

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Curso de Pós-graduação em Engenharia Elétrica

*REDES NEURAIS ARTIFICIAIS PARA
ESTIMAÇÃO DE VELOCIDADE EM
MÁQUINAS DE INDUÇÃO*

Lucilene Ferreira Mouzinho

São Luís - março/2003

CDD 999.99

CDU 998.98

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Curso de Pós-graduação em Engenharia Elétrica

*REDES NEURAIS ARTIFICIAIS PARA
ESTIMAÇÃO DE VELOCIDADE EM
MÁQUINAS DE INDUÇÃO*

Lucilene Ferreira Mouzinho

Dissertação apresentada ao Programa de
Pós-graduação em Engenharia de Eletricidade da UFMA,
como parte dos requisitos necessários para obtenção do
grau de mestre em Engenharia Elétrica.

São Luís - março/2003

Mouzinho, Lucilene Ferreira

REDES NEURAS ARTIFICIAIS PARA ESTIMAÇÃO DE VELOCIDADE EM MÁQUINAS DE INDUÇÃO / Lucilene Ferreira Mouzinho - março/2003

99.p

1.Máquina Elétrica de Corrente Alternada 2. Estimação 3.Sistema Adaptativo 4. Modelo de Rede Neural Artificial 5. Estimação de velocidade. I.Título.

CDD 999.99

CDU 998.98

**REDES NEURAIS ARTIFICIAIS PARA
ESTIMAÇÃO DE VELOCIDADE EM
MÁQUINAS DE INDUÇÃO**

Lucilene Ferreira Mouzinho

Aprovado em 12/03/2003

BANCA EXAMINADORA

Prof^o. João Viana da Fonseca Neto

Dr. em Engenharia Elétrica

Prof^o. Sebastian Yuri Cavalcanti Catunda

Dr. em Engenharia Elétrica

Prof^o. Raimundo Nazareno Cunha Alves

Dr. em Engenharia Elétrica

O supremo grau da sabedoria é ter sonhos suficientemente grandes para não perdê-los de vista enquanto os perseguimos. William Faulkner

Dedicatória

Aos meus pais, Elizia e Raimundo.

Ao meu marido, Lucius.

Aos meus filhos, Sâmia e Vinicius.

Aos meus irmãos e sobrinhos.

Aos meus alunos do CEFET-MA.

Agradecimentos

Ao Mestre dos mestres, pela presença constante em minha vida.

A minha mãe, Elizia, sem a qual alguém e algo não existiriam.

Ao "mestre", Prof^o. Dr. João Viana da Fonseca Neto, pela sublime orientação, amizade, contribuição, paciência e compreensão durante todo este trabalho.

Ao Prof^o. Dr. Sebastian Yuri Catunda por disponibilizar o uso do laboratório LIEA, pelo apoio e contribuição nas referências bibliográficas.

À Prof^a. Dr^a. Maria da Guia, pelo apoio e incentivo.

Aos professores e amigos do CEFET-MA, João Eyder, Reginaldo Miranda, Marcelino Pacelli, Eliúde Trovão, Paulo César Doval, Carlos César e Reginaldo Cortelleti pelo apoio, incentivo e sugestões.

Aos amigos do mestrado José Alano, José Pinheiro e Valeska pelo apoio e incentivo.

Ao Wagner Élvio, pelo apoio técnico.

A todos que colaboraram de maneiras diversas para a realização deste trabalho.

RESUMO

Nesta dissertação apresenta-se um modelo de uma rede neural artificial para estimar ou medir indiretamente a velocidade em máquinas de indução trifásicas para fins de controle. Realiza-se uma análise comparativa do estimador neural com os seguintes tipos de estimadores de velocidade fundamentados: fluxo do rotor, força eletromotriz e modelo adaptativo de referência. Os algoritmos dos estimadores são construídos e acoplados ao modelo dq0 da máquina de indução. A partir de resultados de simulações computacionais, obtidos dos modelos máquina-estimadores, realiza-se a análise de desempenho dos estimadores que tem como objetivo verificar quais dos estimadores atingiu em menor tempo uma velocidade de referência. Este trabalho também apresenta um estudo sobre a aplicação de redes neurais em máquinas de indução, abordando os seguintes tópicos: controle, falhas, supervisão, diagnóstico, identificação e estimação.

RESUMO

This work presents an artificial neural network model to estimate or indirectly measure the speed of three-phase induction machines for control purpose. A comparative analysis of the neural estimator is performed with the following types of speed estimators: rotor flux, electromotive force and model reference adaptive. The algorithms of the estimators are assembled and connected to the dq0 model of the induction machine. Computational simulation results, obtained from machine-estimators model, are used to carry out a comparative analysis of the speed estimators performance. This work also presents a survey on neural network applications in induction machines, covering the following issues: control, failure, supervision, diagnosis, identification and estimation.

Conteúdo

1	Introdução	10
1.1	Medição Indireta	11
1.2	Objetivos	12
1.3	Motivação	12
1.4	Organização da Dissertação	13
2	Redes Neurais Artificiais em Máquinas de Indução	14
2.1	Introdução	14
2.2	O Problema da Identificação de Sistemas	15
2.2.1	Processo para Identificação	15
2.3	Conceitos de Redes Neurais Artificiais	17
2.4	Formas de Identificação Neural	20
2.4.1	Modelagem Direta	20
2.4.2	Modelagem Inversa	22
2.5	Aplicações de RNA's para MI	24
2.5.1	Controladores via RNA	24
2.5.2	Falhas, supervisão e diagnóstico via RNA	25
2.5.3	Identificação e Estimação via RNA	27
3	Técnicas para Estimar a Velocidade da MI	29
3.1	Introdução	29
3.2	Modelo da Máquina de Indução	30

3.3	Estimação via Fluxo do Rotor	31
3.4	Estimação via FEM	35
3.5	Estimação baseada na técnica MRAS	36
3.5.1	Estimação via Fluxo do Rotor baseada na técnica MRAS	37
4	Estimação via Redes Neurais Artificiais	41
4.1	Introdução	41
4.2	Estimação via RNA's	42
5	Análise Comparativa entre Estimadores-ω_r	50
5.1	Introdução	50
5.2	Parâmetros da Máquina	51
5.3	Estimador via Fluxo	51
5.4	Estimador via FEM	52
5.5	Estimador via Fluxo utilizando MRAS	53
5.6	Estimador via RNA	55
5.7	Comentários	59
6	Conclusão	73
6.1	Perspectivas Futuras	74
A	Modelo da Máquina de Indução	76
A.1	Representação no Modelo Discreto	77
B	Estruturas dos Blocos para MATLAB/SIMULINK	80
B.1	Bloco Alimentação	80
B.2	Bloco Motor	82
	Referências Bibliográficas	86

Lista de Tabelas

5.1	Varição dos parâmetros α e η para estimar a velocidade via RNA's	57
5.2	Resultados da velocidade estimada via RNA para variações de α e η - valor de pico (V_p), tempo de estabilização (t_e) e Erro.	58
5.3	Resultados da velocidade estimada via fluxo e FEM - valor de pico (V_p), tempo de estabilização (t_e) e Erro.	62

Lista de Figuras

2.1	Passos para Identificação de Processos	16
2.2	Diagrama de unidade RNA de McCullock - Pitts	18
2.3	Diagrama para treinamento do Perceptron	19
2.4	Diagrama em Blocos do Modelo Direto para Estimação via RNA's	21
2.5	Diagrama em Blocos para Modelagem Neural Inversa	23
3.1	Modelo para estimação via fluxo do Rotor	34
3.2	Modelo para estimação via FEM do Rotor	37
3.3	Modelo para Estimação baseada em MRAS	38
3.4	Modelo para estimação via fluxo do rotor baseada em MRAS . . .	40
4.1	Estrutura do Sistema para Estimar ω_r	43
4.2	RNA de Duas Camadas	45
5.1	Velocidade estimada via fluxo do Rotor	52
5.2	Velocidade estimada via FEM do Rotor	54
5.3	Velocidade estimada via fluxo utilizando MRAS	56
5.4	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,5$ e $\eta=0,5$	59
5.5	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,5$ e $\eta=0,5$	60
5.6	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,9$	61
5.7	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,9$	62
5.8	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,5$	63
5.9	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,5$	64

5.10	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,2$	65
5.11	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,2$	66
5.12	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,00$	67
5.13	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,0$	68
5.14	Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,99$ e $\eta=0,01$	69
5.15	Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,99$ e $\eta=0,01$	70
5.16	Velocidade estimada com menor erro de regime via RNA com $t =$ 100 ms para $\alpha = 0,0$ e $\eta = 0,1$	71
5.17	Velocidade estimada com menor erro de regime via RNA com $t = 20$ ms para $\alpha = 0,0$ e $\eta = 0,1$	72
B.1	Descrição dos Blocos da Simulação da Máquina de Indução	81
B.2	Algoritmo para o cálculo do torque e da velocidade do rotor . . .	85

Lista de Abreviaturas e Siglas

CA	Corrente Alternada
AD	Analógico Digital
AG	Algoritmo Genético
DSP	Processamento Digital de Sinais (Digital Signal Processors)
DTC	Controle de Torque Direto (Direct Torque Control)
EPROM	Erasable Programmable Read Only Memory
FEM	Força Eletromotriz
FOC	Controle por Campo Orientado (Field Oriented Control)
FPGA	Field Programmable Gate Array
IA	Inteligência Artificial
MA	Máquina Assíncrona
MI	Máquina de Indução
MLP	Perceptron Multicamada (Multi Layer Perceptron)
MRAS	Sistema Adaptativo Modelo de Referência (Model Reference Adaptive System)
PC	Personal Computer
PI	Proportional Integral
PID	Proportional Integral Derivativo
PWM	Modulação por Lagura de Pulso (Pulse Width Modulation)
RNA	Rede Neural Artificial
SOFM	Mapeamento alto-organizado (Self-Organizing Feature Map)

Lista de Símbolos

- Subíndices
 - s variáveis e parâmetros do estator
 - r variáveis e parâmetros do rotor
 - q variáveis com referência de eixo em quadratura
 - d variáveis com referência de eixo direto
- Superíndice
 - \wedge variáveis e parâmetros estimados
- Operadores
 - \circ produto interno de vetores
 - \otimes produto vetorial
 - $\frac{\partial}{\partial t}$ derivada parcial
 - p derivativo
 - $\frac{1}{p}$ integral
- Variáveis
 - e_r força eletromotriz de rotor
 - e_{qr} força eletromotriz no rotor em quadratura
 - e_{dr} força eletromotriz no rotor em eixo direto
 - i_s corrente de estator
 - i_r corrente de rotor
 - i_{qs} corrente de estator em quadratura
 - i_{ds} corrente de estator em eixo direto
 - i_{qr} corrente de rotor em quadratura
 - i_{dr} corrente de rotor em eixo direto
 - i_m corrente de magnetização
 - v_s tensão de estator
 - v_r tensão de rotor

v_{qs}	tensão de estator em quadratura
v_{ds}	tensão de estator em eixo direto
λ_s	fluxo de estator
λ_r	fluxo de rotor
λ_{qs}	fluxo de estator em quadratura
λ_{ds}	fluxo de estator em eixo direto
λ_{1qr}	fluxo de rotor em quadratura
λ_{1dr}	fluxo de rotor em eixo direto
$\dot{\lambda}_{1qr}$	derivada do fluxo de rotor em quadratura
$\dot{\lambda}_{1dr}$	derivada do fluxo de rotor em eixo direto
λ_{2qr}	fluxo de rotor em quadratura
λ_{2dr}	fluxo de rotor em eixo direto
$\dot{\lambda}_{2qr}$	derivada do fluxo de rotor em quadratura
$\dot{\lambda}_{2dr}$	derivada do fluxo de rotor em eixo direto
λ_{mq}	fluxo mútuo em quadratura
λ_{md}	fluxo mútuo em eixo direto
ω_s	velocidade angular síncrona
ω_r	velocidade angular do rotor
$\hat{\omega}_r$	velocidade angular estimada do rotor
ω_{ls}	velocidade angular de escorregamento
x_{ls}	reatância de dispersão no estator
x_{lr}	reatância de dispersão no rotor
x_m	reatância de magnetização
x_M	reatância equivalente
T_m	conjugado eletromagnético
θ	posição espacial do vetor fluxo do rotor
T	período de amostragem
T_r	constante de tempo do rotor
ε	erro

J_m momento de inércia

W matriz de parâmetros ajustáveis ou pesos da RNA

- Parâmetros da Máquina de Indução

R_s resistência de estator por fase

R_r resistência de rotor por fase (referida ao estator)

L_{ls} indutância de dispersão do estator

L_{lr} indutância de dispersão do rotor

L_m indutância mútua de magnetização (estator/rotor)

L_s indutância própria do estator

L_r indutância própria do rotor (referida ao estator)

L_{sm} indutância mútua no enrolamento do estator

L_{rm} indutância mútua no enrolamento do rotor

P número de pólos

σ coeficiente de dispersão

- Parâmetros da RNA

α constante de momento da RNA

η taxa de aprendizagem da RNA

CAPÍTULO 1

Introdução

Informação sobre a velocidade é necessária para obter um alto desempenho do motor de indução controlado vetorialmente, (Telford *et al.*, 2000), (Idris e Yattim, 2000). Esta informação geralmente é fornecida por sensores, tal qual um tacômetro. Estes sensores contrapõem-se à robustez, em seus aspectos construtivos, e à simplicidade do motor de indução. Deste ponto de vista e para propósitos gerais a eliminação do sensor tende a reduzir significativamente custos e um *drive* de velocidade sem sensor é preferível, (Ben-Brahim, 1995), (Jacobina *et al.*, 2000), (Aller *et al.*, 2000), (Dunnigan *et al.*, 1998), (Ke *et al.*, 1997), (Llas *et al.*, 1996), (Kim e Hyun, 1994), (Tajima e Hori, 1993), (Oh *et al.*, 1991). Um modo de implementar esta solução é através de um algoritmo de estimação que substitua o sensor de velocidade. Este algoritmo juntamente com amperímetros e voltímetros, realizam uma medição indireta da velocidade, técnica essa denominada *sensorless*.

Várias aplicações *sensorless* estão sendo desenvolvidas, por exemplo, em (Boilleaux e Flaus, 2000) é proposto um modelo básico de medição indireta que permite a estimação de variáveis que não podem ser medidas diretamente. Este método é especialmente desenvolvido com o objetivo de permitir projeto em controle de processos que possuam desenvolvimento matemático limitado, a estimação de estado do processo pode ser monitorada ou controlada.

1.1 Medição Indireta

A medição indireta é muito importante em Controle e Automação de Processos Industriais. Em muitas situações reais, tem-se a necessidade de conhecer o valor de alguma(s) grandeza(s) com dificuldades de medição direta, (Solopchenko *et al.*, 1994). Através da estimação pode-se obter uma boa aproximação do valor dessa(s) grandeza(s) com baixo custo, (Andò *et al.*, 1997). As seguintes referências salientam a importância da medição indireta, (Asaii *et al.*, 1996), (Luetzig *et al.*, 1997), (Xiong e Zhou, 2001), (Raol e Madhuranath, 1996), (Boillereaux e Flaus, 2000), (Moudden e Boubal, 2001), (Shimazu *et al.*, 1992).

Um método indireto que realiza a estimação da precisão em corte de metais, utilizando redes neurais artificiais (RNA's), (Luetzig *et al.*, 1997), apresenta resultados por simulação. A utilização de uma RNA recorrente fornece uma boa habilidade em integrar corretamente valores estimados. A arquitetura possui saída e vetor de entrada atrasados de modo a melhorar o desempenho da RNA. O pré-processamento da entrada estabiliza o valor final estimado, caso haja falha no sensor.

Em (Xiong e Zhou, 2001), apresenta-se a medição indireta para estimação de parâmetros dinâmicos de um sensor eletromagnético de velocidade com bobina móvel. Através das aproximações indiretas e dos modelos de algoritmos no domínio da frequência, esses parâmetros foram estimados com bastante precisão.

A utilização de lógica fuzzy através de intervalos denominados intervalos fuzzy, (Solopchenko *et al.*, 1994), serve como ferramenta base para diminuir os erros de estimação em medidas indiretas.

Um modelo básico de sensor indireto faz a estimação de medidas de variáveis que não podem ser medidas diretamente, (Boillereaux e Flaus, 2000), utilizam-se dois métodos: um algoritmo de estimação com desenvolvimento matemático limitado e as RNA's. Inicialmente, o modelo não linear é identificado pela RNA e, em seguida, utiliza-se o algoritmo de estimação. Esse algoritmo de estimação

consiste em substituir o problema dinâmico pelo critério estático de minimização da soma quadrática dos erros entre os processos medidos e o modelo de saída, através de um ganho horizontal.

Devido ao custo e a sua robustez, a máquina de indução possui uma variedade de aplicações. Aplicações aeroespaciais, inclusive lançamento espacial, em estimação de parâmetros em tempo real, (Stephan *et al.*, 1994). (Stephan *et al.*, 1994), propõe a estimação de parâmetros em um curto período de tempo. O algoritmo é rápido e simples, sendo possível implementá-lo em tempo real com *hardware*. Este método providencia, simultaneamente, o fluxo do rotor estimado e mede as incertezas dessa estimação.

1.2 Objetivos

Os objetivos deste trabalho são:

1. Apresentar o desenvolvimento de um modelo de uma RNA para estimar a velocidade da MI, chamado de estimador neural,
2. Desenvolver algoritmos para estimadores de velocidade da MI,
3. Comparar o desempenho do estimador neural com outras técnicas de estimação.

1.3 Motivação

Devido à necessidade, sob o ponto de vista tecnológico, de substituir o sensor físico por um atuador de velocidade, (Ben-Brahim e Tadakuma, 1998), que não realize esta medição de forma direta (*sensorless*), realizou-se uma análise comparativa do desempenho de algumas técnicas de estimação de modo que seja desenvolvido um modelo que substitua esse sensor. Enfatiza-se a técnica utilizando RNA's por mostrar-se uma alternativa promissora na estimação da velocidade da MI.

1.4 Organização da Dissertação

A seguir, descreve-se a organização da dissertação e o conteúdo dos seus Capítulos e Apêndices. O conteúdo abrange os seguintes tópicos: modelos da máquina de indução, estimadores de velocidade, resultados de simulações e conclusões.

No Capítulo 2, destacam-se aplicações em identificação e controle, abordando o problema existente em identificação e as formas de identificação neural. Ainda no Capítulo 2 apresenta-se uma revisão bibliográfica sobre as mais diversas aplicações de RNA's, tais como: controle, identificação, estimação, falhas, diagnóstico e supervisão. No Capítulo 3, mostram-se os modelos baseados em fluxo do rotor, força eletromotriz e modelo adaptativo de referência para estimar velocidade em máquinas assíncronas (MA) que serão comparados com o modelo neural. No Capítulo 4, apresenta-se um tratamento matemático para estimação da velocidade da MA via RNA. Os algoritmos utilizados para obtenção da velocidade estimada em cada uma das técnicas são mostrados no Capítulo 5, bem como, os resultados de simulações computacionais e análises para cada uma das técnicas. No Capítulo 6 apresenta-se a conclusão.

O texto é constituído de dois Apêndices: Apêndice A - Modelo da Máquina de Indução e Apêndice B - Estruturas dos Blocos para MATLAB/SIMULINK.

Redes Neurais Artificiais em Máquinas de Indução

2.1 Introdução

As RNA's são consideradas técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimentos através da experiência. Representam uma tecnologia que tem raízes em várias áreas, (Elloumi *et al.*, 1998), (Haykin, 2001).

Uma visão geral e conceitual do problema de identificação de sistemas e estimação de parâmetros é apresentada na secção 2.2, enfatizando os tipos, classes dos modelos e as dificuldades que ocorrem no processo de modelagem. Na secção 2.3, apresenta-se uma introdução e conceitos básicos das RNA's, abordando o treinamento de retropropagação, a Regra Delta e os passos para a identificação do processo, estes conceitos são utilizados no Capítulo 4 que trata da estimação da velocidade da MI via RNA's. Na secção 2.4 abordam-se as formas de identificação neural direta e indireta, enfocando a modelagem direta. Nas três últimas secções, apresenta-se uma revisão bibliográfica sobre as aplicações de RNA's em MI, envolvendo controle, falhas, supervisão, diagnóstico, identificação e estimação.

2.2 O Problema da Identificação de Sistemas

Para a utilização prática da teoria de controle, necessita-se construir uma ponte entre o mundo real e a teoria matemática que rege o projeto de controladores. A obtenção da informação sobre o processo a ser controlado, a partir de observações é fundamental para a praticidade da Teoria de Controle. Essa ponte é o procedimento de modelagem ou identificação, onde o modelo descreve, segundo algum objetivo, o mundo real a partir dessas informações. A identificação de sistemas trata a respeito da construção de modelos matemáticos de sistemas dinâmicos a partir de suas medições. Isto é, seleciona-se um modelo para um processo (sistema a ser estudado), usando um número limitado de medições nas entradas e saídas. É necessário determinar uma estrutura de modelos e essa escolha é obtida a partir de algum conhecimento prévio sobre o sistema que gera as medições, (J.Schoukens e Pintelon, 1991). O grau de conhecimento sobre o processo a ser identificado determina a classe do modelo a ser utilizado.

2.2.1 Processo para Identificação

Os modelos podem ser qualitativos ou quantitativos. Este último modelo é utilizado para a determinação do parâmetro do modelo desconhecido, pois é usada uma quantia limitada de informações (medidas), tendo como base a teoria de identificação de sistemas e estimação de parâmetros. Na figura 2.1, ligeira modificação do processo de identificação sugerido por J. Schoukens e Pintelon, mostra-se que a identificação de sistemas é um processo complexo, enquanto a estimação é somente uma etapa. Na estimação de parâmetros, determinam-se experimentalmente os valores dos parâmetros que governam o comportamento dinâmico e/ou não-linear, considerando que a estrutura do modelo do processo seja conhecida, (J.Schoukens e Pintelon, 1991).

Os modelos são agrupadas em classes de acordo com o nível de informações sobre o processo a ser identificado. Como resultado, tem-se os modelos caixa-branca

(modelado pela física ou natureza do processo), indicando que existe um conhecimento total sobre o processo, ou seja, as relações entre as variáveis que descrevem o comportamento dinâmico do sistema são conhecidas. O modelo caixa-preta (modelo empírico, (Aguirre, 2000)) quando não se tem conhecimento algum sobre o processo e os modelos caixa-cinza, se existe algum conhecimento do processo, mas não se conhecem alguns parâmetros ou algumas relações entre as variáveis que descrevem o comportamento dinâmico do sistema.

