

Medição Indireta Ótima Estocástica e Neuronal em Sistemas Dinâmicos

Lucilene Ferreira Mouzinho

Tese de Doutorado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da
Informação/Instrumentação e Controle

Benedito Antonio Luciano, Dr.
João Viana da Fonseca Neto, Dr.
Raimundo Carlos Silvério Freire, Dr.
Orientadores

Campina Grande, Paraíba, Brasil
©Lucilene Ferreira Mouzinho, Junho de 2007

Medição Indireta Ótima Estocástica e Neuronal em Sistemas Dinâmicos

Lucilene Ferreira Mouzinho

Benedito Antonio Luciano, Dr.
João Viana da Fonseca Neto, Dr.
Raimundo Carlos Silvério Freire, Dr.
Orientadores

Professor Celso Pascoli Bottura, PhD.

Professor Sebastian Yuri Cavalcanti Catunda, D.Sc.

Professor Pablo Javier Alsina, D.Sc.

Campina Grande, Paraíba, Brasil, Junho de 2007

Dedicatória

Dedico este trabalho, com muito amor e carinho, aos meus pais, Elizia e Raimundo.
Aos meus filhos, Sâmia e Vinicius, flores do meu jardim, e aos quais reneguei um tempo precioso em busca desta realização.

Aos meus irmãos e sobrinhos pelo carinho, convívio e pelo privilégio de ser um dos membros dessa grande família.

Agradecimentos

Agradeço este trabalho a Deus criador, Doutor dos doutores que me deu forças e me proporcionou o dom de concatenar meus pensamentos, organizar, estruturar e escrever este trabalho.

Aos meus pais, Elizia e Raimundo, por me permitirem ter acesso à educação e, especialmente, a minha mãe pelo desejo infinito de dar aos seus filhos aquilo que ela não pôde ter.

A todos os meus bons professores que conheci ao longo do tempo e que contribuíram com a minha educação por meio de conteúdo, exemplos e palavras.

Ao meu Orientador Prof. Dr. Benedito Antonio Luciano pela amizade, contribuições, observações, discussões, paciência, dedicação, simpatia e pelas palavras de apoio nesta jornada. Meu agradecimento e admiração.

Ao meu Orientador, Prof. Dr. João Viana da Fonseca Neto pela sua amizade, pelo incentivo na realização do doutorado, pela credibilidade depositada em minha pessoa, contribuições, paciência, compreensão, discussões, observações, por disponibilizar o Laboratório de Automação e Controle da UFMA e fornecer suporte para este trabalho. Minha gratidão e meu carinho.

Ao Prof. Dr. Raimundo Carlos Silvério Freire pela amizade, pela oportunidade de fazer o doutorado, pela paciência e por todos os meios colocados a minha disposição, entre eles, o Laboratório de Instrumentação e Metrologia Científicas, para a realização deste trabalho. Meu agradecimento e minha gratidão.

Aos colegas do Laboratório de Automação e Controle da UFMA, Ivanildo, Patrícia, Pedro Turibe, Joaquim, Inácio, Fábio, Roller Ibañez, Jorge Farid e Jorge Henrique pelo ambiente agradável, pela solidariedade e pela amizade.

Aos colegas do meio científico do Laboratório de Instrumentação e Metrologia Científicas (LIMC) e demais laboratórios da UFCG, Prof. Brunelli, Zurita, Moacir, Alves, Cláudia, Cleonilson Protásio, Alfranque, Magdalena, Prof. Sabiniano, Prof. Ilton, Marcelo Sylvio, Walter, Anderson, Joelle, Marcelo Cardoso, Ricardo, Paula, Cintia, Monica, Patryckson, Prof. Reginaldo e Welflen pela amizade.

Ao Prof. Lucius Vinicius Rocha Machado (CEFET-MA) pelo apoio, correções e sugestões.

Ao Prof. Dr. Sebastian Yuri Cavalcanti Catunda (UFMA) pelas suas observações, críticas e pelas contribuições.

Às amigas e professoras Ana Carolina (FUCAPI-AM), Josélia (CEFET-BA), e Ângela Timótia (UEA-AM) pela agradável companhia, pelas contribuições, pelo carinho, pelas discussões, pela paciência e pela compreensão.

Aos professores e colegas Prof. Gutembergue (UFCEG) e Prof. Guilherme (UFPA) pelas sugestões durante este trabalho.

Aos colegas do Centro de Lançamento de Alcântara - MA, Jonas de J. Barros e José Alano Peres de Abreu pela amizade e pelas contribuições.

Ao colega da Elyte Tecnologia-AM e professor da UEA-AM, Angilberto M. Ferreira Sobrinho pela amizade e pelas contribuições neste trabalho.

Ao Dr. Klaus Körner, Institut für Technische Optik (ITO)- University Stuttgart pelas sugestões e correções.

Aos meus irmãos e, especialmente a minha irmã, Eliziane, por não ter medido esforços para me acomodar e disponibilizar meios agradáveis em São Luís para o desenvolvimento deste trabalho.

Ao fiel amigo, Barbosa pelos momentos de lazer, pelo auxílio e pelas palavras de incentivo.

À COPELE e a Miniblibio da Engenharia Elétrica pelo apoio dado durante toda essa caminhada.

À Biblioteca Setorial da UFMA pelo apoio.

Ao PROCAD nº0152/01, a FAPEMA e ao CNPq pela concessão das bolsas de estudo que foram fornecidas, intercaladamente, durante esses anos.

Aos professores do CEFET-MA do Departamento de Eletro-Eletrônica por terem suprido minha ausência.

Aos membros da Banca Examinadora pelas valiosas contribuições realizadas durante a defesa do Exame de Qualificação.

Epígrafe

”O que Deus mais usa em Seus desígnios
são instrumentos de medição defeituosos
para obter resultados exatos.”

Lucilene F. Mouzinho.

Resumo

A medição indireta é uma alternativa para situações nas quais se têm dificuldades em realizar uma medição por meio de sensores diretamente associados com variações da grandeza a ser medida. O objetivo deste trabalho é desenvolver procedimentos para sistemas de medição indireta em tempo real ou não. Os sistemas de medição indireta (SMI) são classificados nesta tese, de forma a especificar o tipo de tratamento a ser usado para obtenção de uma determinada grandeza e de como analisar esse resultado estimado. Cada método deve ser empregado conforme a aplicação de forma isolada ou em conjunto, para se obter a grandeza de interesse. Nesta tese, são realizadas abordagens sobre duas técnicas utilizadas nos estudos de casos: a *Filtragem de Kalman* e as *Redes Neurais Artificiais*. Considerando-se a complexidade das aplicações realizadas nos estudos de casos, é fundamental desenvolver esses métodos direcionados para a aplicação, bem como, a simulação e análise prévia dos resultados até a implementação do sistema. São desenvolvidos quatro SMI: medição indireta da velocidade de um veículo aeroespacial; medição indireta da temperatura no interior de um objeto dentro de uma estufa; medição indireta da velocidade rotórica de um motor de indução e, como acréscimo, a este estudo de caso, a medição indireta robusta de velocidade. A implementação dos SMI possibilita verificar a viabilidade do uso de ferramentas matemáticas para a realização de medições indiretas ótima estocásticas e neurais em sistemas dinâmicos.

Palavras-chave: medição, erros de medição, estimação de estado, redes neurais artificiais, sistema estocástico, filtragem de Kalman, modelagem, sistema em tempo real e sensor virtual.

Abstract

Indirect measuring is an alternative wherever is difficult to perform a measuring by means of sensors directly associated with variations in the magnitude to be measured. This work aims at developing procedures for indirect measurement systems, whether this measuring is done in real time or not. Indirect Measurement Systems (IMS) are classified in this Thesis in such a way as to specify the sort of treatment to be used for the attainment of a given magnitude, also including the means to analyze those results. Each method must be used according to the application, whether isolated or in groups, as to obtain the magnitude of interest. In this thesis, two techniques used in case studies are approached: the Kalman Filter and the Artificial Neural Network. Considering the complexity of the applications performed in case studies, it is crucial to develop methods addressed to the application as well as simulation and prior analysis of the results until the system's implementation. Four *IMS* are developed: indirect speed measuring of an space aircraft; indirect temperature measuring of an object inside an oven; indirect rotor speed measuring of an induction motor and, as supplement to this case study, the robust indirect speed measuring. The *IMS* implementation allows us to check the viability of using mathematical tools to conduct indirect optimal measurements, both, stochastic and neuronal in dynamic systems.

Keywords: measurement, measurement errors, state estimation, artificial neural networks, stochastic system, Kalman Filter, modeling, real time system and virtual system.

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	5
1.2	Descrição do Problema	12
1.3	Objetivos da Tese	12
1.3.1	Objetivo Geral	13
1.3.2	Objetivos Específicos	13
1.4	Metodologia Utilizada	14
1.5	Estrutura do Documento	16
2	Sistema de Medição Indireta	19
2.1	Medição Clássica - Instrumentação Real	21
2.2	Sensor Virtual	22
2.3	Instrumentação Virtual	24
2.4	Sistema de Medição Indireta	26
2.4.1	Uma Taxonomia dos Sistemas de Medição Indireta	27
2.4.2	Etapas do Projeto de um SMI	30
2.5	Conclusão	31
3	Métodos de Medição Indireta	33
3.1	Rede Neuronal Artificial	33
3.1.1	Redes Biológicas	35
3.1.2	O Perceptron	36
3.1.3	Processo de aprendizado	38
3.1.4	Modelagem Direta	42

3.2	Filtragem Ótima Estocástica para <i>Hardware</i> de Tempo Real	43
3.2.1	Modelo do Sistema Real	45
3.2.2	Decomposição do Modelo	46
3.2.3	Os Processos de Estimação	49
3.2.4	Algoritmo <i>Kalman</i> -Padrão	52
3.3	Comentários	53
4	Considerações Sobre Projetos de SMI-TR	55
4.1	<i>Hardware</i> e <i>Software</i> em Tempo Real	56
4.2	Desempenho do Sistema	58
4.2.1	Considerações do Desempenho	59
4.2.2	Integração <i>Hardware/Software</i> em Tempo Real	60
4.3	Qualidade da Medição Indireta	62
4.4	Modelo do Sistema	65
4.5	Análise de Erros do Estimador	68
4.6	Incerteza em Medição Indireta	69
4.7	Sistema Embarcado Ótimo de <i>Kalman</i>	72
4.7.1	Funções Objetivo	74
4.7.2	Mapeamento da memória	78
4.8	Conclusões	83
5	Estudos de Casos	84
5.1	Medição Indireta da Velocidade de um Veículo Aeroespacial	85
5.1.1	Arquitetura para o Filtro de <i>Kalman</i>	88
5.1.2	Resultados Experimentais	92
5.2	Medição Indireta da Temperatura no Interior de um Objeto dentro de uma Estufa	94
5.2.1	Modelo do Sistema Térmico	95
5.2.2	Medição Indireta da Temperatura	100
5.3	Calor Específico Estimado	105
5.4	Testes Computacionais	107
5.4.1	Descrição Numérica do Espaço de Estados	108

5.4.2	Procedimento Computacional	109
5.4.3	Procedimentos Experimentais	110
5.4.4	Resultados Experimentais	113
5.5	Medição Indireta da Velocidade Rotórica de um Motor de Indução	115
5.5.1	Modelo da Máquina de Indução	116
5.5.2	Medição Indireta Robusta da Velocidade	130
5.6	Conclusões	134
6	Conclusão	139
7	Sugestões para Trabalhos Futuros	143
	Referências Bibliográficas	144
A	Modelo do Sistema de Rastreo	166
A.1	Algoritmo do Filtro de Kalman	170
A.2	Resultados Computacionais	172
B	Desenvolvimento da Regra Delta	177
	Anexo A LM35 – Precision Centigrade Temperature Sensor	185

Lista de Símbolos e Abreviaturas

A/D	Analógico/Digital
ADC	<i>Analog-to-Digital Converter</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ASIC	<i>Application Specific Integrated Circuit</i>
CLA-MA	Centro de Lançamento de Alcântara - Alcântara/Maranhão/Brasil
CT	Contínuo no Tempo
DAC	<i>Digital-to-Analog Converter</i>
DSP	<i>Digital Signal Processors</i>
DT	Discreto no Tempo
E/S	Entrada/Saída
FK	Filtro de <i>Kalman</i>
FKE	Filtro de <i>Kalman</i> Estendido
FKU	Filtro de <i>Kalman Unscented</i>
FPGA	<i>Field Programmable Gate Array</i>
GPIO	<i>General Purpose Interface Bus</i>
GUM	<i>Guide to the Expression of Uncertainty in Measurement</i>
HP	<i>Hewlett-Packard</i>
HP-VEE	<i>Hewlett-Packard Visual Engineering Environment</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
LabVIEW	<i>Laboratory Virtual Instrument Engineering Workbench</i>
LCD	<i>Liquid Crystal Display</i>
LTI	<i>Linear Time Invariant</i>
M8C	<i>8-bit microcontroller</i>
MIPS	Milhões de Instruções Por Segundo

MLP	<i>Multilayer Perceptron</i>
MATLAB®	<i>MATrix LABoratory</i>
MRUA	Movimento Retilíneo Uniformemente Acelerado
MV	Máxima Verossimilhança
PI	Ponto de Impacto
PSoC™	<i>Programmable System-on-Chip</i>
PWM	<i>Pulse-Width Modulation</i>
RAM	<i>Random Access Memory</i>
RNA	Rede Neuronal Artificial
SoC	<i>System-on-Chip</i>
SMD	Sistema de Medição Direta
SMI	Sistema de Medição Indireta
SMI-TR	Sistema de Medição Indireta em Tempo Real
SNR	<i>Signal-to-Noise Ratio</i>
SRAM	<i>Static Random Access Memory</i>
STDL	Sistemas de Tratamento de Dados de Localização
STR	Sistema em Tempo Real
TRF	Transformada Rápida de Fourier
TU	Transformada <i>Unscented</i>
VHDL	<i>Very High Speed Integrated Circuit Hardware Description Language</i>
VIM	Vocabulário Internacional de Termos Fundamentais e Gerais de Metrologia
VS-40	Veículo de sondagem projetado e desenvolvido no IAE (Instituto de Atividades Aeroespaciais)

Termos utilizados para a Rede Neuronal Artificial

w	peso da rede neuronal artificial
x	sinais apresentados à entrada da rede neuronal artificial
μ	polarização aplicada externamente à rede neuronal artificial
y	neurônio de saída da rede neuronal artificial
d	resposta desejada
m	número de neurônios na camada de entrada
ε	diferença entre a resposta desejada e a resposta obtida
η	taxa de aprendizagem ou coeficiente de treinamento
δ	valor da <i>Regra Delta</i> no neurônio para a correção do peso
$\sigma(\cdot)$	função de ativação

Termos utilizados para Modelos no Espaço de Estados

x	vetor de estado
A	matriz dinâmica do sistema
B	matriz entrada-controle
u	vetor de entrada
G	matriz ruído do sistema
y	vetor de saída
C	matriz saída
D	matriz controle-saída
$n \times n$ ou $n \times m$ ou $n \times p$ ou $q \times n$ ou $q \times m$	representam as ordens das matrizes A , B , G , C e D

Termos utilizados para o Filtro de *Kalman*

E	média
V	variância
Q	matriz covariância do ruído do sistema
R	matriz covariância do ruído da medida
Y	vetor de saída estocástico
Φ	matriz de transição de estados
P	matriz covariância do erro
K	matriz ganho de <i>Kalman</i>
I	matriz identidade
ξ	ruídos do sistema
η	ruídos da medida

Índices Sobrescritos

D	sistema determinístico
E	sistema estocástico

Termos utilizados no Núcleo do Algoritmo de Kalman - Figura 5.5

K_k	matriz ganho de <i>Kalman</i> de k medidas
\hat{x}_k^-	estado estimado <i>a priori</i> de k medidas
\hat{x}_k	estado estimado <i>a posteriori</i> de k medidas
\hat{x}_{k-1}	estado estimado <i>a posteriori</i> de $k - 1$ medidas
P_k^-	matriz covariância do erro de estimação <i>a priori</i> de k medidas
P_k	matriz covariância do erro de estimação <i>a posteriori</i> de k medidas
P_{k-1}	matriz covariância do erro de estimação <i>a posteriori</i> de $k - 1$ medidas
y_k	valores da medição
G	matriz ruído do sistema
I	matriz identidade

Termos utilizados para Considerações de Projetos SMI-TR

D_s	desempenho do sistema
T_{ea}	tempo de execução da aplicação
T_{ep}	tempo de execução no processador
N_{cc}	número de ciclos de <i>clock</i>
T_{cc}	tempo do ciclo de <i>clock</i>
N_{cpp}	número de ciclos de <i>clock</i> do processador para o programa
N_{ip}	número de instruções pelo programa
N_{cci}	número médio de ciclos de <i>clock</i> por instrução

D_k	conjunto de implementações
E	valor estimado
M	valor medido
μ_{ij}	resultado da análise entre os valores estimados e medidos
\mathfrak{S}	fidelidade da medição indireta
n	número de valores medidos e estimados analisados
X	grandeza medida diretamente
$\Delta a, \Delta b, \Delta c$	limite de erro dos valores das grandezas das quais X é dependente
ΔX	limite de erro da grandeza medida diretamente
$x(t)$	estado do filtro
u, w	entradas do sistema
y	saída do sistema
z	valor medido no sistema
\hat{z}	valor estimado(saída do filtro)
\hat{x}	estimação do estado
\tilde{z}	erro de estimação
x_0	estado inicial do sistema
σ	desvio padrão

Outros Termos utilizados nos Estudos de Casos

T_0	temperatura ambiente
T_1	temperatura no meio 1
T_2	temperatura no meio 2
m_1	massa do objeto do meio 1
m_2	massa do objeto do meio 2
c_1	calor específico do objeto do meio 1
c_2	calor específico do objeto do meio 2
Q_h	fluxo de calor
k_{20}	condutância térmica do meio 2 para o ambiente
k_{12}	condutância térmica do meio 1 para o meio 2
$u(t)$	excitação do sistema térmico
R	resistência elétrica da estufa
$R_{t,conv,2}$	resistência térmica por convecção no meio 2
$R_{t,cond}$	resistência térmica por condução
$T_{s,1}$	temperatura na superfície do material
e_k^-	erro de estimação <i>a priori</i>
e_k	erro de estimação <i>a posteriori</i>
y	medidas do sistema
\hat{x}_k^-	estado estimado <i>a priori</i>
\hat{x}_k	estado estimado <i>a posteriori</i>
P_k^-	covariância do erro de estimação <i>a priori</i>
P_k	covariância do erro de estimação <i>a posteriori</i>
y_k^-	predição da medida
\bar{E}	erro médio
T_{1d}	temperatura medida diretamente no meio 1
T_{1i}	temperatura medida indiretamente no meio 1
n	número de medições da temperatura

g	aceleração gravitacional
M	massa do foguete
MFo	massa inicial do combustível
MT	massa total
TQ	taxa de queima do combustível
F	empuxo na direção do eixo do foguete provocado pela queima do combustível
t_v	tempo de vôo na vertical
X, Y, Z	eixos no espaço tridimensional
d, az, el	coordenadas da distância, azimute e elevação do veículo aeroespacial
Ac	aceleração na coordenada Z
V	velocidade na coordenada Z
rm	ruído da medida de posição
ra	ruído da aceleração
y_m	sinais medidos
x_e	vetor de estados
N	número de amostras de sinais do veículo aeroespacial
v_s	tensão no estator
v_r	tensão no rotor
i_s	corrente no estator
R_s	resistência do estator
R_r	resistência do rotor
λ_s	fluxo no estator
R_r	resistência do rotor
i_r	corrente no rotor
λ_r	fluxo no rotor
ω_r	velocidade rotórica
E	energia do erro
λ_a ou λ_{ar}	vetor fluxo atual no rotor
λ_d ou λ_{dr}	vetor fluxo desejado no rotor

L_s	indutância no estator
L_r	indutância no rotor
L_m	indutância de magnetização
σ	coeficiente de dispersão
T_r	constante de tempo do rotor
T	tempo de amostragem

Lista de Figuras

1.1	Etapas para a Identificação de Processos.	3
1.2	Diagrama em blocos para a concepção de um <i>SMI</i>	4
1.3	Instrumento de Medição Direta de uma Grandeza.	6
1.4	Instrumento de Medição Indireta de uma Grandeza.	6
1.5	Excitação e resposta de uma estufa.	11
1.6	Excitação e resposta de um objeto dentro de uma estufa.	11
2.1	Etapas para o desenvolvimento de um <i>SMI</i>	30
3.1	Diagrama de unidade da <i>RNA</i> de <i>McCulloch - Pitts</i>	34
3.2	Estrutura de um <i>MLP</i>	37
3.3	Diagrama para treinamento do <i>Perceptron</i>	39
3.4	Representação da ativação da rede e treinamento por erro de propagação.	39
3.5	Diagrama em blocos do modelo direto para estimação via redes neuronais.	42
3.6	a) Filtragem Digital; b) Predição Ótima; c) Filtragem Digital Suave.	51
4.1	Graus de rigor e criticalidade de um sistema em tempo real	61
4.2	Configuração para filtragem.	67
4.3	Fatores que interferem no resultado de uma medição indireta.	72
4.4	Pseudocódigo do núcleo do algoritmo de <i>Kalman</i>	80
4.5	Etapas do filtro de <i>Kalman</i> : predição e correção.	80
5.1	Filtro de <i>Kalman</i> em tempo real.	90
5.2	Representação de um sistema térmico.	96
5.3	Representação da estufa a energia elétrica.	98

5.4	a) Distribuição de temperatura através do material; b) Circuito térmico equivalente.	100
5.5	Procedimentos utilizados pela filtragem de <i>Kalman</i> para a medição indireta da temperatura.	105
5.6	Evolução do ganho de <i>Kalman</i>	110
5.7	Diagrama em blocos da medição indireta de temperatura no meio 1.	111
5.8	Cubo de alumínio utilizado.	112
5.9	Comportamento das temperaturas	114
5.10	Circuito elétrico equivalente.	117
5.11	Estrutura da rede neuronal para estimar ω_r	121
5.12	Dois neurônios com três entradas e duas saídas lineares.	124
5.13	Algoritmo de retropropagação modificado para treinar a <i>RNA</i>	126
5.14	Núcleos do observador neuronal de fluxo/velocidade rotórica.	127
A.1	Trajetória de um veículo aeroespacial.	168
A.2	Fluxograma do filtro de <i>Kalman</i> implementado.	171
A.3	Posição do veículo aeroespacial.	172
A.4	Velocidade do veículo aeroespacial.	173
A.5	Aceleração do veículo aeroespacial.	174
A.6	Curvas de ganho de <i>Kalman</i> para posição, velocidade e aceleração no modelo.	175
B.1	Tensão no estator obtida por simulação.	183
B.2	Velocidade rotórica medida indiretamente.	184

Capítulo 1

Introdução

A observação dos fenômenos da natureza é uma grande fonte de inspiração para os pesquisadores. O caminho mais imediato e natural para tratar e compreender um problema físico se inicia com a caracterização e a descrição por meio de um modelo que, na maioria das áreas, utiliza a matemática como ferramenta para tal finalidade. Os modelos matemáticos empregam métodos e técnicas que envolvem, no sentido mais amplo, a resolução de equações, cujas soluções subsidiam a comparação de resultados, por meio de implementações e de observações, com o fenômeno investigado. Na prática, o modelo é tanto mais adequado à descrição do fenômeno quanto menor for o desvio da solução do problema físico com relação ao mundo real.

Ao longo dos últimos três decênios do Século XX e do primeiro do século XXI vêm-se obtendo modelos de sistemas com o objetivo de medir indiretamente uma grandeza. O conhecimento de uma ou mais grandezas por meio indireto é útil em diversas situações nas quais se tem dificuldade para medição. A conotação dada à expressão *medição*

indireta, nesta tese, se refere aos procedimentos utilizados para se obter o valor de uma grandeza desconhecida por meio de métodos computacionais, geralmente complexos, tais como, *Redes Neurais Artificiais* (RNA), *Filtro de Kalman* (FK) e *Transformada Rápida de Fourier* (TRF).

A obtenção da informação a partir de observações é fundamental para modelos de sistemas dinâmicos. Estas observações podem ser as medidas de sinais provenientes de dispositivos/sistemas de medição que, nesta tese, são classificados em *sistemas de medição indireta ou direta*. Esta classificação tem como referência a maneira como é obtido o mensurando. Se o dispositivo gera um sinal elétrico a partir de um sensor que tem por objetivo detectar variações da grandeza de interesse, o sistema de medição é denominado *sistema de medição direta* (SMD). Caso o mensurando seja obtido a partir de grandezas que possuam alguma relação com a grandeza que se deseja conhecer, o sistema é denominado *sistema de medição indireta* (SMI). Por exemplo, pode-se inferir a velocidade do eixo do rotor de uma máquina de indução a partir de medições de tensão e corrente estatóricas. A medição da velocidade também pode ser obtida por meio de um tacômetro que converte as rotações do eixo em um sinal elétrico, sendo caracterizado como um dispositivo de medição direta, pois converte a grandeza de interesse em um sinal elétrico a partir de um sensor que está diretamente relacionado com a grandeza a ser medida. Para o caso do *SMI*, a velocidade pode ser obtida a partir de grandezas elétricas que, indiretamente, refletem as variações da grandeza velocidade.

Para se chegar a um modelo de um *SMI* pode-se partir do processo de identificação

sugerido por *J. Schoukens e Pintelon* [1], ilustrado na Figura 1.1. Na Figura 1.2, extraída da Figura 1.1, tem-se o diagrama em blocos para a concepção de um *SMI*. O primeiro bloco especificado por *sistema real* se refere aos diversos componentes que interagem entre si, em conformidade com as leis da Natureza. Em seguida, a *medição indireta* é caracterizada pela escassez de recursos para se medir diretamente uma grandeza desse sistema. O bloco *modelo matemático* se refere à representação do sistema, conforme o propósito do modelo do sistema dinâmico. No bloco *métodos* tem-se o uso do modelo para determinar a medição indireta por meio de algoritmos computacionais.

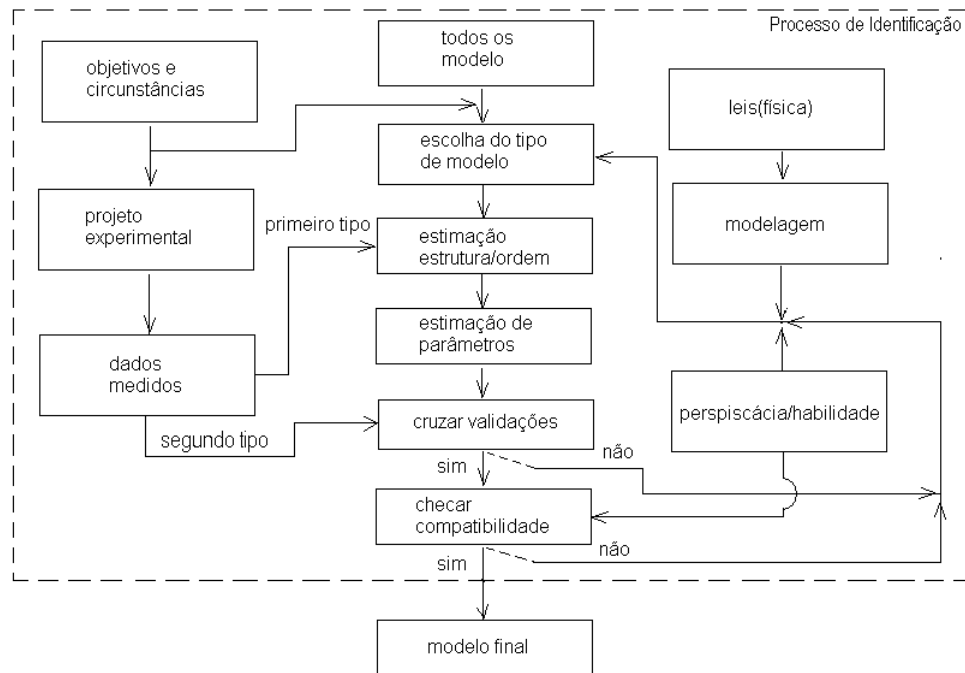


Figura 1.1: Etapas para a Identificação de Processos.

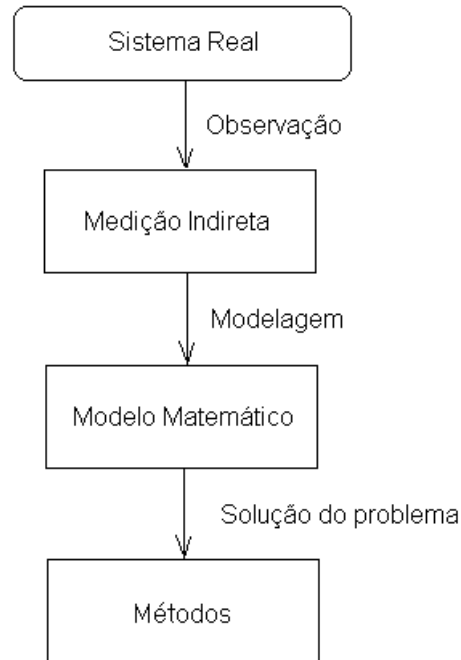


Figura 1.2: Diagrama em blocos para a concepção de um *SMI*.

Neste Capítulo são evidenciadas a contextualização e a motivação do corpo desta tese, item 1.1; a declaração do problema, item 1.2, sendo este o item que motivou a realização deste trabalho, direcionando o desenvolvimento do texto. Em seguida, no item 1.3, são descritos os objetivos da tese, separando-os em objetivos gerais e específicos; a metodologia é apresentada no item 1.4 e, por fim, a estrutura, contendo a divisão do trabalho no item 1.5.

1.1 Contexto e Motivação

Medição Indireta

A medição indireta tem sido normalmente utilizada para solucionar problemas isolados que, na maioria das vezes, podem ser agrupados para apresentarem soluções diversificadas por um único modelo do sistema. Dentre as soluções de problemas isolados encontram-se o uso de alguns algoritmos, que são mencionados nesta tese para estimar parâmetros de máquinas de indução, como em [2]-[19]; filtrar sinais e estimar a velocidade de um veículo aeroespacial, conforme [20]; além de outros *SMI*, tais como em [21]-[25].

Os *SMI* neste trabalho são sistemas em que os algoritmos podem ser processados a partir de sinais obtidos em tempo real ou não, e possuem as seguintes características básicas em comum com os *SMD*:

- a medição ocorre a partir da natureza física de um mensurando;
- os sinais do mensurando são apresentados na entrada do sistema e analisados;
- o método de medição é escolhido de acordo com a aplicação.

Nas Figuras 1.3 e 1.4 têm-se os diagramas em blocos ilustrativos desses dois conceitos de *SMD* e *SMI*. O bloco *Elementos funcionais* presente nesses diagramas se refere à todas as funções básicas consideradas necessárias para a descrição de um instrumento de medição direta, como exemplo, elemento sensível e elemento de conversão. O bloco *mensurando* se refere à grandeza que é submetida à medição. Na Figura 1.4 há a

representação de dois mensurandos de um sistema: *mensurando A* e *mensurando B*. A medição da grandeza referente ao *mensurando A* do sistema é obtida indiretamente. Neste caso, a grandeza referente ao *mensurando B*, desse mesmo sistema, que está sendo submetida diretamente ao bloco *Elementos funcionais* deve ter uma relação com o *mensurando A*. Ainda em relação à Figura 1.4, o bloco *Cálculo computacional baseado no modelo do sistema* se refere ao *método de medição indireta* que é escolhido e desenvolvido para se conhecer a grandeza do *mensurando A*.

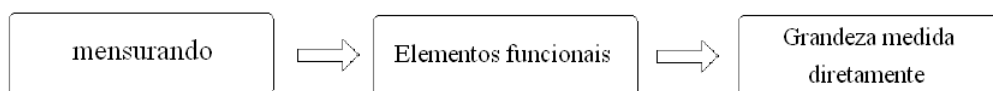


Figura 1.3: Instrumento de Medição Direta de uma Grandeza.

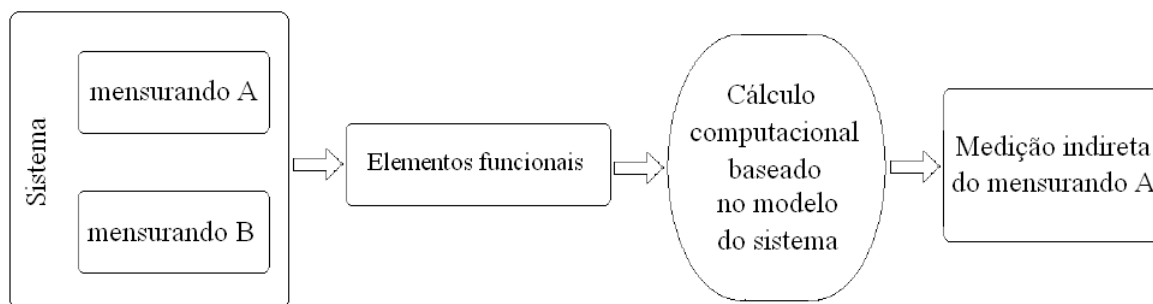


Figura 1.4: Instrumento de Medição Indireta de uma Grandeza.

Verifica-se que a medição direta de uma ou mais grandezas é intermediária para a medição indireta com base em modelo. Para o cálculo computacional, baseado em modelo, é necessário ter os valores das grandezas que são medidas diretamente e que

possuam alguma relação com a grandeza que se deseja conhecer, bem como, alguns parâmetros do sistema. A partir destes, é desenvolvido e processado o algoritmo que fornece o valor estimado.

Normalmente, os erros estão associados a qualquer tipo de medição. Nas medições indiretas esses erros estão presentes nas variáveis utilizadas no cálculo computacional para determinação da grandeza de interesse. Há uma propagação da incerteza da medição direta durante o cálculo computacional até a estimação do valor da grandeza. O que não se pode evitar é que a medida não seja exata, uma vez que o valor real da grandeza de entrada não é exato. Como existem incertezas na função que relaciona a grandeza a ser medida com a grandeza secundária, então, o valor medido indiretamente não será exato. Nesta acepção, para realizar a medição indireta foram necessários alguns parâmetros adicionais ou especiais, como ajustes, correções e compensação que contribuíram para que o resultado da medição fosse o mais próximo possível do real. No contexto desta tese, verifica-se que existem diversas etapas no processo da medição indireta que são passíveis de gerar erros.

A seguir, tem-se uma breve descrição dos *SMI* de alguns trabalhos já desenvolvidos e as estratégias utilizadas para o desenvolvimento dos *SMI* nos estudos de casos apresentados nesta tese.

Sistema de Medição Indireta

Segundo *Moudden* [26], a concepção de um *SMI* baseia-se em encontrar resposta(s)

para a estimação de uma grandeza a que se tem interesse a partir de um ou mais observadores que podem ser representados por sensores de diferentes naturezas, por exemplo. O objetivo é obter um sistema, cuja(s) saída(s) esteja(m) relacionada(s) à grandeza de interesse. A qualidade do *SMI* proposto pode ser avaliada de acordo com a aplicação e a função do estimador desenvolvido. O desempenho do estimador, porém, apresenta algumas limitações, tais como:

- as informações iniciais do(s) observador(es) sobre a grandeza de interesse;
- a experiência em obter essas informações (escolha do método de medição indireta);
- o algoritmo desenvolvido para extrair de modo eficiente tais informações.

Dentre os requisitos citados, o que possui maior impacto no desenvolvimento do sistema é a escolha do método de medição indireta a ser utilizado, de modo que seja possível extrair adequadamente as informações das grandezas de entrada.

A escolha de um e/ou outro método requer certa habilidade e experiência na coleta das informações que servirão de entrada. Quando não se pode aplicar diretamente um e/ou outro algoritmo já estabelecido, é necessário desenvolver um modelo matemático do sistema para se estimar o valor da grandeza em que se tem interesse. Esse modelo matemático, por sua vez, requer um levantamento detalhado do sistema.

O problema da escolha do método constitui uma das limitações para o desenvolvimento do *SMI*. Uma escolha inadequada pode levar a um resultado indesejado, tempo de processamento elevado e, conseqüentemente, resposta lenta.

Uma área que possui muitas aplicações da medição indireta é a robótica, na qual muitos pesquisadores enfocam, prioritariamente, o desenvolvimento de algoritmos para planejamento do movimento e navegação, manipulação de objetos e visão computacional por meio da computação reconfigurável. No trabalho de *Aragão* [27], tem-se a proposta de um robô móvel implementado com a tecnologia de computação reconfigurável. Ela abrange estudos nos seguintes tópicos: projeto de lógica digital, modelagem de sistemas digitais com *VHDL*¹, exploração de paralelismo em arquiteturas de computadores, técnicas de inteligência artificial e *RNA*² aplicadas à robótica, teoria de controle, lógica *fuzzy*, *kernel* de sistemas operacionais de tempo real e, finalmente, a computação reconfigurável propriamente dita.

Bucci[28] apresenta um *SMI* em tempo real para um motor de indução. O algoritmo de estimação da amplitude e da frequência fundamental da tensão de linha no motor de indução é executado em tempo real. O método de medição indireta usado é o baseado em *FK*. Inicialmente, foram realizadas algumas considerações de *hardware* e *software* em relação à arquitetura para as exigências de tempo real. O comportamento dinâmico da medição é monitorado para verificação do desempenho em relação à exatidão e velocidade de execução.

São desenvolvidos algoritmos baseados na *TRF* para medição indireta em tempo real no processamento de sinais e imagens por *Uzun* [29]. Neste caso, a implemen-

¹ *Very High Speed Integrated Circuit Hardware Description Language* - linguagem usada para facilitar o projeto/concepção de circuitos digitais em *FPGA* e *ASIC*.

² Redes Neurais Artificiais - buscam mimetizar o sistema nervoso por meio de elementos processadores simples denominados neurônios.

tação foi realizada em uma arquitetura reconfigurável, obtendo-se alto desempenho e custo reduzido. Em *Bonato* [30], o processamento de imagens coloridas aplicado a robôs móveis é realizado por meio do método de medição indireta utilizando *Rede Neuronal Booleana* para o reconhecimento de padrões aplicando uma abordagem diferente para a construção da *Rede Neuronal*. A plataforma é desenvolvida e implementada em *hardware* reconfigurável dentro do conceito de sistemas *SoC*.

