma, para cada fase medida, são necessários dois integradores. Enquanto um integrador disponibiliza a sua saída para a medição, o segundo integrador efetua a operação de integração (Figura B.10). Desta forma, a defasagem máxima entre o valor aplicado à máquina e a medição é de apenas um período de PWM, independente da frequência do sinal medido.

A medição das correntes de fase pode ser realizada de duas formas, com a aplicação de filtros Butterworth passa-baixa de segunda ordem, sintonizados em aproximadamente $600Hz$, ou através do sinal direto do circuito de aquisição de corrente, sem a presença de filtros.

Todas as grandezas analógicas medidas, tanto tensão quanto correntes, são disponibilizadas por um conector DB25 para a placa de interface. Este conector é responsável, também, por fornecer o sinal de sincronismo (TRG) necessário ao circuito dos integradores. O circuito de medição analógica de corrente e tensão possui, ainda, um sistema para a conversão das grandezas trifásicas para o sistema de eixos $\alpha\beta$ fixos no estator (Figura B.11).

Um outro conector DB9 é responsável por fornecer a este circuito os valores de tensão e correntes provenientes do circuito de aquisição de corrente e tensão. As Tabelas 5.8 e 5.9 mostram as pinagens e os sinais de cada um dos conectores existentes nesta placa.

Tabela 5.8: Pinagem do conector P22 responsável por fornecer os valores de tensão e corrente provenientes do circuito de aquisição de corrente e tensão.

Pino	Sinal
1,2 e 6	GND
3	Corrente medida na fase <i>ca</i>
4	Corrente medida na fase <i>bc</i>
5	Corrente medida na fase <i>ab</i>
7	Tensão medida na fase <i>ca</i>
8	Tensão medida na fase <i>bc</i>
9	Tensão medida na fase <i>ab</i>

Tabela 5.9: Pinagem do conector P21 responsável disponibilizar para a placa de interface as grandezas analógicas medidas.

Pino	Sinal
1	Tensão <i>ab</i> utilizando filtro
2	Tensão <i>ca</i> utilizando filtro
3	Tensão <i>bc</i> utilizando integração em um período de PWM
4	GND
5	Corrente <i>bc</i> utilizando filtro
6	Corrente <i>ab</i> sem utilizar filtro
7	Corrente <i>ca</i> sem utilizar filtro
8	Tensão de eixo α a partir da tensão filtrada
9	Tensão de eixo α a partir da tensão integrada
10	GND
11	Corrente de eixo β a partir da corrente filtrada
12	Corrente de eixo β a partir da corrente não filtrada
13	GND
14	Tensão <i>bc</i> utilizando filtro
15	Tensão <i>ab</i> utilizando integração em um período de PWM
16	Tensão <i>ca</i> utilizando integração em um período de PWM
17	Corrente <i>ab</i> utilizando filtro
18	Corrente <i>ca</i> utilizando filtro
19	Corrente <i>bc</i> sem utilizar filtro
20	GND
21	Tensão de eixo β a partir da tensão filtrada
22	Tensão de eixo β a partir da tensão integrada
23	Corrente de eixo α a partir da corrente filtrada
24	Corrente de eixo α a partir da corrente não filtrada
25	TRG

5.7 Circuito de Medições Digitais de Tensão e Velocidade

Responsável pela medição digital das tensões de fase e da velocidade do motor (Figura B.4).

Este circuito é composto por três EPLD (*Erasable Programmable Logic Device*), da ALTERA (CIs Tensão CI-a, Tensão CI-b e Encoder). EPLDs são dispositivos totalmente programáveis que permitem o desenvolvimento de circuitos digitais complexos em um reduzido espaço. Sua utilização confere ao circuito uma grande versatilidade e modularidade ao projeto.

O CI Encoder é responsável pela medição de velocidade do encoder incremental, o qual está conectado ao eixo da máquina de indução, e pela multiplexação dos sinais tanto da própria medição de velocidade quanto dos outros CIs de medição de tensão.

As Tabelas 5.10, 5.11 e 5.12 mostram as pinagens e os sinais de cada um dos conectores existentes nesta placa. São utilizados dois conectores DB9 e um conector DB25. Um conector DB9 (P4) é responsável pela alimentação do encoder e por disponibilizar os sinais do encoder para a placa desenvolvida. O segundo conector DB9 (P5) é responsável por disponibilizar para a placa desenvolvida os sinais oriundos da placa de aquisição de corrente e tensão. O conector DB25 (P3) é responsável por conectar esta placa à placa de interface.

Tabela 5.10: Pinagem do conector P4 responsável por conectar a placa de medição digital ao encoder incremental.

Pino	Sinal
1	Sinal <i>a</i>
2	Sinal <i>b</i>
3	Sinal <i>zero</i>
4	Alimentação do encoder - +5V
5	Alimentação do encoder - GND
6, 7 e 8	não conectado

O circuito de medição digital utiliza quatro sinais de controle, oriundos da placa de interface. Três destes sinais de controle indicam qual a grandeza que está sendo disponibilizada através dos 13 bits de dados (*D0* a *D12*). O quarto sinal de controle

Tabela 5.11: Pinagem do conector P5 responsável por conectar a placa de medição digital à placa de aquisição de corrente e tensão.

Pino	Sinal
1	Diferença de tensão entre a fase <i>a</i> e o terminal negativo do barramento DC
2	Diferença de tensão entre a fase <i>c</i> e o terminal negativo do barramento DC
3	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>a</i>
4	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>c</i>
5,7 e 9	GND
6	Diferença de tensão entre a fase <i>b</i> e o terminal negativo do barramento DC
8	Diferença de tensão entre o terminal positivo do barramento DC e a fase <i>b</i>

Tabela 5.12: Pinagem do conector P3 utilizado para conectar a placa de medição digital placa de interface.

Pino	Sinal
1	Dado de saída <i>D12</i>
2	Dado de saída <i>D11</i>
3	Dado de saída <i>D10</i>
4	Dado de saída <i>D9</i>
5	Dado de saída <i>D8</i>
6	Dado de saída <i>D7</i>
7	Dado de saída <i>D6</i>
8	Dado de saída <i>D5</i>
9	Dado de saída <i>D4</i>
10	Dado de saída <i>D3</i>
11	Dado de saída <i>D2</i>
12	Dado de saída <i>D1</i>
13	Dado de saída <i>D0</i>
14	Sentido de rotação da máquina
15	Sinal de controle <i>CRT0</i>
16	Sinal de controle <i>CRT1</i>
17	Sinal de controle <i>tempo</i>
18	Sinal de controle <i>Strobe</i>
19	Sinal de controle <i>TRG</i>
20, 21 e 22	não conectado
23, 24 e 25	GND

(*Strobe*) é responsável pela aquisição do dado de saída. Desta forma, pode-se sincronizar o momento no qual a placa de aquisição de dados inicia a amostragem dos sinais analógicos com o instante no qual os dados digitais serão armazenados e disponibilizados na saída do circuito digital, garantindo-se, assim, que todos os dados (digitais e analógicos) sejam amostrados no mesmo instante de tempo. A Tabela 5.13 mostra a relação entre os sinais de controle e as variáveis disponibilizadas na saída da placa de medições digitais.

Tabela 5.13: Sinais de controle da placa de medição digital e a respectiva grandeza disponibilizada na saída do circuito.

<i>CTR0</i>	<i>CTR1</i>	<i>Tempo</i>	Sinal medido
0	0	0	Medição de velocidade por pulso
0	0	1	Medição de velocidade por tempo
0	1	x	Tensão entre a fase <i>a</i> e o barramento
1	0	x	Tensão entre a fase <i>b</i> e o barramento
1	1	x	Tensão entre a fase <i>c</i> e o barramento

5.7.1 Medição de Velocidade

O encoder utilizado na máquina gera dois sinais, defasados em 90 graus, de 2048 pulsos/rotação e um terceiro sinal de apenas um pulso por rotação.

A velocidade da máquina é realizada utilizando-se dois métodos diferentes, medição por tempo ou medição por pulso. O tipo de medição a ser realizada é controlada por um sinal de controle proveniente do conector P3, o qual conecta esta placa à placa de interface.

A Figura 5.6 mostra o diagrama implementado na EPLD responsável pela medição de velocidade.

5.7.1.1 Medição de velocidade por pulso

Neste tipo de medição, conta-se a quantidade de pulsos existente em um período fixo de tempo. Para gerar o período fixo de tempo utilizou-se um oscilador a cristal de $10MHz$.

O sinal do encoder, de 2048 pulsos passa através de um filtro digital, composto por

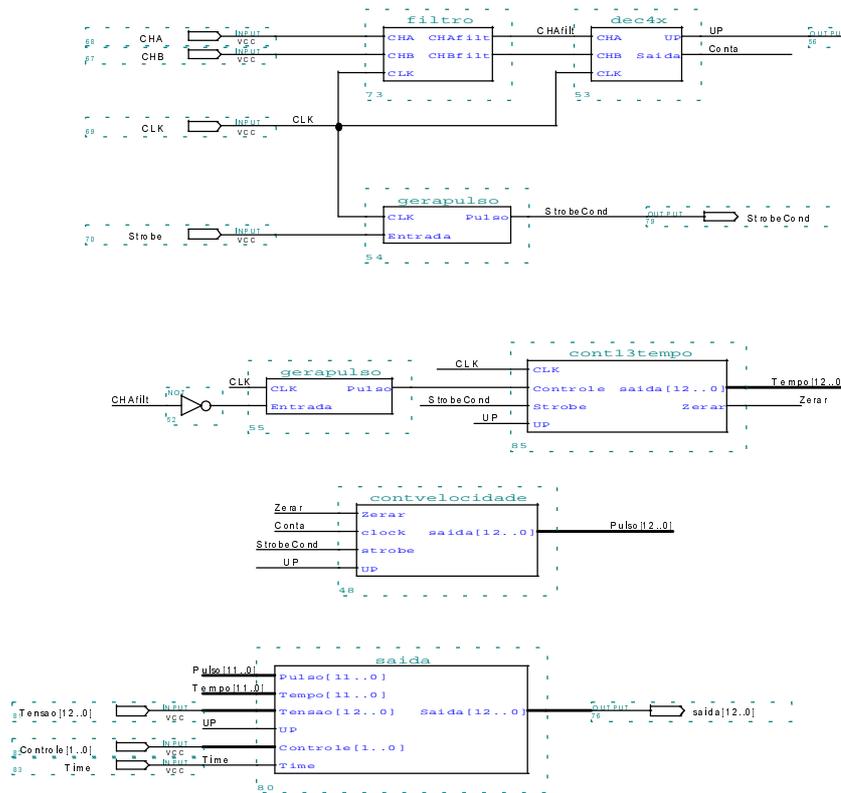


Figura 5.6- Diagrama esquemático do sistema implementado na EPLD do circuito de medição digital, responsável pela medição de velocidade e multiplexação dos sinais de saída.

flip-flops tipo D e, logo após aplicado a um decodificador de quadratura. Este decodificador, utilizando os dois sinais do encoder, defasados em 90 graus, gera um sinal na saída com quatro vezes a frequência do encoder. Com isto consegue-se aumentar a resolução do encoder. Um outro sinal proveniente do decodificador em quadratura é o sentido de giro da máquina. A Figura 5.7 mostra o diagrama de tempo dos dois sinais do encoder (sinal *a* e *b*) quando a máquina gira nos dois sentidos. Observa-se que o decodificador de quadratura é, na verdade, um decodificador de estados, os quais são definidos pelos sinais *a* e *b*. O sentido de transição entre os estados define, também, a direção de rotação da máquina.

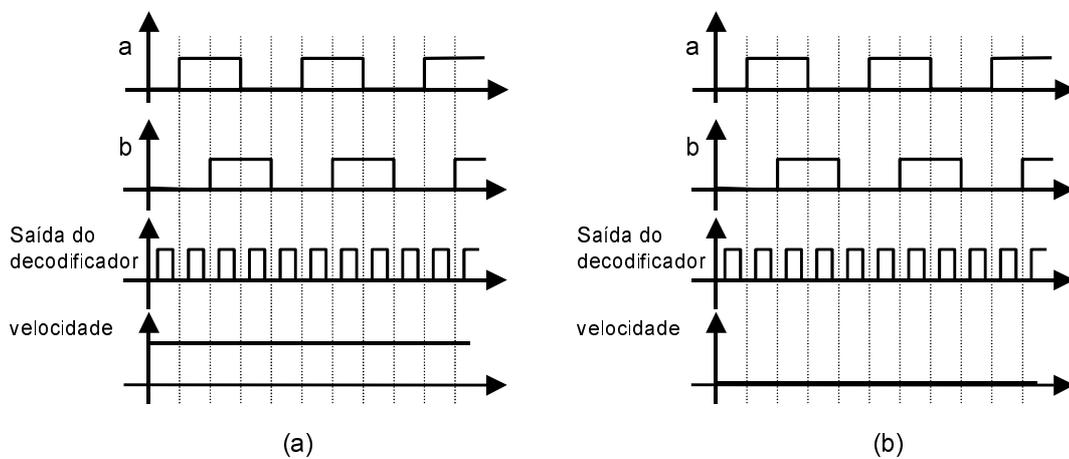


Figura 5.7- Diagrama de tempo do decodificador de quadratura. (a) máquina girando no sentido horário e (b) máquina girando no sentido anti-horário.

A saída do decodificador de quadratura é aplicada ao sinal de clock de um contador, o qual é habilitado na transição de subida do sinal do oscilador de cristal. No pulso subsequente do oscilador, a saída do contador é transferida para um latch de saída e, então, zerada, preparando-o para uma nova contagem. A velocidade da máquina é, então, diretamente proporcional ao número de pulsos contados. Desta forma, considerando *saida* o valor final do contador, a velocidade da máquina é calculada através da seguinte equação:

$$\omega = k_p \cdot \text{saida} \quad (5.3)$$

onde k_p é uma constante de proporcionalidade ($[k_p] = \text{rad/s}$) entre o valor do circuito de medição e a velocidade do motor.

Considerando-se o encoder de 2048 pulsos e f_{clk} a frequência de oscilação do cristal, tem-se:

$$k_p = \frac{\pi \cdot f_{clk}}{4096} \quad (5.4)$$

5.7.1.2 Medição de velocidade por tempo

Neste tipo de medição utilizou-se o mesmo oscilador a cristal de $10MHz$, responsável por gerar um trem de pulso. Desta forma, conta-se o número de pulsos existentes no intervalo de dois pulsos subseqüentes do sinal do encoder. Quando existe uma borda de descida no sinal do encoder, um contador, cujo sinal de clock é proveniente do oscilador, é habilitado, iniciando a contagem. Na borda de subida subseqüente, o contador é desabilitado e o sinal final da contagem é transferido a um latch de saída. Logo após, antes da borda de subida, o contador é zerado para que ele possa iniciar uma nova contagem. Desta forma, tem-se, na saída do circuito um sinal que é inversamente proporcional à velocidade da máquina. Sendo *saída* o valor final do contador, pode-se determinar a velocidade da máquina através da seguinte equação:

$$\omega = k_t \cdot \frac{1}{saída} \quad (5.5)$$

onde k_t é uma constante de proporcionalidade ($[k_t] = rad/s$) entre o valor do circuito de medição e a velocidade do motor. Para o encoder de 2048 pulsos e considerando f_{clk} a frequência de oscilação do cristal, tem-se:

$$k_t = \frac{\pi \cdot f_{clk}}{1024} \quad (5.6)$$

Neste circuito, não se utilizou o decodificador de quadratura do encoder, obtendo-se, portanto, 2048 pulsos por rotação da máquina.

5.7.2 Medição de Tensão

As Figuras 5.8 e 5.9 mostram os dois diagramas implementados em cada uma das EPLDs responsáveis pela medição de tensão. O sistema de medição de tensão será explicado a seguir.

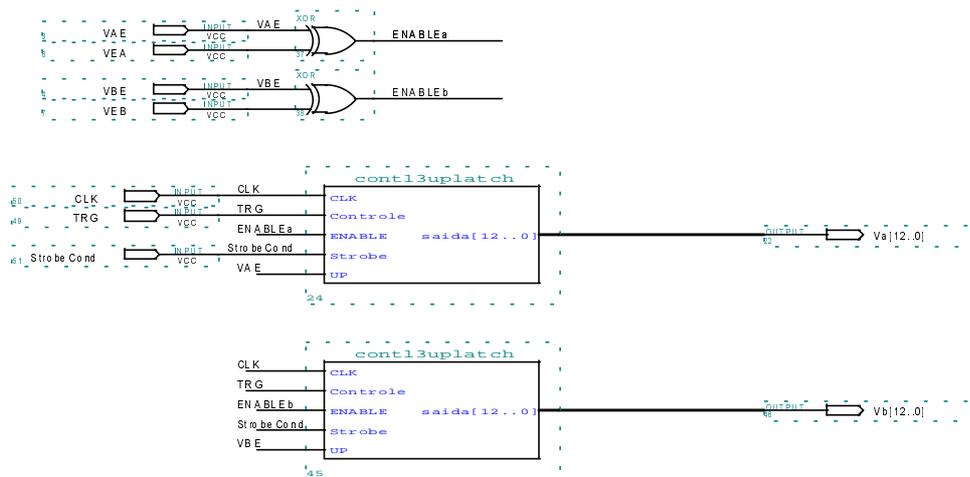


Figura 5.8- Diagrama implementado em uma das EPLDs responsáveis pela medição de tensão (CI Tensão CI-a).

Rocha e outros [Rocha et al., 1999], apresentaram um interessante e robusto método de medição digital da tensão de PWM, bem como de sua integral. O método consiste em um sistema composto por: um detector dos estados das tensões de linha da saída do inversor; um sistema de lógica combinacional responsável por fornecer a direção de contagem dos contadores utilizados como integradores, bem como de quando iniciar ou parar a contagem; um grupo de retificadores utilizados para fornecer a tensão média da amplitude dos pulsos de PWM; um VCO (oscilador controlado por tensão), responsável por gerar um trem de pulsos cuja frequência é proporcional à amplitude dos pulsos de PWM e, finalmente, por dois EPLDs (*Erasable Programmable Logic Devices*), responsáveis pela implementação dos contadores e demais circuitos lógicos combinacionais necessários.

Em sua implementação, Rocha e outros [Rocha et al., 1999] utilizam os sinais de comando do PWM para a detecção do estados das chaves. Desta forma, para cada par de

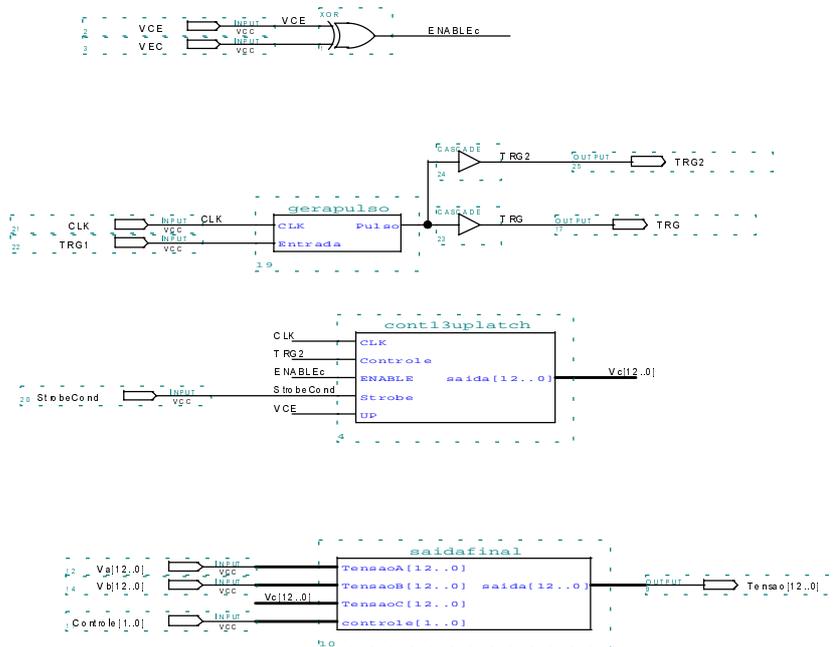


Figura 5.9- Diagrama implementado em uma das EPLDs responsáveis pela medição de tensão (CI Tensão CI-b).

Tabela 5.14: Tabela Verdade dos estados possíveis assumidos pelas chaves $ch3$ e $\overline{ch3}$ e consequente saída dos amplificadores operacionais

$Ch3$	$\overline{Ch3}$	V_{E+C}	V_{CE-}
<i>aberta</i>	<i>aberta</i>	<i>GND</i>	<i>GND</i>
<i>aberta</i>	<i>fechada</i>	<i>+VCC</i>	<i>GND</i>
<i>fechada</i>	<i>aberta</i>	<i>GND</i>	<i>+VCC</i>

chaves é necessário se obter os sinais de comando provenientes do circuito responsável pela geração do PWM. Para tal é necessário a utilização de um par de optoacopladores que, dada a frequência de chaveamento, precisam ter uma boa resposta de frequência. Esta implementação também não leva em consideração a presença do tempo morto, uma vez que este tempo é incluído depois da geração dos sinais de comando de PWM, pelas placas dos drivers de comando e não pelo circuito responsável pelo PWM. De maneira semelhante, são necessários diodos rápidos para a retificação da tensão entre fases que será utilizada pelo circuito do VCO.

Uma nova forma para a medição da tensão aplicada aos terminais da máquina, considerando a integração digital do sinal, foi desenvolvida neste trabalho. Neste sistema não é necessário a utilização de optoacopladores e nem de diodos rápidos, simplificando o sistema de medição e reduzindo os custos de implementação. Neste novo circuito, a presença do tempo morto no chaveamento é, também, considerada. É importante ressaltar que no sistema desenvolvida por Rocha e outros, o tempo morto não era considerado.

Para a implementação do sistema de medição, são necessários dois amplificadores operacionais (AO), configurados no modo diferencial e saturados, para cada braço do inversor, conforme mostra o diagrama esquemático da Figura 5.10. Desta forma o circuito final é implementado de tal forma que cada AO pode assumir dois estados, convenientemente colocados em $+VCC$ e GND , conforme mostra a Tabela 5.14.

A Figura 5.11 exemplifica os sinais medidos e a consequente geração do sinal de habilitação ($ENABLE_c$) pelo circuito mostrado na Figura. 5.10. Os sinais de comando das chaves $ch3$ e $\overline{ch3}$ são gerados a partir do PWM simétrico, sincronizado com o sinal de TRG . O sinal TRG^* é gerado com o sinal TRG e o sinal de Clock do oscilador, de tal forma que quando TRG muda para nível alto é gerado um pulso de clock. Pode-se observar pela figura que o tempo morto é detectado através do sinal de $ENABLE_c$.

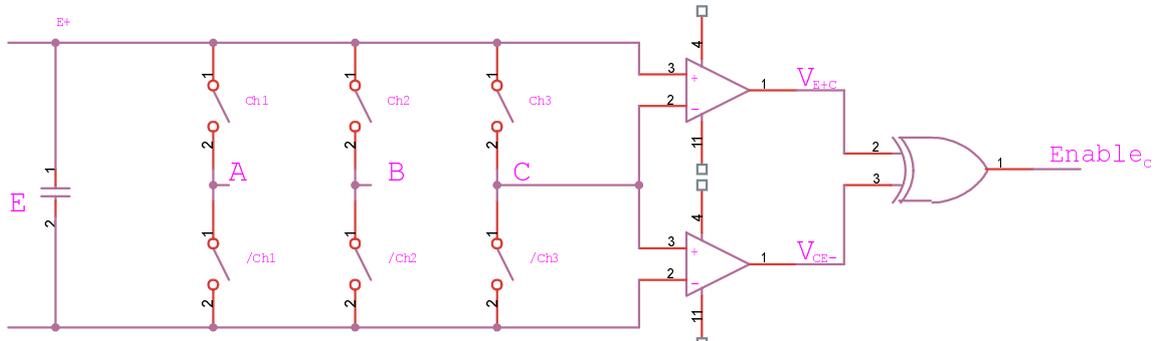


Figura 5.10- Diagrama esquemático do circuito necessário para a medição das tensões sobre cada chave que serão posteriormente utilizadas pela EPLD.

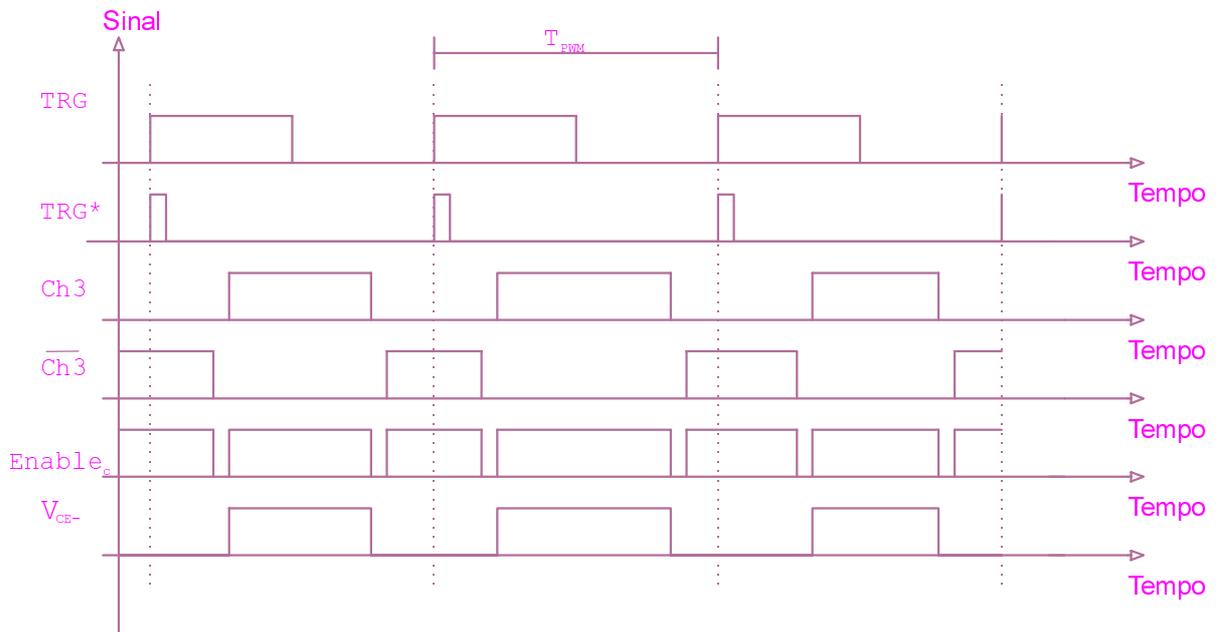


Figura 5.11- Sinais de comando provenientes do sistema de medição. $ch3$ e $\overline{ch3}$ representam os estados das chaves após a inclusão do tempo morto proveniente das placas dos drivers de comando das chaves de potência.

Uma vez determinado o sinal de habilitação, pode-se utilizar tanto a tensão V_{E+C} quanto a tensão V_{CE-} para a integração utilizando-se um integrador crescente/decrescente simples, no qual o sinal de clock é oriundo de um oscilador a cristal. Para facilitar a determinação das tensões entre fase foi utilizada a tensão V_{CE-} .

Todas as demais fases são medidas da mesma maneira, de tal forma que, ao fim de um período de PWM, obtêm-se os sinais V_{AE-} , V_{BE-} e V_{CE-} . Utilizando-se estes sinais, podem-se determinar as tensões entre fases:

$$V_{AB} = V_{AE-} - V_{BE-}$$

$$V_{BC} = V_{BE-} - V_{CE-}$$

$$V_{CA} = V_{CE-} - V_{AE-}$$

Uma vez que todos os sinais digitais estão conectados a um mesmo barramento, é necessária a utilização de um multiplexador controlado pelo programa de controle para que os dados necessários sejam adquiridos um de cada vez. O circuito de determinação do sinal de habilitação ($ENABLE_c$), os contadores necessários, circuito para geração de TRG^* e o multiplexador foram implementados dentro de uma EPLD da ALTERA[®].

A implementação deste sistema requer apenas um sinal de sincronismo com o PWM (sinal TRG). Por utilizar uma medição indireta da tensão entre fases, os pulsos de PWM medidos pelos AO possuem largura maior do que os pulsos que seriam obtidos se fosse realizada uma medição direta da tensão entre fases. Outra característica é a não necessidade de retificadores, sendo que os sinais de saída dos circuitos comparadores já são sinais digitais (níveis $+V_{cc}$ e GND). Desta forma pode-se utilizar AOs que não possuem um alto *slew rate*, não sendo necessária a utilização de opto-acopladores especiais. Nesta implementação, a frequência do oscilador é mantida constante, uma vez que a tensão do barramento é amostrada para a implementação do controle digital, e o ganho do circuito de medição digital da tensão é determinado como uma função desta tensão amostrada, não sendo, portanto, necessária a utilização de circuitos retificadores e

filtros, eliminando a necessidade de diodos rápidos.

5.8 Circuito de Medição de Fluxo e Conjugado

Este circuito é responsável pela medição do fluxo e do torque da máquina de indução (Figuras B.6, B.7 e B.8).

O sistema de medição de torque envolve um torquímetro rotacional instalado entre a máquina de corrente contínua e a máquina de indução. Este torquímetro disponibiliza um sinal proporcional ao torque no eixo do motor, utilizando uma ponte completa de strain-gauges.

Desta forma, o circuito implementado (Figura B.8) possui um amplificador de instrumentação seguido por um filtro passa baixa de Butterworth de segunda ordem, sintonizado em $2kHz$ e um sistema para a compensação de amplitude e off-set. O off-set do valor medido é ajustado através do potenciômetro R689 e o ganho final através do potenciômetro R676. O torquímetro é alimentado com uma tensão de $+10V$, com relação ao terra comum do circuito. Desta forma, para propiciar esta alimentação, utilizou-se um circuito integrado regulador de tensão para garantir o baixo ripple na alimentação do torquímetro. A Tabela 5.15 mostra a pinagem do conector DB9 (P7) utilizado para conectar a placa de medição de fluxo e torque ao torquímetro.

Tabela 5.15: Pinagem do conector P7 responsável por conectar a placa de medição de torque e fluxo ao torquímetro.

Pino	Sinal
1	Sinal 1 da ponte de strain gauges
2	Sinal 2 da ponte de strain gauges
3, 4 e 8	não conectado
5	GND
6	Sinal 1 da ponte de strain gauges
7	Sinal 2 da ponte de strain gauges
9	Tensão de alimentação do torquímetro $+10V$

Para a medição do fluxo, rebobinou-se a máquina de indução para incluir dois sistemas distintos:

1. medição através de bobinas exploradoras [Lipo, 1977];
2. medição através da tensão parcial nas bobinas das fases [Lipo & Chang, 1986];

Desta forma, para o funcionamento dos circuitos de medição de fluxo, são necessários sinais de tensão provenientes da máquina de indução. Estes sinais são disponibilizados para a placa de medição de fluxo e torque através de um conector DB15 (P6). A Tabela 5.16 mostra a pinagem e os sinais elétricos oriundos da máquina.

Tabela 5.16: Pinagem do conector P6 responsável por conectar a placa de medição de torque e fluxo a máquina de indução.

Pino	Sinal
1	Tensão da bobina <i>A</i> do eixo α
2	Tensão da bobina <i>C</i> do eixo α
3	Tensão da bobina <i>A</i> do eixo β
4	Tensão da bobina <i>C</i> do eixo β
5	Tensão terminal 1 da fase <i>A</i>
6	Tensão terminal 2 da fase <i>A</i>
7	Tensão terminal 1 da fase <i>B</i>
8	Tensão terminal 2 da fase <i>B</i>
9	Terminal comum do eixo α
10, 12 e 14	não conectado
11	Terminal comum do eixo β
13	Terminal comum da fase <i>A</i>
15	Terminal comum da fase <i>B</i>

5.8.1 Medição do fluxo através de bobinas exploradoras

O sistema de medição através de bobinas exploradoras foi inicialmente proposto por Lipo, em 1977 [Lipo, 1977]. O Apêndice C detalha melhor a forma de implementação do sistema.

O circuito implementado (Figura B.6) para a medição do fluxo utilizando esta proposta inclui um novo sistema de compensação do off-set existente nos integradores utilizados. Desta forma, o circuito compreende um filtro passa-alta e um sistema de detecção de saturação. Quando os integradores utilizados estão próximos da saturação, o sistema de detecção é ativado e todos os integradores são zerados. Desta forma, impede-se que o sistema fique saturado, ao mesmo tempo que se evita a necessidade de um sistema realimentado, o que poderia ser instável (devido a presença de um integrador).

O filtro passa-alta é responsável por eliminar a componente contínua existente devido ao off-set dos operacionais ou devido às condições iniciais do sistema.

5.8.2 Medição do fluxo através da tensão parcial nas bobinas das fases

Esta proposta foi apresentada por Lipo, em 1986 [Lipo & Chang, 1986] e não requer a presença de enrolamentos extras ao enrolamento já existente na máquina, como é o caso de bobinas exploradoras. O sistema proposto por Lipo está detalhado no Apêndice D.

De maneira equivalente ao circuito anterior, o circuito implementado (Figura B.7) apresenta também o mesmo sistema de compensação de off-set descrito anteriormente. Desta forma garante-se que o valor medido pela placa de aquisição de dados estará sempre desprovido de componentes contínuas, evitando, também, a saturação dos integradores utilizados.

A placa desenvolvida para o circuito de medição de fluxo e torque possui, também, um conector DB9 (P8) responsável por disponibilizar os valores medidos para a placa de interface. A Tabela 5.17 mostra a pinagem e os sinais disponibilizados por este conector.

Tabela 5.17: Pinagem do conector P8 responsável por conectar a placa de medição de torque e fluxo a placa de interface.

Pino	Sinal
1	Fluxo do eixo α utilizando bobinas exploradoras
2	Fluxo do eixo α utilizando tensão parcial
3	não conectado
4, 5 e 9	GND
6	Fluxo do eixo β utilizando bobinas exploradoras
7	Fluxo do eixo β utilizando tensão parcial
8	torque

5.9 Circuito de Medição do Tacogerador

Este circuito é responsável pela medição analógica de velocidade através do tacogerador (Figura B.15).

Este circuito é composto por um amplificador operacional diferencial seguido por

um filtro de Butterworth passa-baixa de segunda ordem, sintonizado em, aproximadamente, $200Hz$ (utilizado para eliminar os ruídos de alta frequência pela presença de escovas e comutador) e transforma este sinal de tensão em um sinal de corrente de 4 a $20mA$, conferindo, desta forma, uma maior imunidade a ruídos. O sinal de corrente é convertido novamente em um sinal de tensão pela placa de interface.

Este circuito possui um conector DB9, responsável pela alimentação do circuito e pela saída do sinal de corrente. A Tabela 5.18 mostra a pinagem e os sinais deste conector.

Tabela 5.18: Pinagem do conector P2 responsável pela alimentação do circuito do tacogerador e por disponibilizar o sinal de corrente de saída do circuito.

Pino	Sinal
1	Tensão de alimentação de $+15V$
2, 3 e 4	GND
5	Corrente de saída
6	Tensão de alimentação de $-15V$
7, 8 e 9	não conectado

5.10 Circuito de Interface

Este circuito é responsável pela interface entre o inversor e a placa de aquisição de dados (Figuras B.2 e B.3).

Por este circuito passam todas as medições, digitais e analógicas, as quais são, então, enviadas à placa de aquisição de dados. Devido a uma característica da placa de aquisição de dados, a qual não pode ter nenhum sinal aplicado às suas entradas sem que ela esteja previamente alimentada, a placa de interface, além da alimentação normal do inversor, utiliza a alimentação da própria placa de aquisição. Desta forma, se a placa de aquisição não estiver devidamente alimentada, não existirá nenhum sinal aplicado às suas entradas.

Na placa de interface podem ser selecionados os sinais medidos que serão realmente amostrados pela placa de aquisição. Desta forma, através de jumpers, são selecionados, dentre os vários métodos de medição, aqueles que serão realmente utilizados pelo sistema de controle implementado.

A placa de interface possui, também, sistemas de proteção para impedir que os níveis de tensão das grandezas medidas ultrapassem o nível máximo suportado pela placa de aquisição de dados.

Esta placa possui sete conectores DB, utilizados para conectá-la às demais placas do inversor e mais um conector DB50 utilizado para conectá-la à placa de aquisição de dados. A Tabela 5.19 mostra a pinagem e os sinais do conector DB50 (P14). Todos os demais conectores já foram descritos nas sessões anteriores.

Tabela 5.19: Pinagem do conector P14 utilizado para conectar a placa de aquisição de dados à placa de interface.

Pino	Sinal	Pino	Sinal
1	Referência de corrente da fase <i>A</i>	26	Sinalização de erro do choper
2	Tensão de alimentação de +12V	27	medição digital - dado <i>D11</i>
3	Tensão de alimentação de -12V	28	medição digital - Dado <i>D3</i>
4	Corrente de eixo β	29	medição digital - Dado <i>D5</i>
5	Corrente de eixo α	30	medição digital - Dado <i>D3</i>
6	Tensão de eixo α	31	medição digital - Dado <i>D1</i>
7	Fluxo de eixo α	32	Controle ligar inversor
8	Velocidade (tacogerador)	33	Controle reset placas de comando dos IGBTs
9	sinalização inversor ligado	34	
10	Sinalização erro nas placas de comando dos IGBTs	35	Referência de corrente da fase <i>c</i>
11	medição digital - Dado <i>D12</i>	36	Tensão V_{bc}
12	medição digital - Dado <i>D6</i>	37	Tensão V_{ca}
13	medição digital - Dado <i>D4</i>	38	Corrente I_{ab}
14	medição digital - Dado <i>D2</i>	39	Corrente I_{bc}
15	medição digital - Dado <i>D0</i>	40	Corrente I_{ca}
16	Controle desligar inversor	41	extra
17	Tensão de alimentação +5V	42	GND
18	Referência de corrente da fase <i>B</i>	43	medição digital - Dado <i>D8</i>
19	Tensão V_{av}	44	medição digital - Dado <i>D9</i>
20	Torque	45	medição digital - Dado <i>D10</i>
21	nãoconectado	46	Controle Habilitação sinais de gate
22	Tensão de eixo β	47	Controle <i>tempo</i>
23	fluxo de eixo β	48	controle <i>CTR1</i>
24	Tensão do barramento DC	49	Controle <i>CTR0</i>
25	Controle <i>Strobe</i>	50	Controle reset do chopper

5.11 Conjunto Motor de Indução/Motor Corrente Contínua

O sistema a ser acionado é composto de um motor de indução (MI) acoplado a um motor

Tabela 5.20: Parâmetros da máquina de indução utilizada

Tensão de alimentação [V] - ligação Δ/Y	92.4/160
Corrente Nominal [A] - ligação Δ/Y	14.4/8.3
Potência [CV]	2.0
Velocidade nominal [rpm]	1715
Resistência de estator [Ω]	0.995
Indutância de dispersão de estator [H]	$2.362E - 03$
Resistência de rotor [Ω]	0.696
Indutância de dispersão de rotor [H]	$3.525E - 03$
Indutância de Magnetização [H]	$45.601E - 3$
Coefficiente de perdas rotacionais [$W \cdot s^2/rad^2$]	$8.718E - 04$
Momento de Inércia [$N \cdot m \cdot s^2$]	$6.547E - 03$

Tabela 5.21: Parâmetros da máquina de corrente contínua utilizada

Resistência de Armadura [Ω]	1.610
Indutância de Armadura [H]	$8.643E - 3$
Km [V/rpm]	$81.10E - 3$

de corrente contínua (MCC). O MCC é utilizado como carga do MI, desta forma pode-se simular a aplicação e retirada de carga, bem como a aplicação de um determinado perfil de carga ao MI.

Rebobinou-se o MI de forma a incluir dois sistemas de medição de fluxo (Apêndices C e D), além de reduzir a tensão de alimentação do motor. O conjunto MI-MCC possui, além da medição de fluxo, um torquímetro rotacional, um tacogerador acoplado ao eixo do MCC e um encoder incremental acoplado ao eixo do MI (Figura 5.12).