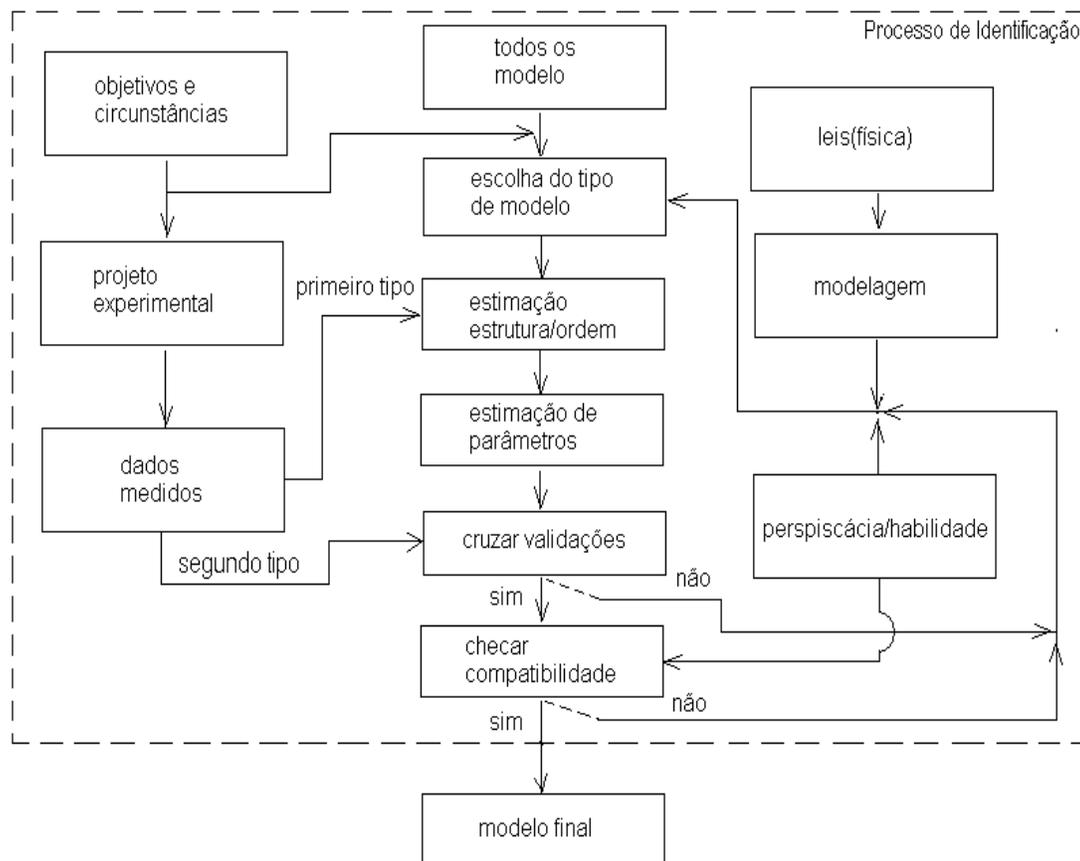


Figura 2.1: Passos para Identificação de Processos

Em aplicações reais é impossível obter uma estrutura de modelo que descreva o sistema de forma exata, pois quase todos os sistemas são não-lineares. Em

vez de tentar encontrar esta representação exata, fazem-se algumas considerações sobre o sistema a ser identificado, tal que a estrutura do modelo seja o mais próximo possível do sistema real. Uma consideração utilizada na identificação de sistemas é que o sistema desconhecido seja linear, ou seja, que em certa região de operação o comportamento possa ser linearizado. Isto é muito difícil de ser verdade no mundo real, mas em muitos casos esta consideração é uma aproximação adequada, considerando que seja possível usar o modelo sem comprometimento das propriedades importantes para ele. Entretanto, se o modelo é necessariamente não-linear ou se é relaxada a consideração de linearidade, fica difícil trabalhar com estes modelos não-lineares. Tais dificuldades são decorrentes do estudo de estabilidade do modelo e da escolha da estrutura do modelo que é uma das etapas mais importantes a ser considerada, (Irwin *et al.*, 1995).

Como o modelo é uma descrição matemática do sistema a ser estudado, pode-se a partir deste obter diferentes categorias de modelo a fim de utilizar um critério específico, (J.Schoukens e Pintelon, 1991).

O uso de modelos parametrizados ou não-parametrizados é definido de acordo com o número limitado de características quantitativas (parâmetros do sistema) ou caso o sistema seja caracterizado por medidas de uma função que possua um grande número de pontos. O modelo não-parametrizado requer, por si próprio, baixo conhecimento do sistema, porém, no parametrizado, a concentração de informações é mais substancial, (Aguirre, 2000).

As RNA's têm uso bastante promissor na identificação de sistemas dinâmicos não-lineares. Considera-se, portanto, que a RNA a ser selecionada está apta a representar a estrutura do sistema em estudo.

2.3 Conceitos de Redes Neurais Artificiais

As RNA's encontram aplicações em diversos campos em virtude de uma importante propriedade: a habilidade de aprender a partir de dados de entrada, (Haykin,

2001). Elas são compostas por várias unidades de processamento, verifica-se, resumidamente, na figura 2.2, a operação de uma unidade de processamento, proposta por McCullock e Pitts¹. O comportamento inteligente da RNA vem das iterações entre essas unidades de processamento da rede, (Haykin, 2001).

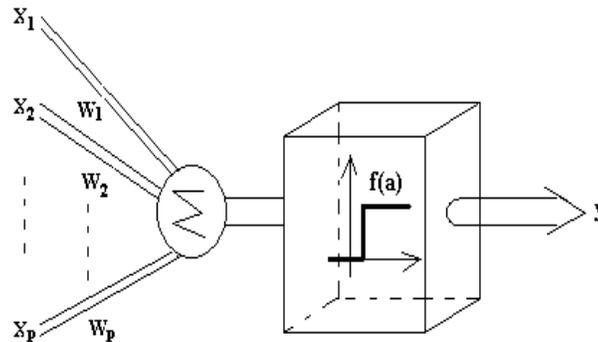


Figura 2.2: Diagrama de unidade RNA de McCullock - Pitts

Na figura 2.2, tem-se:

- $(X_1, X_2, \dots, X_p) \implies$ *sinais* apresentados à entrada;
- $(W_1, W_2, \dots, W_p) \implies$ *peso sináptico*, multiplicador de cada *sinál*, indicando a sua influência na saída da unidade;
- $(W_1X_1 + W_2X_2 + \dots + W_pX_p) \implies$ *soma* ponderada dos *sinais* que produz um nível de atividade;
- $y \implies$ resposta de saída, caso o nível de atividade exceda um certo limite.

¹apresentaram o primeiro modelo de Redes Neurais simulando "máquinas- McCullock, W.S.; Pitts, W.H. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bull Math Biophys, p.115-133.

Dentre os processos de treinamento, tem-se o treinamento supervisionado utilizando retropropagação que consiste em ajustar os pesos e os limites de suas unidades de modo a obter a classificação desejada.

Quando um padrão é inicialmente apresentado à rede, ela produz uma saída. Calcula-se o erro entre a resposta atual ou obtida (modelo neural) e a desejada (ideal, calculada). Com esse erro são realizados os ajustes apropriados nos pesos das conexões até que este erro tenda para o ponto de mínimo, (Haykin, 2001).

O erro é, assim, propagado a partir da camada de saída até as camadas de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados. Esse procedimento é denominado *Regra Delta*, figura 2.3.

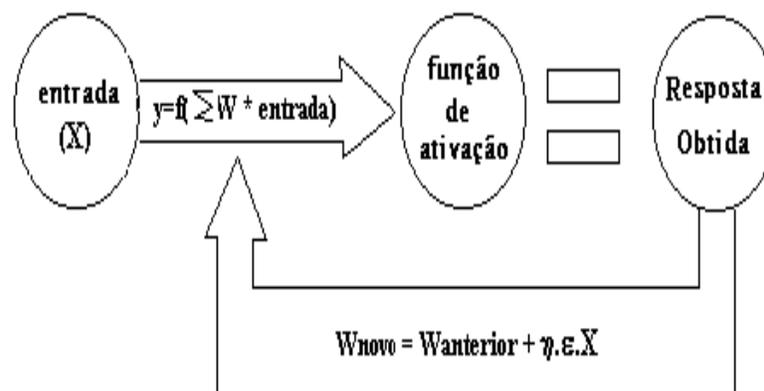


Figura 2.3: Diagrama para treinamento do Perceptron

sendo

η \longrightarrow a taxa de aprendizagem da RNA,

ε \longrightarrow o erro dado pela diferença entre a resposta obtida e a resposta desejada,

X \longrightarrow os sinais de entrada da RNA,

$W_{anterior}$ \longrightarrow peso anterior da RNA,

W_{novo} \longrightarrow novo peso da RNA,

$y \longrightarrow$ nível de ativação dado pela soma ponderada dos sinais de entrada pelos pesos e limitado pela função de ativação.

2.4 Formas de Identificação Neural

Um tópico importante em identificação de sistemas é a questão da *identificabilidade* de sistemas que consiste em analisar se um dado método de identificação produz um único valor para os seus parâmetros e/ou se o modelo identificado corresponde ao sistema real, (?). Na ausência de resultados teóricos, no caso das RNA's, considera-se que todos os sistemas pertençam a uma classe na qual a RNA é capaz de representar, (Irwin *et al.*, 1995).

2.4.1 Modelagem Direta

O procedimento de treinamento da RNA para representar diretamente a dinâmica de um sistema pode ser referido como um modelo direto de acordo com a figura 2.4. O modelo da RNA (M) é colocado em paralelo com o sistema (P) e a diferença entre a saída do modelo neural (y^m) e a saída do sistema (y^p), produz um erro (erro de predição ou resíduo) que é utilizado como sinal de ajuste dos pesos da RNA. Esta estrutura de aprendizagem é um problema clássico de aprendizagem supervisionada sendo que o sistema (P) fornece diretamente os valores das saídas desejadas utilizadas para adaptar o aprendiz (modelo neural). Para o caso em que a rede seja do tipo perceptron multicamada, o erro de predição ou resíduo possibilita um algoritmo de treinamento, (Irwin *et al.*, 1995).

A determinação da natureza dinâmica do sistema estudado e como caracterizar esta dinâmica na RNA é extremamente importante. Uma possibilidade de fazer esta caracterização é introduzir dinâmica na RNA através de laços de realimentação, determinando assim o uso de RNA recorrentes. Outra maneira de introduzir a informação de dinâmica na RNA é através do uso de um conjunto de sinais atrasados no tempo na entrada RNA. Tem-se ainda outra forma de carac-

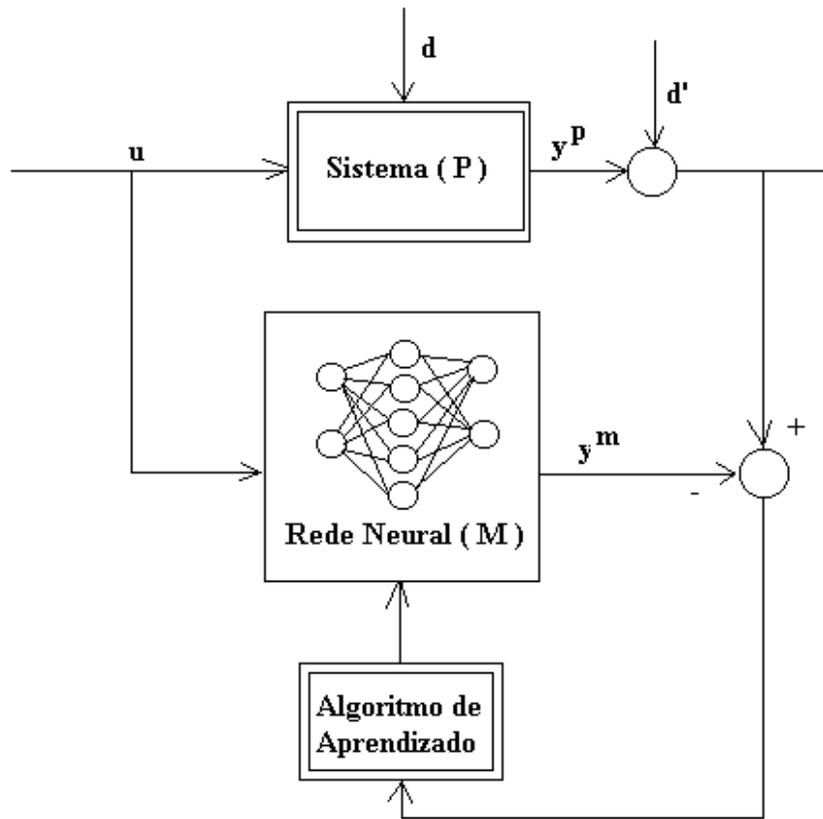


Figura 2.4: Diagrama em Blocos do Modelo Direto para Estimação via RNA's

terizar a dinâmica na RNA que é através do uso de neurônios dinâmicos com a passagem do modelo entrada-saída para o modelo no espaço de estados, (Irwin *et al.*, 1995).

Considera-se que o sistema dinâmico (P) seja governado pela seguinte equação a diferenças não-lineares discreto no tempo,

$$y^p(t+1) = f(y^p(t), \dots, y^p(t-n+1); u(t), \dots, u(t-m+1)). \quad (2.1)$$

A saída do sistema y^p no instante $t+1$ depende dos n valores passados da saída do sistema e dos m valores passados da entrada u do sistema. Utiliza-se somente

a parte dinâmica da resposta do sistema, não sendo representados distúrbios da planta.

Aplicando o modelo não-linear, eq.(2.1), para a saída da rede neural y^m , segue que:

$$y^m(t+1) = \hat{f}(y^p(t), \dots, y^p(t-n+1); u(t), \dots, u(t-m+1), W), \quad (2.2)$$

\hat{f} representa o mapeamento não-linear realizado pela RNA (isto é, aproximação de f) e W é a matriz de parâmetros ajustáveis da rede (os pesos da RNA) através do algoritmo de aprendizagem (o utilizado neste trabalho é o de retropropagação). A entrada da RNA inclui valores passados da saída do sistema real (a rede não apresenta laços internos de realimentação), sendo assim um modelo de RNA com dinâmica externa, (Irwin *et al.*, 1995).

Assumindo que após um período de treinamento suficiente, a RNA consiga uma boa representação do sistema, isto é, $y^m \approx y^p$, então pode-se propor um outro treinamento subsequente, onde a saída da RNA fornecerá os sinais atrasados no tempo para a sua própria entrada. Nesta situação a rede pode ser usada independentemente da planta (sistema real) para a qual foi treinada. Esta outra estrutura de RNA é descrita por:

$$y^m(t+1) = \hat{f}(y^m(t), \dots, y^m(t-n+1); u(t), \dots, u(t-m+1), W). \quad (2.3)$$

O modelo neural, eq.(2.3), também pode ser utilizado para treinar a rede, sem o pré-treinamento feito anteriormente.

2.4.2 Modelagem Inversa

A modelagem inversa é também denominada *aprendizado inverso generalizado*, onde tem-se um conjunto de treinamento específico para uma dada entrada do sistema. A saída do sistema é utilizada como entrada da RNA e a saída da RNA

é comparada com a entrada do sistema. A diferença entre a saída da RNA e a entrada do sistema gera um erro que é usado para treinar a rede, conforme figura 2.5. Tal estrutura força a RNA a representar o modelo inverso do sistema, porém apresenta algumas desvantagens, entre elas, o procedimento de aprendizado não é feito de forma direcionada a uma meta específica e o conjunto de treino deve ser escolhido sob o conjunto de todas as possíveis entradas do sistema; modelos inversos incorretos podem ser obtidos, caso o mapeamento não-linear inverso não seja único. Existem outras abordagens para a modelagem inversa que evitam tais problemas, (Irwin *et al.*, 1995).

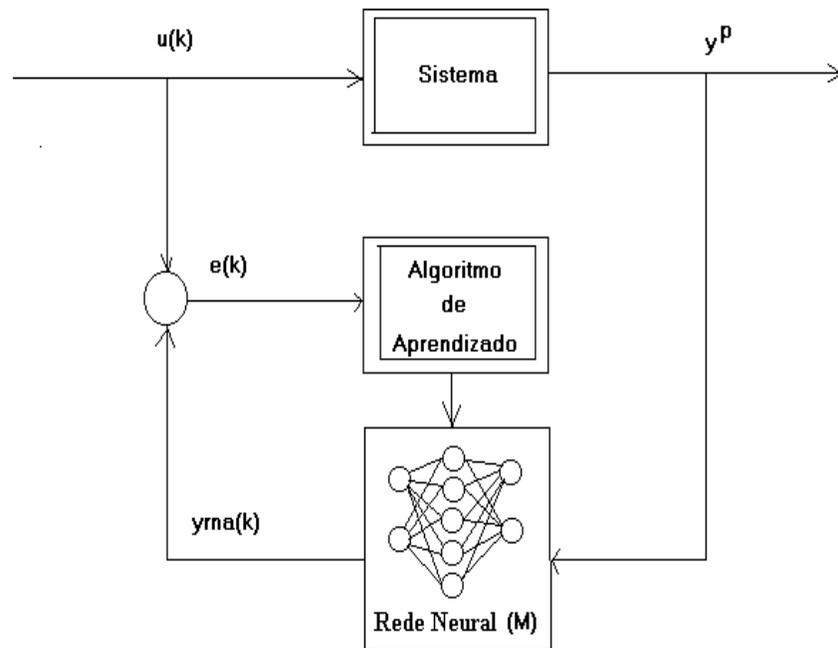


Figura 2.5: Diagrama em Blocos para Modelagem Neural Inversa

2.5 Aplicações de RNA's para MI

Nesta secção são apresentadas algumas aplicações de RNA's para MI como controladores, falhas, supervisão, diagnóstico, identificação e estimação.

2.5.1 Controladores via RNA

Os controladores via RNA's são chamados de neuro-controladores ou controladores neurais ou controladores vetoriais neurais, (Miloudi e Draou, 2001), (Cabrera *et al.*, 1994), (Toqeer e Bayindir, 2000) e (P.Resende e Silvino, 1997) quando aplicados no controle de máquinas de indução. Estes executam tarefas como controles de velocidade, direto de torque, (Wang e Elbuluk, 1996) e de eficiência energética.

O controle de velocidade, (Miloudi e Draou, 2001), controle direto de torque, (Cabrera *et al.*, 1994) e (Toqeer e Bayindir, 2000) são recentes aplicações de RNA's em MI.

Um controlador neural é projetado para o ajuste da velocidade em (Miloudi e Draou, 2001). São apresentados resultados computacionais, o motor atinge rapidamente a velocidade de referência e sem *overshoot*. Reposta ao degrau são rapidamente acompanhadas e sem *overshoot*. Além do mais, este controlador mostra habilidade em lidar com variações dos parâmetros do motor. O controlador neural é treinado a partir de um controlador PI, sua entrada e saída são simuladas e essas são utilizadas para o treinamento da RNA com uma camada oculta.

Em (P.Resende e Silvino, 1997) é proposto um esquema de RNA para orientação do campo e controle do torque da MI, usando a técnica de ganho programado. Um outro esquema de RNA é também proposto em (Zhang e Hasan, 1999) para fazer o controle eficiente de energia da MI. Esta técnica treina a RNA utilizando dados da MI controlada vetorialmente.

A RNA é empregada para simular o controle de torque direto a partir do selector de estado em (Wang e Elbuluk, 1996). O treino é realizado a partir de Algoritmo Genético (AG) que usa dados na representação binária e em ponto

flutuante. É feita uma comparação entre essas duas representações.

Em (Toqeer e Bayindir, 2000), um neuro controlador-MLP é projetado para substituir o controle de campo orientado (FOC) e o controlador PI. Resultados das simulações computacionais do FOC são utilizados para o treinamento do controlador proposto e o algoritmo de treinamento é o *backpropagation*. Resultados computacionais mostram que o neuro controlador é uma alternativa para o controle FOC desacoplado. De modo similar ao neuro-controlador, (Miloudi e Draou, 2001), este controlador apresenta a vantagem de ser insensível à variação dos parâmetros do motor.

A aplicação de uma RNA para o controle de uma MI utilizando DTC é apresentado em (Cabrera *et al.*, 1994). A RNA é utilizada para emular o seletor de estado do DTC. Esta é treinada através de quatro algoritmos que são: *backpropagation*, modelo adaptativo de neurônio, Filtro de Kalman Estendido e erro de predição paralelo-recursivo. Esses métodos de treinamentos são comparados através de simulações e os dois últimos apresentam-se como melhores alternativas.

2.5.2 Falhas, supervisão e diagnóstico via RNA

(Frosini e Petrecca, 2001) apresenta a possibilidade em usar a RNA para monitoramento do torque da MI. As RNA's não-supervisionadas são usadas para detectar possíveis falhas do torque, enquanto as supervisionadas, para identificar o valor médio do regime permanente da carga.

Modelo de RNA é empregado no processo de aprendizagem para o caso em que o comportamento da máquina dinâmica apresenta frequência e distúrbios de torque diferentes, podendo, portanto, ser uma alternativa eficiente na realização de uma correspondência entre tais valores, (Filippetti *et al.*, 1998b).

O desenvolvimento de um modelo empírico baseado em diagnóstico de falha do sistema de uma MI, usando a dinâmica da RNA recorrente e o método de processamento de sinal multi-resolução é apresentado em (Kim e Parlos, 2002). O diagnóstico do sistema usa medidas no terminal do motor da corrente, tensão

e, também, da velocidade do rotor.

Em (Wai e Chang, 2002) é proposto um sistema de controle de posição do rotor e do campo orientado indireto do *drive* do servo motor de indução inteligente que é treinado *on-line* por um controlador Neural com taxa de aprendizagem adaptativa.

Desenvolvimentos mais recentes são apresentados em vários artigos no campo da Inteligência Artificial (IA), atuando em diagnóstico de sistemas para MI e *drives* (Filippetti *et al.*, 2000). Uma combinação das técnicas Sistemas Inteligentes, Lógica Nebulosa e RNA pode ser integrada para resolver problemas específicos. Essas tecnologias em diagnóstico *on-line* para MI tem uso bastante promissor na indicação de falhas existentes, (Filippetti *et al.*, 1998a).

Um algoritmo de treinamento *on-line* por RNA baseado em esquema de detecção de falhas é proposto em (Tallam *et al.*, 2001). O algoritmo elimina a necessidade de uma memória com grande capacidade e um longo tempo de treinamento que é justamente a limitação da maioria dos métodos diagnóstico baseado em IA para aplicações comerciais. Ainda usando RNA *on-line*, em (Tallam *et al.*, 2000), é mostrado o diagnóstico e detecção de falhas no enrolamento do estator. Esse esquema consiste na RNA com realimentação direta combinada com um SOFM (Self-Organizing Feature Map) para visualizar no display a condição de operação da máquina em duas dimensões. Em (L.Collamati *et al.*, 1996) são apresentados os benefícios que a condição de monitoramento fornece à máquina. Essa estratégia operacional procura fazer uma avaliação da integridade da máquina, identificação de falhas e uma extensão da vida útil. O diagnóstico é projetado usando RNA e pode ser ativado para detectar falhas graves.

Há ainda em (Stefano *et al.*, 1994) o uso da RNA em diagnóstico de falhas no estator da MI. O algoritmo apresentado fornece uma boa precisão nos resultados. Já em (Filippetti *et al.*, 1995) a RNA é utilizada para diagnosticar falhas no rotor da MI.