Nos *SMI* desenvolvidos nesta tese, as leis fundamentais da Física são apresentadas para cada sistema utilizado nos estudo de casos, apresentados nos itens de 5.1 a 5.5; bem como são definidos os parâmetros associados a cada um desses sistemas. Como resultado, tem-se uma equação diferencial ou um conjunto de equações diferenciais que descrevem os sistemas e as equações de estado correspondentes. A análise do comportamento de cada um dos sistemas em estudo e o desenvolvimento das equações desses sistemas são reforçadas por meio de diagramas em blocos.

No sistema térmico apresentado no item 5.2, considera-se que a excitação do sistema é a energia elétrica; a resposta está em função da entrada. A fonte de energia térmica é uma resistência elétrica, a temperatura interna aumenta com o tempo, fornecendo uma resposta escalar, cuja dimensão é expressa em °C. Para um objeto colocado dentro da estufa, a excitação pode ser representada pela energia térmica aplicada ao objeto, também escalar, e é expressa em calorias. Assim, a temperatura no interior do objeto depende da energia térmica aplicada. Havendo variação da energia térmica, haverá variação tanto na temperatura interna da estufa quanto na temperatura no interior do

objeto. A resposta, para o objeto, é caracterizada pela temperatura no seu interior. Dessa forma, mantendo-se o conceito de sistema, há uma relação entre a quantidade de energia térmica aplicada ao objeto e a temperatura do sistema (Figuras 1.5 e 1.6).

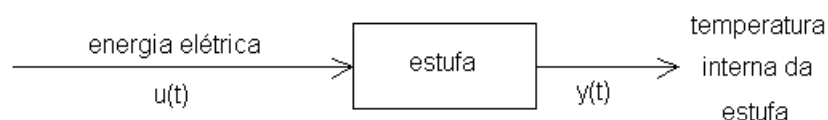


Figura 1.5: Excitação e resposta de uma estufa.

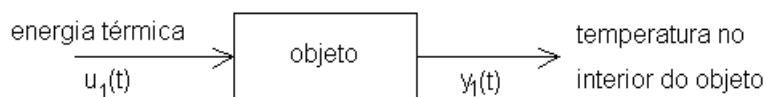


Figura 1.6: Excitação e resposta de um objeto dentro de uma estufa.

Já no sistema elétrico utilizado nos Estudos de Casos dos itens 5.5.1 e 5.5.2 para Medição Indireta da Velocidade Rotórica de um Motor de Indução, os sinais de excitação são caracterizados pelas tensões e correntes estatóricas e os sinais de resposta do sistema, são representados pelos fluxos rotóricos. No desenvolvimento do modelo matemático do sistema envolvendo o veículo aeroespacial, apresentado no item 5.1, as excitações são representadas pelos sinais vindos do radar na forma das coordenadas azimute, distância e elevação com relação ao sistema cartesiano de coordenadas do referido radar de um veículo aeroespacial que é monitorado pelo radar; as respostas de saída do sistema são dadas pela posição e velocidade do veículo aeroespacial.

1.2 Descrição do Problema

Os dispositivos de medição direta existentes não são suficientes para atender solicitações em que se tem dificuldades para realizar medições. Essa lacuna no âmbito das medições é em decorrência de diversos fatores, tais como: ausência de sensores reais, segurança, custos e impossibilidade de realizar a medição. Outro fator importante é a necessidade de se conhecer o valor de grandezas desconhecidas para a teoria de controle. Embora os sistemas de controle reais sejam geralmente não-lineares, eles podem ser aproximados por meio de modelos matemáticos lineares; é possível utilizar ferramentas matemáticas para o desenvolvimento de um sistema de medição indireta agregado ao sistema de controle de modo que este forneça informações do(s) valor(es) da(s) grandeza(s) necessária(s) para o controle.

1.3 Objetivos da Tese

Neste item são mencionados os objetivos geral e os específicos desta tese. No objetivo geral têm-se o alvo essencial deste trabalho que é proposto para a solução do problema. Nos específicos são apresentados os aspectos analisados que apresentam maior concretude e delimitam a investigação desta tese.

1.3.1 Objetivo Geral

O objetivo desta tese é desenvolver procedimentos para sistemas de medição indireta em tempo real. Com base nesses procedimentos, é possível se verificar e implementar sistemas de medição indireta, buscando apresentar as ferramentas mais adequadas para projeto e análise de modelos lineares e não-lineares de sistemas dinâmicos, as métricas de desempenho e as considerações de otimalidade do *SMI*.

Pretende-se assegurar a viabilidade do desenvolvimento de algoritmos para medição indireta por meio de ferramentas matemáticas de alto nível e que as implementações desses algoritmos em sistemas embarcados ou mesmo microcontroladores já disponíveis tendam a substituir os dispositivos para medição direta de grandezas quando este procedimento for necessário. Como benefícios, tem-se a redução de custos por serem agregados aos sistemas de controle para fornecimento de informações necessárias ao controle de um determinado sistema.

Para validação dos procedimentos, são apresentados neste trabalho quatro estudos de casos para aplicações em sistemas lineares e não-lineares. As ferramentas matemáticas usadas para desenvolver os algoritmos dos sistemas de medição indireta são as *RNA* e o *FK*.

1.3.2 Objetivos Específicos

Os objetivos específicos desta tese são:

- desenvolver métodos/procedimentos para a medição indireta por meio de filtragem ótima estocástica e redes neurais artificiais em sistemas dinâmicos;
- direcionar a teoria desenvolvida por *Kalman* para medição direta e indireta em sistemas;
- formular as incertezas em medição indireta;
- apresentar o método computacional mais adequado para sistemas não-lineares;
- descrever uma taxonomia dos sistemas de medição indireta;
- apresentar métricas de desempenho de *hardware* e *software* que influenciam na qualidade, exatidão e tempo de resposta de uma medição indireta.

1.4 Metodologia Utilizada

- Revisão Bibliográfica

Inicialmente, foi realizada uma revisão bibliográfica a respeito dos métodos que são utilizados para estimação de parâmetros de modelos em sistemas lineares e não-lineares em tempo real ou não. Esta revisão teve como objetivo estudar as ferramentas matemáticas mais adequadas aos problemas relacionados com esta tese e estabelecer o conceito de medição indireta que é abordado.

- Abordagem da Medição Indireta

Nesta etapa, foi estabelecido o conceito de medição indireta empregado nesta tese a partir de um levantamento sobre o que se considera sobre as definições de medição indireta. No estudo de alguns conceitos abordados em Metrologia e outros apresentados por alguns autores, verificou-se que existia a necessidade de se mencionar nesta tese a conotação que está sendo dada em relação à expressão medição indireta.

- Modelos dos Sistemas Dinâmicos

Em cada um dos sistemas foi desenvolvido o modelo; levou-se em consideração as grandezas de entrada que podem fazer uma correlação com a grandeza que se deseja medir indiretamente; procurou-se obter um modelo que corresponda ao sistema real, sem a inclusão de parâmetros desnecessários ao cálculo computacional e, ao mesmo tempo, incluindo parâmetros extremamente úteis proporcionando um resultado da medição mais exato.

- Implementação do *SMI*

Após o desenvolvimento dos modelos dos sistemas e a escolha do método, realizou-se a simulação de forma a verificar o desempenho tanto dos parâmetros do método empregado quanto do algoritmo desenvolvido. Uma vez verificando a convergência do algoritmo, o *SMI* foi implementado usando uma arquitetura reconfigurável.

1.5 Estrutura do Documento

Para abordar o problema da medição indireta, este documento está dividido em quatro partes, sendo 7 Capítulos, 2 Apêndices e 1 Anexo:

Parte 1: nesta parte, Introdução (Capítulo 1), apresenta-se a relação entre a medição direta e a medição indireta, especificando-se as características comuns entre ambas as medições e a necessidade da medição direta como intermediária da medição indireta. Identificam-se, inicialmente, algumas limitações para o bom desempenho de um sistema de medição indireta. O termo *SMI* é enfatizado, mencionando a sua concepção e uma breve abordagem sobre o desempenho do estimador. São mencionados alguns textos científicos referentes às medições indiretas implementadas em *hardware*. Os métodos de medição indireta utilizados nos estudos de casos são apresentados brevemente. Tem-se ainda nesta parte, a descrição do problema da tese, os objetivos gerais e específicos e a metodologia utilizada para o desenvolvimento deste trabalho.

Parte 2: esta parte compreende os Capítulos 2, 3 e 4. No Capítulo 2, apresenta-se, resumidamente, os diversos métodos de medição utilizados, dando-se ênfase aos sistemas de medição indireta. No Capítulo 3 faz-se a abordagem de uma taxonomia dos sistemas de medição indireta, bem como as considerações e as etapas necessárias para o desenvolvimento de um *SMI*, apresentadas no Capítulo 4.

Parte 3: esta parte se refere aos estudos de casos desenvolvidos no Capítulo 5. São realizados quatro estudos de casos, abordando os procedimentos utilizados para se obter indiretamente os valores de grandezas desconhecidas, conforme cada aplicação. No Estudo de Caso 5.1, os sinais de trajetórias de um veículo aeroespacial fornecidos pelos radares são filtrados e, posteriormente, é realizada a medição indireta da velocidade do veículo aeroespacial. Ainda em relação a esse estudo, no item 4.7 faz-se uma abordagem sobre o sistema embarcado ótimo de *Kalman*. No Estudo de Caso 5.2 realiza-se a medição indireta da temperatura no interior de um objeto que se encontra dentro de uma estufa. Nesses dois primeiros estudos de casos, utiliza-se o método de medição indireta filtro de *Kalman*. No Estudo de Caso 5.5.1 tem-se o tratamento da obtenção do valor da velocidade rotórica de uma máquina de indução por meio do método *RNA*. No item 5.5.2, realiza-se o desenvolvimento da robustez da rede neuronal para a aplicação do Estudo de Caso 5.5.1.

Parte 4: nesta última parte têm-se a Conclusão, item 6, na qual são mencionadas as contribuições e conclusões desta tese. Tem-se também as Sugestões para Trabalhos Futuros, item 7, apresentando sugestões de linhas de pesquisas como prosseguimento ou desdobramento dos temas expostos nesta tese. Ainda nesta parte tem-se o Apêndice A e B no qual é abordado o Modelo do Sistema de Rastreamento e o Desenvolvimento da Regra Delta para os Estudos de Casos 5.5.1 e 5.5.2. Há

um anexo, Anexo *A*, no qual são apresentadas as características do sensor de temperatura utilizado no Estudo de Caso 5.2.

Capítulo 2

Sistema de Medição Indireta

De acordo com o *Vocabulário Internacional de Termos Fundamentais e Gerais de Metrologia* (VIM), [31], a medição é um conjunto de operações que tem por objetivo determinar o valor de uma grandeza. No *VIM* não são especificados os conceitos de medição direta e indireta. Entretanto, historicamente, sabe-se que a medição direta de uma grandeza é aquela na qual o valor medido é obtido a partir da leitura direta de um instrumento de medição. Já na medição indireta, o valor medido é obtido por cálculos a partir de outras grandezas medidas diretamente. Porém, a abordagem dada neste trabalho sobre o *SMI* se refere às medições que são realizadas em sistemas, nos quais o processo para se obter o valor de uma grandeza indiretamente, leva em consideração que os valores especificados como entrada sejam oriundos de uma medição direta, ou seja, de grandezas que foram medidas diretamente e que o cálculo utilizado seja baseado em ferramentas de alto nível, por exemplo, *RNA*. Os critérios para se utilizar um *SMI* seguindo esta abordagem estão relacionados com a impossibilidade de medição direta,

já que em algumas situações a medição direta não pode ser realizada por questão de segurança, tempo, acesso ou ainda de inexistência de sensor ou transdutor. A medição indireta também pode ser empregada para a obtenção de resultados mais confiáveis que os resultados obtidos por métodos convencionais ou ainda para reduzir um conjunto de procedimentos de modo a se obter o valor de uma grandeza ou custo, considerando que algumas medições diretas podem ser muito dispendiosas.

Nesse contexto, tem-se que a concepção de um *SMI* baseia-se em encontrar resposta(s) para a estimação de uma grandeza a que se tem interesse a partir de um ou mais observadores que podem ser representados por sensores de diferentes naturezas [26]. O objetivo, então, é obter um sistema, cuja(s) saída(s) esteja(m) relacionada(s) à grandeza de interesse. A verificação da qualidade de um *SMI* leva em consideração o tipo de aplicação, o método de medição indireta escolhido e os requisitos de *hardware*. Existem algumas limitações que interferem no bom desempenho do *SMI* desenvolvido, tais como:

- informações iniciais do(s) observador(es) sobre a grandeza de interesse;
- experiência em obter essas informações (escolha do método de medição indireta);
- o algoritmo desenvolvido para extrair de modo eficiente tais informações;
- operação para uma dada banda de frequência.

Dentre os requisitos citados, o que possui maior impacto no desenvolvimento do *SMI*

é a escolha do método de medição indireta a ser utilizado de modo que seja possível extrair adequadamente as informações das grandezas de entrada. Aliada a essa escolha está o desenvolvimento do modelo do sistema que deverá representar o mais próximo possível o sistema real, conforme a aplicação.

2.1 Medição Clássica - Instrumentação Real

A medição é empregada em diversas áreas e, neste trabalho, torna-se necessário enfatizar os tipos de medição. Conforme mencionado anteriormente, a medição pode ser realizada de forma direta e indireta. Estabelecer a metodologia entre uma forma e outra gera, em parte, certa discussão quando se analisa intensamente o método de obtenção da grandeza medida. *Cook*, [32], considera que uma medição é direta quando há um processo de comparação direta. Essa comparação direta pode ser realizada por:

- comparação entre duas grandezas que é determinada pelo senso humano, e então a grandeza tem que ser uma que esteja dentro das capacidades da experiência humana;
- a grandeza a ser medida deve ser comparada com outra grandeza da mesma dimensão, considerada padrão.

Busca-se, neste Capítulo, não somente mencionar sobre os outros métodos de medição, mas também focar a abordagem de medição dada neste trabalho. Como reforço,

pode-se, ainda, mencionar que a medição indireta que está sendo considerada trata-se da medição de grandezas por meio de ferramentas matemáticas que já estão estabelecidas ou que venham a ser desenvolvidas para aplicação em sistemas dinâmicos. As ferramentas matemáticas utilizadas como estimadores, também podem ser consideradas como sensores virtuais, dependendo da complexidade do sistema e da ferramenta. O objetivo é conhecer o valor da(s) grandeza(s) que se tem dificuldade(s) em medir em sistemas dinâmicos a partir da relação entre o(s) valor(es) da(s) grandeza(s) medida(s) diretamente e o(s) valor(es) da(s) grandeza(s) que se deseja medir.

Levando-se em consideração o conceito já estabelecido da medição clássica, a instrumentação real consiste de um conjunto de técnicas para o projeto de desenvolvimento e construção de equipamentos eletrônicos. Esses equipamentos eletrônicos manipulam sinais elétricos que representam grandezas físicas. Normalmente, possuem uma entrada de leitura de dados, que é um transdutor, a interface entre o mundo e o instrumento, uma parte que faz o processamento do sinal propriamente dito e uma saída de dados, o terminal, que é outra interface entre o sistema e o ser humano [33].

2.2 Sensor Virtual

O modelo matemático de sistemas pode reduzir a necessidade de se realizar medições diretas em dispositivos. Uma das principais estratégias é obter um modelo completo

que represente o sistema de forma a garantir que o resultado obtido seja o mais próximo possível do desejado, ou seja, que o resultado seja mais exato. A partir de alguns modelos é possível obter a estimação de diversos parâmetros. A estimação dos parâmetros está correlacionada com outro dispositivo, o sensor virtual. O sensor virtual em um *SMI*, pode ser considerado como o método de medição indireta escolhido, levando-se em consideração o quão complexo é esse método.

O termo denominado por sensor virtual se refere aos dispositivos usados para reconstruir dados que foram perdidos, que se tem dificuldades em obter ou ainda para os quais inexistem um sensor real. Os sensores virtuais são, frequentemente, caixas pretas (por exemplo, *RNA*) e são especialmente valiosos quando o processo físico é muito complexo ou incerto, enquanto houver bastantes dados para serem processados/treinados [34], [35].

Os algoritmos das *RNA* desenvolvidos nesta tese atuam como um sensor virtual que, a partir da aprendizagem da rede, fornecem o valor da velocidade rotórica de uma máquina de indução, por exemplo. *Habtom*, [36], utiliza a *RNA* como sensor virtual onde as informações disponíveis são insuficientes para treinar a rede. Outras aplicações envolvendo sensor virtual podem ser verificadas em [37]-[40].

Os sensores virtuais oferecem propriedades atrativas, tais como:

- representam uma alternativa de baixo custo para dispositivos com preço elevado;
- podem fornecer informações em paralelo com sensores reais, tornando o sistema

mais seguro;

- podem ser implementados em *hardware* já existente;
- contribuem para a estimação em tempo real, superando os atrasos oriundos de sensores reais lentos, melhorando assim, o desempenho em controle.

2.3 Instrumentação Virtual

Há que se mencionar neste item a diferença entre a instrumentação virtual e a simulação por computador. A primeira se refere ao emprego do computador para ampliar a capacidade funcional dos instrumentos. A segunda corresponde ao uso do computador para simular o comportamento de processos, sistemas, dispositivos, meios e materiais. A instrumentação virtual e a simulação por computador surgiram e evoluíram independentemente uma da outra. Na instrumentação virtual, pode-se enviar dados reais em tempo real para um modelo simulado, que processa esses dados e retorna ao sistema físico, por meio de interfaces adequadas, estímulos, sinais e efeitos. Atualmente, em sua grande maioria, a instrumentação virtual está preparada para facilitar a transferência de dados entre sistemas por meio de *GPIB* (*General Purpose Interface Bus*, protocolo desenvolvido pela *HP* que passou a ser denominado padrão pelo *IEEE 488*).

O instrumento virtual é um sistema formado por um computador mais um instrumento de medição ou equipamento de comando (reais), colocados em comunicação. Um

programa executando no computador torna o instrumento ou o controlador acessível ao operador mediante uma interface gráfica de *software*. Essa interface é dotada de botões, chaves, mostradores, indicadores e painéis de exibição de gráficos apresentados como objetos interativos, animados sob ação do operador por meio do apontador do *mouse*. O instrumento conectado ao computador pode ser desde um equipamento completo, como um osciloscópio ou frequencímetro ou mesmo um simples sensor como um termopar ou um extensômetro.

Os botões e indicadores que aparecem na tela do instrumento virtual podem não corresponder aos controles reais do instrumento que está conectado ao computador. Isto é, usando o computador, pode-se ampliar a funcionalidade de um instrumento acrescentando-lhe novas funções executadas pelo computador com as medidas fornecidas pelo instrumento. Por exemplo, se acoplarmos ao computador um osciloscópio digital que não tenha a função de análise espectral, pode-se obter os dados do sinal adquiridos pelo osciloscópio, processá-los no computador usando-se um algoritmo de *TRF* (Transformada Rápida de Fourier) e assim realizar uma análise espectral do sinal. O conjunto osciloscópio digital mais o computador executando a *TRF*, forma um Analisador de Espectro Virtual.

O *HP-VEE*, por exemplo, é um ambiente de instrumentação virtual criado pela *Hewlett-Packard* para trabalhar com seus próprios equipamentos e demais instrumentos compatíveis. A linguagem de programação visual desenvolvida pela *HP* é orientada para o controle de instrumentos, aquisição e análise de dados. Já o *LabVIEW* (*Laboratory*

Virtual Instrument Engineering Workbench) foi criado pela *National Instruments Corporation* como um ambiente de programação voltado ao desenvolvimento de aplicações, realizado por meio de programação visual orientada pelo fluxo de dados, com o foco em instrumentação virtual. O nome *LabVIEW* designa o ambiente de desenvolvimento e a linguagem de programação visual denomina-se *G* (de *Graphics*, numa acepção à linguagem *C*). Tem-se ainda como ferramenta virtual também bastante utilizada o *VeePro*, da *Agilent*, [41]-[43].

2.4 Sistema de Medição Indireta

Para o desenvolvimento de um projeto de medição indireta em sistemas dinâmicos é necessário haver compatibilidade com o Guia da Expressão da Incerteza em Medição (ISO GUM). Dessa maneira, respeitando as normas do *ISO-GUM*, os *SMI* são classificados, segundo a metodologia empregada, em: Métodos por Modelo Matemático, Métodos por Otimização e Métodos Heurísticos. Considera-se que esses três métodos envolvem as ferramentas que podem ser utilizadas para o desenvolvimento de um *SMI*. A escolha de um dos métodos depende da aplicação, da experiência do projetista entre outros fatores que contribuem para que o resultado da medição seja mais próximo do desejado e mais rápido.

2.4.1 Uma Taxonomia dos Sistemas de Medição Indireta

O objetivo deste item é apresentar uma visão do estado da arte sobre os sistemas de medição indireta. A compreensão aprofundada das alternativas existentes para se realizar uma medição indireta pode ser apreendida a partir de um estudo dos diferentes métodos de medição indireta disponíveis.

O primeiro passo para estabelecer uma taxonomia de sistemas de medição indireta é escolher um conjunto adequado de *métodos de medição indireta*. A escolha de um conjunto de métodos de medição indireta deve ter como objetivo a medição ou a estimação de uma ou mais grandezas desconhecidas independentes entre si. A idéia foi atingir a seleção de um conjunto ótimo de métodos de medição indireta, e com isto atingir uma taxonomia completa. Antes, o objetivo foi expor alguns dos mais relevantes métodos de medição indireta e abordar aplicações em medição indireta nos casos de estudo deste trabalho de tese. É importante perceber que o avanço tecnológico implica a revisão constante destes métodos, bem como o acréscimo de novos métodos e mudança na importância relativa de métodos estabelecidos. A seguir, tem-se uma taxonomia dos sistemas de medição indireta formulada sob um ponto de vista metodológico.

1. Métodos por Modelo Matemático

Neste método, busca-se obter um modelo que represente o comportamento de um sistema normalmente por meio de equações diferenciais ou equações de diferenças [44]. Algumas restrições são consideradas de forma que o modelo matemático possa ser simplificado, tais como a linearização do processo e que o sistema seja invariante no tempo. Devido às simplificações, o desenvolvimento de um modelo matemático visando adequá-lo a uma situação real pode não descrever o sistema de forma significativa e realista. Embora essa metodologia seja empregada por não se ter conhecimento ou ainda um método que descreva o sistema, deve-se levar em consideração que outros fatores como fenômenos físicos ou químicos, distúrbios externos e valores inexatos também afetam a exatidão do modelo. Os modelos determinísticos e estocásticos são normalmente utilizados em sistemas de controle robustos, caracterizando-se por levar em consideração perturbações desconhecidas, porém com limites ou amplitudes conhecidas. Nos modelos estocásticos, por exemplo, pode-se acrescentar ruídos aos sinais por meio de parâmetros estocásticos [45].

2. Métodos por Otimização

Os *Métodos por Otimização* são considerados, nesta proposta, como todos os algoritmos já estabelecidos que são utilizados para aplicações em medição indireta.

A escolha ótima de um algoritmo está associada a uma verificação do sistema, se é linear ou não, ou ainda se possui restrições [46], [47]. A seguir tem-se alguns algoritmos que foram aplicados em medição indireta: algoritmos *Genéticos*; algoritmo de *RNA*; algoritmo de *FK*; outros métodos computacionais, tais como: algoritmos evolucionários, algoritmos distribuídos, métodos de "memória mínima", regiões de confiança, minimização de quadráticas, sistemas de equações não lineares, minimização irrestrita e busca linear, minimização unidimensional, restrições lineares, penalização, gradiente reduzido generalizado, métodos sem derivadas, método *Lagrangeano*, métodos homotópicos e minimização global.

3. Métodos Heurísticos

Classifica-se aqui como *Métodos Heurísticos*, os casos de medição indireta em que são utilizados métodos baseados na experiência prévia com regras práticas e estratégias já freqüentemente usadas para se obter o valor de uma determinada grandeza. Neste caso, a restrição de linearidade não é mais necessária ou mesmo relevante, uma vez que a função entrada-saída é descrita ponto a ponto. Em um *SMI heurístico* existe a possibilidade de se construir uma função de entrada *versus* saída e útil na descrição de uma planta ou processo. Neste tipo de medição a ferramenta considerada mais adequada, segundo *Instone*[48], é a lógica *fuzzy*.

2.4.2 Etapas do Projeto de um SMI

Um dos itens a ser mencionado ainda neste Capítulo está relacionado com as etapas necessárias para o projeto de um *SMI*, Figura 2.1. O modelo que representará o sistema real é uma das etapas importantes, pois é por meio dele que será possível especificar quais os parâmetros mais adequados para se obter a grandeza desejada; quais sinais medidos diretamente podem ser utilizados; se é possível fazer uma correlação entre as variáveis de entrada do sistema com as variáveis que se deseja conhecer.

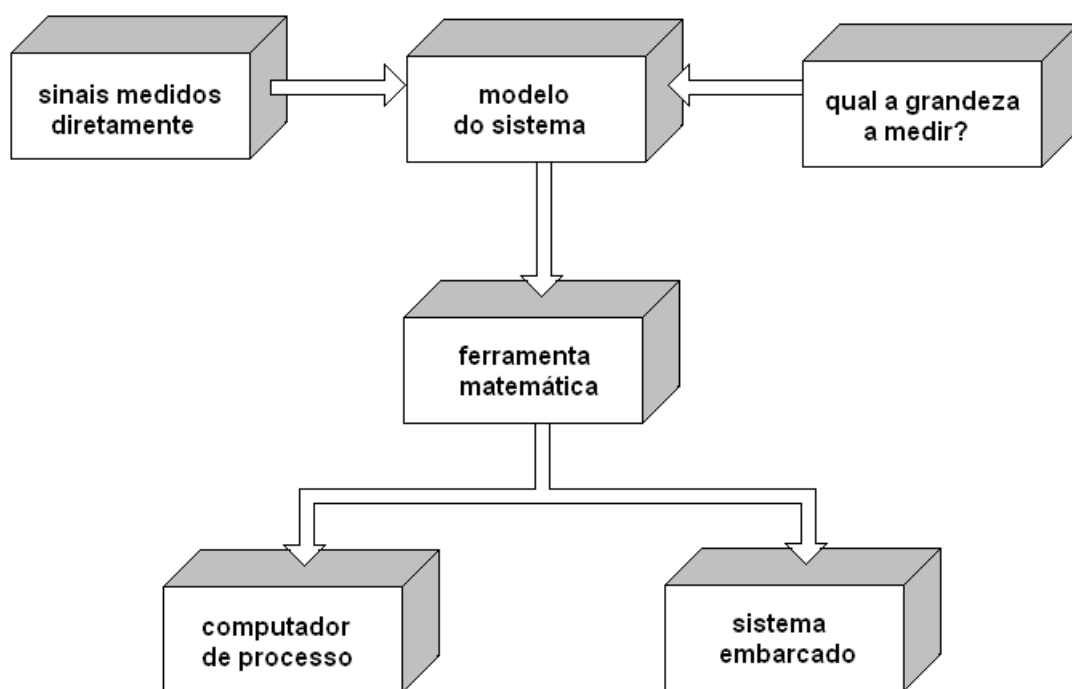


Figura 2.1: Etapas para o desenvolvimento de um *SMI*.

Algumas diretrizes são sugeridas para o desenvolvimento de um sistema de medição indireta:

- conhecer o sistema;
- desenvolver o modelo do sistema real de forma que seja possível realizar a correlação entre os sinais de entrada e os sinais que se deseja conhecer;
- discriminar os sinais de entrada medidos diretamente que podem ter relação com a(s) grandeza(s) que se tem interesse;
- definir a ferramenta matemática que seja mais adequada para a aplicação do *SMI*.

A obtenção de grandezas de forma indireta por meio de *modelos matemáticos*, *métodos por otimização* e *heurísticos* pode ser verificada em diversos trabalhos, tais como [49]-[85].

2.5 Conclusão

Neste Capítulo, são mencionados alguns métodos existentes para aquisição de dados; apresenta-se resumidamente os conceitos de instrumentação real, sensor virtual e instrumentação virtual, além do conceito de medição indireta e uma taxonomia para esse tipo de medição mencionados neste trabalho.

Independente do método de medição empregado para se conhecer o valor de uma grandeza, todos são importantes, tanto na área urbana quanto nas indústrias. Como consequência, a qualidade da medição torna-se uma exigência cada vez mais frequente

em diversos setores: industrial, aeroespacial, naval, médico e hospitalar. Devido à evolução tecnológica, além dos instrumentos reais, é possível se utilizar instrumentos virtuais e sistemas de medição indireta para contribuir com o setor industrial, propiciando e auxiliando nos processos da medição e controle. A partir de modelos matemáticos são desenvolvidas estratégias matemáticas de forma a reduzir a necessidade de se realizar uma medição por meio de dispositivos eletromecânicos ou eletrônicos.

Capítulo 3

Métodos de Medição Indireta

Neste Capítulo são apresentados os métodos de medição indireta utilizados nos estudos de casos desenvolvidos neste trabalho. Em cada um dos métodos é apresentada uma abordagem direcionada para a aplicação. Sendo assim, apresenta-se a teoria matemática para filtragem de *Kalman* utilizada nesta tese: o algoritmo computacional e aplicação para problema de vôo em veículos aeroespaciais, as limitações em tempo real, bem como, o modelo e os procedimentos para se medir indiretamente a temperatura interna de um objeto que se encontra dentro de uma estufa. As *RNA* são utilizadas para a estimação de velocidade de uma máquina de indução.

3.1 Rede Neuronal Artificial

As *RNA* são aplicadas em diversos campos em virtude de uma de suas importantes propriedades: a habilidade de aprender a partir de dados de entrada [86]. As *RNA*

são compostas por várias unidades de processamento. Ilustra-se, resumidamente, na Figura 3.1, a operação de uma unidade de processamento, proposta por *McCulloch e Pitts*(1943). Eles demonstraram que o modelo poderia ser usado para construir qualquer expressão lógica finita. O neurônio de *McCulloch e Pitts* consiste em uma soma ponderada dos pesos, seguido por uma função não linear, denominada, função de ativação ou, originalmente, função limiar. Formalmente, tem-se

$$y_k = \begin{cases} 1, & \text{se } \sum_j w_j x_j \geq 0 \\ 0, & \forall \end{cases} .$$

O comportamento inteligente da *RNA* vem das iterações entre essas unidades de processamento da rede [87].

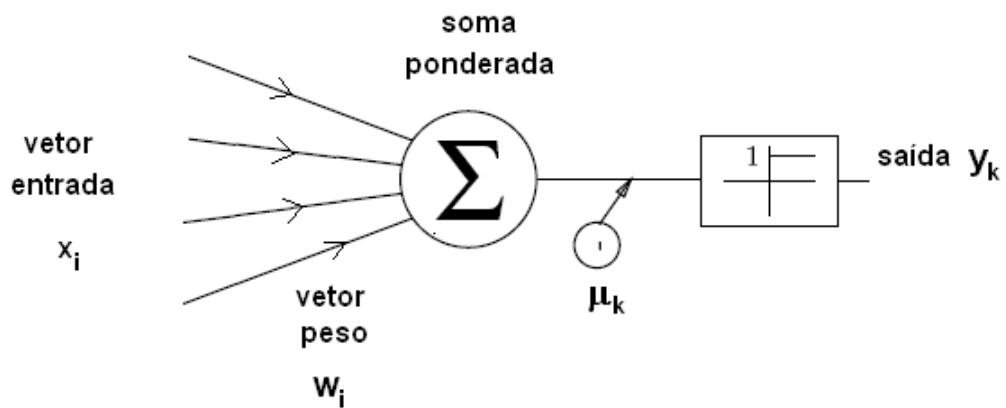


Figura 3.1: Diagrama de unidade da *RNA* de *McCulloch - Pitts*.

De acordo com a Figura 3.1, tem-se: $(x_1, x_2, \dots, x_i) \implies \text{sinais}$ apresentados à

entrada; $(w_1, w_2, \dots, w_i) \implies$ pesos multiplicadores de cada *signal*, indicando a sua influência na saída da unidade; $(w_1x_1 + w_2x_2 + \dots + w_ix_i) \implies$ soma ponderada dos *sinais* que produz um nível de atividade; $\mu_k \implies$ polarização aplicada externamente; $y_k \implies$ resposta de saída, caso o nível de atividade exceda um certo limite.

3.1.1 Redes Biológicas

A estrutura individual dos nodos, a topologia de suas conexões e o comportamento das *RNA* tiveram como base as redes biológicas. O cérebro humano contém cerca de 10^{11} neurônios, sua célula fundamental. Cada um desses neurônios processa e se comunicam com aproximadamente 10^4 neurônios continuamente e em paralelo. A rede de nodos tem a capacidade de reconhecer padrões e relacioná-los, usar e armazenar conhecimento por experiência, além de interpretar observações [88].

O tempo de resposta de um neurônio é aproximadamente 10^{-3} s, muito mais lento do que um computador que é de 10^{-10} s, tempo de resposta de uma porta lógica. O tempo de reconhecimento de uma figura é de cerca de 10^{-1} s. Algumas especulações são abordadas, dentre elas: em relação ao tempo de resposta do neurônio ser relativamente lento, em comparação com a velocidade de processamento de informação. O sistema de aprendizagem biológica é extremamente paralelo e distribuído, entretanto não possui um número muito grande de camadas. Um outro item a mencionar está relacionado com a compatibilidade entre as *RNA* e as redes biológicas: existem incompatibilidades

entre as *RNA* e as unidades do sistema biológico de aprendizagem. A saída de uma unidade artificial é um valor constante, enquanto que as unidades biológicas emitem uma série complexa de sinais.

3.1.2 O Perceptron

O conceito de aprendizado em *RNA* foi introduzido por *Rosenblatt*(1958). O perceptron foi proposto por *Rosenblatt* para reconhecimento e classificação de padrões; é construído em torno de um neurônio não linear. O perceptron representa o progresso do modelo proposto por *McCulloch e Pitts*, pois os pesos são variáveis e a rede é capaz de aprender.

Em relação à capacidade representacional pode-se mencionar que o perceptron define um subespaço multidimensional, classificando como positivos os exemplos que estão de um lado e como negativos os demais. O perceptron pode ser usado para representar as funções lógicas *AND*, *OR*, *NOR* e *NAND*. É importante mencionar que qualquer função booleana pode ser representada por uma rede de perceptrons, que executam as operações citadas. A função booleana pode ser representada na forma normal disjuntiva, ou seja, disjunção de cláusulas [89]-[91].

MLP - Perceptron Multicamadas

Nos estudos de casos desenvolvidos nesta tese em que a *RNA* é usada como método matemático, utilizou-se a *RNA* do tipo *perceptron multicamada* (MLP). O erro de

predição ou resíduo possibilita um algoritmo de treinamento. Neste tipo de rede, todas as camadas são completamente ligadas por pesos, exceto a primeira camada onde ocorre a recepção dos padrões de entrada.

Na Figura 3.2 tem-se a estrutura de um *MLP*. As entradas formam a camada de entrada da rede; as saídas são formadas pela camada de saída. A segunda camada que é visível para a camada de entrada, mas não para a camada de saída, é denominada camada oculta ou camada intermediária e, ao contrário das camadas de entrada e saída, seu tamanho não é fixo. A camada oculta é geralmente utilizada para forçar a rede a obter um modelo simples do sistema, gerando dados com habilidade para generalizar padrões previamente ocultos [10], [87], [91].

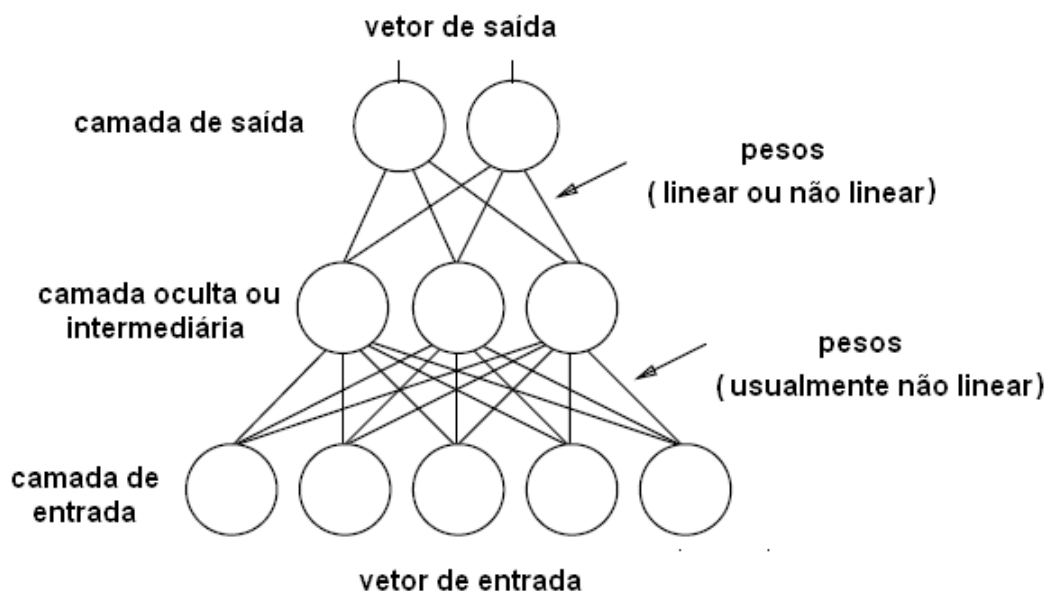


Figura 3.2: Estrutura de um *MLP*.

As redes *MLP* apresentam um poder computacional muito maior que as redes sem camadas intermediárias e podem tratar com dados não linearmente separáveis. As funções de ativação precisam ser diferenciáveis para que o gradiente possa ser calculado, direcionando o ajuste dos pesos.

3.1.3 Processo de aprendizado

O treinamento de uma *RNA* usa um conjunto de dados correspondentes a uma amostra de sinais para entrada e saída do sistema. Esse treinamento é realizado por meio dos algoritmos de aprendizado.

O treinamento supervisionado de retropropagação consiste em ajustar os pesos e os limites de suas unidades de modo a obter o mapeamento desejado. As redes *MLP* com retropropagação são bem utilizadas na prática.

O procedimento é realizado da seguinte forma: quando um padrão é inicialmente apresentado à rede neuronal ela produz uma saída. Em seguida, obtém-se a diferença entre a resposta atual ou obtida pelo modelo neuronal e a desejada. Com esse erro são realizados os ajustes apropriados nos pesos das conexões de modo a reduzir esta diferença.