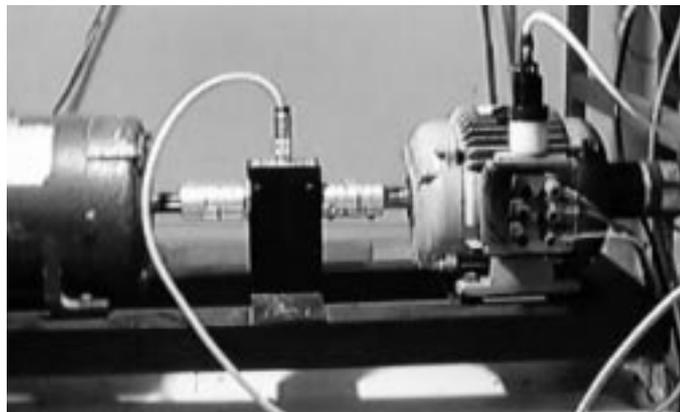


Figura 5.12- Detalhe da instalação do torquímetro rotacional acoplado entre a máquina de indução e a máquina de corrente contínua.

As Tabelas 5.20 e 5.21 mostram os parâmetros da máquina de indução e da máquina de corrente contínua utilizada [Parma et al., 1997b], respectivamente.

Para a medição do fluxo e aquisição dos dados do encoder e do torquímetro, foi necessário desenvolver uma placa de circuito capaz de fazer a interface entre estes sensores e a placa de aquisição de dados presente no microcomputador, Apêndice B.

5.11.1 Torquímetro

Para a medição direta do conjugado imposto pela máquina de indução, foi instalado um torquímetro rotacional, marca HBM, modelo T5/20Nm, entre a máquina de indução e a máquina de corrente contínua.

5.11.2 Encoder

Instalou-se um encoder incremental, com 2048 pulsos/revolução, marca Dynapar, modelo HA5262048B5E3, no eixo da máquina de indução.

5.11.3 Medição de fluxo

Conforme mencionado anteriormente, foram implementados dois sistemas de medição de fluxo da máquina de indução. Para tal, teve-se que rebobinar a máquina de indução. A Figura 5.13 mostra o detalhe do conector de saída da máquina de indução. Através deste conector os sinais de tensão necessários aos circuitos de medição de fluxo são transmitidos às placas de tratamento do sinal.

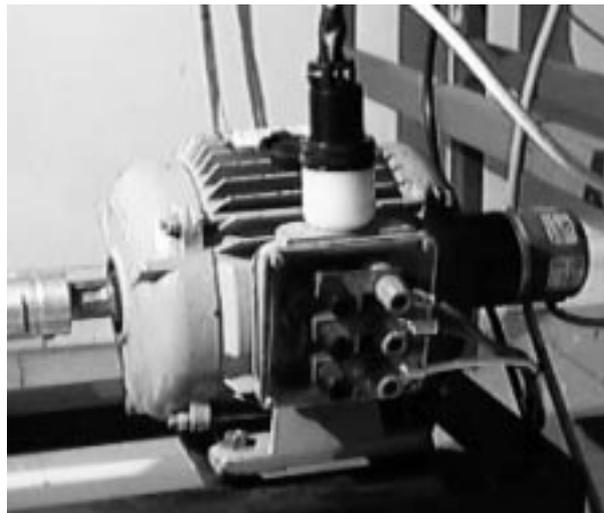


Figura 5.13- Detalhe da máquina de indução utilizada na bancada experimental. O conector disponibiliza os sinais de tensão necessários à medição de fluxo.

5.12 Conclusões

Neste capítulo foi relatado o desenvolvimento de uma bancada de testes para o acionamento de motores capaz de efetuar, em uma mesma estrutura de *hardware*, diversas medições, inclusive redundantes. Estas medições redundantes possibilitarão a

análise e comparação, em uma mesma estrutura de controle, da utilização de diferentes técnicas de medição, resultando na determinação da melhor técnica para um determinado problema. Além deste fato, será possível a aferição de alguns sistemas de observadores e estimadores de grandezas envolvidas no acionamento como, por exemplo, observadores de fluxo e torque. Será possível, também, a análise da influência de técnicas de medição distintas, de uma mesma grandeza elétrica, no sistema de acionamento em estudo.

Foi apresentada, também, uma nova técnica de medição de tensão. Esta técnica, além de ser uma implementação digital, leva em consideração a presença do tempo morto, o que resultará em medidas mais precisas da tensão aplicada aos terminais da máquina. Por ser um sistema digital, o valor medido é mais imune a ruídos do que as técnicas de medição analógica, além de possuir uma defasagem com relação ao sinal real fixa, ou seja, independente da frequência na qual se faz o acionamento. Esta técnica também é uma importante contribuição para os sistemas de acionamentos elétricos.

Capítulo 6

Resultados de Simulação e Experimentais

Este capítulo será dedicado à avaliação do desempenho dos algoritmos propostos no Capítulo 4. Após uma comparação prévia entre os algoritmos de treinamento *off-line* propostos e o algoritmo *Backpropagation* padrão, todos os algoritmos propostos são comparados com alguns algoritmos encontrados na literatura, tendo sido definida uma plataforma de testes composta de três classes de problemas distintos, mostradas abaixo.

1. Aproximação de função: escolheram-se algumas funções e comparou-se o tempo de treinamento obtido com os algoritmos propostos com o tempo de treinamento obtido com outros algoritmos clássicos. Nesta classe de problema foram testados os algoritmos para treinamento *on-line* e *off-line*;
2. Classificação: Escolheu-se um problema típico de classificação e comparou-se o tempo de treinamento obtido com os algoritmos propostos com o tempo de treinamento obtido com outros algoritmos clássicos. Nesta classe de problemas foram testados os algoritmos *off-line*. Comparou-se, também, o nível de acerto da rede treinada com cada um dos tipos de algoritmos;
3. Controle do acionamento do motor de indução: Utilizaram-se RNAs na malha de controle do motor de indução, obtendo-se resultados de simulação e experimentais para os algoritmos de treinamento *on-line*.

Todas as redes implementadas são MLP, cuja dimensão é dependente do problema em questão. Os algoritmos de treinamento *off-line* serão chamados de, a partir deste ponto, SMC1 e SMC2, relativos à primeira proposta de treinamento *off-line* (seção 4.1) e segunda proposta de treinamento *off-line* (seção 4.1), respectivamente.

6.1 Treinamento de um Único Neurônio

Utilizou-se um único neurônio com apenas uma entrada, capaz, portanto, de discriminar apenas classes linearmente separáveis, para possibilitar a visualização do ajuste dos pesos durante o treinamento. Como o neurônio utilizado possui apenas um peso e um termo de polarização, é possível visualizar no plano definido pelas variáveis *peso* X *bias* (espaço dos pesos), a evolução do erro da saída em função do ajuste dos pesos. Desta forma pode-se, graficamente, avaliar a trajetória do erro gerada pelos algoritmos de treinamento *off-line* propostos (SMC1 e SMC2), comparando-as com a trajetória fornecida pelo algoritmo *Backpropagation*.

Utilizou-se, portanto, apenas um neurônio com dois pesos (um peso e um termo de *bias*) para a separação de duas classes composta por dois vetores de treinamento, sendo uma entrada e uma saída $\{x, y\} : \{-3.0, 0.4\}$ e $\{2.0, 0.8\}$. A tangente hiperbólica foi utilizada como função de ativação do neurônio. Como o neurônio utilizado possui apenas dois parâmetros (peso e bias), pode-se traçar a superfície de erro em um espaço tri-dimensional, mostrado na Figura 6.1.

Pode-se observar pela Figura 6.1 que a superfície de erro possui duas regiões planas. O treinamento do neurônio utilizando métodos de gradiente descendente pode ser demasiadamente lento se os pesos iniciais forem escolhidos nestas duas regiões. O mínimo para esta superfície de erro, determinado analiticamente, ocorre para o par $(bias, peso)$ igual à $(-1.7, 0.3)$.

Escolheram-se quatro diferentes condições iniciais para o par $(bias, peso)$. Em cada uma delas o treinamento possui um diferente nível de dificuldade. Desta forma pode-se avaliar o desempenho dos algoritmos de treinamento em função das condições iniciais do treinamento. A Tabela 6.1 mostra a escolha do par $(bias, peso)$ para cada uma das simulações. Estes mesmos valores iniciais estão denotados na Figura 6.1.

Realizaram-se diversas simulações para cada uma das condições iniciais possibilitando ajustar os ganhos envolvidos nos algoritmos de forma a obter a melhor

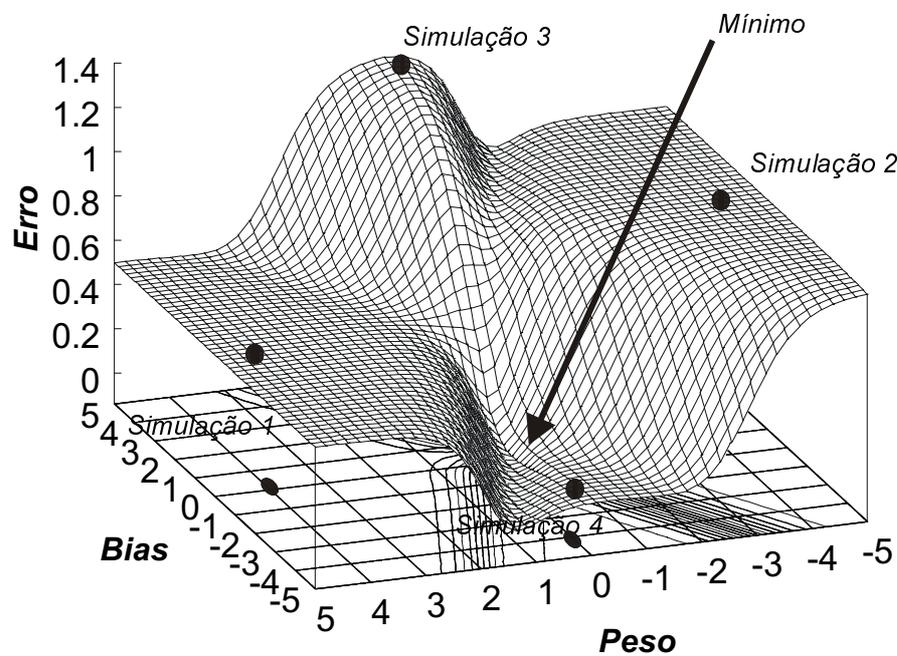


Figura 6.1- Superfície de erro para um neurônio treinado para reconhecer os pares $\{-3.0, 0.4\}$ e $\{2.0, 0.8\}$.

Tabela 6.1: Condições iniciais para o par $(bias, peso)$ utilizado no treinamento de um único neurônio.

Simulação	Par $(bias, peso)$
1	$(0, 4)$
2	$(0, -4)$
3	$(4, 0)$
4	$(-4, 0)$

Tabela 6.2: Ganhos dos treinamentos utilizados por cada um dos algoritmos utilizados.

Algoritmo	Ganhos do treinamento
<i>Backpropagation</i>	$\eta = 0.8$
SMC1	$\alpha = 0.3$ e $C = 10$
SMC2	$\alpha = 0.3$ e $C = 10$

performance, no que diz respeito ao número de *epochs*, de cada um dos algoritmos utilizados. Foram implementados os algoritmos SMC1, SMC2 e *Backpropagation*. A Tabela 6.2 mostra os ganhos dos treinamentos de cada um dos algoritmos.

A Figura 6.2 mostra a projeção da trajetória do erro sobre o espaço dos pesos (plano *peso X bias*) para cada um das condições iniciais das simulações. Apenas a trajetória do erro está denotada nesta figura, entretanto, o número de *epochs* necessárias ao treinamento, em cada uma das condições iniciais, está mostrado na Tabela 6.3. Adotou-se um limite de erro igual a 10^{-4} para todos os algoritmos. Observa-se, através da Tabela 6.3, a dependência do algoritmo *Backpropagation* com as condições iniciais do treinamento. Os algoritmos propostos mantiveram, aproximadamente, o mesmo número de *epochs* em todas as simulações, independente da condição inicial do sistema.

O erro em função do número de *epochs*, para cada uma das condições iniciais, está mostrado nas Figuras 6.3, 6.4, 6.5 e 6.6. Pode ser observado, através destas figuras, que os algoritmos propostos aumentam a velocidade de convergência em regiões planas (Simulações 1 e 2).

O erro mínimo alcançado em todos os algoritmos ocorre para o par (*bias, weight*) igual à $(-1.66, 0.27)$ o qual é muito próximo do par que corresponde ao erro mínimo real. Apenas o número de *epochs* é diferente, como pode ser observado na Tabela 6.3. Desta forma pode-se verificar que a inclusão da teoria de modos deslizantes é capaz de acelerar o treinamento da rede pois, agora, existe uma direção pré-definida (ajustada através da superfície de deslizamento) de ajuste dos pesos.

6.2 Aproximação de Funções

Foram realizados dois testes distintos. Inicialmente, os algoritmos propostos para o

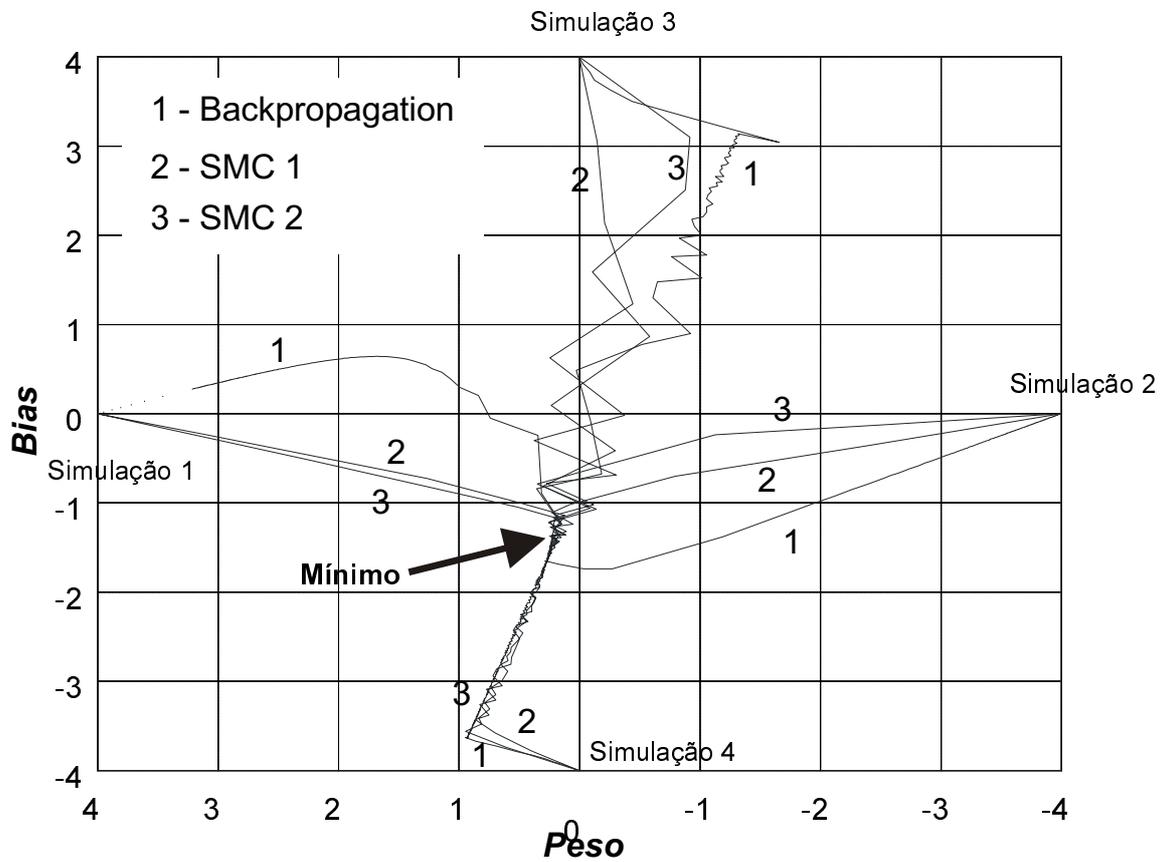


Figura 6.2- Evolução dos pesos em cada uma das simulações para o treinamento de um único neurônio.

Tabela 6.3: Número de *epochs* necessárias ao treinamento de um único neurônio. Limite de erro estipulado em 10^{-4} .

Algoritmo	Número de <i>epochs</i>			
	Simulação 1	Simulação 2	Simulação 3	Simulação 4
<i>Backpropagation</i>	1676	215	72	362
SMC1	13	13	17	29
SMC2	11	11	18	30

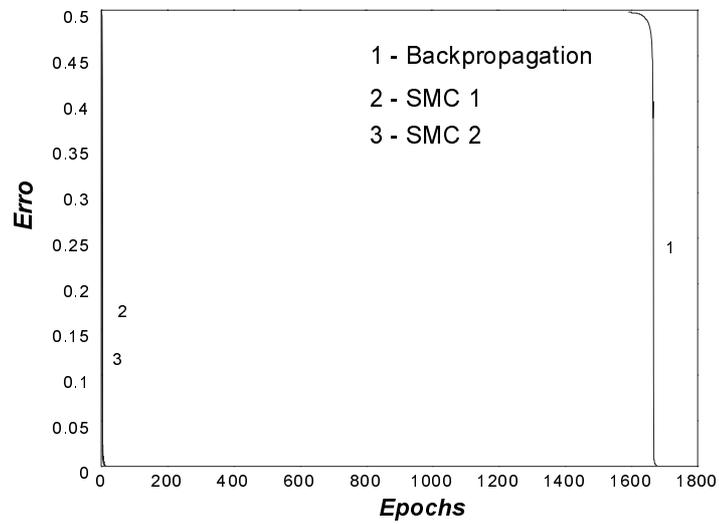


Figura 6.3- Evolução do erro em função do número de *epochs* para a simulação 1.

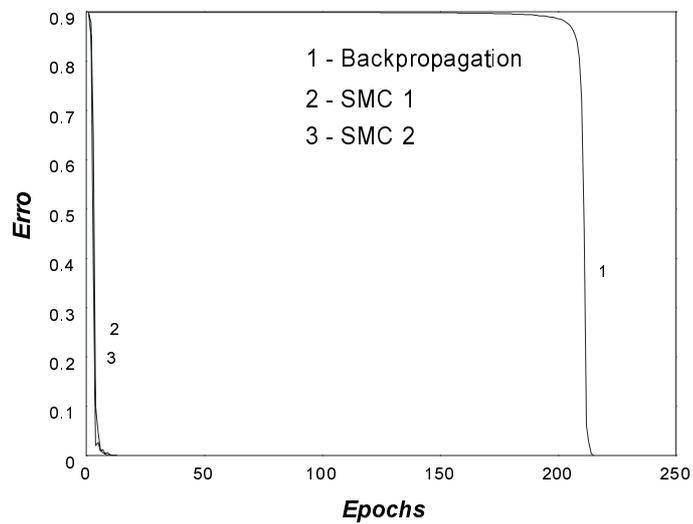


Figura 6.4- Evolução do erro em função do número de *epochs* para a simulação 2.

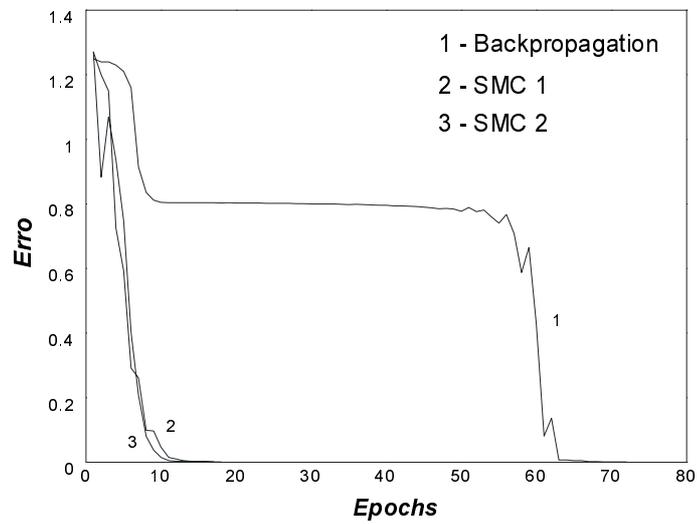


Figura 6.5- Evolução do erro em função do número de *epochs* para a simulação 3.

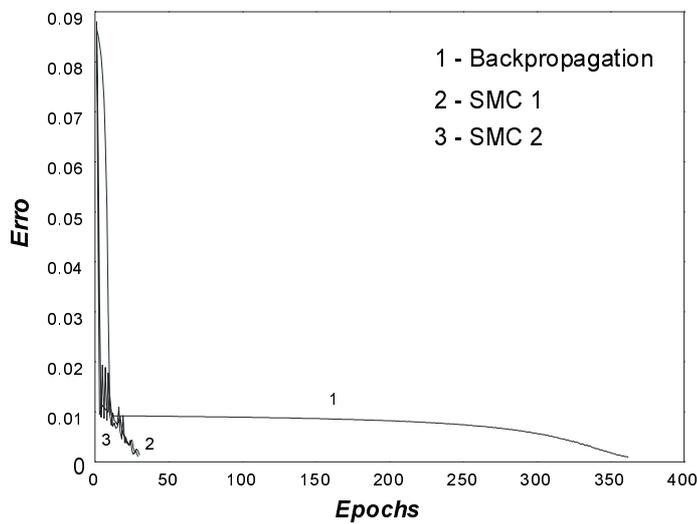


Figura 6.6- Evolução do erro em função do número de *epochs* para a simulação 4.

treinamento *off-line* foram comparados com alguns algoritmos encontrados na literatura e discutidos no Capítulo 2. Em seguida, os algoritmos propostos para o treinamento *on-line* foram comparados com o algoritmo *Backpropagation*.

6.2.1 Treinamento *Off-line*

Além dos algoritmos SMC1 e SMC2, os seguintes algoritmos foram implementados, possibilitando a comparação das performances dos algoritmos de treinamento:

1. *Backpropagation* [Rumelhart et al., 1986]. Utilizou-se este algoritmo porque ele é baseado apenas no gradiente do erro, sendo o precursor de vários algoritmos de treinamento.
2. *Backpropagation com momentum* [Bertsekas & Tsitsiklis, 1996]. Esta é a versão mais conhecida do algoritmo *Backpropagation*, sendo, inclusive, mais rápida do que ele.
3. *Levenberg-Marquardt* [Marquardt, 1963]. Como é bem conhecido na literatura, este algoritmo, considerado de segunda ordem, é um dos algoritmos mais rápidos, considerando apenas o número de *epochs*, para o treinamento de MLPs. Entretanto, ele necessita de cálculos matemáticos mais complexos e, portanto, requer maiores recursos computacionais.
4. *Rprop* [Riedmiller & Braun, 1993]. Este algoritmo foi utilizado como principal ponto de comparação pois ele é considerado um dos algoritmos de primeira ordem com melhor performance, no que diz respeito ao tempo de treinamento, em *epochs* [Schiffmann et al., 1992], encontrado na literatura.
5. *Qprop* [Fahlman, 1988b]. Utilizou-se este algoritmo para poder comparar os algoritmos propostos com um algoritmo de treinamento em lote, considerado, também, um algoritmo de primeira ordem bastante eficiente.
6. *Adaptive Learning Rate 1* [Demuth & Beale, 1994]. Esta versão do *Backpropagation* altera o ganho do treinamento em função do erro de saída. Caso o erro de saída decresça, o ganho do treinamento é aumentado em relação ao ajuste anterior dos pesos. Se o erro de saída aumentar, o ganho do treinamento é reduzido

e o ajuste dos pesos é desfeito.

7. *Adaptive Learning Rate 2* [Demuth & Beale, 1994]. Este algoritmo é similar ao anterior, entretanto ele faz uso do termo de momento, o qual é mantido constante durante todo o treinamento da rede.

Todos os algoritmos foram utilizados sob a mesma condição e implementados em linguagem C. Os parâmetros envolvidos no treinamento, para cada um dos algoritmos, foram escolhidos de forma a obter o melhor resultado para cada um deles, individualmente. Apenas os melhores resultados, dentre várias simulações, serão apresentados.

Para a comparação dos algoritmos de treinamento utilizados, utilizaram-se duas funções distintas, escolhidas em função da dificuldade do treinamento de cada uma delas, representadas nas Figuras 6.8 e 6.7:

1. $f_1(x) = -3.2 \cdot x^2 + 3.2 \cdot x, x \in [0, 1]$;
2. $f_2(x) = x^4, x \in [0, 1]$;

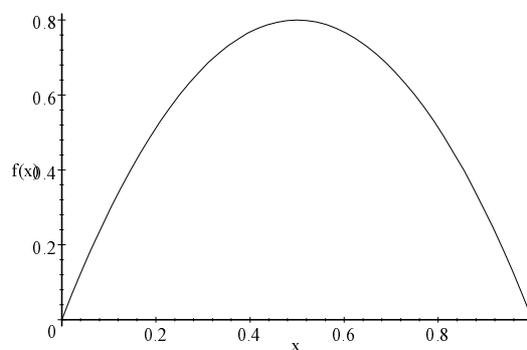


Figura 6.7- Segunda função utilizada na comparação dos algoritmos de treinamento *off-line*.
 $f_1(x) = -3.2 \cdot x^2 + 3.2 \cdot x$.

Os algoritmos foram comparados adotando-se o seguinte procedimento:

1. Uma vez escolhido o conjunto de dados e definida a dimensão da MLP, são feitas tantas cópias da RNA quantos forem os algoritmos de treinamento utilizados. A

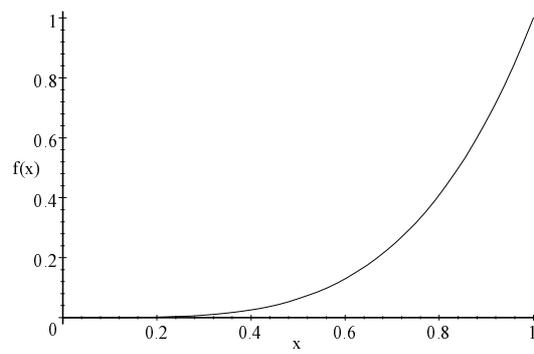


Figura 6.8- Função utilizada para a comparação dos algoritmos de treinamento *off-line*. $f_2(x) = x^4$.

dimensão da MLP utilizada é definida em função da melhor resposta possível, com o menor número de nodos escondidos, sem comprometer a capacidade de generalização da RNA;

2. Os pesos iniciais da RNA são escolhidos aleatoriamente de uma amostra com distribuição normal e média nula. As condições iniciais de todas as redes são idênticas;
3. Um mesmo vetor de treinamento, escolhido aleatoriamente do conjunto de dados, é apresentado a todas as RNAs e, conseqüentemente, a todos os algoritmos de treinamento;
4. O critério para parar o treinamento é adotado para todos os algoritmos implementados. Adotaram-se dois critérios:
 - (a) limite de erro;
 - (b) número máximo de *epochs*;
5. Os parâmetros envolvidos em cada algoritmo de treinamento foram escolhidos de forma a possibilitar o melhor treinamento possível;
6. Considera-se que o treinamento obteve sucesso quando o limite de erro foi atingido antes de completar o número máximo, previamente estipulado, de *epochs*.

Para as duas funções em questão, utilizou-se uma MLP com uma entrada, três nodos na camada escondida e uma saída, utilizando tangente hiperbólica e a função sigmoide como funções de ativação da camada escondida e da camada de saída, respectivamente. Após exaustivos testes, esta configuração se mostrou a mais adequada, possibilitando o treinamento das RNAs pela maioria dos algoritmos de treinamentos implementados. Os parâmetros envolvidos em cada um dos algoritmos utilizados, para cada uma das funções, estão mostrados na Tabela 6.4 (para a função $f_1(x)$) e Tabela 6.5 (para a função $f_2(x)$).

Adotou-se um limite máximo de *epochs* máximo igual à 5000 com um limite de erro igual à $2.5 \cdot 10^{-3}$ para as duas funções utilizadas. A Tabela 6.6 mostra os valores médios do número de *epochs* necessárias ao treinamento das RNAs por cada um dos algoritmos, para cada uma das funções. Este número foi calculado considerando-se diversas simulações, variando-se apenas as condições iniciais dos pesos das redes e a seqüência na qual os

Tabela 6.4: Parâmetros dos algoritmos de treinamentos utilizados para a função f_1 .

Algoritmo	Função $f_1(x)$
	Parâmetros
<i>Backpropagation</i>	$\eta = 0.99$
Momentum	$\eta = 0.99$ e $\mu = 0.1$
Levenberg-Marquardt	$\mu = 0.01$
RProp	$\eta^+ = 1.2$ e $\eta^- = 0.5$
Qprop	$\mu = 1.3$ e $\eta = 0.7$
Adaptive Learning Rate 1	$\eta = 0.99$ $\eta^+ = 1.05$ e $\eta^- = 0.7$
Adaptive Learning Rate 2	$\eta = 0.99$ e $\mu = 0.1$ $\eta^+ = 1.07$ e $\eta^- = 0.7$
SMC1	$\alpha = 2$ e $C = 1$ $\beta = 2$ e $C_H = 1$
SMC2	$\alpha = 0.045$ e $C = 1$ $\beta = 4$ e $C_H = 1$

Tabela 6.5: Parâmetros dos algoritmos de treinamentos utilizados para a função f_2 .

Algoritmo	Função $f_2(x)$
	Parâmetros
<i>Backpropagation</i>	$\eta = 0.9$
Momentum	$\eta = 0.9$ e $\mu = 0.05$
Levenberg-Marquardt	$\mu = 0.01$
RProp	$\eta^+ = 1.2$ e $\eta^- = 0.5$
QPropt	$\mu = 0.5$ e $\eta = 0.84$
Adaptive Learning Rate 1	$\eta = 0.9$ $\eta^+ = 1.05$ e $\eta^- = 0.7$
Adaptive Learning Rate 2	$\eta = 0.9$ e $\mu = 0.5$ $\eta^+ = 1.07$ e $\eta^- = 0.7$
SMC1	$\alpha = 2$ e $C = 10$ $\beta = 2$ e $C_H = 1$
SMC2	$\alpha = 0.034$ e $C = 10$ $\beta = 7.9$ e $C_H = 10$

Tabela 6.6: Número de *epochs* necessárias ao treinamento da RNA para as função apresentadas. Limite de erro igual a $2.5 \cdot 10^{-3}$ e número máximo de *epochs* igual a 5000.

Algoritmo	Número de <i>epochs</i>	
	$f_1(x)$	$f_2(x)$
<i>Backpropagation</i>	1348	-
Momentum	1348	-
Levenberg-Marquardt	23	21
RProp	246	285
Qprop	-	640
Adaptive Learning Rate 1	1084	2983
Adaptive Learning Rate 2	898	1424
SMC1	560	212
SMC2	220	100

vetores de treinamento são apresentados. Observa-se, na referida tabela, que nem todos os algoritmos conseguiram treinar as redes, considerando-se os limites estipulados. Nestes casos, o número de *epochs* está representado, na tabela, com um hífen, denotando que aquele algoritmo não atingiu o resultado esperado, nas condições dadas (limite de erro e número máximo de *epochs*).

A Tabela 6.7 mostra o tempo de processamento dos algoritmos utilizados para o treinamento da RNA, considerando cada uma das funções. Este tempo de processamento inclui, além do processamento das equações do algoritmo, propriamente dito, a escolha do vetor de treinamento a ser apresentado e a armazenagem dos dados resultantes do treinamento para posterior análise.

Observa-se, na Tabela 6.6, que para as duas funções utilizadas, a segunda proposta de treinamento apresentou, considerando-se apenas o número de *epochs*, o menor tempo de treinamento após o algoritmo de Levenberg-Marquadt. Entretanto, como pode ser observado na Tabela 6.7, o algoritmo SMC1 apresentou um menor tempo de processamento do que o algoritmo de Levenberg-Marquadt. Observando-se as equações do algoritmo de Levenberg-Marquadt (seção 2.2), verifica-se que o custo computacional deste algoritmo é maior do que o custo computacional dos algoritmos propostos (seções 4.1 e 4.2).

Analisando, ainda, a Tabela 6.6, observa-se que, para a função $f_2(x)$, não foi possível obter os parâmetros de treinamento para os algoritmos *Backpropagation* e

Tabela 6.7: Tempo de processamento (em segundos) necessário a cada algoritmo para o treinamento da RNA, considerando cada uma das funções e os limites impostos.

Algoritmo	Tempo de processamento (em segundos)	
	$f_1(x)$	$f_2(x)$
<i>Backpropagation</i>	22.68	-
Momentum	22.67	-
Levenberg-Marquardt	0.69	0.96
RProp	4.26	4.86
Adaptive Learning Rate 1	17.90	51.35
Adaptive Learning Rate 2	14.73	22.93
SMC1	1.68	0.70
SMC2	0.66	0.33

Momentum que possibilitasse o treinamento no número máximo de *epochs* estipulado. Também não foi possível ajustar os parâmetros do algoritmo QProp que possibilitasse o treinamento da RNA para a função $f_1(x)$.

As Figuras 6.9 e 6.10 mostram a evolução do erro em função das *epochs*, para um dos treinamentos utilizando as funções $f_1(x)$ e $f_2(x)$, respectivamente. Observa-se, nestas figuras, o desempenho dos algoritmos propostos em comparação aos demais algoritmos.

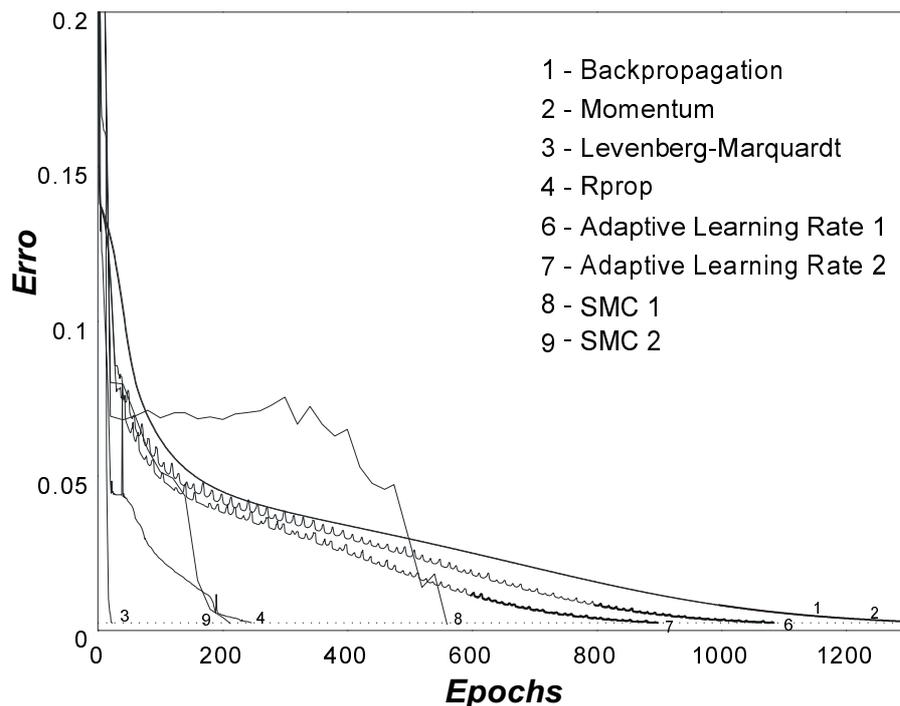


Figura 6.9- Evolução do erro durante o treinamento utilizando a função $f_1(x) = -3.2 \cdot x^2 + 3.2 \cdot x$.

É importante dizer que, uma vez treinadas, todas as RNAs apresentaram, aproximadamente, a mesma capacidade de generalização. Tal fato é um indicativo de que todas as redes atingiram o mesmo mínimo.

Através destas simulações pôde-se observar que os parâmetros do algoritmo RPROP, quando utilizado para o treinamento de MLP em problemas de aproximação de funções, apresentaram-se dependentes das condições iniciais da RNA. Não existindo uma metodologia bem definida para o ajuste destes parâmetros, tendo sido, pois, escolhidos de maneira empírica. Da mesma forma, não existe um estudo aprofundado sobre

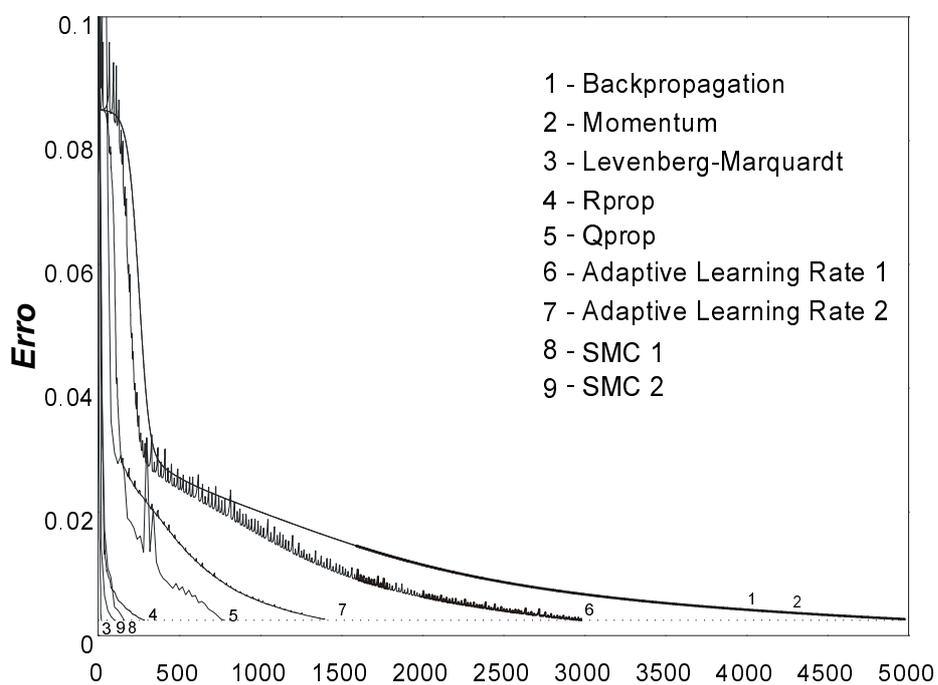


Figura 6.10- Evolução do erro durante o treinamento utilizando a função $f_2(x) = x^4$.

a convergência deste algoritmo. O algoritmo, entretanto, é bastante veloz, quando comparado com o algoritmo *Backpropagation*.

O algoritmo QProp se mostrou menos susceptível às variações das condições iniciais. Entretanto, se utilizado apenas em sua forma original ([Fahlman, 1988b]), sem a consideração do termo de gradiente e sem a alteração na derivada da função de ativação e sem a limitação da máxima variação dos pesos, o algoritmo é bastante ineficiente. Com a consideração de todos estes termos, o ajuste dos parâmetros envolvidos é realizado de maneira empírica e o algoritmo fica dependente do problema e da dimensão da RNA utilizada.

Os algoritmos *Backpropagation* padrão e Momentum também não apresentam uma metodologia para a escolha dos parâmetros envolvidos no treinamento. O algoritmo Momentum mostrou-se bastante sensível à escolha do termo de momento, podendo passar da estabilidade para a instabilidade com uma pequena variação neste termo.

O ganho na velocidade do treinamento obtida pela 1a. Proposta, quando comparado ao algoritmo *Backpropagation* padrão, é devido à superfície de deslizamento. A direção de ajuste dos pesos, agora, não é dependente apenas do gradiente do erro. Desta forma, possibilita-se delinear uma direção, definida pela superfície de deslizamento, na superfície de erro. O ajuste dos pesos tende, portanto, a permanecer nesta superfície, enquanto que, no algoritmo *Backpropagation* padrão, o ajuste dos pesos segue o caminho definido pelo gradiente do erro, o qual não é, necessariamente, o melhor caminho a ser seguido.