Em (Shoudao e Youjie, 2001), o estimador neural observa a corrente de estator para detectar a carga na resistência do estator. O esquema para estimação da

resistência para controle de torque direto da MI é apresentado, utilizando toolbox da Rede Neural do MATLAB.

O uso de técnicas de Inteligência Computacional (RNA's e AG) é também apresentado em (Elsimary, 1997) como método para detecção de falhas na rotação da máquina. Neste caso, o AG é utilizado para treinar a RNA de modo a realçar o detector neural. O uso da RNA em diagnóstico de falhas é empregado em (Filippetti *et al.*, 1996) na combinação do rotor com o momento de inércia da carga do rotor na MI.

2.5.3 Identificação e Estimação via RNA

Em (Asaii *et al.*, 1996) utiliza-se a RNA para controlar a velocidade através de medição indireta (*sensorless*) da MI. Neste artigo tem-se duas redes: a primeira que é treinada para estimar a velocidade e o torque da MI os quais serão utilizados na realimentação do controle do sistema e, a segunda, que obtém a tensão (V) e a frequência (f).

Cada RNA possui três camadas, sendo o número de neurônios das camadas de entrada e saída igual ao número de entradas e saídas do sistema. São mostrados os modos de Controle da MI em malha aberta e em malha fechada.

A RNA tem aplicação de grande potencial na MI, usando Controle de Torque Direto (Cabrera *et al.*, 1997), isto é reforçado em (Wishart e Harley, 1993) em que esta é utilizada para Identificar e Controlar a MI. São considerados dois sistemas: um envolvendo a corrente de estator e outro, a velocidade do rotor.

A estrutura da RNA é implementada em linguagem de máquina em (Bordon *et al.*, 1999), utiliza uma memória de dados para armazenar o valor rms da corrente de rotor (I_r) como uma função da frequência de controle e a frequência de escorregamento de modo a fazer um ajuste da velocidade da MI usando PID com ganho planejado.

A implementação do protótipo do *hardware* que treina a RNA *on-line* para identificar a dinâmica da MI e controlar a corrente de estator para um inversor

PWM é apresentada em (Burton *et al.*, 1998).

O uso da RNA no processo como ferramenta para o Controle e Identificação de Sistemas cresceu rapidamente em pouco tempo. Em diversos artigos são feitas análises do uso dessa ferramenta tanto para Identificação quanto para o Controle (Martinez e Martinez, 1997). São apresentados em (Burton e Harley, 1994) novos métodos computacionais de modo a incrementar a RNA. Em (Martinez e Martinez, 1997), o processo de aprendizagem da RNA é analisado com diferentes números de neurônios de camadas intermediárias, taxa de aprendizagem e de amostragem diferentes.

Em (Grzesiak e Skoczylas, 1996), a RNA é utilizada como estimador para se obter o torque e o fluxo do estator de posse do conhecimento da corrente e da velocidade do estator. A RNA, utilizando o algoritmo de aprendizagem retro-propagação, é também empregada em (D'Angelo e CostaJr, 2000) para estimar o torque e a velocidade da MI. Os modelos dos estimadores são baseados nas equações da MI.

Técnicas para Estimar a Velocidade da MI

3.1 Introdução

As MI's são bastantes utilizadas na indústria. Normalmente utilizam-se duas técnicas para o controle da MI: controle escalar, tal como: v/f , e o controle vetorial. O controle escalar v/f , também considerado controle em malha aberta, não possui características dinâmicas satisfatórias. É necessária uma realimentação da velocidade. O controle vetorial é mais utilizado devido ao seu alto desempenho no controle de sistemas, (Eloumi *et al.*, 1998), considerado controle em malha fechada necessita de informação da velocidade para sua implementação.

No trabalho de (Rajashekara *et al.*, 1996) as técnicas *sensorless*, para o controle da MI, são classificadas em dois tipos: controle de velocidade em malha aberta com compensação de escorregamento e controle em malha fechada com velocidade estimada. No controle de velocidade em malha aberta, a velocidade é regulada enquanto a frequência de escorregamento estimada é utilizada para compensar modificações na carga. No controle em malha fechada, a velocidade do motor estimada é utilizada para regulação da velocidade. Algumas dessas técnicas são:

cálculo da frequência de escorregamento, estimação de fluxo, controle direto de torque e fluxo, MRAS, filtragem de Kalman e controle *sensorless* com adaptação de parâmetros.

Modelos e aplicações de estimação *sensorless* da velocidade podem ser verificados em (Armstrong *et al.*, 1997), (Rajashekara *et al.*, 1993) e (FonsecaNeto e Mouzinho, 2003). Estes autores comparam através de simulações e/ou experimentos diversos tipos de técnicas para estimar a velocidade, tais como, fluxo do rotor, FEM, MRAS, Filtro de Kalman Estendido e RNA's.

Neste capítulo são apresentadas técnicas para estimação da velocidade em MI que são: estimação via fluxo do rotor, estimação via Força Eletromotriz (FEM) e estimação baseada na técnica Sistema Adaptativo do Modelo de Referência (MRAS). Nesta última técnica analisa-se a estimação da velocidade através do fluxo.

3.2 Modelo da Máquina de Indução

Esta seção apresenta de forma sucinta o modelo do processo a ser controlado, no caso, a MI. No Apêndice A, esse modelo é apresentado de forma mais detalhada.

O modelo utiliza as grandezas tensão e a corrente do motor referenciadas a um sistema estacionário dq0. As equações da máquina em termos das quantidades vetoriais são, (Ben-Brahim, 1995), (Elloumi *et al.*, 1998), (Ben-Brahim e Tadakuma, 1998), (Ong, 1998):

$$\vec{v}_s = R_s \vec{i}_s + \dot{\vec{\lambda}}_s, \quad (3.1)$$

$$0 = R_r \vec{i}_r + \dot{\vec{\lambda}}_r + j\omega_r \vec{\lambda}_r, \quad (3.2)$$

$$\vec{\lambda}_s = L_s \vec{i}_s + L_m \vec{i}_r, \quad (3.3)$$

$$\vec{\lambda}_s = L_s \vec{i}_s + \vec{\lambda}_m, \quad (3.4)$$

$$\vec{\lambda}_r = L_m \vec{i}_s + L_r \vec{i}_r, \quad (3.5)$$

$$\vec{\lambda}_r = L_r \vec{i}_r + \vec{\lambda}_m, \quad (3.6)$$

sendo

$$\vec{v}_s = \begin{bmatrix} \vec{v}_{ds} & \vec{v}_{qs} \end{bmatrix}' \quad \text{a tensão no estator,}$$

$$R_s \quad \text{a resistência no estator,}$$

$$\vec{i}_s = \begin{bmatrix} \vec{i}_{ds} & \vec{i}_{qs} \end{bmatrix}' \quad \text{a corrente no estator,}$$

$$\dot{\vec{\lambda}}_s = \frac{d\vec{\lambda}_s}{dt} \quad \text{tensão induzida no estator (derivada do fluxo em relação ao tempo).}$$

Considerando $\vec{v}_r = \begin{bmatrix} \vec{v}_{dr} & \vec{v}_{qr} \end{bmatrix}' = 0$ (rotor curto-circuitado) e $\omega = 0$. As eqs.(3.1), (3.2), (3.3) e (3.5) na forma matricial, ficam

$$\begin{bmatrix} \vec{v}_s \\ 0 \\ \vec{\lambda}_s \\ \vec{\lambda}_r \end{bmatrix} = \begin{bmatrix} R_s & \frac{d}{dt} & 0 & 0 \\ 0 & 0 & R_r & \frac{d}{dt} \\ L_s & 0 & L_m & 0 \\ L_m & 0 & L_r & 0 \end{bmatrix} \begin{bmatrix} \vec{i}_s \\ \vec{\lambda}_s \\ \vec{i}_r \\ \vec{\lambda}_r \end{bmatrix} + j\omega_r \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \vec{\lambda}_r \\ 0 \\ 0 \end{bmatrix}. \quad (3.7)$$

3.3 Estimação via Fluxo do Rotor

O método de estimação de velocidade utiliza o modelo da MI descrito na secção anterior. A velocidade do rotor, ω_r , é obtida com orientação do fluxo rotórico (λ_r), (Boldea e Nasar, 1998). Baseia-se na estimação de ω_r a partir das eqs.(3.8) e (3.9). Como a magnitude do fluxo do rotor se mantém constante durante o regime permanente da velocidade, ele é conveniente para o cálculo da velocidade (Elloumi *et al.*, 1998). Das eqs.(3.1), (3.2), (3.3) e (3.5) o fluxo do rotor pode ser escrito como:

$$\dot{\vec{\lambda}}_{1r} = \frac{L_r}{L_m} (\vec{v}_s - R_s \vec{i}_s - \sigma L_s \dot{\vec{i}}_s) \quad (3.8)$$

ou

$$\dot{\vec{\lambda}}_{2r} = \left(-\frac{1}{T_r} + \omega_r j\right) \vec{\lambda}_r + \frac{L_m}{T_r} \vec{i}_s, \quad (3.9)$$

sendo $\dot{\vec{\lambda}}_{1r} = \frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{1dr} \\ \vec{\lambda}_{1qr} \end{bmatrix}$ e $\dot{\vec{\lambda}}_{2r} = \frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{2dr} \\ \vec{\lambda}_{2qr} \end{bmatrix}$,

$\dot{\vec{\lambda}}_{1r}$ e $\dot{\vec{\lambda}}_{2r}$ são expressões equivalentes. Ao contrário de $\dot{\vec{\lambda}}_{2r}$, o fluxo $\dot{\vec{\lambda}}_{1r}$ não está em função de ω_r .

As eqs.(3.8) e (3.9) do fluxo na referência dq, são:

$$\frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{1dr} \\ \vec{\lambda}_{1qr} \end{bmatrix} = \frac{L_r}{L_m} \begin{bmatrix} \vec{v}_{ds} \\ \vec{v}_{qs} \end{bmatrix} - \begin{bmatrix} (R_s + \sigma L_s \frac{d}{dt}) & 0 \\ 0 & (R_s + \sigma L_s \frac{d}{dt}) \end{bmatrix} \begin{bmatrix} \vec{i}_{ds} \\ \vec{i}_{qs} \end{bmatrix} \quad (3.10)$$

e

$$\frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{2dr} \\ \vec{\lambda}_{2qr} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_r} & -\omega_r \\ \omega_r & -\frac{1}{T_r} \end{bmatrix} \begin{bmatrix} \vec{\lambda}_{2dr} \\ \vec{\lambda}_{2qr} \end{bmatrix} + \frac{L_m}{T_r} \begin{bmatrix} \vec{i}_{ds} \\ \vec{i}_{qs} \end{bmatrix}. \quad (3.11)$$

A eq.(3.11) pode ser representada como

$$\dot{\vec{\lambda}}_{2dr} = -\frac{1}{T_r} \vec{\lambda}_{2dr} - \omega_r \vec{\lambda}_{2qr} + \frac{L_m}{T_r} \vec{i}_{ds} \quad (3.12)$$

e

$$\dot{\vec{\lambda}}_{2qr} = -\frac{1}{T_r} \vec{\lambda}_{2qr} + \omega_r \vec{\lambda}_{2dr} + \frac{L_m}{T_r} \vec{i}_{qs}. \quad (3.13)$$

O ângulo do vetor fluxo do rotor dado por θ é definido como,

$$\theta = \arctan\left(\frac{\lambda_{2qr}}{\lambda_{2dr}}\right). \quad (3.14)$$

Derivando θ , obtém-se a velocidade rotórica,

$$\dot{\theta} = \frac{\lambda_{2dr}\dot{\lambda}_{2qr} - \lambda_{2qr}\dot{\lambda}_{2dr}}{\lambda_{2dr}^2},$$

$$\dot{\theta} = \frac{\lambda_{2dr}\dot{\lambda}_{2qr} - \lambda_{2qr}\dot{\lambda}_{2dr}}{\lambda_{2dr}^2 + \lambda_{2qr}^2},$$

$$\dot{\theta} = \frac{\lambda_{2dr}\dot{\lambda}_{2qr} - \lambda_{2qr}\dot{\lambda}_{2dr}}{\lambda_{2dr}^2 + \lambda_{2qr}^2},$$

ou seja,

$$\dot{\theta} = \frac{\lambda_{2dr}\dot{\lambda}_{2qr} - \lambda_{2qr}\dot{\lambda}_{2dr}}{|\vec{\lambda}_{2r}|^2} \quad (3.15)$$

e substituindo $\dot{\lambda}_{2dr}$ e $\dot{\lambda}_{2qr}$ das eqs.(3.12) e (3.13) na expressão (3.15), vem

$$\dot{\theta} = \frac{\lambda_{2dr}\left(\frac{-1}{T_r}\lambda_{2qr} + \omega_r\lambda_{2dr} + \frac{L_m}{T_r}i_{qs}\right) - \lambda_{2qr}\left(\frac{-1}{T_r}\lambda_{2dr} - \omega_r\lambda_{2qr} + \frac{L_m}{T_r}i_{ds}\right)}{|\vec{\lambda}_{2r}|^2},$$

$$\dot{\theta} = \frac{\frac{-1}{T_r}\lambda_{2dr}\lambda_{2qr} + \omega_r(\lambda_{2dr})^2 + \frac{L_m}{T_r}\lambda_{2dr}i_{qs} + \frac{1}{T_r}\lambda_{2qr}\lambda_{2dr} + \omega_r(\lambda_{2qr})^2 - \frac{L_m}{T_r}\lambda_{2qr}i_{ds}}{|\vec{\lambda}_{2r}|^2},$$

$$\dot{\theta} = \frac{\omega_r|\lambda_{2r}|^2 + \frac{L_m}{T_r}(\lambda_{2dr}i_{qs} - \lambda_{2qr}i_{ds})}{|\vec{\lambda}_{2r}|^2},$$

$$\dot{\theta} = \omega_r + \frac{L_m}{T_r}\left(\frac{\lambda_{2dr}i_{qs} - \lambda_{2qr}i_{ds}}{|\vec{\lambda}_{2r}|^2}\right),$$

$$\omega_r = \dot{\theta} - \frac{L_m}{T_r}\left(\frac{\lambda_{2dr}i_{qs} - \lambda_{2qr}i_{ds}}{|\vec{\lambda}_{2r}|^2}\right). \quad (3.16)$$

Então, substituindo-se $\lambda_{2dr}i_{qs} - \lambda_{2qr}i_{ds}$ por $\vec{\lambda}_{2r} \otimes \vec{i}_s$ na eq.(3.16), tem-se

$$\omega_r = \dot{\theta} - \frac{L_m}{T_r} \frac{|\vec{\lambda}_{2r} \otimes \vec{i}_s|}{|\vec{\lambda}_{2r}|^2}. \quad (3.17)$$

O segundo termo da eq.(3.17) é a velocidade angular de escorregamento ω_{ls} :

$$\omega_{ls} = \frac{L_m}{T_r} \frac{|\vec{\lambda}_{2r} \otimes \vec{i}_s|}{|\vec{\lambda}_{2r}|^2}. \quad (3.18)$$

A velocidade do rotor é determinada a partir da diferença entre a velocidade angular síncrona e a velocidade angular de escorregamento.

$$\omega_r = \omega_s - \omega_{ls}, \quad (3.19)$$

sendo ω_s e ω_{ls} estimadas segundo as eqs.(3.15) e (3.18), respectivamente.

A eq.(3.16) indica que a velocidade angular instantânea do vetor fluxo do rotor, bem como, o escorregamento relativo instantâneo do rotor, pode ser obtida do estimador baseado no fluxo do rotor na eq.(3.9), (Elloumi *et al.*, 1998).

O diagrama em blocos da figura 3.1 representa o modelo para estimação de velocidade via fluxo. Verifica-se que neste modelo as entradas são as tensões \vec{v}_s e as correntes \vec{i}_s , (Elloumi *et al.*, 1998).

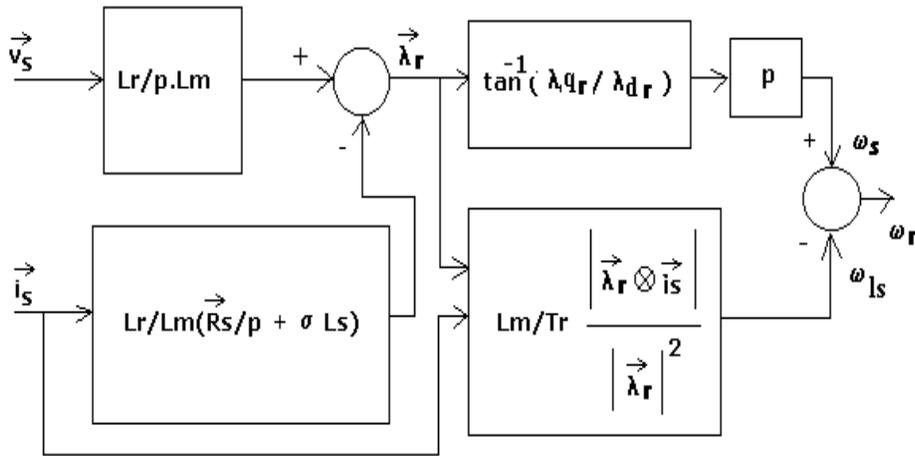


Figura 3.1: Modelo para estimação via fluxo do Rotor

3.4 Estimação via FEM

Neste método de estimação, obtém-se a velocidade do rotor, ω_r , por intermédio da FEM. Baseia-se na estimação de ω_r a partir das eqs.(3.8) e (3.26) que é deduzida a seguir. A eq.(3.26) envolve diretamente a FEM e os parâmetros da máquina.

Assumindo a variação da magnitude do fluxo do rotor em estado estacionário, a FEM pode ser aproximada como segue, (Elloumi *et al.*, 1998):

$$\vec{e}_r = \frac{d}{dt} \vec{\lambda}_r \approx j\omega_s \vec{\lambda}_r, \quad (3.20)$$

sendo,

$$\vec{e}_r = \begin{bmatrix} e_{dr} \\ e_{qr} \end{bmatrix}. \quad (3.21)$$

Os fluxos rotóricos são calculados como

$$\lambda_{1dr} \cong \frac{e_{qr}}{\omega_s} \quad (3.22)$$

e

$$\lambda_{1qr} \cong \frac{-e_{dr}}{\omega_s}, \quad (3.23)$$

tem-se que

$$\begin{bmatrix} \lambda_{1dr} \\ \lambda_{1qr} \end{bmatrix} = \frac{1}{\omega_s} \begin{bmatrix} e_{qr} \\ -e_{dr} \end{bmatrix}, \quad (3.24)$$

sendo e_{dr} e e_{qr} as FEM rotóricas.

Como

$$\vec{\lambda}_{1r} \otimes \vec{i}_s = \lambda_{1dr} i_{qs} - \lambda_{1qr} i_{ds} = \begin{vmatrix} i & j & k \\ \lambda_{1dr} & \lambda_{1qr} & 0 \\ i_{ds} & i_{qs} & 0 \end{vmatrix},$$

substituindo-se os fluxos do determinante acima pelos da eq.(3.24), vem

$$\vec{\lambda}_{1r} \otimes \vec{i}_s = \begin{vmatrix} i & j & k \\ \frac{e_{qr}}{\omega_s} & \frac{e_{dr}}{\omega_s} & 0 \\ i_{ds} & i_{qs} & 0 \end{vmatrix}.$$

Resolvendo

$$\vec{\lambda}_{1r} \otimes \vec{i}_s = \frac{e_{qr}}{\omega_s} i_{qs} - \frac{e_{dr}}{\omega_s} i_{dr} \quad (3.25)$$

e substituindo a eq.(3.25) na eq.(3.17), obtém-se

$$\omega_r = \omega_s - \frac{L_m}{T_r} \omega_s \left(\frac{e_{qr} \cdot i_{qs} - e_{dr} i_{ds}}{e_{dr}^2 + e_{qr}^2} \right),$$

ou seja,

$$\omega_r = \omega_s - \frac{L_m}{T_r} \omega_s \frac{(\vec{e}_r \circ \vec{i}_s)}{|\vec{e}_r|^2}. \quad (3.26)$$

A velocidade angular de escorregamento é dada por, (Elloumi *et al.*, 1998):

$$\omega_{ls} = \frac{L_m}{T_r} \omega_s \frac{(\vec{e}_r \circ \vec{i}_s)}{|\vec{e}_r|^2}, \quad (3.27)$$

sendo \circ , o produto interno de vetores. A FEM, \vec{e}_r , é calculada da eq.(3.8) e o diagrama em blocos que representa o modelo para estimação da velocidade rotórica via FEM é dado na figura 3.2. Este diagrama é similar ao da figura 3.1, a diferença está na ausência da integração no bloco que tem a corrente como entrada, (Elloumi *et al.*, 1998). Com este método ocorre um problema de precisão se a frequência do estator se aproxima do zero.

3.5 Estimação baseada na técnica MRAS

O Sistema Adaptativo do Modelo de Referência - (MRAS-Model Reference Adaptive System) de aproximação usa a redundância de dois modelos da máquina de indução com estruturas diferentes. O modelo que não envolve a quantidade ω_r

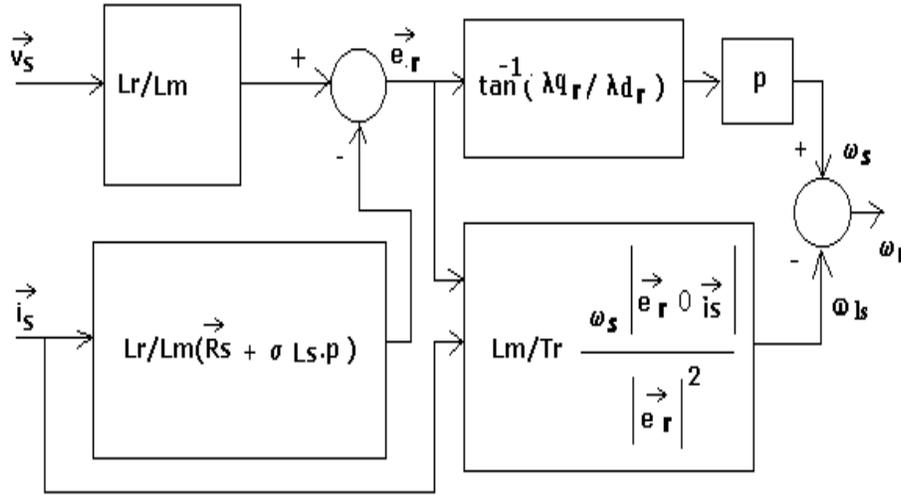


Figura 3.2: Modelo para estimação via FEM do Rotor

é considerado como referência e o que contém ω_r a ser estimada, é o modelo ajustável, (Elloumi *et al.*, 1998). Verifica-se se a variável estimada pelo modelo ajustável se aproxima da variável do modelo de referência e cada modelo tem como entrada diferentes conjuntos de variáveis, (Boldea e Nasar, 1998). O erro de estimação é obtido através da diferença das variáveis X e \hat{X} dos modelos de referência e ajustável, respectivamente. O erro $\varepsilon = X - \hat{X}$ é usado para acionar um mecanismo de adaptação que gera a velocidade do rotor estimada, ω_r , para o modelo ajustável, (Boldea e Nasar, 1998), (Elloumi *et al.*, 1998). O diagrama em blocos para este método é visto na figura 3.3.