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 3.3.

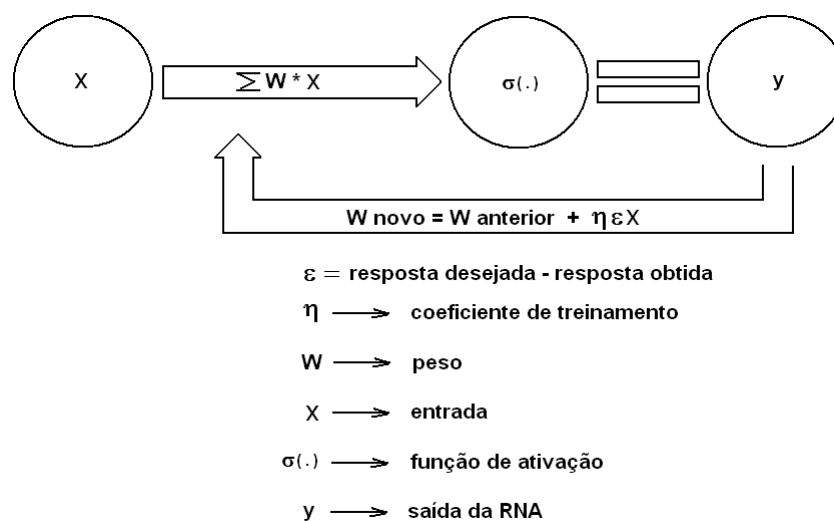


Figura 3.3: Diagrama para treinamento do *Perceptron*.

O algoritmo de retropropagação treina a *RNA* camada por camada, conforme a Figura 3.4 [92].

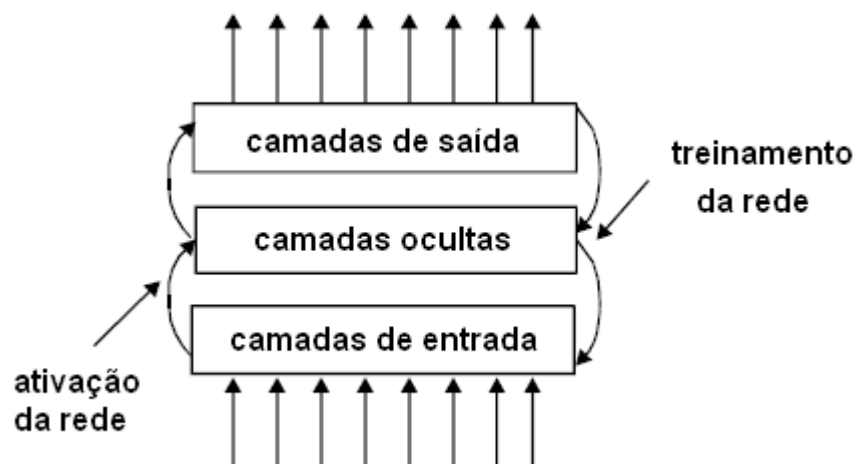


Figura 3.4: Representação da ativação da rede e treinamento por erro de propagação.

A seguir, tem-se as equações do cálculo computacional para frente e para trás do algoritmo de retropropagação [93]:

- propagação para frente

$$y_k^1 = \sigma\left(\sum_{i=0}^{m_0} w_{ki}^1 x_i + \mu_k^1\right), \quad (3.1)$$

para o k -ésimo neurônio da primeira camada,

$$y_k^l = \sigma\left(\sum_{i=0}^{m_{l-1}} w_{ki}^l y_i^{l-1} + \mu_k^l\right), \quad (3.2)$$

$l = 2 \dots L$ para o k -ésimo neurônio da l -ésima camada;

- propagação para trás

$$\delta_k^L = (d_k^L - y_k^L)(y_k^L), \quad (3.3)$$

para o k -ésimo neurônio da camada de saída,

$$\delta_k^l = (y_k^l) \sum_i^{m_{l+1}} \delta_i^{l+1} w_{ik}^{l+1}, \quad (3.4)$$

$l=1 \dots L$ para o k -ésimo neurônio da l -ésima camada intermediária,

$$w_{ki}^l = w_{ki}^l + \Delta w_{ki}^l = w_{ki}^l + \eta \delta_k^l y_i^{l-1}. \quad (3.5)$$

Sendo w , o peso entre dois neurônios; d , a resposta desejada e x , a entrada; y , o neurônio de saída; σ , a função de ativação; η , a taxa de aprendizagem; l , o número de camadas, onde 1 (um) representa a primeira camada intermediária e L a camada de saída; i ou k representa o número de neurônios em cada camada. Dessa forma, y_k^l é a saída do k -ésimo neurônio da l -ésima camada intermediária; w_{ki}^l é o peso entre o k -ésimo neurônio na l -ésima camada e o i -ésimo neurônio na $(l-1)$ -ésima camada. d_k^l é a resposta desejada do k -ésimo neurônio na l -ésima camada. μ_k^l é a polarização no k -ésimo neurônio. δ_k^l é o k -ésimo valor da *Regra Delta* no neurônio para a correção do peso. m_0 é o número de neurônios na camada de entrada, m_{l-1} é o número de neurônios na $(l-1)$ -ésima camada.

A fase de treinamento é seguida da fase de estimação, onde a *RNA* se comporta deterministicamente, ou seja, para cada combinação de valores de entrada, a saída será sempre a mesma. Uma vez treinada, a *RNA* pode estimar, reconhecer e classificar dados desconhecidos ou incompletos, inferindo soluções e até mesmo capturando relações entre os dados.

O treinamento de um *MLP* também pode ser realizado com auxílio do *Filtro de Kalman Estendido*, (FKE). Neste caso, o *FKE* "tenta" estimar o estado de um sistema que pode ser modelado por um sistema linear com a adição de *ruído Gaussiano*, tendo as medidas disponíveis como combinações lineares dos estados do sistema, sido corrompidas pela adição do *ruído Gaussiano*. No treinamento da *RNA* os pesos da rede são os estados do sistema a serem estimados e as saídas da *RNA* são as medidas disponíveis usadas

pelo *FKE*. O algoritmo de retropropagação é uma simplificação do *FKE* [86], [88], [90], [91], [94].

3.1.4 Modelagem Direta

Na modelagem direta, Figura 3.5, o modelo neuronal é colocado em paralelo com o sistema e a saída desse modelo é utilizada como sinal de ajuste dos pesos da *RNA*. Esta estrutura de aprendizagem é utilizada para resolver um problema clássico de aprendizagem supervisionada em que o sistema fornece diretamente os valores das saídas desejadas que servem para adaptar o modelo neuronal.

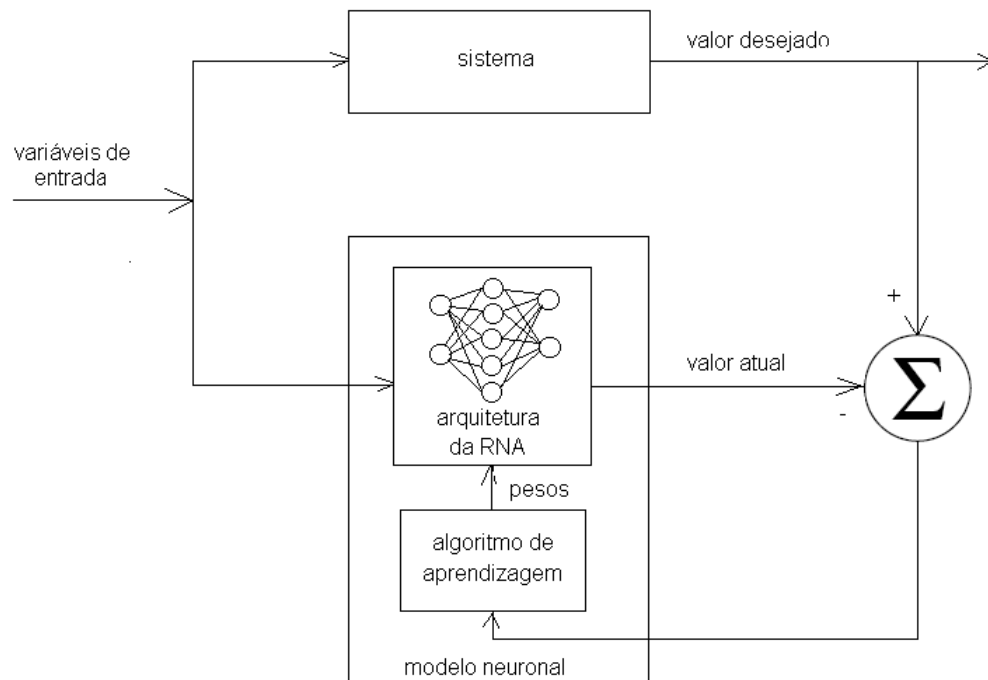


Figura 3.5: Diagrama em blocos do modelo direto para estimação via redes neurais.

3.2 Filtragem Ótima Estocástica para *Hardware* de Tempo Real

Uma das características do *FK* é sua natureza recursiva que estima o estado de um sistema a partir de medidas incertas e incompletas; necessita de pouca capacidade de memória. O *FK* é amplamente utilizado devido a sua simplicidade, tratabilidade e robustez e por esses aspectos o método de medição indireta utilizando *FK*, torna-se extremamente útil quando se trata de aplicações em tempo real.

A propriedade fundamental do *FK* é que ele atua como um estimador de estados para um sistema dinâmico linear. Quando a dinâmica do sistema e os modelos de observações são lineares, os modelos de probabilidade condicional são distribuições normais, e o *FK* pode ser utilizado. As equações do *FK* podem ser algebricamente manipuladas em diversas formas; podem ser escritas usando funções não aproximadas do sistema. O *FK* é um estimador recursivo com estrutura de predição-correção.

Entretanto, em muitas aplicações de interesse, as condições de linearidade não são satisfeitas e extensões do *FK* devem ser utilizadas. Nesses casos, pode-se utilizar o *Filtro de Kalman Estendido*, *FKE*, que é baseado na expansão da série de *Taylor* do sistema não linear. O *FKE* explora a suposição de que todas as transformações são quase lineares e lineariza todas as transformações não lineares, de tal forma que as equações tradicionais do *FK* possam ser utilizadas. Dessa forma, na maioria das aplicações, o *FKE* é utilizado em sistemas não lineares de primeira ordem que apresentem linearidade

local. Para sistemas não lineares em que a ordem é mais elevada, não é vantajoso se utilizar o *FKE* devido às representações pobres das funções não lineares e distribuições de probabilidade de interesse, resultando em estimativas incorretas.

O *Filtro de Kalman Unscented* (FKU) busca superar as limitações do *FKE* por meio da utilização do modelo não linear real, utilizando para tal uma abordagem de amostragem determinística. O *FKU* baseia-se na Transformada *Unscented* (TU) que fornece um mecanismo para transformar as informações de média e covariância, suprindo as deficiências impostas pela linearização realizada pelo *FKE* [90], [95]-[99].

Um *FK* corretamente projetado permite que algumas grandezas sejam observadas, e então, a partir dessas observações, seja possível reconstruir ou estimar o estado interno completo de um sistema. Num *FK* também é possível se utilizar funções para filtragem passa-baixa e amplificação; pode ser elaborado para fornecer compensação de temperatura, rejeição de modo comum e correção de tensão de *offset* [100].

Os principais resultados do *FK* nos estudos de casos deste trabalho são utilizados para desenvolver um algoritmo de predição e correção para aplicações em tempo real da filtragem *Kalman*. Este algoritmo é utilizado como estimador do vetor de estados. O desenvolvimento do algoritmo fundamenta-se no critério de otimalidade dos mínimos quadrados para estimação não polarizada do vetor de estado com ponderação ótima [101].

No item 3.2.1, o sistema do mundo real é representado por um modelo determinístico e outro estocástico. A decomposição deste modelo é apresentada no item 3.2.2.

3.2.1 Modelo do Sistema Real

Um sistema real pode ser representado por um modelo linear determinístico com ruídos aleatórios. As equações a diferenças finitas modelam ou representam o sistema real e podem ser utilizadas na análise do comportamento desse sistema, bem como para fins de projeto de filtro. O referido sistema é representado de modo linear no espaço de estados. Acrescentam-se ruídos do sistema e da medida ao modelo determinístico e obtém-se assim o sistema linear determinístico/estocástico [101],

$$x_{k+1} = A_k x_k + B_k u_k + G_k \xi_k, \quad (3.6)$$

$$y_k = C_k x_k + D_k u_k + \eta_k. \quad (3.7)$$

Sendo que $A_k^{n \times n}$, $B_k^{n \times m}$, $G_k^{n \times p}$, $C_k^{q \times n}$ e $D_k^{q \times m}$ representam a dinâmica do sistema, entrada-controle, o ruído da planta, a saída e controle-saída, respectivamente. As matrizes A_k , B_k , G_k , C_k e D_k são conhecidas e constantes. As seqüências $x_k^{n \times 1}$, $u_k^{m \times 1}$ e $y_k^{q \times 1}$ são vetores de estado, entrada e saída, respectivamente. Os valores de ξ_k e η_k são os ruídos do sistema e da medida, respectivamente.

A Equação (3.6) representa o modelo do sistema que descreve a dinâmica do estado no tempo. O estado em $k + 1$ é obtido a partir do produto entre a matriz de transição A_k e o estado previsto x_k . Este modelo, porém, é imperfeito devido a adição do vetor de ruído do sistema. A Equação (3.7) representa o modelo da medição e a partir desta,

tem-se uma relação entre as medidas e o estado, já que usualmente, não se pode medir diretamente o estado. As seqüências de ruídos desconhecidas tem medidas estatísticas conhecidas denominadas ξ_k e η_k e representam os ruídos do sistema e da observação (medida), respectivamente. As médias são nulas e as variâncias e covariâncias são medidas estatísticas conhecidas [101], [102].

As duas primeiras restrições de tempo real a seguir são estabelecidas para a dimensão tanto a estrutural do modelo quanto a das variáveis que devem estar associadas com o *hardware* do sistema embarcado.

Restrição 1 - *Restrições para garantir a realização estrutural*

$$1 \leq m, p, q \leq n.$$

A restrição 1 relaciona a dimensão de espaço de estado com dimensão da entrada-controle, m , dimensão da seqüência de ruído, p , e dimensão da saída, q .

3.2.2 Decomposição do Modelo

O sistema linear determinístico/estocástico, Equações (3.6) e (3.7), é decomposto em dois sistemas lineares; um dos sistemas é determinístico e o outro, estocástico. A solução do sistema determinístico é conhecida e esta condição justifica a decomposição do sistema [101].

Sistema Linear Determinístico:

$$x_{k+1}^D = A_k x_k^D + B_k u_k, \quad (3.8)$$

$$y_k^D = C_k x_k^D + D_k u_k. \quad (3.9)$$

Sistema Linear Estocástico:

$$x_{k+1}^E = A_k x_k^E + G_k \xi_k, \quad (3.10)$$

$$y_k^E = C_k x_k^E + \eta_k. \quad (3.11)$$

Restrição 2 - Modelo Determinístico/Estocástico

1. As seqüências $\{\xi_k\}$ e $\{\eta_k\}$ são ruídos brancos *Gaussianos*, não correlacionados,
 - Média zero: $E(\xi_k) = 0$ e $E(\eta_k) = 0$;
 - Variâncias: $\text{Var}(\xi_k) = Q_k$, $\text{Var}(\eta_k) = R_k$ são matrizes definidas positivas e

$$E(\xi_k \eta_l) = 0, \quad \forall \quad k \text{ e } l.$$
2. O estado inicial x_0 é independente de $\{\xi_k\}$ e de $\{\eta_k\}$: $E(x_0 \xi_k^T) = 0, \quad \forall \quad k$;

$$E(x_0 \eta_k^T) = 0, \quad \forall \quad k.$$

A solução do sistema linear, Equações (3.6) e (3.7), após a decomposição é a soma das soluções do sistemas determinístico e estocástico,

$$x_k = x_k^D + x_k^E, \quad (3.12)$$

sendo x_k^D a solução do sistema determinístico e x_k^E a solução do sistema estocástico [101], [103].

A solução do sistema determinístico é conhecida e é dada pela equação de transição de estados,

$$x_k^D = (A_{k-1} \dots A_0)x_0^D + \sum_{i=1}^k (A_{k-1} \dots A_0)B_{i-1}u_{i-1}. \quad (3.13)$$

A solução da equação de estado estocástica fornece uma estimativa do vetor de estado, x_k^E , e conseqüentemente, a saída da equação determinística/estocástica. A estimativa ótima \hat{x} conduz a estimativa ótima da saída,

$$\hat{x}_k = x_k^D + \hat{x}_k^E.$$

A solução da equação estocástica é formulada para fornecer uma estimativa ótima de y_k . A componente estocástica se faz presente nos ruídos, na forma de informação estatística, tais como médias e variâncias.

A estimativa \hat{x}_k de \hat{x}_k^E consiste na escolha de uma matriz de ponderação ótima que fornece a variância mínima de \hat{x}_k . O método de solução segue a abordagem dos mínimos quadrados.

3.2.3 Os Processos de Estimação

Neste item, os processos de estimação se referem à relação temporal entre a quantidade k de valores da medida e a quantidade k de valores das variáveis de estados que são utilizados para caracterização do processo. Podem-se obter estimadores diferentes dos estados das Equações (3.6) e (3.7), dependendo das medidas disponíveis.

Considerando-se que $y_k = \{y(j), u(j) | j \leq k\}$ é conhecido e usando y_k para estimar $x(k+m)$, haverá três situações descritas para caracterização das restrições deste processo de estimação e ilustradas na Figura 3.6.

Restrição 3 - Caracterização dos Processos de Estimação

1 Processo de Filtragem Digital: quando $m = 0$, ou seja, $x(k)$, Figura 3.6a - Relacionando o vetor de estados x_k com as medidas y_j .

$$k = j, \quad \text{Notação} \quad \hat{x}_k = \hat{x}_{k|k}. \quad (3.14)$$

2 Processo de Predição Ótima de x_k : quando $m > 0$, ou seja, $x(k+1)$, Figura 3.6b.

$$k > j, \quad \text{Notação: } \hat{x}_k = \hat{x}_{k|j}. \quad (3.15)$$

3 Processo de Filtragem Digital Suave: quando $m < 0$, ou seja, $x(k-1)$, Figura 3.6c.

$$k < j, \quad \text{Notação: } \hat{x}_k = \hat{x}_{k|j}. \quad (3.16)$$

Caso não seja necessário trabalhar de modo recursivo ou em tempo real, os valores estimados podem ser obtidos usando o *FK* suave, conforme ilustrado na Figura 3.6c.

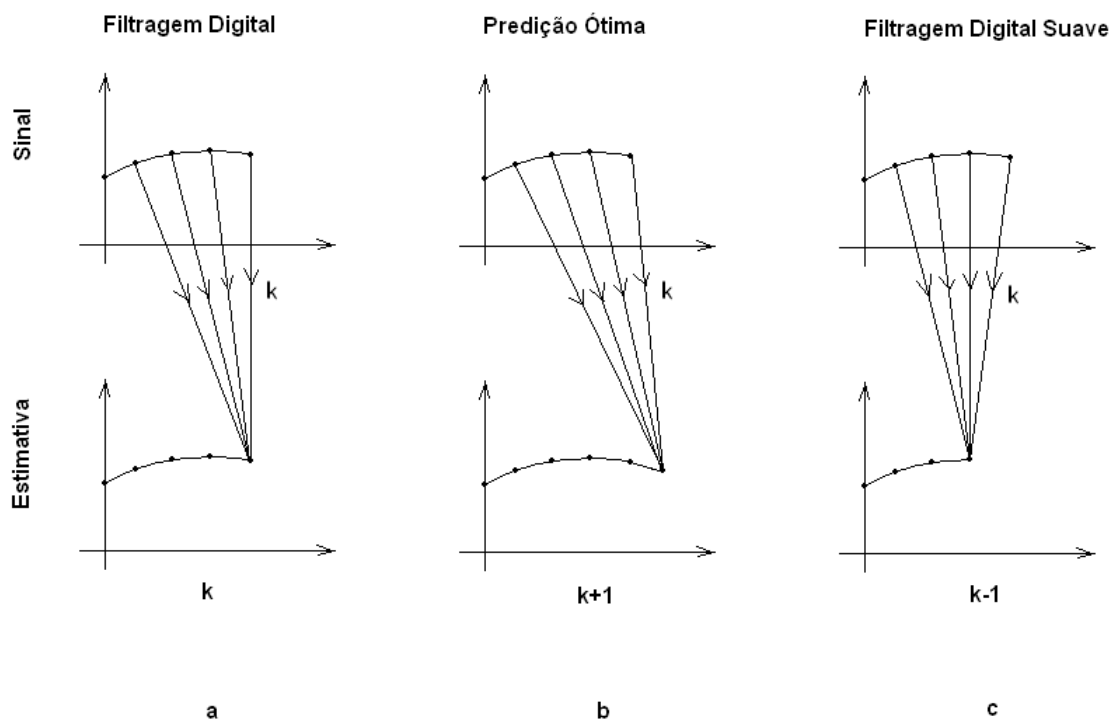


Figura 3.6: a) Filtragem Digital; b) Predição Ótima; c) Filtragem Digital Suave.

Restrição 4 - Hardware de Tempo Real

- 1 Se o estado estimado $\hat{x}_k = \hat{x}_{k|k}$ é determinado utilizando todos os sinais y_0, \dots, y_k , o processo de estimação não é aplicável para problemas em tempo real para valores quando k é muito grande.
- 2 Motivo: o armazenamento das medições e solicitações computacionais aumentam com o tempo.

3.2.4 Algoritmo *Kalman*-Padrão

Uma formulação recursiva da estimação $\hat{x}_k = \hat{x}_{k|k}$ contorna o problema da restrição 4 de *Hardware* de tempo real. Nesta formulação os sinais digitais são lidos a cada passo k e desta forma pode-se garantir um pequeno armazenamento de dados. Esta formulação é o algoritmo de filtragem de *Kalman*-Padrão.

Algoritmo 5 - *Filtragem de Kalman-Padrão, processo de filtragem digital suave para determinar $\hat{x}_{k|k}$ a partir da predição $\hat{x}_{k|k-1}$.*

Condições Iniciais:

$$P_{0,0} = \text{Var}(x_k), \quad (3.17)$$

$$\hat{x}_{0,0} = E(x_0). \quad (3.18)$$

Filtragem:

$$K_k = P_{k/k-1} C_k^T (C_k P_{k/k-1} C_k^T + R)^{-1}, \quad (3.19)$$

$$\hat{x}_{k/k} = \hat{x}_{k/k-1} + K_k (y_k - C_k \hat{x}_{k/k-1}), \quad (3.20)$$

$$P_{k/k} = (I - K_k C_k) P_{k/k-1}. \quad (3.21)$$

Predição:

$$\hat{x}_{k/k-1} = A_{k-1} \hat{x}_{k-1/k-1}, \quad (3.22)$$

$$P_{k/k-1} = A_{k-1} P_{k-1/k-1} A_{k-1}^T + G_{k-1} Q G_{k-1}^T, \quad (3.23)$$

para $k = 1, 2, \dots$. Sendo Q e R as matrizes covariância dos ruídos do sistema e da observação, respectivamente e P , a matriz covariância do erro. As variáveis com sub-índices $0, 0$ representam as condições iniciais.

As Equações (3.19) e (3.21) representam a filtragem e as Equações (3.22) e (3.23), a função de predição do processo de filtragem de *Kalman*.

A partir da observação y_k , tem-se a relação da filtragem tendo em vista o mapeamento [104], [105],

$$H_X : \hat{x}_{k|k-1} \longrightarrow \hat{x}_{k|k}. \quad (3.24)$$

A função H_X é baseada na forma clássica de *Gauss-Legendre*, para realizar uma correção na predição $\hat{x}_{k|k-1}$ do estado em k , Equação (3.20).

3.3 Comentários

Embora as *RNA* sejam baseadas na neurociência, a maioria dessas redes está relacionada com modelos matemáticos e/ou estatísticos como classificação de padrões não paramétricos, agrupamentos de algoritmos, filtros não-lineares e modelos de regressão estatística que podem ser associados com o modelo neurobiológico. A *RNA* busca uma aproximação do modelo do cérebro e do modo como este processa a informação. É utilizada como aproximador universal e é considerada uma ferramenta que aprende a

partir do ambiente; que encontra dependências não evidentes entre dados. A partir de sinais medidos de alguma fonte conhecida ou não, a *RNA* pode ser treinada para aplicações como: classificação de padrão; síntese e reconhecimento da fala; estimação; simulação e predição de processos; interfaces adaptáveis entre o homem e os sistemas físicos complexos; função de aproximação; compressão de imagens; memória associativa; otimização combinatorial; modelamento e controle de sistemas não-lineares.

O filtro de *Kalman* é, essencialmente, um conjunto de equações matemáticas que implementa um estimador do tipo predição-correção que é ótimo quando minimiza a covariância do erro de estimação, desde que sejam consideradas algumas restrições. Desde a introdução do filtro de *Kalman* simples, *Kalman*(1960)[106], esta técnica de medição tem sido alvo de extensivas pesquisas e aplicações, particularmente em áreas que contribuem com melhorias na navegação. A relativa simplicidade e a natureza robusta do próprio filtro são vantagens para sua implementação.

Apesar do *FK* ser bastante aplicado em processos regidos por equações a diferenças estocásticas lineares, existem diversas aplicações em sistemas não-lineares. A atuação do *FK* em sistemas não-lineares é caracterizada pela linearização dos estados estimados. Nesse novo filtro, o *FKE*, é utilizada uma expansão da série de *Taylor* para reter somente os termos de primeira ordem, obtendo-se novas equações que linearizam as estimativas do estados. A operação básica do *FKE* é muito similar ao *FK*.

Capítulo 4

Considerações Sobre Projetos de SMI-TR

Em um Sistema em Tempo Real, *STR*, a execução das tarefas deve obedecer as restrições de tempo. Caso o limite máximo para o tempo de resposta de uma tarefa não seja respeitado, a aplicação pode falhar, trazendo em casos extremos consequências graves. Ao tentar corrigir falhas em sistemas de tempo real, deve-se inicialmente identificar as tarefas que não estão cumprindo os requisitos temporais especificados. Descrições semelhantes de *STR* podem ser verificadas em [107], [108].

Um sistema operando em tempo real é fundamental em importantes áreas de aplicação, como o controle de processos, plantas de geração de energia, controle de tráfego aéreo e multimídia. Em particular, a maioria dos sistemas de segurança críticos e muitos sistemas embarcados são sistemas de tempo real.

As métricas mais importantes comumente utilizadas para caracterizar projetos são o custo e o desempenho do *software* e implementações de *hardware*. Embora existam

outras métricas, apresentadas neste Capítulo, muitas decisões de alto-nível são completamente baseadas nas métricas acima mencionadas [109].

Normalmente, os dispositivos que possuem arquiteturas reconfiguráveis são aplicados em ambientes que exigem elevado desempenho, segurança e rapidez de processamento. Para um Sistema de Medição Indireta em Tempo Real, *SMI-TR*, em que o tempo de execução é tão importante quanto o resultado exato das saídas, uma arquitetura reconfigurável torna-se extremamente relevante. Tanto o projeto de um *SMI-TR* quanto a sua reconfigurabilidade precisam ser cuidadosamente analisados de modo que o desempenho do estimador, na prática, apresente o(s) valor(es) da(s) grandeza(s) que se deseja medir ou filtrar com menor erro possível do estimador.

4.1 *Hardware e Software em Tempo Real*

O *hardware* em tempo real pode consistir de componentes considerados padrões, porém deve suportar as restrições de projeto em tempo real. O grau de rigor de uma aplicação específica que, sem exceção, os sistemas estritos (rigorosos) em tempo real devem satisfazer a todas as suas restrições temporais. Caso uma restrição seja violada, o sistema falha. Em outro extremo tem-se os sistemas tolerantes de tempo real que realizam suas tarefas com algumas restrições. O *hardware*, por exemplo, tem influência sobre a previsibilidade de um *STR*.

Os *STR* são baseados em cálculos computacionais. Alguns envolvem minicomputadores, outros microcontroladores ou ainda arquiteturas reconfiguráveis. Nesta acepção, é necessário o uso de uma linguagem de programação.

As linguagens de alto nível apresentam algumas vantagens e desvantagens quando se refere a projetos de sistemas em tempo real. Estas linguagens geram facilidades na programação, como definição ampla de variáveis, realização de equações ou funções, definição fácil de *loops* e de condições de decisão.

Em relação à linguagem *assembly*, elas apresentam vantagens do tipo:

- facilidade de desenvolvimento;
- documentação de melhor nível e a portabilidade para outras máquinas, causando melhor produtividade, qualidade e maior tempo de vida;
- manutenção mais fácil do *software*.

No entanto, para projetos em tempo real, a linguagem mais adequada é a linguagem *assembly*, isto por que esta linguagem por estar mais próxima da linguagem de máquina, apresenta menor tempo de execução dos programas, menor consumo de memória e facilidades para operação com certos tipos de interfaces reais. Porém, nada impede que sistemas embarcados em tempo real sejam escritos somente em linguagens de alto nível ou ainda que se utilize linguagem *assembly* com linguagem de alto nível em um mesmo projeto, pois as características das linguagens devem ser adequadas às necessidades do sistema, seja ele em tempo real ou não.

De acordo com *Farines* [110] é improvável que uma única linguagem seja abrangente o suficiente para acomodar todos os requisitos temporais das abordagens existentes e mais os novos que certamente deverão surgir nos próximos anos.

As linguagens em tempo real atualmente mais utilizadas são *assembly*, *C*, *C++* e *Ada95* [111]. As linguagens que foram desenvolvidas nos últimos 20 anos para atuar em aplicações em tempo real são: *Ada95*, *occam-2*, *Esterel*, *PEARL*, *Real-time Euclid*, *Real-time C*, *Real-time C++* e *Java*. Existem, porém, outras linguagens em tempo real e algumas são utilizadas somente em pesquisas que envolvem aplicações especializadas [110].

4.2 Desempenho do Sistema

Para a avaliação do desempenho do *SMI* em tempo real deve-se levar em consideração a complexidade do sistema de *software* utilizado, bem como, a técnica para melhorar o desempenho. A melhora de um *software* em um *hardware* está em função dos fatores de *hardware* que influenciam o desempenho global do sistema. Os fatores de *hardware* que têm maior influência no desempenho do sistema são, [112]: a utilização por parte do compilador das instruções da máquina na geração do código de um programa; a maneira do *hardware* implementar as instruções; a maneira como a memória e os dispositivos de entrada/saída se comportam durante o processamento do programa.

4.2.1 Considerações do Desempenho

Para maximizar o desempenho é necessário minimizar o tempo de resposta ou tempo de execução da aplicação analisada. O desempenho e o tempo de execução do sistema podem ser relacionados como segue,

$$D_s = \frac{1}{T_{ea}}, \quad (4.1)$$

sendo D_s o desempenho do sistema dada em Hertz (Hz) e T_{ea} , o tempo de execução da aplicação dado em segundos (s).

O desempenho pode ser verificado sob diferentes métricas, porém o foco é a análise do desempenho em relação ao processador. A medida básica é o tempo de execução gasto no processador. A Equação (4.2), relaciona o tempo de execução no processador para o programa (T_{ep}) com número de ciclos de *clock* (N_{cc}) e o tempo do ciclo de *clock* (T_{cc}).

$$T_{ep} = N_{cc} = N_{cpp} * T_{cc}, \quad (4.2)$$

sendo N_{cpp} o número de ciclos de *clock* do processador para o programa. O desempenho pode ser melhorado reduzindo o tamanho do ciclo de *clock* ou o número de ciclos de *clock* necessários à execução do programa em análise.

4.2.2 Integração *Hardware/Software* em Tempo Real

O *hardware* e o *software* interagem de diversas maneiras. O compilador gera instruções a serem executadas de modo que o processamento do programa seja realizado. O tempo de execução depende também do número de instruções do programa. Pode-se especificar o número de ciclos de *clock* do processador necessários à execução de um programa como o produto entre o número de instruções de um programa (ip) e o número médio de ciclos de *clock* por instruções (N_{cci}), como:

$$N_{cpp} = N_{ip} * N_{cci}. \quad (4.3)$$

O N_{cci} , sendo uma média dos tempos gastos por todas as instruções executadas pelo programa, permite que sejam realizadas comparações entre implementações diferentes da mesma arquitetura do conjunto de instruções. Os três principais fatores que afetam o desempenho do processador são agrupados na Equação (4.4),

$$T_{ep} = N_{ip} * N_{cci} * T_{cc}. \quad (4.4)$$

A integração entre o *hardware* e o *software* em tempo real ocorre com o sistema embarcado para aplicações em tempo real. Nessa etapa, o sistema em tempo real representa uma unidade no qual será verificado o tempo de resposta a uma dada aplicação.

A unificação do sistema ocorre com o processamento do *software*, momento em

que são eliminando prováveis erros que podem ocorrer, como por exemplo, capacidade de memória insuficiente, erros no *link* de página, adequando às características da arquitetura que será utilizada. Após a eliminação dos erros, o código pode, então, ser transferido definitivamente para a arquitetura. O SMI é testado, podendo ainda ocorrer falhas, indicando erros do *hardware* ou do *software*. Durante essa validação do sistema em tempo real, é necessário registrar alguns itens considerados relevantes para que o problema seja identificado e corrigido. A detecção de falhas em sistemas de tempo real, normalmente, envolve análises precisas sobre a interação das tarefas ao longo do tempo.

Quanto aos graus de rigor e criticalidade, a aplicação de um SMI em tempo real, pode ser representada pela Figura 4.1. No eixo x tem-se um sistema rigoroso e no eixo y , um sistema crítico. As previsibilidades probabilista e determinista são associadas aos dois grupos acima mencionados [110].

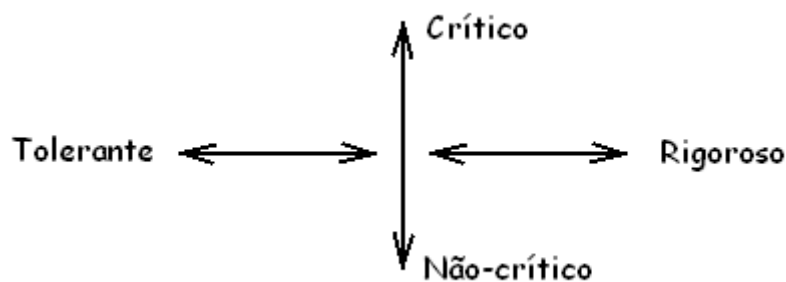


Figura 4.1: Graus de rigor e criticalidade de um sistema em tempo real .

4.3 Qualidade da Medição Indireta

Além da escolha adequada do método de medição utilizado, a qualidade de um *SMI* está associada à necessidade de estimações rápidas com métricas de qualidade, tais como: ciclo de *clock*, tempo de execução de *hardware* e *software* e tamanho da memória de dados e programa. Quando se trata das especificações e projetos de um *SMI* para uso em tempo real é necessário seguir as recomendações das diversas referências sobre sistema em tempo real, tais como:

- estimação do *hardware*: configuração mínima necessária, estimação do ciclo de *clock*: máximo atraso associado a operação, estimação do controle dos procedimentos, estimação do tempo de execução, estimação da taxa de comunicação e estimação da área;
- estimação do *software*: estimação do modelo do processador específico, estimação genérica do modelo, tempo de execução do programa, tamanho de memória do programa, tamanho de memória de dados;
- outras métricas: dissipação de potência, testabilidade e tempo de projeto.

A medição indireta pode ser simplificada para se obter resultados mais rápidos; porém, pode-se incorrer em erros de estimação e baixa exatidão. O alto nível de exatidão nem sempre é necessário desde que o valor estimado seja obtido com métricas de qualidade, permitindo alternativas para tomadas de decisões. O método de medição indireta

escolhido deve apresentar fidelidade, ou seja, um percentual de comparações corretamente preditas entre implementações. Segundo *Gajski et al*, [109], a fidelidade de uma medição indireta é definida considerando-se $D = D_1, D_2 \dots D_n$ como um conjunto de implementações em,

$$\mu_{ij} = \begin{cases} 1, & E(D_i) > E(D_j) \text{ e } M(D_i) > M(D_j) \text{ ou} \\ & E(D_i) < E(D_j) \text{ e } M(D_i) < M(D_j) \text{ ou} \\ & E(D_i) = E(D_j) \text{ e } M(D_i) = M(D_j) \\ 0, & \forall \end{cases} .$$

Sendo E e M valores estimados e medidos, respectivamente; μ_{ij} para todo i, j tal que $1 \leq i, j \leq n$ e $i \neq j$ e n uma análise entre dois valores estimados e medidos. A fidelidade de uma medição indireta, \mathfrak{S} , pode ser escrita por:

$$\mathfrak{S} = 100 \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \mu_{ij}. \quad (4.5)$$

Mesmo que todos os componentes de erro conhecidos ou suspeitos tenham sido avaliados e as correções adequadas tenham sido aplicadas, ainda assim permanece uma incerteza sobre quão correto é o resultado declarado, isto é, uma dúvida acerca de quão corretamente o resultado da medição representa o valor da grandeza que está sendo medida [113].

Nas medições diretas, o termo *incertezas*, no sentido mais amplo, significa dúvida acerca da validade do resultado de uma medição. Dado que numa medição indireta são realizados cálculos envolvendo valores de outras grandezas medidas diretamente, a incerteza do resultado é função, também, dos vários valores medidos diretamente ao longo do processo de estimação. Matematicamente, tem-se que se X é a grandeza cujo limite se deseja determinar e a, b, c, \dots os valores das grandezas de que X depende e, sendo cada uma delas com limite de erro $\Delta a, \Delta b, \Delta c, \dots$ pode-se escrever que $X = f(a, b, c, \dots)$. Então, o limite do erro da grandeza X , ΔX , pode ser dado por,

$$\Delta X = \left| \frac{\partial X}{\partial a} \right| \Delta a + \left| \frac{\partial X}{\partial b} \right| \Delta b + \dots \quad . \quad (4.6)$$

As *incertezas herdadas*, bem como aquelas que estão associadas aos resultados da medição indireta, caracterizam-se por dispersões dos valores que podem ser razoavelmente atribuídos ao mensurando. O valor da incerteza em medição direta pode ser analisado por meios estatísticos de uma série de observações (incerteza tipo A) ou por outros meios que não a análise estatística de uma série de observações (incertezas tipo B) [114]. A ênfase deste trabalho é sobre a medição indireta de acordo com a abordagem que foi mencionada inicialmente, portanto, não será dada maior profundidade acerca desses métodos de análise para a medição direta. Todavia, para entender mais sobre a avaliação da incerteza do tipo A e a incerteza do tipo B , sugere-se que sejam lidas as referências [113]-[115].