De forma semelhante, na 2a. Proposta, a superfície de deslizamento define uma trajetória na superfície de erro a ser seguida pelo ajuste dos pesos. Além deste fato, devido à natureza de memória da somatória existente no módulo do ajuste dos pesos, a cada novo ajuste o peso sofre uma influência de todos os vetores já apresentados. É importante ressaltar que existe uma grande diferença entre o ajuste dos pesos existente nesta proposta e o ajuste feito com, por exemplo, o algoritmo *Backpropagation* para treinamento em lote. No treinamento em lote, um determinado gradiente de erro influencia apenas uma vez no ajuste dos pesos, após a apresentação de todos os vetores de dados, enquanto que, na

2a. Proposta, um determinado gradiente de erro influencia o ajuste dos pesos em todas as vezes, após a apresentação do vetor, dentro da *epoch*. Desta forma podem-se obter tempos de treinamento bastante reduzidos.

6.2.2 Treinamento *On-line*

A primeira proposta de um algoritmo para treinamento *on-line*, Seção 4.3, foi utilizada para o treinamento de duas funções temporais distintas:

1. A primeira função, mostrada na Figura 6.11, possui a seguinte expressão:

$$f_1(t) = \frac{1}{3} [\sin(2 \cdot \pi \cdot 30 \cdot t + 0.7) + \sin(2 \cdot \pi \cdot 20 \cdot t + 0.7) + \sin(2 \cdot \pi \cdot 20 \cdot t + 1)]$$

2. A Segunda função, mostrada na Figura 6.13, possui a seguinte expressão:

$$f_2(t) = \frac{1}{3} \{ \text{Cos} [\sin(2 \cdot \pi \cdot 20 \cdot t) + \sin(2 \cdot \pi \cdot 20 \cdot t)] + \sin(2 \cdot \pi \cdot 10 \cdot t) \}$$

Utilizou-se uma RNA com uma entrada, sete nodos na camada escondida e uma saída, utilizando tangente hiperbólica como função de ativação em ambas camadas, para a aproximação desta função.

A Tabela 6.8 mostra a numeração das curvas adotada nas Figuras 6.11, 6.12 e 6.13.

Tabela 6.8: Numeração utilizada nas figuras para a identificação das curvas.

Numeração	Descrição
1	Dados de treinamento
2	Primeira proposta de treinamento <i>on-line</i>
3	<i>Backpropagation</i>

É importante ressaltar que a periodicidade da curva utilizada no treinamento possibilita à rede o perfeito treinamento, substituindo a repetição da apresentação de todos os vetores de treinamento, como ocorre em um treinamento *off-line*. Por isto, após alguns períodos pode-se considerar que a RNA está treinada.

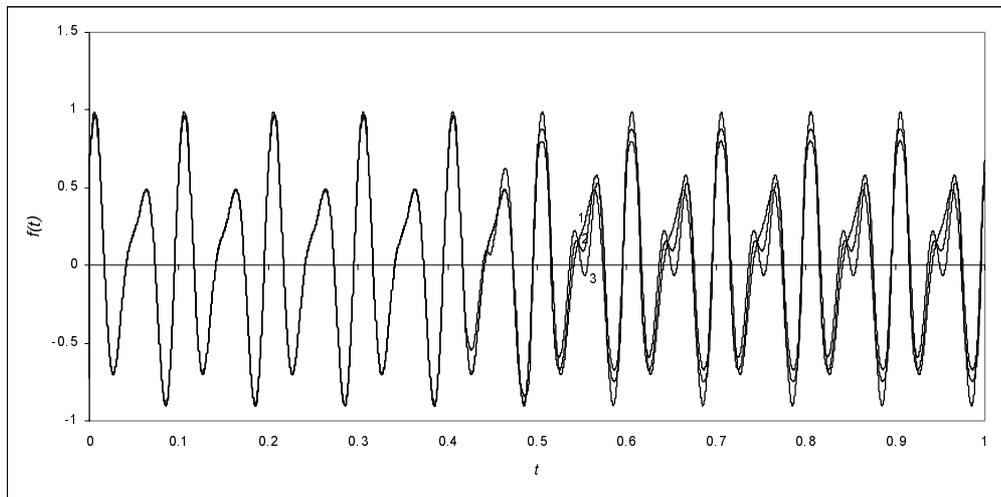


Figura 6.11- Treinamento da RNA utilizando a primeira proposta para treinamento *on-line* e o algoritmo *Backpropagation* . Em $t=0.45s$, os algoritmos de treinamento foram desabilitados.

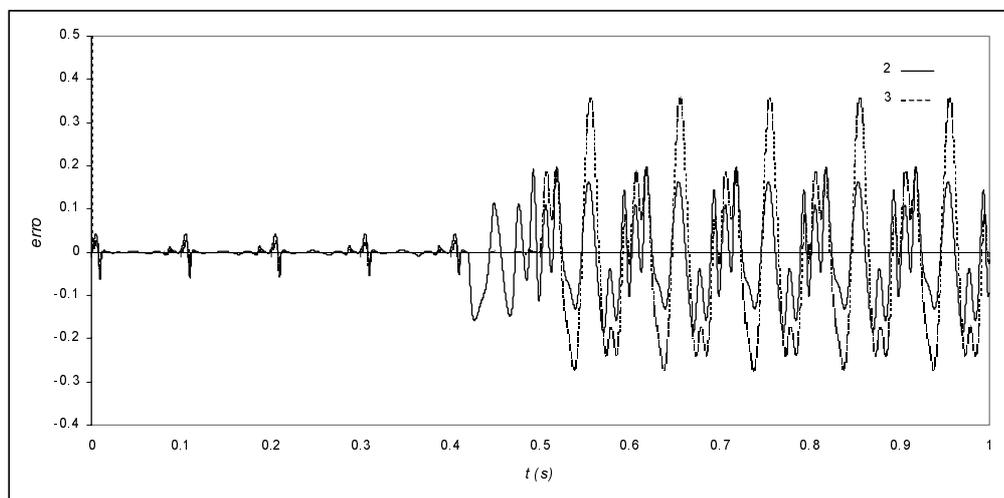


Figura 6.12- Curva de erro referente ao treinamento da RNA utilizando a primeira proposta para treinamento *on-line* e o algoritmo *Backpropagation* . Em $t=0.45s$, os algoritmos de treinamento foram desabilitados.

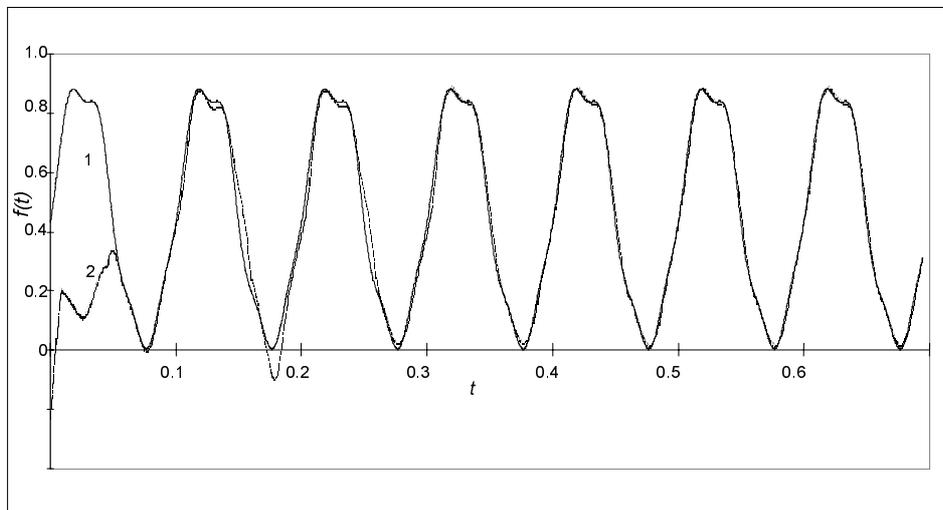


Figura 6.13- Treinamento da MLP, para a segunda função, utilizando a primeira proposta para treinamento *on-line*.

Para o treinamento da RNA utilizada na primeira função implementou-se, além da primeira proposta para treinamento *on-line*, o algoritmo *Backpropagation* padrão, utilizado para treinamento *on-line*. Objetivou-se fazer uma comparação entre o resultado obtido com a utilização das duas técnicas de treinamento. Sabe-se que o algoritmo *Backpropagation* não foi, em princípio, desenvolvido para treinamento *on-line*. Entretanto, encontram-se na literatura diversos trabalhos com esta aplicação [Sabanovic et al., 1996, K. et al., 1997]. Além deste fato, os demais algoritmos de treinamento são menos indicados ainda para o treinamento *on-line* pois eles utilizam, em sua maioria, direta ou indiretamente, os sinais do gradiente do erro, oriundos do algoritmo *Backpropagation*.

Na Figura 6.11, os algoritmos de treinamento foram desabilitados em $t = 0.45s$. Observa-se que, após este tempo, existe uma divergência entre a função a ser aproximada e a saída das RNAs utilizadas por cada um dos algoritmos de treinamento. Entretanto, a resposta da RNA treinada com a primeira proposta para treinamento *on-line* permanece com um menor erro, se comparada com o erro da RNA treinada com o algoritmo *Backpropagation* (Figura 6.12). Este fato pode mostrar que as RNAs alcançaram diferentes pontos de mínimo durante o treinamento e que o algoritmo proposto é menos susceptível a interferência destrutiva, a qual ocorre quando é fornecida uma quantidade maior do que a necessária de uma determinada parte do conjunto de dados, mascarando alguma informação relevante de uma outra parte do conjunto de dados. Este fato ocorre devido a periodicidade da função utilizada.

Observa-se na Figura 6.13 a característica de adaptação da RNA à função que se deseja aproximar, durante o treinamento. Não foi utilizado o algoritmo *Backpropagation* por se tratar apenas de uma figura ilustrativa da capacidade de adaptação da RNA ao sistema na qual ela está inserida, tornando-a uma interessante ferramenta a ser utilizada em sistemas de controle.

6.3 Classificação de Padrões

Para avaliar a performance dos algoritmos de treinamento *off-line* propostos em problemas de classificação, utilizaram-se duas classes de duas variáveis x_1 e x_2 , definidas no plano

$x_1 \times x_2$ (Figura 6.14). As duas classes, $C1$ e $C2$, possuem distribuição Gaussiana, ambas com variância igual a 0.8 e média $(0, 0)$ e $(2, 0)$, respectivamente. O conjunto de treinamento é composto de 200 elementos amostrados aleatoriamente destas duas distribuições com 100 vetores para cada classe. Os vetores para a validação do treinamento foram amostrados da mesma distribuição e foram utilizados para verificar a performance das RNAs treinadas.

Utilizou-se uma RNA com duas entradas (variáveis x_1 e x_2), uma saída (classe $C1$ ou $C2$) e uma camada escondida com quatro neurônios. Utilizou-se a função sigmoide e a função tangente hiperbólica para a ativação dos neurônios da camada escondida e da camada de saída.

A Figura 6.14 mostra os vetores do conjunto de treinamento das duas classes. Como pode ser observado, a média e a variância das classes foram escolhidas de tal forma a existir uma sobreposição entre as classes $C1$ e $C2$, dificultando, portanto, o treinamento das RNAs. A Tabela 6.9 mostra os parâmetros de treinamento utilizados em cada um dos algoritmos implementados. Os mesmos algoritmos utilizados no problema de aproximação de funções foram, aqui, utilizados para comparar os resultados obtidos com algoritmos clássicos com os resultados obtidos com o algoritmo SMC2.

O mesmo procedimento adotado no problema de aproximação de funções foi utilizado no problema de classificação.

A Tabela 6.9 mostra os melhores parâmetros de treinamento utilizados por cada um dos algoritmos implementados. Estes parâmetros foram ajustados de forma a obter o melhor resultado com cada um dos algoritmos. O algoritmo Qprop não apresentou bons resultados neste problema e, por isto, não será incluído na análise deste problema.

As Figuras 6.15 e 6.16 mostram o erro médio quadrático para o conjunto de treinamento. A Figura 6.16 mostra uma ampliação da Figura 6.15 para permitir uma melhor análise do comportamento dos algoritmos. Observa-se, nesta figura e através da Tabela 6.10, que o algoritmo SMC2 alcançou o menor erro de todos os algoritmos, em um menor número de *epochs*.

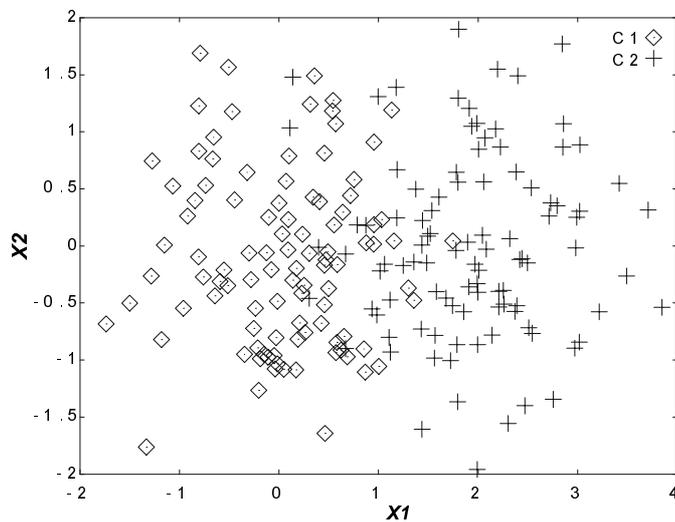


Figura 6.14- Conjunto de treinamento utilizado no problema de classificação.

Tabela 6.9: Parâmetros do treinamento para cada um dos algoritmos utilizados no problema de classificação .

Algoritmo	Parâmetros do Treinamento
<i>Backpropagation</i>	$\eta = 0.9$
Momentum	$\eta = 0.9$ and $\mu = 0.05$
Levenberg-Marquardt	$\mu = 0.01$
RProp	$\eta^+ = 1.2$ and $\eta^- = 0.5$
Adaptive Learning Rate 1	$\eta = 0.9$ $\eta^+ = 1.05$ and $\eta^- = 0.7$
Adaptive Learning Rate 2	$\eta = 0.9$ and $\mu = 0.1$ $\eta^+ = 1.07$ and $\eta^- = 0.7$
SMC2	$\alpha = 0.03$ and $C = 1$ $\beta = 2.$ and $C_H = 1$

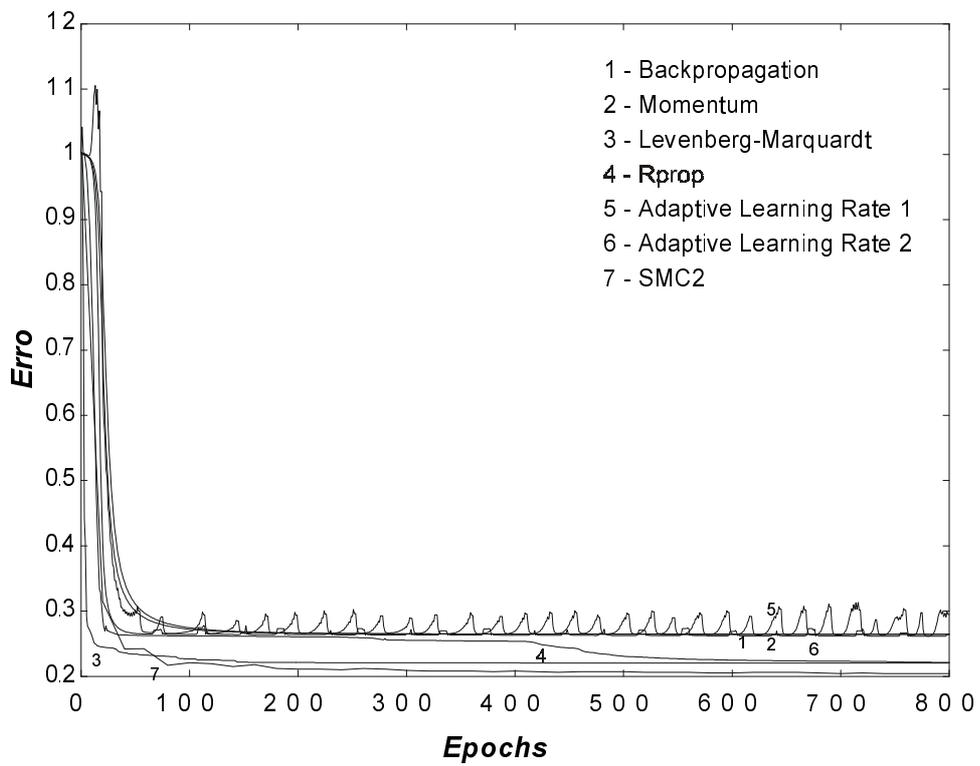


Figura 6.15- Erro durante o treinamento para o problema de classificação de padrões.

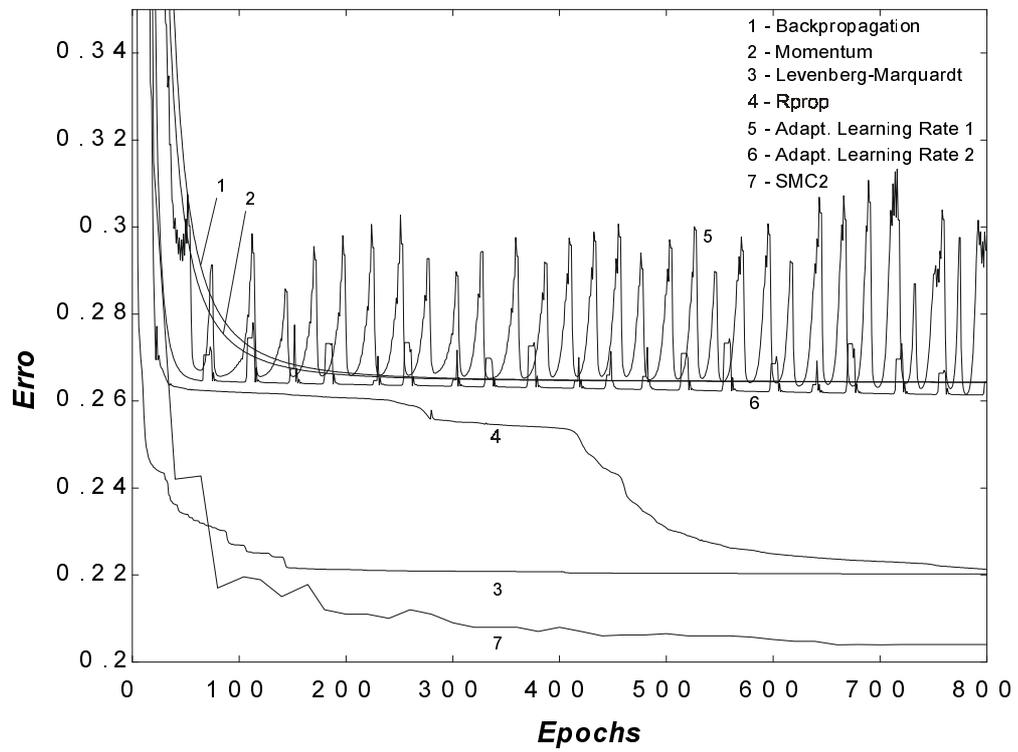


Figura 6.16- Ampliação da figura anterior para uma melhor visualização da convergência dos algoritmos utilizados.

Tabela 6.10: Classificações erradas para o conjunto de treinamento.

Algoritmo	Classificações erradas (total de 200)
<i>Backpropagation</i>	16
Momentum	16
Levenberg-Marquardt	15
RProp	12
Adaptive Learning Rate 1	17
Adaptive Learning Rate 2	16
SMC2	11

A Tabela 6.10 mostra o total de vetores classificados erradamente considerando-se apenas o conjunto de treinamento, para cada um dos algoritmos utilizados.

A Figura 6.17 mostra a linha de decisão encontrada por cada um dos algoritmos utilizados no treinamento. Cada linha de decisão divide o plano $x_1 \times x_2$ em duas partes. A parte esquerda do plano representa a classe $C1$ e a parte direita representa a classe $C2$.

Após o completo treinamento das RNAs, utilizou-se o conjunto de dados para a validação, amostrados das classes $C1$ e $C2$. A Tabela 6.11 mostra o percentual de erro para cada uma das redes. Como pode ser observado, o índice de erro de todos os algoritmos utilizados é, aproximadamente, igual.

Observa-se, pela Tabela 6.11 que o algoritmo de Levenberg-Marquadt apresentou o maior índice de erro, apesar de que este índice é muito próximo do índice dos demais algoritmos.

O algoritmo SMC2 apresentou a maior velocidade de treinamento (Figuras 6.15 e 6.16) e o menor índice de erro para o conjunto de treinamento, apresentando um índice de erro para o conjunto de validação equivalente ao índice dos demais algoritmos, excluindo-se o algoritmo de Levenberg-Marquadt.

6.4 Controle Vetorial Direto

Os controladores normalmente utilizados em sistemas de controle (controladores proporcional e/ou integral e/ou diferencial, controladores adaptativos, controle robusto)

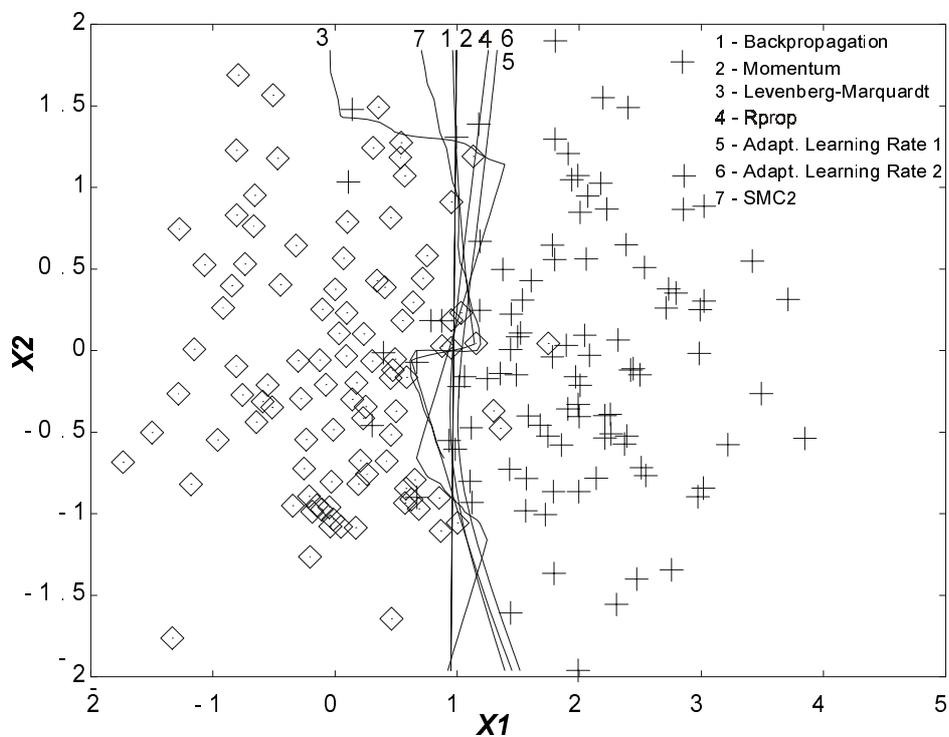


Figura 6.17- Linha de decisão encontrada pelos algoritmos de treinamento utilizados no problema de classificação.

Tabela 6.11: Erro de classificação para o conjunto de validação.

Algoritmo	Erro de classificação (%)
<i>Backpropagation</i>	10.20
Momentum	10.20
Levenberg-Marquardt	11.15
RProp	10.65
Adaptive Learning Rate 1	10.45
Adaptive Learning Rate 2	10.80
SMC2	10.75

normalmente necessitam de um conhecimento prévio da planta que se deseja controlar. Além deste fato, o projeto de alguns controladores consideram a planta como um sistema linear. Muitas vezes é necessário um considerável gasto de tempo para o ajuste dos controladores utilizados. Estas características dificultam a aplicação dos sistemas de controle às plantas.

RNAs, por sua vez, se devidamente treinadas, apresentam-se como uma interessante ferramenta a ser aplicada em sistemas de controle, sejam como observadores de algum estado ou variável necessária ao controle ou como elementos responsáveis pela geração do sinal de controle. As características de generalização e predição conferem às RNAs a capacidade de responderem a uma determinada entrada para a qual elas não foram previamente treinadas, podendo, inclusive, atuarem como elementos filtrantes de sinais espúrios ao controle. Entretanto, para manter a capacidade da RNA de se adaptar às variações paramétricas da planta, é necessário manter um treinamento contínuo da RNA utilizada. Normalmente a RNA é treinada *off-line*, sempre que necessário. Entretanto, este tipo de operação requer que o sistema de controle seja desativado durante a etapa de treinamento, além de ser necessário armazenar uma grande quantidade de dados, durante o funcionamento normal do sistema, para o posterior treinamento.

Por outro lado, a utilização de algoritmos de treinamento *on-line* acarretam um aumento da complexidade do sistema de controle e do tempo de processamento requerido pelo controle. Apesar deste fato, a RNA tem a possibilidade de se adaptar dinamicamente às variações paramétricas da planta a ser controlada, evitando, desta forma, uma redução na qualidade do controle. Evita-se, também, a necessidade de desligamento da planta para uma etapa de treinamento *off-line*, possibilitando um funcionamento ininterrupto do sistema de controle. O algoritmo de controle *on-line* deve possibilitar uma rápida convergência da RNA, sem onerar muito o tempo de processamento do controle. Os algoritmos de treinamento *on-line*, propostos neste trabalho (Seção 4.3 e 4.4), serão utilizados no controle vetorial direto do motor de indução. Objetiva-se obter um sistema de acionamento que seja o mais independente possível dos parâmetros da máquina utilizada.

A Figura 6.18 mostra o sistema de controle vetorial direto, orientado pelo fluxo de estator, a ser implementado [Neves, 1999]. Neste sistema de controle, as variáveis a serem

controladas são o módulo do fluxo ($|\phi_s|$) e a velocidade do rotor do motor (ω_r). O motor de indução, por ser um sistema não linear e com parâmetros variantes no tempo, é uma interessante aplicação de sistemas de controle utilizando RNAs.

O modelo matemático da máquina de indução, em referencial fixo no vetor fluxo de estator (sistema de eixos dq), pode ser descrito pelo seguinte conjunto de equações:

$$\vec{v}_s = R_s \cdot \vec{i}_s + \dot{\vec{\lambda}}_s \quad (6.1)$$

$$0 = R_r \cdot \vec{i}_r + \dot{\vec{\lambda}}_r - j \cdot \omega_r \cdot \vec{\lambda}_r \quad (6.2)$$

$$\vec{\lambda}_s = L_s \cdot \vec{i}_s + L_m \cdot \vec{i}_r \quad (6.3)$$

$$\vec{\lambda}_r = L_r \cdot \vec{i}_r + L_m \cdot \vec{i}_s \quad (6.4)$$

$$J \cdot \frac{d\omega_r}{dt} = T_e - T_l \quad (6.5)$$

Considerando-se a perfeita orientação pelo fluxo de estator, o fluxo de eixo q será nulo e a Equação 6.1 pode ser reescrita da seguinte maneira:

$$v_{sd} = R_s \cdot i_{sd} + \dot{\lambda}_{sd}$$

considerando-se o sistema amostrado e a regra de Euler, tem-se:

$$\lambda_{sd(k+1)} = \lambda_{sd(k)} + (v_{sd(k)} - R_s \cdot i_{sd(k)}) \cdot T$$

onde T é o período de amostragem.

Desta forma, considerando-se que se deseja controlar o fluxo de eixo direto de estator, a variável $\lambda_{sd(k+1)}$ torna-se o fluxo de referência (λ_{sd}^*) e, portanto, a equação acima será reescrita como:

$$v_{sd(k)} = \frac{\lambda_{sd}^* - \lambda_{sd(k)}}{T} + R_s \cdot i_{sd(k)} \quad (6.6)$$

Uma rápida análise da equação 6.6 indica que a tensão de eixo direto de estator

a ser aplicada na máquina é uma função tanto do erro existente entre o fluxo de eixo direto e o fluxo de referência de estator quanto da corrente de estator de eixo direto.

Considerando-se o sistema de eixos fixos no vetor fluxo de estator, pode-se escrever a seguinte equação de torque:

$$T_e = \frac{3}{2} \cdot \frac{P}{2} \cdot \lambda_{sd} \cdot i_{sq}$$

Utilizando-se a equação acima na Equação 6.5 e considerando-se o sistema amostrado, tem-se:

$$J \cdot \frac{\omega_r(k+1) - \omega_r(k)}{T} = \frac{3}{2} \cdot \frac{P}{2} \cdot \lambda_{sd(k)} \cdot i_{sq(k)} - T_{l(k)}$$

Considerando-se que se deseja controlar a velocidade do motor, a variável $\omega_r(k+1)$ torna-se a velocidade de referência (ω_r^*) e, ainda, considerando a perfeita orientação do fluxo, pode-se escrever a equação acima como:

$$v_{sq(k)} = J \cdot \frac{\omega_r^* - \omega_r(k)}{T} \cdot \frac{2}{3} \cdot \frac{2}{P} \cdot \frac{R_s}{\lambda_{sd(k)}} + T_{l(k)} \quad (6.7)$$

Analisando-se a Equação 6.7, observa-se que a tensão de estator de eixo em quadratura a ser aplicada à máquina é uma função do erro de velocidade, do fluxo de estator de eixo direto e do conjugado de carga presente no eixo da máquina, além de alguns outros parâmetros da máquina.

Para a implementação do sistema de controle proposto, é necessária a medição ou observação do fluxo de rotor (módulo e ângulo do fluxo de rotor) e da velocidade. A medição do fluxo pode ser realizada, além da medição através de sensores, com a implementação de um estimador ou de um observador de fluxo [Verghese & Sanders, 1988]. A estratégia do controle a ser utilizada neste trabalho envolve o controle de fluxo com observação e o controle da velocidade com medição direta.

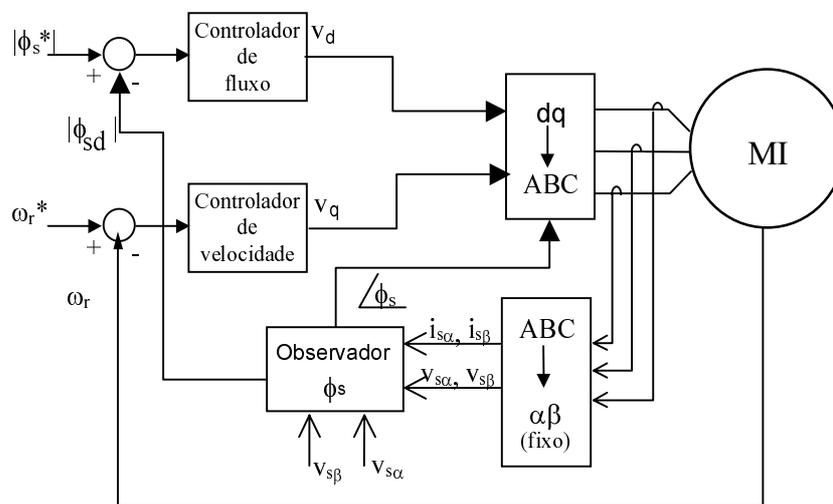


Figura 6.18- Sistema de controle a ser implementado.

6.4.1 Observador Neural de Fluxo Estatórico

Como pode ser observado, o sistema de controle mostrado na Figura 6.18 utiliza a orientação dos eixos segundo o fluxo de estator. Desta forma, desenvolveu-se um observador de fluxo estatórico. Apesar do sistema de controle vetorial baseado no fluxo de estator apresentar um acoplamento entre as variáveis de controle, a determinação do fluxo de estator é menos dependente das grandezas da máquina, além de ser uma variável mais acessível do que as variáveis de rotor.

O observador implementado está mostrado na Figura 6.19, tendo sido apresentado no XII Congresso Brasileiro de Automática [Parma et al., 1998c]. Os resultados aqui apresentados referem-se à simulação do observador.

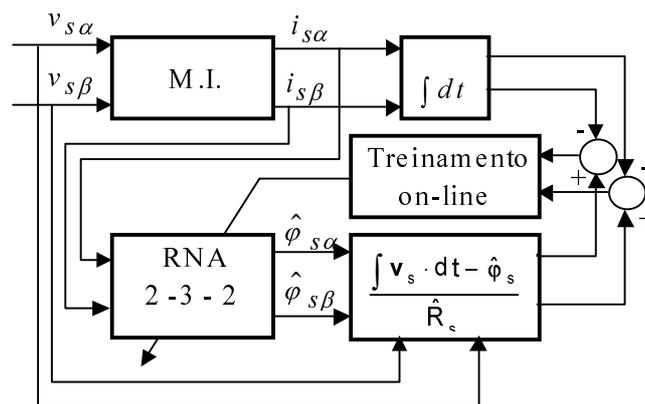


Figura 6.19- Diagrama esquemático do observador neural de fluxo estatórico implementado.

Como observa-se na Figura 6.19, o observador implementado necessita apenas da resistência estatórica, a qual poder ser facilmente determinada através de ensaios no motor [Parma et al., 1997b].

Foi utilizada uma MLP com duas entradas, três nodos na camada escondida e duas saídas (Figura 6.19). A RNA é treinada com a segunda proposta de treinamento *on-line* (Seção 4.4). Os pesos iniciais da RNA são escolhidos aleatoriamente de um conjunto de dados com distribuição normal e média nula.

O observador foi testado em frequência nominal e frequência reduzida de operação, Figuras 6.20 e 6.21, respectivamente. Observa-se que não existem mudanças significativas na resposta do observador para frequência nominal ou para frequência reduzida. Sabe-se que um dos principais problemas de observadores é com relação à operação em frequência reduzida [Verghese & Sanders, 1988].

A numeração das curvas mostradas nas Figuras 6.20, 6.21, 6.22 e 6.23 refere-se à mostrada na Tabela 6.12.

Tabela 6.12: Identificação das curvas mostradas do observador neural de fluxo estatórico.

numeração	Curva
1	Fluxo real
2	Fluxo observado

Para avaliar a dependência do observador com a resistência estatórica, foi realizada uma variação de 20% em seu valor. Sabe-se que essa variação não ocorre em experimentos práticos, tanto com relação à forma quanto com relação à amplitude da variação. Em casos práticos a variação ocorre de maneira gradual, sendo que foi implementada uma variação em degrau.

As Figuras 6.22 e 6.23 mostram a resposta do observador para uma variação de 20% no valor da resistência estatórica, tanto em frequência nominal de operação quanto em frequência reduzida, respectivamente. Pode-se observar que o observador é bastante robusto frente a uma variação na resistência estatórica.

6.4.2 Resultados de Simulação

Implementou-se o sistema mostrado na Figura 6.18, utilizando-se um observador de fluxo de estator de segunda ordem, proposto por Neves [Neves, 1999] e utilizando duas redes MLP no sistema de controle, uma atuando como controlador de fluxo e outra atuando como controlador de velocidade. Ambas as redes possuem duas entradas e uma saída, utilizando a função tangente hiperbólica como função de ativação tanto na camada escondida quanto na camada de saída. As MLPs utilizadas como controlador de fluxo e controlador de velocidade possuem, respectivamente, três e oito neurônios na camada escondida.

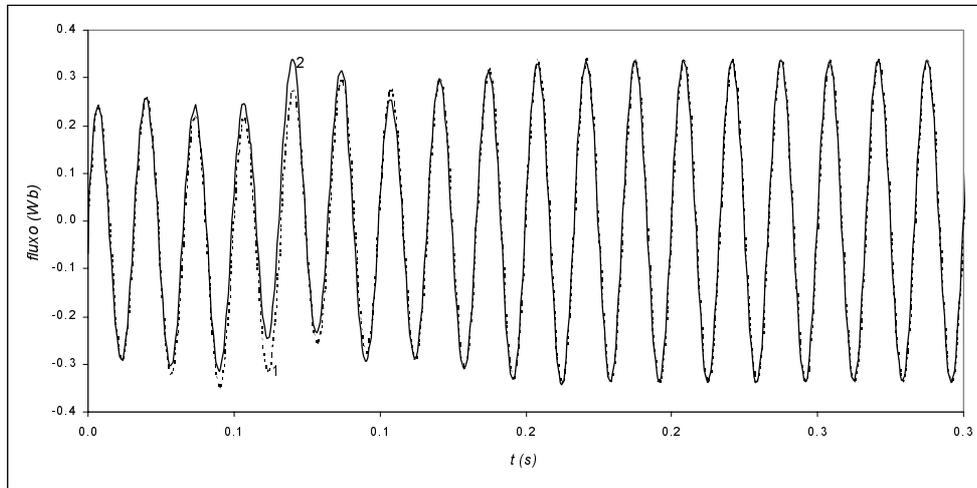


Figura 6.20- Fluxo real e observado para uma frequência de alimentação de 60Hz.

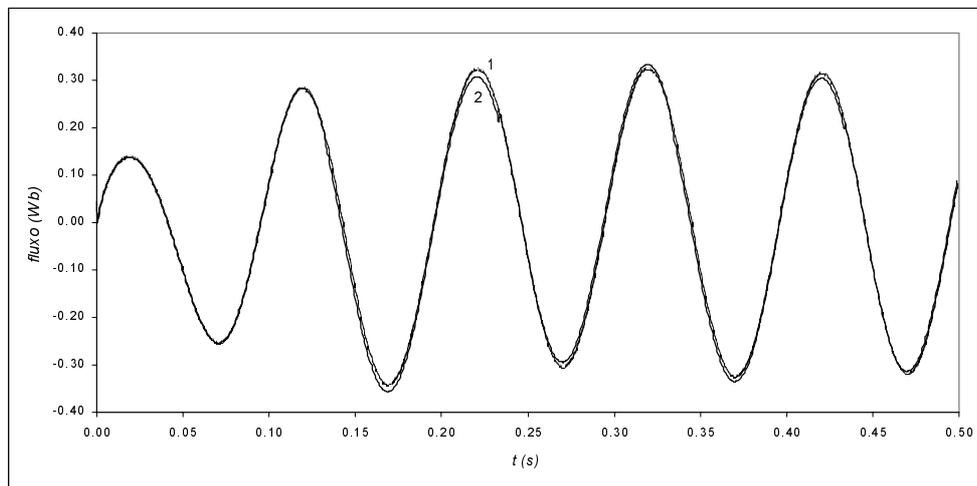


Figura 6.21- Fluxo real e observado para uma frequência de alimentação de 10Hz.

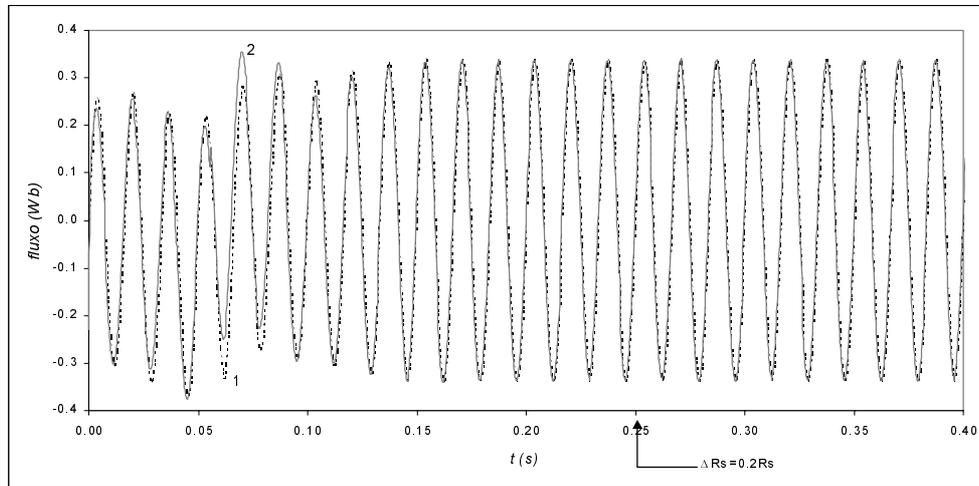


Figura 6.22- Fluxo real e observado para uma freqüência de alimentação de 60Hz. Variação de 20% na resistência estática em $t=0.25s$.