3.5.1 Estimação via Fluxo do Rotor baseada na técnica MRAS

A comparação entre dois estimadores de fluxo, sendo que uma das equações contém a velocidade ω_r , é empregada nesta técnica MRAS via fluxo do rotor.

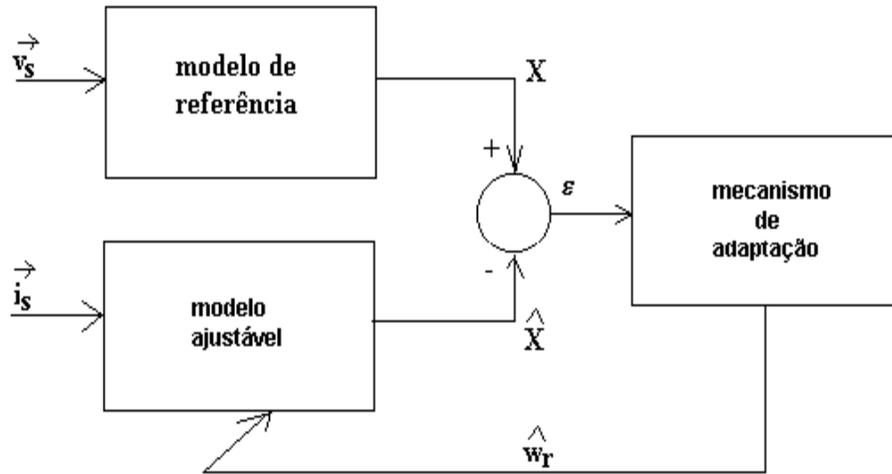


Figura 3.3: Modelo para Estimação baseada em MRAS

Desse modo, o erro entre esses dois estimadores é usado para fornecer um modelo de mecanismo de adaptação que produz a velocidade estimada do rotor, $\hat{\omega}_r$, para o modelo ajustável, (Boldea e Nasar, 1998).

O fluxo do rotor da eq.(3.8) não envolve a velocidade ω_r . Portanto, ele é selecionado para gerar o valor de referência do fluxo do rotor. Todavia, a eq.(3.9) que envolve ω_r gera o valor estimado do fluxo do rotor e, pode ser denominado como modelo ajustável. Tendo completo conhecimento dos parâmetros do motor e da velocidade do motor, as saídas dos dois modelos coincidem. Entretanto, qualquer diferença entre a velocidade usada no modelo ajustável e a velocidade do motor, resultará automaticamente em um erro entre as saídas dos dois estimadores. Este erro, $\varepsilon = \lambda_{1r} - \hat{\lambda}_{2r}$, é usado para ajustar o mecanismo de adaptação que produz a velocidade do motor na eq.(3.9), (Elloumi *et al.*, 1998). O vetor erro possui os seguintes componentes:

$$\begin{aligned}\varepsilon_d &= \lambda_{1dr} - \hat{\lambda}_{2dr} \\ \varepsilon_q &= \lambda_{1qr} - \hat{\lambda}_{2qr} \\ [\varepsilon] &= \begin{bmatrix} \varepsilon_d \\ \varepsilon_q \end{bmatrix},\end{aligned}$$

ou seja,

$$\varepsilon = \lambda_{1r} - \hat{\lambda}_{2r}. \quad (3.28)$$

O modelo do rotor é o modelo ajustável que produz o mesmo vetor de fluxo que o modelo de referência. O modelo para o mecanismo de adaptação da velocidade estimada é dado por, (Schauder, 1992):

$$\hat{\omega}_r = (k_p + \frac{k_i}{p})(\lambda_{1r} \otimes \hat{\lambda}_{2r}), \quad (3.29)$$

ou seja,

$$\hat{\omega}_r = k_p(\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr}) + \frac{k_i}{p}(\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr}), \quad (3.30)$$

sendo p , o operador derivativo.

Então,

$$\hat{\omega}_r = k_p(\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr}) + k_i \int (\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr}). \quad (3.31)$$

O modelo para estimação do fluxo via fluxo do rotor baseado em MRAS é apresentado na figura 3.4.

A presença da integração aberta no estator prevê problemas com condições iniciais. Em (Schauder, 1992) propõe-se a utilização de um filtro passa-baixas ao invés de um integrador. Porém, o uso de filtro passa-baixas degrada a estimação para baixa velocidade durante transitórios, o que impede o seu funcionamento por muito tempo nessa região.

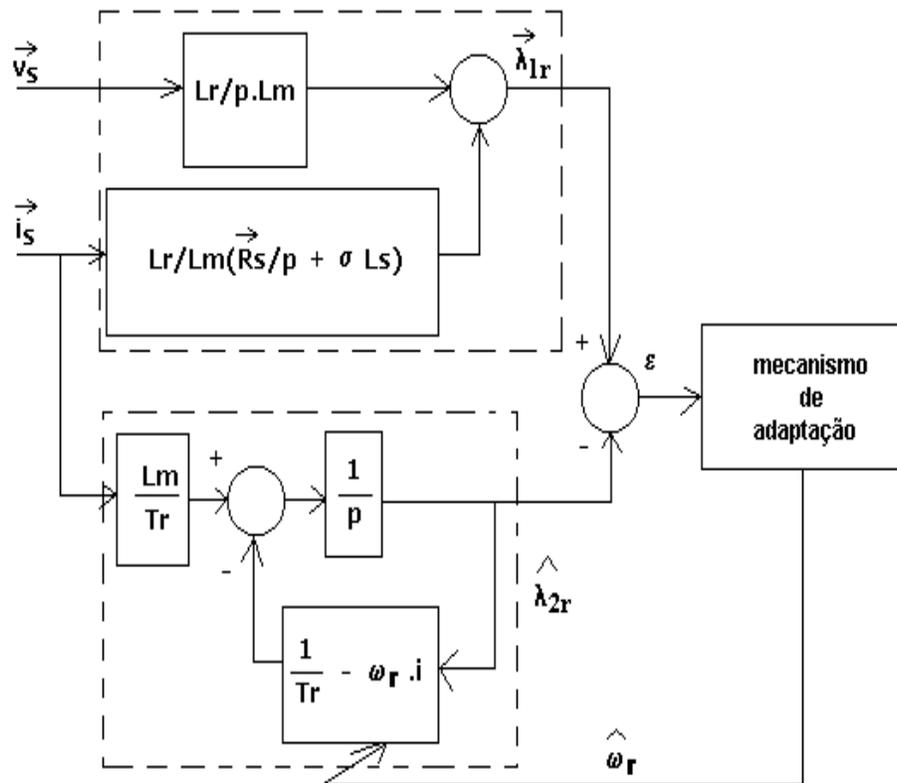


Figura 3.4: Modelo para estimação via fluxo do rotor baseada em MRAS

Estimação via Redes Neurais Artificiais

4.1 Introdução

Neste capítulo apresenta-se o desenvolvimento de um modelo neural para a estimação de velocidade de uma MI. Este modelo tem por base duas equações de fluxo que são calculadas a partir das tensões e correntes do estator. Estas duas equações são utilizadas para montar dois modelos. O primeiro modelo utiliza uma equação de fluxo que não envolve a velocidade da MI, esta acoplada ao modelo da máquina de indução forma um modelo ampliado do sistema real. O segundo modelo é construído a partir de uma equação de fluxo que depende da velocidade e é denominado de modelo neural. A saída destes dois modelos são expressas em termos de fluxo e a diferença entre estas duas saídas gera um erro. Este erro é utilizado para o treinamento da rede neural. Um dos pesos sinápticos da RNA fornece uma expressão para estimar a velocidade e esta é ajustada a cada passo do treinamento da rede.

As secções seguintes apresentam o desenvolvimento algébrico para obtenção do estimador neural, o modelo da rede e o tipo de treinamento utilizado para o

ajuste dos pesos. Dentre as formas de identificação neural, utiliza-se a modelagem direta, conforme a figura 2.4, para a estrutura da RNA que estima ω_r .

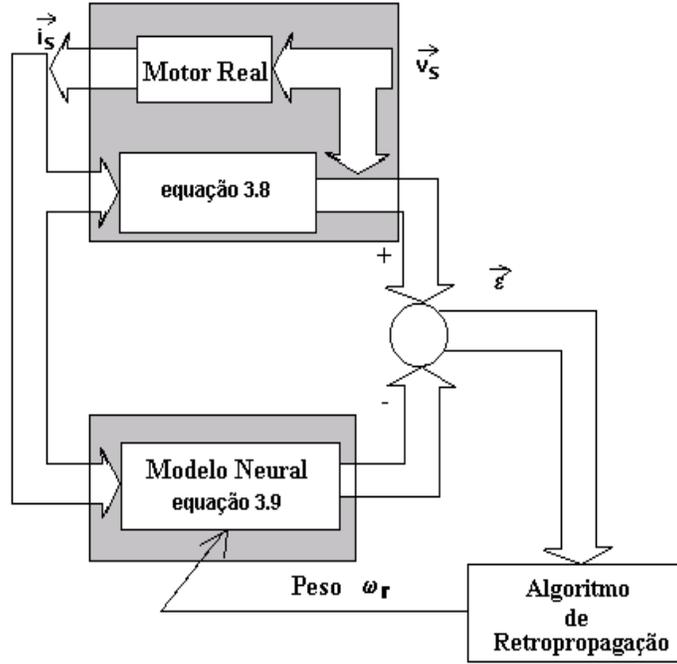
4.2 Estimação via RNA's

A estimação via RNA's apresentada neste trabalho tem por base dois estimadores independentes. Estes são expressos através de uma estrutura estacionária e são usados para estimar os fluxos no rotor, um dos estimadores é baseado na eq.(3.8) e o outro na eq.(3.9), (Elloumi *et al.*, 1998). O estimador baseado na eq.(3.8) que não envolve a velocidade ω_r é selecionado para gerar o valor de referência dado pelo fluxo no rotor, λ_r , e a eq.(3.9) que envolve ω_r gera o fluxo estimado pelo modelo neural, (Ben-Brahim, 1995), (Elloumi *et al.*, 1998). De posse do completo conhecimento dos parâmetros do motor, os dois modelos devem coincidir. Entretanto, qualquer diferença entre a velocidade utilizada no modelo neural e na velocidade do motor pode resultar automaticamente em erro entre a saída dos dois estimadores. Este erro entre o fluxo de referência e o fluxo do modelo neural será utilizado para atualizar os pesos da RNA, ou seja, a velocidade do rotor, ω_r , na eq.(3.9). Esta aproximação, figura 4.1, apresenta o algoritmo de retropropagação derivado de forma tal que o estimador da eq.(3.9) acompanha o mais próximo possível o estimador da eq.(3.8).

Fazendo uma analogia entre a figura 2.4 e a figura 4.1, verifica-se que o sistema (P) é representado pela MI e a entrada u pelas tensões e correntes de estator. A saída do sistema (P), y^p , é dada pela equação do fluxo do rotor, eq.(3.8), e a saída do modelo da rede neural, y^m , é dada pela eq.(3.9).

Para obter o algoritmo de retropropagação, o modelo dos padrões de dados da eq.(3.9) é primeiramente derivado utilizando o método a diferença *backward*.

Como a taxa de variação instantânea de λ_r em relação a T no instante em que $T = k$, é dada por $\lim_{\Delta T \rightarrow 0} \frac{\Delta \lambda_r}{\Delta T} = \lim_{\Delta T \rightarrow 0} \left(\frac{\bar{\lambda}_r(k) - \bar{\lambda}_r(k-1)}{T} \right)$,
então,

Figura 4.1: Estrutura do Sistema para Estimar ω_r .

$$\dot{\vec{\lambda}}_{2r}(k) = \frac{\vec{\lambda}_{2r}(k) - \vec{\lambda}_{2r}(k-1)}{T}. \quad (4.1)$$

Substituindo $\dot{\vec{\lambda}}_{2r}(k)$ da eq.(4.1) pela eq.(3.9), segue que

$$\frac{\vec{\lambda}_{2r}(k) - \vec{\lambda}_{2r}(k-1)}{T} = \left(\frac{-1}{T_r}I + \omega_r J\right)\vec{\lambda}_{2r}(k-1) + \frac{L_m}{T_r}\vec{i}_s(k-1),$$

então,

$$\vec{\lambda}_{2r}(k) = I\vec{\lambda}_{2r}(k-1) + \left(\frac{-1}{T_r}I + \omega_r J\right)T\vec{\lambda}_{2r}(k-1) + \frac{L_m}{T_r}T\vec{i}_s(k-1). \quad (4.2)$$

Organizando os termos em relação às matrizes \mathbf{I} , \mathbf{J} e as variáveis $\vec{\lambda}_{2r}$ e \vec{i}_s vem,

$$\vec{\lambda}_{2r}(k) = \left(1 - \frac{T}{T_r}\right)I\vec{\lambda}_{2r}(k-1) + \omega_r T J \vec{\lambda}_{2r}(k-1) + \frac{L_m}{T_r}T\vec{i}_s(k-1). \quad (4.3)$$

A eq.(4.3) pode ser escrita da seguinte forma:

$$\vec{\lambda}_{2r} = W_1 X_1 + W_2 X_2 + W_3 X_3, \quad (4.4)$$

ou ainda

$$\vec{\lambda}_{2r} = \sum_{i=1}^3 W_i X_i, \quad (4.5)$$

sendo

$$W_1 = 1 - \frac{T}{T_r},$$

$$X_1 = I \vec{\lambda}_{2r} (k - 1),$$

$$W_2 = \omega_r T,$$

$$X_2 = J \vec{\lambda}_{2r} (k - 1),$$

$$W_3 = \frac{L_m}{T_r T},$$

$$X_3 = I \vec{i}_s (k - 1),$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

e T é o período de amostragem.

A eq.(4.4) pode ser representada como um modelo neural, com duas camadas, sendo que W_1 , W_2 e W_3 representam os pesos da RNA, X_1 , X_2 e X_3 , as entradas e $\vec{\lambda}_{2r}$ a saída, conforme a figura 4.2, (Ben-Brahim, 1995).

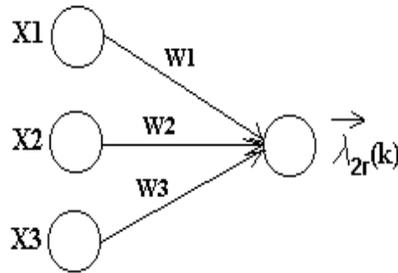


Figura 4.2: RNA de Duas Camadas

O erro de saída entre o fluxo desejado e a saída da planta é dado por:

$$\vec{\varepsilon}(k) = \vec{\lambda}_{1r}(k) - \vec{\lambda}_{2r}(k). \quad (4.6)$$

Os pesos W_1 e W_3 da RNA são considerados constantes dependendo somente dos parâmetros do motor e o peso W_2 dependendo da velocidade do motor, (El-loumi *et al.*, 1998).

Os pesos sinápticos (W_1, W_2, W_3) são sintonizados para minimizar a função de energia, (Haykin, 2001), (Elloumi *et al.*, 1998). O valor instantâneo da energia do erro para esses neurônios é definida como, (Elloumi *et al.*, 1998),

$$E = \frac{1}{2} \varepsilon^2(k), \quad (4.7)$$

ou seja,

$$E = \frac{1}{2} \varepsilon' \varepsilon = \frac{1}{2} \|\vec{\lambda}_{1r}(k) - \vec{\lambda}_{2r}(k)\|^2 \quad (4.8)$$

Dentre os pesos da rede neural somente W_2 varia.

A correção $\Delta W_2(k)$ aplicada pelo algoritmo de retropropagação ao peso sináptico W_2 é dada por, (Haykin, 2001):

$$\Delta W_2(k) \propto -\frac{\partial E}{\partial W_2}. \quad (4.9)$$

Pela regra da cadeia, o gradiente $\frac{\partial E}{\partial W_2}$ pode ser expresso como, (Haykin, 2001):

$$\frac{\partial E}{\partial W_2} = \frac{\partial E}{\partial \varepsilon} \frac{\partial \varepsilon}{\partial X_2} \frac{\partial X_2}{\partial \lambda_{2r}} \frac{\partial \lambda_{2r}}{\partial W_2}, \quad (4.10)$$

sendo $\frac{\partial E}{\partial W_2}$, o fator de sensibilidade que determina a direção de busca no espaço de pesos para o peso sináptico W_2 , (Haykin, 2001).

Diferenciando-se ambos os lados da eq.(4.7) em relação a ε , tem-se

$$\frac{\partial E}{\partial \varepsilon} = \vec{\varepsilon}. \quad (4.11)$$

Substituindo-se $\vec{\lambda}_{2r}$ da eq. $X_2 = J\vec{\lambda}_{2r}(k-1)$ na eq.(4.6) e diferenciando-se ambos os lados dessa equação em relação a X_2 , tem-se

$$\frac{\partial \varepsilon}{\partial X_2} = -1. \quad (4.12)$$

Se $X_2 = J\vec{\lambda}_{2r}(k-1)$, pode-se dizer, então, que

$$X_2 = f(\vec{\lambda}_{2r}). \quad (4.13)$$

Logo, diferenciando-se a eq.(4.13) em relação a $\vec{\lambda}_{2r}$, tem-se

$$\frac{\partial X_2}{\partial \vec{\lambda}_{2r}} = f'(\vec{\lambda}_{2r}). \quad (4.14)$$

Diferenciando-se a eq.(4.4) em relação a W_2 , produz-se

$$\frac{\partial \lambda_{2r}}{\partial W_2} = X_2. \quad (4.15)$$

Substituindo-se as eqs.(4.11), (4.12) e (4.14) na eq.(4.10), tem-se

$$\frac{\partial E}{\partial W_2} = \varepsilon(-1)f'(\vec{\lambda}_{2r})X_2. \quad (4.16)$$

A correção $\Delta W_2(k)$ aplicada a W_2 , definida como Regra Delta, é dada por:

$$\Delta W_2(k) = -\eta \frac{\partial E}{\partial \vec{\lambda}_{2r}}, \quad (4.17)$$

no qual η é o parâmetro da taxa de aprendizagem do algoritmo de retro-propagação. O sinal negativo indica descida do gradiente no espaço de pesos de modo a encontrar uma direção para a mudança de peso a fim de reduzir o valor do erro, ε , (Haykin, 2001) .

Substituindo-se a eq.(4.16) em (4.17), tem-se

$$\Delta W_2(k) = -\eta \varepsilon(-1)f'(\vec{\lambda}_{2r})X_2, \quad (4.18)$$

ou seja,

$$\Delta W_2(k) = -\eta \delta(k)X_2, \quad (4.19)$$

sendo $\delta(k)$ o gradiente local, dado por

$$\delta(k) = -\frac{\partial E}{\partial \vec{\lambda}_{2r}}. \quad (4.20)$$

Diferenciando-se a eq.(4.8) em relação a $\vec{\lambda}_{2r}$, segue que

$$\frac{\partial E}{\partial \vec{\lambda}_{2r}} = \frac{1}{2} \frac{\partial [(\vec{\lambda}_{1r}(k) - \vec{\lambda}_{2r}(k))' (\vec{\lambda}_{1r}(k) - \vec{\lambda}_{2r}(k))]}{\partial \vec{\lambda}_{2r}(k)},$$

$$\frac{\partial E}{\partial \vec{\lambda}_{2r}} = \frac{1}{2} \frac{\partial [\vec{\lambda}_{1r}(k)' \vec{\lambda}_{1r}(k) - \vec{\lambda}_{1r}(k)' \vec{\lambda}_{2r}(k) - \vec{\lambda}_{2r}(k)' \vec{\lambda}_{1r}(k) + \vec{\lambda}_{2r}(k)' \vec{\lambda}_{2r}(k)]}{\partial \vec{\lambda}_{2r}(k)}. \quad (4.21)$$

Como

$$\vec{\lambda}_{1r}(k)' \vec{\lambda}_{2r}(k) = \vec{\lambda}_{2r}(k)' \vec{\lambda}_{1r}(k),$$

então,

$$\frac{\partial E}{\partial \vec{\lambda}_{2r}} = \frac{1}{2} \frac{\partial [(\vec{\lambda}_{1r}(k)^2)' - (2\vec{\lambda}_{1r}(k))' \vec{\lambda}_{2r}(k) + (\vec{\lambda}_{2r}(k)^2)']}{\partial \vec{\lambda}_{2r}(k)}. \quad (4.22)$$

Derivando,

$$\frac{\partial E}{\partial \vec{\lambda}_{2r}} = \frac{1}{2} (-2\vec{\lambda}_{1r}(k)' + 2\vec{\lambda}_{2r}(k)')$$

$$\frac{\partial E}{\partial \vec{\lambda}_{2r}} = (-\vec{\lambda}_{1r}(k)' + \vec{\lambda}_{2r}(k)'). \quad (4.23)$$

Substituindo-se a eq.(4.23) em (4.20), segue que

$$\delta(k) = (\vec{\lambda}_{2r} - \vec{\lambda}_{1r})'. \quad (4.24)$$

Substituindo-se a eq.(4.20) e $X_2 = J\vec{\lambda}_{2r}(k-1)$ na eq.(4.19), tem-se

$$\Delta W_2 = -\eta(\vec{\lambda}_{2r}(k) - \vec{\lambda}_{1r}(k))' J\vec{\lambda}_{2r}(k-1). \quad (4.25)$$

Da Regra Delta, conforme fig.2.3, tem-se que o novo peso será

$$W_2(k) = W_2(k-1) + \eta \Delta W_2(k), \quad (4.26)$$

sendo η , o coeficiente de treinamento e o coeficiente k é incrementado de 1 para cada varredura através do conjunto de entrada-saída, (Elloumi *et al.*, 1998).

A Regra Delta da eq.(4.19) é modificada de maneira a aumentar a taxa de aprendizado sem conduzir a oscilações, incluindo-se, assim, um termo de momento, conforme a eq.(4.27), (Haykin, 2001).

$$\Delta W_2(k) = -\eta\delta(k)X_2 + \alpha\Delta W_2(k-1). \quad (4.27)$$

O coeficiente α , constante de momento, determina o efeito das modificações dos pesos anteriores no peso atual. É preferível, portanto, utilizar a eq.(4.27) ao invés da eq.(4.26).

Sabendo-se que $W_2 = \omega_r T$, então $\omega_r = \frac{W_2}{T}$, variando-se ω_r , vem

$$\Delta\omega_r = \frac{\Delta W_2}{T}. \quad (4.28)$$

E, finalmente, substituindo-se a eq.(4.27) na eq.(4.28), produz-se a velocidade estimada do rotor que é dada por

$$\Delta\omega_r = -\frac{1}{T}\eta\delta(k)X_2 + \frac{1}{T}\alpha\Delta W_2(k-1), \quad (4.29)$$

ou seja,

$$\hat{\omega}_r(k) - \hat{\omega}_r(k-1) = -\frac{1}{T}\eta\delta(k)X_2 + \frac{1}{T}\alpha\Delta W_2(k-1), \quad (4.30)$$

então, a velocidade estimada é, (Elloumi *et al.*, 1998),

$$\hat{\omega}_r(k) = \hat{\omega}_r(k-1) - \frac{1}{T}\eta\delta(k)X_2 + \frac{1}{T}\alpha\Delta W_2(k-1). \quad (4.31)$$

Um dos componentes do vetor peso W da eq.(2.2) fornece a expressão da velocidade estimada ω_r da eq.(4.31) que é ajustada através do algoritmo de aprendizagem - o de retropropagação.