4.4 Modelo do Sistema

Os processos no mundo real são intrinsecamente dinâmicos, não lineares, não *Gaussianos*, não estacionários, com restrições e com interações com outros processos. Mas, uma prática comum para o modelo matemático de um sistema e que é adotada nos estudos de casos deste trabalho é realizar considerações acerca desse sistema. Deve-se considerar que o sistema seja *linear invariante no tempo* (LIT), *causal* e *finito*. E, sendo o sistema *LIT contínuo no tempo* (CT), tem-se a descrição no espaço de estados, como

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (4.7)$$

$$y(t) = Cx(t) + Du(t), \quad (4.8)$$

sendo A , B , C e D matrizes reais de dimensões apropriadas [103].

Karl e *Björn* mencionam em [116] que *Kalman* decompôs o sistema em quatro subsistemas: observável e alcançável; observável, mas não alcançável; não observável, mas alcançável; nem observável e nem alcançável. De acordo com essa decomposição, alguns desses subsistemas podem até ser interconectados, porém a relação entre a entrada e saída somente pode ser verificada em um subsistema que seja observável e alcançável.

As definições de *controlabilidade* e *alcançabilidade* de um sistema diferenciam a partir do ponto em que a seqüência de controle pode ser alcançada. No sistema controlável,

essa seqüência é tal que a origem pode ser alcançada a partir de qualquer estado inicial com tempo finito. Em um sistema alcançável, essa seqüência é tal que um estado arbitrário pode ser alcançado a partir de qualquer estado inicial com tempo finito. Essas duas definições são equivalentes quando a matriz A da Equação (4.7) é inversível.

Os termos *alcançabilidade* e *controlabilidade* podem ser utilizados alternadamente em sistemas CT . A partir do período de amostragem de um sistema CT alcançável, pode-se obter as matrizes de um sistema *discreto no tempo* (DT) alcançável, isto por que os sinais de controle permissíveis de um sistema amostrado são subconjuntos dos sinais de controle permissíveis de um sistema CT .

A forma canônica de sistemas alcançáveis CT ou DT com entrada simples de n -ésima ordem é escrita conforme a Equação (4.9) [116], [117],

$$\left. \begin{array}{l} \dot{x}(t) \\ ou \\ x(t+1) \end{array} \right\} = Ax(t) + Bu(t). \quad (4.9)$$

Para problemas envolvendo filtragem de sinais, as Equações (4.7) e (4.8) são representadas por:

$$\dot{x} = Ax + Bu + G\xi, \quad (4.10)$$

$$y = C_2x + D_2u + \eta. \quad (4.11)$$

com $x(0) = x_0$. Adiciona-se ruídos ao sistema, Equação (4.10), e à medida, Equação (4.11), representados por ξ e η . O valor estimado, \hat{z} , é obtido a partir de medidas ruidosas y . Na Figura 4.2 tem-se a ilustração para uma configuração geral de filtragem,

$$z = C_1x + D_1u. \quad (4.12)$$

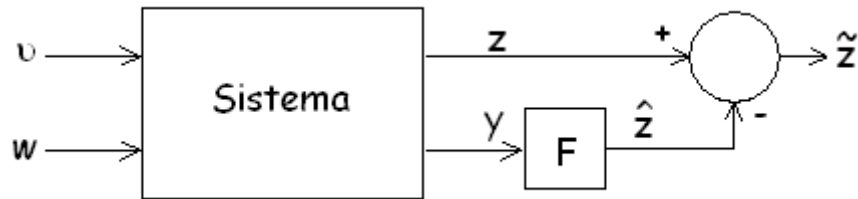


Figura 4.2: Configuração para filtragem.

A equação geral para estimar o valor de z utilizando os valores u e y pode ser escrita como,

$$\dot{x} = \hat{A}x + Bu + By, \quad (4.13)$$

$$\hat{x} = \hat{C}x, \quad (4.14)$$

$$\hat{z} = C_1\hat{x}, \quad (4.15)$$

com $x(0) = x_0$. Sendo x o estado do filtro ou do estimador e \hat{x} , o estado estimado do sistema. A escolha das matrizes nas Equações (4.13), (4.14) e (4.15) conduz a filtros com características diferentes, tais como limitabilidade do erro de estimação e impolarizabilidade do filtro.

Considerando-se que os sinais de entrada e o sistema são limitados e o estimador de estados sendo estável, o erro de estimação deve ser limitado. Em *Seron et al* [103] é definido que um filtro estável apresenta erro de estimação limitado quando para todos os estados iniciais finitos do sistema e do filtro, o erro de estimação \tilde{z} é limitado, sendo as entradas do sistema u e w limitadas.

4.5 Análise de Erros do Estimador

Os erros de estimação para uma medição indireta podem ser divididos em dois grupos: *estatístico* e por *intervalo*. No erro estatístico, considera-se que os erros das medidas $\Delta x_i = \tilde{x}_i - x_i$ são variáveis aleatórias *Gaussianas* independentes com média 0 e desvio padrão conhecido σ_i . Enquanto que no erro por intervalo, somente se considera os erros das medidas Δx_i , desde que $\Delta x_i \in [-\Delta_i, \Delta_i]$, sendo Δ_i conhecido. Quando se utiliza o primeiro grupo, ou seja, o erro da medida sendo estatístico, tem-se um algoritmo de erro de estimação considerado ótimo; utilizando-se o erro por intervalo fixo, tem-se um novo algoritmo que pode até não ser ótimo, porém será melhor que o

previamente conhecido [118].

As considerações mencionadas anteriormente para os erros de estimação estatísticos são baseadas tanto em razões teóricas quanto empíricas.

- 1 Razões teóricas: normalmente, os fabricantes de instrumentos de medição se esforçam no sentido de eliminarem as fontes que alteram o resultado de uma medida. O erro da medida, porém, surge a partir de uma variedade de interferências independentes e, dessa forma, esses erros são descritos como uma soma de pequenas variáveis aleatórias independentes. E, de acordo com o *teorema do limite central*, [104], [119], essa soma, sob condições razoáveis, converge para uma *distribuição normal*.
- 2 Razões empíricas: para a maioria dos instrumentos de medição, o erro da medida é normalmente distribuído, dessa forma, considera-se que as variáveis aleatórias sejam *Gaussianas*.

4.6 Incerteza em Medição Indireta

Sabe-se que na prática é improvável que uma medida seja 100% exata; desse modo, os valores atuais (estimados ou medidos indiretamente) que divergem dos valores desejados podem até ser desprezados, dependendo do algoritmo desenvolvido. Entretanto, o erro da medida será diferente de *zero*.

Em termos metrológicos, o processo de medição raramente consiste de uma medição direta. Para expressar o resultado de uma medição indireta com a incerteza associada, deve-se seguir as recomendações do *Guide to the Expression of Uncertainty in Measurement* (GUM). No GUM[113] tem-se a expressão da incerteza padrão combinada, denominada *Lei de Propagação da Incerteza* para medição indireta, quando o resultado da saída (o mensurando) está relacionado com valores da entrada por um modelo de medição linear, mencionada na Seção 4.3. A utilização dessa expressão é devido à presença de efeitos não lineares significantes no processo de medição. Ao se utilizar métodos para representar a propagação da incerteza na medição indireta abordada neste trabalho, é necessário haver compatibilidade com o GUM [84], [113].

A incerteza acerca do resultado de uma medida indireta está também relacionada com a sensibilidade do método empregado e com as incertezas herdadas da medição direta. No caso do método de medição indireta por meio de Redes Neurais Artificiais, a incerteza do resultado da medição indireta está relacionada com o tipo de treinamento, ou seja, como encontrar os pesos w_{ki} de modo a aproximar a função neuronal da função desejada. No estudo de caso apresentado no item 5.5, foi usado o algoritmo de retropropagação que utiliza o método do gradiente descendente para determinar o menor erro quadrático. O uso do algoritmo de retropropagação é considerado rápido para computar as derivadas parciais.

É interessante abordar neste contexto, os aspectos práticos sobre RNA e o seu algoritmo de treinamento. Algumas considerações devem ser verificadas de modo a obter

a melhor estimativa. Não há regras quanto ao número de camadas ocultas, porém existem algumas ressalvas como por exemplo, o número de camadas ocultas não exceda o número de entradas, pois os vetores dos pesos $\vec{w}_k \triangleq (w_{k1}, \dots, w_{kn})$ são linearmente independentes. Segundo *Nievola*, [120], um *MLP* com duas camadas ocultas é um aproximador universal, ou seja, realiza qualquer mapeamento entrada-saída. Não foi verificada nenhuma regra para a quantidade de neurônios que cada camada oculta deve possuir. Sugere-se, pois, que se inicie com um número pequeno de neurônios e que, gradualmente, aumente-se este número até se encontrar um número ótimo.

As primeiras tentativas para verificação dos valores adequados dos pesos foram realizadas de forma aleatória. Os pesos foram inicializados com valores entre 0 e 1,0, porém os melhores resultados foram obtidos quando os pesos foram inicializados simetricamente em torno de zero, entre $-\frac{1}{2}$ e $\frac{1}{2}$ ou $-1,0$ e $1,0$. Em relação às funções de ativação, baseando-se na argumentação de que as *RNA* são não lineares, alega-se que as funções não devem ser todas lineares. Quanto aos parâmetros, a determinação de quais são melhores, contribuindo para um melhor desempenho é totalmente dependente do problema. Com a variação adequada dos parâmetros é possível obter tanto ativações assimétricas quanto simétricas em relação à zero, da mesma forma que é realizada com a inicialização dos pesos. Bons resultados, entretanto, têm sido obtidos para valores de η entre 0,1 e 10, já para α valores tipicamente entre 0,10 e 0,90. Outros itens também devem ser considerados como: o tipo de treinamento e critério de parada desse treinamento [88], [91], [121], [122].

Devido à larga utilização da rede MLP por diversos pesquisadores, aprimoramentos e melhorias têm sido desenvolvidos no sentido de conduzir a uma diminuição no tempo de treinamento para muitos problemas.

Na Figura 4.3 procurou-se sintetizar os fatores que interferem no resultado em uma medição indireta, levando-se em consideração os itens que são mencionados neste trabalho.

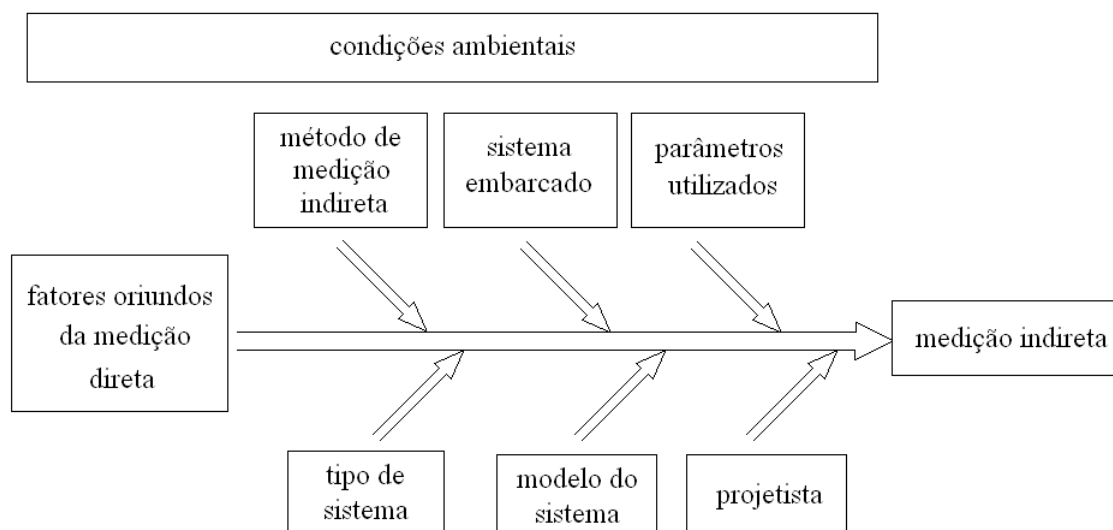


Figura 4.3: Fatores que interferem no resultado de uma medição indireta.

4.7 Sistema Embarcado Ótimo de *Kalman*

Normalmente, um sistema embarcado envolve um microcontrolador com potência limitada e uma arquitetura simples. Entretanto, o projeto de um sistema embarcado não é tão simples quanto a de um microcontrolador devido às condições de restrições

temporais. O que se tem verificado é que sistemas embarcados baseados em microcontroladores estão sendo muito utilizados na indústria de forma a reduzir custos [154].

A especificação de um sistema embarcado ótimo em tempo real de *Kalman*, leva em consideração que um dos eventos são, normalmente, amostras de sinais de trajetórias do veículo aeroespacial. Esse sistema embarcado ótimo consiste de técnicas de otimização para escolher uma estrutura de otimização que determine o *hardware* em função de um conjunto de algoritmos.

A inicialização das constantes em tempo adequado num programa durante a fase inicial de execução em um sistema embarcado, pode conduzir a um desempenho mais rápido em tempo real. Considera-se já o pré-cálculo de algumas constantes que são utilizadas em diversas partes do algoritmo. Uma divisão longa é substituída por uma simples multiplicação, caso a constante já seja calculada pelo compilador e armazenada para posterior uso na área de memória estática [155]. Os melhores valores das variâncias e covariâncias do filtro de *Kalman* podem ser obtidos, inicialmente, por meio de simulação e assim, já armazenados como constantes para uso pelo código em tempo real. Uma vez armazenadas as variáveis consideradas constantes, pode-se ter uma economia de espaço de memória, aumentando-se, assim, as exigências da capacidade de memória mínima no sistema embarcado em tempo real. Neste Capítulo, faz-se uma abordagem acerca das especificações de *hardware* e *software* do sistema embarcado em tempo real ótimo.

4.7.1 Funções Objetivo

O Sistema Ótimo de Medição Indireta consiste da melhor alternativa baseada em otimização para definir o melhor sistema de medição. As restrições representam a conexão entre o *hardware* e o *software* mediante o seguinte mapeamento,

$$H_{HS} : X_{s-ware} \longrightarrow Y_{h-ware}, \quad (4.16)$$

sendo X_{s-ware} os conjuntos de restrições do modelo e do filtro que são mapeadas nos requisitos X_{h-ware} de *hardware*. Uma estrutura de otimização implementa a função H_{HS} , Equação 4.16, de mapeamento,

$$\min \quad Z(Y_{h-ware}) \quad (4.17)$$

sujeito a

$$X_{s-ware} \leq R_{s-ware} \quad (4.18)$$

$$Y_{h-ware} \leq R_{h-ware},$$

sendo R_{h-ware} e R_{s-ware} as especificações de *hardware* e *software*, respectivamente. As especificações R_{h-ware} gerais de *hardware* são as capacidades de memória, tempo de relógio, potência consumida e outras características intrínsecas a um dado *hardware*

embarcado. O conjunto de especificações R_{s-ware} de *software* desenvolvidas nesta seção, tem por base as dimensões dos modelos e estratégias de projeto do algoritmo de filtragem e medição indireta.

As considerações de desempenho, apresentadas no Capítulo 4, fornecem as relações para maximizar a *performance* do sistema embarcado.

A estrutura de otimização é construída a partir da Equação 4.19, representando a função objetivo e as Equações 4.20, 4.21 e 4.22, sendo as restrições,

$$D_s = \frac{1}{T_{ea}}, \quad (4.19)$$

$$T_{ep} = N_{cc} = N_{cpp} * T_{cc}, \quad (4.20)$$

$$N_{cpp} = N_{ip} * N_{cci} \quad (4.21)$$

e

$$T_{ep} = N_{ip} * N_{cci} * T_{cc}. \quad (4.22)$$

$$\max D_s \quad \max \frac{1}{T_{ea}} \quad (4.23)$$

sujeito a

$$T_{ep} \leq N_{cpp} * T_{cc} \quad (4.24)$$

$$N_{cpp} \leq N_{ip} * N_{cci} \quad (4.25)$$

$$T_{ep} \leq N_{ip} * N_{cci} * T_{cc} \quad (4.26)$$

Uma função E_H mapeia as solicitações de *software* em cima de um dado *hardware*,

$$E_H : \{D_{soft-W}\} \longrightarrow \{D_{hard-W}, \} \quad (4.27)$$

sendo $D_{soft-W} = \{E_{mod-p}, E_{mod-g}, T_{p-exe}, E_{mm-t}\}$ o conjunto das especificações de *software* que são obtidas dos algoritmos para medição indireta. E_{mod-p} estabelece a estimação do modelo do processador, E_{mod-g} é a estimação genérica do modelo, T_{p-exe} é o tempo de execução do programa, E_{mm-t} é o tamanho de memória do programa e E_{mem-d} é o tamanho de memória de dados. O conjunto de $D_{hard-W} = \{O_\mu, E_{cont-p}, E_{t-exe}, E_{t-com}, E_{area}, D_{pot}\}$. Para este último conjunto do mapeamento E_H , O_μ representa a operação, E_{cont-p} é a estimação do controle dos procedimentos, E_{t-exe} é a estimação do tempo de execução, E_{t-com} é a estimação da taxa de comunicação, E_{area} é a estimação da área e D_{pot} dissipação de potência.

Procura-se otimizar a fidelidade do sistema em função de medições entre as im-

plementações do mundo real e o desempenho estimado em projeto. A estrutura de otimização, Equações (4.23) a (4.26), fornece um conjunto de opções de *hardware* para um determinado algoritmo em tempo real. Para as alternativas fornecidas monta-se uma nova estrutura de otimização para otimizar a fidelidade do sistema embarcado para medição indireta,

$$\max \quad 100 \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n \mu_{ij} \quad (4.28)$$

sujeito a

$$\mu_{ij} = \begin{cases} 1, & E(D_i) > E(D_j) \quad e \quad M(D_i) > M(D_j) \quad ou \\ & E(D_i) < E(D_j) \quad e \quad M(D_i) < M(D_j) \quad ou \\ & E(D_i) = E(D_j) \quad e \quad M(D_i) = M(D_j) \\ 0, & \forall \end{cases} \quad (4.29)$$

A estrutura de otimização é construída a partir da Equação (4.5) representando a função objetivo e as relações da Equação (4.3) para restrições em função dos E e M , valores estimados e medidos, respectivamente.

4.7.2 Mapeamento da memória

Cada um dos diversos tipos de dados necessita de uma certa quantidade de memória para armazenar a informação por ele representada. Essa quantidade é função do tipo de dado considerado, do tipo de arquitetura e do tipo de linguagem de programação [156].

Na exploração da arquitetura de memória para sistemas embarcados programáveis, considera-se como padrão, o acesso a memória para cada aplicação, a arquitetura de processador, bem como, um compilador ótimo que improvise, significativamente, o comportamento do sistema na memória [157].

A complexidade de um algoritmo está intimamente ligada à da aplicação a que se destina. O algoritmo do filtro de *Kalman* aqui abordado, representa o núcleo, ou seja, o trecho principal do algoritmo e, posteriormente, deve-se realizar uma justaposição de modo a encontrar a solução para a aplicação desejada. Embora o processamento do sistema atue de forma diferenciada para arquiteturas do tipo *Harvard* ou *Von Neumann*, não há diferença entre a memória que contém o algoritmo e a memória que contém os dados usados pelo algoritmo. Em aplicações específicas, o programa pode incluir fórmulas para calcular os elementos de algumas matrizes como, por exemplo, as matrizes A e Q . Os requisitos de memória para o algoritmo tendem a ser independentes da aplicação. Entretanto, para o filtro de *Kalman*, o tamanho do problema está relacionado com as ordens das matrizes do vetor de estado, das medidas e do ruído do processo. Os

requisitos da memória de dados para armazenar as ordens dessas matrizes são muito dependentes tanto do tamanho do problema quanto da estrutura dos dados e, ainda, do modo como a estrutura de dados re-usa os resultados parciais. A memória de dados também pode ser economizada eliminando-se redundâncias na estrutura de dados, como por exemplo, usando a simetria das matrizes de covariância [158].

O núcleo do algoritmo de *Kalman*, Figura 4.4, encerra em si uma seqüência de comandos que é repetida diversas vezes. As variáveis utilizadas nesse núcleo podem ser do tipo inteira, real e indexada, dependendo da aplicação. No Estudo de Caso 5.1, utilizam-se variáveis inteiras, reais e indexadas (matrizes). Dessa forma, na análise do cálculo computacional, leva-se em consideração que não é possível operar diretamente sobre uma matriz como um todo, mas apenas sobre seus componentes, um por vez. Por exemplo, para uma soma entre duas matrizes é necessário somar cada um de seus componentes dois a dois.

Verifica-se por meio do núcleo do algoritmo de *Kalman* que a construção não é linear e sim, cíclica. Cada etapa do ciclo corresponde às etapas do processo da filtragem *Kalman* (Predição e/ou Correção), Figura 4.5. As equações (4.30), (4.31), (4.32) e (4.33) são obtidas a partir do detalhamento realizado em cada etapa do ciclo desse núcleo.

```

1. Condições iniciais:  $\hat{x}_{\min}(0)$  e  $P_{\min}(0)$ ;

   'Loop do Algoritmo de Kalman'

2. For( $k$ , 0,  $n$ ) {

3.  $y_{\min}(k)$  :=  $C(k) * \hat{x}_{\min}(k)$ ;

4.  $g_{\text{anho}}(k)$  := ( $P_{\min}(k) * m_{\text{trans}}(C(k) * \text{minv}(C(k) * P_{\min}(k) * m_{\text{trans}}(C(k)) + R(k)))$ );

5.  $\hat{x}(k)$  :=  $\hat{x}_{\min}(k) + (g_{\text{anho}}(k) * (y(k) - y_{\min}(k)))$ ;

6.  $P(k)$  := ( $\text{id}(n) - (g_{\text{anho}}(k) * C(k))$ ) *  $P_{\min}(k)$ ;

7.  $\hat{x}_{\min}(k+1)$  :=  $A(k) * \hat{x}(k) + G(k) * u(k)$ ;

8.  $P_{\min}(k+1)$  := ( $A(k) * P(k) * m_{\text{trans}}(A(k))$ ) +  $Q(k)$ ;}

```

Figura 4.4: Pseudocódigo do núcleo do algoritmo de *Kalman*.

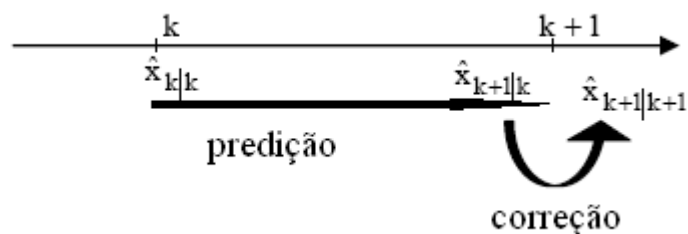


Figura 4.5: Etapas do filtro de *Kalman*: predição e correção.

Para o projeto e análise do filtro de *Kalman* considera-se somente a coordenada Z , com as ordens das matrizes A , G e C sendo 3×3 , 3×1 e 1×3 , respectivamente e, ainda, o vetor de estado, x_k , com ordem 3×1 e os ruídos, ξ_k e η_k , escalares.

No pseudocódigo do núcleo do algoritmo de *Kalman*, considera-se $\text{xhatmin}(k)$ e $\text{xhat}(k)$, sendo o estado estimado *a priori* (\hat{x}_k^-) e *a posteriori* (\hat{x}_k), respectivamente e $\text{Pminus}(k)$ a covariância do erro estimado *a priori*, P_k^- .

Análise do Núcleo do Algoritmo de *Kalman*

Baseando-se no núcleo ilustrado na Figura 4.4, segue que a correção ocorre nos passos de 3 a 6 e a predição, nos passos 7 e 8. Para cada um dos passos e, considerando-se as ordens das matrizes utilizadas, tem-se 92 adições, 132 produtos, 3 divisões e 10 subtrações para cada medida. De um modo genérico para as matrizes A , G e C de ordem $n \times n$, $n \times 1$ e $1 \times n$, respectivamente; o vetor x_k com ordem $n \times 1$ e os ruídos, ξ_k e η_k , sendo escalares, tem-se as seguintes quantidades de operações para cada medida do vetor y_k :

$$SOMA = 3n^3 + n^2 + n - 1; \quad (4.30)$$

$$SUB = n^2 + 1; \quad (4.31)$$

$$PRODUTO = 3n^3 + 4n^2 + 5n; \quad (4.32)$$

$$DIV = n. \quad (4.33)$$

Incertezas Identificadas

Neste trabalho procura-se deixar em evidência que não se consegue determinar o valor exato de uma grandeza. Seria interessante se os dados desejados fossem obtidos de modo exato, porém a experiência nos mostra que somente existe uma aproximação dos valores medidos indiretamente com os valores desejados. A incerteza pode ser substancialmente reduzida quando se utiliza o métodos de medição mais adequado a aplicação e se minimiza ou elimina-se os fatores que possam propiciar resultados indesejáveis, de acordo com cada um desses métodos. Podem ser consideradas também como principais fontes de incertezas nos resultados:

- **incertezas do modelo do sistema** - o modelo pode não descrever o sistema real;
- **incertezas numéricas** - a inexatidão aritmética pode prejudicar as respostas, o algoritmo poder ser, numericamente, instável; os métodos podem ser finalizados prematuramente, ou seja, antes que ocorra a convergência dos resultados;
- **incertezas do *software*** - o programa pode gerar erros devido ao método numérico, ao algoritmo e à biblioteca do *software* que podem ser impróprios ou inexatos; ou ainda, a execução pode estar incorreta devido à erros em compiladores ou dados que se perdem ou que são carregados pelo sistema, usando o programa errado, ou eventos semelhantes.

4.8 Conclusões

Embora se esteja tratando de assuntos da "ciência exata", nas medições de grandezas físicas não se consegue obter os valores exatos dessas grandezas; as medições diretas e/ou indiretas são afetadas pela inexatidão. De modo razoável, pode-se admitir que o valor exato existe e que, embora ele não seja conhecido, seja possível realizar a estimação desse valor e estabelecer limites do intervalo em que ele se encontra. As considerações sobre a incerteza em uma medição indireta devem ser tomadas como forma de avaliar o grau de confiança nos resultados obtidos. Nas medições indiretas é necessário ter bastante cuidado no desenvolvimento das etapas da medição. Com o levantamento detalhado do modelo do sistema, o método que melhor se aplica, entre outros itens que foram mencionados é provável que os resultados obtidos sejam próximos entre si e não se afastem muito do verdadeiro valor. O objetivo deste Capítulo foi apresentar as etapas de desenvolvimento de uma medição indireta para sistemas dinâmicos no qual o valor estimado representa a melhor estimativa do verdadeiro valor da grandeza e mencionar sobre a inexatidão dessa medida. É importante, também, abordar que, independentemente do tipo de medição, o resultado de uma medição está sujeito a erros de diversas fontes.

Capítulo 5

Estudos de Casos

Neste Capítulo são apresentados quatro estudos de casos sobre a medição indireta que focalizam o assunto deste trabalho.

1. Medição Indireta da Velocidade de um Veículo Aeroespacial;
2. Medição Indireta da Temperatura no Interior de um Objeto dentro de uma Estufa;
3. Medição Indireta da Velocidade Rotórica de um Motor de Indução;
4. Medição Indireta Robusta de Velocidade.

Nos Estudos de Casos 1 e 2, utiliza-se o método da filtragem de *Kalman* e nos Estudos de Casos 3 e 4, a Rede Neuronal Artificial.

Uma análise sobre o sistema embarcado ótimo do filtro de *Kalman* é realizada no item 4.7. Esses quatro estudos de casos apresentados refletem a viabilidade do uso de *SMI*. Com base nesses estudos, foram realizadas análises comparativas entre os

resultados obtidos por meio do *SMI* e os valores medidos diretamente. Por meio dessa análise, verifica-se que os resultados obtidos pelos *SMI* apresentam bom desempenho.

5.1 Medição Indireta da Velocidade de um Veículo Aeroespacial

O método de medição indireta baseado em filtro de *Kalman* é utilizado neste item para medir indiretamente a velocidade de um veículo aeroespacial. Os sinais filtrados utilizados nesta aplicação foram do banco de dados do Sistema de Rastreamento de Foguetes do *Centro de Lançamento de Alcântara - Brasil* (CLA) fornecidos pelos radares para um foguete de teste, o *VS-40*. Os sinais de rastreamento medidos pelos radares são transmitidos para computadores locais a uma frequência de 20 *Hz*. Estes sinais consistem de informações do *status* do radar, relação sinal ruído (SNR) e, principalmente, posição do alvo que está sendo rastreado na forma de elevação, azimute e distância do foguete. Na saída do sistema, portanto, têm-se os sinais filtrados e os valores da posição e da velocidade estimada do foguete. Após tratamento, os dados de cada um dos radares são enviados, a uma frequência de 10 *Hz*, aos computadores centrais ou *Sistemas de Tratamento de Dados de Localização* (STDL), que, dentre outras funções, geram a visualização da trajetória percorrida pelo foguete e fazem a previsão do *ponto de impacto* (PI), com a superfície terrestre. O *PI* reflete o nível de ruído do sinal fil-

trado e o tempo de reação do filtro nas transições entre as fases propulsada e balística. O cálculo do PI tem como parâmetros de entrada a posição e a velocidade do veículo aeroespacial. Um sistema de previsão de trajetória de foguete que apresente vantagens em relação ao sistema de filtragem atualmente utilizado é bastante interessante do ponto de vista tecnológico e da segurança de vôo. A previsão do PI depende da exatidão e da estabilidade do filtro, cujos produtos são valores filtrados da posição e da velocidade do alvo em relação à rampa de lançamento definida para o rastreo. Atualmente, o filtro utilizado pelo CLA , filtro $\alpha\beta\gamma$, é sub-ótimo, caracterizando-se por possuir ganhos constantes para algum valor de $\alpha\beta\gamma$ [123], [124].

Neste item, é utilizada uma arquitetura programável da *Cypress System* para implementação em *hardware* do Filtro de *Kalman*. Antes, porém, o desempenho do FK é verificado por meio do MATLAB versão 6.0, linguagem C , e, posteriormente, implementado em *hardware*. O modelo do sistema de rastreo do veículo aeroespacial desenvolvido é apresentado no Apêndice A.

O Processo do Rastreo

Os radares informam somente a posição do foguete e o PI é uma variável estimada a partir de posição e velocidade estimadas, obtidas por meio de técnicas de filtragem. A evolução do PI reflete a qualidade de uma filtragem, em termos de níveis de ruído e velocidade de convergência. Em condições normais de rastreo de foguetes

de multiestágios, tem-se mudanças bruscas de acelerações entre fins de queima de estágios e reinício de queima de estágio seguinte, alterando o comportamento dinâmico. Em outras condições, tem-se a perda de rastreamento por parte dos radares em pleno vôo e posterior reaquisição do rastreamento instantes depois. Tais condições devem ser levadas em consideração no processo de filtragem.

Um dos principais objetivos da Segurança de Vôo durante o rastreamento de foguetes é a capacidade de acompanhar a sua trajetória e acertar a localização do *PI* com a superfície terrestre, considerando a fase balística a cada instante. A implementação de um filtro adaptativo e robusto é necessária devido ao comportamento dinâmico do veículo multiestágio e essas conseqüências são refletidas na aquisição de dados pelos radares. Portanto, os ganhos devem ser adaptados, ao contrário do filtro $\alpha\beta\gamma$ a ganhos constantes e que considera apenas uma fase propulsada e uma balística [124].

O comportamento do filtro deve ser avaliado considerando a velocidade de convergência e o nível de ruído no sinal filtrado. Um bom indicador desses critérios é a estimativa do *PI* nas seguintes situações: a) entre fases fim de queima de um estágio e início de queima do seguinte, filtragem na fase balística final; b) entre certas condições operacionais, tal como instantes entre perda e recuperação do rastreamento do alvo.

Alguns aspectos de tempo real são apresentados para o desenvolvimento do algoritmo. As restrições para a caracterização do processo de filtragem e *hardware* em tempo real são especificadas. O sistema em estudo pode ser considerado como um sistema crítico e, neste caso, possui restrições rígidas quanto ao tempo.

O problema é formulado utilizando métodos de filtragem estocástica de *Kalman*. As considerações para o desenvolvimento do algoritmo de *Kalman* são especificadas no item 3.2 e no Apêndice A, conforme mencionadas em [101], [125] e [126].

5.1.1 Arquitetura para o Filtro de *Kalman*

Um *FK* foi implementado em *DSP* por *Lewis*[127]. Algumas subrotinas foram utilizadas, como: uma subrotina inicial na qual foram armazenados os coeficientes do filtro na memória de dados e as condições iniciais; outra subrotina para obter os valores medidos y_k e os valores de entrada u_k e, em seguida, armazenados na memória de dados. De modo semelhante ao processo adotado para simulação de filtragem do veículo aeroespacial, o autor obteve os valores medidos y_k por meio de simulação e, em seguida, realizou uma comparação entre os resultados obtidos por simulação e por implementação. Nesse trabalho, o autor menciona que a implementação convergiu rapidamente e, provavelmente, devido às considerações iniciais que adotou, como, velocidade estimada ($\hat{v}(0)$) *zero* com período de amostragem 0,05 s. Uma outra arquitetura, o *FPGA*, foi utilizada em [128] por *Al-Dhafer* para implementação do *FK*. Nesta implementação, vários sensores foram usados para fornecer o valor da medida a ser filtrada pelo *FK*. Segundo o autor, a multiplicação entre matrizes implementada em *FPGA* apresentou alto desempenho com uso mínimo da área.

A arquitetura utilizada neste trabalho é da *Cypress Microsystems* [129]. O processador *MC8* possui frequência de até 24 MHz, arquitetura *Harvard* com 4 *MIPS*, de 8 *bits*, e permite a configuração de blocos digitais e analógicos. O *clock* do processador pode funcionar a baixas velocidades, reduzindo o consumo de energia. Neste trabalho, o *FK* é utilizado para filtrar sinais de posição de um veículo aeroespacial e obter, indiretamente, a velocidade do veículo para um foguete de teste do *CLA-MA*. Considera-se somente a coordenada *Z* em km. As características gerais da arquitetura são: 3,0 - 5,25 V, 16 Koctetos memória *flash*, 256 octetos de *RAM*, 8 blocos digitais e 12 blocos analógicos. A memória de programa é somente para armazenar códigos do programa, tabelas constantes, valores iniciais e caracteres para variáveis globais. O compilador gera uma memória imagem em forma de arquivos de saída de hexadecimal em *ASCII*. Na Figura 5.1, considera-se y_k como os sinais de uma aeronave obtidos pelo radar e \hat{x}_k , a medição indireta da posição.

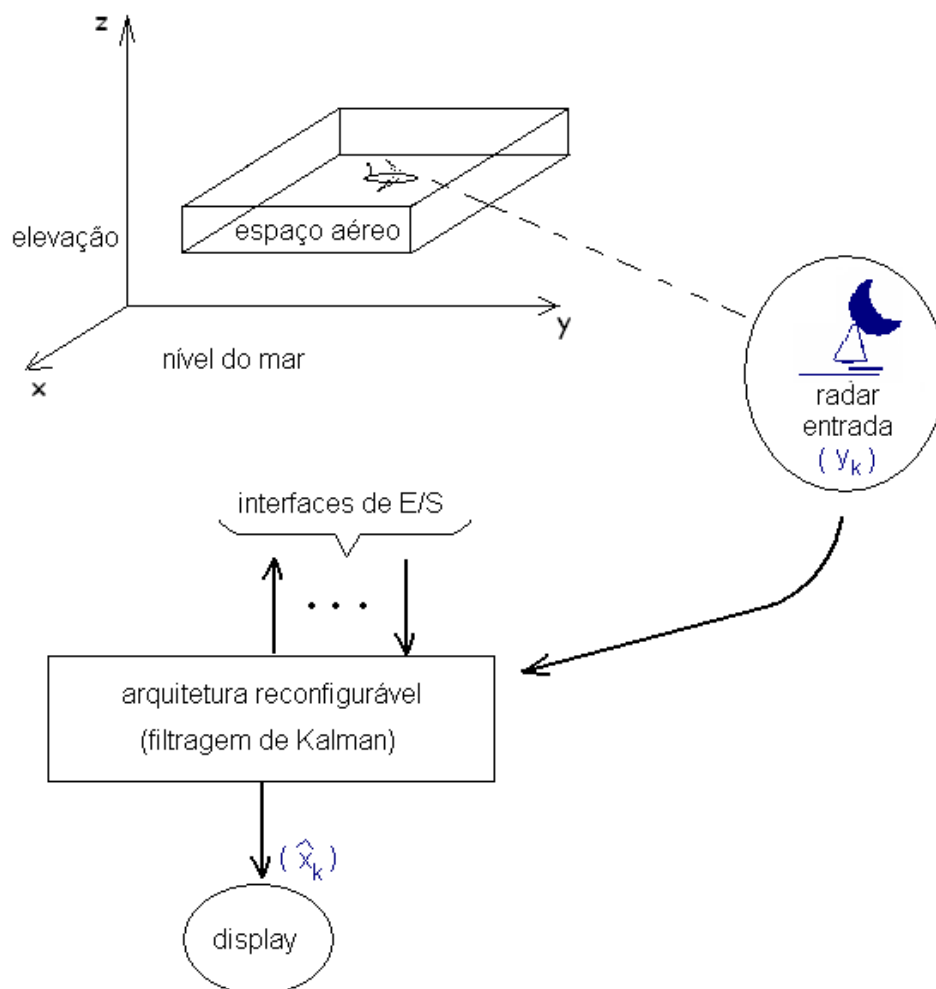


Figura 5.1: Filtro de *Kalman* em tempo real.

Por meio do *software* da arquitetura é possível configurar e programar uma grande variedade de opções. Pode-se adicionar componentes discretos e adequar as configurações da arquitetura quanto à velocidade de processamento e consumo; é possível entrar com ajustes globais necessários para configurar o próprio *chip* [130].

O *FK* é implementado na arquitetura e os valores de referência são os sinais de um

banco de dados de trajetória do *CLA-MA*. Esse algoritmo é também desenvolvido em linguagem *C* versão 4.5 no ambiente *PC Windows*. Utiliza-se a versão 4.1, *Programmable System-on-Chip (PSoC) Designer*.