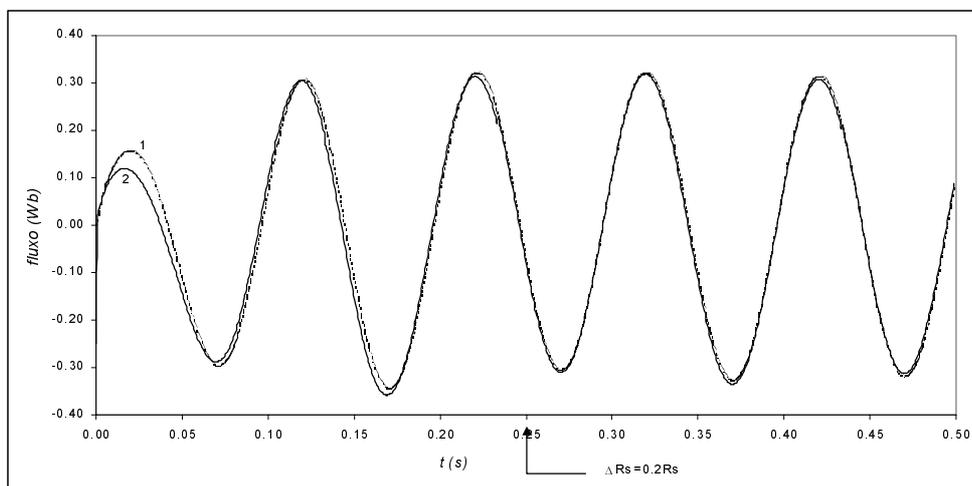


Figura 6.23- Fluxo real e observado para uma freqüência de alimentação de 10Hz. Variação de 20% na resistência estática em $t=0.25s$.

Utilizou-se o erro de fluxo, normalizado em função do fluxo de referência, e a corrente de eixo d como entradas da MLP de controle de fluxo. A saída desta rede é a tensão de eixo d . Para a MLP de controle de velocidade utilizou-se o erro de velocidade e a velocidade de referência, ambos normalizados em função da máxima velocidade elétrica do motor (Figura 6.24). A escolha das variáveis de entrada utilizadas pelas redes foi feita em função da análise das Equações 6.6 e 6.7 sendo que, para esta última, considerando-se que o fluxo de eixo direto esteja perfeitamente controlado, a tensão de estator de eixo em quadratura é uma função apenas do erro de velocidade. Todos os demais parâmetros são admitidos como parâmetros a serem modelados pela RNA.

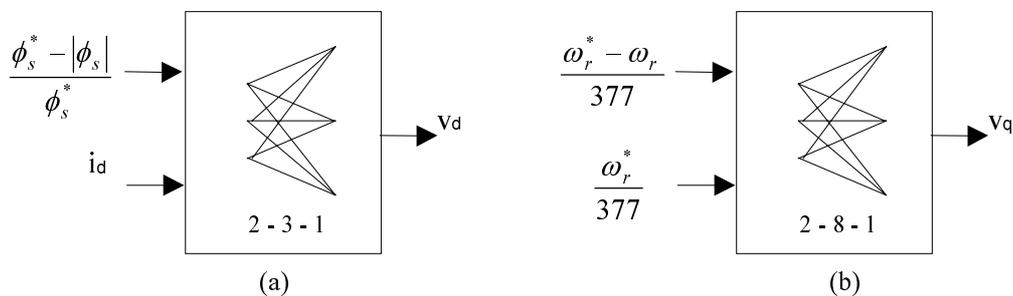


Figura 6.24- Topologia MLP utilizadas como: (a) controlador de fluxo e (b) controlador de velocidade.

O programa de simulação considera a alimentação da máquina de indução utilizando PWM simétrico operando a uma frequência de $8kHz$. Apesar do sistema de simulação trabalhar com um passo de integração, para a simulação da máquina, igual à $1\mu s$, o sistema de controle opera com uma frequência de $4kHz$.

São apresentados resultados de reversão de velocidade e aplicação e retirada de carga.

Os pesos das RNAs utilizadas são amostrados de uma distribuição gaussiana de média nula, quando da primeira partida da máquina. A partir de então, as RNAs são treinadas na operação normal da máquina de indução, utilizando-se a primeira proposta de treinamento *on-line* (seção 4.3). A cada nova partida da máquina, utilizam-se os pesos das RNAs treinadas anteriormente. Todo o algoritmo de simulação foi implementado em linguagem C.

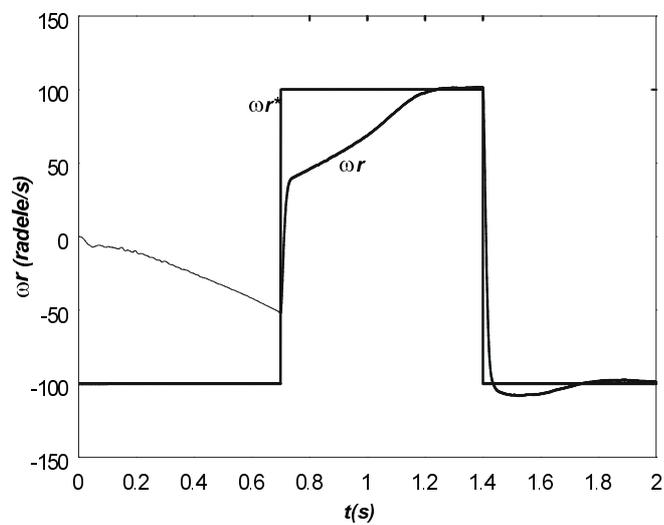


Figura 6.25- Velocidade de referência (± 100 *rad/s*) e medida. Partida e reversão de velocidade. Primeira partida, pesos da RNA amostrados de uma distribuição normal de média nula.

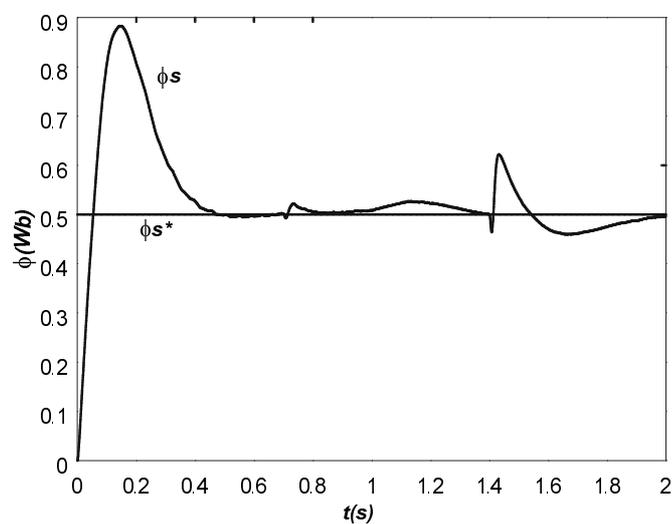


Figura 6.26- Fluxo de estator de eixo direto de referência e fluxo simulado. Partida e reversão de velocidade. Primeira partida da máquina, pesos da RNA amostrados de uma distribuição normal de média nula.

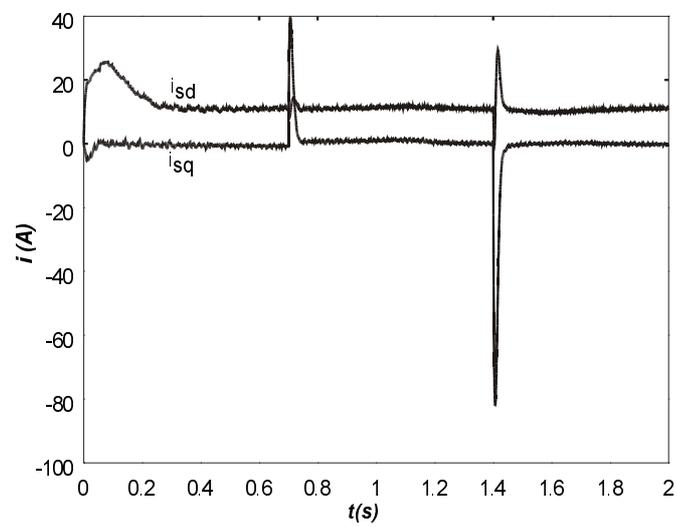


Figura 6.27- Corrente de estator de eixo direto e eixo em quadratura. Partida e reversão de velocidade. Primeira partida da máquina, pesos da RNA amostrados de uma distribuição normal de média nula.

A Figura 6.25 mostra a velocidade de referência e a velocidade medida para a primeira partida do motor de indução. Para esta partida, os pesos das MLPs foram amostrados de uma distribuição normal de média nula. Observa-se que, na primeira partida, a máquina não consegue atingir a velocidade de referência e, logo após, é feita uma reversão de velocidade (em $t = 0.7s$). A Figura 6.26 mostra o fluxo de estator de eixo direto de referência e o fluxo da máquina simulada. Observa-se que o fluxo, na primeira partida, atinge o valor de referência com um sobre-sinal. Na simulação não foi considerada a saturação do fluxo na máquina. Observa-se também que o fluxo sofre perturbações quando da reversão de velocidade. Estas perturbações podem ser melhor observadas na Figura 6.27, a qual mostra as correntes de estator de eixo direto e em quadratura. Pode-se observar que existe um certo acoplamento entre os dois eixos.

A Figura 6.28 mostra o comportamento da velocidade da máquina de indução na quinta partida do motor. Nesta partida, os pesos das MLPs não são mais amostrados de uma distribuição normal de média nula. Neste caso, os pesos são oriundos das partidas anteriores, mostrando a capacidade de aprendizado do sistema. Observa-se que, agora, antes da primeira reversão ($t = 0.75s$), a máquina já atinge a velocidade de referência. Durante todas as reversões seguintes, o sistema de controle mostra uma boa velocidade de resposta. É importante ressaltar que, durante a partida, existe uma maior variação na velocidade pois o fluxo de estator ainda não está totalmente estabelecido. A Figura 6.29 mostra o fluxo de estator de eixo direto para a quinta partida da máquina. Observa-se uma pequena redução no sobre-sinal do fluxo na partida e uma redução nas oscilações quando ocorre uma redução na velocidade da máquina.

A Figura 6.30 mostra as correntes de estator de eixo direto e em quadratura para a quinta partida da máquina. Observa-se que ainda existe um acoplamento entre os eixos. Entretanto, mesmo com este acoplamento, o sistema de controle consegue manter a velocidade medida acompanhando a velocidade de referência, como pode ser observado pela Figura 6.28.

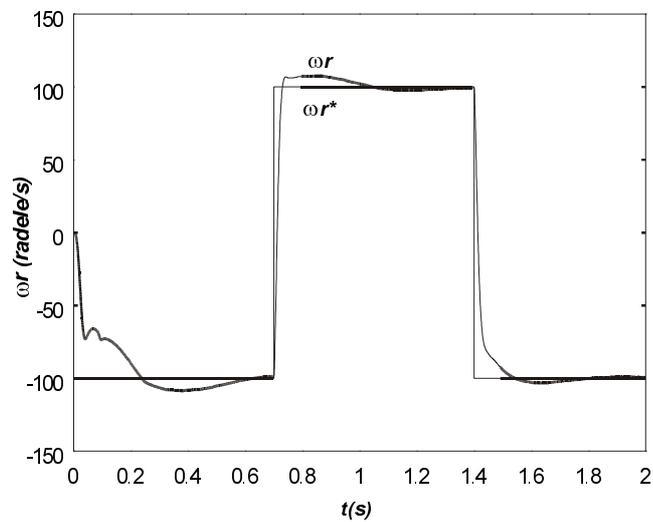


Figura 6.28- Velocidade de referência (± 100 rad/s) e medida. Partida e reversão de velocidade. Quinta partida, pesos da RNA provenientes das partidas anteriores.

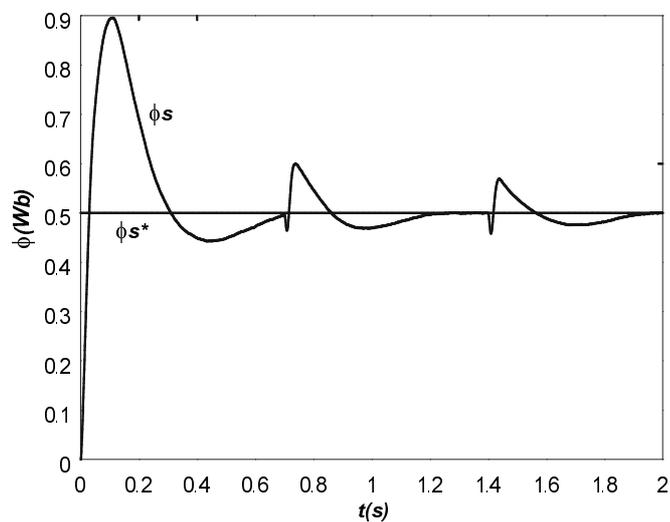


Figura 6.29- Fluxo de estator de eixo direto de referência e fluxo simulado. Partida e reversão de velocidade. Quinta partida da máquina, pesos da RNA oriundos das partidas anteriores.

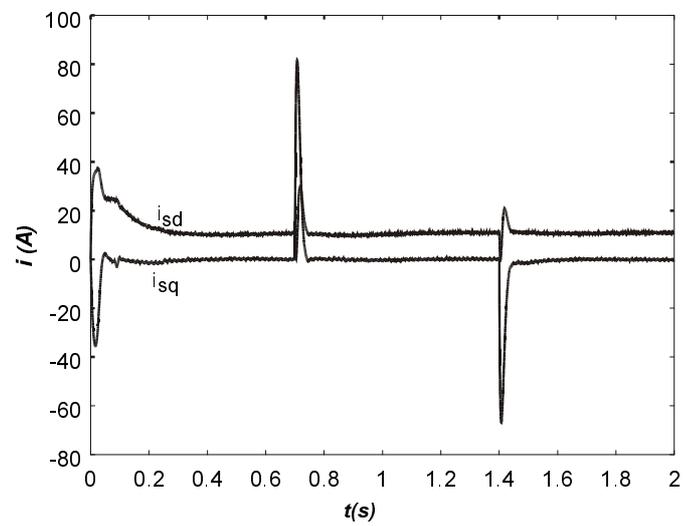


Figura 6.30- Corrente de estator de eixo direto e eixo em quadratura. Partida e reversão de velocidade. Quinta partida da máquina, pesos da RNA oriundos das partidas anteriores.

6.4.3 Controlador Neural de Velocidade

Apesar do sistema de acionamento proposto utilizar a orientação pelo fluxo de estator (Figura 6.18), em uma primeira implementação prática utilizou-se o diagrama de acionamento mostrado na Figura 6.31. Neste acionamento utiliza-se a orientação pelo fluxo de rotor [Novotny & Lipo, 1996][Bose, 1986], ao contrário da alimentação pelo fluxo de estator. Entretanto, nesta implementação (Figura 6.31), é necessária a presença de controladores de corrente, o que pode sobrecarregar o sistema de controle. Mas, por outro lado, com a presença de controladores de corrente pode-se ter um controle direto sobre a corrente do motor, possibilitando a limitação desta corrente.

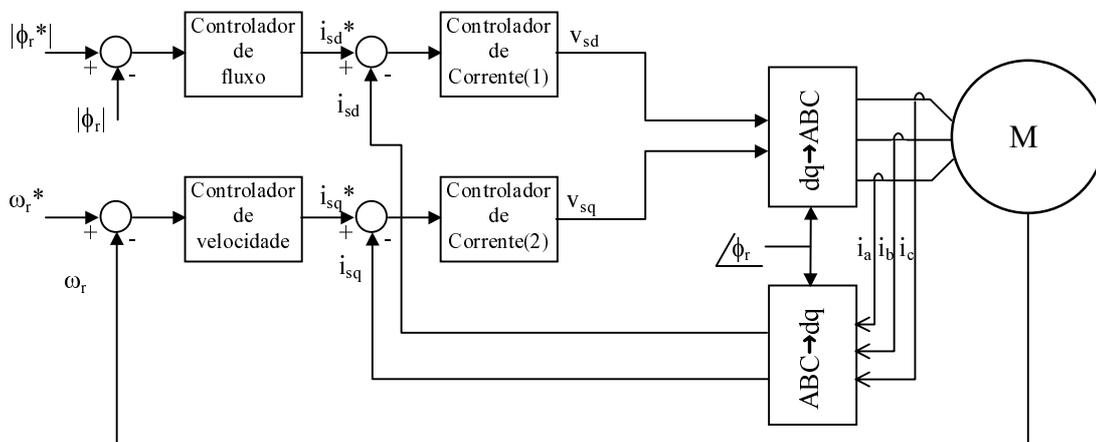


Figura 6.31- Diagrama esquemático do controle vetorial direto do motor de indução utilizando sistema de eixos orientados pelo fluxo de rotor.

Continuando com o desenvolvimento do controle vetorial direto, foi implementado um controlador neural de velocidade, mostrado na Figura 6.32. Utilizou-se uma MLP com duas entradas, cinco nodos na camada escondida e uma saída. As entradas são a velocidade de referência (ω_r^*) e o erro de velocidade ($\omega_r^* - \omega_r$). A saída da RNA é o conjugado de referência, existindo, portanto, uma malha de controle além das mostradas na Figura 6.31. A saída da malha de conjugado é a corrente de estator de eixo q (i_{sq}^*). Todos os demais controladores implementados são do tipo Proporcional e Integral (PI). Os ganhos dos controladores PI foram ajustados para uma determinada máquina, enquanto que os pesos iniciais da RNA utilizada foram escolhidos aleatoriamente de um conjunto de

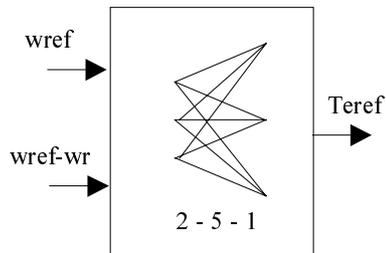


Figura 6.32- MLP utilizada como controlador de velocidade no controle vetorial direto.

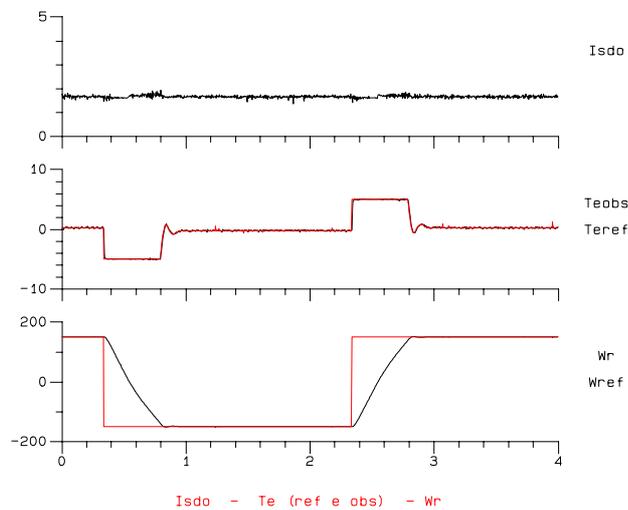


Figura 6.33- Curvas de resposta do controle implementado. Observa-se a curva de velocidade real e de referência (W_r e W_{ref} , respectivamente), conjugado observado e de referência (T_{eobs} e T_{eref} , respectivamente) e corrente de eixo direto (I_{sdo} , responsável pelo fluxo).

dados com distribuição normal e média nula.

Para o treinamento da RNA, utilizou-se a primeira proposta para treinamento *on-line*, descrita na Seção 4.3. Como não se conhece a saída desejada da RNA, o sinal de erro de velocidade foi utilizado para a determinação da superfície de deslizamento. Conforme descrito na seção 4.3, foi necessário implementar uma limitação no ajuste dos pesos para se evitar problemas numéricos durante a execução do algoritmo.

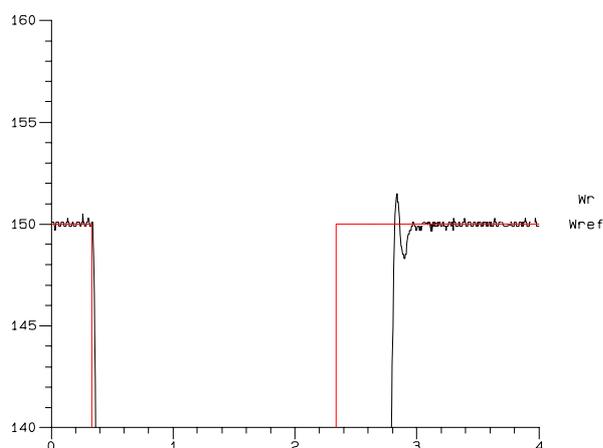


Figura 6.34- Ampliação da curva de velocidade da Figura anterior. Observa-se o pequeno sobre-sinal existente.

As Figuras 6.33, 6.34 e 6.35 são resultados experimentais obtidos no laboratório do GAEEP (Grupo de Acionamentos Elétricos e Eletrônica de Potência), durante uma reversão de velocidade do motor utilizado.

Na Figura 6.33 estão mostradas as curvas da corrente de estator de eixo direto (I_{sdo}), conjugado elétrico observado e de referência (T_{eobs} e T_{eref}) e velocidade do rotor de referência e real (W_r e W_{ref}). A saída do controlador neural de velocidade é o conjugado elétrico de referência (T_{eref}). Pode-se observar a rápida resposta do controlador, com uma pequena oscilação. Na Figura 6.34 é mostrada uma ampliação da curva de velocidade real do motor e de referência.

A Figura 6.35 mostra o gráfico do conjugado do motor em função da velocidade, mostrando a velocidade de resposta do controlador neural utilizado.

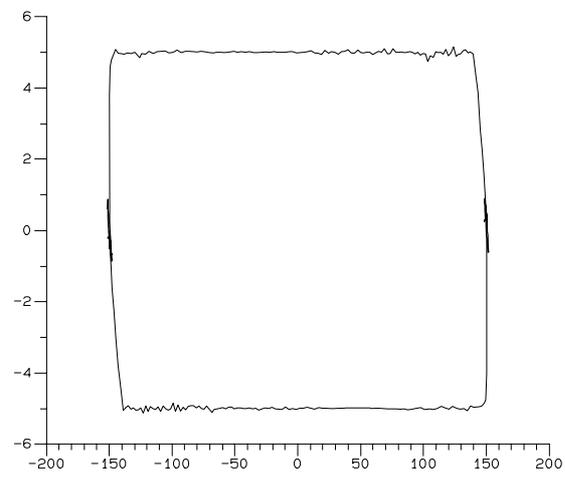


Figura 6.35- Gráfico do conjugado em função da velocidade do motor.

Observa-se pela resposta experimental do controle vetorial direto que está sendo implementado, a aplicabilidade de RNAs em sistemas de controle. Por ser um sistema adaptativo, não foi necessário o prévio ajuste dos parâmetros do controlador, como é necessário com, por exemplo, os controladores PI utilizados no sistema. Comparando com resultados mostrados na literatura [Stopa, 1997], os resultados obtidos até este momento são considerados bons, apresentando uma boa velocidade de resposta transitória e facilidade de implementação.

6.5 Conclusões

Neste capítulo foi feita a comparação dos algoritmos propostos com algoritmos encontrados na literatura, sendo que os algoritmos propostos apresentaram os melhores resultados nos problemas de aproximação de função e classificação, dois problemas típicos na área de RNAs.

Apresentou-se, também, um sistema de acionamento de máquinas de indução utilizando MLPs. Este sistema requer apenas o conhecimento da resistência estatórica para a determinação do fluxo de estator. Os resultados de simulação mostram que é possível desenvolver um sistema de acionamento capaz de se adaptar a máquina que está sendo controlada, utilizando os algoritmos de treinamento *on-line* propostos.

Capítulo 7

Conclusões e Propostas de Continuidade

Neste capítulo serão sumarizadas as principais conclusões obtidas deste trabalho, bem como as propostas de continuidade a serem desenvolvidas.

A primeira conclusão diz respeito a utilização de uma teoria de controle no problema de treinamento de Redes Neurais Artificiais (RNAs). Com a utilização da teoria de Controle por Modos Deslizantes (CMD), a RNA passa a ser analisada como um sistema a ser controlado, no qual as variáveis de controle são os pesos e as variáveis de referência devem ser acompanhadas pela saída da RNA. Desta forma, desenvolve-se um controlador dedicado ao treinamento da RNA. A metodologia empregada possibilita o estudo dos limites dos ganhos envolvidos nos algoritmos de treinamento propostos. Esta metodologia de abordagem do problema de treinamento de RNAs não era, até então, encontrada na literatura.

A superfície de deslizamento, definida nos algoritmos propostos, garante a convergência do sistema se os parâmetros envolvidos no treinamento permanecerem dentro dos limites apresentados. Os valores dos parâmetros de treinamento podem ser determinados a cada iteração, devendo permanecer, sempre, dentro dos limites definidos e possibilitando uma aceleração no treinamento. Como observa-se pelos resultados obtidos, os algoritmos propostos são de fácil implementação e rápido processamento. A inclusão do somatório dos gradientes no ajuste dos pesos da RNA (segunda proposta de treinamento *off-line*) possibilitou, ao algoritmo proposto, o treinamento de MLPs em um número de *epochs* inferior àquele requerido por outros algoritmos encontrados na literatura, enquanto a superfície de deslizamento garante a convergência do treinamento.

Observa-se que a inclusão da superfície de deslizamento na determinação da direção de ajuste dos pesos propicia uma maior aceleração no treinamento da RNA (Figura 6.2). Desta forma, mesmo utilizando sinais provenientes do algoritmo *Backpropagation*, pode-se obter um algoritmo mais rápido, apenas com a modificação da direção de ajuste dos pesos, como é feito na primeira proposta de treinamento *off-line*. Podem-se, entretanto, obter melhores resultados se, além da direção de ajuste dos pesos, modificar-se o módulo do ajuste, como é feito na segunda proposta de treinamento *off-line*. Observa-se, também, a dependência do algoritmo *Backpropagation* com a escolha inicial dos pesos da RNA e a robustez dos algoritmos propostos em função destes parâmetros iniciais.

Apesar do algoritmo de Levenberg-Marquardt apresentar o menor número de *epochs* para o treinamento da RNA utilizada no problema de aproximação de função, o tempo de processamento gasto nestas *epochs* é superior ao tempo de processamento gasto pelos algoritmos propostos. A tabela abaixo mostra o tempo gasto para o processamento de uma *epoch* considerando o algoritmo de Levenberg-Marquardt e as duas propostas de treinamento *off-line*. Observa-se que o tempo gasto pelo algoritmo de Levenberg-Marquardt para o processamento de uma *epoch* é, aproximadamente, 10 e 13 vezes superior ao tempo gasto pelos algoritmos propostos para a primeira e segunda função, respectivamente.

Algoritmo	<i>tempo/epoch</i>	
	$f_1(x)$	$f_2(x)$
Levenberg-Marquardt	0.030	0.046
SMC1	0.003	0.0033
SMC2	0.003	0.0033

No problema de classificação proposto, a segunda proposta de treinamento *off-line* apresentou os melhores resultados dentre todos os algoritmos, com o menor número de *epochs* e menor erro para o conjunto de treinamento. Para a validação do treinamento, todos os algoritmos obtiveram, aproximadamente, a mesma taxa de erro. Pode ser observado que o algoritmo de Levenberg-Marquardt apresentou a superfície de decisão que mais difere da superfície encontrada pelos demais algoritmos de treinamento, apresentando um erro de classificação para o conjunto de treinamento superior ao erro obtido pelo algoritmo Rprop e pela segunda proposta de treinamento (Figura 6.17).

Os algoritmos de treinamento *on-line* foram desenvolvidos para aplicações em tempo real e os resultados obtidos mostram a capacidade de adaptação da RNA a uma determinada situação, tendo sido comparados apenas ao algoritmo *Backpropagation*, pois os demais algoritmos não foram desenvolvidos para este tipo de aplicação.

O resultado obtido com o algoritmo proposto para treinamento *on-line* mostra que o erro mínimo obtido pelo algoritmo é diferente do erro mínimo alcançado com o algoritmo *Backpropagation*, apresentando um melhor resultado quando o algoritmo de treinamento é desabilitado (Figura 6.12).

A bancada experimental desenvolvida é composta por diversos sistemas de medição das principais grandezas de interesse em um acionamento elétrico. Esta bancada possibilitará, portanto, a implementação e comparação de diversas técnicas de controle. Além deste fato, será possível a aferição de alguns sistemas de observadores e estimadores de grandezas envolvidas no acionamento. Será possível, também, a análise da influência de técnicas de medição distintas, de uma mesma grandeza elétrica, no sistema de acionamento em estudo.

O sistema de controle que está sendo implementado para o motor de indução, além de apresentar resultados satisfatórios, é de fácil implementação. Os resultados obtidos, em simulação, para o observador neural de fluxo estatórico encorajam a sua implementação prática, tendo apresentado bons resultados tanto em velocidade nominal quanto em velocidade reduzida. O observador proposto se mostrou bastante robusto frente a variações paramétricas, principalmente por depender apenas da resistência estatórica, parâmetro de mais acessibilidade na máquina de indução.

Os resultados de simulação obtidos com o sistema de acionamento *Direct Torque and Flux Control* (DTFC) proposto, em conjunto com o observador neural, mostram que é possível desenvolver um sistema de acionamento que seja dependente do menor número de parâmetros da máquina. Um dos principais problemas no desenvolvimento de sistemas de acionamento reside na variação paramétrica da máquina de indução e, portanto, um sistema que seja praticamente independente dos parâmetros da máquina de indução não deverá apresentar os problemas resultantes desta variação paramétrica. Desta forma, o

sistema de acionamento proposto será capaz de se adaptar à máquina e às suas variações paramétricas.

Os resultados experimentais obtidos mostram a capacidade da RNA em controlar o motor de indução, independentemente da potência do motor, não tendo sido necessário o ajuste prévio dos parâmetros da rede. Todos os pesos são ajustados durante a operação normal da rede, possibilitando que a RNA se adapte à situação atual do motor de indução.

7.1 Propostas de Continuidade

Observando-se os principais pontos de contribuição deste trabalho, diversas propostas de continuidade podem ser formuladas.

Considerando-se tanto a utilização da metodologia aplicada ao treinamento de MLPs quanto a bancada experimental desenvolvida, podem-se listar as seguintes propostas de continuidade:

1. A princípio, propõem-se a implementação dos algoritmos propostos, para treinamento *on-line* e *off-line*, utilizando ganho variável. Espera-se, com isto, uma redução ainda maior do tempo de treinamento das MLPs;
2. Aplicação da metodologia matemática apresentada para a aplicação dos algoritmos propostos no treinamento de outras topologias de RNAs e não somente para MLPs;
3. Implementação do sistema de acionamento proposto e simulado neste texto, tendo como base a bancada desenvolvida;
4. Implementação de um sistema de acionamento genérico na qual se possam comparar metodologicamente todas as técnicas de medição encontradas na literatura e implementadas no *hardware*, a começar pelas técnicas de medição de tensão;
5. Aferição de alguns dos observadores e estimadores de fluxo e torque, encontrados na literatura, utilizando-se o sistema de medição de fluxo e torque existente no *hardware*;

6. Desenvolvimento de um sistema de acionamento sem sensor de velocidade, utilizando-se uma RNA como observador de velocidade.

Referência Bibliográfica

- [Battiti & Massuli, 1990] Battiti, R. & Massuli, F. (1990). BFGS optimization for faster and automated supervised learning. In *Proceedings of the International Neural Network Conference (Paris, 1990)*, volume 2 (pp. 757–760).
- [Bertsekas & Tsitsiklis, 1996] Bertsekas, D. P. & Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.
- [Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.
- [Bose, 1986] Bose, B. K. (1986). *Power Electronics and AC drives*. New Jersey: Prentice-Hall, Englewood Cliffs.
- [Braga et al., 2000] Braga, A. P., Ludermir, T. B., & de Leon Ferreira Carvalho, A. C. P. (2000). *Redes Neurais Artificiais: Teoria e aplicações*. Brasil: Livros Técnicos e Científicos.
- [Butkov, 1978] Butkov, E. (1978). *Física Matemática*. Editora Guanabara dois S.A.
- [Chin et al., 1998] Chin, T.-H., Nakano, M., & Hirayama, T. (1998). Accurate neasurement of instantaneous values of voltage, current and power for power electronics circuits. In *PESC'98 - 29th Annual IEEE Power Electronics Specialists Conference* (pp. 302–307). USA.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4), 303–314.
- [Darken & Moody, 1991] Darken, C. & Moody, J. (1991). Note on learning rate schedules for stochastic optimization. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in Neural Information Processing Systems 3. Proceedings of the 1990 Conference* (pp. 832–838). San Mateo, CA: Morgan Kaufmann.
- [Demuth & Beale, 1994] Demuth, H. & Beale, M. (1994). *Neural Network Tollbox User's guide - For use with MATLAB*. The Math Works Inc., 3 edition.
- [Fahlman, 1988a] Fahlman, S. E. (1988a). *An Empirical Sudy of Learning Speed in Back-Propagation Networks*. Technical Report CMU-CS-88-162, CMU.
- [Fahlman, 1988b] Fahlman, S. E. (1988b). Faster-learning variations on back-propagation: an empirical study. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist*
-

-
- Models Summer School, Pittsburg* (pp. 38–51). San Mateo, CA: Morgan Kaufmann.
- [Fu, 1996] Fu, L.-C. (1996). Neural network approach to variable structure based adaptive tracking of siso systems. In *1996 IEEE International Workshop on Variabel Structure Systems - VSS'96* (pp. 148–153). Tokyo, Japan.
- [Hagan & Menhaj, 1994] Hagan, M. T. & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989–993.
- [Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior*. New York: Wiley.
- [H.Hassoun, 1995] H.Hassoun, M. (1995). *Fundamentals of Artificial Neural Networks*. Cambridge, Massachusetts: The MIT Press.
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational facilities. *Proceedings of the National Academy of Sciences of the USA*, 79, 2554–2558.
- [Itkis, 1976] Itkis, U. (1976). *Control Systems of variable structure*. New York: John Wiley and Sons.
- [K. et al., 1997] K., J., M., R., R., S., & B., C. (1997). Neural network sliding mode robot control. *ROBOTIC*, 15, 23–30.
- [Kohonen, 1982] Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps. *Biol. Cyb.*, 43(1), 59–69.
- [Kramer & S.-Vincentelli, 1989] Kramer, A. H. & S.-Vincentelli, A. (1989). Efficient parallel learning algorithms for neural networks. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems. Proceedings of the 1988 Conference* (pp. 40–48). San Mateo, CA: Morgan Kaufmann.
- [Kreyszig, 1993] Kreyszig, E. (1993). *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., 7 edition.
- [Lipo, 1977] Lipo, T. A. (1977). Flux sensing and control of static ac drives by the use of flux coils. *IEEE Trans. on Magnetics*, 13(5), 1403–1408.
- [Lipo & Chang, 1986] Lipo, T. A. & Chang, K. C. A. (1986). New approach to flux and torque-sensing in induction machines. *IEEE Trans. on Industry Applications*, 22, 731–737.
- [Luenberger, 1969] Luenberger, D. G. (1969). *Optimization By Vector Space Methods*. New York: Wiley.
- [Marquardt, 1963] Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431–441.
- [McCulloch & Pitts, 1943] McCulloch, W. & Pitts, W. (1943). A logical calculus of ideas
-

- immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133.
- [Minsky & Papert, 1969] Minsky, M. L. & Papert, S. A. (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.
- [Moreira & Fiesler, 1995] Moreira, M. & Fiesler, E. (1995). *Neural Networks with Adaptive Learning Rate and Momentum Terms*. Technical Report 95-04, IDIAP, Martigny, Switzerland.
- [Neves, 1999] Neves, F. A. S. (1999). *Acionamento de alto desempenho com máquinas de indução alimentadas em tensão: desenvolvimento de novas estratégias de controle*. PhD thesis, Escola de Engenharia da UFMG.
- [Neves et al., 1999] Neves, F. A. S., Habetler, T. G., Parma, G. G., Menezes, B. R., & Silva, S. R. (1999). An evaluation of sensorless induction motor drives for low speed operation. In *Proceedings do 5o. Congresso Brasileiro de Eletrônica de Potência*, volume 2 (pp. 659–665). Foz do Iguaçu, Brazil.
- [Novotny & Lipo, 1996] Novotny, D. W. & Lipo, T. A. (1996). *Vector Control and Dynamics of AC Drives*. New York: Oxford University Press Inc.
- [Parma et al., 1997a] Parma, G. G., Menezes, B. R., & Braga, A. P. (1997a). Aprendizado on-line de rnas multi-camadas baseado na teoria de sistemas de estrutura variável. In *IV Simposio Brasileiro de Redes Neurais* (pp. 80–82). Goiania, Brazil: Editora da UFG.
- [Parma et al., 1998a] Parma, G. G., Menezes, B. R., & Braga, A. P. (1998a). Improving backpropagation with sliding mode control. In *Proceedings of the Vth Brazilian Symposium on Neural Networks* (pp. 8–13). Belo Horizonte, Brazil: IEEE Computer Society Press.
- [Parma et al., 1998b] Parma, G. G., Menezes, B. R., & Braga, A. P. (1998b). Sliding mode algorithm for training multi-layer neural networks. *IEE Electronics Letters*, 38(1), 97–98.
- [Parma et al., 1999a] Parma, G. G., Menezes, B. R., & Braga, A. P. (1999a). Aceleração da convergência do algoritmo de treinamento smc-bp através do ajuste da superfície de deslizamento. In *Proceedings do IV Congresso Brasileiro de Redes Neurais*, volume 1 (pp. 188–191).
- [Parma et al., 1999b] Parma, G. G., Menezes, B. R., & Braga, A. P. (1999b). Neural networks learning with sliding mode control: The sliding mode backpropagation algorithm. *International Journal of neural Systems*, 9(3), 187–193.
- [Parma et al., 1999c] Parma, G. G., Menezes, B. R., & Braga, A. P. (1999c). Sliding mode backpropagation: Control theory applied to neural network learning. In *Proceedings of the International Joint Conference on Neural Networks Washington-DC*: IEEE Computer Society Press.