Análise Comparativa entre Estimadores- ω_r

5.1 Introdução

Neste capítulo apresenta-se uma análise do desempenho dos modelos de estimadores de velocidade. Os resultados são obtidos a partir desses modelos da máquina de indução, Apêndice A, e dos estimadores vistos nos Capítulos 3 e 4, respectivamente. Utiliza-se a plataforma computacional MATLAB/SIMULINK para a implementação dos modelos. O bloco para o modelo do motor é abordado no Apêndice B. O termo estimadores- ω_r diz respeito à velocidade estimada do rotor.

Também apresenta-se neste capítulo algoritmos para estimar a velocidade que foram desenvolvidos a partir dos modelos dos estimadores via fluxo, FEM, MRAS e RNA, Capítulos 3 e 4. Estes algoritmos visam implementação em tempo real, necessitando-se de análises mais minuciosas quanto ao tempo de resposta e à precisão dos estimadores.

5.2 Parâmetros da Máquina

Os parâmetros da máquina são: trifásica, quatro pólos, 220 V, frequência 60 Hz, $R_s = 3,35 \Omega$, $R_r = 1,99 \Omega$, $L_{ls} = L_{lr} = 6,94 \text{ mH}$, $L_m = 163,73 \text{ mH}$ e $J_m = 0,1 \text{ kgm}^2$.

5.3 Estimador via Fluxo

O estimador via fluxo produz a velocidade da máquina de indução tendo por base as eqs.(3.8), (3.15), (3.18) e (3.19). O fluxo é calculado pela eq.(3.8) e a partir desta, obtém-se a velocidade ω_s utilizando a eq.(3.15). Em seguida calcula-se a velocidade ω_{ls} usando a eq.(3.18). De posse desses dois valores produz-se o valor estimado da velocidade ω_r que é dado pela diferença entre o valor da velocidade síncrona ω_s e o valor da velocidade de escorregamento ω_{ls} , ou seja, $\omega_s - \omega_{ls}$. A seguir descreve-se o algoritmo para calcular a velocidade estimada ω_r .

ALGORITMO Velocidade Estimada via Fluxo

VARIÁVEIS:

$T_r, L_r, L_m, \sigma, L_s, R_s$: real;

$\omega_s, \omega_r, \omega_{ls}, \lambda_r, v_s, i_s$: vetor

INÍCIO da Estimação via Fluxo

LER L_r, L_m, R_s, σ

FAZER $i \leftarrow 0$ até ∞

$$\lambda_{1r} \leftarrow \int [(L_r/L_m)(v_s - R_s i_s - \sigma L_s \frac{di_s}{dt})]$$

$$\omega_s \leftarrow \frac{\lambda_{1dr} \lambda_{1qr} - \lambda_{1qr} \lambda_{1dr}}{\lambda_{1dr}^2 + \lambda_{1qr}^2}$$

$$\omega_{ls} \leftarrow \frac{L_m}{T_r} \left(\frac{\lambda_{1dr} i_{qs} - \lambda_{1qr} i_{ds}}{|\lambda_{1r}|^2} \right)$$

$$\omega_r \leftarrow \omega_s - \omega_{ls}$$

FIM FAZER

FIM da Estimação via Fluxo

Na figura 5.1 tem-se a velocidade estimada ω_r . Verifica-se que a velocidade cresce de forma linear, atinge um valor máximo de 387 rad/s num instante próximo

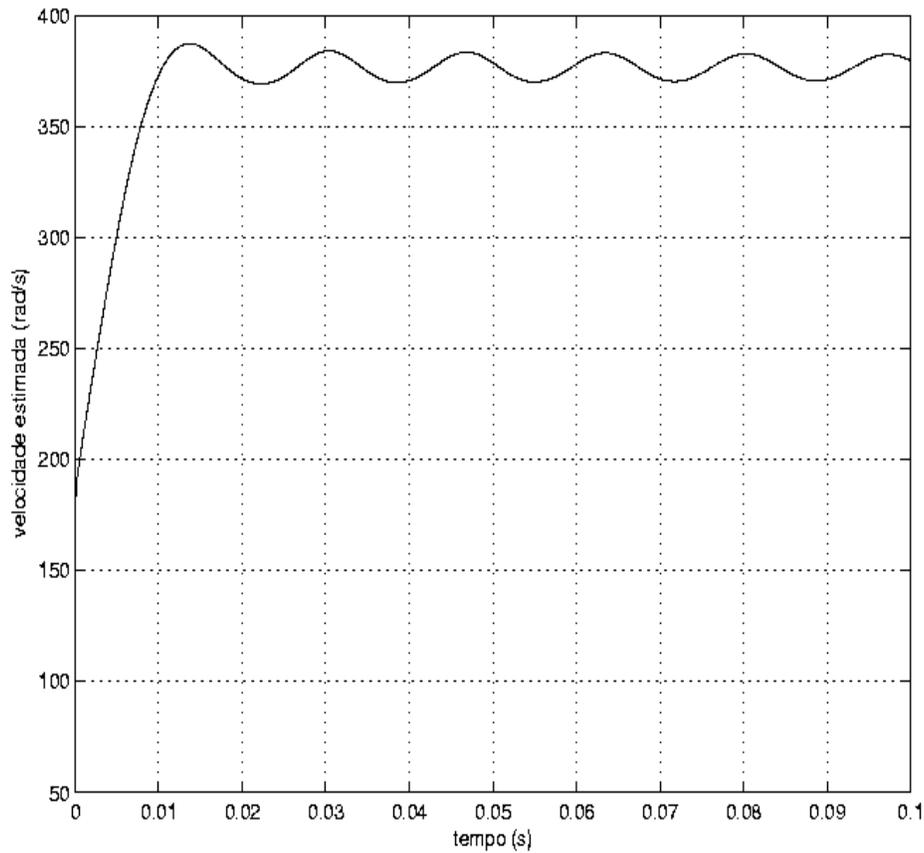


Figura 5.1: Velocidade estimada via fluxo do Rotor

de $t = 10$ ms e em seguida apresenta oscilações que vão diminuindo o erro de regime, tendendo para o valor especificado de referência.

5.4 Estimador via FEM

Neste algoritmo, o estimador para ω_r utiliza as seguintes equações: (3.8), (3.15), (3.27) e (3.19). Segue o mesmo procedimento do item anterior, porém as equações são apresentadas em função da força eletromotriz e_r . A eq.(3.8) é a mesma para a estimação por fluxo, porém sem a integração. Novamente o fluxo é calculado

para posteriormente ser utilizado no cálculo da velocidade ω_s e o valor da força eletromotriz para o cálculo de ω_{ls} . De posse desses dois valores produz-se o valor estimado da velocidade ω_r que é dado pela diferença entre a velocidade síncrona ω_s e a velocidade de escorregamento ω_{ls} , ou seja, $\omega_s - \omega_{ls}$. A seguir descreve-se o algoritmo para obtenção da velocidade estimada via FEM.

ALGORITMO Velocidade Estimada via FEM

VARIÁVEIS:

$T_r, L_r, L_m, \sigma, L_s, R_s$: real;

$\omega_s, \omega_r, \omega_{ls}, \lambda_{1r}, e_r, v_s, i_s$: vetor

INÍCIO da Estimação via FEM

LER L_r, L_m, R_s, σ

FAZER $i \leftarrow 0$ até ∞

$$e_r \leftarrow [(L_r/L_m)(v_s - R_s i_s - \sigma L_s \frac{di_s}{dt})]$$

$$\lambda_{1r} \leftarrow \int e_r$$

$$\omega_s \leftarrow \frac{\lambda_{1dr} \lambda_{1qr} - \lambda_{1qr} \lambda_{1dr}}{\lambda_{1dr}^2 + \lambda_{1qr}^2}$$

$$\omega_{ls} \leftarrow \frac{L_m}{T_r} \omega_s \left(\frac{e_{qr} \cdot i_{qs} - e_{dr} \cdot i_{ds}}{e_{dr}^2 + e_{qr}^2} \right)$$

$$\omega_r \leftarrow \omega_s - \omega_{ls}$$

FIM FAZER

FIM da Estimação via FEM

A figura 5.2 apresenta a velocidade estimada ω_r e pode-se verificar que há o registro de oscilações logo no início em um curto intervalo. Em seguida tem-se o crescimento de forma linear com valor máximo de 390 rad/s, neste caso, próximo de $t = 16$ ms e algumas oscilações que vão diminuindo gradativamente convergindo para o valor de referência.

5.5 Estimador via Fluxo utilizando MRAS

Neste algoritmo, o estimador via Fluxo utilizando MRAS utiliza as seguintes equações: (3.8), (3.9), (3.28) e (3.30). A eq.(3.8) é usada para gerar o fluxo de

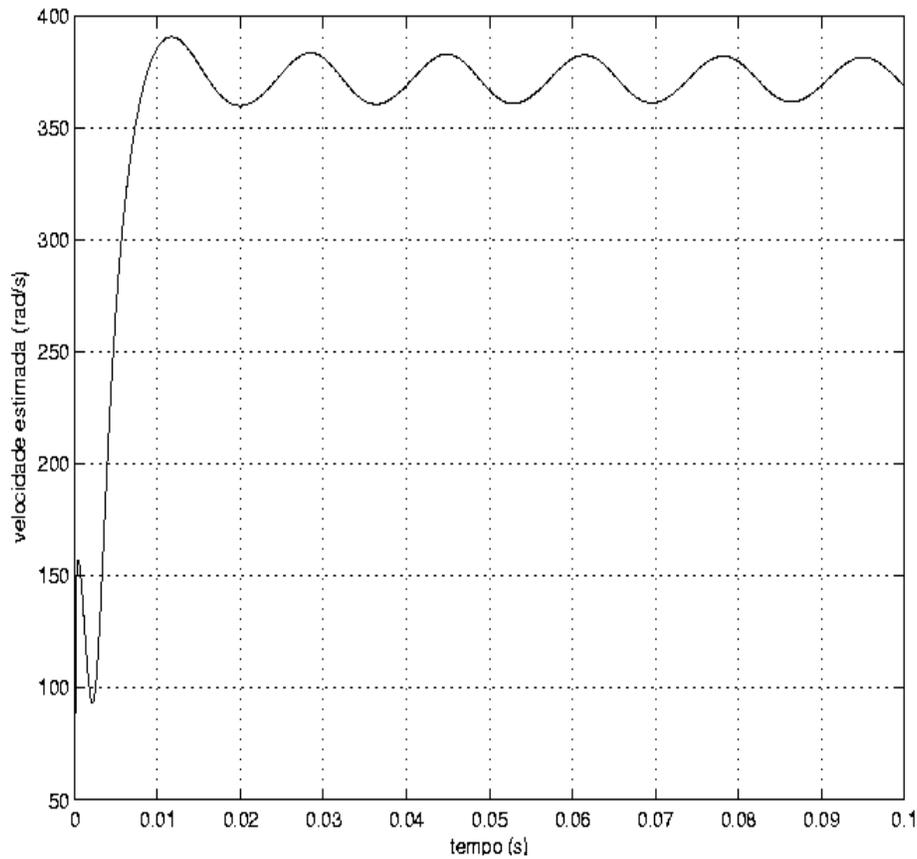


Figura 5.2: Velocidade estimada via FEM do Rotor

referência e a eq.(3.9) para fornecer o fluxo estimado. O mecanismo de adaptação que gera a velocidade estimada ω_r é dado pela eq.(3.9) para ajustar o fluxo estimado até que o erro ε tenda para 0. A descrição do algoritmo para calcular a velocidade estimada via fluxo utilizando MRAS é mostrada a seguir .

ALGORITMO Velocidade Estimada via fluxo utilizando MRAS

VARIÁVEIS:

$T_r, L_r, L_m, \sigma, L_s, R_s, k_p, k_i$: real;

$\omega_r, \lambda_{2r}, \varepsilon, \lambda_{1r}, v_s, i_s$: vetor

INÍCIO da Estimação via Fluxo utilizando MRAS

LER L_r, L_m, R_s, σ

FAZER $i \leftarrow$ de 0 até ∞

$$\lambda_{1r} \leftarrow \int [(L_r/L_m)(v_s - R_s i_s - \sigma L_s \frac{di_s}{dt})]$$

$$\omega_r \leftarrow 0$$

$$\lambda_{2r} \leftarrow \int [(L_m/T_r)i_s + (-\frac{1}{T_r} + \omega_r)\lambda_{2r}]$$

$$\varepsilon \leftarrow \lambda_{1r} - \lambda_{2r}$$

ENQUANTO $\varepsilon \neq 0$ faça

$$\omega_r \leftarrow k_p(\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr}) + \frac{k_i}{p}(\lambda_{1qr}\hat{\lambda}_{2dr} - \lambda_{1dr}\hat{\lambda}_{2qr})$$

$$\lambda_{2r} \leftarrow \int [(L_m/T_r)i_s + (-\frac{1}{T_r} + \omega_r)\lambda_{2r}]$$

$$\varepsilon \leftarrow (\lambda_{1r} - \lambda_{2r})$$

FIM ENQUANTO

FIM FAZER

FIM da Estimação via Fluxo utilizando MRAS

A velocidade estimada obtida para este estimador, figura 5.3, atinge o valor máximo de 450 rad/s em $t = 10$ ms, cai suavemente e próximo do instante $t = 20$ ms apresenta oscilações que vão diminuindo gradativamente tendendo a convergir para a velocidade de referência.

5.6 Estimador via RNA

O algoritmo para estimar a velocidade rotórica ω_r via RNA tem por base as seguintes equações: (3.8), (3.9), (4.24), (4.27) e (4.31). Utiliza-se também a expressão $X_2 = J\vec{\lambda}_r(k-1)$. Inicialmente calcula-se o valor do fluxo a partir da eq.(3.8), em seguida o valor do fluxo pertencente ao modelo neural dado pela eq.(3.9). Calcula-se a diferença entre esses dois fluxos, ou seja, o gradiente δ dado pela eq.(4.24). Enquanto esse gradiente estiver distante de zero, calcula-se a velocidade estimada a partir do algoritmo de retropropagação, a fim de que o fluxo estimado seja ajustado, verificando em seguida se o gradiente já apontou para o ponto de mínimo, ou seja, se o gradiente já tende para zero. O algoritmo

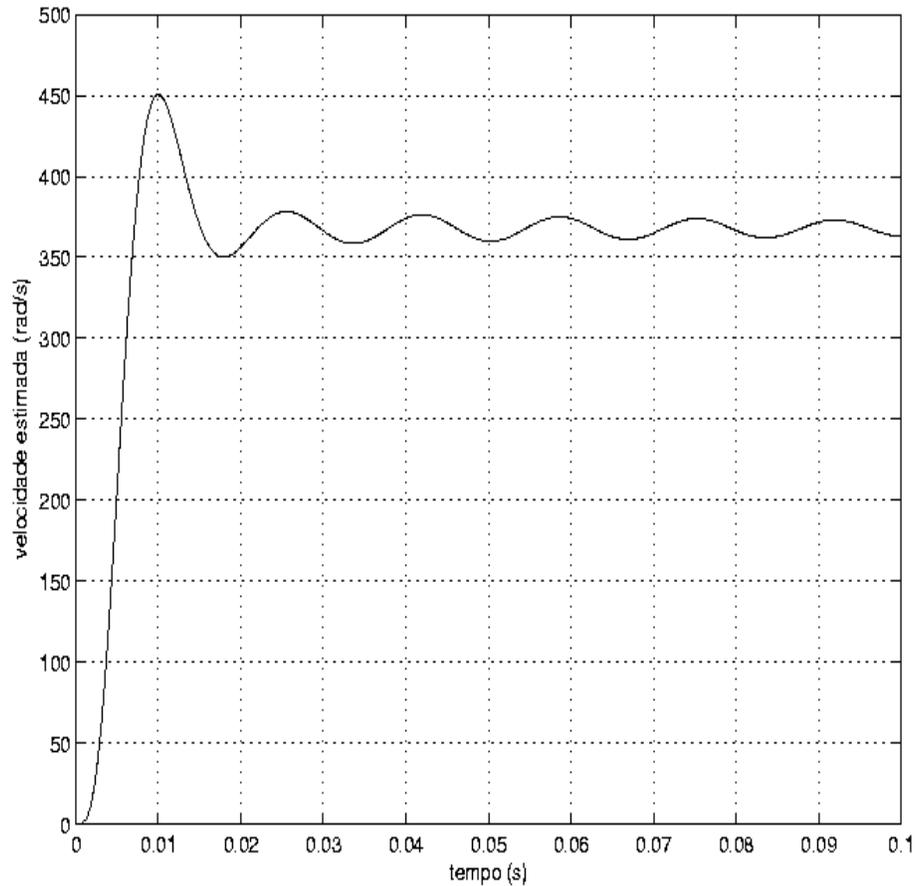


Figura 5.3: Velocidade estimada via fluxo utilizando MRAS

para esse estimador de velocidade é descrito a seguir.

ALGORITMO Velocidade Estimada via RNA

VARIÁVEIS:

$T_r, L_r, L_m, \sigma, L_s, R_s, \alpha, \eta, T$: real;

$\omega_r, \lambda_{1r}, \lambda_{2r}, \delta, \Delta W_2(k-1), v_s, i_s, X_2, I, J$: vetor

INÍCIO da Estimação via RNA

LER L_r, L_m, R_s, σ

FAZER $i \leftarrow 0$ até ∞

$$\lambda_{r1} \leftarrow \int [(L_r/L_m)(v_s - R_s i_s - \sigma L_s \frac{di_s}{dt})]$$

$$\omega_r \leftarrow 0$$

$$\lambda_{r2} \leftarrow \int [-(\frac{1}{T_r})\lambda_{2r} + \omega_r \lambda_{2r} + (\frac{L_m}{T_r})i_s]$$

$$\delta \leftarrow \lambda_{2r} - \lambda_{1r}$$

ENQUANTO $d \neq 0$ faça

$$X_2 \leftarrow J\lambda_{2r}$$

$$\omega_r \leftarrow \omega_r - (\frac{1}{T_r})\eta\delta X_2 + (\frac{1}{T_r})\alpha\Delta W_2(k-1)$$

$$\lambda_{r2} \leftarrow \int [-(\frac{1}{T_r})\lambda_{2r} + \omega_r \lambda_{2r} + (\frac{L_m}{T_r})i_s]$$

$$\delta \leftarrow \lambda_{2r} - \lambda_{1r}$$

FIM ENQUANTO

FIM FAZER

FIM da Estimação via RNA

A tabela 5.1 apresenta a variação dos parâmetros da RNA α e η utilizados para obter a velocidade estimada.

Tabela 5.1: Variação dos parâmetros α e η para estimar a velocidade via RNA's

α	η	figura
0,5	0,5	5.4
0,9	0,9	5.6
0,9	0,5	5.8
0,9	0,2	5.10
0,9	0,0	5.12
0,99	0,01	5.14
0,0	0,1	5.16

A velocidade estimada ω_r na figura 5.4 cresce quase linearmente atingindo um valor máximo um pouco acima de 3000 rad/s no espaço de tempo entre 5 ms e 90 ms e logo em seguida apresenta pouca oscilação e vai se estabilizando em torno de 1500 rad/s, distanciando-se bastante do valor de referência. De modo semelhante

ocorre com a velocidade estimada mostrada na figura 5.6, porém nesta o valor máximo já diminui e fica entre 2000 e 2500 rad/s para o mesmo intervalo de tempo mencionado anteriormente. Apresenta poucas oscilações estabilizando-se em torno de 1250 rad/s. A velocidade estimada da figura 5.8 tende a se estabilizar em torno de 1000 rad/s e inicialmente chega a atingir próximo de 2000 rad/s para o mesmo intervalo de tempo da figura 5.4. Na velocidade estimada mostrada na figura 5.10, verifica-se que esta tende a se estabilizar em torno de 600 rad/s e seu valor máximo no início do tempo não chega a 1600 rad/s.

Os resultados da velocidade estimada apresentada nas figura 5.12 e figura 5.14 indicam que há uma diminuição do valor no início do tempo, porém não atinge o valor de referência, ficando em torno de 700 rad/s.

Observa-se que a velocidade estimada para os parâmetros da rede neural, $\alpha = 0.0$ e $\eta = 0.1$, atinge um valor máximo próximo de 430 rad/s no intervalo de tempo inferior a 20 ms em seguida apresenta oscilação que vai diminuindo tendendo a convergir para o valor de referência, figura 5.16. Considera-se $T = 0,2$ ms.

Tabela 5.2: Resultados da velocidade estimada via RNA para variações de α e η - valor de pico (V_p), tempo de estabilização (t_e) e Erro.

Estimador	Parâmetros α / η	V_p (rad/s)	t_e (ms)	Erro (%)
Neural	0,5 / 0,5	3000	20	297
	0,9 / 0,9	2300	20	245
	0,9 / 0,5	2000	20	169
	0,9 / 0,2	1500	20	73
	0,9 / 0,0	1300	20	86
	0,99 / 0,01	1250	20	85
	0,0 / 0,1	430	20	8

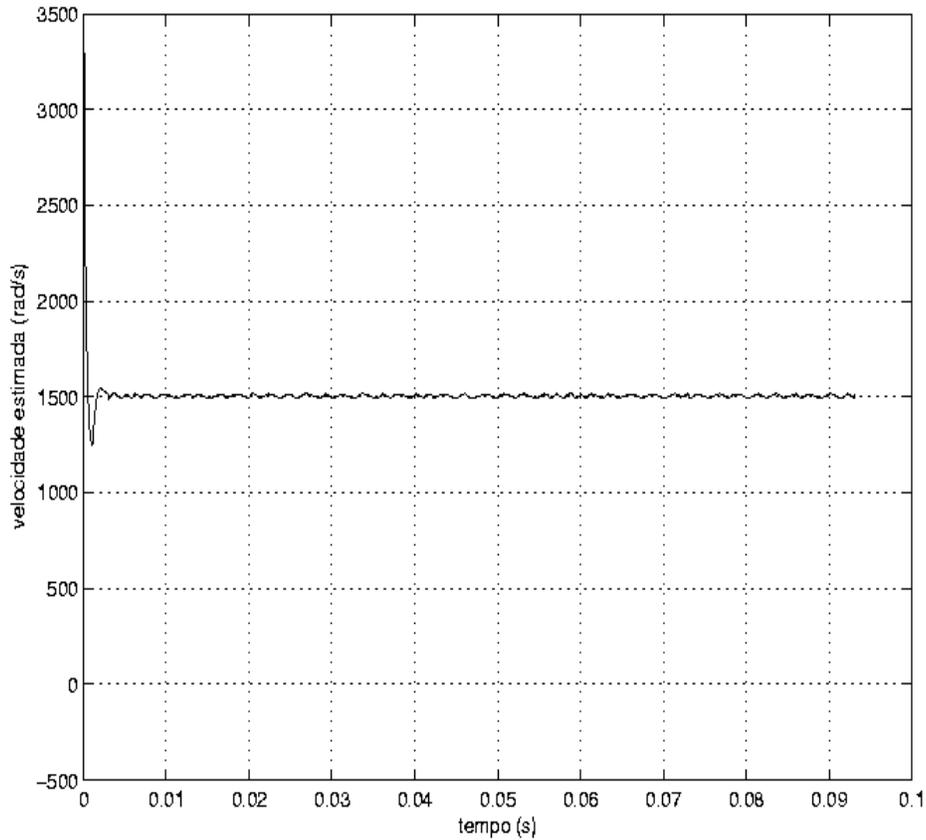


Figura 5.4: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,5$ e $\eta=0,5$

5.7 Comentários

Os algoritmos apresentados são baseados nas equações da máquina na referência dq0. As equações dos fluxos rotóricos são obtidas a partir das medições das correntes e tensões estatóricas.