O algoritmo é desenvolvido em 81 linhas de código *C*, com 159 octetos de RAM e 55% de memória *flash*. Utiliza-se 100 amostras de sinais de dados da coordenada *Z*. Esse conjunto de amostras é armazenado em uma tabela constante na memória *flash*, ocupando 400 octetos. As variáveis em ponto flutuante e inteiras são de 4 octetos e 2 octetos, respectivamente. Embora a arquitetura utilizada tenha sido de 8 *bits*, a implementação usa palavras de 16 *bits*. Da porta 0 obtém-se os valores de saída da posição do veículo na coordenada *Z*, sendo visualizada por meio de um *display* externo. O módulo *LCD* da arquitetura é utilizado para realizar a conexão com o *LCD* externo. O módulo *TIMER* de resolução 1 μ s é acrescentado ao projeto somente para verificar o tempo de execução que é de 367 ms. O arquivo principal do algoritmo em *C* do projeto tem tamanho de 8 Koctetos e os arquivos auxiliares *.rom* e *.hex* têm tamanhos 29 Koctetos e 36 Koctetos, respectivamente. **Dados da Arquitetura Reconfigurável:**

- frequência: *clock* de 24 MHz;
- tensão: 4,95 V.

A arquitetura reconfigurável usada neste projeto pode ser ajustada usando um bloco *conversor A/D* para receber diretamente esses sinais dos radares e realizar o processo de filtragem. Se esses sinais forem enviados pelos radares aos computadores a uma taxa

de 20 Hz, então o valor máximo de *clock* da CPU do PSoCTM pode ser de até 24 MHz. Os módulos a serem utilizados, neste caso, são: *módulo do conversor A/D*, *módulo LCD* e o *módulo TIMER*.

5.1.2 Resultados Experimentais

Na simulação computacional é possível verificar o desempenho do *FK*. Considerando os recursos da simulação, o desenvolvimento do filtro pode ser observado a partir dos gráficos da estimação da posição, da velocidade e da aceleração do veículo aeroespacial na coordenada *Z*. Verifica-se, principalmente, por meio do gráfico da velocidade ilustrado no Apêndice A, Figura A.4, que durante a fase propulsada as equações do movimento se assemelham às do *Movimento Retilíneo Uniformemente Acelerado* (MRUA), isto por que nessa fase o veículo apresenta aceleração de propulsão que varia rapidamente de *zero* até um certo patamar de aceleração, mantendo-se, aproximadamente, constante até instantes finais do estágio, caindo rapidamente para zero. Na fase balística, tem-se um movimento semelhante às de um projétil, instantes em que o veículo está sujeito unicamente à força gravitacional, despreza-se, neste caso, a resistência do ar. Com uma taxa de amostragem de 0,1 s, observa-se um *overshoot* em torno de 2,0 s provocado pela ação do filtro na estimação da velocidade, após a mudança para a fase balística. O comportamento do filtro é ainda mais pronunciado para a estimação da aceleração, conforme pode ser observado no Apêndice A, Figura A.5. Nesta figura, verifica-se que

a aceleração está ligeiramente defasada na fase propulsada e que logo após a passagem brusca para a fase balística, há um *overshoot* da curva estimada para posterior acomodação. Na Figura A.6, Apêndice A, tem-se as curvas do ganho de *Kalman*. Após o processo de simulação da filtragem de *Kalman* no ambiente *MATLAB*, é realizado o desenvolvimento do algoritmo na linguagem *C*, versão 4.5 e em seguida na linguagem própria da arquitetura para verificar indiretamente a velocidade do veículo aeroespacial. O desempenho do filtro é verificado e comparado com os valores de sinais de trajetória disponíveis no banco de dados do *CLA-MA*.

5.2 Medição Indireta da Temperatura no Interior de um Objeto dentro de uma Estufa

O conhecimento do comportamento de um sistema térmico pode ser avaliado a partir de observações de certas variáveis. Algumas dessas variáveis apresentam dificuldades para se observar devido ao acesso para se inserir o sensor e/ou devido ao elevado nível de ruído. O método de medição indireta baseado em modelos de processos permite a avaliação dos estados do sistema por meio de outras variáveis medidas diretamente.

Nesse item é apresentado o desenvolvimento do *modelo estocástico* que representou o sistema térmico e o sistema embarcado. No item 5.2.1 são verificadas as metodologias para medição da temperatura. O item 5.2.2, Medição Indireta da Temperatura, enfatiza o desenvolvimento da teoria matemática utilizada para estimar a temperatura; apresenta-se o sistema térmico descrito no espaço de estados e o estimador. No item 5.2.2 são apresentados os resultados computacionais por meio de simulações. Os resultados experimentais mencionados no item 5.4.4 foram obtidos no sistema embarcado implementado em uma arquitetura reconfigurável que executou a medição indireta da temperatura do sistema. E, no último item são apresentadas as conclusões e comentários acerca do sistema de medição indireta desenvolvido. A fundamentação teórica desse estudo de caso tem como referências [105], [131]-[139].

5.2.1 Modelo do Sistema Térmico

O desenvolvimento do modelo do sistema térmico é realizado baseado nos conceitos da termodinâmica, da transferência de calor e da mecânica dos fluidos. Apresenta-se um exemplo do mundo real e a partir de sinais de medições obtém-se indiretamente os parâmetros de uma função de transferência de um sistema térmico. Deseja-se monitorar a temperatura no interior de um objeto que se encontra dentro de uma estufa sujeita à variações da temperatura. São consideradas as variações da temperatura ambiental e um resistor para realizar o controle de temperatura do objeto.

Considera-se o sistema térmico da Figura 5.2. Nesse sistema tem-se T_2 que representa a temperatura interna da estufa; T_1 , sendo a temperatura no interior de um objeto; T_0 , a temperatura ambiente e R , a resistência elétrica da estufa. A metodologia de identificação apresentada em *Modelo Dinâmico* é útil para monitoração e controle de temperatura quando todos os objetos inseridos na estufa estão associados a um sensor. Já na abordagem apresentada em *Modelo Térmico*, a metodologia de identificação é adaptada para medição indireta. Nesse caso, os modelos são baseados nas leis da Física para estimar a temperatura no interior de um objeto que se encontra dentro da estufa que está sob variações de temperatura por meio de uma resistência elétrica.

Modelo Dinâmico

O modelo dinâmico do sistema térmico é representado pela equação à diferenças,

$$y_k = -a_{n-1}y_{k-1} + \dots - a_0y_{k-n} + b_{n-1}u_{k-1} + \dots + b_0u_{k-n}, \quad (5.1)$$

sendo y_k e u_k as variáveis discretas que representam a temperatura $T_1(t)$ e a excitação $u(t)$ da fonte de calor contínuas no tempo, respectivamente. Reescrevendo a Equação (5.1), tem-se,

$$T_k = -a_{n-1}T_{k-1} + \dots - a_0T_{k-n} + b_{n-1}u_{k-1} + \dots + b_0u_{k-n}, \quad (5.2)$$

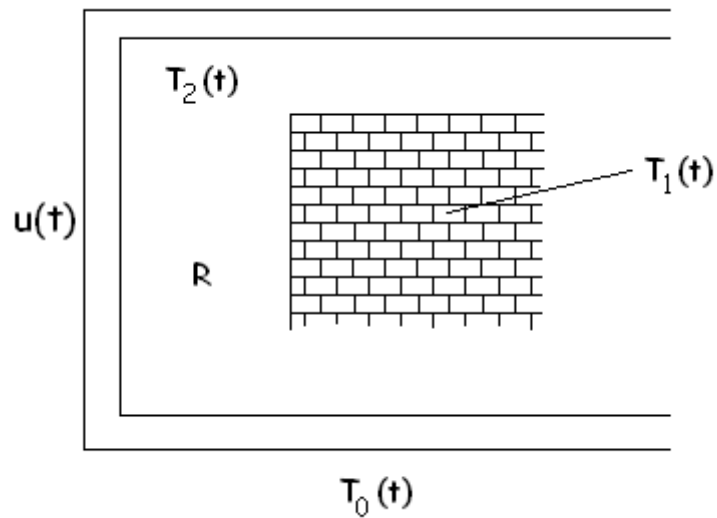


Figura 5.2: Representação de um sistema térmico.

Baseando-se nas referências e aplicando as leis da Física que regem os sistemas térmicos, considera-se que uma equação de segunda ordem seja suficiente para representar o comportamento dinâmico da temperatura no objeto. Dessa forma, a estrutura do modelo pode ser escrita como:

$$T_k = -a_1 T_{k-1} - a_0 T_{k-2} + b_1 u_{k-1} + b_0 u_{k-2}, \quad (5.3)$$

ou seja,

$$T_k = \begin{bmatrix} -T_{k-1} & -T_{k-2} & u_{k-1} & u_{k-2} \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \\ b_1 \\ b_0 \end{bmatrix}, \quad (5.4)$$

sendo a_1 , a_0 , b_1 e b_0 os parâmetros desse modelo.

Modelo Térmico

Na Figura 5.3 tem-se a representação da estufa, na qual são colocados os principais elementos do sistema térmico a ser modelado. A estufa utilizada é do fabricante *Odontobrás*, cujas dimensões internas são *largura* = 31,00 cm, *altura* = 32,00 cm e *profundidade* = 29,5 cm; com o peso líquido de 28,50 kg e volume de 30 l. Possui

um resistor, R , que dissipa uma potência igual à taxa de transferência de calor do meio 2, $T_2(t)$, para o meio externo, $T_0(t)$. A temperatura do modelo térmico é baseada no conceito de capacitância térmica. O calor total armazenado é dado pelo calor específico, c_i , e pela massa do objeto que se encontra dentro da estufa. A temperatura varia como uma função do tempo quando a energia térmica é armazenada na estufa devido à taxa de variação do fluxo de calor.

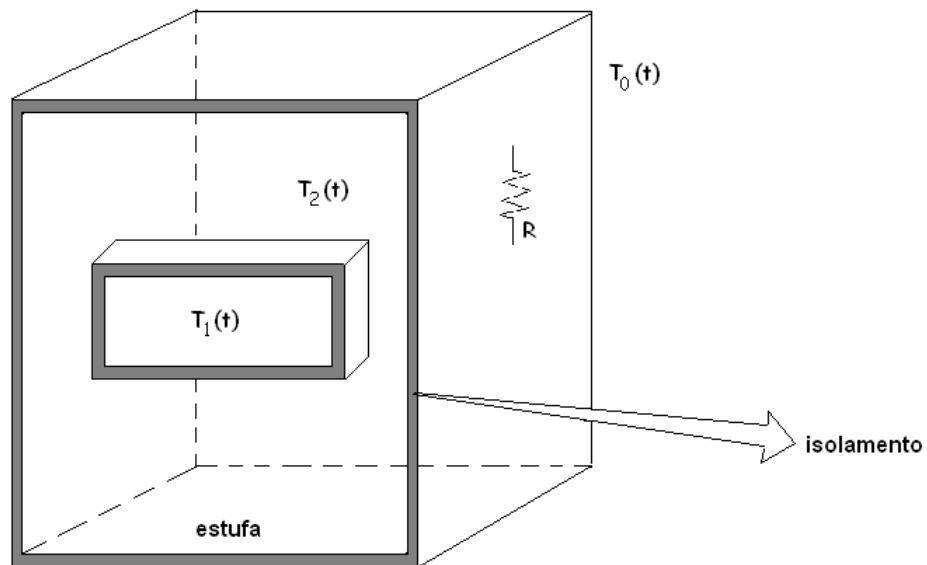


Figura 5.3: Representação da estufa a energia elétrica.

O modelo do sistema térmico baseia-se nos princípios da conservação de energia e nas leis da termodinâmica que representam a transferência de calor, ou seja,

$$m_i c_i \frac{dT}{dt} = Q_h, \quad (5.5)$$

sendo m_i e c_i a massa e o calor específico do i -ésimo meio e Q_h é o fluxo de calor. A Equação (5.5) tem a forma da variável de fluxo que é proporcional à taxa de variação da temperatura como uma função do tempo.

Esse modelo térmico representa a transferência de calor entre os dois meios, ou seja, o meio 2 e o meio 1 na presença da temperatura ambiente $T_0(t)$. Para cada um dos meios, a equação diferencial resultante que representa o comportamento do sistema é expressa como uma função da temperatura:

$$m_1 c_1 \frac{dT_1}{dt} = k_{12}(T_2 - T_1) \quad (5.6)$$

e

$$m_2 c_2 \frac{dT_2}{dt} = -k_{12}(T_2 - T_1) - k_{20}(T_2 - T_0) + u(t), \quad (5.7)$$

sendo k_{ij} condutâncias térmicas entre os meios [133].

As variações da temperatura nas Equações (5.6) e (5.7) representam o potencial de impulso que provoca o fluxo de calor.

A seguir, na Figura 5.4, tem-se o sistema de fluxo de calor e o circuito térmico equivalente. Sendo $R_{t,conv,2}$, a variável que representa a resistência térmica por convecção no meio 2; $R_{t,cond}$, a resistência térmica por condução que o material oferece ao fluxo de calor e $T_{s,1}$, a temperatura na superfície do material. A resistência térmica é a

recíproca da condutância térmica. A resistência térmica total, R_{total} é dada por,

$$R_{total} = R_{t,conv,2} + R_{t,cond}. \quad (5.8)$$

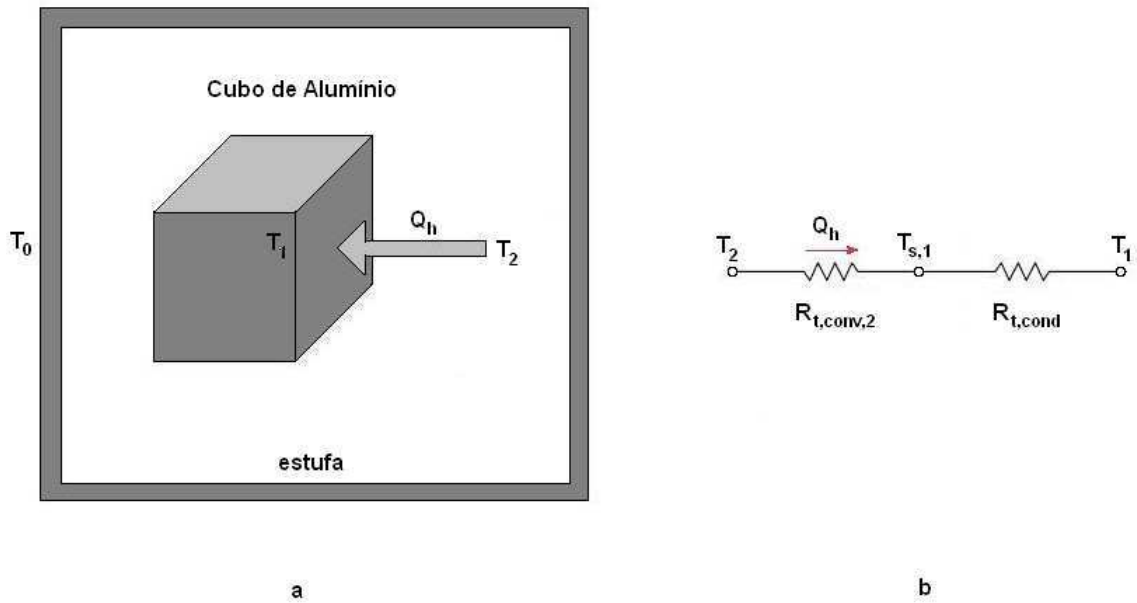


Figura 5.4: a) Distribuição de temperatura através do material; b) Circuito térmico equivalente.

5.2.2 Medição Indireta da Temperatura

Neste item, apresenta-se o modelo da medição indireta da temperatura na estufa. O modelo do sistema e a forma de estimação de *Gauss-Legendre* são os elementos básicos do *SMI* para o processo de estimação, predição e filtragem de sinais reais representados

pelo modelo estocástico. O modelo do sistema é utilizado para prever as variáveis e a forma de *Gauss-Legendre* levou em consideração os valores medidos para melhorar o resultado da medição indireta. O método de medição indireta baseado em filtro de *Kalman* é utilizado para filtrar e prever o comportamento da temperatura na estufa que pode ser controlada por meio de um resistor. A descrição no espaço de estados a seguir apresenta o modelo da estufa baseado nas leis da Física. No item 5.2.2 tem-se a filtragem de *Kalman* padrão para aplicação em medição indireta da temperatura.

Descrição no Espaço de Estados

O sistema térmico modelado no item 5.2.1 dado pelas Equações (5.4) e (5.7) é representado na forma de espaço de estado por:

$$\dot{x} = Ax + Bu, \quad (5.9)$$

$$y = Cx, \quad (5.10)$$

sendo $A^{n \times n}$, a matriz que representa a dinâmica do sistema; $B^{n \times m}$, a ponderação do controle e $C^{p \times n}$ matriz constante de saída. A saída do sistema, y , é dada por uma combinação linear dos estados. Substituindo o vetor x como $x_1 = T_1$ e $x_2 = T_2$ nas

Equações (5.4) e (5.7) desenvolvidas no item 5.2.1, tem-se:

$$m_1 c_1 \dot{T}_1 = k_{12}(T_2 - T_1) \quad (5.11)$$

e

$$m_2 c_2 \dot{T}_2 = -k_{12}(T_2 - T_1) - k_{20}(T_2 - T_0) + u(t), \quad (5.12)$$

colocando na forma matricial, segue

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \end{bmatrix} = \begin{bmatrix} \frac{-k_{12}}{m_1 c_1} & \frac{k_{12}}{m_1 c_1} \\ \frac{k_{12}}{m_2 c_2} & \frac{-k_{12}-k_{20}}{m_2 c_2} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_2 c_2} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{k_{20}}{m_2 c_2} \end{bmatrix} T_0. \quad (5.13)$$

Estimador de Estados da Temperatura

O algoritmo do filtro de *Kalman* produz uma estimativa ótima da temperatura na estufa, minimizando as incertezas acerca da predição dessa temperatura. Considerando $\hat{x}_k^- \in \mathbb{R}^n$ o estado estimado *a priori* no passo k conhecendo-se o sistema e $\hat{x}_k \in \mathbb{R}^n$ o estado estimado *a posteriori* no passo k , conhecendo-se a medida y_k , tem-se que os erros de estimação de y_k *a priori* e *a posteriori* são,

$$e_k^- = x_k - \hat{x}_k^- \quad (5.14)$$

e

$$e_k = x_k - \hat{x}_k. \quad (5.15)$$

A covariância do erro de estimação *a priori* é

$$P_k^- = E[e_k^- e_k^{-T}] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \quad (5.16)$$

e a covariância do erro de estimação *a posteriori* é,

$$P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]. \quad (5.17)$$

Segue, então, a predição da medida,

$$y_k^- = C_k \hat{x}_k^-. \quad (5.18)$$

Na Equação (5.19) obtém-se a estimativa *a posteriori* \hat{x}_k como uma combinação linear da estimação *a priori* \hat{x}_k^- com a inovação da medida (resíduo), $(y_k - y_k^-)$, que fornece a discrepância entre a medida atual e a medida prevista:

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - y_k^-). \quad (5.19)$$

De modo semelhante ao item 5.1, segue que para as equações diferenciais lineares estocásticas (A.3), $x_{k+1}^i = A_k^i x_k^i + G_k^i \xi_k^i$, e (A.4), $y_k^i = C x_k^i + \eta_k^i$, considerou-se que os ruí-

dos do sistema ξ_k e da medida η_k são independentes e com distribuições probabilísticas normais:

$$p(\xi_k) \sim N(0, Q_k) \quad (5.20)$$

$$p(\eta_k) \sim N(0, R_k) \quad (5.21)$$

$$E(\xi_k \eta_i^T) = 0 \quad (5.22)$$

$$E(\xi_k \xi_i^T) = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}$$

$$E(\eta_k \eta_i^T) = \begin{cases} R_k, & i = k \\ 0, & i \neq k. \end{cases}$$

Na Figura 5.5 têm-se os procedimentos *Kalman* utilizados para a medição indireta da temperatura. O processamento do algoritmo é inicializado com a correção da medida e em seguida com a predição da medida, ou seja, com a predição da temperatura no interior do objeto.

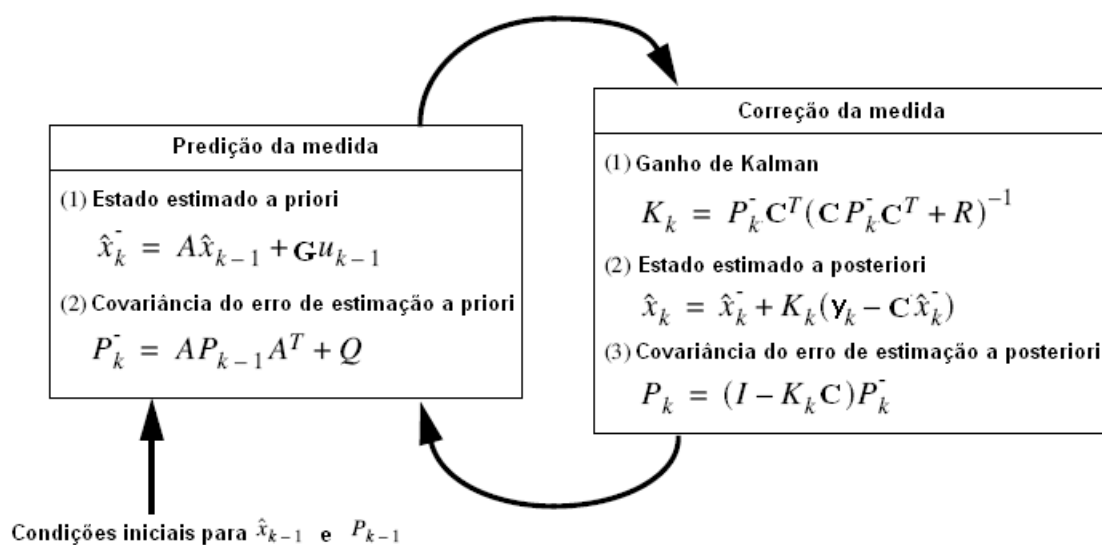


Figura 5.5: Procedimentos utilizados pela filtragem de *Kalman* para a medição indireta da temperatura.

5.3 Calor Específico Estimado

O calor específico, c_i , dos materiais (metais) utilizados neste estudo de caso é estimado por meio do *Laboratório Virtual de Química* (ChemLab), versão 1.2a. Neste laboratório o calor específico do material é estimado por meio do aquecimento. O material é aquecido inicialmente e, em seguida, colocado numa quantidade conhecida de água em um calorímetro. Mede-se a temperatura resultante.

O calor é transferido entre o metal aquecido e a água, tal que a perda de calor pelo metal é igual ao ganho de calor pela água, sendo expresso por,

$$(m_i) \times (c_i) \times [T(0)_i - T(F)_i] = (m_{H_2O}) \times (c_{H_2O}) \times [T(0)_{H_2O} - T(F)_{H_2O}]$$

Sendo m_i , a massa do metal; c_i , o calor específico do metal; $T(0)_i$, a temperatura inicial do metal; $T(F)_i$, a temperatura final do metal; m_{H_2O} , a massa da água; c_{H_2O} , o calor específico da água; $T(0)_{H_2O}$, a temperatura inicial da água e $T(F)_{H_2O}$, a temperatura final da água.

A temperatura final da água será a mesma do metal. Considera-se o calor específico da água sendo 1 cal/gm°C.

Utilizou-se os procedimentos especificados pelo Laboratório na seguinte seqüência:

- a amostra do metal é colocada em um tubo de ensaio. Em seguida, é adicionado um termômetro dentro do tubo de ensaio para medir a temperatura inicial do material;
- utiliza-se um *beaker* contendo 150 ml de água e, posteriormente, inserido o tubo de ensaio com o material;
- o conjunto é aquecido até o ponto de ebulição da água, 100 °C;
- O material aquecido é retirado do *beaker* e derramado dentro de um calorímetro, contendo 100 ml de água. O calorímetro estando com temperatura de 20 °C;
- O calorímetro é fechado. O valor da temperatura final dentro do calorímetro é obtido por meio de um termômetro e em seguida registrado.

Após realizar os procedimentos anteriores, o valor do calor específico de cada um dos materiais pode ser obtido por meio das seguintes anotações:

- temperatura inicial do metal;
- temperatura final do metal;
- temperatura inicial da água no calorímetro;
- temperatura final da água no calorímetro.

5.4 Testes Computacionais

Os resultados computacionais apresentados neste item são obtidos por meio de simulações do *SMI* proposto. Esses resultados são utilizados para análise e síntese de dispositivos digitais baseados nos conceitos que são apresentados no item 5.2.2. O sistema embarcado é o núcleo do dispositivo de medição indireta. O sistema embarcado é implementado em uma arquitetura reconfigurável que é usada para desempenhar as principais tarefas da medição indireta da temperatura. O desempenho do algoritmo de *Kalman* é verificado para prever a temperatura no interior do objeto, ou seja, de um cubo de alumínio com peso de 19,50 g inserido dentro da estufa, conforme representado na Figura 5.3.

5.4.1 Descrição Numérica do Espaço de Estados

A descrição numérica do espaço de estados da estufa é a representação do sistema real. Esta representação é obtida substituindo-se na descrição do espaço de estados, Equação (5.13), os valores aproximados para $m_1c_1 = 3,88$, $m_2c_2 = 3049,5$, $k_{12} = 2,08$ e $k_{20} = 0,46$.

$$\begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \end{bmatrix} = \begin{bmatrix} -0,54 & 0,54 \\ 0,00068 & -0,00083 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{3049,5} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{0,46}{3049,5} \end{bmatrix} T_0 \quad (5.23)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}. \quad (5.24)$$

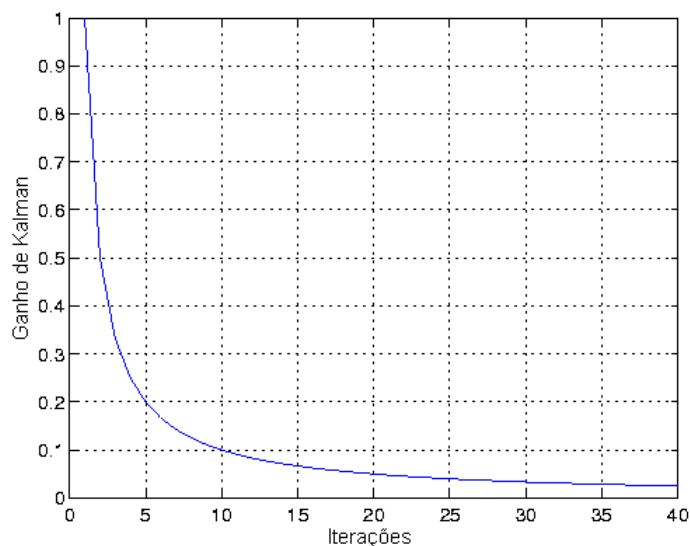
As equações no espaço de estado contínuas são mapeadas em suas equivalentes equações de diferença no espaço de estados. O modelo discreto realiza o passo de predição do algoritmo de filtragem *Kalman*, conforme Figura 5.5.

5.4.2 Procedimento Computacional

Para encontrar os melhores valores da covariância Q e variância R de forma que haja convergência do filtro são realizadas duas simulações. Os resultados obtidos são comparados com valores de temperatura medidos diretamente por meio de um sensor de temperatura colocado no interior do objeto, medindo T_1 , dentro da estufa.

Na primeira simulação, a covariância $Q = 1$ e a variância $R = 0,0001$. Na segunda simulação, os valores de Q e R , são 10 e 0,01, respectivamente. Essa última simulação apresenta os melhores valores quando comparados aos valores de temperatura medidos diretamente no interior do objeto T_1 .

Na Figura 5.6 tem-se a ilustração da evolução do ganho de *Kalman*, e, como pode ser verificado, o valor tende a se fixar em torno da trigésima iteração.

Figura 5.6: Evolução do ganho de *Kalman*

5.4.3 Procedimentos Experimentais

Conforme ilustrado no diagrama da Figura 5.7, é utilizado um sensor de temperatura para medir a temperatura interna da estufa, T_2 e uma arquitetura na qual é inserido o filtro de *Kalman* para medir indiretamente a temperatura no interior do objeto que se encontra dentro da estufa. Além disso, o valor da temperatura é apresentado em um *display*. O *conversor A/D* da arquitetura converte os valores analógicos de tensão em valores digitais. Esse *conversor A/D* interno da arquitetura opera com 8 *bits*. O sensor de temperatura analógico utilizado, *LM35*, tem a linearidade como característica principal. Esse sensor fornece 10 mV/°C. Na Figura 5.8 tem-se ilustrado os aspectos

do objeto utilizado para se verificar o valor da temperatura em seu interior. Na parte superior do cubo de alumínio é realizada uma abertura, Figura 5.8, na qual é inserido um dos sensores *LM35*. Posteriormente, os valores medidos por esse sensor são utilizados para realizar a análise comparativa entre os valores medidos indiretamente pela arquitetura e os valores medidos diretamente pelo sensor de temperatura, denominado por T_1 , conforme ilustra o diagrama da Figura 5.7.

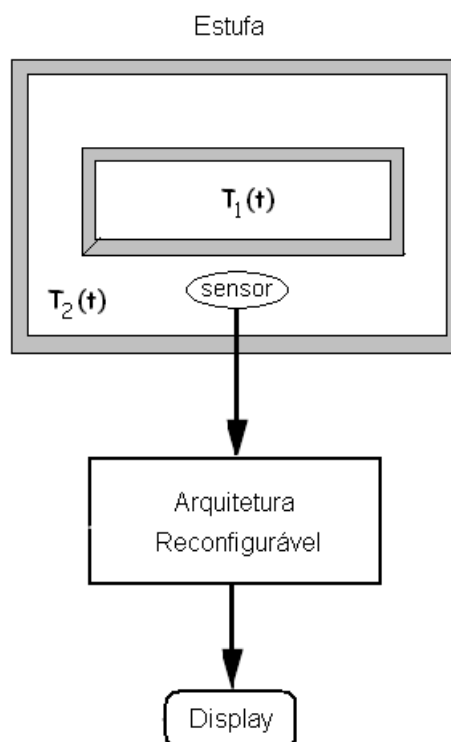


Figura 5.7: Diagrama em blocos da medição indireta de temperatura no meio 1.

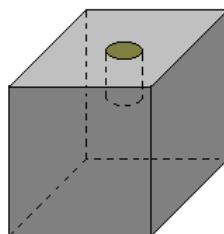


Figura 5.8: Cubo de alumínio utilizado.

Sistema Embarcado

Utiliza-se uma arquitetura reconfigurável para implementar o núcleo do *SMI*. A arquitetura é da *Cypress System - PSoC*, versão 4.0, *módulo LCD* e o *módulo ADC*. Uma *porta de E/S* da arquitetura reconfigurável, (*P0* a *P7*), é utilizada para verificar o valor da temperatura no meio 1. Essa porta é conectada a um *display* externo para visualizar o valor da temperatura. Internamente, essa porta é conectada ao *módulo LCD* da arquitetura. Esta arquitetura possui as seguintes características: frequência - 24 *MHz*; tensão de alimentação - 3,0 a 5,25 V; memória *flash* - 16 Koctetos; RAM - 256 octetos; 8 blocos digitais (temporizadores, contadores, PWM, etc) e 12 blocos analógicos (ADC, DAC, filtros programáveis, comparadores, etc).

O algoritmo do filtro de *Kalman* é desenvolvido e processado em linguagem *C* própria da arquitetura. A partir dessa ferramenta, mede-se indiretamente o valor da temperatura no interior do objeto (cubo de alumínio) que se encontra dentro da estufa.

O desempenho do filtro de *Kalman* é comparado com 340 amostras de valores medidos diretamente a partir dos dois sensores de temperatura: um deles inserido dentro do objeto e outro fora do objeto, porém dentro da estufa. As limitações apresentadas neste projeto correspondem aos formatos dos dados e a biblioteca disponível pela linguagem *C* da arquitetura, capacidade de memória, além da temperatura máxima permissível pelo fabricante da estufa.

5.4.4 Resultados Experimentais

Os valores da temperatura obtidos diretamente e indiretamente no interior do objeto que se encontra dentro da estufa, apresentam algumas discrepâncias, principalmente no início do processo da medição. É necessário levar em consideração que tais discrepâncias foram oriundas de alguns fatores, tais como: as condições iniciais - a massa do material, o calor específico e a condutância térmica que não estavam devidamente ajustadas; a dissipação entre o material e a estufa.

Na Figura 5.9 tem-se a ilustração das temperaturas T_1 no interior do objeto obtida por meio direto e no interior do objeto obtida por meio indireto.

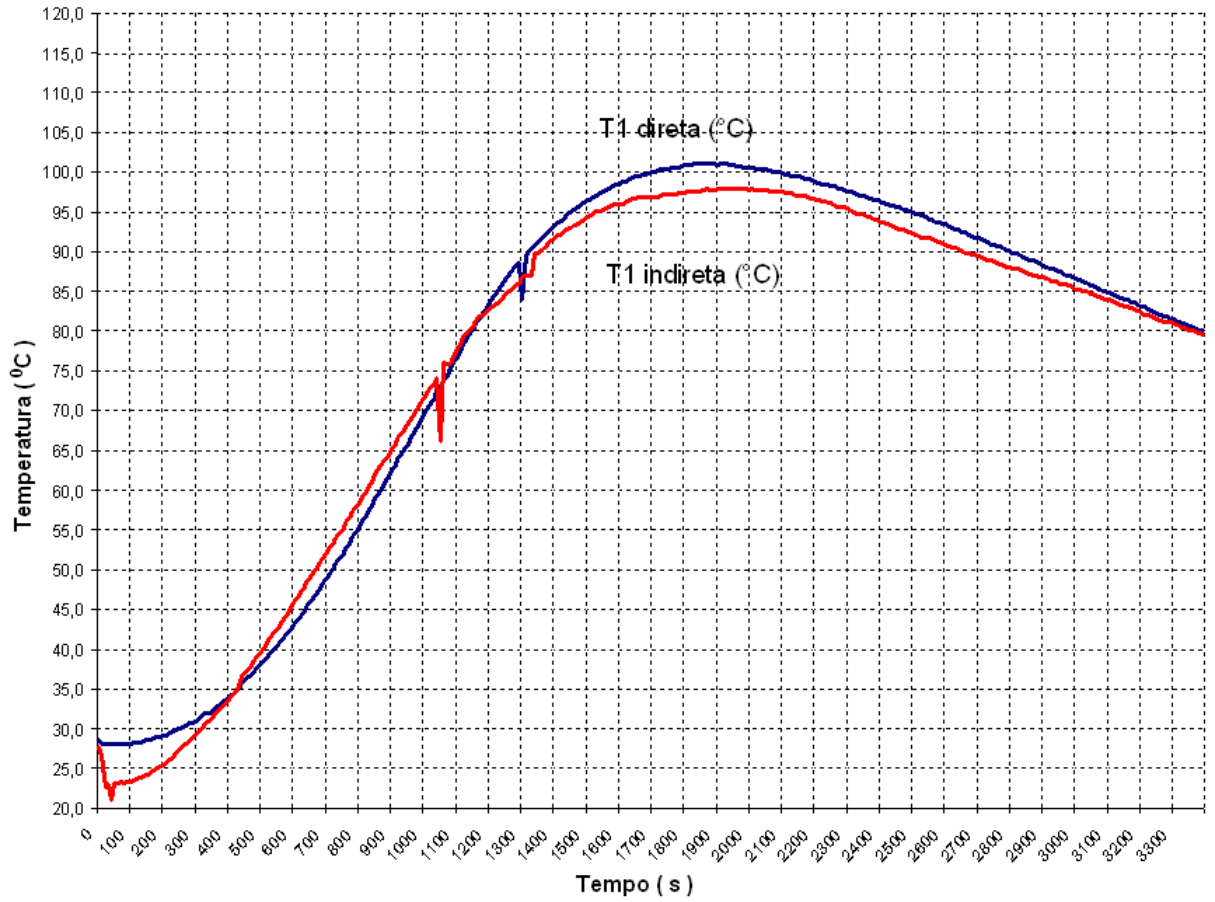


Figura 5.9: Comportamento das temperaturas

A análise comparativa do desempenho entre as medições de temperatura, direta e indireta, é verificada a partir do erro médio, \bar{E} , dado pela Equação (5.25):

$$\bar{E} = \frac{1}{n} \sum_{j=1}^n |T_{1d(j)} - T_{1i(j)}|, \quad (5.25)$$

sendo T_{1d} a medição direta no meio 1, T_{1i} , a medição indireta no meio 1 e n , o número de medições ($n = 340$). O erro médio entre os valores de temperatura medidos indire-

tamente no interior do objeto é de $2,36\text{ }^{\circ}\text{C}$, aproximadamente. A exatidão do sensor utilizado, o *LM35*, conforme Anexo A, é ± 1.5 e é definida como o erro entre a tensão de saída e $10\text{ mV}/^{\circ}\text{C}$, dentro das condições especificadas de tensão, corrente e temperatura expressa em $^{\circ}\text{C}$.

5.5 Medição Indireta da Velocidade Rotórica de um Motor de Indução

Neste item é apresentado o estudo de caso no qual se obtém indiretamente a velocidade rotórica em uma máquina de indução. Nos dois casos desenvolvidos, o método de medição indireta utilizado é o da *RNA*. Analisou-se o comportamento das tensões e correntes estatóricas, fluxo rotórico e velocidade rotórica no ambiente MATLAB. Em outro momento é desenvolvido o algoritmo da *RNA* que, a partir do conjunto de sinais de entrada, realiza o treinamento. A *RNA* estima o valor do fluxo rotórico. A velocidade rotórica é um dos três pesos da *RNA* e, com a minimização do erro, obtém-se a medição da velocidade rotórica. Os resultados são analisados e comparados por meio de simulação e implementação em *hardware*.

5.5.1 Modelo da Máquina de Indução

Neste item, tem-se o modelo da máquina de indução que representa o sistema. É dada uma abordagem sobre o desenvolvimento do estimador neuronal de velocidade rotórica. A excitação da rede neuronal é realizada por meio das tensões e correntes de estator referenciadas a um sistema estacionário $dq0$.