-
- [Parma et al., 1998c] Parma, G. G., Menezes, B. R., Braga, A. P., Aguirre, L. A., & Oliveira, J. C. R. (1998c). Observador neural de fluxo estatístico com treinamento on-line. In *XII Congresso Brasileiro de Automática* (pp. 1301–1306). Uberlândia, Brazil.
- [Parma et al., 1997b] Parma, G. G., Rocha, A. V., & Landim, R. P. (1997b). *Ensaio Paramétricos - Motor de Indução e Máquina de Corrente Contínua*. Technical report, UFMG, Belo Horizonte, Brasil.
- [Riedmiller & Braun, 1993] Riedmiller, M. & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. of the IEEE Intl. Conf. on Neural Networks* (pp. 586–591). San Francisco, CA.
- [Rocha et al., 1999] Rocha, A. V., de Oliveira, J. C. R., & Silva, S. R. (1999). Digital acquisition of voltages in electrical drives for application in observers and vector control. In *COBEP99 - 5o. Congresso Brasileiro de Eletrônica de Potência*, volume 2 (pp. 651–655). Foz do Iguaçu - Paraná, Brasil.
- [Rodic et al., 1996] Rodic, M., Jezernik, K., Safaric, R., & Sabanovic, A. (1996). Sliding mode based neural network learning procedure. In *Robotics in Alpe-Adria Region* (pp. 547–552). Budapest, Hungary.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage in the brain. *Psychological review*, 65, 386–408.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representation by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1 (pp. 318–362). Cambridge, MA: MIT Press.
- [Sabanovic et al., 1996] Sabanovic, A., Jezernik, K., & Rodic, M. (1996). Neural network application in sliding mode control systems. In *Proc. IEEE Workshop on Variable Structure Systems USA*.
- [Sarpturk et al., 1987] Sarpturk, S. Z., Istefanopulos, Y., & Kaynak, O. (1987). On the stability of discrete time sliding mode control systems. *IEEE Transactions on Automatic Control*, 39(10), 930–932.
- [Schiffmann et al., 1992] Schiffmann, W., Joost, M., & Werner, R. (1992). *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. Technical report, Institute of Physics, University of Koblenz, Koblenz, Germany.
- [Schmidhuber, 1989] Schmidhuber, J. (1989). Accelerated learning in back-propagation nets. In *Connectionism in Perspective* (pp. 439–445). Amsterdam: Elsevier.
- [Silva & Almeida, 1990] Silva, F. M. & Almeida, L. B. (1990). Speeding up backpropagation. In R. Eckmiller (Ed.), *Advanced Neural Computers* (pp. 151–158). Amsterdam: Elsevier North Holland.
-

-
- [Stopa, 1997] Stopa, M. M. (1997). Controle do motor de indução por orientação de campo: Uma implementação utilizando um conversor reversível. Master's thesis, Escola de Engenharia da UFMG.
- [Tollenaere, 1990] Tollenaere, T. (1990). SuperSAB: Fast adaptive back propagation with good scaling properties. *Neural Networks*, 3(5), 561–573.
- [Utkin, 1978] Utkin, V. (1978). *Sliding modes and their application in Variable Structure Systems*. Moscow: MIR.
- [Verghese & Sanders, 1988] Verghese, G. C. & Sanders, S. R. (1988). Observers for flux estimation in induction machines. *IEEE Transactions on Industrial Electronics*, 35(1).
- [Werbos, 1974] Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Boston, MA: Doctoral Dissertation, Applied Mathematics, Harvard University.
- [Widrow, 1962] Widrow, B. (1962). Generalization and information storage in networks of adaline “neurons”. In M. C. Yovits, G. T. Jacobi, & G. D. Goldstein (Eds.), *Self-Organizing Systems 1962* (pp. 435–461). Washington: (Chicago 1962) Spartan.
- [Widrow & Hoff, 1960] Widrow, B. & Hoff, M. E. (1960). Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, volume 4 (pp. 96–104). New York: IRE.
- [Zurada, 1992] Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*. St. Paul: West Publishing Company.
-

Apêndice A

Equacionamento Matemático

Neste apêndice, as principais equações da topologia MLP serão mostradas. Estas equações serão, então, utilizadas ao longo de todo o texto.

A.1 Equações básicas da RNA MLP

Utilizando-se a nomenclatura adotada, tem-se que, para uma RNA MLP:

$$R_{i(k)} = \sum_{l=1}^{n+1} (Z_{il(k)} \cdot T_{l(k)}) \quad (\text{A.1})$$

$$V_{j(k)} = \sum_{l=1}^{m+1} (W_{jl(k)} \cdot Y_{Hl(k)}) \quad (\text{A.2})$$

$$Y_{Hi(k)} = f_H(R_{i(k)}) \quad (\text{A.3})$$

$$Y_{j(k)} = f(V_{j(k)}) \quad (\text{A.4})$$

$$e_{j(k)} = Y_{dj(k)} - Y_{j(k)} \quad (\text{A.5})$$

$$\Delta X_{(k)} = X_{(k)} - X_{(k-1)} \quad (\text{A.6})$$

$$E_{(k)} = \frac{1}{2} \sum_{j=1}^p e_{j(k)}^2 \quad (\text{A.7})$$

A Equação A.7 (somatório dos erros médios quadráticos da camada de saída no instante k) será utilizada como função de custo a ser minimizada pela maioria dos algoritmos de

treinamento.

A.2 Gradiente do erro em função dos pesos

Para uma rede MLP de duas camadas, deseja-se determinar o valor do gradiente do erro em função dos pesos, tanto da camada de saída quanto da camada escondida.

- Camada de Saída

Seja o critério de erro definido pela Equação A.7. Deseja-se, portanto, determinar o valor de $\nabla E_{(k)} = \frac{\partial E_{(k)}}{\partial W_{ji}}$.

Portanto, pode-se determinar o gradiente do erro, para o peso W_{ji} da camada de saída, como se segue:

$\nabla E_{(k)} = \frac{\partial E_{(k)}}{\partial W_{ji}}$, utilizando-se a Equação A.7 e A.5:

$$\nabla E_{(k)} = \frac{\partial \left(\frac{1}{2} \sum_{j=1}^p (Y_{dj(k)} - Y_{j(k)})^2 \right)}{\partial W_{ji}}$$

A partir da regra da cadeia para a derivação:

$$\nabla E_{(k)} = -1 \cdot (Y_{dj(k)} - Y_{j(k)}) \cdot \frac{\partial Y_{j(k)}}{\partial W_{ji}}$$

Utilizando a Equação A.4, A.2,

$$\nabla E_{(k)} = -1 \cdot (Y_{dj(k)} - Y_{j(k)}) \cdot \frac{\partial f(V_{j(k)})}{\partial V_{j(k)}} \frac{\partial V_{j(k)}}{\partial W_{ji}}$$

Conseqüentemente, utilizando-se a Equação A.5:

$$\nabla E_{(k)} = -e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} \quad (\text{A.8})$$

O termo $e_{j(k)} \cdot f'(V_{j(k)})$ da equação acima é, muitas vezes, referido como "gradiente local

” e definido por $\delta_{j(k)}$. Ou seja,

$$\delta_{j(k)} = e_{j(k)} \cdot f'(V_{j(k)}) \quad (\text{A.9})$$

e, conseqüentemente, a Equação A.8 pode ser reescrita como:

$$\nabla E_{(k)} = -\delta_{j(k)} \cdot Y_{Hi(k)} \quad (\text{A.10})$$

- Camada Escondida

O gradiente do erro com relação ao peso $Z_{ih(k)}$ da camada escondida pode ser calculado como é mostrado a seguir:

$$\nabla E_{(k)} = \frac{\partial E_{(k)}}{\partial Z_{ih(k)}}$$

$$\frac{\partial E_{(k)}}{\partial Z_{ih(k)}} = \frac{\partial E_{(k)}}{\partial Y_{Hi(k)}} \frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}} \frac{\partial R_{i(k)}}{\partial Z_{ih(k)}}$$

Para facilitar a compreensão da dedução matemática, os termos serão estudados separadamente.

Termo 1: $\frac{\partial E_{(k)}}{\partial Y_{Hi(k)}}$, considerando o erro definido pela Equação A.7 e A.5, tem-se que:

$$\frac{\partial E_{(k)}}{\partial Y_{Hi(k)}} = \frac{1}{2} \sum_{j=1}^p \frac{\partial (Y_{dj(k)} - Y_{j(k)})^2}{\partial Y_{Hi(k)}}$$

$$\frac{\partial E_{(k)}}{\partial Y_{Hi(k)}} = \frac{1}{2} \sum_{j=1}^p \frac{\partial (Y_{dj(k)} - Y_{j(k)})^2}{\partial Y_{j(k)}} \frac{\partial Y_{j(k)}}{\partial V_{j(k)}} \frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}}$$

$$\text{Termo 1.1: } \frac{\partial (Y_{dj(k)} - Y_{j(k)})^2}{\partial Y_{j(k)}} = -2 \cdot (Y_{dj(k)} - Y_{j(k)});$$

$$\text{Termo 1.2: } \frac{\partial Y_{j(k)}}{\partial V_{j(k)}} = f'(V_{j(k)}), \text{ por definição;}$$

Termo 1.3: $\frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}}$, a partir da Equação A.2, tem-se que: $\frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}} = W_{ji(k)}$,

logo,

$$\frac{\partial E_{(k)}}{\partial Y_{Hi(k)}} = - \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}];$$

ou, considerando a Equação A.9,

$$\frac{\partial E_{(k)}}{\partial Y_{Hi(k)}} = - \sum_{j=1}^p [\delta_{j(k)} \cdot W_{ji(k)}]$$

Termo 2: $\frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}}$ onde, utilizando-se a Equação A.3, $Y_{Hi(k)} = f_H(R_{i(k)})$ e, portanto,

$$\frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}} = \frac{\partial f_H(R_{i(k)})}{\partial R_{i(k)}} \text{ que, por definição, é igual à } f'_H(R_{i(k)});$$

Termo 3: $\frac{\partial R_{i(k)}}{\partial Z_{ih(k)}}$, considerando a Equação A.1, tem-se que: $\frac{\partial R_{i(k)}}{\partial Z_{ih(k)}} = T_{h(k)}$,

logo,

$$\frac{\partial E_{(k)}}{\partial Z_{ih(k)}} = -T_{h(k)} \cdot f'_H(R_{i(k)}) \cdot \sum_{j=1}^p [\delta_{j(k)} \cdot W_{ji(k)}] \quad (\text{A.11})$$

De forma semelhante à camada de saída, define-se o termo $f'_H(R_{i(k)}) \cdot \sum_{j=1}^p [\delta_{j(k)} \cdot W_{ji(k)}]$ como sendo o gradiente local da camada escondida, ou seja,

$$\delta_{i(k)} = f'_H(R_{i(k)}) \cdot \sum_{j=1}^p [\delta_{j(k)} \cdot W_{ji(k)}] \quad (\text{A.12})$$

Portanto, pode-se reescrever a Equação A.11 como:

$$\frac{\partial E_{(k)}}{\partial Z_{ih(k)}} = -\delta_{i(k)} \cdot T_{h(k)} \quad (\text{A.13})$$

A.3 Outras Relações de interesse

Serão mostradas, agora, a determinação de algumas variáveis de interesse que são utilizadas nas análises matemáticas dos algoritmos propostos neste trabalho.

Inicialmente, define-se que:

$$\Delta X_{(k)} = X_{(k)} - X_{(k-1)} \quad (\text{A.14})$$

onde:

$X_{(k)}$ é uma variável qualquer de interesse no instante k ;

Deseja-se, agora, determinar algumas variáveis de interesse que serão utilizadas nas deduções posteriores.

Várias deduções estão baseadas no primeiro termo da série de Taylor ($\Delta x = \frac{\partial x}{\partial y} \Delta y$) e na regra da cadeia para a derivação, como poderá ser observado nas equações subseqüentes.

A.3.0.1 Variação da saída da RNA em função da variação do peso da camada de saída

Deseja-se determinar qual o efeito da variação de um peso na camada de saída no valor da saída final da RNA. Logo, deseja-se determinar a função $f(x)$ tal que : $\Delta Y_{j(k)} = f(\Delta W_{ji(k)})$

Utilizando-se a Equação A.4, tem-se:

$$\Delta Y_{j(k)} = \Delta f(V_{j(k)})$$

A partir o primeiro termo da série de Taylor e da regra da cadeia:

$$\Delta Y_{j(k)} = \frac{\partial f(V_{j(k)})}{\partial V_{j(k)}} \frac{\partial V_{j(k)}}{\partial W_{ji(k)}} \Delta W_{ji(k)}$$

sendo,

Termo 1: $\frac{\partial f(V_{j(k)})}{\partial V_{j(k)}} = f'(V_{j(k)})$

Termo 2: $\frac{\partial V_{j(k)}}{\partial W_{ji(k)}}$, utilizando-se a Equação A.2, tem-se que: $\frac{\partial V_{j(k)}}{\partial W_{ji(k)}} = Y_{Hi(k)}$;

logo,

$$\Delta Y_{j(k)} = f'(V_{j(k)}) \cdot Y_{Hi(k)} \cdot \Delta W_{ji(k)} \quad (\text{A.15})$$

A.3.0.2 Variação da saída da RNA em função da variação do peso da camada escondida

Deseja-se determinar qual o efeito da variação de um peso na camada de saída no valor da saída final da RNA. Logo, deseja-se determinar a função $f(x)$ tal que: $\Delta Y_{j(k)} = f(\Delta Z_{ih(k)})$

Utilizando-se a Equação A.4, o primeiro termo da série de Taylor e a regra da cadeia para a derivação:

$$\Delta Y_{j(k)} = \Delta f(V_{j(k)})$$

$$\Delta Y_{j(k)} = \frac{\partial f(V_{j(k)})}{\partial V_{j(k)}} \frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}} \frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}} \frac{\partial R_{i(k)}}{\partial Z_{ih(k)}} \Delta Z_{ih(k)}$$

sendo,

Termo 1: $\frac{\partial f(V_{j(k)})}{\partial V_{j(k)}} = f'(V_{j(k)})$;

Termo 2: $\frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}}$, utilizando-se a Equação A.2, tem-se que: $\frac{\partial V_{j(k)}}{\partial Y_{Hi(k)}} = W_{ji(k)}$;

Termo 3: $\frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}}$, a partir da Equação A.3, $\frac{\partial Y_{Hi(k)}}{\partial R_{i(k)}} = \frac{\partial f_H(R_{i(k)})}{\partial R_{i(k)}} = f'_H(R_{i(k)})$;

Termo 4: $\frac{\partial R_{i(k)}}{\partial Z_{ih(k)}}$, calculando a partir da Equação A.1, tem-se que: $\frac{\partial R_{i(k)}}{\partial Z_{ih(k)}} = T_{h(k)}$,

logo,

$$\Delta Y_{j(k)} = f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \Delta Z_{ih(k)} \quad (\text{A.16})$$

A.4 Detalhamento matemático para o Treinamento *Off-line* - 1a. Proposta

A.4.1 Camada de saída

Substituindo a Equação 4.9 e A.8 em 4.10:

$$X_{2ji(k)} = X_{1ji(k)} - X_{1ji(k-1)}$$

$$X_{2ji(k)} = -e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot Y_{Hi(k-1)}$$

Considerando-se a definição A.6 para as variáveis $f'(V_{j(k-1)})$ e $Y_{Hi(k-1)}$,

$$X_{2ji(k)} = -e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [f'(V_{j(k)}) - \Delta f'(V_{j(k)})] \cdot (Y_{Hi(k)} - \Delta Y_{Hi(k)})$$

$$X_{2ji(k)} = -e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

$$X_{2ji(k)} = [-Y_{dj(k)} + Y_{dj(k-1)} + Y_{j(k)} - Y_{j(k-1)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

novamente, utilizando a definição A.14 para a variável $Y_{j(k)}$

$$X_{2ji(k)} = [-\Delta Y_{dj(k)} + \Delta Y_{j(k)}] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Utilizando-se a Equação A.15 e 4.14 na expressão acima, obtém-se:

$$X_{2ji(k)} = [-\Delta Y_{dj(k)} + f'(V_j(k)) \cdot Y_{Hi(k)} \cdot (-\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}))] \cdot f'(V_j(k)) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_j(k)) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_j(k)) \cdot Y_{Hi(k)} - f'(V_j(k)) \cdot \Delta Y_{Hi(k)}]$$

$$X_{2ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \cdot [f'(V_j(k)) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f'(V_j(k)) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_j(k)) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_j(k)) \cdot Y_{Hi(k)} - f'(V_j(k)) \cdot \Delta Y_{Hi(k)}]$$

Substituindo a Equação 4.9 e a expressão acima determinada em 4.8:

$$S_{ji(k)} = -C \cdot e_{j(k)} \cdot f'(V_j(k)) \cdot Y_{Hi(k)} - \alpha \cdot |X_{1ji(k)}| \cdot \text{sgn}(S_{ji(k)}) \cdot [f'(V_j(k)) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f'(V_j(k)) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_j(k)) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_j(k)) \cdot Y_{Hi(k)} - f'(V_j(k)) \cdot \Delta Y_{Hi(k)}]$$

$$S_{ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot [f'(V_j(k)) \cdot Y_{Hi(k)}]^2 \cdot \text{sgn}(S_{ji(k)}) - [C \cdot e_{j(k)} + \Delta Y_{dj(k)}] \cdot f'(V_j(k)) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_j(k)) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_j(k)) \cdot Y_{Hi(k)} - f'(V_j(k)) \cdot \Delta Y_{Hi(k)}]$$

Definindo:

$$X_{(k)} = |X_{1ji(k)}| \cdot [f'(V_j(k)) \cdot Y_{Hi(k)}]^2$$

$$Y_{(k)} = -[C \cdot e_{j(k)} + \Delta Y_{dj(k)}] \cdot f'(V_j(k)) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f'(V_j(k)) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_j(k)) \cdot Y_{Hi(k)} - f'(V_j(k)) \cdot \Delta Y_{Hi(k)}]$$

A.4.2 Camada escondida

Desta forma, substituindo-se as Equações 4.12 e A.11 em 4.13 tem-se:

$$X_{2ih(k)} = -f'_H(R_i(k)) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_j(k)) \cdot W_{ji(k)}] + f'_H(R_i(k-1)) \cdot T_{h(k-1)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Utilizando-se a definição A.6 para as variáveis $f'_H(R_i(k-1))$ e $T_{h(k-1)}$:

$$X_{2ih(k)} = -f'_H(R_i(k)) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_j(k)) \cdot W_{ji(k)}] + [f'_H(R_i(k)) - \Delta f'_H(R_i(k))] \cdot$$

$$[T_{h(k)} - \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$\begin{aligned} X_{2ih(k)} &= -f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \\ &\sum_{j=1}^p [(Y_{dj(k-1)} - Y_{j(k-1)}) \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot \\ &T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

$$\begin{aligned} X_{2ih(k)} &= f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{- [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot \\ &f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \\ &\sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

novamente, considerando a definição A.6 para as variáveis $f'(V_{j(k-1)})$ e $W_{ji(k-1)}$:

$$\begin{aligned} X_{2ih(k)} &= f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{- [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot \\ &(f'(V_{j(k)}) - \Delta f'(V_{j(k)})) \cdot (W_{ji(k)} - \Delta W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot \\ &T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

$$\begin{aligned} X_{2ih(k)} &= \\ &f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-Y_{dj(k)} + Y_{j(k)} + Y_{dj(k-1)} - Y_{j(k-1)})] + [e_{j(k-1)} \cdot \\ &(\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \\ &\Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

$$\begin{aligned} X_{2ih(k)} &= \\ &f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-\Delta Y_{dj(k)} + \Delta Y_{j(k)})] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot \\ &(\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \\ &\Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \end{aligned}$$

Substituindo a Equação A.16 e 4.15 na equação acima,

$$X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-\Delta Y_{dj(k)} + f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot (-\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}))] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{-\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot T_{h(k)}\} + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$X_{2ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Substituindo a expressão acima determinada e a Equação 4.12 na Equação 4.11, obtém-se:

$$S_{ih(k)} = -C_H \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - \beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$\begin{aligned}
 S_{ih(k)} = & -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + \\
 & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - [C_H \cdot e_{(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot \\
 & (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \\
 & \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]
 \end{aligned}$$

utilizando-se a Equação A.9, define-se:

$$\begin{aligned}
 X_{(k)} = & |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 \\
 Y_{(k)} = & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] \\
 & - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot \\
 & (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} \\
 & + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \\
 & \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}]
 \end{aligned}$$

A.5 Detalhamento matemático para o Treinamento *Off-line* - 2a. Proposta

A.5.1 Camada de saída

Substituindo-se as Equações 4.23 e 4.24 em 4.22:

$$S_{ji(k)} = C \cdot \sum_{l=1}^k \frac{\partial E_{(l)}}{\partial W_{ji(l)}} + \frac{\partial E_{(k)}}{\partial W_{ji(k)}}$$

$$S_{ji(k)} = C \cdot \frac{\partial E_{(k)}}{\partial W_{ji(k)}} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E_{(l)}}{\partial W_{ji(l)}} + \frac{\partial E_{(k)}}{\partial W_{ji(k)}}$$

$$S_{ji(k)} = (C + 1) \cdot \frac{\partial E(k)}{\partial W_{ji(k)}} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

utilizando-se a Equação A.8, tem-se:

$$S_{ji(k)} = (C + 1) \cdot [-e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

considerando as Equações A.5 e A.6 para a variável $Y_{j(k)}$,

$$S_{ji(k)} = (C + 1) \cdot \left\{ -[Y_{dj(k)} - (\Delta Y_{j(k)} + Y_{j(k-1)})] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} \right\} + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

$$S_{ji(k)} = (C + 1) \cdot \Delta Y_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} - (C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

usando a Equação A.15 e 4.28:

$$S_{ji(k)} = -\alpha \cdot \operatorname{sgn}(S_{ji(k)}) \cdot (C + 1) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \left| \sum_{l=1}^k \frac{\partial E(l)}{\partial W_{ji(l)}} \right| - (C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

define-se, portanto:

$$X_{(k)} = (C + 1) \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \left| \sum_{l=1}^k \frac{\partial E(l)}{\partial W_{ji(l)}} \right|$$

$$Y_{(k)} = -(C + 1) \cdot [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)}] + C \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial W_{ji(l)}}$$

A.5.2 Camada escondida

O estudo da convergência prossegue de maneira análoga à utilizada para a camada de

saída, assim sendo, substituindo a Equação 4.26 e 4.27 em 4.25:

$$S_{ih(k)} = C_H \cdot \sum_{l=1}^k \frac{\partial E(l)}{\partial Z_{ih(l)}} + \frac{\partial E(k)}{\partial Z_{ih(k)}}$$

$$S_{ih(k)} = C_H \cdot \frac{\partial E(k)}{\partial Z_{ih(k)}} + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}} + \frac{\partial E(k)}{\partial Z_{ih(k)}}$$

$$S_{ih(k)} = (C_H + 1) \cdot \frac{\partial E(k)}{\partial Z_{ih(k)}} + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

utilizando a Equação A.11,

$$S_{ih(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

considerando, agora, as Equações A.5 e A.6 para a variável $Y_{j(k)}$:

$$S_{ih(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - (\Delta Y_{j(k)} + Y_{j(k-1)})) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

$$S_{ih(k)} = (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [\Delta Y_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

Substituindo-se a Equação A.16, e posteriormente a Equação 4.29, na expressão acima, tem-se:

$$S_{ih(k)} = (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot [-\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot f'_H(R_{i(k)})] \cdot f'(V_{j(k)}) \cdot W_{ji(k)} \} - (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

$$S_{ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot (C_H + 1) \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 - (C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

Define-se, portanto:

$$X_{(k)} = |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot (C_H + 1) \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2$$

$$Y_{(k)} = -(C_H + 1) \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [(Y_{dj(k)} - Y_{j(k-1)}) \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + C_H \cdot \sum_{l=1}^{k-1} \frac{\partial E(l)}{\partial Z_{ih(l)}}$$

A.6 Detalhamento matemático para o Treinamento *On-line* - 1a. proposta

A.6.1 Camada de saída

Substituindo-se a Equação 4.37 em 4.38, considerando-se a Equação A.6, e efetuando a derivada, tem-se

$$X_{2ji(t)} = \frac{\partial (e_j(t))}{\partial t}$$

$$X_{2ji(t)} = \frac{\partial (Y_{dj(t)} - Y_{j(t)})}{\partial t}$$

$$X_{2ji(t)} = \frac{\partial Y_{dj(t)}}{\partial t} - \frac{\partial Y_{j(t)}}{\partial t}$$

utilizando uma expressão análoga à Equação A.15, para o tempo contínuo, $\frac{\partial Y_{j(t)}}{\partial t} = f'(V_{j(k)}) \cdot Y_{Hi(k)} \cdot \dot{W}_{ji(k)}$, em conjunto com a Equação 4.42, para a determinação de $X_{2ji(t)}$

obtém-se:

$$X_{2ji(t)} = \frac{\partial Y_{dj(t)}}{\partial t} - f'(V_{j(k)}) \cdot Y_{Hi(k)} \cdot \frac{\alpha \cdot |X_{1ji(t)}| \cdot \operatorname{sgn}(S_{ji(t)}) \cdot Y_{Hi(t)}}{f'(V_{j(k)})}$$

$$X_{2ji(t)} = \frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \quad (\text{A.17})$$

Substituindo a Equação A.17 em 4.36,

$$S_{ji(t)} = C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2$$

$$|S_{ji(t)}| \cdot \operatorname{sgn}(S_{ji(t)}) = C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2$$

$$\left[|S_{ji(t)}| + \alpha \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right] \cdot \operatorname{sgn}(S_{ji(t)}) = \left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right| \cdot \operatorname{sgn}\left(C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right)$$

$$\therefore \operatorname{sgn}\left(C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right) = \operatorname{sgn}(S_{ji(t)})$$

$$S_{ji(t)} = \left[\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right| - \alpha \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right] \cdot \operatorname{sgn}(S_{ji(t)}) \quad (\text{A.18})$$

Define-se, portanto, um primeiro limite para a variável α para que a expressão acima seja válida:

$$\alpha < \frac{\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right|}{|X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2} \quad (\text{A.19})$$

Derivando a Equação A.17,

$$\frac{\partial X_{2ji(t)}}{\partial t} = \frac{\partial}{\partial t} \left[\frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right]$$

$$\begin{aligned} \frac{\partial X_{2ji(t)}}{\partial t} &= \frac{\partial^2 Y_{dj(t)}}{\partial t^2} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot \frac{\partial}{\partial t} \left[|X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right] \\ \frac{\partial X_{2ji(t)}}{\partial t} &= \frac{\partial^2 Y_{dj(t)}}{\partial t^2} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot \left[[Y_{Hi(t)}]^2 \cdot \frac{\partial |X_{1ji(t)}|}{\partial t} + 2 \cdot |X_{1ji(t)}| \cdot Y_{Hi(t)} \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \right] \\ \frac{\partial X_{2ji(t)}}{\partial t} &= \frac{\partial^2 Y_{dj(t)}}{\partial t^2} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot [Y_{Hi(t)}]^2 \cdot \left[\frac{\partial |X_{1ji(t)}|}{\partial t} + 2 \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \right] \quad (\text{A.20}) \end{aligned}$$

Derivando a Equação 4.36 para que se possa prosseguir na análise da condição 4.44,

$$\frac{\partial S_{ji(t)}}{\partial t} = C \cdot \frac{\partial X_{1ji(t)}}{\partial t} + \frac{\partial X_{2ji(t)}}{\partial t}$$

utilizando a Equação 4.38,

$$\frac{\partial S_{ji(t)}}{\partial t} = C \cdot X_{2ji(t)} + \frac{\partial X_{2ji(t)}}{\partial t}$$

Substituindo as Equações A.17 e A.20 na expressão acima,

$$\begin{aligned} \frac{\partial S_{ji(t)}}{\partial t} &= C \cdot \left[\frac{\partial Y_{dj(t)}}{\partial t} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot |X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2 \right] + \frac{\partial^2 Y_{dj(t)}}{\partial t^2} - \alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot \\ & [Y_{Hi(t)}]^2 \cdot \left[\frac{\partial |X_{1ji(t)}|}{\partial t} + 2 \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \right] \end{aligned}$$

$$\begin{aligned} \dot{S}_{ji(t)} &= -\alpha \cdot \operatorname{sgn}(S_{ji(t)}) \cdot [Y_{Hi(t)}]^2 \cdot \left\{ 2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right\} + \\ & \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \end{aligned}$$

Considerando a Equação 4.44, na convergência do sistema, $\operatorname{sgn}(S_{ji(t)}) = -\operatorname{sgn}(\dot{S}_{ji(t)})$, desta forma pode-se escrever:

$$\begin{aligned} \dot{S}_{ji(t)} &= +\alpha \cdot \operatorname{sgn}(\dot{S}_{ji(t)}) \cdot [Y_{Hi(t)}]^2 \cdot \left\{ 2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right\} + \\ & \left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right| \cdot \operatorname{sgn} \left(\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right) \end{aligned}$$

$$\left[\left| \dot{S}_{ji(t)} \right| - \alpha \cdot [Y_{Hi(t)}]^2 \cdot \left\{ 2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right\} \right] \cdot \operatorname{sgn} \left(\dot{S}_{ji(t)} \right) = \left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right| \cdot \operatorname{sgn} \left(\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right)$$

$$\operatorname{sgn} \left(\dot{S}_{ji(t)} \right) = \operatorname{sgn} \left(\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right)$$

$$\therefore \dot{S}_{ji(t)} = \left\{ \alpha \cdot [Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right] + \left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right| \right\} \cdot \operatorname{sgn} \left(\dot{S}_{ji(t)} \right)$$

Analisando a expressão acima, tem-se que, necessariamente,

$$\alpha \cdot [Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right] + \left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right| > 0$$

$$\alpha > - \frac{\left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right|}{[Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right]}$$

Portanto, a expressão acima, juntamente com a Equação A.19, define os limites para o ganho α .

$$\frac{\left| \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right|}{[Y_{Hi(t)}]^2 \cdot \left[2 \cdot \frac{\partial Y_{Hi(t)}}{\partial t} \cdot \frac{|X_{1ji(t)}|}{Y_{Hi(t)}} + C \cdot |X_{1ji(t)}| + \frac{\partial |X_{1ji(t)}|}{\partial t} \right]} < \alpha$$

$$< \frac{\left| C \cdot X_{1ji(t)} + \frac{\partial Y_{dj(t)}}{\partial t} \right|}{|X_{1ji(t)}| \cdot [Y_{Hi(t)}]^2}$$

A.6.2 Camada escondida

Substituindo a Equação 4.40 em 4.41, considerando-se a Equação A.6, e derivando,

$$X_{2ih(t)} = \frac{\partial}{\partial t} \left[\frac{1}{2} \cdot \sum_{j=1}^p [Y_{dj(t)} - Y_j(t)]^2 \right]$$

$$X_{2ih(t)} = \frac{1}{2} \cdot \sum_{j=1}^p \frac{\partial}{\partial t} [Y_{dj(t)} - Y_j(t)]^2$$

$$X_{2ih(t)} = \sum_{j=1}^p e_j(t) \cdot \left[\frac{\partial Y_{dj(t)}}{\partial t} - \frac{\partial Y_j(t)}{\partial t} \right]$$

Considerando a expressão análoga à Equação A.16 para tempo contínuo,

$$\frac{\partial Y_{j(k)}}{\partial t} = f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t}$$

$$X_{2ih(t)} = \sum_{j=1}^p e_j(t) \cdot \left[\frac{\partial Y_{dj(t)}}{\partial t} - f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t} \right]$$

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ e_j(t) \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \sum_{j=1}^p \left\{ e_j(t) \cdot f'(V_{j(k)}) \cdot f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t} \right\}$$

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ e_j(t) \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \frac{\partial Z_{ih(k)}}{\partial t} \cdot \sum_{j=1}^p \left\{ e_j(t) \cdot f'(V_{j(k)}) \cdot W_{ji(k)} \right\}$$

Utilizando as Equações 4.43 e 4.37

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ e_j(t) \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \frac{\beta \cdot |X_{1ih(t)}| \cdot \text{sgn}(S_{ih(t)}) \cdot T_{h(k)}}{f'_H(R_{i(k)}) \cdot \sum_{j=1}^p [e_j(t) \cdot f'(V_{j(t)}) \cdot W_{ji(t)}}$$

$$\sum_{j=1}^p \left\{ e_j(t) \cdot f'(V_{j(k)}) \cdot W_{ji(k)} \right\}$$

$$X_{2ih(t)} = \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \text{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \quad (\text{A.21})$$

Utilizando a expressão acima e a Equação 4.40 na expressão 4.39,

$$S_{ih(t)} = C_H \cdot \frac{1}{2} \cdot \sum_{j=1}^p [e_{j(t)}]^2 + \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

Substituindo a Equação 4.37,

$$S_{ih(t)} = C_H \cdot \frac{1}{2} \cdot \sum_{j=1}^p [X_{1ji(t)}]^2 + \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

$$S_{ih(t)} = \frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

Prosseguindo de maneira análoga à utilizada durante a análise da convergência para a camada de saída tem-se:

$$|S_{ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) = \frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

$$\left[|S_{ih(t)}| + \beta \cdot |X_{1ih(t)}| \cdot [T_{h(k)}]^2 \right] \cdot \operatorname{sgn}(S_{ih(t)}) = \left| \frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} \right| \cdot$$

$$\operatorname{sgn} \left(\frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} \right)$$

$$\operatorname{sgn}(S_{ih(t)}) = \operatorname{sgn} \left(\frac{1}{2} \cdot \sum_{j=1}^p \left\{ C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} \right)$$

$$S_{ih(t)} = \left\{ \left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right| - \beta \cdot |X_{1ih(t)}| \cdot [T_{h(k)}]^2 \right\} \cdot \operatorname{sgn}(S_{ih(t)})$$

Conseqüentemente,

$$\left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right| - \beta \cdot |X_{1ih(t)}| \cdot [T_{h(k)}]^2 > 0$$

$$\beta < \frac{\left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right|}{|X_{1ih(t)}| \cdot [T_{h(k)}]^2} \quad (\text{A.22})$$

Derivando a expressão 4.39,

$$\frac{\partial}{\partial t} S_{ih(t)} = C_H \cdot \frac{\partial}{\partial t} X_{1ih(t)} + \frac{\partial}{\partial t} X_{2ih(t)}$$

$$\dot{S}_{ih(t)} = C_H \cdot X_{2ih(t)} + \frac{\partial}{\partial t} X_{2ih(t)}$$

Para a determinação da parcela $\frac{\partial}{\partial t} X_{2ih(t)}$, a expressão A.21 é derivada, obtendo:

$$\frac{\partial X_{2ih(t)}}{\partial t} = \frac{\partial}{\partial t} \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - \frac{\partial}{\partial t} \beta \cdot |X_{1ih(t)}| \cdot \text{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2$$

$$\begin{aligned} \frac{\partial X_{2ih(t)}}{\partial t} &= \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial}{\partial t} \frac{\partial Y_{dj(t)}}{\partial t} + \frac{\partial Y_{dj(t)}}{\partial t} \cdot \frac{\partial}{\partial t} X_{1ji(t)} \right\} - \beta \cdot \text{sgn}(S_{ih(t)}) \cdot \\ &\left[|X_{1ih(t)}| \cdot \frac{\partial}{\partial t} [T_{h(k)}]^2 + [T_{h(k)}]^2 \cdot \frac{\partial}{\partial t} |X_{1ih(t)}| \right] \end{aligned}$$

$$\begin{aligned} \frac{\partial X_{2ih(t)}}{\partial t} &= \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + \frac{\partial Y_{dj(t)}}{\partial t} \cdot \frac{\partial}{\partial t} X_{1ji(t)} \right\} - \beta \cdot \text{sgn}(S_{ih(t)}) \cdot \\ &\left[2 \cdot |X_{1ih(t)}| \cdot \frac{\partial T_{h(k)}}{\partial t} + [T_{h(k)}]^2 \cdot \frac{\partial |X_{1ih(t)}|}{\partial t} \right] \end{aligned}$$

utilizando 4.38,

$$\begin{aligned} \frac{\partial X_{2ih(t)}}{\partial t} &= \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} - \beta \cdot \text{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \\ &\left[2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] \end{aligned}$$

Substituindo a expressão acima determinada, juntamente com a Equação 4.51 na expressão de $\dot{S}_{ih(t)}$,

$$\begin{aligned} \dot{S}_{ih(t)} &= C_H \cdot \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right\} - C_H \cdot \beta \cdot |X_{1ih(t)}| \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 + \\ &\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \frac{\partial^2 Y_{dj(t)}}{\partial t^2} + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} - \beta \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \left[2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \right. \\ &\left. \frac{\partial |X_{1ih(t)}|}{\partial t} \right] \end{aligned}$$

$$\begin{aligned} \dot{S}_{ih(t)} &= -\beta \cdot \operatorname{sgn}(S_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \\ &\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \end{aligned}$$

Considerando a Equação 4.44, na convergência do sistema, necessariamente, $\operatorname{sgn}(S_{ih(t)}) = -\operatorname{sgn}(\dot{S}_{ih(t)})$, conseqüentemente,

$$\begin{aligned} \dot{S}_{ih(t)} &= \beta \cdot \operatorname{sgn}(\dot{S}_{ih(t)}) \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \\ &\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \end{aligned}$$

$$\begin{aligned} &\left\{ |\dot{S}_{ih(t)}| - \beta \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] \right\} \cdot \operatorname{sgn}(\dot{S}_{ih(t)}) = \\ &\left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right| \cdot \operatorname{sgn} \left(\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + \right. \right. \right. \\ &\left. \left. \left. C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right) \end{aligned}$$

$$\operatorname{sgn}(\dot{S}_{ih(t)}) = \operatorname{sgn} \left(\sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right)$$

$$\begin{aligned} \dot{S}_{ih(t)} &= \left\{ \beta \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \right. \\ &\left. \left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right| \right\} \cdot \operatorname{sgn}(\dot{S}_{ih(t)}) \end{aligned}$$

e, conseqüentemente,

$$\beta \cdot [T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right] + \left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right| > 0$$

$$\beta > - \frac{\left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right|}{[T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right]}$$

A inequação acima descrita, juntamente com a inequação A.22, define os limites para o ganho β .

$$\frac{\left| \frac{1}{2} \cdot \sum_{j=1}^p \left[C_H \cdot \frac{[X_{1ji(t)}]^2}{2} + X_{1ji(t)} \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] \right|}{|X_{1ih(t)}| \cdot [T_{h(k)}]^2} > \beta$$

$$> - \frac{\left| \sum_{j=1}^p \left\{ X_{1ji(t)} \cdot \left[\frac{\partial^2 Y_{dj(t)}}{\partial t^2} + C_H \cdot \frac{\partial Y_{dj(t)}}{\partial t} \right] + \frac{\partial Y_{dj(t)}}{\partial t} \cdot X_{2ji(t)} \right\} \right|}{[T_{h(k)}]^2 \cdot \left[C_H \cdot |X_{1ih(t)}| + 2 \cdot \frac{|X_{1ih(t)}|}{T_{h(k)}} \cdot \frac{\partial T_{h(k)}}{\partial t} + \frac{\partial |X_{1ih(t)}|}{\partial t} \right]}$$

A.7 Detalhamento matemático para o Treinamento *On-line* - 1a. proposta

A.7.1 Camada de saída

Substituindo a Equação 4.56 e A.8 em 4.57:

$$X_{2ji(k)} = \frac{X_{1ji(k)} - X_{1ji(k-1)}}{T}$$

$$X_{2ji(k)} = \frac{1}{T} [-e_{j(k)} \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot Y_{Hi(k-1)}]$$

Considerando-se a Equação A.5 e a definição A.6 para as variáveis $f' (V_{j(k-1)})$ e $Y_{Hi(k-1)}$,

$$T \cdot X_{2ji(k)} = -e_{j(k)} \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [f' (V_{j(k)}) - \Delta f' (V_{j(k)})] \cdot (Y_{Hi(k)} - \Delta Y_{Hi(k)})$$

$$T \cdot X_{2ji(k)} = -(Y_{dj(k)} - Y_{j(k)}) \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + (Y_{dj(k-1)} - Y_{j(k-1)}) \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + (Y_{dj(k-1)} - Y_{j(k-1)}) \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

$$T \cdot X_{2ji(k)} = [-Y_{dj(k)} + Y_{dj(k-1)} + Y_{j(k)} - Y_{j(k-1)}] \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + (Y_{dj(k-1)} - Y_{j(k-1)}) \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

novamente, utilizando a definição A.6 para a variável $Y_{j(k)}$

$$T \cdot X_{2ji(k)} = [-\Delta Y_{dj(k)} + \Delta Y_{j(k)}] \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Utilizando-se a Equação A.15 e 4.61 na expressão acima, obtém-se:

$$T \cdot X_{2ji(k)} = [-\Delta Y_{dj(k)} + f' (V_{j(k)}) \cdot Y_{Hi(k)} \cdot (-\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn} (S_{ji(k)}))] \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

$$T \cdot X_{2ji(k)} = -\alpha \cdot |X_{1ji(k)}| \cdot \text{sgn} (S_{ji(k)}) \cdot [f' (V_{j(k)}) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + e_{j(k-1)} \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Substituindo a Equação 4.56 e a expressão acima determinada em 4.55:

$$S_{ji(k)} = -C \cdot e_{j(k)} \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} - \frac{1}{T} \cdot \{ \alpha \cdot |X_{1ji(k)}| \cdot \text{sgn} (S_{ji(k)}) \cdot [f' (V_{j(k)}) \cdot Y_{Hi(k)}]^2 - \Delta Y_{dj(k)} \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + (Y_{dj(k-1)} - Y_{dj(k-1)}) \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}] \}$$

$$S_{ji(k)} = -\frac{1}{T} \cdot \alpha \cdot |X_{1ji(k)}| \cdot [f' (V_{j(k)}) \cdot Y_{Hi(k)}]^2 \cdot \text{sgn} (S_{ji(k)}) - \left[C \cdot e_{j(k)} + \frac{\Delta Y_{dj(k)}}{T} \right] \cdot f' (V_{j(k)}) \cdot Y_{Hi(k)} + \frac{1}{T} \cdot e_{j(k-1)} \cdot [\Delta f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f' (V_{j(k)}) \cdot Y_{Hi(k)} - f' (V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

Define-se, portanto:

$$X_{(k)} = \frac{|X_{1ji(k)}| \cdot [f'(V_{j(k)}) \cdot Y_{Hi(k)}]^2}{T}$$

$$Y_{(k)} = - \left[C \cdot e_{j(k)} + \frac{\Delta Y_{aj(k)}}{T} \right] \cdot f'(V_{j(k)}) \cdot Y_{Hi(k)} + \frac{1}{T} \cdot e_{j(k-1)} \cdot [\Delta f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)} - \Delta f'(V_{j(k)}) \cdot Y_{Hi(k)} - f'(V_{j(k)}) \cdot \Delta Y_{Hi(k)}]$$

A.7.2 Camada escondida

A análise do estudo da convergência para a camada escondida é realizada de forma semelhante à utilizada para a camada de saída.