Observa-se em todas as técnicas oscilações no início do tempo das formas de onda. As velocidades estimadas pelo fluxo, pela FEM e pelo fluxo utilizando MRAS apresentam algumas oscilações até se aproximarem do valor de referência. A velocidade estimada via RNA's é apresentada nas figuras 5.4, 5.6, 5.8, 5.10, 5.12 e 5.14, porém somente para os parâmetros da RNA, $\alpha = 0,0$, $\eta = 0,1$, obteve-se o

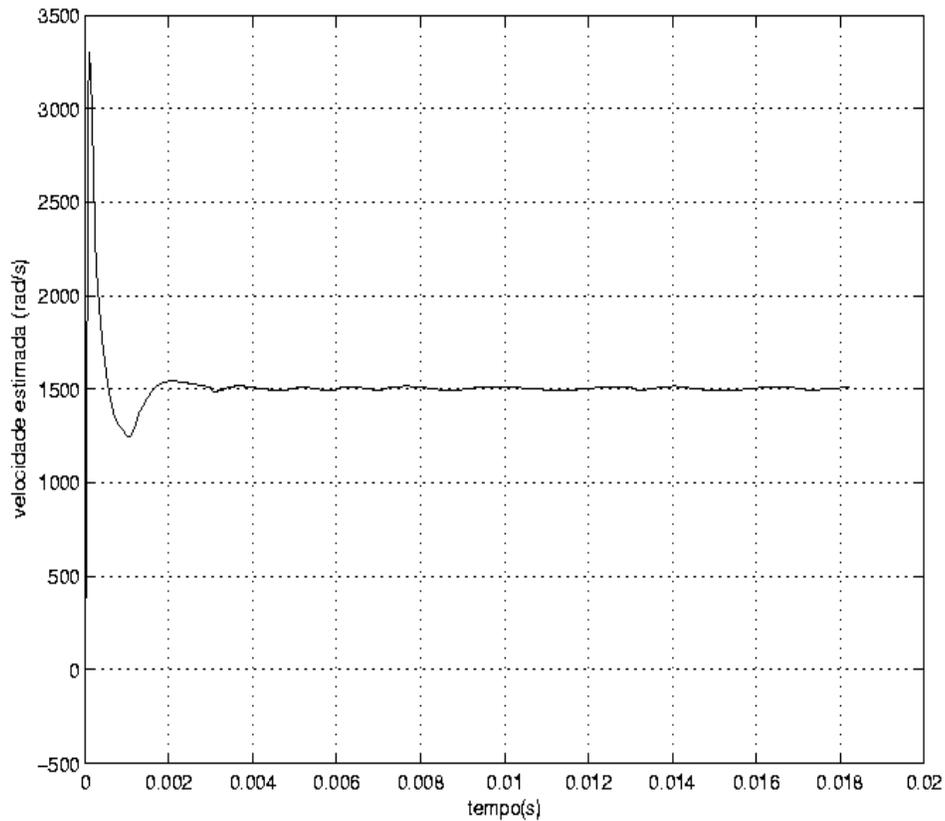


Figura 5.5: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,5$ e $\eta=0,5$

valor estimado da velocidade convergindo para o valor da velocidade de referência, figura 5.16, num tempo aceitável $t = 60$ ms. As figuras 5.4, 5.6, 5.8, 5.10, 5.12 e 5.14, são mostradas também pelas figuras 5.5, 5.7, 5.9, 5.11, 5.13 e 5.15, em um tempo menor de modo a verificar mais detalhadamente o comportamento da velocidade estimada nos primeiros instantes.

O resultado da velocidade estimada ω_r através do fluxo com e sem MRAS e da FEM apresenta convergência para o valor da velocidade de referência. O fluxo com MRAS apresenta um valor máximo maior que o da FEM e o do fluxo. Estas três técnicas apresentam uma boa aproximação da velocidade estimada, erro médio de

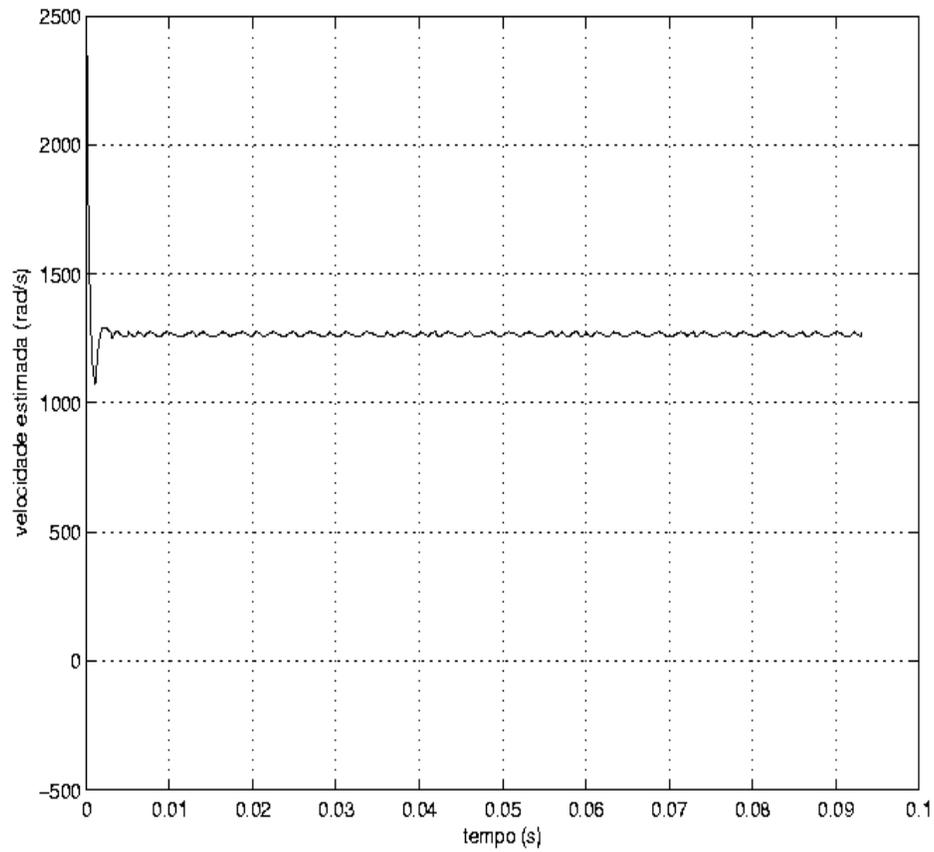


Figura 5.6: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,9$

3.6% com um tempo de estabilização $t = 60$ ms.

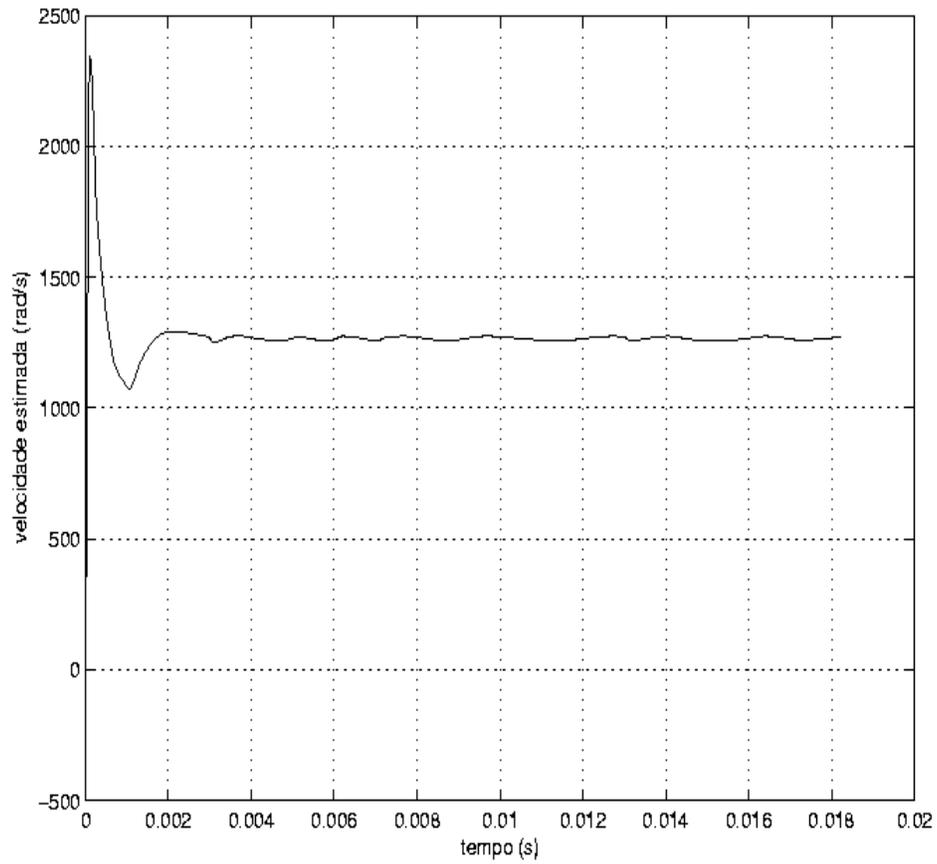


Figura 5.7: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,9$

Tabela 5.3: Resultados da velocidade estimada via fluxo e FEM - valor de pico (V_p), tempo de estabilização (t_e) e Erro.

Estimador	MRAS	V_p (rad/s)	t_e (ms)	Erro (%)
FLUXO	Sem	387	60	3,3
	Com	450	60	4,0
FEM	Sem	390	60	3,5

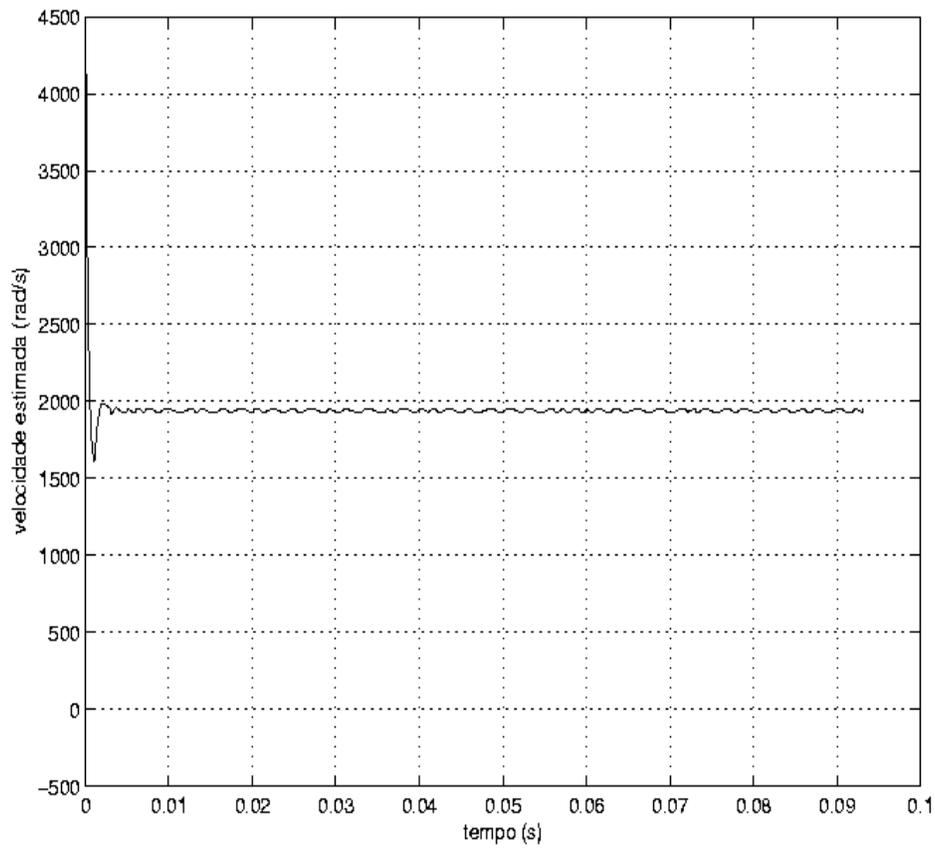


Figura 5.8: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,5$

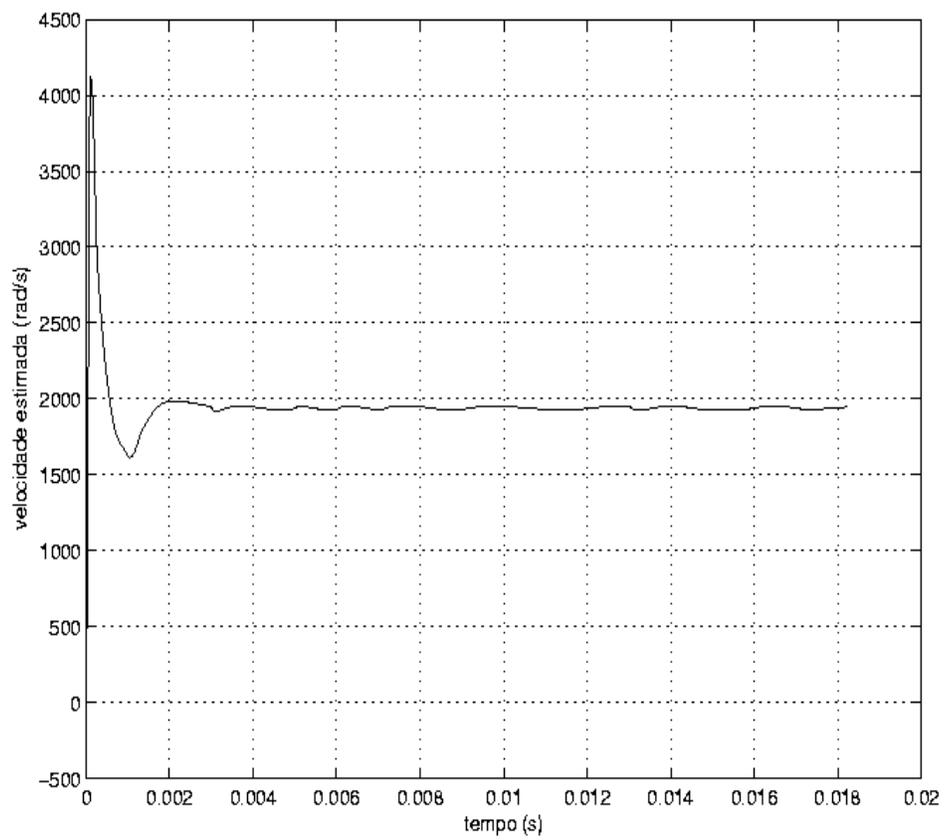
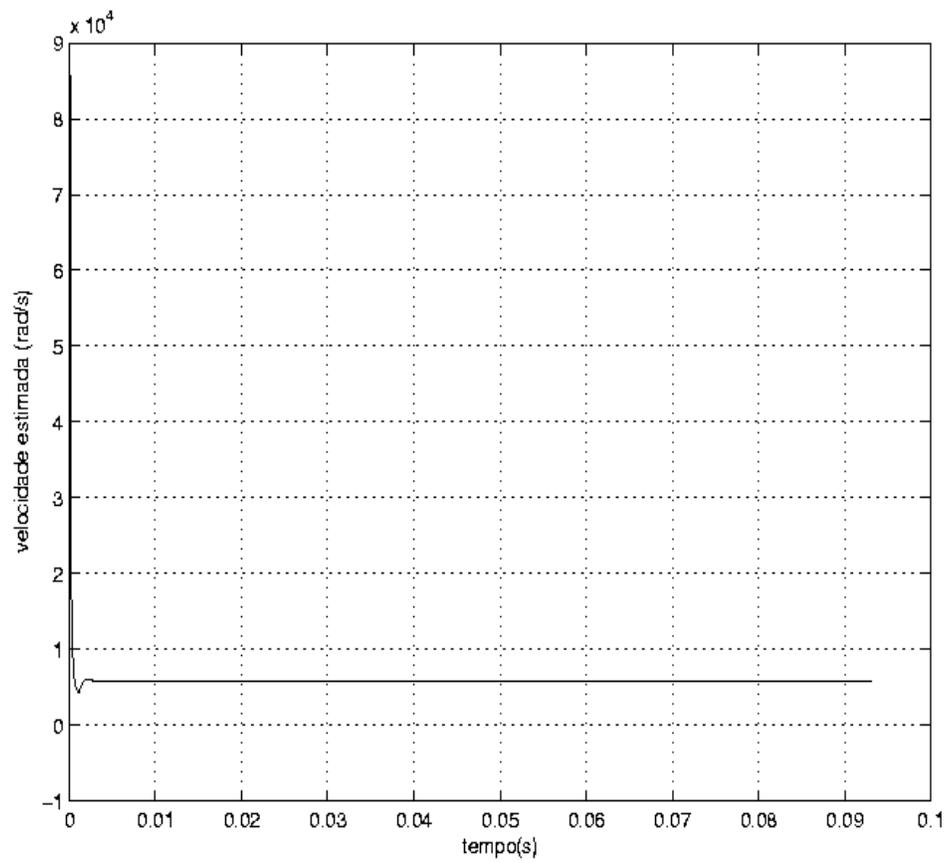


Figura 5.9: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,5$

Figura 5.10: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,2$

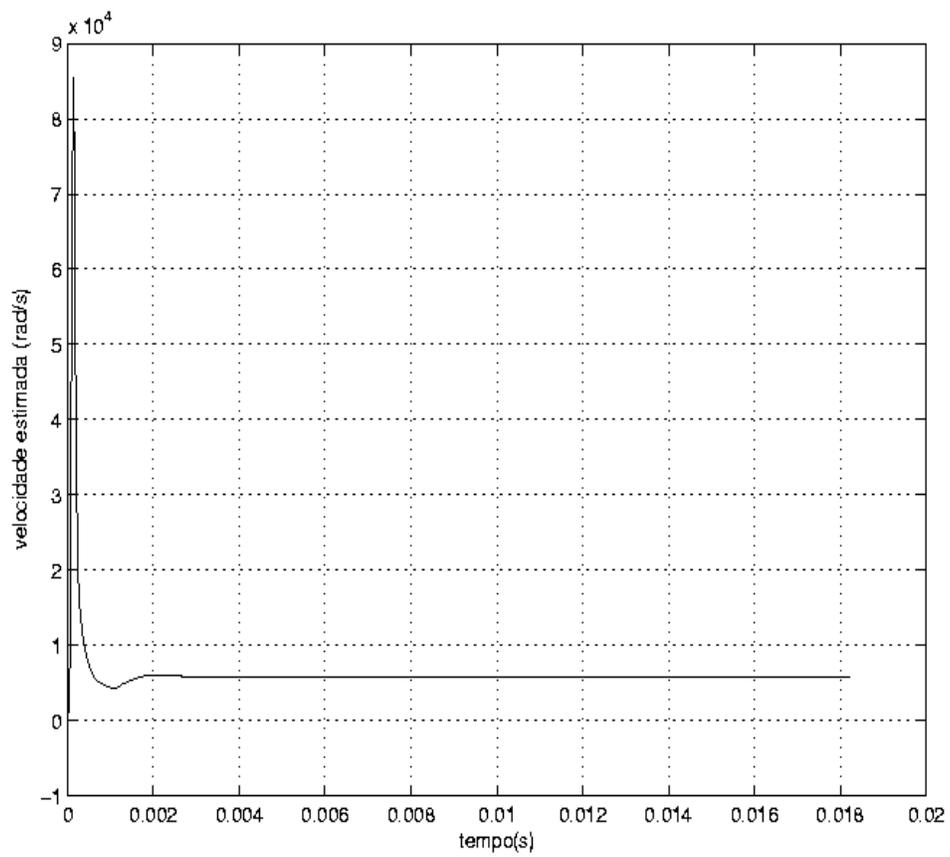


Figura 5.11: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,2$

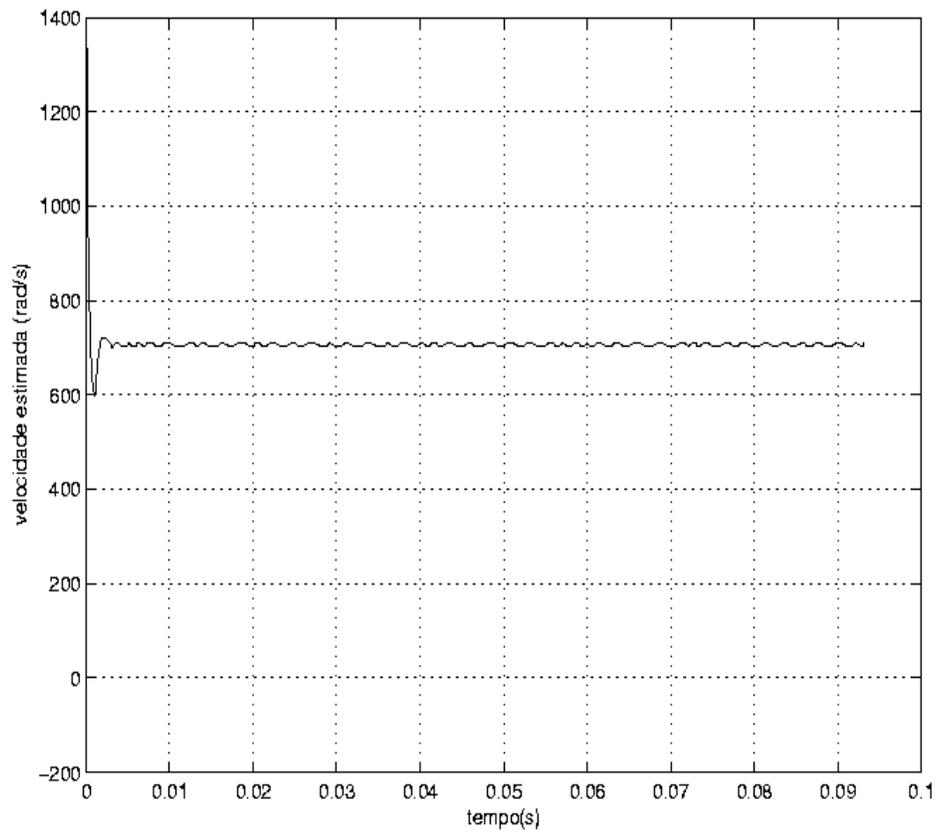


Figura 5.12: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,9$ e $\eta=0,00$