As equações da máquina foram baseadas em [6], [8], [140] e [141]. O circuito elétrico equivalente para operação no espaço de estados de uma máquina de indução simétrica é ilustrado na Figura 5.10,

$$\vec{v}_s = R_s \vec{i}_s + \dot{\vec{\lambda}}_s, \quad (5.26)$$

$$0 = R_r \vec{i}_r + \dot{\vec{\lambda}}_r + j\omega_r \vec{\lambda}_r, \quad (5.27)$$

$$\vec{\lambda}_s = L_s \vec{i}_s + L_m \vec{i}_r, \quad (5.28)$$

$$\vec{\lambda}_s = L_s \vec{i}_s + \vec{\lambda}_m, \quad (5.29)$$

$$\vec{\lambda}_r = L_m \vec{i}_s + L_r \vec{i}_r, \quad (5.30)$$

$$\vec{\lambda}_r = L_r \vec{i}_r + \vec{\lambda}_m, \quad (5.31)$$

sendo λ , o fluxo concatenado; L , a indutância; v , a tensão; R , a resistência; i , a corrente; $\sigma = 1 - \frac{L_m^2}{L_r L_s}$, o coeficiente de dispersão; $T_r = \frac{L_r}{R_r}$, a constante de tempo do rotor; ω_r , a velocidade rotórica e L_m , a indutância de magnetização.

Os subscritos r e s representam valores referentes ao rotor e ao estator, respectivamente e os subscritos d e q representam componentes no eixo dq .

Segue que, $\vec{v}_s = \begin{bmatrix} \vec{v}_{ds} & \vec{v}_{qs} \end{bmatrix}'$, $\vec{i}_s = \begin{bmatrix} \vec{i}_{ds} & \vec{i}_{qs} \end{bmatrix}'$ e $\dot{\vec{\lambda}}_s = \frac{d\vec{\lambda}_s}{dt}$, sendo a tensão induzida no estator.

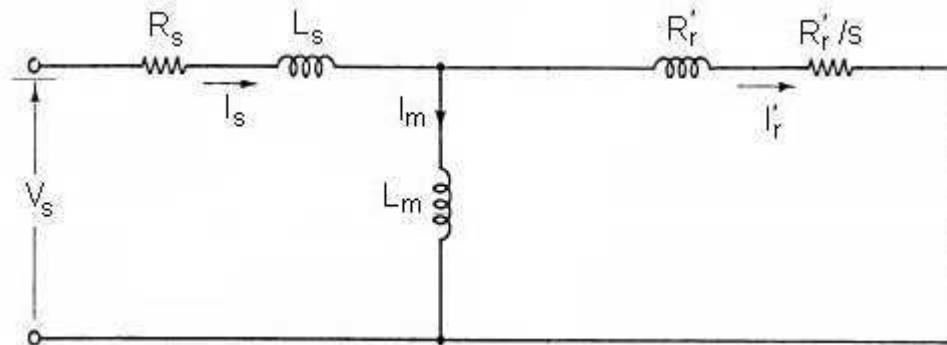


Figura 5.10: Circuito elétrico equivalente.

Considerando $\vec{v}_r = \begin{bmatrix} \vec{v}_{dr} & \vec{v}_{qr} \end{bmatrix}'$ sendo 0 (rotor curto-circuitado) e $\omega = 0$. As

Equações (5.26), (5.27), (5.28) e (5.30) na forma matricial,

$$\begin{bmatrix} \vec{v}_s \\ 0 \\ \vec{\lambda}_s \\ \vec{\lambda}_r \end{bmatrix} = \begin{bmatrix} R_s & \frac{d}{dt} & 0 & 0 \\ 0 & 0 & R_r & \frac{d}{dt} \\ L_s & 0 & L_m & 0 \\ L_m & 0 & L_r & 0 \end{bmatrix} \begin{bmatrix} \vec{i}_s \\ \vec{\lambda}_s \\ \vec{i}_r \\ \vec{\lambda}_r \end{bmatrix} + j\omega_r \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \vec{\lambda}_r \\ 0 \\ 0 \end{bmatrix}. \quad (5.32)$$

A velocidade rotórica, ω_r , é obtida com orientação do fluxo rotórico, λ_r [143]. A estimação da velocidade rotórica é baseada nas Equações (5.33) e (5.34). O fluxo rotórico é utilizado para o cálculo da velocidade [7]. Das Equações (5.26), (5.27), (5.28) e (5.30) o fluxo rotórico pode ser escrito como:

$$\dot{\lambda}_{dr} = \frac{L_r}{L_m} (\vec{v}_s - R_s \vec{i}_s - \sigma L_s \dot{\vec{i}}_s) \quad (5.33)$$

ou

$$\dot{\lambda}_{ar} = \left(-\frac{1}{T_r} + \omega_r j\right) \vec{\lambda}_r + \frac{L_m}{T_r} \dot{\vec{i}}_s. \quad (5.34)$$

Sendo, $\dot{\lambda}_{dr} = \frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{ddr} \\ \vec{\lambda}_{dqr} \end{bmatrix}$ e $\dot{\lambda}_{ar} = \frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{adr} \\ \vec{\lambda}_{aqr} \end{bmatrix}$, $\dot{\lambda}_{dr}$ e $\dot{\lambda}_{ar}$ são expressões equivalentes.

A tensão induzida dada por $\dot{\lambda}_{ar}$ está em função de ω_r , enquanto $\dot{\lambda}_{dr}$ não possui essa dependência.

As Equações (5.33) e (5.34) do fluxo na referência dq são:

$$\frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{ddr} \\ \vec{\lambda}_{dqr} \end{bmatrix} = \frac{L_r}{L_m} \begin{bmatrix} \vec{v}_{ds} \\ \vec{v}_{qs} \end{bmatrix} - \begin{bmatrix} (R_s + \sigma L_s \frac{d}{dt}) & 0 \\ 0 & (R_s + \sigma L_s \frac{d}{dt}) \end{bmatrix} \begin{bmatrix} \vec{i}_{ds} \\ \vec{i}_{qs} \end{bmatrix} \quad (5.35)$$

e

$$\frac{d}{dt} \begin{bmatrix} \vec{\lambda}_{adr} \\ \vec{\lambda}_{aqr} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_r} & -\omega_r \\ \omega_r & -\frac{1}{T_r} \end{bmatrix} \begin{bmatrix} \vec{\lambda}_{adr} \\ \vec{\lambda}_{aqr} \end{bmatrix} + \frac{L_m}{T_r} \begin{bmatrix} \vec{i}_{ds} \\ \vec{i}_{qs} \end{bmatrix}. \quad (5.36)$$

A Equação (5.36) pode ser representada como:

$$\dot{\vec{\lambda}}_{adr} = -\frac{1}{T_r} \vec{\lambda}_{adr} - \omega_r \vec{\lambda}_{aqr} + \frac{L_m}{T_r} \vec{i}_{ds}, \quad (5.37)$$

$$\dot{\vec{\lambda}}_{aqr} = -\frac{1}{T_r} \vec{\lambda}_{aqr} + \omega_r \vec{\lambda}_{adr} + \frac{L_m}{T_r} \vec{i}_{qs}. \quad (5.38)$$

A saída do bloco que representa a máquina de indução é dada pelo fluxo desejado λ_d , que é obtido a partir da tensão induzida, Equação (5.33). Na saída do modelo neuronal tem-se o fluxo atual (λ_a), Equação (5.34). Os fluxos são independentes, expressos por meio de uma estrutura estacionária.

Aprendizagem

A aprendizagem da rede é realizada mediante o algoritmo de retropropagação estático e supervisionado, conforme descrito no item 3.1.3. Este tipo de algoritmo utiliza

pares de entrada/saída para, por meio de um mecanismo de correção de erros, ajustar os pesos da rede. Quando o fluxo atual, λ_a , se aproxima do fluxo desejado, λ_d , ou seja, o fluxo rotórico da Equação (5.34) se aproxima do fluxo da Equação (5.33), a *RNA* passa a fornecer os sinais atrasados no tempo para a sua própria entrada [10].

A *RNA* é utilizada como estimador para obter o fluxo do rotor. O erro entre o fluxo de referência, λ_d , e o fluxo estimado, λ_a é usado pelo algoritmo de aprendizagem para ajustar os pesos da rede. A *RNA* possui três pesos, sendo dois deles considerados constantes.

O ajuste do peso pela arquitetura da rede é realizado por meio da velocidade rotórica ω_r que, por sua vez, atualiza λ_a até que este se aproxime de λ_d . Conhecendo-se os parâmetros do motor, os dois modelos tendem a coincidir. Entretanto, diferenças entre a velocidade utilizada no modelo neuronal e na velocidade do motor podem resultar em erro entre a saída dos dois estimadores. Este erro entre λ_d e λ_a é responsável pela atualização do peso do modelo da rede neuronal, ou seja, a velocidade rotórica, ω_r , na Equação (5.34). Na Figura 5.11 tem-se esta aproximação, sendo que o algoritmo de retropropagação é derivado de forma tal que o estimador da Equação (5.34) acompanha o mais próximo possível o estimador da Equação (5.33). Para obter o algoritmo de retropropagação, o modelo dos padrões de dados da Equação (5.34) é, primeiramente, derivado.

Como a taxa de variação instantânea de λ_r em relação a T no instante em que

$t = Tk$, é dada por $\lim_{\Delta T \rightarrow 0} \frac{\Delta \lambda_r}{\Delta T} = \lim_{\Delta T \rightarrow 0} \left(\frac{\vec{\lambda}_r(k) - \vec{\lambda}_r(k-1)}{T} \right)$ então,

$$\dot{\vec{\lambda}}_a(k) = \frac{\vec{\lambda}_a(k) - \vec{\lambda}_a(k-1)}{T}. \quad (5.39)$$

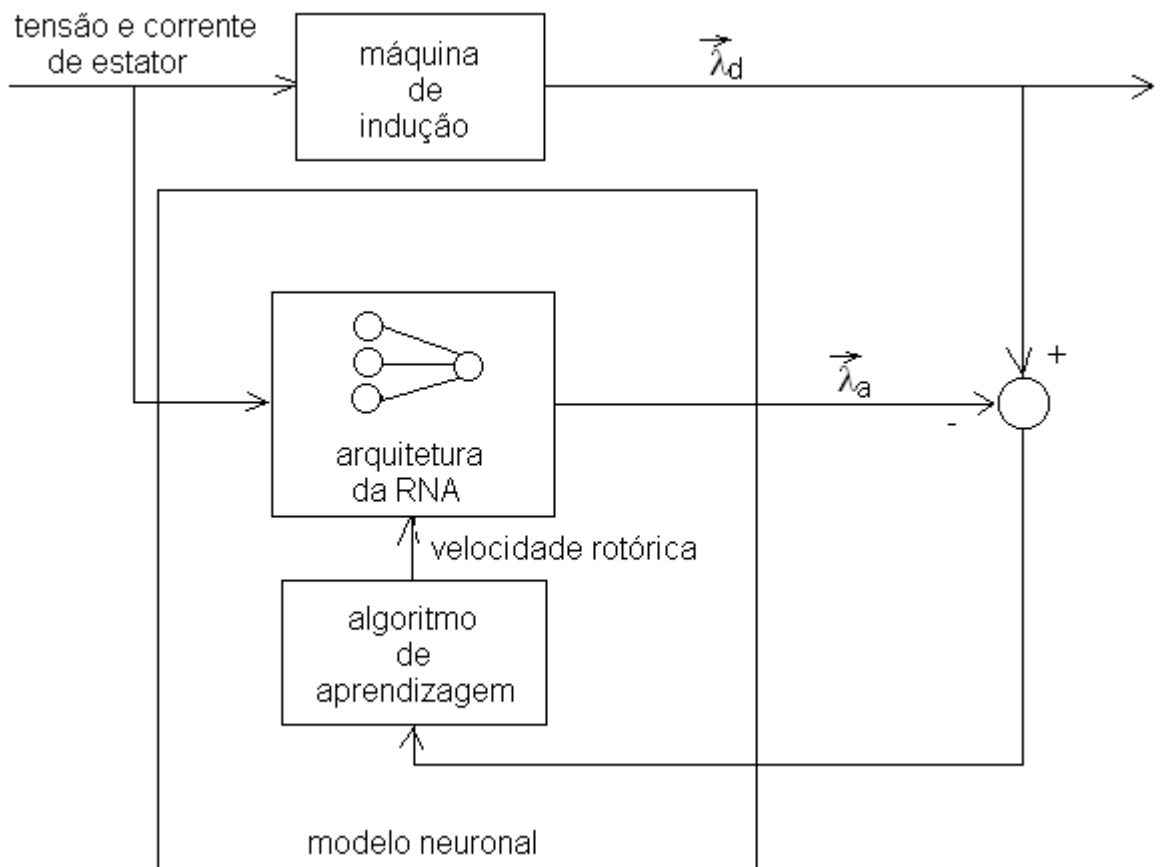


Figura 5.11: Estrutura da rede neuronal para estimar ω_r .

Aplicando a recursividade no lado direito da Equação (5.39), igualando à Equação

(5.34), vem

$$\frac{\vec{\lambda}_a(k) - \vec{\lambda}_a(k-1)}{T} = \left(\frac{-1}{T_r}I + \omega_r J\right)\vec{\lambda}_a(k-1) + \frac{L_m}{T_r}\vec{i}_s(k-1) \quad (5.40)$$

então,

$$\vec{\lambda}_a(k) = I\vec{\lambda}_a(k-1) + \left(\frac{-1}{T_r}I + \omega_r J\right)T\vec{\lambda}_a(k-1) + \frac{L_m}{T_r}T\vec{i}_s(k-1). \quad (5.41)$$

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

$$\vec{\lambda}_a(k) = \left(1 - \frac{T}{T_r}\right)I\vec{\lambda}_a(k-1) + \omega_r T J \vec{\lambda}_a(k-1) + \frac{L_m}{T_r}T\vec{i}_s(k-1). \quad (5.42)$$

A Equação (5.42) pode ser escrita da seguinte forma:

$$\vec{\lambda}_a = w_1\vec{x}_1 + w_2\vec{x}_2 + w_3\vec{x}_3 \quad (5.43)$$

ou ainda

$$\vec{\lambda}_a = \sum_{i=1}^3 w_i\vec{x}_i, \quad (5.44)$$

com $w_1 = 1 - \frac{T}{T_r}$, $\vec{x}_1 = I\vec{\lambda}_a(k-1)$, $w_2 = \omega_r T$, $\vec{x}_2 = J\vec{\lambda}_a(k-1)$, $w_3 = \frac{L_m}{T_r}$, $\vec{x}_3 = T\vec{i}_s(k-1)$,

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \text{ e } T \text{ é o período de amostragem.}$$

Arquitetura da Rede Neuronal Artificial

Há diversas notações do *MLP*, porém, para facilitar o desenvolvimento, adota-se uma nova notação tanto da arquitetura quanto da formulação algébrica do algoritmo de aprendizado. A notação adotada neste item é:

1. x_{idq} - camada de entrada, onde i varia de 1 a 3. Os índices dq corresponderam aos sinais no eixo direto e em quadratura;
2. λ_{adq} - camada de saída;
3. w_i - peso do neurônio da camada de entrada i para a camada de saída.

Na Figura 5.12 tem-se a representação da arquitetura da rede *MLP* composta por duas camadas; dois neurônios, ou seja, com duas saídas representando o fluxo magnetizante nas variáveis d e q . Os pesos da rede neuronal são w_1 , w_2 e w_3 ; x_1 , x_2 e x_3 representando as entradas e a função de ativação linear é utilizada.

Segundo Braga [88], o número de unidades das camadas intermediárias de uma rede pode crescer exponencialmente com o número de entradas, porém a solução neuronal mais eficiente é aquela em que o número de unidades cresce apenas polinomialmente com o aumento do número de unidades de entradas. Neste trabalho, a solução do estimador neuronal de velocidade rotórica é constituída por uma camada de entrada e uma camada de saída para cada um dos dois neurônios.

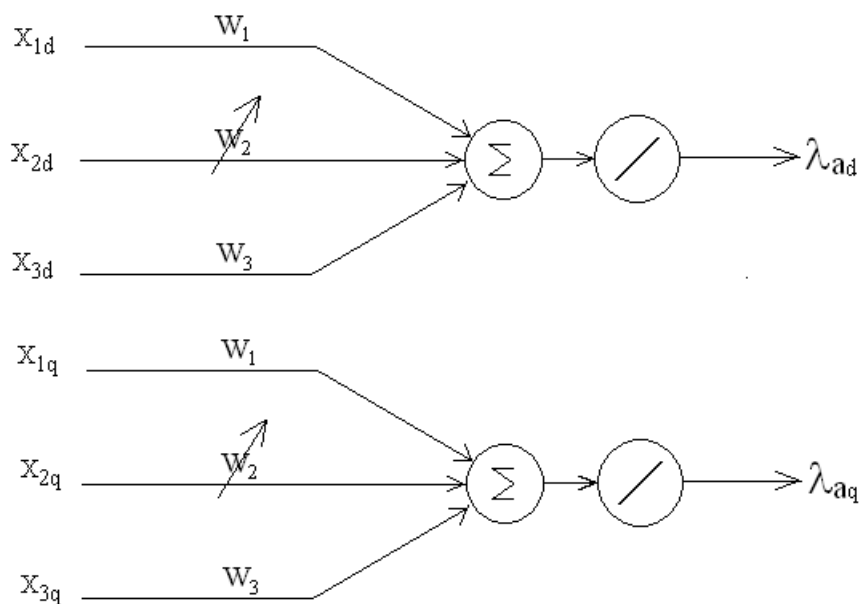


Figura 5.12: Dois neurônios com três entradas e duas saídas lineares.

Organizando λ_a , x e w , tem-se que,

$$\vec{\lambda}_a = \begin{bmatrix} \lambda_{ad} \\ \lambda_{aq} \end{bmatrix},$$

$$\vec{x} = \begin{bmatrix} x_{1d} & x_{2d} & x_{3d} \\ x_{1q} & x_{2q} & x_{3q} \end{bmatrix} \quad (5.45)$$

e

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}. \quad (5.46)$$

O erro entre o fluxo desejado e o fluxo atual, matematicamente, é dado por:

$$\vec{\varepsilon}(k) = \vec{\lambda}_d(k) - \vec{\lambda}_a(k). \quad (5.47)$$

Os pesos w_1 e w_3 da rede neuronal são considerados constantes, pois dependem somente dos parâmetros do motor. O peso w_2 que depende da velocidade do motor é variável [7]. A velocidade estimada é dada por [143], [144],

$$\hat{\omega}_r(k) = \hat{\omega}_r(k-1) - \frac{1}{T}\eta\delta(k)X_2 + \frac{1}{T}\alpha\Delta w_2(k-1). \quad (5.48)$$

O desenvolvimento da velocidade estimada pode ser verificado no Apêndice B.

Combinando a Equação (5.43) e a Equação (B.25) que é desenvolvida no Apêndice B, segue

$$(\Delta w_k, y_a(k)) = F(w_k, x_k, y_d(k)), \quad (5.49)$$

sendo que F utiliza os pesos w_k , as entradas x_k e o valor desejado y_d como entradas e Δw_k e $y_a(k)$ como saídas. Na Figura 5.13, tem-se a ilustração do fluxograma do algoritmo de retropropagação modificado para treinar a *RNA*. A principal diferença

entre este algoritmo e o algoritmo de retropropagação apresentado no item 3.1.3 é que a saída da *RNA* não fornece o valor da grandeza de interesse. O valor realmente desejado é obtido por meio da variação do peso w_2 , do qual é extraído o valor da velocidade rotórica da máquina de indução, conforme é verificado na Equação (B.27) do Apêndice B e reforçado na Figura 5.14.

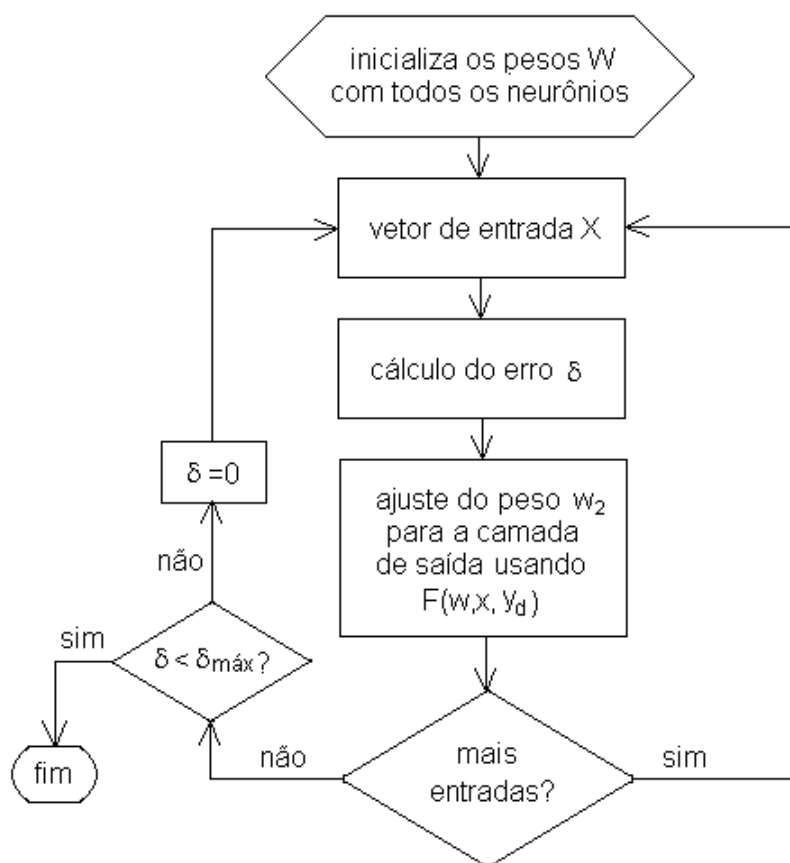


Figura 5.13: Algoritmo de retropropagação modificado para treinar a *RNA*.

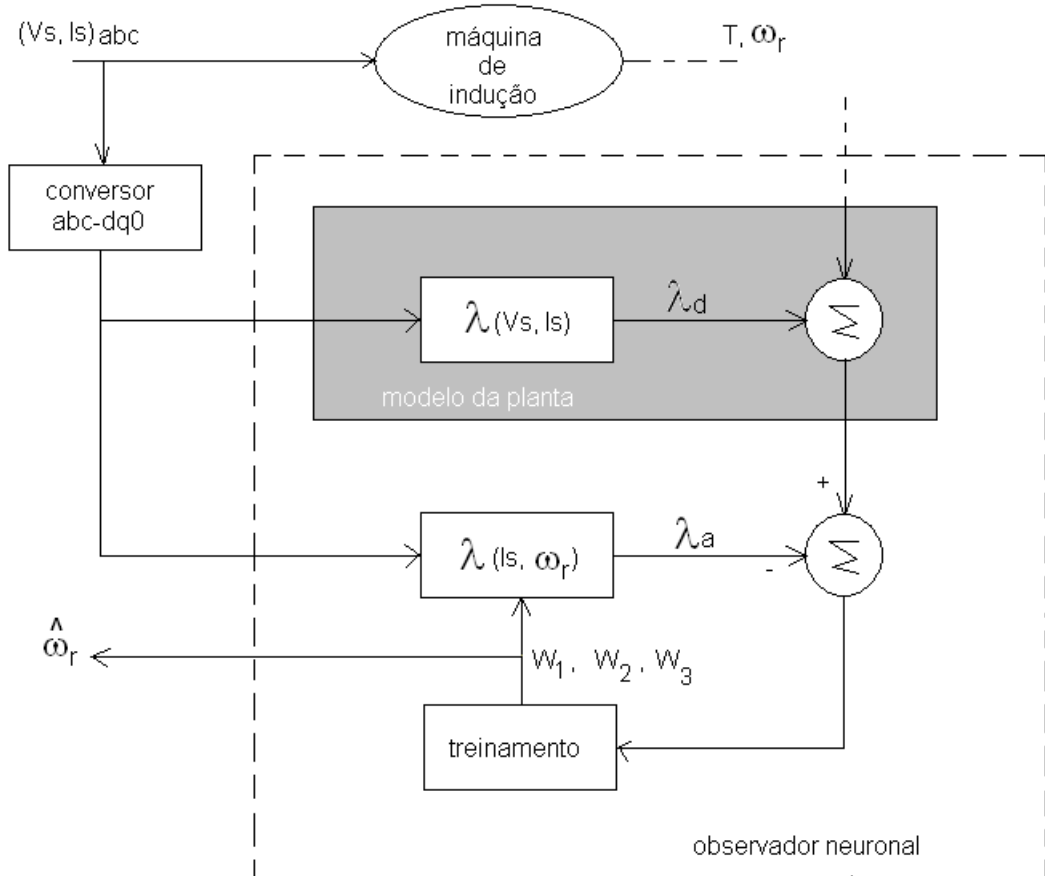


Figura 5.14: Núcleos do observador neuronal de fluxo/velocidade rotórica.

Sistema Experimental

O PSoCTM, *Programmable System on Chip*[145], é usado neste projeto para implementar o estimador neuronal. Considera-se a capacidade de memória disponível na arquitetura. Os dados utilizados na simulação são obtidos do *catálogo do fabricante* e dados de placa da máquina de indução. O desenvolvimento do algoritmo é codificado

tanto em linguagem *C* própria da arquitetura PSoCTM quanto em *C Builder* versão

4.5. As características principais da máquina de indução são:

- **Motor de Indução:** trifásico, tensão: 380 V em Y, velocidade: 1700 rpm, potência: 2,25 kW e fator de potência: 0,82;
- **Parâmetros:** $L_s=134,5$ mH, $L_r=76,55$ mH, $R_r=1,55$ Ω e $R_s=2,1$ Ω , obtidos experimentalmente.

A tecnologia da arquitetura programável desenvolvida pela *Cypress System* e o *software PSoC Designer versão 4.0* são utilizados. A partir do *módulo LCD* do *software* os valores de saída da velocidade estimada são fornecidos e disponíveis na *porta 0* do *chip*. O valor da velocidade estimada é visualizada mediante a conexão com um *display* externo. Dados da arquitetura:

- **Arquitetura Programável:** velocidade do processador MC8: 24 MHz, Tensão de operação: 4,95 V, memória *flash* usada: 8 Koctetos, *módulo LCD* e *módulo TIMER*.

Análises

O programa desenvolvido em linguagem *C* é previamente processado para posterior utilização do código *C* da arquitetura. Verifica-se a influência do tamanho das variáveis, principalmente devido às dimensões das matrizes utilizadas. Algumas estratégias são

utilizadas para acelerar o método de modo que a execução do algoritmo seja favorável, permitindo que a matriz aparente ter um menor número de pontos e/ou tenha número de condição menor.

O algoritmo é desenvolvido contendo 143 linhas de código *C*. Um conjunto de amostras de tensão e corrente estatóricas na referência *dq0* é armazenado em uma tabela e considerado, inicialmente, constante, utilizando desse modo, a memória *flash* da arquitetura. Cada amostra em ponto flutuante ocupa 4 octetos e as variáveis inteiras 2 octetos. A arquitetura é de 8 *bits*, porém utiliza-se palavras de 16 *bits*. Os módulos *LCD* e *TIMER* são utilizados para visualização do valor da velocidade rotórica e verificação do tempo de execução, respectivamente. A porta 0 é utilizada como saída do valor da velocidade rotórica medida indiretamente. O arquivo principal, contendo o código em *C* possui tamanho de 8 Koctetos.

Uma análise comparativa entre as velocidades obtidas por meio de simulação e pelo estimador neuronal evidencia uma boa estimação em regime permanente. O bom desempenho do estimador no espaço de estados é verificado quando comparado aos valores do tacômetro e com o sistema embarcado em regime. Neste caso, no tacômetro com 1790,00 rpm, considerando o motor sem carga. A velocidade obtida por simulação é 1798,80 rpm e a velocidade obtida pelo estimador neuronal implementada na arquitetura é 1791,94 rpm.

5.5.2 Medição Indireta Robusta de Velocidade

Devido à temperatura, saturação e outros efeitos não-lineares, existem variações nos parâmetros da máquina de indução. Diversos métodos de estimação de parâmetros em máquinas de indução são utilizados, conforme pode ser verificado em [4], [9], [12], [16], [146]. A variação na resistência do rotor pode ser de até 50%, causando variação na constante de tempo do rotor que, por sua vez, influencia no controle de fluxo orientado [147].

Neste item tem-se o desenvolvimento de uma *RNA-robusta* para medir indiretamente a velocidade rotórica em uma máquina de indução. O método de medição indireta utilizado é o da *RNA* e os procedimentos são semelhantes aos do item 5.5.1. Na aprendizagem robusta, todos os parâmetros devem aprender com mesma velocidade.

Modelo da Máquina de Indução

Considerando-se o modelo da máquina apresentado no item 5.5.1 e de modo semelhante a [148]-[150], para verificar a robustez da *RNA*, considerou-se que os parâmetros reais da máquina de indução possuíam as seguintes incertezas em porcentagem: $\frac{\Delta R_s}{R_s} = c$, $\frac{\Delta R_r}{R_r} = d$, $\frac{\Delta L_r}{L_r} = e$, $\frac{\Delta L_s}{L_s} = f$, $\frac{\Delta L_m}{L_m} = g$ e $\frac{\Delta \sigma}{\sigma} = h$. As Equações (5.26), (5.27), (5.28) e

(5.30) na forma matricial são:

$$\begin{bmatrix} \vec{v}_s \\ 0 \\ \vec{\lambda}_s \\ \vec{\lambda}_r \end{bmatrix} = \begin{bmatrix} (R_s + \Delta R_s) & \frac{d}{dt} & 0 & 0 \\ 0 & 0 & (R_r + \Delta R_r) & \frac{d}{dt} \\ (L_s + \Delta L_s) & 0 & (L_m + \Delta L_m) & 0 \\ (L_m + \Delta L_m) & 0 & (L_r + \Delta L_r) & 0 \end{bmatrix} \begin{bmatrix} \vec{i}_s \\ \vec{\lambda}_s \\ \vec{i}_r \\ \vec{\lambda}_r \end{bmatrix} + j\omega_r \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \vec{\lambda}_r. \quad (5.50)$$

A velocidade rotórica, ω_r , baseia-se na estimação de ω_r a partir das equações (5.33) e (5.34). Das equações (5.26), (5.27), (5.28) e (5.30) o fluxo rotórico, considerando as variações paramétricas, pode ser escrito como:

$$\dot{\lambda}_{dr} = \frac{(L_r + \Delta L_r)}{(L_m + \Delta L_m)} (\vec{v}_s - (R_s + \Delta R_s) \vec{i}_s - (\sigma + \Delta\sigma)(L_s + \Delta L_s) \dot{\vec{i}}_s) \quad (5.51)$$

ou

$$\dot{\lambda}_{ar} = \left(-\frac{1}{(T_r + \Delta T_r)} + \omega_r j \right) \vec{\lambda}_r + \frac{(L_m + \Delta L_m)}{(T_r + \Delta T_r)} \dot{\vec{i}}_s. \quad (5.52)$$

Desenvolvendo-se as Equações (5.51) e (5.52), tem-se

$$\dot{\lambda}_{dr} = \frac{L_r}{L_m} (A \vec{v}_s - B R_s \vec{i}_s - C \sigma L_s \dot{\vec{i}}_s), \quad (5.53)$$

$$\dot{\lambda}_{ar} = \left(-\frac{1}{T_r D} + \omega_r j \right) \vec{\lambda}_a + \frac{L_m}{T_r} F \dot{\vec{i}}_s, \quad (5.54)$$

sendo $A = \frac{(1+e)}{(1+g)}$, $B = A(1+c)$, $C = A(1+f)(1+h)$, $D = (1 + \frac{e}{d})$ e $F = \frac{(1+g)}{D}$.

A saída do bloco que representa a máquina de indução é dada pelo fluxo desejado λ_d , que é obtido a partir da tensão induzida, Equação (5.53). Na saída do modelo neuronal tem-se o fluxo real (λ_a), Equação (5.54).

Aprendizagem

A aprendizagem da rede é também realizada mediante o algoritmo de retropropagação estático e supervisionado. Quando o fluxo real, λ_a , se aproxima do fluxo desejado, λ_d , ou seja, o fluxo rotórico da Equação (5.54) se aproxima do fluxo da Equação (5.53), o *Observador-Neuronal* passa a fornecer os sinais atrasados no tempo para a sua própria entrada, conforme Equação (5.55), [10], [150], [151],

$$y^{maq}(t+1) = f(y^{maq}(t), \dots, y^{maq}(t-n+1); u(t), \dots, u(t-m+1)). \quad (5.55)$$

A saída do sistema, representado pela máquina de indução, y^{maq} , no instante $t+1$ depende dos n valores passados da saída do sistema e dos m valores passados da entrada u do sistema, sendo $(u(t), y^{maq}(t))$ os pares entrada/saída.

A *RNA* robusta é utilizada como estimador para obter o fluxo do rotor. O erro entre o fluxo de referência, λ_d , e o fluxo estimado, λ_a é utilizado pelo algoritmo de aprendizagem para ajustar os pesos da rede. A rede contém três pesos, sendo dois deles considerados constantes.

O algoritmo de retropropagação é derivado de forma tal que o estimador da Equação

(5.54) acompanha o mais próximo possível o estimador da Equação (5.53). Para obter o algoritmo de retropropagação, o modelo dos padrões de dados da Equação (5.54) é, primeiramente, derivado e, aplicando-se a recursividade no lado direito da Equação (5.39), e igualando à Equação (5.54), tem-se,

$$\frac{\vec{\lambda}_a(k) - \vec{\lambda}_a(k-1)}{T} = \left(\frac{-1}{T_r D} I + \omega_r J\right) \vec{\lambda}_a(k-1) + F \frac{L_m}{T_r} \vec{i}_s(k-1) \quad (5.56)$$

então,

$$\vec{\lambda}_a(k) = I \vec{\lambda}_a(k-1) + \left(\frac{-1}{T_r D} I + \omega_r J\right) T \vec{\lambda}_a(k-1) + F \frac{L_m}{T_r} T \vec{i}_s(k-1). \quad (5.57)$$

Organizando os termos em relação às matrizes I , J e as variáveis $\vec{\lambda}_a$ e \vec{i}_s vem:

$$\vec{\lambda}_a(k) = \left(1 - \frac{T}{T_r D}\right) I \vec{\lambda}_a(k-1) + \omega_r T J \vec{\lambda}_a(k-1) + F \frac{L_m}{T_r} T \vec{i}_s(k-1). \quad (5.58)$$

A Equação (5.58) pode ser escrita da seguinte forma:

$$\vec{\lambda}_a = w_1 \vec{x}_1 + w_2 \vec{x}_2 + w_3 \vec{x}_3 \quad (5.59)$$

ou ainda

$$\vec{\lambda}_a = \sum_{i=1}^3 W_i \vec{X}_i, \quad (5.60)$$

com $w_1 = 1 - \frac{T}{T_r D}$, $\vec{x}_1 = I\vec{\lambda}_a(k-1)$, $w_2 = \omega_r T$, $\vec{x}_2 = J\vec{\lambda}_a(k-1)$, $w_3 = F\frac{L_m}{T_r T}$,
 $\vec{x}_3 = I\vec{i}_s(k-1)$, $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ e T é o período de amostragem.

Arquitetura da RNA Robusta

Semelhante ao item 5.5.1, a representação da arquitetura do *MLP* é composta por duas camadas; dois neurônios, ou seja, com duas saídas representando o fluxo magnetizante nas variáveis d e q . Os pesos da rede neuronal são w_1 , w_2 e w_3 ; x_1 , x_2 e x_3 representando as entradas e a função de ativação linear é utilizada.

Após as substituições das equações, verifica-se que a velocidade estimada que é obtida a partir do peso w_2 permanece a mesma que é desenvolvida no Apêndice B, Equação (B.27), entretanto os pesos w_1 e w_3 se alteram quando se aplica a robustez.

5.6 Conclusões

Nos quatro estudos de casos apresentados, verificou-se que a metodologia utilizada no desenvolvimento dos *SMI* é satisfatória. Os resultados e análises realizados em cada um dos casos demonstram que o uso de *SMI* é viável e pode ser adotado em sistemas que apresentem razões para o uso da medição indireta. Os algoritmos das ferramentas matemáticas escolhidas apresentam melhor desempenho quando são desenvolvidos de

tal forma que os parâmetros relevantes, todos os parâmetros relevantes e somente os parâmetros relevantes estejam inseridos. O conhecimento da importância dos parâmetros utilizados para o bom desempenho do método é obtido a partir do momento em que se faz uma correlação entre as variáveis de entrada e as variáveis de saída. Esses aspectos são importantes para que o resultado de uma medição indireta seja mais próximo do desejado e que o processamento do algoritmo apresente uma resposta mais rápida.

No Estudo de Caso 5.1, o *FK* é utilizado para medir indiretamente a velocidade de um veículo aeroespacial. Verifica-se por meio dos gráficos a proximidade do valor estimado com o valor de referência da velocidade e da posição; a velocidade de reação do filtro e a convergência do filtro durante a mudança da fase propulsada para a fase balística; o nível do ruído do sinal filtrado. Este nível de ruído é visualizado no gráfico da posição, no qual é considerada somente a coordenada Z . Verifica-se também, por meio dos gráficos, mudanças abruptas do *FK* no início do processo, entretanto, o filtro consegue recuperar os valores medidos. No gráfico da aceleração, entretanto, o filtro não apresenta bons resultados, os resultados são inexatos; porém, esta componente não tem utilidade no desempenho da dinâmica do rastreamento. Esses resultados são realizados para uma fase inicial de estudo, na qual o objetivo é localizar o melhor ajuste para a sintonia do filtro. O *FK* apresenta bons resultados para filtrar posição e medir indiretamente a velocidade, é regular no tempo de reação e inexato para se obter a aceleração.

Dessa forma, com base nos resultados computacionais e implementações em *hardware*, pode-se mencionar que o método *FK* apresenta bom desempenho e satisfaz os requisitos

necessários para implementações em tempo real.

O método do *FK* também é utilizado no Estudo de Caso 5.2. Nesta aplicação, o *FK* é utilizado para medir indiretamente a temperatura no interior de um objeto que se encontra dentro de uma estufa. O modelo térmico é desenvolvido para o sistema e representado por meio de equações diferenciais. Novamente, o desempenho do algoritmo proposto pode ser verificado por método computacional e implementação em *hardware*. O algoritmo é também implementado em uma arquitetura reconfigurável de baixo custo que possui blocos de *conversores A/D*. O *conversor A/D* recebe os valores de temperatura medidos pelo sensor colocado no meio 2. A partir desses valores, o algoritmo é executado, visualizando-se no *display* o valor da temperatura medida indiretamente no meio 1, ou seja, o valor da temperatura no interior do objeto. Neste Estudo de Caso, baseando-se na análise do modelo, leva-se em consideração a massa, o calor específico e a condutância térmica tanto do objeto quanto da estufa. Não é realizada uma análise quanto ao volume e/ou a forma geométrica da estufa e do objeto, entretanto, por meio das medições diretas utilizando dois sensores é possível comprovar as variações no interior do objeto em relação à temperatura interna da estufa. O dispositivo de medição indireta proposto pode ser considerado como uma alternativa de estimativa ótima de temperatura para monitoração e controle.