Desta forma, substituindo-se as Equações 4.59 e A.11 em 4.60 tem-se:

$$T \cdot X_{2ih(k)} = -f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + f'_H(R_{i(k-1)}) \cdot T_{h(k-1)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Utilizando-se a definição A.14 para as variáveis $f'_H(R_{i(k-1)})$ e $T_{h(k-1)}$:

$$T \cdot X_{2ih(k)} = -f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [f'_H(R_{i(k)}) - \Delta f'_H(R_{i(k)})] \cdot [T_{h(k)} - \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$T \cdot X_{2ih(k)} = -f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ - [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

novamente, considerando a Equação A.5 e a definição A.6 para as variáveis $f'(V_{j(k-1)})$ e $W_{ji(k-1)}$:

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ - [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (f'(V_{j(k)}) - \Delta f'(V_{j(k)})) \cdot (W_{ji(k)} - \Delta W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{ [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-Y_{dj(k)} + Y_{j(k)} + Y_{dj(k-1)} - Y_{j(k-1)})] + [(Y_{dj(k-1)} - Y_{j(k-1)}) \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] \} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [(Y_{dj(k-1)} - Y_{j(k-1)}) \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

admitindo a definição A.6, tem-se:

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-\Delta Y_{dj(k)} + \Delta Y_{j(k)})] + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Substituindo a Equação A.16 e 4.62 na equação acima,

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)} \cdot (-\Delta Y_{dj(k)} + f'(V_{j(k)}) \cdot$$

$$f'_H(R_{i(k)}) \cdot W_{ji(k)} \cdot T_{h(k)} \cdot (-\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)})) + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})] + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$T \cdot X_{2ih(k)} = f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{-\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 \cdot f'_H(R_{i(k)}) \cdot T_{h(k)}\} + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$T \cdot X_{2ih(k)} = -\beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

Substituindo a expressão acima determinada e a Equação 4.12 na Equação 4.11, obtém-se:

$$S_{ih(k)} = -C_H \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p [e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - \frac{1}{T} \cdot \beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + \frac{1}{T} \cdot f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \{[-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)})]\} + \frac{1}{T} \cdot [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]$$

$$S_{ih(k)} = -\frac{1}{T} \cdot \beta \cdot |X_{1ih(k)}| \cdot \text{sgn}(S_{ih(k)}) \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2 + f'_H(R_{i(k)}) \cdot$$

$$\begin{aligned}
 & T_{h(k)} \cdot \sum_{j=1}^p \left\{ \frac{1}{T} \cdot [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] - [C_H \cdot e_{j(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] + \frac{1}{T} \cdot [e_{j(k-1)} \cdot \right. \\
 & \left. (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)}) \right\} + \\
 & \frac{1}{T} \cdot [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [e_{j(k-1)} \cdot \\
 & f'(V_{j(k-1)}) \cdot W_{ji(k-1)}]
 \end{aligned}$$

Utilizando-se a Equação A.9, define-se, portanto, as seguintes variáveis

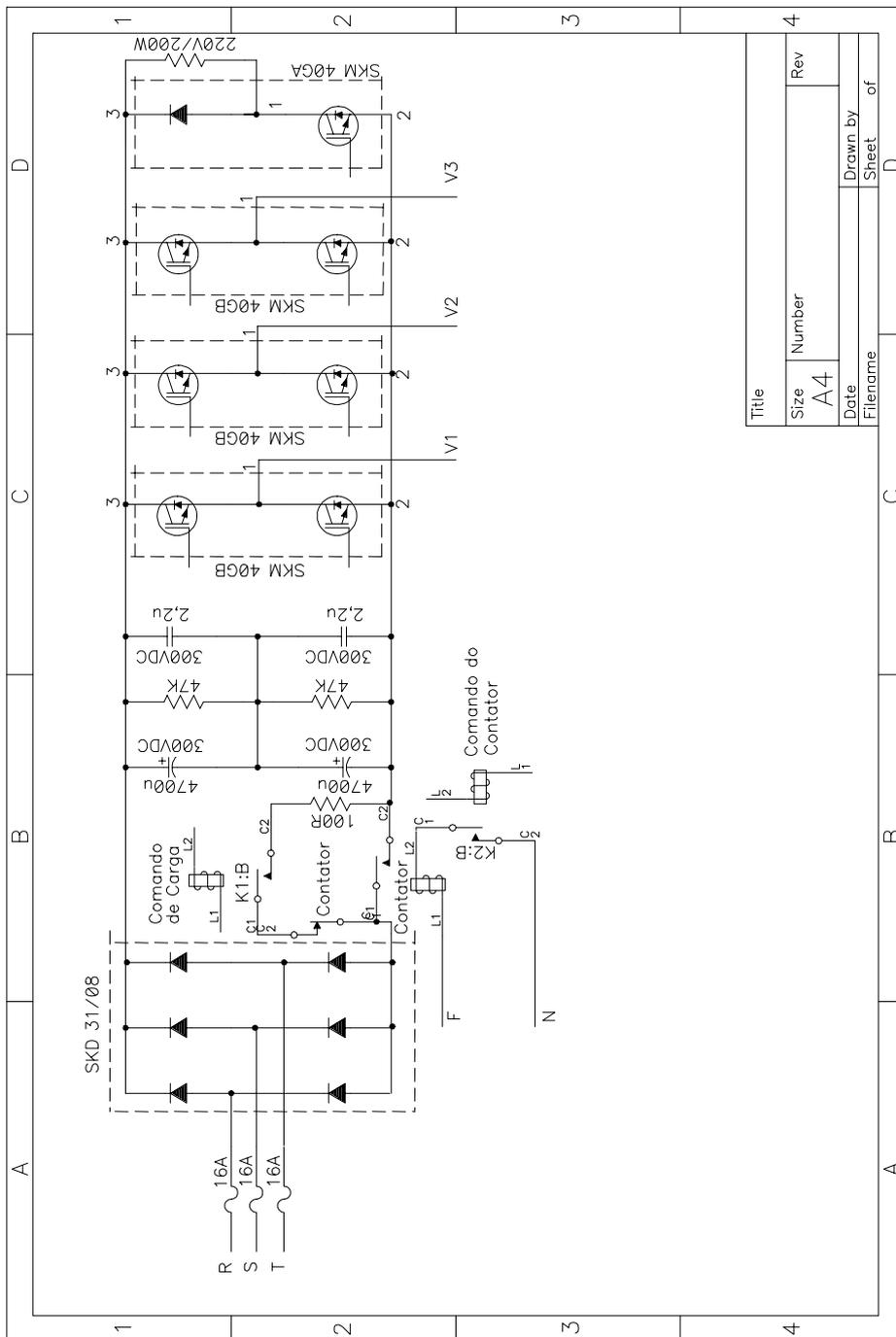
$$X_{(k)} = \frac{1}{T} \cdot |X_{1ih(k)}| \cdot [f'_H(R_{i(k)}) \cdot T_{h(k)}]^2 \cdot \sum_{j=1}^p [f'(V_{j(k)}) \cdot W_{ji(k)}]^2$$

$$\begin{aligned}
 Y_{(k)} = & f'_H(R_{i(k)}) \cdot T_{h(k)} \cdot \sum_{j=1}^p \left\{ \frac{1}{T} \cdot [-\Delta Y_{dj(k)} \cdot f'(V_{j(k)}) \cdot W_{ji(k)}] \right. \\
 & \left. - [C_H \cdot \delta_{j(k)} \cdot W_{ji(k)}] + \frac{1}{T} \cdot [e_{j(k-1)} \cdot (\Delta f'(V_{j(k)}) \cdot \Delta W_{ji(k)} \right. \\
 & \left. - f'(V_{j(k)}) \cdot \Delta W_{ji(k)} - \Delta f'(V_{j(k)}) \cdot W_{ji(k)}) \right\} + \frac{1}{T} \cdot [\Delta f'_H(R_{i(k)}) \cdot \Delta T_{h(k)} \\
 & - \Delta f'_H(R_{i(k)}) \cdot T_{h(k)} - f'_H(R_{i(k)}) \cdot \Delta T_{h(k)}] \cdot \sum_{j=1}^p [\delta_{j(k-1)} \cdot W_{ji(k-1)}]
 \end{aligned}$$

Apêndice B

Diagramas Esquemáticos da Bancada Experimental

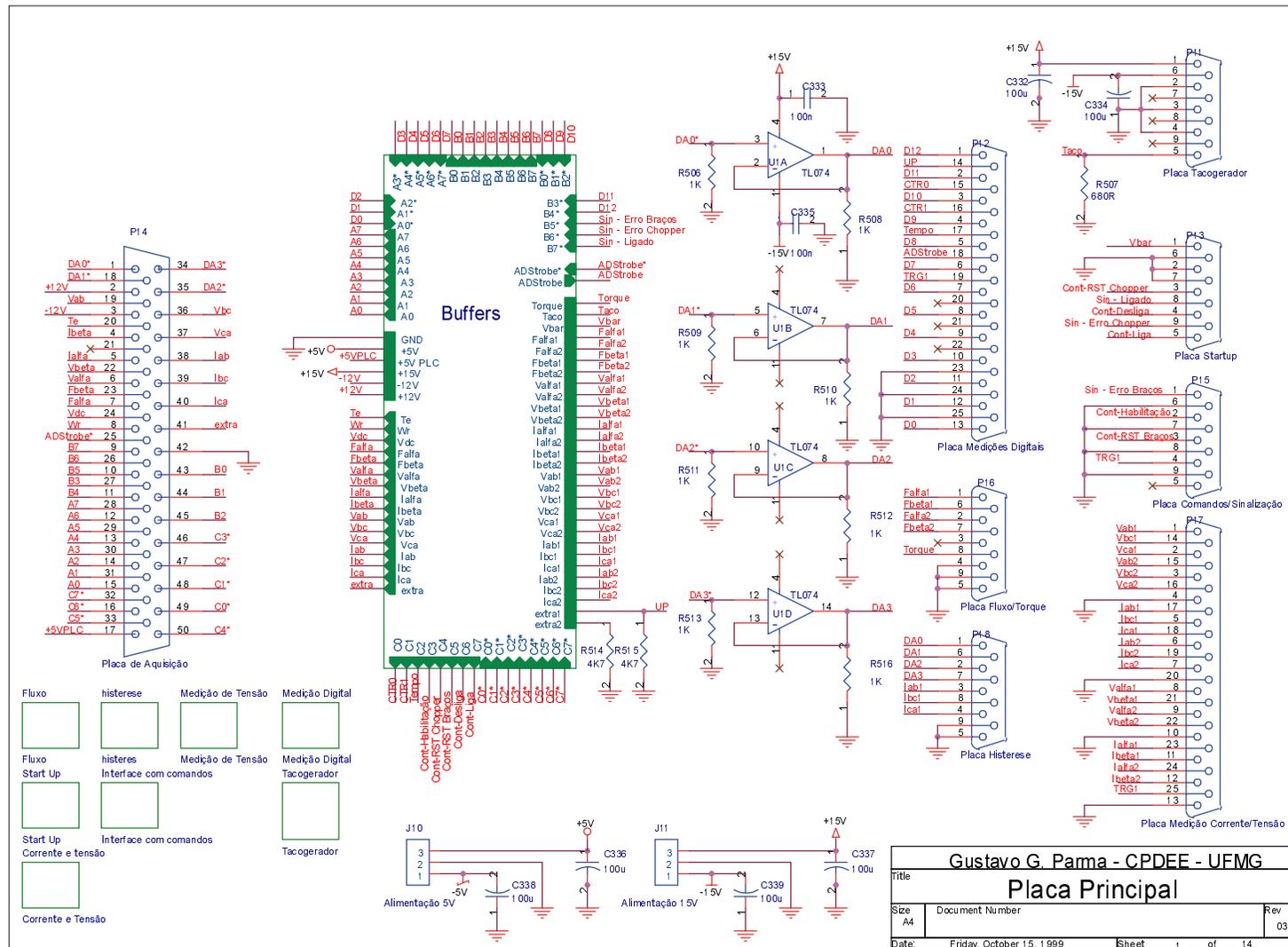
Neste apêndice estão os circuitos elétricos da bancada experimental desenvolvida neste trabalho.



Title	
Size	Number
A4	
Date	Drawn by
Filename	Sheet of
	4

Figura B.1. Diagrama de ligações do barramento DC utilizado pela bancada experimental desenvolvida.

Figura B.2. Circuito elétrico da placa de interface.



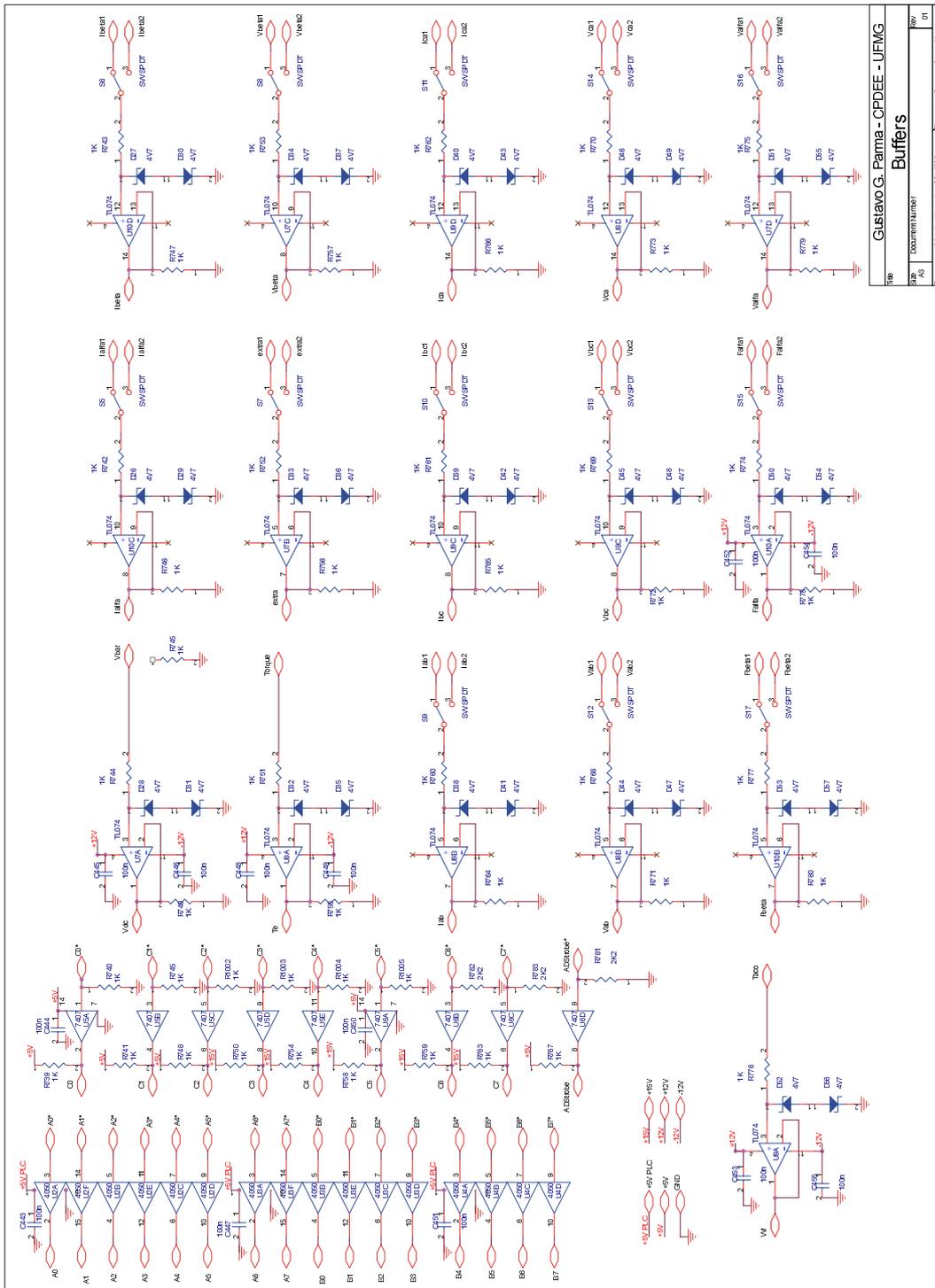


Figura B.3. Circuito elétrico dos buffers utilizados pela placa de interface.

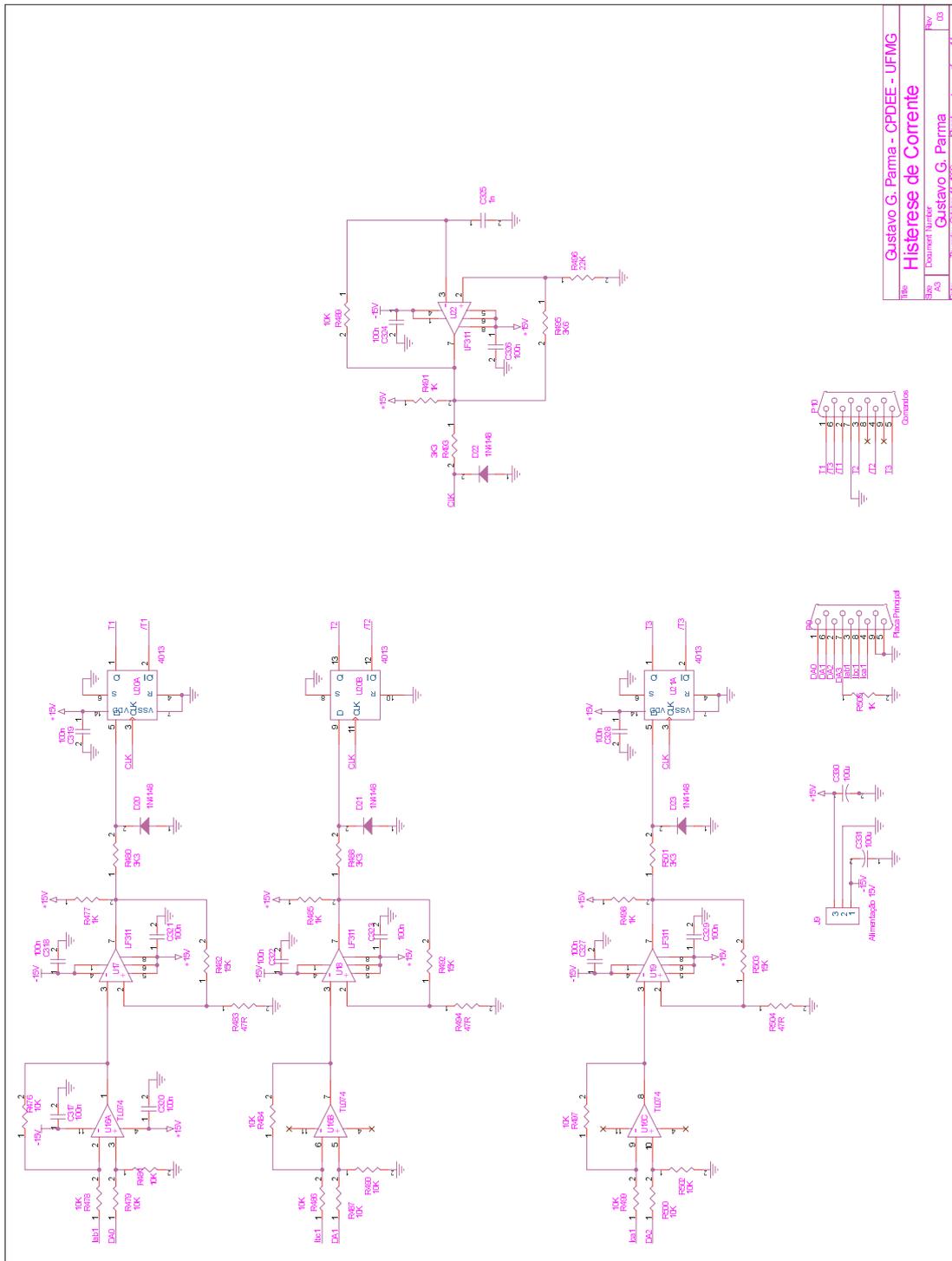
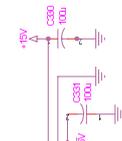
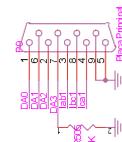
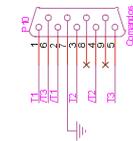


Figura B.5. Circuito de histerese desenvolvido.



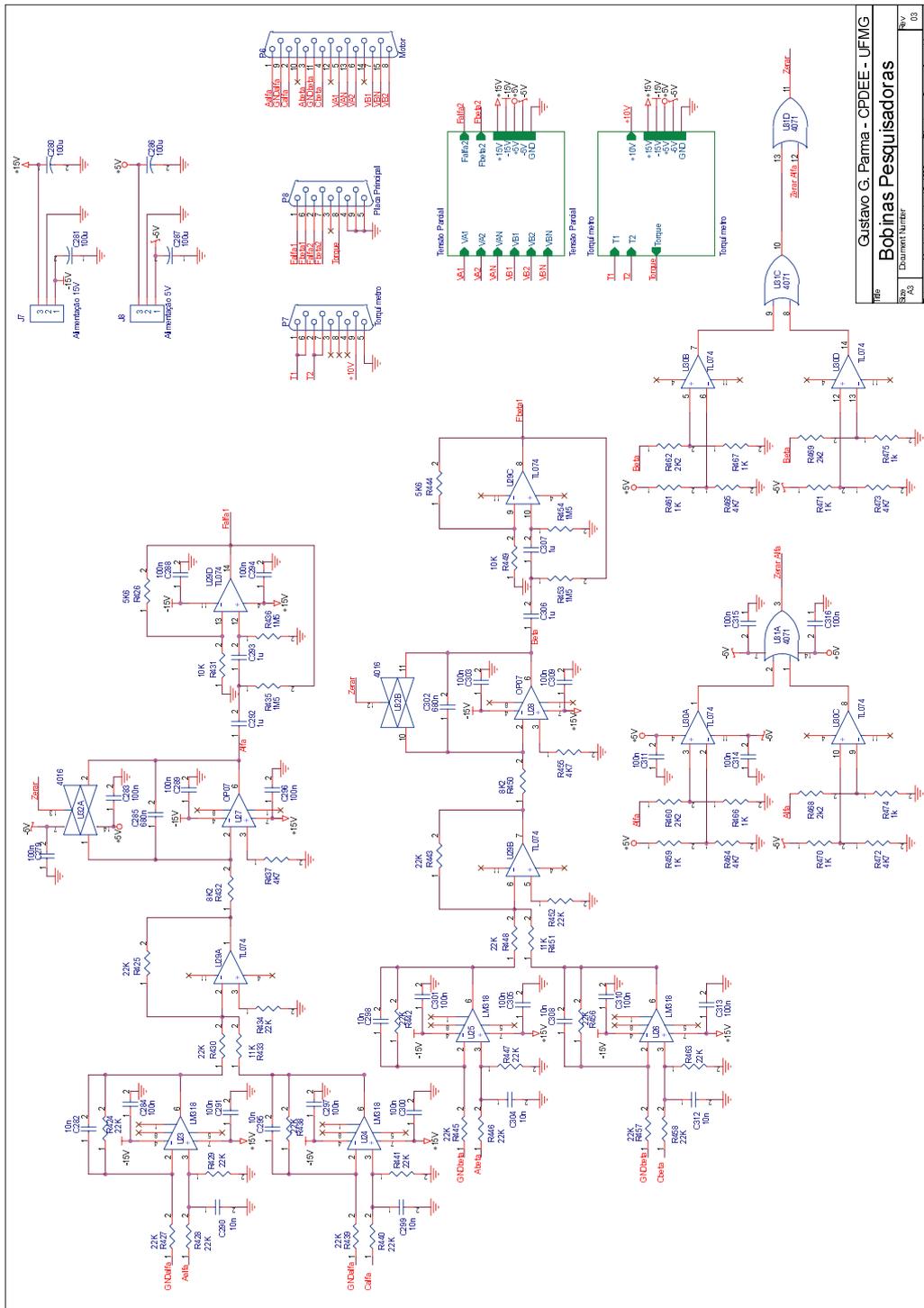


Figura B.6. Circuito eletrônico para a medição de fluxo utilizando bobinas pesquisadoras.

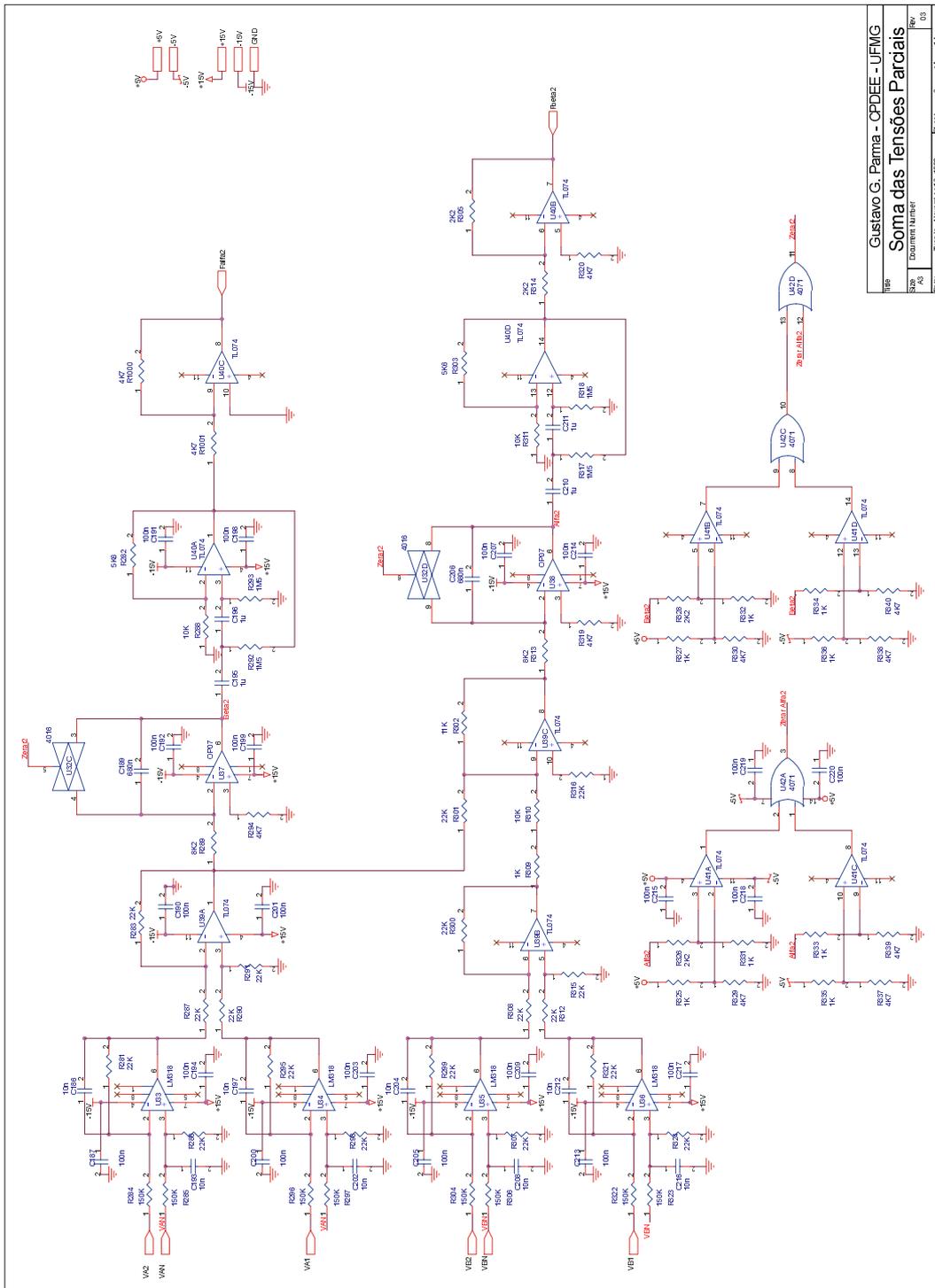


Figura B.7. Circuito para a medição de fluxo utilizando tensões parciais.

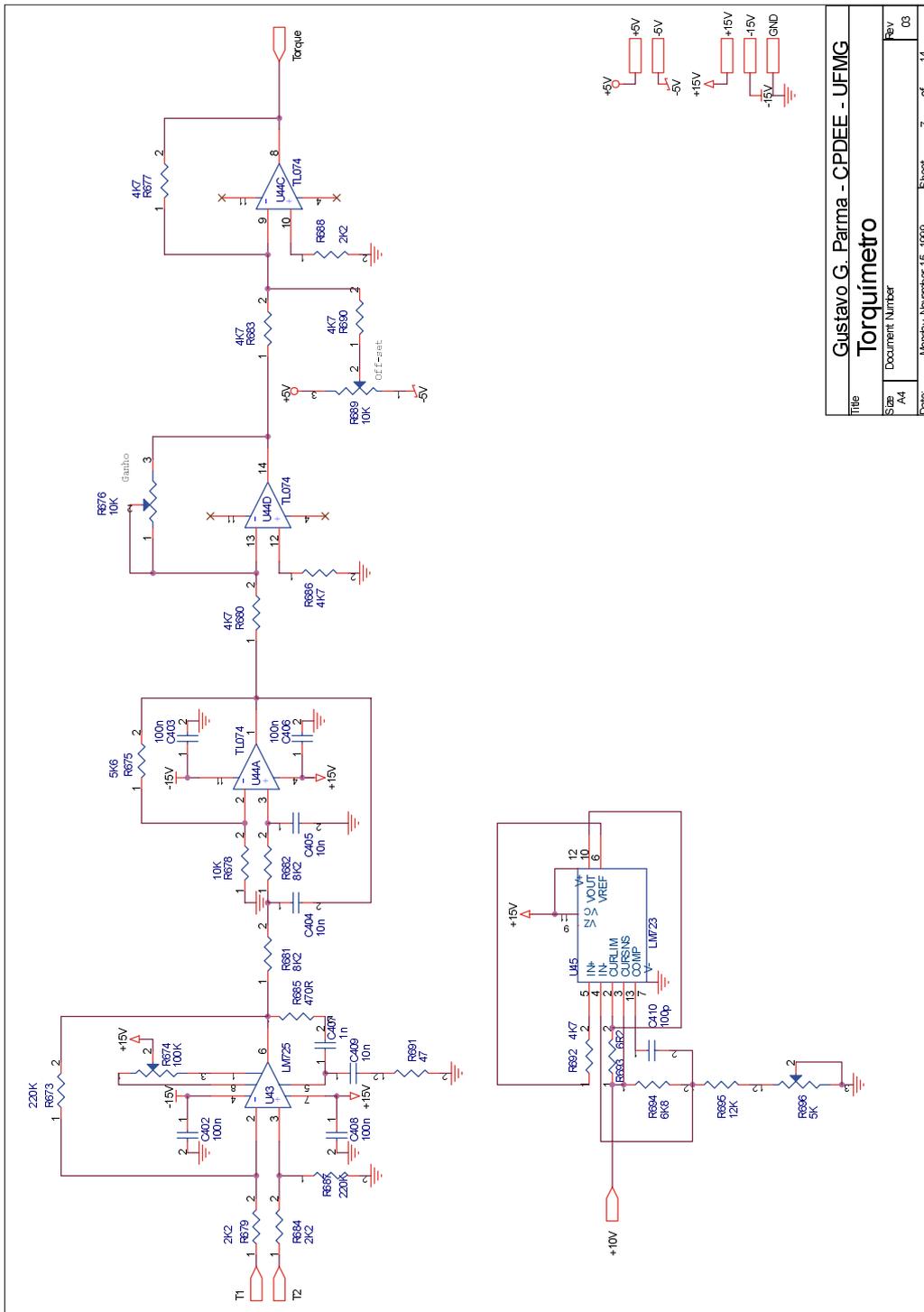
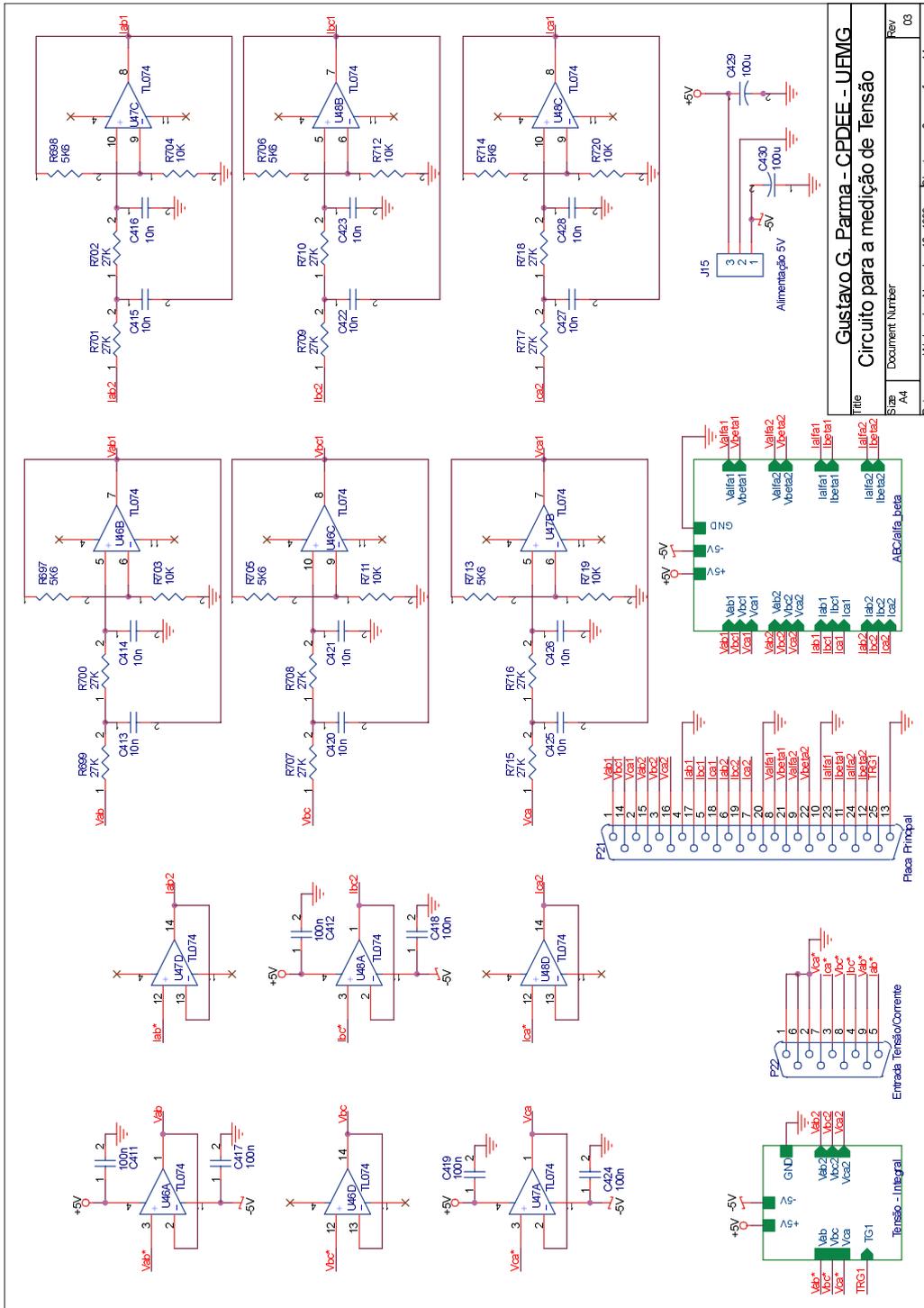


Figura B.8. Circuito para a medição do torque.

Title		Gustavo G. Parma - CPDEE - UFMG	
Size		Torquímetro	
Sheet		Document Number	
Date		Rev	
Monday, November 15, 1999		A4	
Sheet		7 of 14	



Gustavo G. Parma - CPDEE - UFMG
 Circuito para a medição de Tensão

File
 Size A4
 Document Number
 Date: Wednesday, November 10, 1999 Sheet 8 of 14

Figura B.9. Circuito para a medição analógica das tensões e correntes.

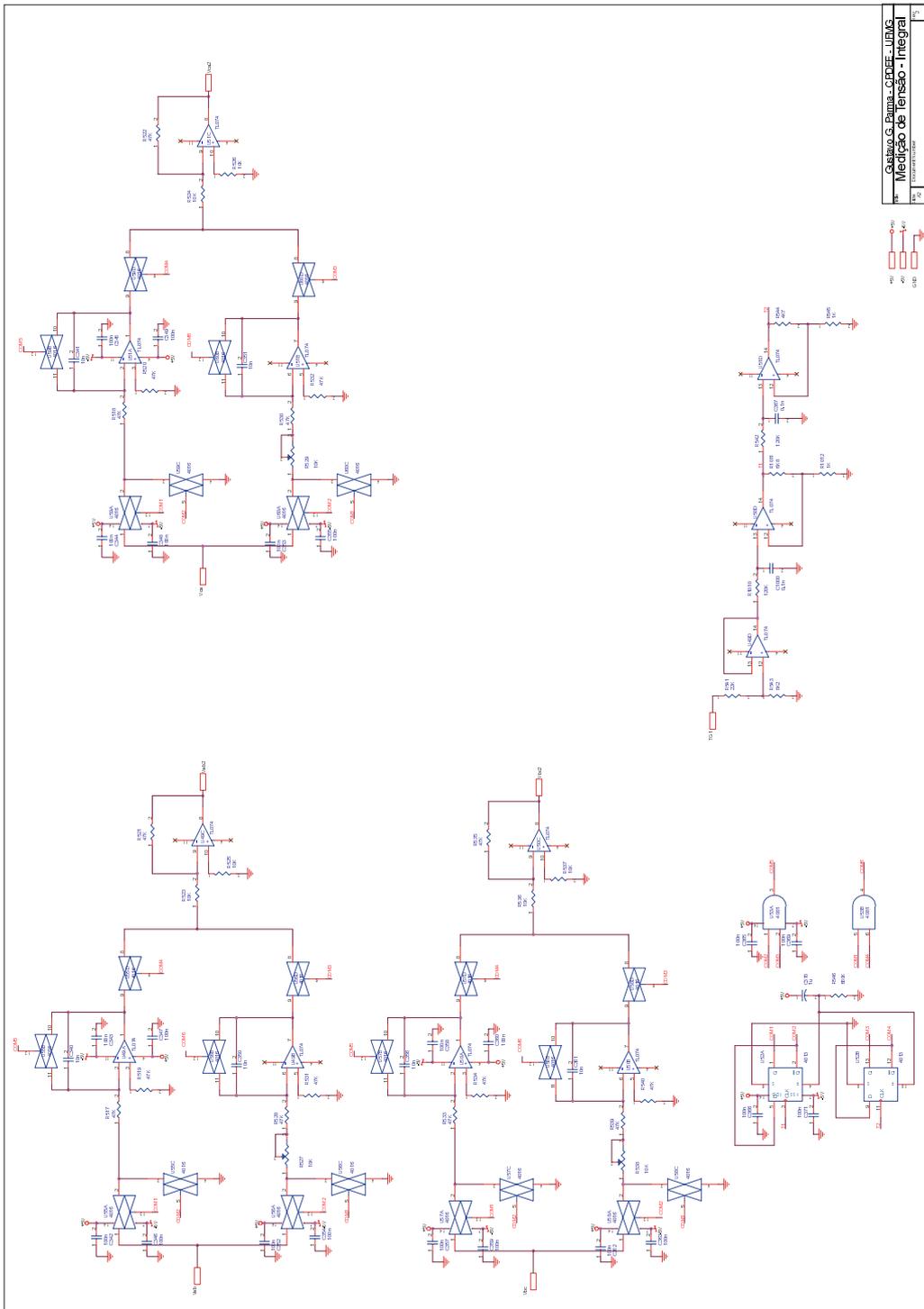


Figura B.10. Circuito para a medição da tensão aplicada à máquina, utilizando a integração no período de PWM.