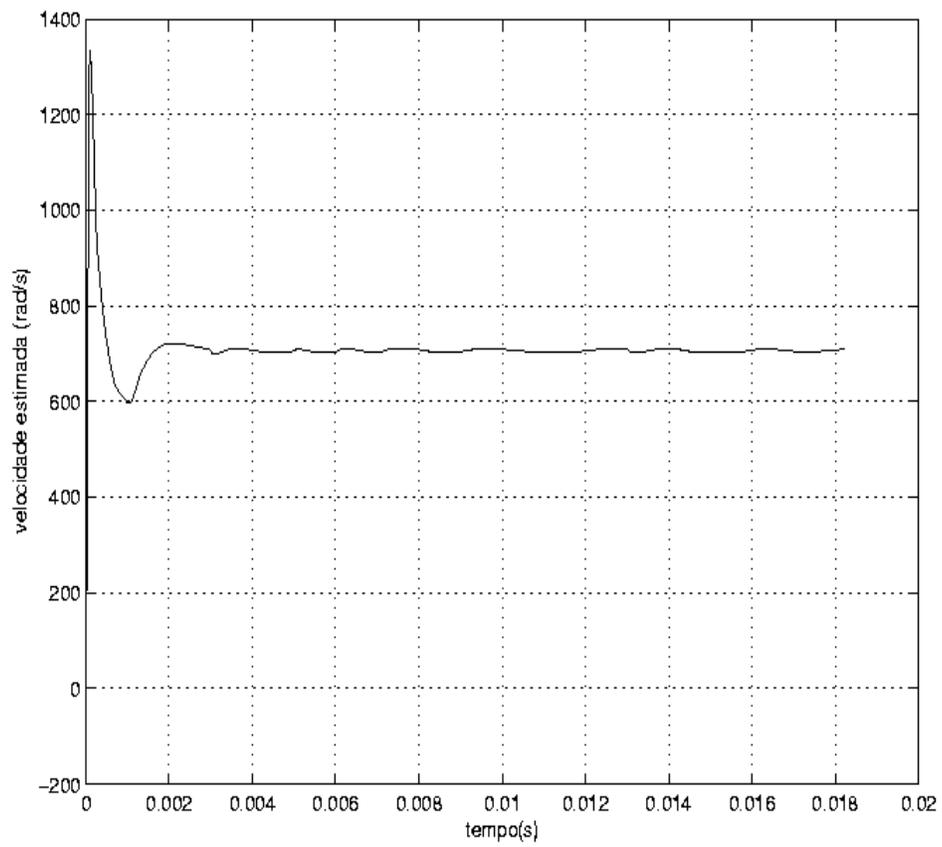


Figura 5.13: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,9$ e $\eta=0,0$

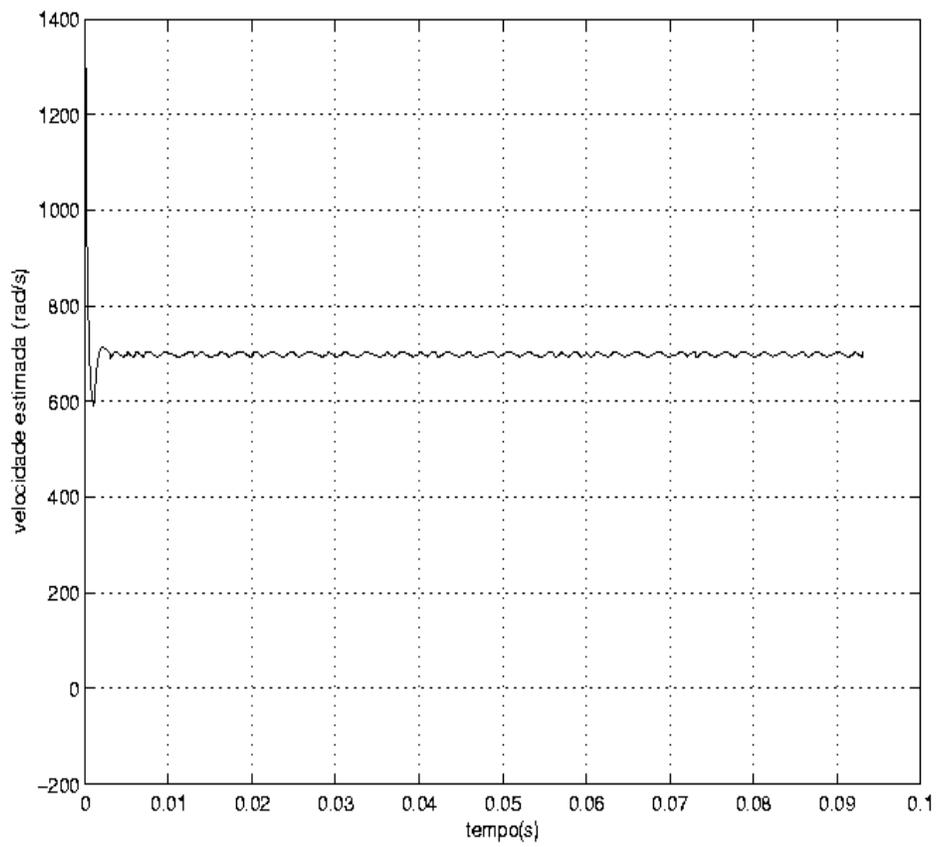


Figura 5.14: Velocidade estimada via RNA com $t = 100$ ms para $\alpha=0,99$ e $\eta=0,01$

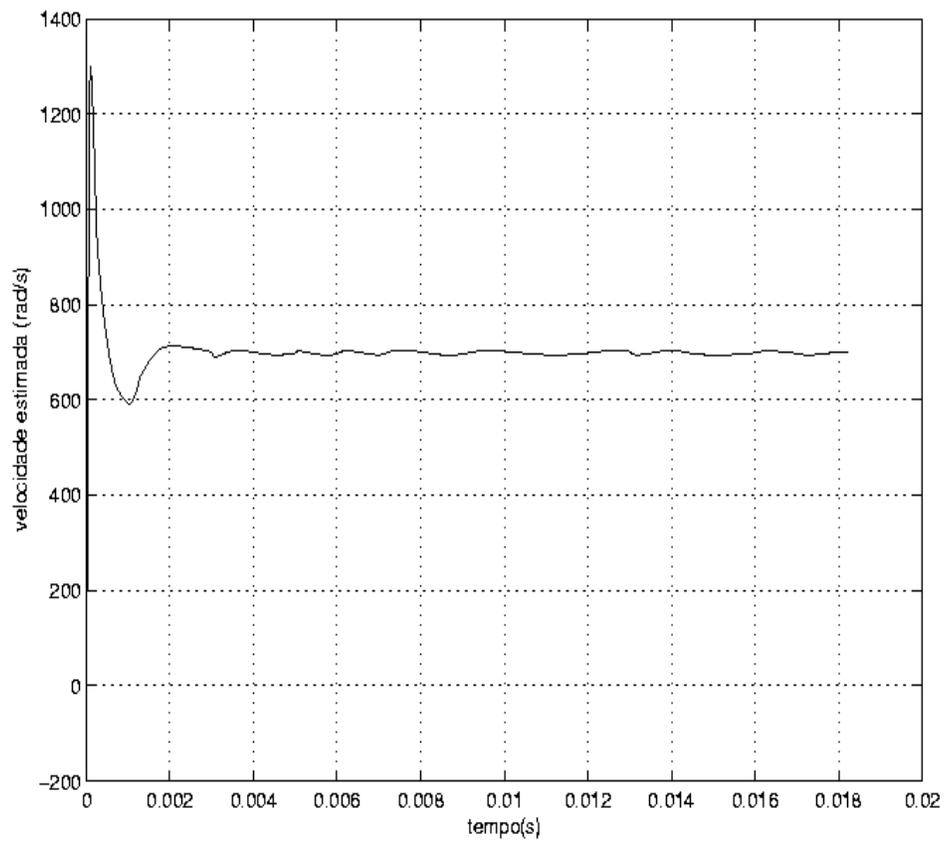


Figura 5.15: Velocidade estimada via RNA com $t = 20$ ms para $\alpha=0,99$ e $\eta=0,01$

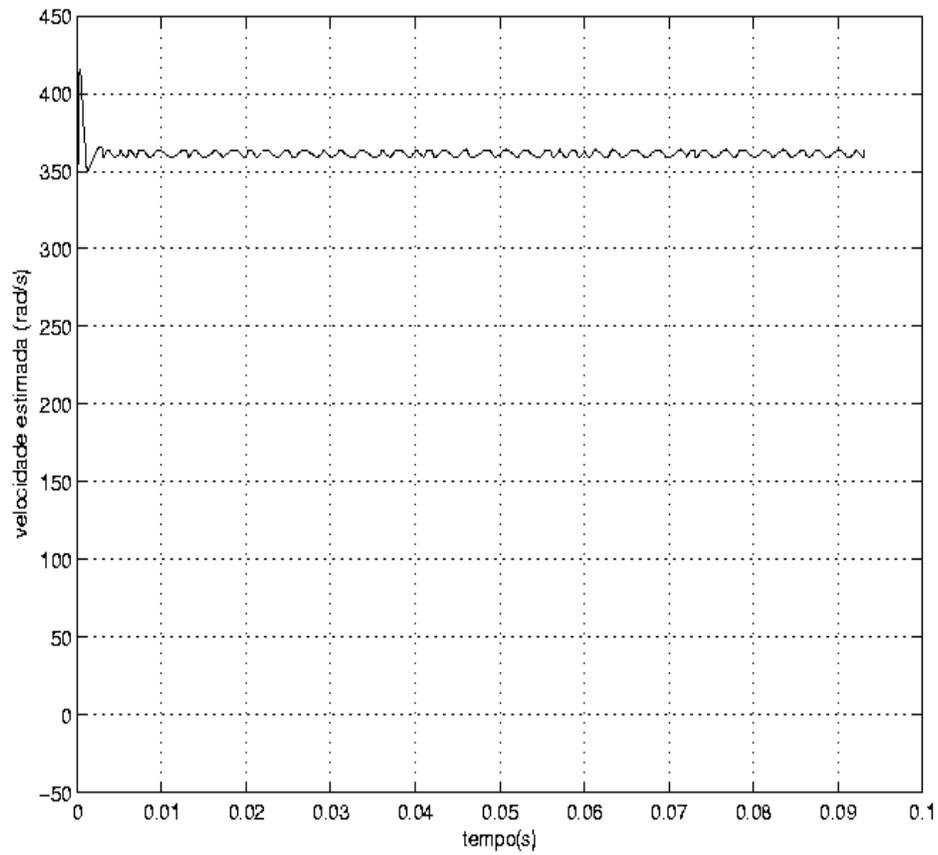


Figura 5.16: Velocidade estimada com menor erro de regime via RNA com $t = 100$ ms para $\alpha = 0,0$ e $\eta = 0,1$

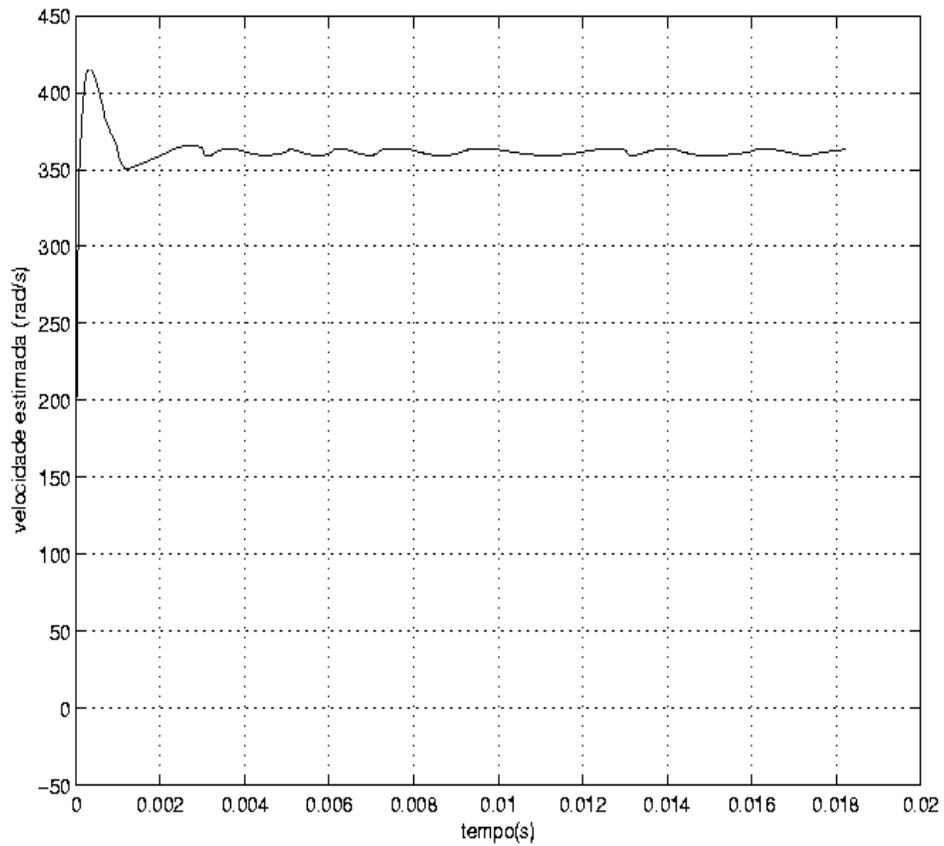


Figura 5.17: Velocidade estimada com menor erro de regime via RNA com $t = 20$ ms para $\alpha = 0,0$ e $\eta = 0,1$

CAPÍTULO 6

Conclusão

Neste trabalho mostrou-se estruturas de modelos para identificar sistemas, abordando as dificuldades existentes devido à não-linearidade do mundo real. Verificase que para se obter modelos matemáticos é necessário ter um certo número de medições e que, tais modelos, sendo lineares ou não-lineares, não podem distanciar-se de suas propriedades reais.

Apresentou-se de forma detalhada os modelos de estimação de velocidade em MI que, por sua vez, foram convertidos em algoritmos de modo a calcular a velocidade estimada ω_r . Abordou-se a técnica via fluxo com e sem utilização MRAS. Mostrou-se estimadores- ω_r através da FEM sem uso da técnica MRAS e também com a utilização desta.

Mostrou-se que aplicações mais recentes em MI são desenvolvidas na área de IA. Nessas abordagens, combinações feitas entre ferramentas inteligentes como, Lógica Nebulosa, Algoritmo Genético e Redes Neurais, tem sido usadas na eliminação de problemas freqüentes em MI. O crescente uso de RNA's em todas as áreas de conhecimento, inclusive em Identificação e Controle de MI, mostra que esta ferramenta fornece uma boa aproximação como estimador, para controle direto de torque, na orientação de campo, no controle de energia da MI, no monitoramento do torque, na detecção e diagnóstico de falhas.

Partindo-se das formas de identificação via RNA's, usou-se o modelo direto para estimar a velocidade através de simulações e, posteriormente, análises. Fez-se o algoritmo para calcular a velocidade estimada ω_r baseando-se nas equações dos fluxos. Deduziu-se uma expressão para a velocidade estimada ω_r que foi utilizada pelo algoritmo de aprendizagem de modo a ajustar o peso da RNA.

A rede recebe os dados de entrada que, juntamente com os dados desejados da saída do sistema, contribuem para a aprendizagem desta. Utilizou-se o treinamento supervisionado por retropropagação, sendo o erro propagado a partir da camada de saída até a camada de entrada e o peso da conexão ω_r , sendo modificado conforme o erro seja retropropagado. Trabalhou-se com uma variação da Regra Delta: a regra delta generalizada.

A partir do desenvolvimento matemático, oriundo das equações da MI, obteve-se modelos para estimar a velocidade e, como contribuição, desenvolveu-se algoritmos visando análises em tempo real. Estes algoritmos foram acoplados ao modelo da máquina e verificou-se o desempenho do sistema máquina-estimadores através de simulações. Os resultados mostraram que é viável estimar ω_r com boa precisão através de RNA's.

6.1 Perspectivas Futuras

Como trabalhos futuros propõe-se:

1. Implementar em microcontroladores os algoritmos desenvolvidos para estimar a velocidade da MI,
2. Desenvolver sistemas embarcados para medição indireta da velocidade da MI.

Pode-se propor ainda métodos, baseados em computação evolutiva, para determinar os parâmetros da RNA α e η ótimos e também algoritmos que considerem a variação dos parâmetros da máquina T_r , R_s , R_r , L_s e L_r . Como não foram

apresentadas neste trabalho as técnicas para estimar a velocidade usando Filtro de Kalman Estendido, a técnica MRAS para estimar a velocidade através da potência reativa e FEM, propõe-se a análise e elaboração de algoritmos para a obtenção da velocidade estimada ω_r para cada uma dessas técnicas.

APÊNDICE A

Modelo da Máquina de Indução

Neste apêndice são apresentadas as equações para o modelo da máquina de indução. Efeitos indesejáveis como histerese, corrente parasita, saturação magnética e outros fatores são geralmente desconsiderados nos modelos para o projeto de controle, (Stephan *et al.*, 1994). Naturalmente, é desejável que o erro não seja tão significativo ao se desconsiderar tais fatores. Devido a dificuldade prática de medição dos fluxos tanto estáticos quanto rotóricos nas máquinas, deve-se obter o modelo da máquina de modo que se possa estimar os parâmetros a partir de medições das correntes e tensões estáticas.

O modelo da máquina de indução pode ser representado na forma de equação de estado a seguir:

$$\begin{bmatrix} \frac{d\lambda_{qr}}{dt} \\ \frac{d\lambda_{dr}}{dt} \\ \frac{d\omega_r}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -(\omega - \omega_r) & 0 \\ -(\omega - \omega_r) & -\frac{R_r}{L_r} & 0 \\ -\frac{3P^2 L_m}{8J_i L_r} i_{ds} & \frac{3P^2 L_m}{8J_i L_r} i_{qs} & -\frac{K}{J} \end{bmatrix} \begin{bmatrix} \lambda_{qr} \\ \lambda_{dr} \\ \omega_r \end{bmatrix} + \quad (A.1)$$

$$\begin{bmatrix} \frac{L_m R_r}{L_r} & 0 & 0 \\ 0 & \frac{L_m R_r}{L_r} & 0 \\ 0 & 0 & -\frac{P}{2J_i} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ T_{at} \end{bmatrix}.$$

A.1 Representação no Modelo Discreto

Devido à predominância de processadores digitais, abordagens no tempo discreto são também comuns na implementação de sistemas de acionamento de máquina de indução, (Åström e Wittenmark, 1997).

A obtenção de modelos discretos a partir de sistemas inerentemente contínuos é comumente realizada através de processos de amostragem. O procedimento de discretização deve ser feito a cada passo de amostragem, considerando que os parâmetros variantes da matriz de estado são constantes entre dois instantes consecutivos de amostragem. Reescrevendo a eq.(A.1), tem-se, (Zuben, 1993),

$$\begin{bmatrix} \frac{d\lambda_{qr}}{dt} \\ \frac{d\lambda_{dr}}{dt} \\ \frac{d\omega_r}{dt} \end{bmatrix} = A_n \begin{bmatrix} \lambda_{qr} \\ \lambda_{dr} \\ \omega_r \end{bmatrix} + B \begin{bmatrix} i_{qs} \\ i_{ds} \\ T_{at} \end{bmatrix}, \quad (\text{A.2})$$

sendo

$$A_n = \begin{bmatrix} -\frac{R_r}{L_r} & -\omega_s(nT) & 0 \\ -\omega_s(nT) & -\frac{R_r}{L_r} & 0 \\ -\frac{3P^2 L_m}{8J_i L_r} i_{ds} & \frac{3P^2 L_m}{8J_i L_r} i_{qs} & -\frac{K}{J} \end{bmatrix} \quad (\text{A.3})$$

e

$$B = \begin{bmatrix} \frac{L_m R_r}{L_r} & 0 & 0 \\ 0 & \frac{L_m R_r}{L_r} & 0 \\ 0 & 0 & -\frac{P}{2J_i} \end{bmatrix}, \quad (\text{A.4})$$

sendo T o período de amostragem, ω_s é a velocidade de escorregamento e A_n é a matriz de estado definida no n -ésimo período de amostragem.

A representação de tempo discreto da eq.A.2 do tipo $\dot{x} = Ax + Bu$ assume a forma $x[(k+1)T] = G(T)x(kT) + H(T)u(kT)$, ou seja

$$\begin{bmatrix} \frac{d\lambda_{qr}}{dt} \\ \frac{d\lambda_{dr}}{dt} \\ \frac{d\omega_r}{dt} \end{bmatrix} = A_n \begin{bmatrix} \lambda_{qr} \\ \lambda_{dr} \\ \omega_r \end{bmatrix} + B \begin{bmatrix} i_{qs} \\ i_{ds} \\ T_{at} \end{bmatrix}, \quad (\text{A.5})$$

fica

$$\begin{bmatrix} \lambda_{qr}(k+1) \\ \lambda_{dr}(k+1) \\ \omega_r \end{bmatrix} = G(T) \begin{bmatrix} \lambda_{qr} \\ \lambda_{dr} \\ \omega_r \end{bmatrix} + H(T) \begin{bmatrix} i_{qs}(k) \\ i_{ds}(k) \\ T_{at}(k) \end{bmatrix}, \quad (\text{A.6})$$

sendo

$$G(T) = e^{A_n T} = \begin{bmatrix} e^{-T \frac{R_r}{L_r}} \cos[\omega_s(nT)T] & e^{-T \frac{R_r}{L_r}} \sin[\omega_s(nT)T] & 0 \\ -e^{-T \frac{R_r}{L_r}} \sin[\omega_s(nT)T] & e^{-T \frac{R_r}{L_r}} \cos[\omega_s(nT)T] & 0 \\ \zeta \frac{3}{8} P^2 \frac{L_m}{J_i L_r} i_{ds}(nT) & \zeta \frac{3}{8} P^2 \frac{L_m}{J_i L_r} i_{qs}(nT) & e^{-\frac{K}{J_i} T} \end{bmatrix} \quad (\text{A.7})$$

e

$$H(T) = \int_0^T e^{A_n t} dt B = \begin{bmatrix} h_1 & h_2 & 0 \\ -h_2 & h_1 & 0 \\ 0 & 0 & -\zeta \frac{P}{2} \end{bmatrix}, \quad (\text{A.8})$$

com

$$h_1 = \frac{R_r L_m L_r}{\omega_s^2 L_r^2 + R_r^2} \left[e^{-T \frac{R_r}{L_r}} (\omega_s \sin(\omega_s) T - \frac{R_r}{L_r} \cos(\omega_s T)) \right] \quad (\text{A.9})$$

$$h_2 = -\frac{R_r L_m L_r}{\omega_s^2 L_r^2 + R_r^2} \left[e^{-T \frac{R_r}{L_r}} (\omega_s \cos(\omega_s) T - \frac{R_r}{L_r} \sin(\omega_s T)) \right] \quad (\text{A.10})$$

$$\zeta = \begin{bmatrix} \frac{1}{K} (1 - e^{-\frac{K}{J_i} T}) & \text{se } K \neq 0 \\ T & \text{se } K = 0 \end{bmatrix}. \quad (\text{A.11})$$

As matrizes $G(T)$ e $H(T)$ podem ser obtidas por diferentes métodos:

- Expansão de série da matriz exponencial,
- Transformada de Laplace,
- Teorema de Carley-Hamilton,
- Transformação para forma Jordan.