Já nos Estudos de Casos 5.5.1 e 5.5.2, o método de medição indireta utilizado é a *RNA*. O modelo do sistema descrito no item 5.5.1 é desenvolvido baseado no modelo

neuronal e implementado em uma arquitetura. Os resultados computacionais e experimentais provam que o uso do estimador neuronal é viável para medição indireta da velocidade. No caso robusto, os procedimentos para o desenvolvimento são semelhantes aos utilizados no item 5.5.1, adiciona-se aqui a robustez da máquina. Verifica-se ao fim do desenvolvimento matemático que a equação do peso w_2 abordada no Apêndice B, Equação (B.27), ou seja, da velocidade rotórica é a mesma para os dois casos de estudo, entretanto, os termos dos pesos w_1 e w_3 são alterados devido ao acréscimo da robustez. A eficiência do método pode ser verificada a partir da análise comparativa entre os valores obtidos diretamente por meio do tacômetro e os valores medidos indiretamente por meio do observador neuronal.

Uma análise comparativa é realizada em [144] por meio do *Matlab/Simulink* entre os seguintes estimadores de velocidade rotórica: estimador via fluxo; estimador via *fem*; estimador via fluxo usando *MRAS* e estimador via *fem* usando *MRAS*. Em todas as técnicas analisadas ocorreram oscilações no início do tempo das formas de onda. As velocidades estimadas pelo fluxo e pela *fem* apresentaram mais oscilações até se aproximarem do valor de referência. Na estimação do fluxo utilizando *MRAS*, a velocidade estimada aproximou-se mais rápido do valor de referência. A estimação da *fem* utilizando *MRAS* distanciou-se um pouco da velocidade de referência, ficando 25% abaixo desta. Na simulação realizada para o dispositivo de medição indireta, no observador neuronal, inicialmente são atribuídos diversos valores para os parâmetros da rede neuronal, porém somente os valores $\alpha = 0,6$ e $\eta = 0,5$ foram mais adequados

ao algoritmo de treinamento, aproximando-se do valor de referência. Nesse trabalho, o autor comenta que a velocidade estimada ω_r se apresenta melhor para a técnica *MRAS* com fluxo, ou seja, converge mais rapidamente para o valor de referência em seguida o estimador via redes neurais atingindo o valor de referência um pouco depois do estimador via fluxo por *MRAS*. A técnica via *fem* por *MRAS*, embora tenha se estabilizado rapidamente em um valor de velocidade, distanciou do valor de referência. Logo depois, as técnicas via fluxo e via *fem* sem *MRAS*, ambas não garantiram que a velocidade estimada convergisse para o valor de referência.

Capítulo 6

Conclusão

A principal contribuição desta tese é o desenvolvimento de metodologias para projeto e implementação de *SMI*, utilizando estimações estocásticas e neuronais de estados. Mostra-se a viabilidade do uso de ferramentas matemáticas para a realização de medições indiretas ótima estocásticas e neuronais em sistemas dinâmicos. O desenvolvimento de cada um dos *SMI* apresentados nos estudo de casos é abordado nas seguintes etapas: desenvolvimento do modelo do sistema dinâmico; método de medição indireta considerado mais adequado a aplicação; utilização de um *software* para verificar antecipadamente o comportamento do sistema; integração entre o *software* e o *hardware*. Destaca-se também o desenvolvimento de uma proposta de um *SMI* robusto para máquinas de indução via *RNA*. Com o uso de *SMI* é possível reduzir a necessidade de se realizar medições por meio de dispositivos eletromecânicos ou eletrônicos quando for considerado necessário.

Normalmente, os dispositivos utilizados para realizar medições diretas são limitados,

principalmente por causa da disponibilidade tecnológica para o seu desenvolvimento. Essa lacuna existente no sistema de medição direta pode ser preenchida com o auxílio das medições indiretas. O uso destas medições também pode reduzir custos em sistemas de medição. O tratamento dos dados é realizado por meio de um algoritmo especificado conforme o método utilizado para se realizar a medição. A aquisição desses dados pode ser realizada por meio de medições diretas, por parâmetros do próprio sistema em análise ou por meio de sinais armazenados em um banco de dados. Todavia, o desenvolvimento de tais sistemas requer a integração de diversos conhecimentos em Engenharia, Eletrônica, Informática, Física e outras áreas que se fizerem necessárias de acordo com o sistema no qual se deseja realizar a medição indireta.

Os procedimentos de controle, medições e automação de processos tradicionalmente utilizam sensores reais. Por outro lado, existem situações nas quais se tem dificuldades para realizar uma medição direta por questões de segurança, tempo, acesso, inexistência de sensor ou transdutor. Recomenda-se como alternativa para esses casos, a medição indireta. As medições indiretas podem ser, também, utilizadas quando se deseja obter resultados mais confiáveis do que aqueles obtidos por métodos convencionais ou ainda para reduzir um conjunto de procedimentos necessários para se conhecer o valor de uma grandeza.

Como o processo de medição de grandezas é fundamental para as diversas áreas, torna-se extremamente necessário o uso de recursos que auxiliem e melhorem significativamente as medições. Em muitas situações, o estado pode não estar disponível

para medida, o custo de sensores pode ser demasiadamente elevado. Dessa forma, se beneficiando das medições indiretas é possível ter conhecimento de algumas grandezas. Um *SMI* utilizando o método de medição indireta por *FK*, por exemplo, pode ser empregado para se obter informações em comunicações, navegação, para realimentação em controle e outras aplicações nas quais ele pode ser utilizado com exatidão.

A premissa básica de qualquer medição é de que as informações a serem manipuladas sejam corretas, pois os erros inerentes à medição direta e o uso inadequado de parâmetros do sistema contribuem para que o resultado da medição indireta se distancie do valor desejado. Dessa forma, as etapas que antecedem o resultado de uma medição indireta devem ser realizadas com conhecimento, experiência, habilidade, perspicácia, cautela e bom-senso.

O uso de um *software* é necessário como uma das etapas iniciais para a verificação do modelo do sistema em estudo. Para as análises dos gráficos, é utilizado o *MATLAB*[®] como plataforma computacional. Por meio desse *software* é verificado o comportamento das grandezas relevantes de cada um dos sistemas estudados. A partir dessas análises é possível desenvolver métodos de ajuste de alguns parâmetros do método de medição indireta empregado de modo que haja convergência para os resultados desejados.

Os algoritmos apresentam melhor desempenho quando são desenvolvidos de tal forma que somente os parâmetros relevantes no processo da medição indireta estejam inseridos. O método de medição indireta *FK* é utilizado para medir indiretamente a velocidade de um veículo aeroespacial e medir a temperatura no interior de um objeto

que estava dentro de uma estufa. Por meio deste método de medição obtém-se bons resultados para filtrar posição e medir indiretamente a velocidade do veículo aeroespacial e também a temperatura no interior do objeto. Dessa forma, com base nos resultados computacionais e experimentais, pode-se mencionar que o uso do método *FK* apresenta-se bastante promissor desde o projeto até a fase de implementação em *hardware*. Um outro método de medição que se mostra bastante promissor é o uso das *RNA*. Os resultados computacionais e experimentais provam que o uso do estimador neuronal é viável para medição indireta da velocidade rotórica em máquina de indução. A eficiência do método pode ser verificada a partir da análise comparativa entre os valores obtidos diretamente por meio do tacômetro e os valores medidos indiretamente por meio do observador neuronal.

O desenvolvimento dos *SMI* dos estudos de casos neste trabalho surgiram devido à necessidade do emprego de uma metodologia alternativa, tentando ultrapassar as concepções predominantes no processo da medição. Não é possível nem desejável negar a importância das medições diretas no processo da medição indireta, tendo em vista que estas são dependentes daquelas.

Capítulo 7

Sugestões para Trabalhos Futuros

Como prosseguimento ou desdobramento dos temas expostos e nos estudo de casos apresentados neste trabalho de tese, seguem-se algumas sugestões de linhas de pesquisas:

1. aplicações para controle sob o ponto de vista de observador de estado. Por exemplo, tornar factível a implementação do projeto *Regulador Linear Quadrático* (LQR), de modo a estimar os estados que não podem ser mensurados diretamente;
2. a aplicação da metodologia proposta para sistemas complexos do mundo real e dispositivos em que ela reduz o custo e o espaço físico;
3. desenvolvimento de um método para otimizar a implementação em função do custo, energia, tamanho e confiabilidade;
4. análise de métodos de medição indireta para sinais em baixas e altas frequências;
5. estudo da viabilidade de combinar métodos de medição indireta para fusão de dados.

Referências Bibliográficas

- 1 SCHOUKENS, J.; PINTELON, R. *Identification of Linear Systems*. 1.ed., Pergamon Press, 1991.
- 2 MOUZINHO, Lucilene F.; FONSECA NETO, João V.; LUCIANO, Benedito A.; FREIRE, Raimundo C.S.. Induction machine neural estimator using embedded system. *IEEE Instrumentation and Measurement Technology Conference*, Ottawa, Canada, p. 1291-1296, May 2005.
- 3 MOUZINHO, Lucilene F.. *Implementação do estimador de velocidade da máquina de indução via redes neuronais utilizando uma arquitetura programável*. 2004. 42f. Projeto e Pesquisa–Universidade Federal de Campina Grande, Campina Grande, Paraíba.
- 4 SOUZA FILHO, Eurico Bezerra de. Estimação dos parâmetros do modelo do motor de indução usando o algoritmo talus. *Exacta*, v. 2, p. 23-26, nov. 2004.

- 5 FONSECA NETO, João V.; MOUZINHO, Lucilene F.. Analysis of speed estimation techniques for asynchronous machines. *The 14th IASTED International Conference on Modelling and Simulation MS*. California, Feb. 2003.
- 6 BEN-BRAHIM, Lazhar. Motor speed identification via neural network. *IEEE Industry Applications Magazine*, v. 1, p. 28-32, Jan/Feb. 1995.
- 7 ELLOUMI, Mohsen; BEN-BRAHIM, Lazhar; ALHAMADI, Mohamed Survey of speed sensorless controls for IM drives. *Industrial Electronics Society, IECON. Proceedings of the 24th Annual Conference of the IEEE*, v. 2, p. 1018-1023, Aachen, Germany, Sep. 1998.
- 8 BEN-BRAHIM, Lazhar; TADAKUMA, Susumu. Pratical considerations for sensorless induction motor drive system. *Industrial Electronics Society, IECON. Proceedings of the 24th Annual Conference of the IEEE*, v. 2, p. 1002-1007, Aachen, Germany, Sep. 1998.
- 9 STEPHAN, Jennifer; BODSON, Marc; CHIASSON, John. Real-time estimation of the parameters and fluxes of induction motors. *IEEE Transactions on Industry Applications*, v. 3, p. 746-759, May/June 1994.
- 10 IRWIN, George W.; WARWICK, Kevin; HUNT, Ken J.. *Neural Network Applications In Control*. The Institution of Electrical Engineers, 1. ed., Oct. 1995.
- 11 KIM, Nam-Joon; HYUN, Dong-Seok. Very low speed control of induction machine by instantaneous speed and inertia estimation. *Industrial Electronics Control and Instrumentation - IECON*, v. 1, p. 605-610, 1994.

- 12 OH, D.S.; CHO, K.Y.; YOUN, M.J.. New rotor parameter estimation for a flux and speed control of induction machine considering saturation effects. *Industrial Electronics Control and Instrumentation - IECON*, v. 1, p. 561-566, Oct./Nov.1991.
- 13 GRZESIAK, L.M.; SKOCZYLAS, J.. Ac machine torque and stator flux estimation using a neural network based on the steady-state 2d field model. *International Symposium on Industrial Electronics-ISIE, Proceedings of the IEEE*, v. 1, p. 358-361, Marsaw, Poland, June 1996.
- 14 D'ANGELO, Marcos F.S.V.; COSTA JR., P.P.. State estimation for induction machines using a neural network backpropagation technique. *IEEE International Conference on Systems, Man, and, Cybernetics*, v. 4, p. 2613-2618, Nashville, TN, USA, 2000.
- 15 LUETZIG, Guenter; SÀNCHEZ-CASTILLO, Manuel; LANGARI, Reza. On tool wear estimation through neural networks. *IEEE International Conference on Neural Networks*, p. 2359-2363, Houston, TX, USA, June 1997.
- 16 XIONG, Shen-Shu; ZHOU, Zhao-Ying. Dynamic parameter estimation of velocity sensors using an indirect measurement approach. *IEEE Instrumentation and Measurement*, p. 794-797, May 2001.
- 17 RAOL, J.R.; MADHURANATH. Neural networks architectures for parameter estimation of dynamical systems. *IEEE transactions on control theory applications*, 143, 1996.

- 18 ARMSTRONG, Gavin J.; ATKINSON, David J.; ACARNLEY, Paul P.. A comparison of estimation techniques for sensorless vector controlled induction motor drives. *IEEE International Conference on Power Electronics and Drive Systems*, v. 1, p. 110-116, May 1997.
- 19 ERENTURK, Koksai; ERENTURK, Saliha; TABIL, Lope G.. A comparative study for the estimation of dynamical drying behavior of *echinacea angustifolia*: regression analysis and neural network. *Computers and Electronics in Agriculture*, v. 45, p. 71-90, June 2004.
- 20 MOUZINHO, Lucilene F.; FONSECA NETO, João V.; LUCIANO, Benedito A.; FREIRE, Raimundo C.S.. Kalman filter in real time for indirect measurement of space vehicle position using a reconfigurable architecture. *IEEE Instrumentation and Measurement Technology Conference*, p. 853-858, Ottawa, Canada, May 2005.
- 21 MIAO, Jianwei. A computer aided method for design of noninvasive indirect measurement of arterial blood pressure. *Southeastcon, Proceedings, IEEE*, p. 772-775, Birmingham, AL, USA, April 1992.
- 22 WANG, Jia-Jung *et al.*. Model-based synthetic fuzzy logic controller for indirect blood pressure measurement. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, v. 32, p. 306-315, June 2002.
- 23 PAPP, Z.; DRIESSEN, B.J.F.. An efficient adaptive fuzzy control scheme. *IEEE Instrumentation and Measurement Technology Conference*, p. 741-746, Waltham, MA, USA, April 1995.

- 24 HOLEJSOVSKA, Pavla; PEROUTKA, Zdenik; EENGERY, Jioi. Non-invasive monitoring of the human blood pressure. *Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems (CBMS-03)*, p. 301-306, June 2003.
- 25 URSINO, Mauro; CRISTALLI, Cristina. Mathematical analysis of the oscillometric technique for indirect blood pressure evaluation. *Engineering in Medicine and Biology Society. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE*, v. 2, p. 1286-1287, Baltimore, MD, USA, 1994.
- 26 MOUDDEN, Yassir; BOUBAL, Olivier. Indirect measurement of cylinder pressure peak position in an internal combustion engine. *IEEE Instrumentation and Measurement*, p. 21-23, May 2001.
- 27 ARAGÃO, Antônio; ROMERO, Roseli; MARQUES, Eduardo Marques. Computação reconfigurável aplicada à robótica. São Carlos-SP *Universidade de São Paulo-USP, Instituto de Ciências Matemáticas e de Computação - ICM*. Disponível em: [http : // www. unipan. br / fernando / OAC / pesquisa](http://www.unipan.br/fernando/OAC/pesquisa). Acesso em 2005.
- 28 BUCCI, G. et al.. Real-time transputer-based measurement apparatus: Performance testing. *IEEE Transactions On Instrumentation And Measurement*, v. 43, p. 251-256, April 1994.
- 29 UZUN, I.S.; AMIRA, A.; BOURIDANE, A.. FPGA implementations of Fast Fourier Transforms for real-time signal and image processing. *IEE proceedings. Vision, image and signal processing*, v. 152, p. 283-296, June 2005.

- 30 BONATO, Vanderlei. *Projeto de um módulo de aquisição e pré-processamento de imagem colorida baseado em computação reconfigurável e aplicado a robôs móveis*. 2004. Dissertação de Mestrado Universidade de São Paulo - USP, São Carlos.
- 31 INMETRO. *Vocabulário Internacional de Termos Fundamentais e Gerais de Metrologia - VIM*, 3. ed., Rio de Janeiro, 2003.
- 32 COOK, Nathan H.; RABINOWICZ, Ernest. *Physical Measurement and Analysis*. Addison-Wesley Publishing Company, London, 1. ed., 1963.
- 33 WERNECK, Marcelo Martins. *Transdutores e Interfaces*. Livros Técnicos e Científicos Editora, Rio de Janeiro, 1. ed. 1996.
- 34 PROKHOROV, Danil. Virtual sensor and their automotive applications. *International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference.*, p. 411-416, Dec. 2005.
- 35 GOODWIN, G.C.. Evaluating the performance of virtual sensors. *Information, Decision and Control*, p. 5-12, February 1999.
- 36 HABTOM, Resson; LITZ, L.. Virtual sensors based on recurrent neural networks and the extended Kalman filter. *IEEE International Conference on Control Applications*, v. 1, p. 163-167, Trieste, Italy, Sep. 1998.
- 37 TONG, Kai Yu; GRANAT, M.H.. Using neural network with virtual sensors to generate optimum FES gait controllers. *International Conference of the IEEE Engineering in Medicine and Biology Society*, v. 5, p. 2582-2585, Oct./Nov. 1998.

- 38 HANZEVACK, E.L. et al.. Virtual sensors for spark ignition engines using neural networks. *American Control Conference*, v. 1, p. 669-673, Albuquerque, NM, USA, June 1997.
- 39 TING, T.L.. Development of a neural network based virtual sensor for automatic transmission slip. *IEEE International Symposium on Intelligent Control*, p. 721-727, Oct. 2002.
- 40 LEON, M.A.A. Arroyo; CASTRO, A. Ruiz; ASCENCIO, R.R. Leal. An artificial neural network on a field programmable gate array as a virtual sensor. *III International Workshop on Design of Mixed-Mode Integrated Circuits and Applications*, p. 114-117, Puerto Vallarta, Mexico, July 1999.
- 41 KÖGLER, João Jr.. Laboratório de Sistemas Integráveis, São Paulo. Disponível em: www.lsi.usp.br/kogler. Acesso em 2004.
- 42 BALBINOT, Alexandre; BRUSAMARELLO, Valner João. *Instrumentação e Fundamentos de Medidas*. 1. ed. Rio de Janeiro: LTC, 2006.
- 43 RODRIGUES, Helder Teixeira et al.. *Implementação De Sistema De Caracterização e Análise Automática De Dispositivos Microeletrônicos*. Recife, 2006. Disponível em www.semicondutores.ufpe.br/pesquisa/publicacoes. Acessado em 2006.
- 44 SHAW, Ian S.; SIMÕES, Marcelo Godoy. *Controle e Modelagem Fuzzy*. 1. ed. São Paulo, Edgard Blücher, LTDA, 1999.
- 45 WEINMANN, Alexander. *Uncertain Models and Robust Control*. New York, Springer-Verlag, June 1991.

- 46 MARTINEZ, José Mário; SANTOS, Sandra Augusta. *Métodos Computacionais de Otimização*. 20º Colóquio Brasileiro de Matemática, Rio de Janeiro - Brasil, 1995.
- 47 LYNCH, Nancy A.. *Distributed Algorithms*. 8. ed. San Francisco-Califórnia, 1996.
- 48 INSTONE, Ian Instone. Calculating the uncertainty of a single measurement. *IEE Colloquium on Uncertainties in Electrical Measurement*, p. 1/1-1/5, London, UK, May 1993.
- 49 BAILI, H.; FLEURY, G.A.. Indirect measurement within dynamical context: probabilistic approach to deal with uncertainty. *IEEE Transactions on Instrumentation and Measurement*, v. 53, p. 1449-1454, Dec. 2004.
- 50 XIONG, Shen-Shu; ZHOU, Zhao-Ying. Dynamic parameter estimation of velocity sensors using an indirect measurement approach. *Instrumentation and Measurement Technology Conference - IMTC*, v. 2, p. 794-797, Budapest, Hungary, May 2001.
- 51 BUCCI, G. et al.. Indirect measurements for the on-line characterization of linear induction motors. *Instrumentation and Measurement Technology Conference - IMTC*, v. 1, p. 380-383, May 1994.
- 52 BUCCI, G.; LANDI, C.. A model based measurement technique applied to linear induction motors. *Instrumentation and Measurement Technology Conference - IMTC*, p. 686, Waltham, MA, USA , April 1995.
- 53 GEDDES, L.A.. Counterpressure: the concept that made the indirect measurement of blood pressure possible. *Engineering in Medicine and Biology Magazine*, v. 17, p. 85-87, Nov. 1998.

- 54 SOLOPCHENKO, G.N.; REZNIK, L.K.; JOHNSON, W.C.. Fuzzy intervals as a basis for measurement theory. *Proceedings of the First International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference - NAFIPS/IFIS/NASA*, p. 405-406, Dec. 1994.
- 55 CROPLEY, D.H.. A graphical modelling method for the representational form of measurement theory. *Instrumentation and Measurement Technology Conference - IMTC*, v. 2, p. 1338-1343, Ottawa, Ont., Canada, May 1997.
- 56 LEE, J. *et al.*. Mathematical model of venous occlusion plethysmography for diagnosing deep vein thrombosis. *Engineering in Medicine and Biology Society*, v. 1, p. 124-125, Oct. 2001.
- 57 MOUDDEN, Yassir; BOUBAL, Olivier. Indirect measurement of cylinder pressure peak position in an internal combustion engine. *IEEE Instrumentation and Measurement Technology Conference*, v. 2, p. 772-777, Budapest, Hungary, 2001.
- 58 BAILI, H.; FLEURY, G.. Statistical characterization of a measurement approach using mcmc. *Instrumentation and Measurement Technology Conference*, v. 2, p. 1373-1376, May 2002.
- 59 CAPUA, C. De; ROMEO, E.. A t-norm based fuzzy approach to the estimation of measurement uncertainty. *Instrumentation and Measurement Technology Conference*, v. 1, p. 229-233, May 2004.
- 60 YANG, Y. *et al.*. Reflectance as an indirect measurement of the extent of laser-induced coagulation. *IEEE Transactions on Biomedical Engineering*, v. 37, p. 466-473, May 1990.

- 61 OVASKA, S.J.; VALIVIITA, S.. Angular acceleration measurement: a review. *IEEE Transactions on Instrumentation and Measurement*, v. 47, p. 1211-1217, Oct. 1998.
- 62 SCHEICK, L.Z.; McNULTY, P.J.; ROTH, D.R.. Dosimetry based on the erasure of floating gates in the natural radiation environments in space. *IEEE Transactions on Nuclear Science*, v. 45, p. 2681-2688, Dec. 1998.
- 63 MAYHAN, R.; BAILEY, R.. An indirect measurement of the effect dielectric constant and loss tangent of typical concrete roadways. *IEEE Transactions on Antennas and Propagation*, v. 23, p. 565-569, Jul 1975.
- 64 BAKALARCZYK, Jerzy. The measurements of internal signals of induction and synchronous motors. *Akademia Techniczno-Rolnicza*, p. 1-4, sa.
- 65 KREINOVICH, Vladik; TREJO, Raul. *Error estimations for indirect measurements: Randomized vs. deterministic algorithms for blackbox programs*. Disponível em: [www. cs. utep.Edu / vladik / 2000 / tr00-17.pdf](http://www.cs.utep.edu/vladik/2000/tr00-17.pdf). Acesso em 2004.
- 66 TAFAZOLI, S. et al.. Parameter estimation and actuator friction analysis for a mini excavator. *IEEE International Conference on Robotics and Automation*, v. 1, p. 329-334, Minneapolis, MN, USA, Apr 1996.
- 67 GATTU, G.; ZAFIRIOU, E.. State estimation model based algorithm for on-line optimization and control of batch processes. *American Control Conference*, v. 1, p. 480-484, June/July 1994.

- 68 TAKAHASHI, K.; SAKAGUCHI, T.; OHYA, J.. Real-time estimation of human body postures using Kalman filter. *8th IEEE International Workshop on Robot and Human Interaction. RO-MAN '99.*, p. 189-194, Pisa, Italy Sep. 1999.
- 69 HOFBAUR, M.W.; WILLIAMS, B.C.. Hybrid estimation of complex systems. *IEEE Transactions on Systems, Man and Cybernetics*, v. 34, p. 2178-2191, Oct. 2004.
- 70 HWANG, Inseok; BALAKRISHNAN, H.; TOMLIN, C.. Performance analysis of hybrid estimation algorithms. *IEEE Conference on Decision and Control*, v. 5, p. 5353-5359, Dec. 2003.
- 71 WIT, C. Canudas de; AUBIN, A.. Robot parameter identification via sequential hybrid estimation algorithm. *IEEE International Conference on Robotics and Automation*, v. 2, p. 952-957, Sacramento, CA, USA, Apr. 1991.
- 72 LUMELSKY, Vladimir J.. Estimation and prediction of unmeasurable variables in the steel mill soaking pit control system. *General Electric Company*, v. 28, p. 388-400, Mar. 1983.
- 73 BIALKOWSKI, W.L.. Application of Kalman filters to the regulation of dead time processes. *IEEE Transactions on Computer System*, p. 400-410, Sep. 1983.
- 74 NORTH, P.R.; PLUMMER, S.E.. Estimation of conifer bi-directional reflectance using a Monte Carlo method. *Geoscience and Remote Sensing Symposium, IGARSS, IEEE*, v. 1, p. 114-116, Piscataway, NJ, Aug. 1994.

- 75 FERRARI, P.; ANGENIEUX, G.. A simulation technique for the evaluation of random error effects in time-domain measurement systems. *IEEE Transactions on Instrumentation and Measurement*, v. 50, p. 665-671, June 2001.
- 76 CRISTALDI, L. *et al.*. The impact of internet transmission on the uncertainty in the electric power quality estimation by means of a distributed measurement system. *IEEE Transactions on Instrumentation and Measurement*, v. 52, p. 1073-1078, Aug. 2003.
- 77 HOWE, D.A. *et al.*. A 100 GHz AM and PM noise measurement system: preliminary design and performance. *IEEE International on Frequency Control Symposium and PDA Exhibition*, p. 690-698, 2002.
- 78 HOU, Zeng-Guang; GUPTA, M.M.. A rail damage detection and measurement system using neural networks. *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, p. 4-9, July 2004.
- 79 MARTIN, L.J.; FREI, J.. Application of statistical tools and methods for high density substrate process development [printed circuit board fabrication]. *Electronic Components and Technology Conference*, p. 690-699, May 2002.
- 80 D'ANTONA, G.; ROCCA, L.. A comparative analysis between different inversion algorithms for process tomographic measurements. *Instrumentation and Measurement Technology Conference*, v. 2, p. 1439-1443, May 2004.

- 81 BAILI, H.; FLEMY, G.. Probabilistic approach for indirect measurement within dynamical uncertain context application from electrical engineering. *Instrumentation and Measurement Technology Conference*, v. 1, p. 31-35, May 2003.
- 82 RAO, G. Sridhara; THILAGAR, S. Hosimin. A novel method for the determination of stray load losses of induction motor using genetic algorithm. *Electrical and Computer Engineering-IEEE CCECE*, v. 1, p. 259-262, May 2003.
- 83 GLAERUM, Sigurd. Talus-an object oriented air combat simulation. *Winter Simulation Conference*, v. 2, p. 1160-1167, Phoenix, AZ, USA, 1999.
- 84 ZANOBINI, A.; IUCULANO, G.; PELLEGRINI, G.. The evaluation of the combined standard uncertainty for nonlinear models. *IEEE Instrumentation and Measurement Technology Conference*, v. 1, p. 359-363, May 2002.
- 85 ANDO, B. *et al.*. A smart sensor for pressure measurement. *Instrumentation and Measurement Technology Conference*, v. 2, p. 1284-1287, Ottawa, Ont., Canada, May 1997.
- 86 HAYKIN, Simon. *Redes Neurais: princípios e prática*. 2. ed. Porto Alegre: Ed. Bookman, 2001.
- 87 MICHIE, D.; SPIEGELHALTER, D.J.; TAYLOR, C.C.. *Machine Learning, Neural and Statistical Classification*. Feb. 1994.

- 88 BRAGA, Antônio de Pádua; LUDERMIR, Teresa Bernarda; CARVALHO, André Carlos P. de L.F.. *Redes Neurais Artificiais - Teoria e Aplicações*. 1. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2000.
- 89 CAMMARATA, Silvio. *Reti Neuronalì: Dal Perceptron alle reti caotiche e neuro-fuzzy*. 2. ed. Milano: Etaslibri, 1997.
- 90 HAYKIN, Simon. *Kalman Filtering and Neural Networks*. 1. ed. New York, 2001.
- 91 CABRAL JR., Euvaldo. *Redes Neurais Artificiais: modelos em C*. 1. ed. São Paulo: Grupo ReNeArt, 2003.
- 92 EMARY, Ibrahiem M. M. El. On the application of artificial neural networks in analyzing and classifying the human chromosomes. *Journal of Computer Science*, v. 1, p. 72-75, 2006.
- 93 LIOU, Cheng-Yuan; KUO, Yen-Ting. Data flow design for the backpropagation algorithm. Disponível em: red.csie.ntu.edu.tw/publications/, p. 1-6. Acesso em 2002.
- 94 NASCIMENTO JR., Cairo Lúcio; YONEYAMA, Takashi. *Inteligência Artificial em Controle e Automação*. 1. ed. São Paulo: FAPESP, 2000.
- 95 VOSS, Henning U.; TIMMER, Jens. Nonlinear dynamical system identification from uncertain and indirect measurements. *International Journal of Bifurcation and Chaos*, v. 14, p. 1905-1932, 2004.
- 96 HUTCHINGS, Haward. *Interfacing with C*. 1. ed. Boston: Newnes Publishing Staff, 1995.

- 97 JULIER, S.; UHLMANN, J.. A general method for approximating nonlinear transformations of probability distributions. *Technical report, Dept. of Engineering Science, University of Oxford*, Nov. 1996.
- 98 JULIER, S.; UHLMANN, J.. A new extension of the Kalman filter to nonlinear systems. *In Proceedings of the International Symposium on Optical Science - Engineering and Instrumentation*, 1997.
- 99 JULIER, S.; UHLMANN, J.. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, v. 92, p. 401-422, Mar 2004.
- 100 LEWIS, F.L.. Wireless sensor networks in Smart Environments: Technologies, Protocols, and Applications. Ed. D. J. Cook and S. K. Das, John Wiley, pp. 1-18, Nova Iorque, 2004.
- 101 CHUI, Charles K.; GUANRONG. *Kalman Filtering with Real-Time Applications*. 2. ed. New York: Springer Series in Information and Sciences, v. 17, 1991.
- 102 LIN, Yu Kweng; CAI, Guo Qiang. *Probabilistic Structural Dynamics*. 5. ed. New York, 1995.
- 103 SERON, Maria M.; BRASLAVSKY, Julio.H.; GOODWIN, Graham C.. *Fundamental Limitations in Filtering and Control*. Springer, Feb. 2004.
- 104 BROWN, Robert Grover; HWANG, Patrick Y.C.. *Introduction to Random Signals and Applied Kalman Filtering*. 10. ed., New York: John Wiley & Sons, 1997.
- 105 WELCH, Greg; BISHOP, Gary. An introduction to the Kalman filter. *University of North Carolina at Chapel Hill Chapel Hill*, p. 1-16, April 2004.

- 106 KALMAN, R. E.. A new approach to linear filtering and prediction problems. *Transactions of the ASME/Journal of Basic Engineering / Research Institute for Advanced Study*, p. 35-45, 1960.
- 107 SHAW, Alan C.. *Sistemas e Software de Tempo Real*. 1. ed. Porto Alegre, 2003.
- 108 RIBEIRO NETO, Pedro Fernandes. *Gerenciamento de qualidade de serviços para banco de dados em tempo real*. 2004. Exame de qualificação - Universidade Federal de Campina Grande, Campina Grande.
- 109 GAJSKI, Daniel D. *et al.*. *Specification and Design of Embedded Systems*. 10. ed. University of California at Irvine, 1991.
- 110 FARINES, Jean-Marie; FRAGA, Joni da Silva; OLIVEIRA, Rômulo Silva de. *Sistema em Tempo Real*. Departamento de Automação e Sistemas, Florianópolis, julho 2000.
- 111 LAPLANTE, Phillip A.. *Real-Time Systems Design and analysis – An Engineer's Handbook*. 10. ed. New York: IEEE Press, 1997.
- 112 PETERSON, David A.; HENNESSY, Jonh L.. *Organização e Projetos de Computadores: a interface hardware/software*. 2. ed. Rio de Janeiro, 2000.
- 113 ABNT-INMETRO. *ISO-GUIA para a Expressão da Incerteza de Medição*. 3. ed. Rio de Janeiro: Sociedade Brasileira de Metrologia, 2003.
- 114 INMETRO. Expressão da incerteza da medição. Norma N° NIT-DICLA-021, 2003.

- 115 PIZZOLATO, Morgana. *Incertezas nas medições*. Porto Alegre, maio 2005.
- 116 ÅSTRÖM, Karl J.; WITTENMARK, Björn. *Computer-Controlled Systems Theory and Design*. 3. ed. New Jersey: Prentice Hall, 1997.
- 117 DAHLEH, Mohammed; DAHLEH, Munther A.; VERGHESE, George. Lectures on dynamic systems and control. *Massachusetts Institute of Technology - Department of Electrical Engineering and Computer Science*, cap.2/cap.3, 2005.
- 118 KREINOVICH, Vladik; TREJO, Raúl. Error estimation for indirect measurements: randomized vs deterministic algorithms for black-box programs. *University of Texas at El Paso*, p. 1-57, 2000.
- 119 HSU, Hwei Piao. *Theory and Problems of Probability, Random Variables, & Random Process*. 20. ed. New Jersey: MacGraw-Hill, 1997.
- 120 NIEVOLA, Julio Cesar. *Redes Neurais Artificiais*. Disponível em: www.ppgia.pucpr.br/nievola/DadosWEB/XII-ERI-RNA.ppt, Paraná. Acesso em 2006.
- 121 HASHEM, S.. Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. *International Joint Conferences on Neural Networks*, v. 1, p. 419-424, Baltimore, MD, USA , June 1992.
- 122 TAO, Chin-Wang *et al.* Sensitivity analysis of neural control. *University of Texas at El Paso*, p. 1-5, 2003.

- 123 ROSA, Mauricio A. Pinheiro *et al.*. A rocket tracking analysis using constant-gain filters. *4th International Conference on Inverse Problems in Engineering*. Rio de Janeiro, 2002.
- 124 BARROS, Jonas de J.. *Filtragem em Tempo Real: Adaptativa e Robusta para Rastreamento de Foguetes Multiestágios*. 2004. Dissertação de Mestrado – Universidade Federal do Maranhão, São Luís.
- 125 BROOKNER, Eli. *Tracking and Kalman Filtering Made Easy*. New York: John Wiley & Sons, Inc. 1998.
- 126 BOZIC, S.M.. *Digital and Kalman Filtering: An Introduction to Discrete-Time Filtering and Optimum Linear Estimation*. 2. ed. New York: John Wiley & Sons, Inc. 1994.
- 127 LEWIS, Frank L.. *Applied optimal control and estimation: Digital design and implementation*. 1. ed. Prentice Hall and Digital Signal Processing Serie, 1996.
- 128 AL-DHAHER, A.H.G.; FARSI, E.A.; MACKESY, D.. Data fusion architecture - an FPGA implementation. *Instrumentation and Measurement Technology Conference*, v. 3, p. 1985-1990, May 2005.
- 129 Cypress MicroSystem. PsoC designer. Disponível em www.cypressmicro.com. Acesso em 2004.
- 130 NAIK, Ankoor; VON DER MARK, Stefan. Muse wakeup receiver - psoc block device datasheet. *WINLAB (Wireless Information Network Laboratory)*, 2003.

- 131 WOODS, Robert L.; LAWRENCE, Kent L.. *Modeling and Simulation of Dynamic Systems*. Prentice Hall, 1997.
- 132 JACQUOT, Raymond G.. *Modern Digital Control Systems*. 2. ed. New York: CRC, 1997.
- 133 LUYBEN, William L.; LUYBEN, Michael L.. *Essentials of Process Control Chemical Engineering*. McGraw-Hill Companies, 1997.
- 134 FRANKLIN, Gene F.; POWELL, J. David; EMMAMI-NAEINI, Abbas. *Feedback Control of Dynamic Systems*. 4. ed. Prentice Hall, 2002.
- 135 BEJAN, Adrian. *Transferência de Calor*. 4. ed. São Paulo, Edgard Blücher, 1996.
- 136 MORAN, Michael *et al.* *Introdução à Engenharia de Sistemas Térmicos: Termodinâmica, Mecânica dos Fluidos e Transferência de Calor*. 1. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2005.
- 137 OGATA, Katsuhiko. *Engenharia de Controle Moderno*. 3. ed. Rio de Janeiro: Prentice Hall do Brasil, 1997.
- 138 BROGAN, William L.. *Modern Control Theory*. 6. ed. Prentice Hall, 1990.
- 139 GERTHSEN, C.; KNESER; VOGEL, H.. *Física*. 2. ed. Lisboa: Fundação Calouste Gulbenkian, 1998.
- 140 KRAUSE, Paul C.. *Analysis of Electric Machinery*. 10. ed. New York, 1986.