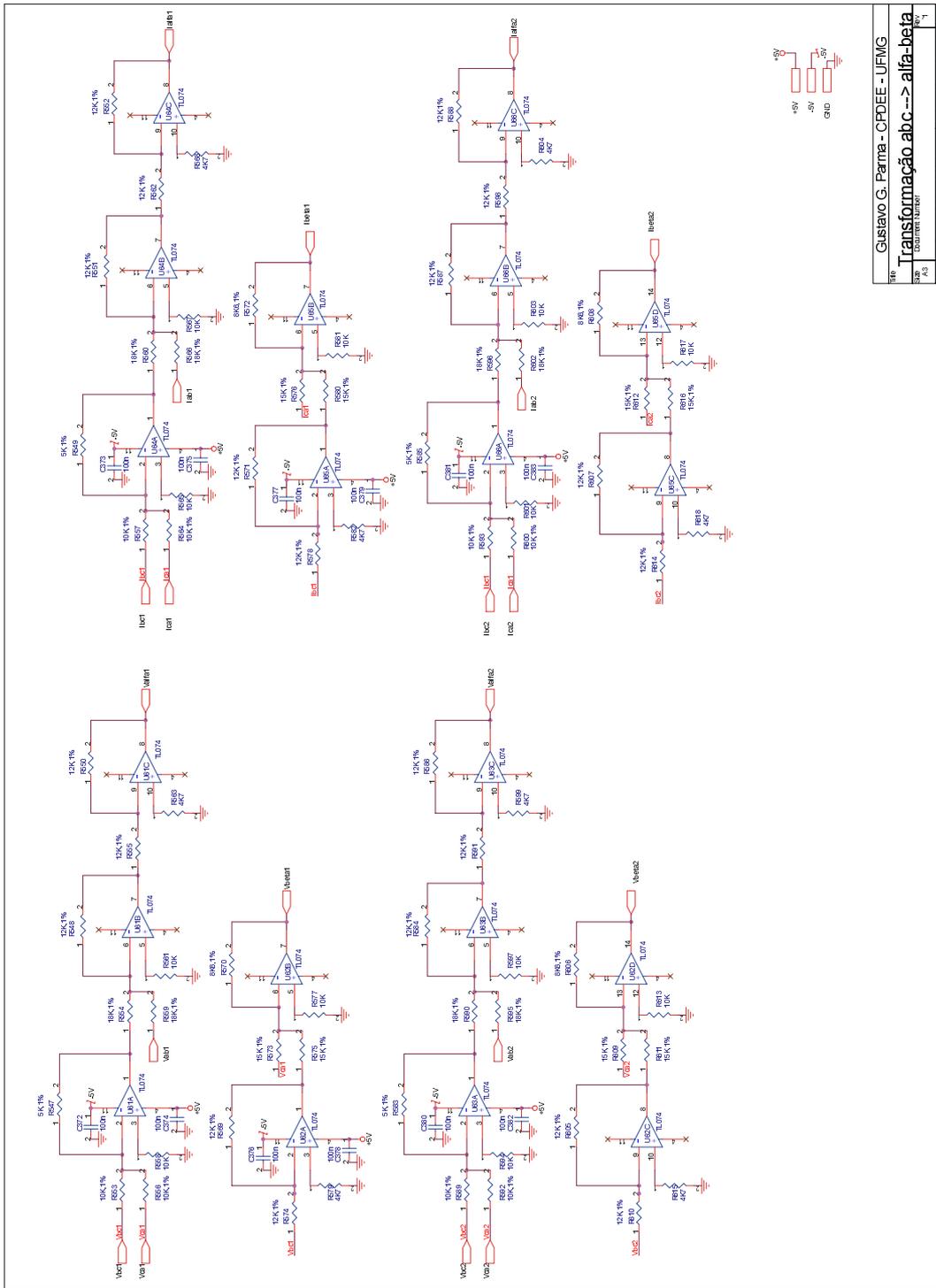
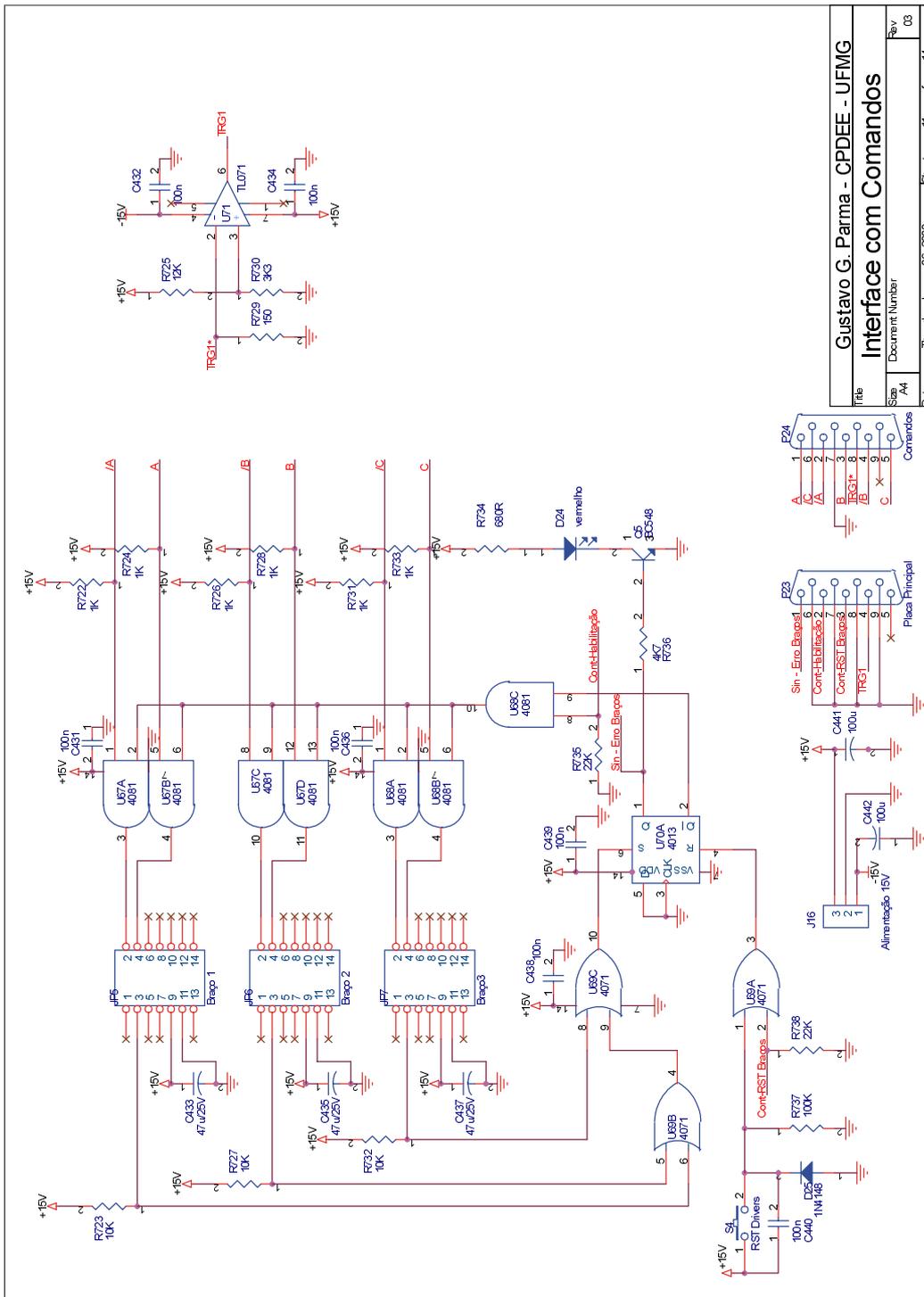


Figura B.11. Circuito para a transformação das grandezas trifásicas em grandezas do sistema de eixos $\alpha\beta$ fixos no estator.



Gustavo G. Parma - CPDEE - UFMG			
Interface com Comandos			
File	Document Number	Rev	03
Sheet	744		
Date	Thursday, January 06, 2000	Sheet	11 of 14

Figura B.12. Circuito para a interface entre as placas de comando dos IGBTs e o inversor.

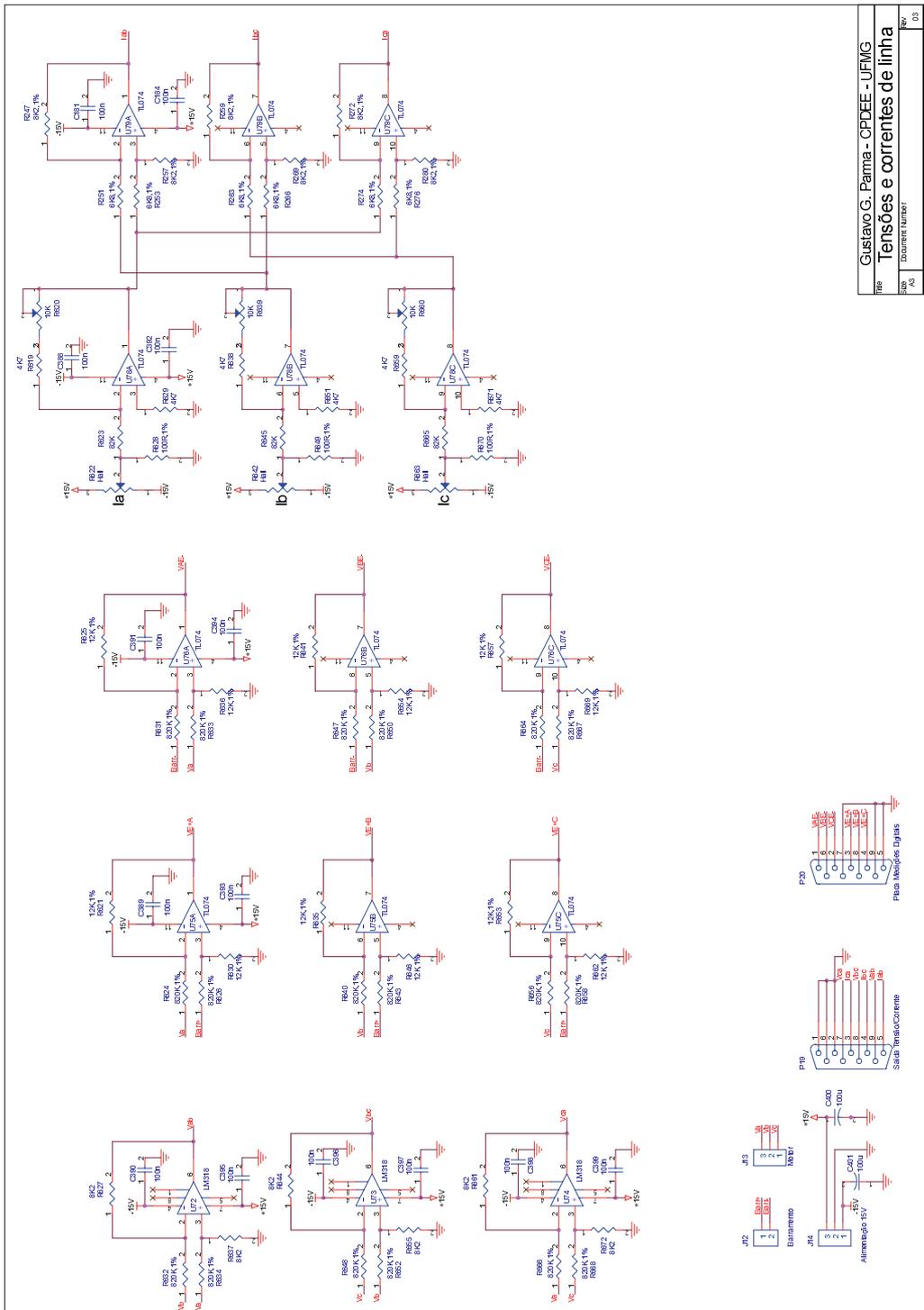
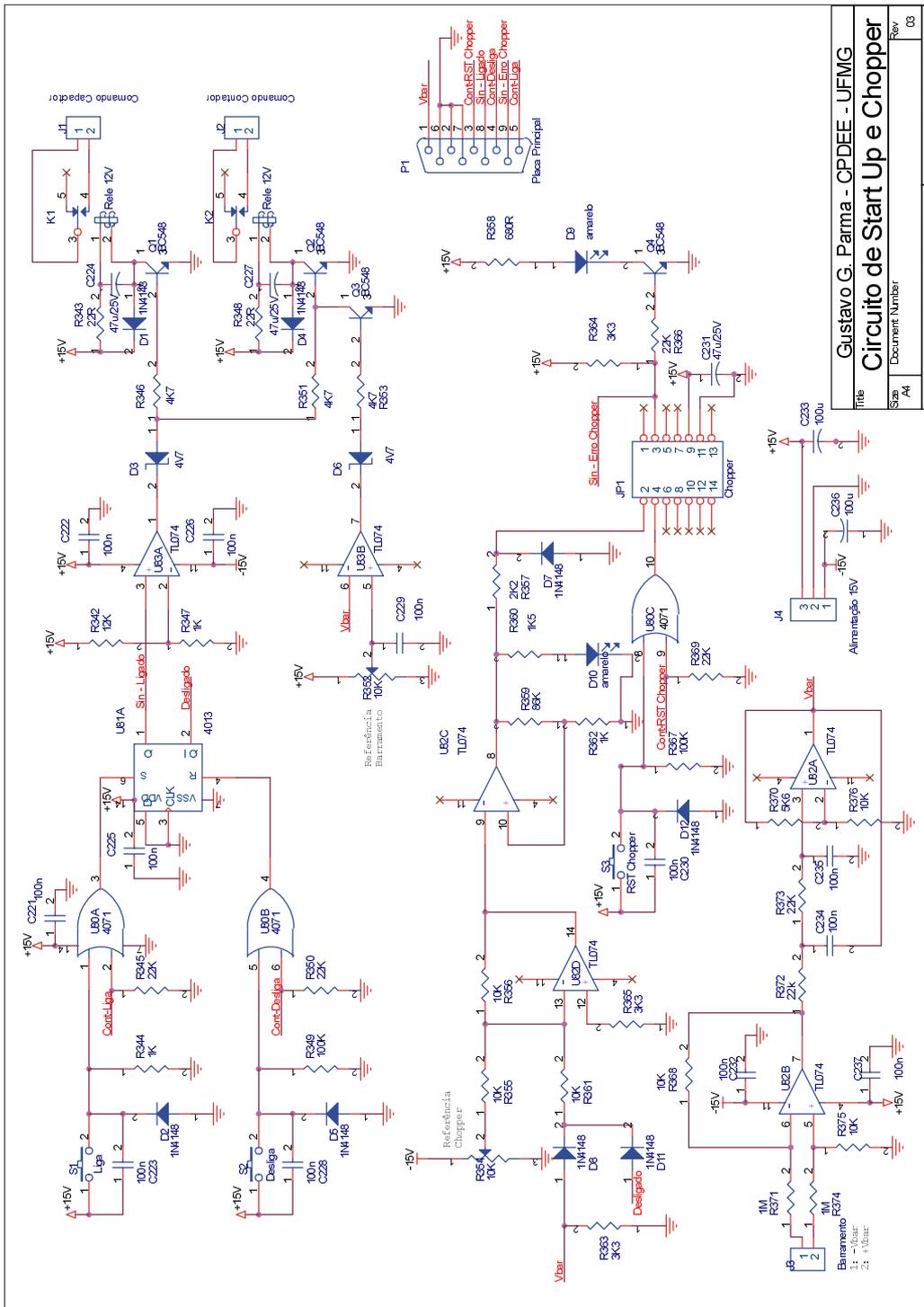


Figura B.13. Circuito para a aquisição das correntes e tensões na máquina.



Gustavo G. Parma - CPDEE - UFMG	
Title	Circuito de Start Up e Chopper
Size	A4
Document Number	
Date	Monday, November 22, 1999
Sheet	13 of 14
Rev	03

Figura B.14. Circuito para o controle da tensão do barramento DC.

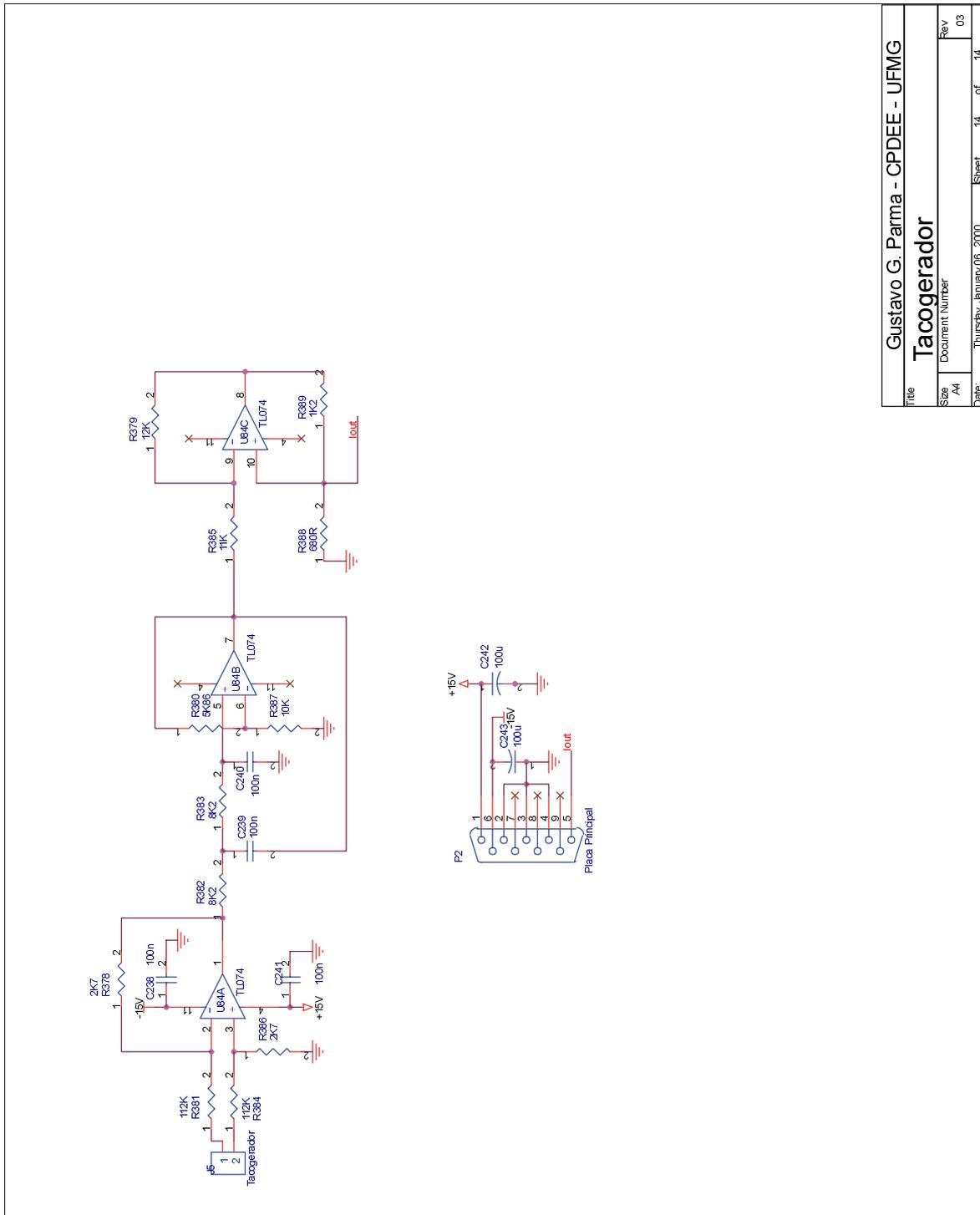


Figura B.15. Circuito para a medição da velocidade utilizando o tacogerador.

Title		Gustavo G. Parma - CPDEE - UFMG
Size		A4
Document Number		Rev 03
Date:	Thursday, January 06, 2000	Sheet 14 of 14

Apêndice C

Medição de Fluxo - Bobinas Exploradoras

Bobinas exploradoras são espiras feitas de metal fino que podem ser centradas em um dente ou em uma ranhura do estator. Devido a presença de ranhuras, a distribuição do fluxo da máquina não é perfeitamente senoidal, como desejado, apresentando diversos harmônicos que tendem a distorcer o sinal medido.

Sendo p o número de pares de pólos, T o número de ranhuras de estator e R o número de ranhuras de rotor, sabe-se [Lipo, 1977] que as ranhuras de estator produzirão campos com $T - p$ e $T + p$ pares de pólos girando no mesmo sentido e no sentido contrário, respectivamente, da velocidade síncrona. As ranhuras de rotor produzirão campos com $R + p$ e $R - p$ pares de pólos. Os pólos do rotor irão girar com a velocidade dada pela seguinte equação, relativo ao estator.

$$\omega = 1 \pm \frac{(1 - s) \cdot R}{p} \quad (\text{C.1})$$

onde s é o escorregamento.

Para máquinas pequenas (abaixo de 100 HP), de rotor fundido, os harmônicos produzidos pelas ranhuras de rotor podem ser desconsiderados se as ranhuras forem dispostas inclinadamente ("skewing").

Além dos harmônicos produzidos pelas ranhuras, existem ainda os harmônicos devido ao número discreto de ranhuras de estator. Os harmônicos de ordem ímpar são somados à

fundamental, distorcendo o sinal medido. Os principais harmônicos são o quinto e o sétimo, o primeiro girando negativamente e o segundo positivamente. Deve-se lembrar, também, do terceiro harmônico e múltiplos que começam a surgir com a saturação da máquina.

Todos estes fatores são fontes de erros indesejados, dessa forma, as bobinas exploradoras devem ser construídas de tal forma a minimizar, e até eliminar, as perturbações mais significativas, as quais são: O terceiro harmônico, e as componentes $R + p$ e $R - p$ do rotor.

C.1 Aspectos Construtivos

O eixo magnético de uma máquina pode ser centrado em um dente ou uma ranhura, dependendo do número de ranhuras por polo. A máquina utilizada possui dois pares de pólos ($p = 2$) e vinte e quatro ranhuras de estator ($T = 24$), tendo, pois, um número par de ranhuras por polo. Dessa forma, os eixos estarão centrados ambos em um dente ou ambos em uma ranhura. Foi adotada a configuração dos eixos sobre um dente, como mostra a Figura C.1.

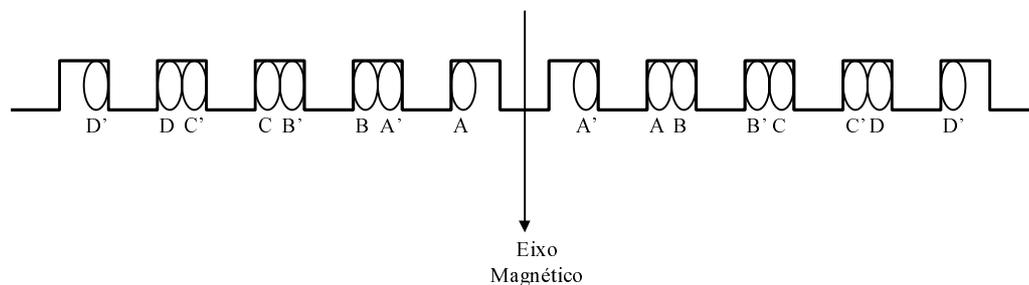


Figura C.1. Bobinas Pesquisadora: Configuração adotada. Eixo magnético centrado em um dente.

Na configuração adotada, convencionando-se a entrada de corrente induzida pela letra com apóstrofe e a saída de corrente induzida pela letra sem apóstrofe, a tensão induzida total é proporcional ao número de espiras de cada uma das bobinas (A'A, B'B, C'C, D'D), podendo ser calculada por:

$$V_a = -N \cdot \frac{d\phi(t)}{dt} \quad (C.2)$$

onde V_a é a tensão total induzida; N o número de espiras da bobina e ϕ o fluxo concatenado

por uma espira da bobina.

Na Figura C.1, as letras (A'A, B'B, C'C, D'D), além de identificar as bobinas, também se referem ao número de espiras por bobina. Ou seja, a bobina A'A possui A espiras.

O fluxo é, então, determinado pela integração da equação acima descrita, ou seja,

$$\phi_{(t)} = -\frac{1}{N} \cdot \int_0^t V_a \cdot dt \quad (\text{C.3})$$

Considerando a disposição adotada para as bobinas exploradoras (Figura C.1), obtém-se as seguintes equações, as quais são resolvidas para que tenha-se a minimização dos harmônicos indesejados e a maximização do fluxo resultante.

- Componente fundamental (campo p e $T \pm p$). Deve resultar na maior tensão induzida possível:

$$\begin{aligned} (D - C) \cdot \sin\left(\frac{7 \cdot \pi \cdot p}{T}\right) + (C - B) \cdot \sin\left(\frac{5 \cdot \pi \cdot p}{T}\right) + \\ (B - A) \cdot \sin\left(\frac{3 \cdot \pi \cdot p}{T}\right) + A \cdot \sin\left(\frac{\pi \cdot p}{T}\right) = K \end{aligned} \quad (\text{C.4})$$

- Campos $R - p$ e $R + p$, produzidos pelo Rotor. Tensões indesejadas, devendo ser nulas:

$$\begin{aligned} (D - C) \cdot \sin\left(\frac{7 \cdot \pi \cdot (R-p)}{T}\right) + (C - B) \cdot \sin\left(\frac{5 \cdot \pi \cdot (R-p)}{T}\right) + \\ (B - A) \cdot \sin\left(\frac{3 \cdot \pi \cdot (R-p)}{T}\right) + A \cdot \sin\left(\frac{\pi \cdot (R-p)}{T}\right) = 0 \end{aligned} \quad (\text{C.5})$$

$$\begin{aligned} (D - C) \cdot \sin\left(\frac{7 \cdot \pi \cdot (R+p)}{T}\right) + (C - B) \cdot \sin\left(\frac{5 \cdot \pi \cdot (R+p)}{T}\right) + \\ (B - A) \cdot \sin\left(\frac{3 \cdot \pi \cdot (R+p)}{T}\right) + A \cdot \sin\left(\frac{\pi \cdot (R+p)}{T}\right) = 0 \end{aligned} \quad (\text{C.6})$$

- Componente do terceiro harmônico. Tensão indesejada, devendo ser anulada.

$$\begin{aligned} (D - C) \cdot \sin\left(\frac{21 \cdot \pi \cdot p}{T}\right) + (C - B) \cdot \sin\left(\frac{15 \cdot \pi \cdot p}{T}\right) + \\ (B - A) \cdot \sin\left(\frac{9 \cdot \pi \cdot p}{T}\right) + A \cdot \sin\left(\frac{3 \cdot \pi \cdot p}{T}\right) = 0 \end{aligned} \quad (\text{C.7})$$

Dividindo as Equações C.4-C.7 por A, pode-se resolver o sistema de equações em função das variáveis D/A , C/A e B/A , obtendo-se um resultado relativo ao número de espiras

utilizado na bobina A . Resolvendo o sistema para a máquina utilizada, os seguintes resultados foram obtidos:

$$\frac{B}{A} = 0 \quad (\text{C.8})$$

$$\frac{C}{A} = 0.5 \quad (\text{C.9})$$

$$\frac{D}{A} = 0 \quad (\text{C.10})$$

$$\frac{K}{A} = -0.4483 \quad (\text{C.11})$$

Utilizando o sistema de eixos $\alpha\beta$ fixos no estator, com o eixo α sobre o eixo da fase a , considerando ainda que a máquina utilizada possui dois pares de polos, o espaço físico do estator pode ser dividido com quatro conjuntos de pares ordenados $\alpha\beta$, simetricamente distribuídos (Figura C.2), sendo que todos os eixos α (ou β) apresentam o mesmo valor instantâneo. Desta forma, na implementação prática das bobinas exploradoras, serão utilizados dois eixos α e β não subseqüentes, facilitando a implementação do sistema.

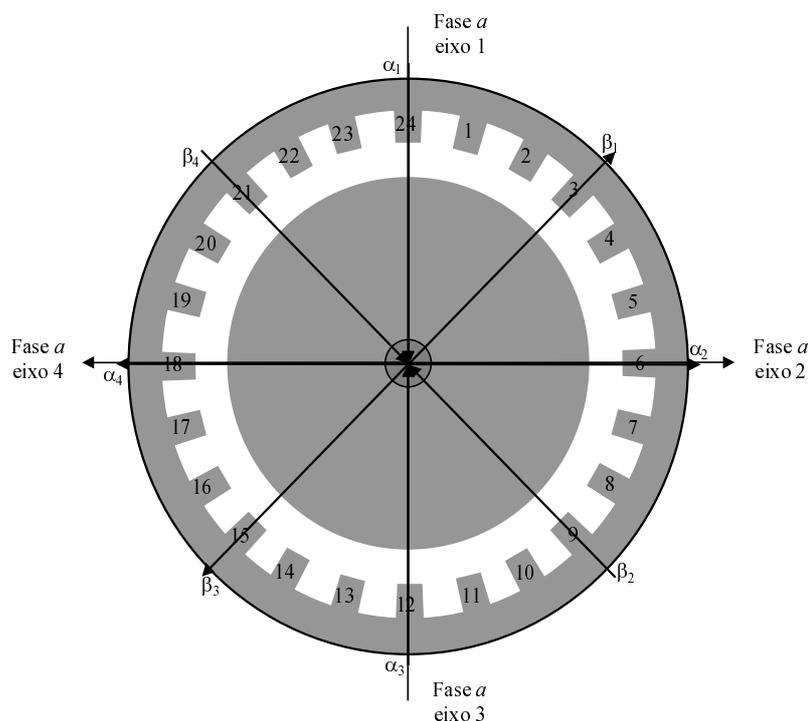


Figura C.2. Bobinas pesquisadoras: Distribuição espacial dos eixos $\alpha\beta$ no estator da máquina utilizada. Os números indicados se referem às ranhuras do estator.

Considerando que tanto o eixo α quanto o eixo β estão centrados em um dente, a tensão induzida em ambos é similar quanto à forma bastando, portanto, determinar o número de espiras para apenas um dos eixos e reproduzindo o resultado no outro eixo. Uma vez que é necessário integrar a tensão resultante das bobinas exploradoras em cada eixo (Equação C.3), optou-se por fazer todas as bobinas exploradoras com o mesmo número de espiras, sendo que a relação necessária para a eliminação das tensões indesejadas (Equações C.8 à C.11) é implementada na relação de ganho do integrador. A Figura C.3 mostra o diagrama esquemático do sistema implementado.

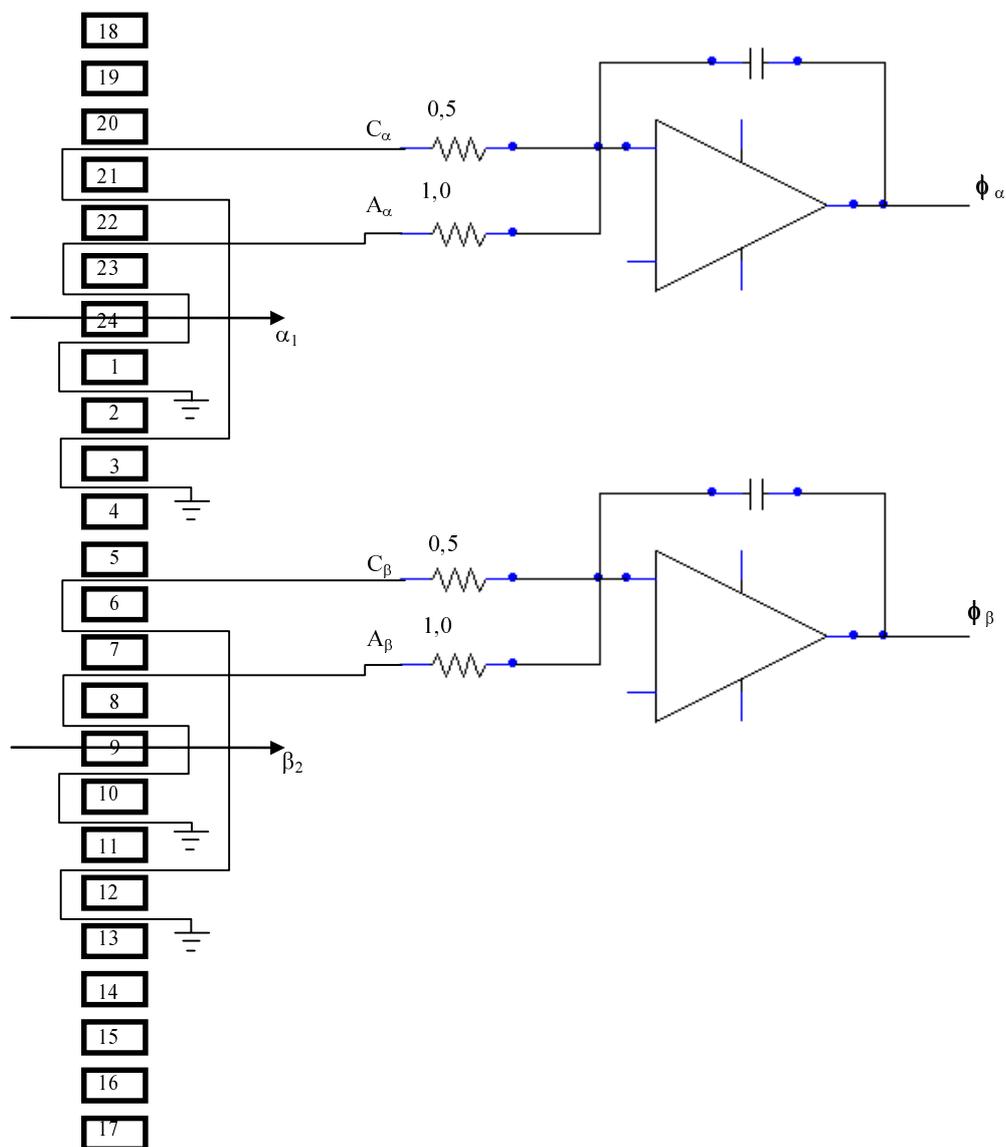


Figura C.3. Implementação prática das bobinas pesquisadoras para o eixo α e β . A numeração refere-se ao número de ranhuras de estator.

Apêndice D

Medição de Fluxo - Soma das Tensões Parciais

Ao contrário da utilização de bobinas exploradoras, as quais requerem um maior trabalho para a instalação e manutenção, pode-se utilizar as próprias bobinas dos enrolamentos da máquina para a medição do fluxo de estator [Lipo & Chang, 1986].

A Figura D.1 mostra a disposição espacial das bobinas de estator da máquina utilizada (24 ranhuras e 2 pares de pólos). As bobinas mostradas se referem a apenas uma das três fases do motor (fase a). O método de medição de fluxo utilizado só se aplica à disposição das bobinas como mostrada na Figura D.1. O motor de indução foi rebobinado para contemplar esse tipo de disposição.

Observa-se, pela Figura D.1, a existência de uma diferença angular espacial entre as espiras n_1 e n_2 . Considerando essa diferença angular igual à $2 \cdot \epsilon$, medidos em graus elétricos, pode-se determinar o diagrama fasorial destas tensões, representado na Figura D.2, letra (a). Estão representados, também, a soma (b) e a diferença (c) das tensões induzidas. A soma das tensões induzidas é a própria tensão da fase a . Entretanto, a diferença das tensões induzidas $(V_{n1} - V_{n2})_a$ está 90° defasada do eixo da fase a , estando sobre o eixo β do sistema de eixos coordenados $\alpha\beta$, fixos no estator.

O processo de subtração também elimina certos termos parasitas, que poderiam introduzir erros na leitura do fluxo. Em geral, a tensão induzida nas espiras pode ser dividida em três termos:

1. Tensão induzida pelo fluxo desejado;
2. Tensão do fluxo de dispersão;
3. Tensão na queda resistiva, considerando-se a resistência das bobinas.

Os termos 2 e 3 são dependentes apenas da corrente que flui pela própria espira, independentemente da corrente que flui nas demais espiras. Desde que a corrente que flui em n_1 é a mesma que flui em n_2 , a queda $R \cdot I$ é a mesma nas duas espiras, desta forma, efetuando a diferença $(V_{n1} - V_{n2})_a$, o termo da tensão devido a queda resistiva é anulado. O mesmo ocorre com a tensão devida ao fluxo de dispersão, considerando-se que a indutância de dispersão é a mesma para as duas bobinas. Desta forma, apenas a tensão devido ao fluxo mútuo, a qual desejamos medir, será responsável pelo resultado de $(V_{n1} - V_{n2})_a$.

A mesma análise pode ser realizada com as bobinas das fases b e c . A diferença das tensões parciais de cada fase será sempre defasada de 90^0 com relação aos eixos de cada fase, como mostra a Figura D.3.

Efetuando-se a diferença de $(V_{n1} - V_{n2})_b - (V_{n1} - V_{n2})_c$, a tensão resultante, em fase com o eixo da fase a , é, a menos de um fator de escala, a tensão sobre o eixo α . Desta forma obtém-se as tensões do eixo α e β e, a partir destas, calcula-se o fluxo em cada eixo.

As tensões V_{n1} e V_{n2} podem ser escritas como:

$$V_{n1} = R_1 \cdot I_{as} + \frac{d\lambda_{1s}}{dt} + \frac{d\lambda_{1r}}{dt} \quad (D.1)$$

$$V_{n2} = R_1 \cdot I_{as} + \frac{d\lambda_{2s}}{dt} + \frac{d\lambda_{2r}}{dt} \quad (D.2)$$

onde:

λ_{1s} e λ_{2s} representa o fluxo entre a espira n_1 e n_2 e as demais espiras do estator, respectivamente;

λ_{1r} e λ_{2r} representa o fluxo entre a espira n_1 e n_2 e as demais espiras do rotor, respectivamente.

Considerando espiras idênticas, a partir das Equações D.1 e D.2, tem-se:

$$(V_{n1} - V_{n2})_a = \left[\frac{d\lambda_{1s}}{dt} - \frac{d\lambda_{2s}}{dt} \right] + \left[\frac{d\lambda_{1r}}{dt} - \frac{d\lambda_{2r}}{dt} \right] \quad (D.3)$$

Observa-se que a diferença da tensão induzida é independente da resistência do estator.