Utilizou-se Transformada de Laplace(s) e subsequente transformação para representação discreta transformada(Z).

APÊNDICE B

Estruturas dos Blocos para MATLAB/SIMULINK

Neste apêndice é apresentado o modelo da Máquina de Indução utilizado para verificar o comportamento dos estimadores de velocidade.

Os modelos foram implementados no MATLAB/SIMULINK. Os parâmetros da máquina são apresentados no Capítulo 5. A fig.B.1 mostra o diagrama de montagem dos seguintes blocos: bloco alimentação e bloco motor.

B.1 Bloco Alimentação

Este bloco gera as tensões de alimentação no eixo em quadratura e no eixo direto. Entra-se com os valores das tensões eficazes e a frequência da rede. Na seqüência *abc*, tem-se que, (Ong, 1998):

$$V_a = V_m \cos(\omega t), \quad (\text{B.1})$$

$$V_b = V_m \cos\left(\omega t - \frac{2\pi}{3}\right), \quad (\text{B.2})$$

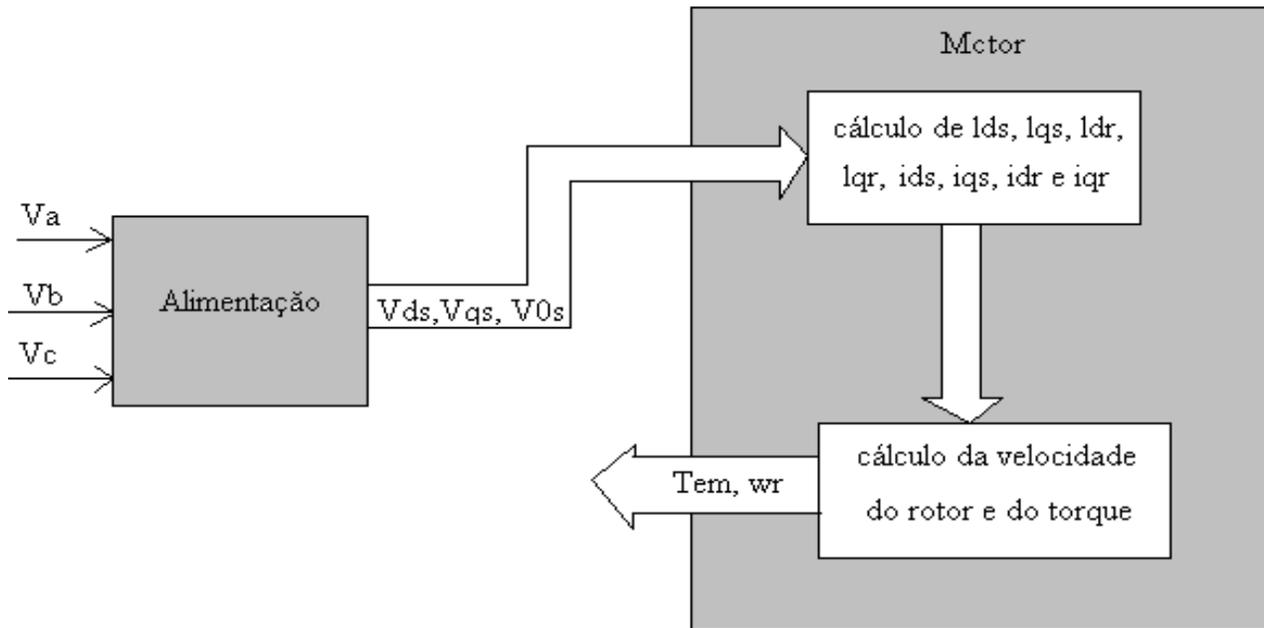


Figura B.1: Descrição dos Blocos da Simulação da Máquina de Indução

$$V_c = V_m \cos\left(\omega t + \frac{2\pi}{3}\right), \quad (\text{B.3})$$

sendo,

$$\omega = 2\pi f,$$

v_a , v_b , e v_c \longrightarrow tensões nas fases a , b e c ,

l_{ds} \longrightarrow fluxo de estator de eixo direto,

l_{qs} \longrightarrow fluxo de estator de eixo em quadratura,

l_{dr} \longrightarrow fluxo de rotor de eixo direto,

l_{qr} \longrightarrow fluxo de rotor de eixo em quadratura.

Substituindo os valores das tensões e frequência nas expressões (B.1), (B.2) e

(B.3), segue que:

$$V_a = 220 \cos(120\pi), \quad (\text{B.4})$$

$$V_b = 220 \cos(120\pi - \frac{2\pi}{3}), \quad (\text{B.5})$$

$$V_c = 220 \cos(120\pi + \frac{2\pi}{3}). \quad (\text{B.6})$$

A equação de transformação de abc para o eixo de referência dq0 é dada por, (Ong, 1998):

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = T \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix}, \quad (\text{B.7})$$

podendo a variável f ser tensão por fase, corrente por fase ou fluxo da máquina e a matriz \mathbf{T} é dada por:

$$T = \begin{bmatrix} \cos \theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin \theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad (\text{B.8})$$

sendo θ o ângulo elétrico (rad/s).

B.2 Bloco Motor

De posse dos parâmetros do motor, obtém-se o modelo da Máquina de Indução no eixo dq a partir das equações dadas a seguir:

Para o cálculo dos fluxos, tem-se

$$\frac{d}{dt} \begin{bmatrix} \lambda_{ds} \\ \lambda_{qs} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \end{bmatrix} + \frac{R_s}{x_{ls}} \begin{bmatrix} \lambda_{md} - \lambda_{ds} \\ \lambda_{mq} - \lambda_{qs} \end{bmatrix}, \quad (\text{B.9})$$

$$\frac{d}{dt} \begin{bmatrix} \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{dr} \\ v_{qr} \end{bmatrix} + \begin{bmatrix} 0 & -\omega_r \\ \omega_r & 0 \end{bmatrix} \begin{bmatrix} \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} - \frac{R_s}{x_{ls}} \begin{bmatrix} \lambda_{md} - \lambda_{dr} \\ \lambda_{mq} - \lambda_{qr} \end{bmatrix} \quad (\text{B.10})$$

sendo,

$$\begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} = \frac{1}{x_{ls}} \begin{bmatrix} \lambda_{ds} - \lambda_{md} \\ \lambda_{qs} - \lambda_{mq} \end{bmatrix}, \quad (\text{B.11})$$

$$\begin{bmatrix} i_{dr} \\ i_{qr} \end{bmatrix} = \frac{1}{x_{lr}} \begin{bmatrix} \lambda_{dr} - \lambda_{md} \\ \lambda_{qr} - \lambda_{mq} \end{bmatrix}, \quad (\text{B.12})$$

$$\lambda_{mq} = x_M \left(\frac{\lambda_{qs}}{x_{ls}} + \frac{\lambda_{qr}}{x_{lr}} \right), \quad (\text{B.13})$$

$$\lambda_{md} = x_M \left(\frac{\lambda_{ds}}{x_{ls}} + \frac{\lambda_{dr}}{x_{lr}} \right), \quad (\text{B.14})$$

$$\frac{1}{x_M} = \frac{1}{x_m} + \frac{1}{x_{ls}} + \frac{1}{x_{lr}}, \quad (\text{B.15})$$

ou ainda de modo equivalente

$$\frac{d}{dt} \begin{bmatrix} \lambda_{ds} \\ \lambda_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{ds} \\ v_{qs} \\ v_{dr} \\ v_{qr} \end{bmatrix} - \begin{bmatrix} R_s & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 \\ 0 & 0 & R_r & 0 \\ 0 & 0 & 0 & R_r \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega_r \\ 0 & 0 & \omega_r & 0 \end{bmatrix} \begin{bmatrix} \lambda_{ds} \\ \lambda_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} \quad (\text{B.16})$$

O modelo das equações utilizadas para o cálculo do fluxo, pode ainda ser obtido das eqs.(3.8) e (3.9).

O cálculo do torque pode ser obtido da equação (B.16):

$$T_{em} = \frac{3P}{2} \frac{1}{2} \begin{bmatrix} \lambda_{ds} & \lambda_{qs} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix}, \quad (\text{B.17})$$

sendo P o número de pólos.

A velocidade do rotor relacionada ao conjugado eletromagnético é dada por:

$$\frac{d}{dt}\omega_r = \frac{P}{2} \frac{1}{J_m} (T_{em} + T_{mec} - T_{at}), \quad (\text{B.18})$$

sendo J_m o momento de inércia

O algoritmo da fig.B.2 a seguir mostra os passos para o cálculo do torque e da velocidade do rotor.

A partir do algoritmo da fig.B.2, obtém-se os fluxos, e a velocidade angular do rotor que poderão ser utilizados como referência para as diversas técnicas de estimação de velocidade empregadas neste trabalho. Obtém-se também os valores de v_s e i_s que são necessários para as técnicas apresentadas.

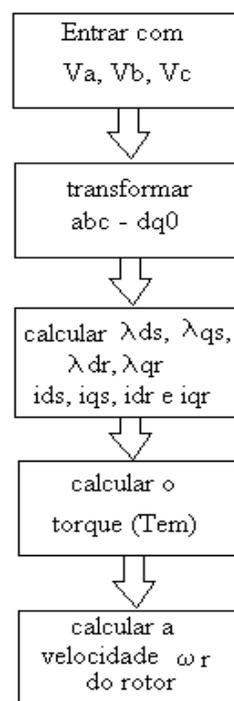


Figura B.2: Algoritmo para o cálculo do torque e da velocidade do rotor

Bibliografia

- Aguirre, Luis Antonio (,2000). *Introdução à Introdução de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. 1ª ed.. editora UFMG.
- Aller, J. M., J. A. Restrepo, A. Bueno, T. Paga, V. M. Guzman and Gimenez (,2000). Sensorless speed control of the induction machine combining field oriented method and dtc. *IEEE International Caracas conference* pp. 791–796.
- Andò, B., S. Baglio, A. Cocuccio, S. Graziani and A. La Terra (,1997). A smart sensor for pressure measurement. *IEEE* pp. 1284–1287.
- Armstrong, Gavin J., David J. Atkinson and Paul P. Acarnley (,1997). A comparison of estimation techniques for sensorless vector controlled induction motor drives. *IEEE*.
- Asaii, B., D.F. Gosden and S.Sathiakumar (,1996). Neural network applications in control of electric vehicle induction machine drives. *Power Electronics and Variable Speed Drives, 1996. Sixth International Conference on (Conf. Publ. No. 429)* pp. 273–278.
- Åström, Karl J. and Björn Wittenmark (,1997). *Computer-Controlled Systems Theory and Design*. third edition ed.. Prentice Hall.
- Ben-Brahim, Lazhar (,1995). Motor speed identification, via neural network. *IEEE Industry Applications Magazine*.

- Ben-Brahim, Lazhar and Susumu Tadakuma (,1998). Pratical considerations for sensorless induction motor drive system. *IEEE*.
- Boillereaux, Lionel and Jean-Marie Flaus (,2000). A new approach for designing model-based indirect sensors. *IEEE* pp. 601–608.
- Boldea, Ion and S.A. Nasar (,1998). *Electric Drives*. CRC press.
- Bordon, M.E., I.N. da Silva and A.N. de Souza (,1999). New approach of induction machine drive using an adaptive digital pid controller with gain planning based on artificial neural networks. *American Control Conference, 1999. Proceedings of the 1999* **3**, 2153–2157.
- Burton, B. and R.G. Harley (,1994). Reducing the computational demands of continually online trained artificial neural networks for system identification and control of fast processes. *Industry Applications Society Annual Meeting, 1994., Conference Record of the 1994 IEEE* **3**, 1836–1843.
- Burton, B., R.G. Harley, G. Diana and J.L. Rodgerson (,1998). Implementation of a neural network to adaptively identify and control vsi-fed induction motor stator currents. *Industry Applications, IEEE Transactions on* **34**, 580–588.
- Cabrera, L.A., M.E. Elbuluk and D.S. Zinger (,1994). Learning techniques to train neural networks as a state selector for inverter-fed induction machines using direct torque control. *Power Electronics Specialists Conference, PESC '94 Record., 25th Annual IEEE* **1**, 233–242.
- Cabrera, L.A., M.E. Elbuluk and D.S. Zinger (,1997). Learning techniques to train neural networks as a state selector for inverter-fed induction machines using direct torque control. *Power Electronics, IEEE Transactions on* **12**, 788–799.
- D'Angelo, Marcos F.S.V. and P.P. CostaJr (,2000). State estimation for induction machines using an neural network backpropagation technique. *Systems, Man, and Cybernetics, 2000 IEEE International Conference on* **4**, 2613–2618.

- Dunnigan, M. W., S. Wade, B. W. Williams and X. Yu (,1998). Position control of a vector controlled induction machine using slotine's sliding mode control approach. *Electric Power Applications - IEEE Proceedings* pp. 231–238.
- Elloumi, Mohsen, Lazhar Ben-Brahim and Mohamed A Al-Hamadi (,1998). Survey of speed sensorless controls for im drives. *IEEE*.
- Elsimary, Hamed (,1997). Implementation of neural network and genetic algorithms for novelty filters for fault detection. *Circuits and Systems, 1996., IEEE 39th Midwest symposium on* **3**, 1432–1435.
- Filippetti, F., A. Uncini, C. Piazza, P. Campolucci, C. Tassoni and G. Franceschini (,1996). Neural network architectures for fault diagnosis and parameter recognition in induction machines. *Electrotechnical Conference, 1996. MELECON'96., 8th Mediterranean* **1**, 289–293.
- Filippetti, F., G. Franceschini and C. Tassoni (,1995). Neural networks aided on-line diagnostics of induction motor rotor faults. *Industry Applications, IEEE Transactions on* **31**, 892–899.
- Filippetti, F., G. Franceschini and C. Tassoni (,2000). Recent developments of induction motor drives fault diagnosis using ai techniques. *Vas, P. Industrial Electronics, IEEE Transactions on* **47**, 994–1004.
- Filippetti, F., G. Franceschini, C. Tassoni and Peter Vas (,1998a). Ai techniques in induction machines diagnosis including the speed ripple effect. *Industry Applications, IEEE Transactions on* **34**, 98–108.
- Filippetti, F., G. Grellet, G. Salles, G. Franceschini and C. Tassoni (,1998b). Monitoring of induction machines load torque disturbances: an alternative nn-based method. *Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting. The 1998 IEEE* **1**, 103–110.

- FonsecaNeto, J. V. and L. Mouzinho (,2003). Analysis comparative of speed estimation techniques for asynchronous machines. *IASTED*.
- FonsecaNeto, João Viana Da (,1996). Identificação da velocidade de motores de indução via rede neural.
- Frosini, Lucia and Giovanni Petrecca (,2001). Neural networks for load torque monitoring of an induction motor. *Applied Soft Computing* **1**, 215–223.
- Grzesiak, L.M. and J. Skoczylas (,1996). Ac machine torque and stator flux estimation using a neural network based on the steady-state 2d field model. *Industrial Electronics, 1996. ISIE '96., Proceedings of the IEEE International Symposium on* **1**, 358–361.
- Haykin, Simon (,2001). *Redes Neurais: princípios e prática*. 2 ed.. Ed. Bookman.
- Idris, N. R. N. and A. H. M. Yatim (,2000). Modeling of a new torque controller for direct torque control of induction machines. *TENCON* **2**, 39–45.
- Irwin, G. W., K. Warwick and K. J. Hunt (,1995). *Neural Network Applications In Control*. 1 ed.. The Institution of Electrical Engineers.
- Jacobina, C.B., J. Fo Bione, F. Salvadori, A. M. N. Lima and L.A.S. Ribeiro (,2000). A simple indirect field oriented control of an induction machine. *Industry Applications Conference*.
- J.Schoukens and R. Pintelon (,1991). *Identification of Linear Systems*. 1 ed.. Pergamon Press.
- Ke, J., B. Wu and R. Sotudeh (,1997). Dsp based induction motor torque and parameter identification. *Power Electronics and Drives Systems* **2**, 679–683.
- Kim, K. and A.G. Parlos (,2002). Model-based fault diagnosis of induction motors using non-stationary signal segmentation. *Mechanical Systems and Signal Processing* **16**, 223–253.

- Kim, Nam-Joon and Dong-Seok Hyun (,1994). Very low speed control of induction machine by instantaneous speed and inertia estimation. *Industrial Electronics Control and Instrumentation - IECON* **1**, 605–610.
- L.Collamati, F. Filippetti, G. Franceschini, S. Pirani and C. Tassoni (,1996). Induction machine stator fault on-line diagnosis based on labview environment. *Electrotechnical Conference, 1996. MELECON '96., 8th Mediterranean* **1**, 495–498.
- Ljung, Lennart(,1987). *System Identification Theory For The User*. 10 ed.. Prentice-Hall.
- Llas, C., G. Griva and F. Profumo (,1996). On-line identification of mechanical parameters and load torque in induction machine field oriented drives. *Industrial Electronics - IEEE* **2**, 985–989.
- Luetzig, Guenter, Manuel Sánchez-Castillo and Reza Langari (,1997). On tool wear estimation through neural networks. *IEEE* pp. 2359–2363.
- Martinez, L.Z. and A.Z Martinez (,1997). Identification of induction machines using artificial neural networks. *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on* **3**, 1259–1264.
- Miloudi, A. and A. Draou (,2001). Neural controller design for speed control of an indirect field oriented induction machine drive. *Industrial Electronics Society - IEEE* **2**(27), 1225–1229.
- Moudden, Yassir and Olivier Boubal (,2001). Indirect measurement of cylinder pressure peak position in an internal combustion engine. *IEEE* pp. 772–777.
- Oh, D. S., K. Y. Cho and M. J. Youn (,1991). New rotor parameter estimation for a lux and speed control of induction machine considering saturation effects. *Industrial Electronics Control and Instrumentation - IECON* **1**, 561–566.

- Ong, Chee-Mun (,1998). *Dynamic Simulation of Eleetric Machinery*. 10 ed.. Prentice Hall PTR.
- Peng, Fang Z. (,2000). Speed and flux sensorless field oriented control of induction motors for electric vehicles. *IEEE* pp. 133–139.
- P.Resende, C.A. Maia and J.L. Silvino (,1997). A neural vector control for induction machine. *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on* **3**, 1265–1269.
- Rajashekara, Kaushik, Atsuo Kawamura and Kouki Matsuse (,1993). Sensorless control of ac motor drives - speed and position sensorless operation. *IEEE*.
- Rajashekara, Kaushik, Atsuo Kawamura and Kouki Matsuse (,1996). Sensorless control of ac motor drives - speed and position sensorless operation. *IEEE*.
- Raol, J. R. and Madhuranath (,1996). Neural networks architectures for parameter estimation of dynamical systems. *IEEE transactions on control theory applications*.
- Saccomano, Alaor Mousa (,2003). Controle de velocidade e torque de motores trifásicos. *Mecatrônica Atual*.
- Schauder, Colin (,1992). Adaptive speed identification for vector control of induction motors without rotational transducers. *IEEE, IAS Annual Meeting*.
- Shimazu, Hideaki, Hiroshi Ito, Takuo Hashimoto, Masakatsu, Toshiaki Tamai, Shigenori Nakamura and Ikuo Ohtaka (,1992). Indirect measurement of intracranial pressure by using a ventricle-pritoneal shunt swystem. *IEEE* pp. 117–118.
- Shoudao, Huang and Lin Youjie (,2001). Study of neural network application on direct torque control of induction machine. *Electrical Machines and Systems. ICEMS 2001. Proceedings of the Fifth International Conference on* **47(5)**, 1282–1285.

- Solopchenko, Gennady N., Leonid K. Reznik and W. Carroll Johnson (,1994). Fuzzy intervals as a basis for measurement theory. *IEEE* pp. 405–406.
- Stefano, R. Di, S. Meo and M. Scarano (,1994). Induction motor faults diagnostic via artificial neural network (ann). *Industrial Electronics, 1994. Symposium Proceedings, ISIE '94., 1994 IEEE International Symposium on* pp. 220–225.
- Stephan, Jennifer, Marc Bodson and John Chiasson (,1994). Real-time estimation of the parameters and fluxes of induction motors. *IEEE Transactions on Industry Applications* **30**(3), 746–759.
- Tajima, H. and Y. Hori (,1993). Speed sensorless field-orientation control of the induction machine. *Industry Applications - IEEE* **29**, 175–180.
- Tallam, R.M., T.G. Habetler and R.G Harley (,2001). Continual on-line training of neural networks with applications to electric machine fault diagnostics. *Power Electronics Specialists Conference, 2001. PESC. 2001 IEEE 32nd Annual* **4**, 2224–2228.
- Tallam, R.M., T.G. Habetler, R.G. Harley, D.J. Gritter and B.H. Burton (,2000). Neural network based on-line stator winding turn fault detection for induction motors. *Industry Applications Conference. Conference Record of the IEEE* **1**, 375–380.
- Telford, D., M. W. Dunnigan and B. W. Willians (,2000). A comparison of vector control and direct torque control of an induction machine. *Power Electronics Specialists Conference*.
- Toqeer, R.S and N.S Bayindir (,2000). Neurocontroller for induction motors. *Microelectronics, 2000. ICM 2000. Proceedings of the 12th International Conference on* pp. 227–230.

- Wai, Rong-Jong and Jia-Ming Chang (,2002). Intelligent control of induction servo motor drive via wavelet neural network. *Electric Power System Research* **61**, 67–76.
- Wang, Xiufeng and M. Elbuluk (,1996). Neural network control of induction machines using genetic algorithm training. *Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS '96., Conference Record of the 1996 IEEE* **3**, 1733–1740.
- Wishart, Michael T. and Ronald G. Harley (,1993). Identification and control of induction machines using artificial neural networks. *IEEE*.
- Xiong, Shen-Shu and Zhao-Ying Zhou (,2001). Dynamic parameter estimation of velocity sensors using an indirect measurement approach. *IEEE Instrumentation and Measurement* pp. 794–797.
- Zhang, Li and K. M. Hasan (,1999). Neural network aided energy efficiency control for a field-orientation induction machine drive. *Electrical Machines and Drives. Ninth International Conference on (Conf. Publ. No. 468)* pp. 356–360.
- Zuben, Fernando José Von (,1993). Redes neurais aplicadas ao controle de máquina de indução. Master's thesis. UNICAMP.