O fluxo entre a espira n_1 e n_2 e as demais espiras do estator pode ser escrito, considerando-se fluxo de dispersão e o fluxo mútuo, como:

$$\lambda_{1s} = \lambda_{1ls} + \lambda_{1ms} \quad (D.4)$$

$$\lambda_{2s} = \lambda_{2ls} + \lambda_{2ms} \quad (D.5)$$

onde, λ_{1ls} e λ_{2ls} são os fluxos das espiras n_1 e n_2 que não cortam o rotor;

λ_{1ms} e λ_{2ms} são os fluxos mútuos entre a espira n_1 e n_2 e as demais espiras do estator, respectivamente.

Considerando o passo da bobina não é pleno, a dupla-camada resulta em ranhuras contendo espiras de fases diferentes. Neste caso, pode-se observar a existência de um acoplamento associado apenas com as componentes de enlace de fluxo. Sendo L_{IT} e L_{IB} a indutância associada com a espira colocada em cima e em baixo da ranhura e L_{ITB} a indutância mútua entre elas, pode-se escrever a parcela de fluxo que não corta o rotor como:

$$\lambda_{1ls} = (L_{IT} + L_{IB}) \cdot i_{as} + L_{ITB} \cdot i_{as} - i_{bs} \cdot L_{ITB} + L_{le} \cdot i_{as} \quad (D.6)$$

$$\lambda_{2ls} = (L_{IT} + L_{IB}) \cdot i_{as} + L_{ITB} \cdot i_{as} - i_{cs} \cdot L_{ITB} + L_{le} \cdot i_{as} \quad (D.7)$$

onde L_{le} representa a indutância devido ao fim de espira e demais componentes responsáveis pela indutância de dispersão.

Subtraído-se as Equações D.6 e D.7, tem-se:

$$\lambda_{1ls} - \lambda_{2ls} = L_{ITB} \cdot (I_{cs} - I_{bs}) \quad (D.8)$$

Considerando a corrente de estator de eixo β determinada por: $I_{\beta s} = \frac{1}{\sqrt{3}} \cdot (I_{cs} - I_{bs})$, obtém-se:

$$\lambda_{1ls} - \lambda_{2ls} = \sqrt{3} \cdot L_{ITB} \cdot I_{\beta s} \quad (D.9)$$

A parcela do fluxo descrita pela Equação D.9 é considerada parasita, sendo que seu total cancelamento seria decorrente da utilização do passo pleno no bobinamento do motor.

A componente fundamental do fluxo mútuo (λ_{1ms} e λ_{2ms}) pode ser descrito, considerando a simetria existente entre as bobinas n_1 e n_2 e o eixo da fase a , como:

$$\lambda_{1ms} = i_{as} \cdot L_{mls} \cdot \cos(-\varepsilon) + i_{bs} \cdot L_{mls} \cdot \cos(-120^\circ - \varepsilon) + i_{cs} \cdot L_{mls} \cdot \cos(120^\circ - \varepsilon)$$

$$\lambda_{2ms} = i_{as} \cdot L_{mls} \cdot \cos(\varepsilon) + i_{bs} \cdot L_{mls} \cdot \cos(\varepsilon - 120^\circ) + i_{cs} \cdot L_{mls} \cdot \cos(120^\circ + \varepsilon)$$

onde L_{mls} é a indutância mútua entre cada espira e os demais enrolamentos da fase a .

Subtraindo estas duas Equações tem-se:

$$\lambda_{1ms} - \lambda_{2ms} = \sqrt{3} \cdot L_{mls} \cdot \sin(\varepsilon) \cdot (i_{cs} - i_{bs}) \quad (D.10)$$

Considerando-se a corrente do eixo β pode-se reescrever a Equação acima como:

$$\lambda_{1ms} - \lambda_{2ms} = 3 \cdot L_{mls} \cdot \sin(\varepsilon) \cdot I_{\beta s} \quad (D.11)$$

Para a determinação do fluxo enlaçado pelas bobinas n_1 e n_2 , devido às correntes de rotor, é necessário realizar uma mudança de variáveis, referenciando todas as variáveis ao mesmo sistema de eixos $\alpha\beta$, fixos no estator. Para tal mudança será considerado o sistema de eixos mostrado na Figura D.4.

O fluxo que corta as espiras n_1 e n_2 , devido as correntes de rotor podem, então, ser descritas como:

$$\lambda_{1r} = L_{mlr} \cdot I_{dr}^r \cdot \cos(\theta_r + \varepsilon) + L_{mlr} \cdot I_{qr}^r \cdot \sin(\theta_r + \varepsilon)$$

$$\lambda_{2r} = L_{mlr} \cdot I_{dr}^r \cdot \cos(\theta_r - \varepsilon) + L_{mlr} \cdot I_{qr}^r \cdot \sin(\theta_r - \varepsilon)$$

Onde:

I_{dr}^r e I_{qr}^r são as correntes de rotor referidas ao sistema de eixos fixos no rotor (sistema de eixos dq);

L_{mlr} representa o máximo acoplamento entre uma das espiras e uma das correntes equivalentes do rotor.

Subtraindo-se estas Equações, obtém-se:

$$\lambda_{1r} - \lambda_{2r} = -2 \cdot L_{mlr} \cdot \sin(\varepsilon) \cdot [I_{dr}^r \cdot \sin(\theta_r) - I_{qr}^r \cdot \cos(\theta_r)] \quad (D.12)$$

Referenciando-se as correntes I_{dr}^r e I_{qr}^r do rotor para o sistema de eixos β fixo no estator através da Equação de transformação $I_r^\beta = I_{qr}^r \cdot \cos(\theta_r) - I_{dr}^r \cdot \sin(\theta_r)$, tem-se:

$$\lambda_{1r} - \lambda_{2r} = 2 \cdot L_{mlr} \cdot \sin(\varepsilon) \cdot I_r^\beta \quad (D.13)$$

Sabe-se, também, que $L_{mlr} = \frac{3}{2} \cdot L_{mls}$, portanto:

$$\lambda_{1r} - \lambda_{2r} = 3 \cdot L_{mls} \cdot \sin(\varepsilon) \cdot I_r^\beta \quad (D.14)$$

Substituindo-se as Equações D.14, D.11 e D.9 na Equação D.3, obtém-se:

$$(V_{n1} - V_{n2})_a = \frac{d}{dt} \left[\sqrt{3} \cdot L_{ITB} \cdot I_{\beta s} + 3 \cdot L_{mls} \cdot \sin(\varepsilon) \cdot (I_{\beta s} + I_r^\beta) \right] \quad (D.15)$$

Considerando o fluxo que enlaça o enrolamento do eixo β , descrito por $\lambda_{m\beta} = \frac{3}{2} \cdot L_{ms} (I_{\beta s} + I_r^\beta)$, pode-se reescrever a equação acima como:

$$(V_{n1} - V_{n2})_a = \frac{d}{dt} \left[\sqrt{3} \cdot L_{ITB} \cdot I_{\beta s} + 3 \cdot L_{mls} \cdot \sin(\varepsilon) \cdot \left(\frac{2}{3} \cdot \frac{\lambda_{m\beta}}{L_{ms}} \right) \right] \quad (D.16)$$

onde L_{ms} é a indutância de magnetização de uma fase do estator.

Como já foi mencionado, a primeira parcela ($\sqrt{3} \cdot L_{ITB} \cdot I_{\beta s}$) é caracterizada como um termo parasita, dependente do passo de bobina, sendo, portanto, desconsiderada na determinação final do fluxo. Poderia-se introduzir um termo de correção, uma vez que a corrente de estator é medida. Entretanto, devido à saturação da máquina, a qual afeta o termo L_{ITB} diferentemente de $L_{m\ell s}$, o procedimento de correção, em função do ponto de operação da máquina, se tornaria demasiadamente complexo. Desta forma, o fluxo mútuo de eixo β pode ser determinado como:

$$\lambda_{m\beta} = \frac{1}{2 \cdot \sin(\varepsilon)} \cdot \frac{L_{ms}}{L_{m\ell s}} \cdot \int (V_{n1} - V_{n2})_a \cdot dt \quad (D.17)$$

Considerando-se a fase a como a referência (0 graus), pode-se escrever as tensões $(V_{n1} - V_{n2})_b$ e $(V_{n1} - V_{n2})_c$ como:

$$(V_{n1} - V_{n2})_b = V_{m\acute{a}x} \cdot \cos(\omega \cdot t - 30^\circ)$$

$$(V_{n1} - V_{n2})_c = V_{m\acute{a}x} \cdot \cos(\omega \cdot t - 150^\circ)$$

Portanto,

$$(V_{n1} - V_{n2})_b - (V_{n1} - V_{n2})_c = \sqrt{3} V_{m\acute{a}x} \cdot \cos(\omega \cdot t) \quad (D.18)$$

Observa-se (Figura D.3) que a diferença $(V_{n1} - V_{n2})_b - (V_{n1} - V_{n2})_c$ está diretamente sobre o eixo α , podendo-se, portanto, determinar o fluxo sobre este eixo de maneira similar à utilizada para o eixo β .

Considerando-se a Lei de Gauss, pode-se escrever:

$$-(V_{n1} - V_{n2})_c = (V_{n1} - V_{n2})_a + (V_{n1} - V_{n2})_b, \text{ logo,}$$

$$(V_{n1} - V_{n2})_b - (V_{n1} - V_{n2})_c = (V_{n1} - V_{n2})_a + 2 \cdot (V_{n1} - V_{n2})_b.$$

Portanto,

$$\lambda_{m\alpha} = \frac{1}{2 \cdot \sqrt{3} \cdot \sin(\varepsilon)} \cdot \frac{L_{ms}}{L_{m\ell s}} \cdot \int [(V_{n1} - V_{n2})_a + 2 \cdot (V_{n1} - V_{n2})_b] \cdot dt \quad (D.19)$$

O torque eletromagnético pode ser expresso em função dos fluxos nos eixos α e β e pelas correntes de estator, como mostrado:

$$Te = \frac{3}{2} \cdot \frac{p}{2} \cdot (\lambda_{m\alpha} \cdot i_{s\beta} - \lambda_{m\beta} \cdot i_{s\alpha}) \quad (\text{D.20})$$

onde p é o número de pares de polos.

Desta forma, obtém-se o fluxo em eixo α (Equação D.19), o fluxo em eixo β (Equação D.17) e o torque eletromagnético (Equação D.20), sem a necessidade de incluir bobinas exploradoras, utilizando apenas as tensões parciais de cada bobina de duas fases.

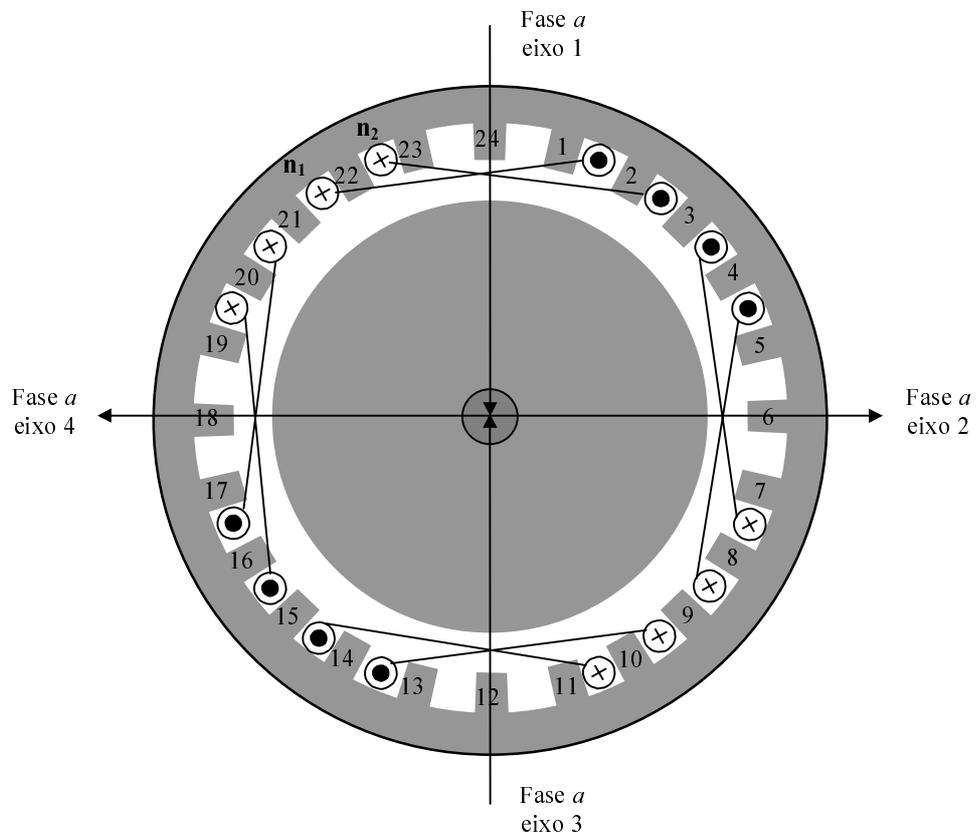


Figura D.1. Distribuição espacial dos enrolamentos de uma fase do estator.

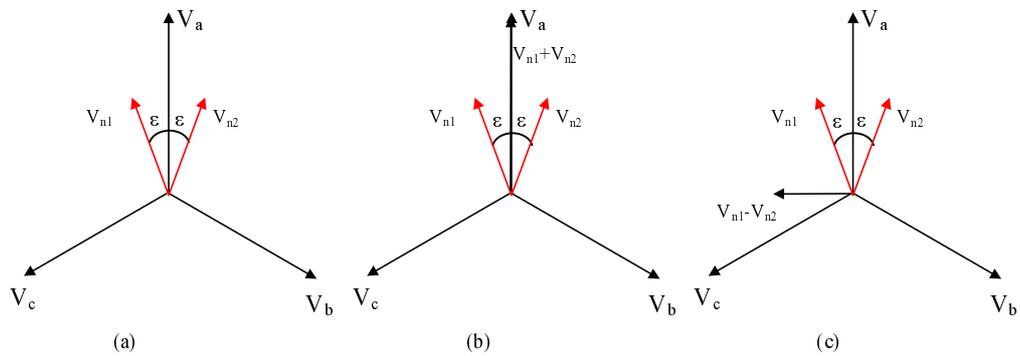


Figura D.2. Diagrama fasorial das tensões induzidas nas bobinas n_1 e n_2 .

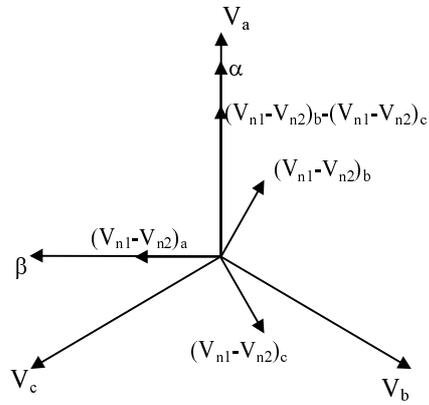


Figura D.3. Diagrama fasorial para a diferença das tensões parciais nas bobinas de cada uma das fases.

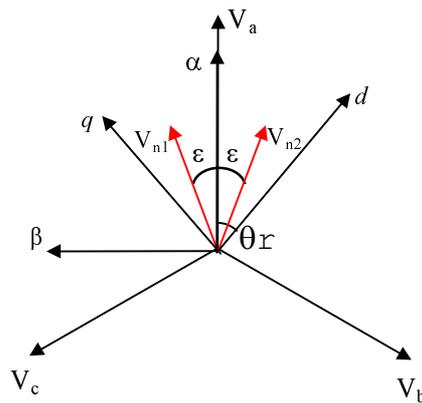


Figura D.4. Sistema de eixos utilizado para a mudança de variáveis.

Apêndice E

SemiDriver SKHI23

Para a construção do inversor trifásico, optou-se por adquirir os drivers de comando dos IGBTs da Semikron, modelo SKHI23. Devido à dificuldade de encontrar as especificações técnicas do fabricante, as principais informações a respeito dos drivers de comando serão descritas.

E.1 Características Técnicas

Cada placa de driver é responsável por um braço do inversor (dois IGBTs), sendo responsável pela inclusão do tempo morto e circuitos de proteção (sobrecorrente no IGBT e queda da tensão de alimentação dos drivers).

A Tabela E.1 mostra os valores típicos do driver de comando. A Figura E.1 mostra a disposição dos jumpers e conectores existentes no semidriver de comando. A Tabela E.2 mostra a pinagem do conector de entrada, modelo DIN 41651, de 14 pinos.

As conexões do driver de comando para os IGBTs podem ser feitas usando apenas um conector MOLEX de 12 pinos ou usando dois conectores com 5 pinos, modelo 41791, para cada IGBT. Na Tabela E.3 está mostrada a pinagem referente a um conector molex com 5 pinos.

A capacidade de saída de alta potência foi desenvolvida para chavear altas correntes em módulos simples ou duplos de IGBT's (ou IGBT's paralelos). Os buffers de saída foram projetados para chavear módulos de correntes de até 200A e frequências tão altas quanto

Tabela E.1: Valores típicos do driver de comando

Símbolo	Significado	Valor
V_s	Tensão de alimentação primária	+15, 0V (típico)
V_{it+}	Limite de tensão de entrada HIGHmin	+12, 5V (para nível +15V) +2, 4V (para nível +5V)
V_{it-}	limite de tensão de entrada LOWmáx	+3, 6V (para nível +15V) +0, 5V (para nível +5V)
V_{gon}	Tensão de disparo do <i>gate</i>	+15V
V_{goff}	Tensão de corte do <i>gate</i>	-8V
R_{in}	Resistência de entrada	10 Ω
$td_{(on)io}$	Tempo de propagação entrada -saída - disparo	1, 4s
$td_{(off)io}$	Tempo de propagação entrada -saída - corte	1, 4s
ttd	Tempo morto	10s (ajuste de fábrica)
R_{on}	Resistor de <i>gate</i> interno para disparo	22 Ω
R_{off}	Resistor de <i>gate</i> interno para corte	22 Ω

20kHz.

Este driver possui uma nova função adicionada ao circuito de proteção de curto-circuito (soft turn off). Ela aumenta automaticamente o tempo de corte do IGBT reduzindo os picos de sobre-tensão no lado DC.

Conversores DC/DC com isolamento galvânica garantem a proteção contra a alta tensão no secundário.

A alimentação dos drivers podem ser a mesma da placa de controle (V_s) sem necessitar isolamento. Toda informação transmitida entre primário e secundário é feita utilizando transformador de ferrite.

Os estágios de entrada do driver são conectados diretamente da saída da placa de controle, podendo ser usados dois níveis diferentes de tensão de entrada (+15V ou +5V), selecionados por J1. Para cabos longos de entrada, não é aconselhável utilizar nível de +5V.

O circuito de interlock impede que dois IGBT's de um braço permaneçam ligados, introduzindo um tempo morto que pode ser ajustado através de J3/K3 e J4/K4. O circuito de interlock pode ser desabilitado através de J5/K5.

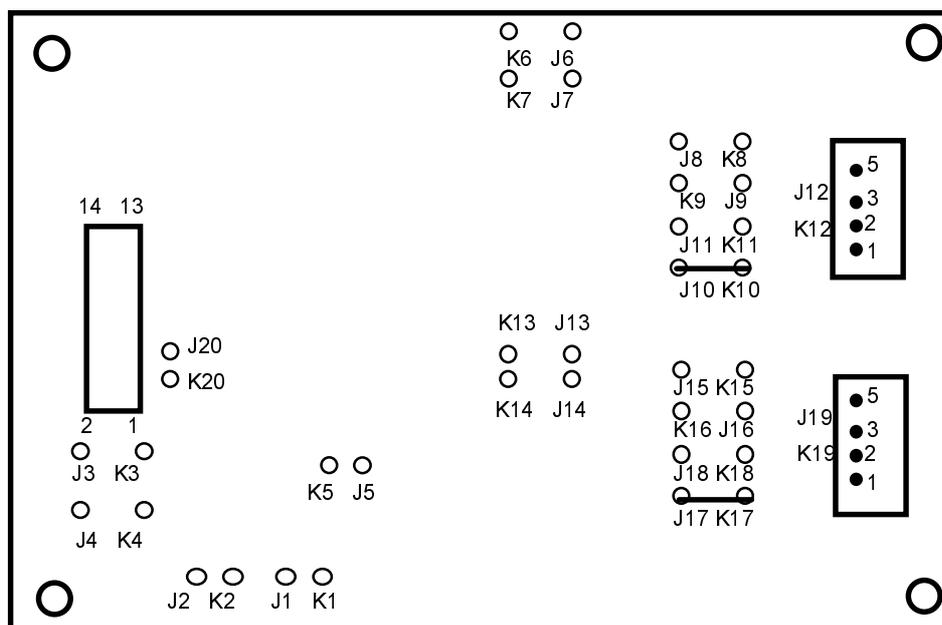


Figura E.1. Disposição dos jumpers e conectores do driver de comando SKHI23.

E.1.1 Descrição do diagrama de blocos

As principais características do driver de comando serão analisadas, utilizando, como referência, o diagrama de blocos (Figura E.2) do circuito.

1 - Nível do circuito de entrada

O nível de entrada pode ser ajustado para trabalhar com o sinal de controle em $+15V$ ou $+5V$, selecionado por J1/K1.

Utiliza-se o nível de $+15V$ para minimizar ruídos ou quando o cabo utilizado para a conexão entre a placa de controle e o driver ultrapassar $50cm$ (máximo de 1 metro). Neste caso utilizar cabo blindado. Conectar a blindagem ao pino 1 (conector de entrada) e acoplar a $0V$ através de um capacitor, resistor ou jumper (J20/K20).

Como a impedância do seletor de nível de entrada é muito alta, um resistor interno de pull-down mantém o IGBT cortado no caso da conexão de V_{in} ser interrompida ou ter sido deixada não conectada.

Tabela E.2: Conector de entrada - 14 pinos, modelo DIN 41651

Pino	Descrição
1	Blindagem do cabo de conexão entre a placa de controle e o driver
2	Sinal de entrada do driver inferior ($V_{in\ BOT}$)
3	Sinal de erro do circuito de "error memory"
4	Sinal de entrada do driver superior ($V_{in\ TOP}$)
5	
6	
7	
8	Tensão de alimentação $+V_s$
9	Tensão de alimentação $+V_s$
10	GND
11	GND
12	
13	
14	

A Tabela E.6 mostra as tensões de disparo (V_{it+}) e corte (V_{it-}) do IGBT, compreendidas pelo driver de comando, para as tensões de alimentação de $+15V$ e $+5V$:

2 - Buffer de entrada

Este circuito habilita e possibilita que o sinal de entrada V_{in} seja transmitido para o transformador de pulso e impede que sinais espúrios sejam transmitidos para o secundário.

3 - Error memory e sinal de reset

No caso de curto-circuito, o circuito de monitoramento de V_{ce} manda um sinal através do transformador de pulso para o primário fornecendo a informação para um transistor de coletor-aberto (pino X1,3), o qual pode ser conectado ao circuito de controle como uma mensagem de erro em nível ALTO. Pode-se colocar o sinal de erro em nível baixo fechando-se J2/K2.

Se a tensão de alimentação cair abaixo de $13V$ por mais de $0,5ms$, o sinal de erro também será ativado.

Para nível alto (default), deve-se utilizar um resistor externo R_C , conectado, preferencialmente, na placa de controle.

Se for utilizado nível baixo (J2/K2 fechado) um resistor de pull-up (conectado internamente

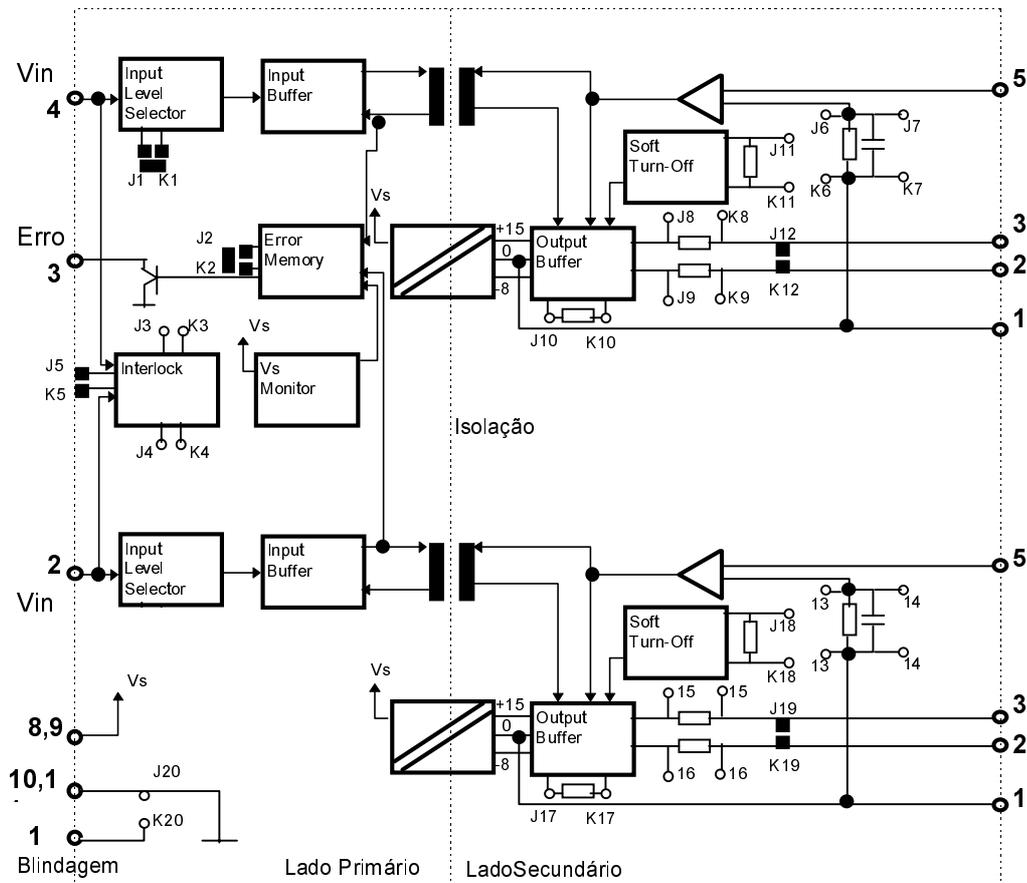


Figura E.2. Diagrama esquemático do semidriver de comando.

a V_s) é fornecido e os sinais de erro de mais de um driver podem ser conectados formando uma função wire-AND.

O sinal de erro pode ser desabilitado (Sinal de Reset) colocando os dois sinais de entrada em zero ($V_{in-top} = V_{in-bot} = 0$) por mais de $5ms$, ou desligando a tensão de alimentação do driver.

A Tabela E.7 mostra a relação entre o sinal de erro e os sinais de comando de disparo do IGBT.

Para sinal de erro em nível alto, o transistor de coletor-aberto deve ser conectado a um resistor de pull-up a uma fonte externa, limitando a corrente em $6mA$ ($I_{sink} < 6mA$).

4 - Transformador de Pulso

Tabela E.3: Conector de saída dos sinais de comando para o IGBT, modelo MOLEX 41791.

Pino	Descrição
1	Emissor do IGBT
2	Saída do resistor R_{goff}
3	Saída do resistor R_{gon}
5	Coletor do IGBT (monitoramento de V_{ce})

Responsável pela transmissão dos sinais de disparo e corte para o secundário. Em sentido inverso, transmite o sinal de erro do circuito de monitoramento de V_{ce} . Isolação de 4KVAC.

5 - Conversor DC/DC

No primário, um inversor em meia-ponte transfere a energia necessária de V_s para o secundário, através de um transformador de ferrite. No secundário um conversor de ponte completa fornece os níveis DC (+15V e -8V) que são estabilizados por um circuito regulador.

6 - Buffer de saída

Alimentado com +15V e -8V pelo conversor DC/DC, amplifica o sinal de controle recebido pelo transformador de pulso. Se a operação ocorre normalmente (sem erro), o sinal é transmitido para o *gate* do IGBT através de R_{gon} e R_{goff} .

O circuito de saída possui dois MOSFET os quais são capazes de fornecer uma corrente de pico de 8A para o *gate*, melhorando o tempo de disparo/corte do IGBT.

Adicionalmente, pode-se selecionar $I_{r_{goff}}$ para descarregar a capacitância de *gate* através de uma fonte de tensão (padrão) ou por uma fonte de corrente, especialmente projetada para as séries de IGBT's de 1700V (melhorando o tempo de corte do IGBT).

7 - Soft turn-off

No caso de curto-circuito, este circuito aumenta a resistência em série com R_{goff} , aumentando o tempo de corte do IGBT. Este aumento reduz o pico de tensão no lado DC, reduzindo di/dt sobre o link DC.

Tabela E.4: Jumpers utilizados na placa do driver de comando.

Jumper	Descrição
J1/K1	Seleção do nível de entrada do sinal de controle Aberto: +15V Fechado: +5V
J2/K2	Seleção do sinal de erro do circuito de "error memory": Aberto: erro em nível alto; Fechado: erro em nível baixo
J3/K3 (RTD1 superior) J4/K4 (RTD2 inferior)	Ajuste de Tempo Morto (ver tabela)
J5/K5	Interconexão entre driver superior e inferior (tempo morto): Aberto: com tempo morto; Fechado: sem tempo morto
J6/K6	R_{ceTOP} (ver Tabela E.5) - $18K\Omega$ (SKHI 23/12) $36K\Omega$ (SKHI 23/17)
J7/K7	C_{ceTOP} (ver Tabela E.5) - $330pF$ (SKHI 23/12) $470pF$ (SKHI 23/17)
J8/K8	R_{gonTOP} (ver Tabela E.5) - 22Ω
J9/K9	$R_{goffTOP}$ (ver Tabela E.5) - 22Ω
J10/K10	$I_{rgoff-TOP}$ (ver Tabela E.5) - equipado com 0Ω
J11/K11	$R_{goff-SC-TOP}$ (ver Tabela E.5)- equipado com 22Ω
J12/K12	Drive superior - 1 IGBT ou IGBT's paralelos: Aberto: 2 cabos para gate; Fechado: 1 cabo para gate
J13/K13	R_{ceBOT} (ver Tabela E.5) - $18K\Omega$ (SKHI 23/12) $36K\Omega$ (SKHI 23/17)
J14/K14	C_{ceBOT} (ver Tabela E.5) - $330pF$ (SKHI 23/12) $470pF$ (SKHI 23/17)
J15/K15	R_{gonBOT} (ver Tabela E.5) - 22Ω
J16/K16	$R_{goffBOT}$ (ver Tabela E.5) - 22Ω
J17/K17	$I_{rgoff-BOT}$ (ver Tabela E.5) - equipado com 0Ω
J18/K18	$R_{goff-SC-BOT}$ (ver Tabela E.5)- equipado com 22Ω
J19/K19	Drive Inferior - 1 IGBT ou IGBT's paralelos: Aberto: 2 cabos para gate; Fechado: 1 cabo para gate
J20/K20	Blindagem: Aberto: não ligado; Fechado: Ligado à GND

Este tempo pode ser reduzido conectando resistores em paralelo com o resistor $R_{goff-SC}$, que já existe na placa do driver.

8 - Monitoramento de V_{ce}

Este circuito é responsável pela sensibilidade ao curto-circuito. Através da medição direta do V_{ce} do IGBT, o circuito bloqueia a saída do buffer (pelo circuito de soft turn-off), no caso de curto-circuito e envia um sinal para o circuito de *error memory*.

O reconhecimento do nível de V_{ce} de curto-circuito é função de R_{ce} e C_{ce} e depende do IGBT utilizado. Os valores típicos utilizados são mostrados na curva 2, Figura E.3.

O valor de $V_{ce_{ref}}$ não é estático, mas uma referência dinâmica que decai exponencialmente, começando em, aproximadamente, $15V$ e decresce até $V_{ce_{stat}}$ ($5V < V_{ce_{stat}} < 10V_m$, determinado por R_{ce}) com uma constante de tempo t ($0,5ms < t < 1ms$, controlado por

Tabela E.5: Componentes de ajuste utilizados na placa do driver de comando.

Resistor	Descrição
R_{gon}	Resistor de disparo (<i>gate</i>) do IGBT
R_{goff}	Resistor de corte (<i>gate</i>) do IGBT
$R_{goff-SC}$	Resistor para reduzir o tempo do “soft turn-off”
Ir_{goff}	Resistor de corte (<i>gate</i>) através de fonte de corrente
R_{CE}	Resistor de ajuste do nível de V_{ce} de curto-circuito
C_{CE}	Capacitor de ajuste do tempo de decaimento do V_{ce} de curto-circuito
R_{TD}	Resistor de ajuste do tempo morto

C_{ce}).

O valor de V_{cestat} deve ser ajustado para permanecer acima do nível de V_{cesat} (V_{ce} de saturação) em operação normal .

Para evitar uma indicação falsa de erro quando o IGBT começa a conduzir, deve-se utilizar um tempo de decaimento para $V_{ce_{ref}}$. Como o valor de V_{ce} é limitado, internamente, em 10V, o tempo de decaimento de $V_{ce_{ref}}$ deve alcançar este nível após V_{ce} ou uma falsa indicação de curto-circuito ocorrerá (curva 1, Figura E.3).

O tempo no qual o IGBT alcança os 10V (representado por um “□” na Figura E.3) depende do próprio IGBT e do valor de R_{gon} utilizado.

Os valores de R_{ce} e C_{ce} podem ser encontrados da Figura. 7 (ver original) , utilizando V_{cestat} e T_{min} como valores de entrada e com as seguintes limitações:

- $R_{ce} > 10K\Omega$
- $C_{ce} < 2,7\eta F$

9 - R_{gon} e R_{goff}

Estes dois resistores são responsáveis pela velocidade de chaveamento de cada IGBT.

O driver possui dois resistores de R_g (22 Ω) na placa. Estes valores podem ser reduzidos, para o uso em módulos maiores ou frequências mais altas, colocando resistores em paralelo com os existentes.

Tabela E.6: Valores típicos para o disparo ou corte do IGBT.

V_{it+}	V_s	min.	tip.	máx
	15V	9.5V	11.0V	12.5V
	5V	1.8V	2.0V	2.4V
V_{it-}	V_s	min.	tip.	máx
	15V	3.6V	4.2V	4.8V
	5V	1.8V	0.65V	0.8V

Tabela E.7: Tabela Verdade para o sinal de erro em nível alto.

Falta	Sinal de Reset	Sinal de Erro	Disparo do IGBT
não	não	0	habilitado
não	ativo	0	desabilitado
sim	não	1	desabilitado
sim	ativo	0	desabilitado

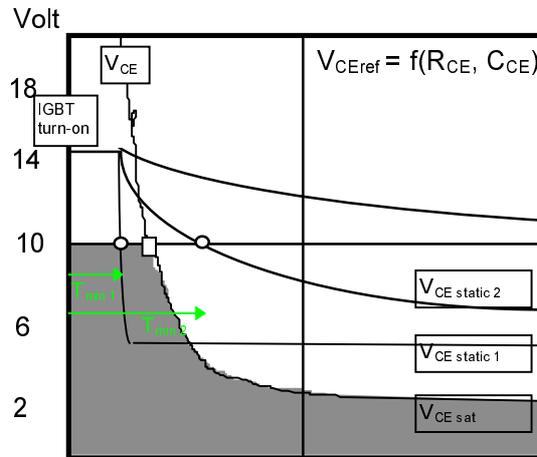


Figura E.3. Forma de onda de V_{CE} e, função dos parâmetros R_{CE} e C_{CE} .

As saídas G_{on} e G_{off} são feitas para a conexão de mais de um IGBT por driver (IGBT's paralelos). Neste caso necessita-se dos sinais de ON e OFF separadamente para a conexão de resistores externos de R_{gon} e R_{goff} independentes para cada IGBT.

Se apenas um IGBT é utilizado, sugere-se que as duas saídas (G_{on} e G_{off}) sejam conectadas através de J12/K12 e J9/K9. Sugere-se também que sejam utilizados dois resistores para R_{gon} e dois para R_{goff} quando utiliza-se valores baixos de resistência pois o pico de corrente (8A) pode danificar um único resistor.

10 - Interlock

Este circuito impede que o IGBT dispare antes da descarga completa do $gate$ do outro IGBT. Pode-se ajustar o tempo morto através de R_{TD} (J3/K3 e J4/K4).

Pode-se desabilitar este circuito através de J5/K5. Deste modo, tem-se dois drivers independentes.

A Tabela E.8 mostra a variação do tempo morto (t_{TD}) em função dos resistores R_{TD} .

E.2 Procedimento de operação

Tabela E.8: Variação do tempo morto em função do resistor R_{TD}

$R_{TD1} = R_{TD2}$	Tempo Morto (t_{TD})
10K Ω	0.9 μ s
22K Ω	1.8 μ s
33K Ω	2.5 μ s
47K Ω	3.2 μ s
68K Ω	4.0 μ s
100K Ω	5.0 μ s
930K Ω	7.7 μ s
não conectado	10 μ s

Tabela E.9: Valores característicos dos componentes para IGBT 1200V

SK -IGBT Módulo	$R_{gon} (\Omega)$	$R_{goff} (\Omega)$	$C_{CE} (pF)$	$R_{CE} (K\Omega)$	$I_{r_{goff}} (\Omega)$
SKM 75GB123D	22	22	330	18	0
SKM 100GB123D	15	15	990	18	0
SKM 145GB123D	12	12	330	18	0
SKM 150GB123D	12	12	330	18	0
SKM 200GB123D	10	10	330	18	0
SKM 300GB123D	8,2	8,2	330	18	0

E.2.1 Um IGBT por drive

Para um correto chaveamento e monitoramento de curto-circuito do crive, alguns componentes adicionais podem ser utilizados. Estes componentes são funções do IGBT utilizado e estão mostrados nas Tabelas E.9 e E.10.

O ajuste de Rgoff-SC deve ser feito observando a sobretensão no módulo, em caso de curto-circuito. Se o link DC for indutivo o módulo pode ser chaveado mais rapidamente.

E.2.2 IGBT's paralelos

A conexão em paralelo de IGBT's é recomendada apenas quando for utilizado IGBT's com estrutura homogênea (IGHT), desta forma eles terão o mesmo coeficiente de temperatura, resultando numa perfeita divisão de corrente sem o auxílio de elementos externos.

Deve-se ter cuidado para otimizar o circuito e conseguir a melhor performance do IGBT.

Os IGBT's devem ter resistores independentes de Rgon e Rgoff e um resistor auxiliar de emissor R_E , assim como um resistor de coletor auxiliar R_C . Os resistores auxiliares R_{gonx} , R_{goffx} , R_{ex} e R_{Cx} devem ser montados em um circuito adicional, próximo aos módulos paralelos e R_{gon} , R_{goff} devem ser mudados para zero ohms.

R_{ex} tem o valor de $0,5\Omega$ e sua função é de evitar que a corrente de carga circule pelo emissor auxiliar, o que poderia fazer com que a tensão de emissor-terra ficasse desbalanceada.

R_{Cx} assume o valor de 47Ω e sua função é de dividir a tensão de V_{cesat} no caso de curto-

Tabela E.10: Valores característicos dos componentes para IGBT 1700V

SK -IGBT Módulo	$R_{gon} (\Omega)$	$R_{goff} (\Omega)$	$C_{CE} (pF)$	$R_{CE} (K\Omega)$	$Ir_{goff} (\Omega)$
SKM 75GB173D	15	15	470	36	0
SKM 100GB173D	12	12	470	36	0
SKM 150GB173D	10	10	470	36	0
SKM 200GB173D	8,2	8,2	470	36	0
SKM 300GB173D	6,8	6,8	470	36	0

circuito, para o monitoramento de V_{ce} .