- 141 ONG, Chee-Mun. *Dynamic Simulation of Electric Machinery*. 10. ed. Prentice Hall PTR, 1998.
- 142 BOLDEA, Ion; NASAR, S.A.. *Electric Drives*. CRC press, 1998.
- 143 MOUZINHO, Lucilene F.; FONSECA NETO, João V.; LUCIANO, Benedito A.; FREIRE, Raimundo C.S.. Induction machine speed neural estimator implemented in programmable architecture. *VII CBRN - VII Congresso Brasileiro de Redes Neurais*, p. 1-6, Natal, RN, out. 2005.
- 144 MOUZINHO, Lucilene F.. *Redes neurais em estimação de velocidade para máquinas de indução*. 2003. Dissertação de Mestrado – Universidade Federal do Maranhão, São Luís.
- 145 Cypress MicroSystem. PsoC designer. Disponível em www.cypressmicro.com. Acesso em 2004.
- 146 CHAVES FILHO, João Edgar. *Estimação de Parâmetros e Detecção de Falhas Incipientes de Máquinas Assíncronas*. 2001. Tese de Doutorado – Universidade Federal da Paraíba, Campina Grande.
- 147 MEHROTRA, Prashant; QUAICOE, John E.; VENKATESAN, R.. Development of an artificial neural network based induction motor speed estimator. *IEEE Power Electronics Specialists Conference*, v. 1, p. 682-689, Baveno, Italy, Jun. 1996.
- 148 MARINO, Pompeo; MILANO, Michele; VASCA, Francesco. Robust neural network observer for induction motor control. *IEEE Power Electronics Specialists Conference*, v. 1, p. 699-705, St. Louis, MO, USA, Jun. 1997.

- 149 MARINO, Pompeo; MILANO, Michele; VASCA, Francesco. Linear quadratic state feedback and robust neural estimator for field oriented controlled induction motors. *IEEE Transactions on Industrial Electronics*, v. 46, p. 150-161, Feb. 1999.
- 150 FODOR, D.; GRIVA, G.; PROFUMO, F.. Compensation of parameters variations in induction motor drives using a neural network. *Power Electronics Specialists Conference PESC. 26th Annual IEEE*, v. 2, p. 1307-1311, Jun. 1995.
- 151 TING-HO LO, James. Robust adaptive identification of dynamic systems by neural networks. *IEEE International Conference on Neural Networks*, v. 2, p. 1121-1126, Houston, TX, USA, Jun. 1997.
- 152 Wishart, M.T.; Harley, R.G.. Identification and control of induction machines using artificial neural networks. *IEEE Transactions on Industry Applications*, v. 31, p. 612-619, May/June 1995.
- 153 MOUZINHO, Lucilene F.. *Plataforma de teste do algoritmo de kalman para estimação de posição de veículo espacial*. 2004. Projeto e Pesquisa – Universidade Federal de Campina Grande, Campina Grande.
- 154 CHENG, Chen-Mou. *An Operating System Architecture for Embedded Systems-Design and Implementation*. PhD thesis, National Taiwan University, June 1998.
- 155 EMBREE, Paul M.. *C Algorithms for Real-Time DSP*. New Jersey, 1 edition, 1995.
- 156 SALIBA, Walter Luiz Caram. *Técnicas de Programação: uma abordagem estruturada*. McGraw-Hill, São Paulo, 1 edition, 1992.

- 157 GRUN, Peter; DUTT, Nikil; NICOLAU, Alexandru. *Memory Architecture Exploration for Programmable Embedded Systems*. New York, 2 edition, 2003.
- 158 GREWAL, Mohinder S. ; ANDREWS, Angus P. . *Kalman Filtering: Theory and Practice*. John Wiley and Sons, New York, 2 edition, 2001.
- 159 FONSECA NETO, João V.; MOUZINHO, Lucilene F.; LUCIANO, Benedito A.; FREIRE, Raimundo C.S.. Aplicações de redes neuronais artificiais e medição indireta de velocidade via observador-neuronal em máquinas de indução. *VII SBAI - VII Brazilian Symposium on Intelligent Automation*, p. 1-8, São Luís, MA, set. 2005.
- 160 KUNG, Sun Yuang. *Digital Neural Networks*. 10. ed. New Jersey: PTR Prentice Hall, 1993.

Apêndice A

Modelo do Sistema de Rastreo

Um modelo estocástico linear é utilizado para representar o deslocamento do veículo aeroespacial a partir dos sinais de trajetória provenientes das medições realizadas pelos radares de rastreo. Este modelo é descrito por um conjunto de equações diferenciais ordinárias de primeira ordem. Sejam d , az , el a distância, o azimute e a elevação, respectivamente, do alvo, com o radar localizado na origem. Considera-se d , az , e el como funções do tempo com primeira e segunda derivadas denotadas por \dot{d} , \dot{az} , \dot{el} , \ddot{d} , \ddot{az} , \ddot{el} , respectivamente. Seja $T > 0$ o período de amostragem para $d_k = d(kT)$, $\dot{d}_k = \dot{d}(kT)$ e $\ddot{d}_k = \ddot{d}(kT)$. Usando a aproximação polinomial do segundo grau de *Taylor*, o modelo de rastreo do radar é descrito em espaço de estado linear estocástico por[153]:

$$\begin{bmatrix} x^1_{k+1} \\ x^2_{k+1} \\ x^3_{k+1} \end{bmatrix} = \begin{bmatrix} A_k^1 & \bar{0} & \bar{0} \\ \bar{0} & A_k^2 & \bar{0} \\ \bar{0} & \bar{0} & A_k^3 \end{bmatrix} \begin{bmatrix} x^1_k \\ x^2_k \\ x^3_k \end{bmatrix} + \begin{bmatrix} G_k^1 & \bar{0} & \bar{0} \\ \bar{0} & G_k^2 & \bar{0} \\ \bar{0} & \bar{0} & G_k^3 \end{bmatrix} \begin{bmatrix} \xi^1_k \\ \xi^2_k \\ \xi^3_k \end{bmatrix} \quad (\text{A.1})$$

e

$$\begin{bmatrix} y^1_k \\ y^2_k \\ y^3_k \end{bmatrix} = \begin{bmatrix} C^1_k & \bar{0} & \bar{0} \\ \bar{0} & C^2_k & \bar{0} \\ \bar{0} & \bar{0} & C^3_k \end{bmatrix} \begin{bmatrix} x^1_k \\ x^2_k \\ x^3_k \end{bmatrix} + \begin{bmatrix} \eta^1_k \\ \eta^2_k \\ \eta^3_k \end{bmatrix} \quad (\text{A.2})$$

sendo $x_k = [d_k \ \dot{d}_k \ \ddot{d}_k \ az_k \ a\dot{z}_k \ a\ddot{z}_k \ el_k \ e\dot{l}_k \ e\ddot{l}_k]^T$.

As matrizes $A^i_k = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$, $C^i_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $\{\xi_k\}$, $\{\eta_k\}$ seqüências de ruídos

brancos independentes de média zero com $Var(\xi_k) = Q_k$ e $Var(\eta_k) = R_k$.

Supondo que $Q_k = \begin{bmatrix} Q^1_k & \bar{0} & \bar{0} \\ \bar{0} & Q^2_k & \bar{0} \\ \bar{0} & \bar{0} & Q^3_k \end{bmatrix}$, $R_k = \begin{bmatrix} R^1_k & \bar{0} & \bar{0} \\ \bar{0} & R^2_k & \bar{0} \\ \bar{0} & \bar{0} & R^3_k \end{bmatrix}$, sendo G^i_k sub-

matrizes 3×3 , Q^i_k sub-matrizes simétricas 3×3 não-negativa definidas e R^i_k sub-matrizes

simétricas 3×3 definidas positivas, para $i = 1, 2, 3$. O sistema pode ser decomposto

em três sub-sistemas no espaço de estados. Sejam, $x_k = [x^1_k x^2_k x^3_k]^T$, $x^1_k = [d_k \ \dot{d}_k \ \ddot{d}_k]^T$,

$x^2_k = [az_k \ a\dot{z}_k \ a\ddot{z}_k]^T$, $x^3_k = [el_k \ e\dot{l}_k \ e\ddot{l}_k]^T$, $\xi_k = [\xi^1_k \ \xi^2_k \ \xi^3_k]^T$, $\eta_k = [\eta^1_k \ \eta^2_k \ \eta^3_k]^T$, $y_k = [y^1_k \ y^2_k \ y^3_k]^T$.

O sistema de Equações (A.1) e (A.2) é escrito como:

$$x^i_{k+1} = A^i_k x^i_k + G^i_k \xi^i_k \quad (\text{A.3})$$

e

$$y_k^i = Cx_k^i + \eta_k^i \quad (\text{A.4})$$

com $i = 1, 2, 3$.

Para o projeto e análise do filtro, considera-se somente a coordenada Z . O modelo do sistema é descrito por três equações no espaço de estados para cada um dos eixos do sistema tridimensional de referência (X, Y, Z) . Considera-se que $x(t)$, $0 \leq t < \infty$, especifica a trajetória no espaço de um objeto em vôo, Figura A.1, sendo t a variável tempo e $T > 0$ o período de amostragem, a equação de estado para o modelo discreto é escrita como $x_k \doteq x(kT)$, $k = 0, 1, \dots$

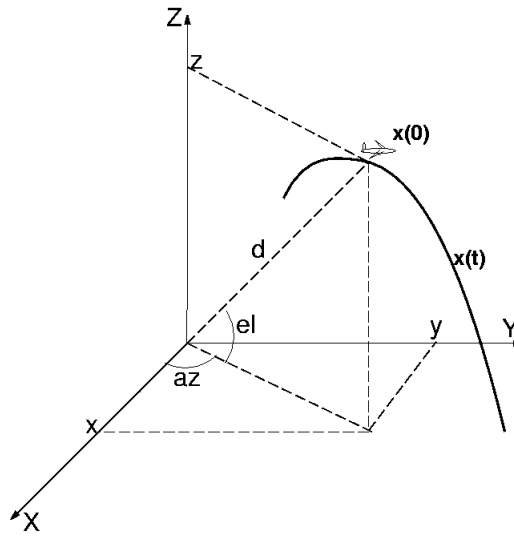


Figura A.1: Trajetória de um veículo aeroespacial.

Considera-se que $x(t)$ tem derivadas de primeira e segunda ordem contínuas, denotadas por \dot{x} e \ddot{x} , respectivamente [101], [126]. As equações da posição e velocidade são

representadas como $x_{k+1} = x_k + T\dot{x}_k + \frac{1}{2}T^2\ddot{x}_k$, $\dot{x}_{k+1} = \dot{x}_k + T\ddot{x}_k$ e $\ddot{x}_{k+1} = \ddot{x}_k$. Sendo $x_k = x_k^1$, $\dot{x}_k = x_k^2$, $\ddot{x}_k = x_k^3$, e $k = 0, 1, \dots$. O sistema é decomposto, levando-se em consideração somente o eixo Z ,

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ x_{k+1}^3 \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_k^1 \\ \xi_k^2 \\ \xi_k^3 \end{bmatrix},$$

$$\text{e } y_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \end{bmatrix} + \eta_k, \text{ sabendo-se que } x_k = \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \end{bmatrix} = \begin{bmatrix} s(t_k) \\ v_{el}(t_k) \\ a(t_k) \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, C = [100] \text{ e } \eta_k = \eta(k) \text{ é um escalar. O modelo de estado e as}$$

medidas da posição em uma direção específica são representados como, $x_{k+1} = Ax_k + G\xi_k$

e $y_k = Cx_k + \eta_k$. Considera-se $\xi_k = \begin{bmatrix} \xi_k^1 & \xi_k^2 & \xi_k^3 \end{bmatrix}^T$ e η_k com seqüências de ruídos

brancos *Gaussiano* de média zero satisfazendo $E(\xi_k) = 0$, $E(\eta_k) = 0$, $E(\xi_k \xi_k^T) = Q_k \delta_{k\ell}$,

$E(\eta_k \eta_k^T) = R_k \delta_{k\ell}$, $E(\xi_k \eta_k) = 0$, $E(x_k \xi_k^T) = 0$, $E(x_k \eta_k) = 0$ e Q_k uma matriz simétrica

não-negativa definida, $Q_k > 0$ para todo k .

A.1 Algoritmo do Filtro de *Kalman*

O algoritmo do *FK* é desenvolvido e testado no ambiente *MATLAB*[®]. Os valores considerados medidos na simulação são obtidos a partir da análise matemática da dinâmica de uma partícula. Aplicando conceitos de cinemática e considerando o movimento de uma partícula no espaço em \mathbb{R}^3 , obtém-se o vetor de posição do foguete no instante t , o vetor da velocidade a partir da derivada do vetor de posição e o vetor de aceleração a partir da derivada do vetor velocidade. Considera-se a massa M do foguete e que seu movimento tenha sido causado por um campo de forças em \mathbb{R}^3 . Aplica-se as *leis de Newton* para obter o modelo matemático adequado e representativo do foguete nas fases balística e propulsada, momento em que é estabelecida a conexão do foguete e da força que produz o movimento, ou seja, quando a massa decresce pelo consumo do combustível; o estado do foguete quando está sem a ação da força e a reação do foguete, considerando o deslocamento vertical sujeito somente a força gravitacional da Terra. Na Figura A.2 é ilustrado o fluxograma do algoritmo utilizado na implementação.

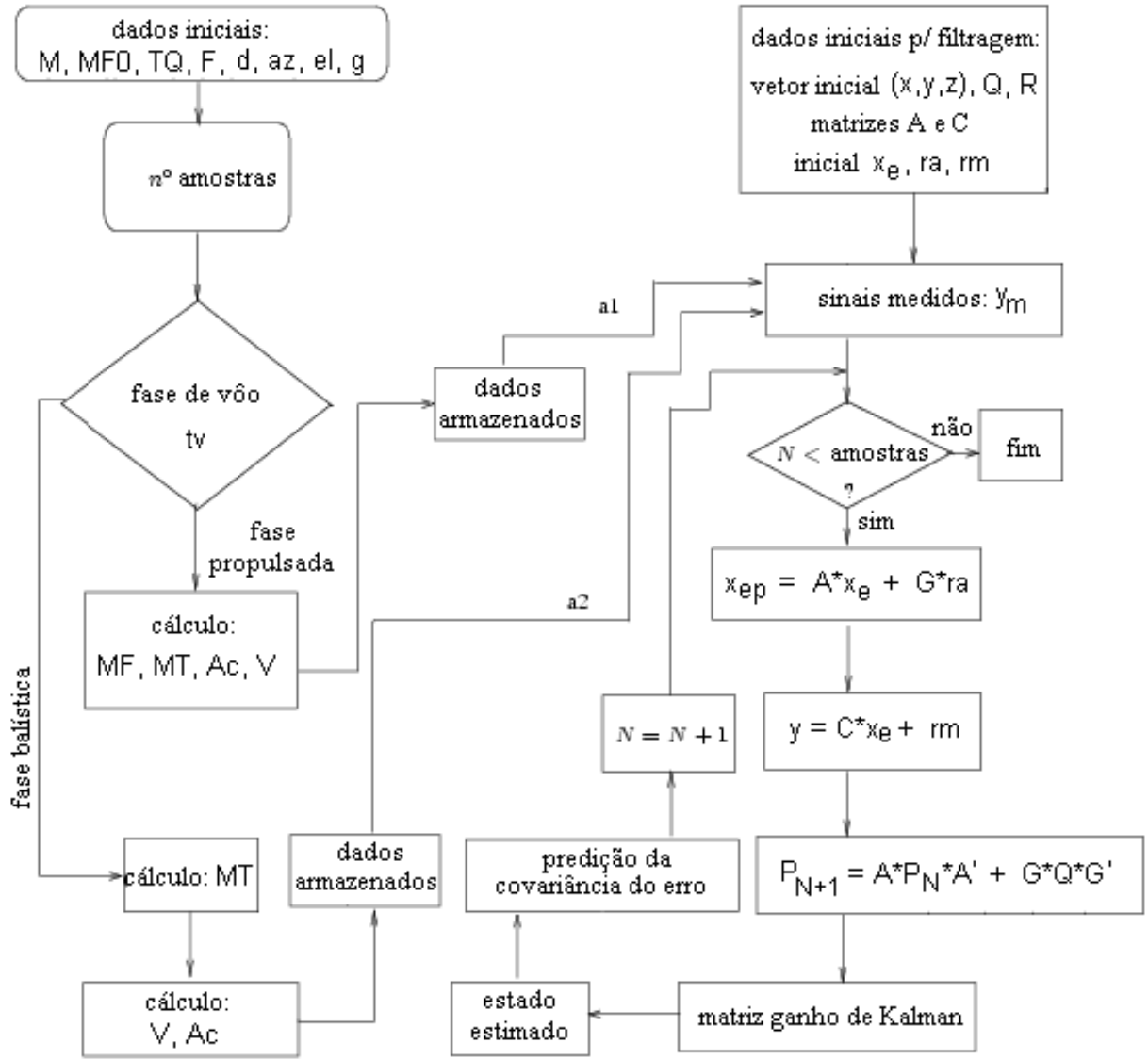


Figura A.2: Fluxograma do filtro de *Kalman* implementado.

A.2 Resultados Computacionais

O algoritmo do fluxograma, Figura A.2, é desenvolvido no ambiente *MATLAB*[®]. Nas Figuras A.3, A.4, A.5 e A.6 são ilustradas as curvas de posição, velocidade, aceleração e ganho obtidas para a coordenada *Z*, considerando 100 amostras de sinal. Os principais arquivos da simulação, nos quais são armazenados os valores medidos durante a fase propulsada e balística, têm tamanhos 11 Koctetos e 15 Koctetos, respectivamente.

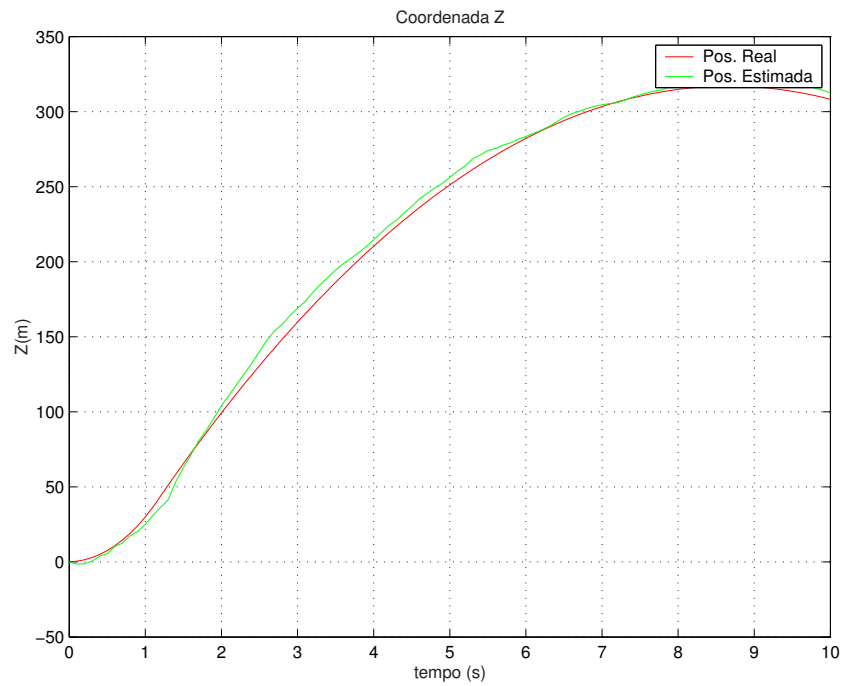


Figura A.3: Posição do veículo aeroespacial.

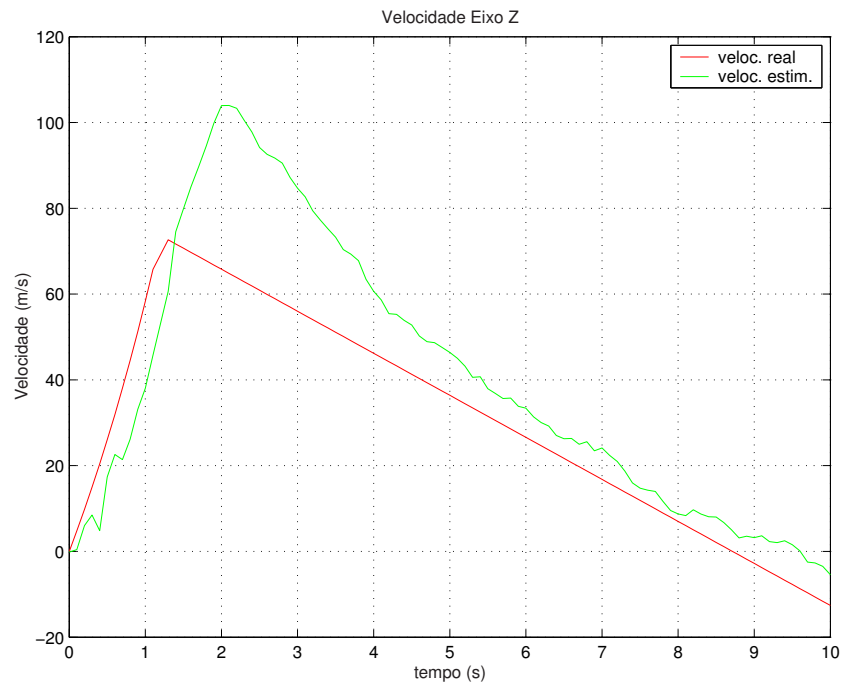


Figura A.4: Velocidade do veículo aeroespacial.

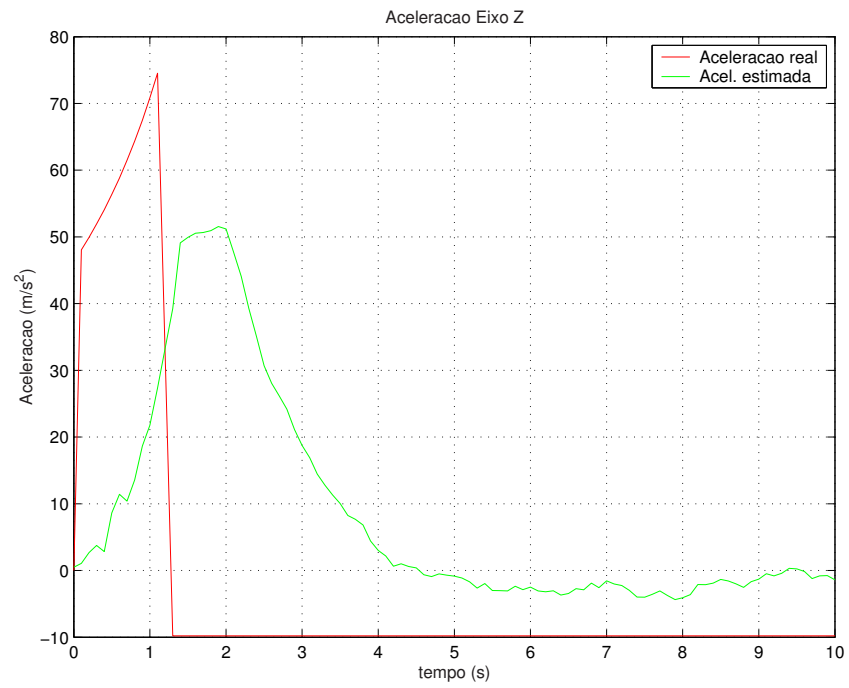


Figura A.5: Aceleração do veículo aeroespacial.

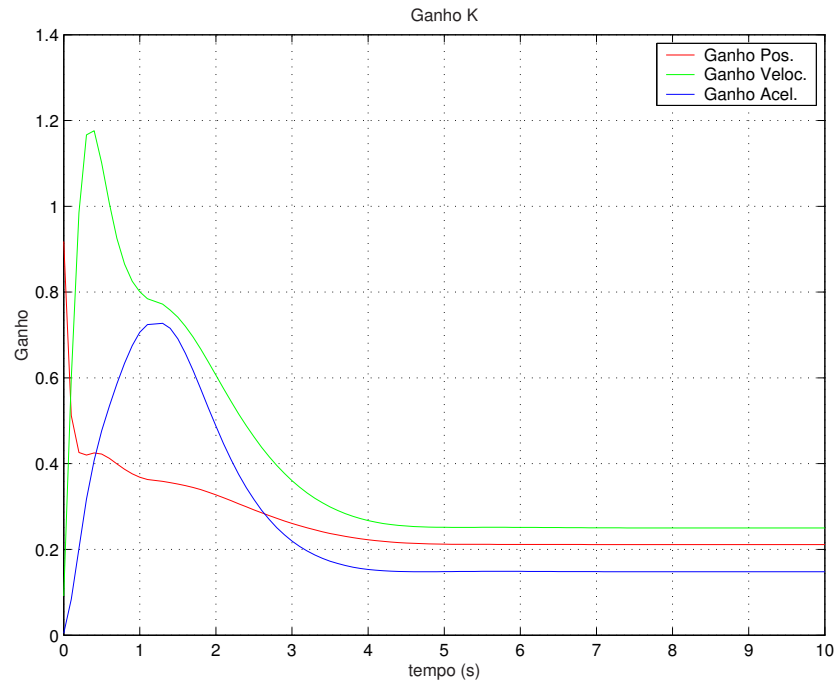


Figura A.6: Curvas de ganho de *Kalman* para posição, velocidade e aceleração no modelo.

Os sinais de trajetória usados pelo programa computacional são sinais medidos pelos radares de um pequeno foguete de teste do *CLA-MA*. A seguir, tem-se as especificações do veículo aeroespacial, o *VS-40*.

- **Dados do Veículo Aeroespacial:**

- massa do foguete: 5,15 kg;
- massa inicial do combustível: 2,6 kg;
- taxa de queima do combustível (tempo de queima 1,10 s): 2,36;
- empuxo na direção do eixo do foguete provocado

- pela queima do combustível: 450 kg/s;
- tempo de vôo na vertical: 10 s;
- aceleração gravitacional: $-9,81 \text{ m/s}^2$.

Apêndice B

Desenvolvimento da Regra Delta

Neste item tem-se o desenvolvimento da *Regra Delta* que foi utilizada na medição indireta de velocidade rotórica da máquina de indução. Os pesos w_1 , w_2 e w_3 também denominados *pesos sinápticos*, são sintonizados para minimizar a função de energia [7], [86], [88], [159]. O valor instantâneo da energia do erro para esses neurônios é definido como:

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

ou seja,

$$E = \frac{1}{2} \varepsilon' \varepsilon = \frac{1}{2} \|\vec{\lambda}_d(k) - \vec{\lambda}_a(k)\|^2. \quad (\text{B.2})$$

Como dentre os pesos da rede neuronal somente W_2 varia, segue que a correção $\Delta w_2(k)$ aplicada pelo algoritmo de retropropagação a esse peso sináptico variável é dada por:

$$\Delta w_2(k) \propto -\frac{\partial E}{\partial w_2}. \quad (\text{B.3})$$

Pela regra da cadeia, o gradiente $\frac{\partial E}{\partial w_2}$ pode ser expresso como [93], [160]:

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial \varepsilon} \frac{\partial \varepsilon}{\partial x_2} \frac{\partial x_2}{\partial \lambda_a} \frac{\partial \lambda_a}{\partial w_2}, \quad (\text{B.4})$$

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 .

Diferenciando-se ambos os lados da Equação (B.1) em relação a ε , tem-se

$$\frac{\partial E}{\partial \varepsilon} = \vec{\varepsilon}. \quad (\text{B.5})$$

Substituindo-se $\vec{\lambda}_a$ da Equação $x_2 = J\vec{\lambda}_a(k-1)$ na Equação (5.47) e diferenciando-se ambos os lados dessa equação em relação a x_2 , tem-se

$$\frac{\partial \varepsilon}{x_2} = -1. \quad (\text{B.6})$$

Se $x_2 = J\vec{\lambda}_a(k-1)$, pode-se dizer, então, que

$$x_2 = f(\vec{\lambda}_a). \quad (\text{B.7})$$

Logo, diferenciando-se a Equação (B.7) em relação a $\vec{\lambda}_a$, tem-se

$$\frac{\partial x_2}{\partial \vec{\lambda}_a} = f'(\vec{\lambda}_a). \quad (\text{B.8})$$

Diferenciando-se a Equação (5.43) em relação a w_2 , produz-se

$$\frac{\partial \lambda_a}{\partial w_2} = x_2. \quad (\text{B.9})$$

Substituindo-se as Equações (B.5), (B.6) e (B.8) na Equação (B.4), tem-se

$$\frac{\partial E}{\partial w_2} = \varepsilon(-1)f'(\vec{\lambda}_a)x_2. \quad (\text{B.10})$$

A correção $\Delta w_2(k)$ aplicada a w_2 , definida como *Regra Delta*, é dada por:

$$\Delta w_2(k) = -\eta \frac{\partial E}{\partial \vec{\lambda}_a}, \quad (\text{B.11})$$

sendo η o parâmetro da taxa de aprendizagem do algoritmo de retropropagação estático e, o sinal negativo, indicando 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 .

Substituindo-se a Equação (B.10) na Equação (B.11), tem-se

$$\Delta w_2(k) = -\eta \varepsilon(-1)f'(\vec{\lambda}_a)x_2, \quad (\text{B.12})$$

ou seja,

$$\Delta w_2(k) = -\eta \delta(k)x_2, \quad (\text{B.13})$$

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

$$\delta(k) = -\frac{\partial E}{\partial \vec{\lambda}_a}. \quad (\text{B.14})$$

Diferenciando-se a Equação (B.2) em relação a $\vec{\lambda}_a$, vem

$$\frac{\partial E}{\partial \vec{\lambda}_a} = \frac{1}{2} \frac{\partial [(\vec{\lambda}_d(k) - \vec{\lambda}_a(k))'(\vec{\lambda}_d(k) - \vec{\lambda}_a(k))]}{\partial \vec{\lambda}_a(k)}, \quad (\text{B.15})$$

$$\frac{\partial E}{\partial \vec{\lambda}_a} = \frac{1}{2} \frac{\partial [\vec{\lambda}_d(k)' \vec{\lambda}_d(k) - \vec{\lambda}_d(k)' \vec{\lambda}_a(k) - \vec{\lambda}_a(k)' \vec{\lambda}_d(k) + \vec{\lambda}_a(k)' \vec{\lambda}_a(k)]}{\partial \vec{\lambda}_a(k)} \quad (\text{B.16})$$

como $\vec{\lambda}_d(k)' \vec{\lambda}_a(k) = \vec{\lambda}_a(k)' \vec{\lambda}_d(k)$ então,

$$\frac{\partial E}{\partial \vec{\lambda}_a} = \frac{1}{2} \frac{\partial [(\vec{\lambda}_d(k)^2)' - (2\vec{\lambda}_d(k))' \vec{\lambda}_a(k) + (\vec{\lambda}_a(k)^2)']}{\partial \vec{\lambda}_a(k)} \quad (\text{B.17})$$

derivando,

$$\frac{\partial E}{\partial \vec{\lambda}_a} = \frac{1}{2} (-2\vec{\lambda}_d(k)' + 2\vec{\lambda}_a(k)'), \quad (\text{B.18})$$

$$\frac{\partial E}{\partial \vec{\lambda}_a} = (-\vec{\lambda}_d(k)' + \vec{\lambda}_a(k)'), \quad (\text{B.19})$$

substituindo-se a Equação (B.19) em (B.14), vem

$$\delta(k) = (\vec{\lambda}_a - \vec{\lambda}_d)'. \quad (\text{B.20})$$

Substituindo-se a Equação (B.14) e $x_2 = J\vec{\lambda}_a(k-1)$ na Equação (B.13), tem-se

$$\Delta w_2 = -\eta(\vec{\lambda}_a(k) - \vec{\lambda}_d(k))' J\vec{\lambda}_a(k-1). \quad (\text{B.21})$$

Da *Regra Delta*, conforme Figura 3.3, tem-se que o novo peso será

$$w_2(k) = w_2(k-1) + \eta\Delta w_2(k), \quad (\text{B.22})$$

sendo η , o coeficiente de treinamento e o coeficiente k é incrementado de 1 para cada varredura por meio do conjunto de entrada-saída.

A *Regra Delta* da Equação (B.13) é modificada de maneira a aumentar a taxa de aprendizado sem conduzir a oscilações, incluindo-se, assim, um termo de momento, conforme a Equação (B.23):

$$\Delta w_2(k) = -\eta\delta(k)x_2 + \alpha\Delta w_2(k-1). \quad (\text{B.23})$$

O coeficiente α , constante de momento, determina o efeito das modificações dos pesos anteriores no peso atual. É preferível, portanto, utilizar a Equação (B.23) ao invés da Equação (B.22).

Sabendo-se que $w_2 = \omega_r T$, então $\omega_r = \frac{w_2}{T}$, variando-se ω_r , vem

$$\Delta\omega_r = \frac{\Delta w_2}{T}. \quad (\text{B.24})$$

E, finalmente, substituindo-se a Equação (B.23) na Equação (B.24), produz-se a velocidade estimada do rotor que é dada por

$$\Delta\omega_r = -\frac{1}{T}\eta\delta(k)x_2 + \frac{1}{T}\alpha\Delta w_2(k-1), \quad (\text{B.25})$$

ou seja,

$$\hat{\omega}_r(k) - \hat{\omega}_r(k-1) = -\frac{1}{T}\eta\delta(k)x_2 + \frac{1}{T}\alpha\Delta w_2(k-1), \quad (\text{B.26})$$

então, a velocidade estimada é [145],

$$\hat{\omega}_r(k) = \hat{\omega}_r(k-1) - \frac{1}{T}\eta\delta(k)x_2 + \frac{1}{T}\alpha\Delta w_2(k-1). \quad (\text{B.27})$$

O comportamento da tensão no eixo estacionário $dq0$ é ilustrado, por meio de simulação computacional, na Figura B.1. Na Figura B.2 é ilustrada a velocidade rotórica medida indiretamente.

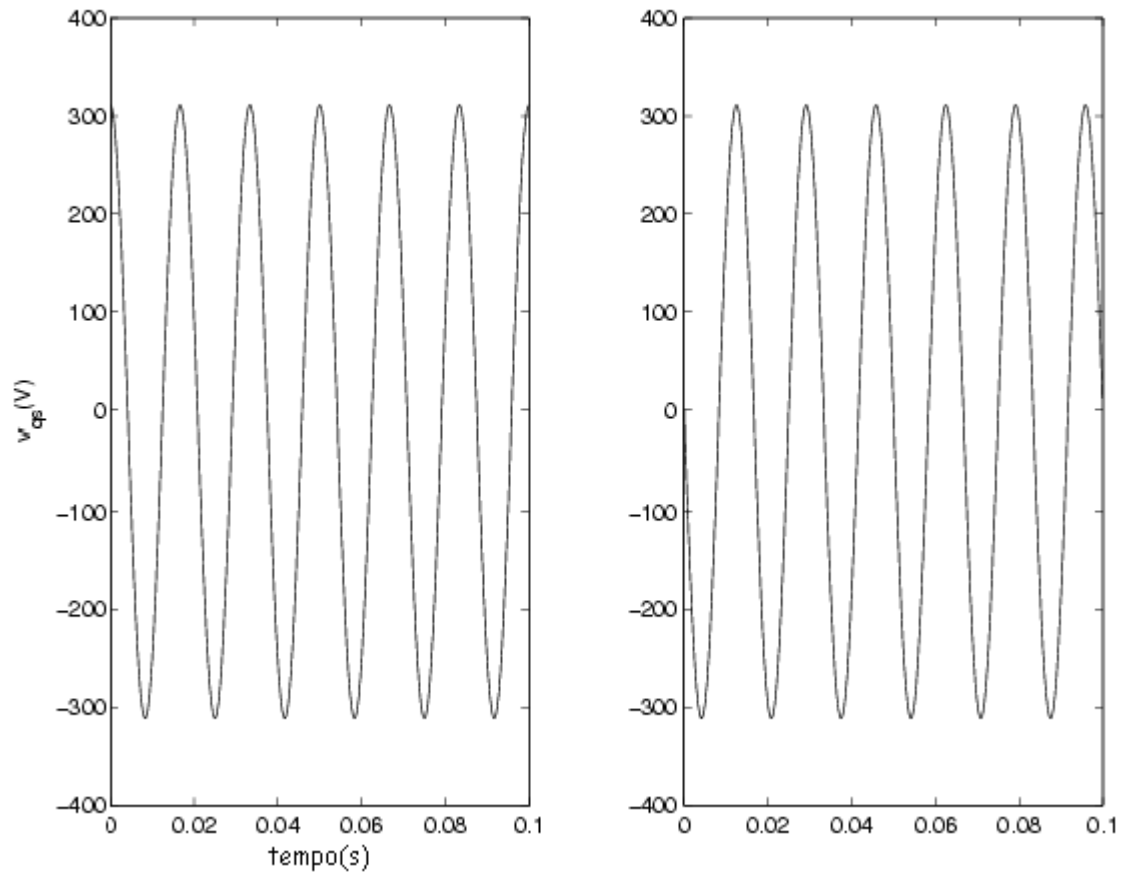


Figura B.1: Tensão no estator obtida por simulação.

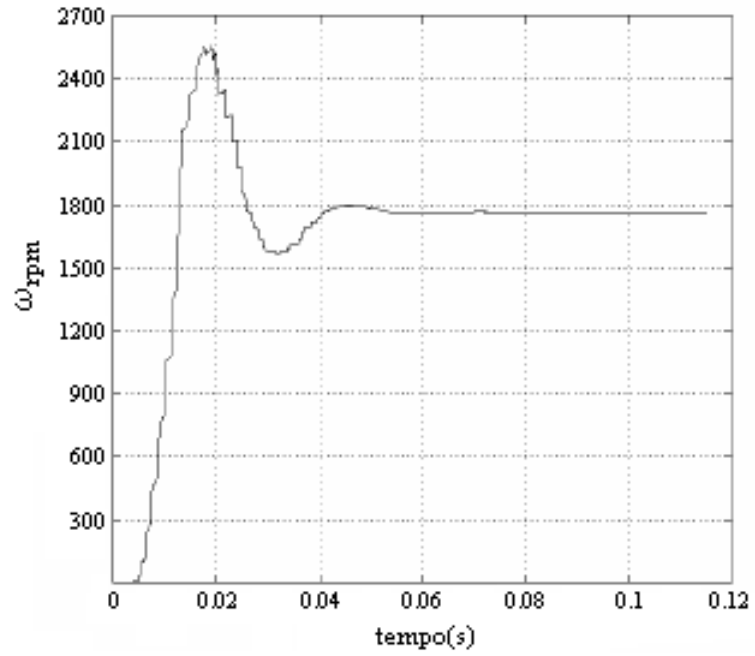


Figura B.2: Velocidade rotórica medida indiretamente.

LM35 Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^\circ\text{C}$ at room temperature and $\pm 3/4^\circ\text{C}$ over a full -55 to $+150^\circ\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only $60\ \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to $+150^\circ\text{C}$ temperature range, while the LM35C is rated for a -40° to $+110^\circ\text{C}$ range (-10° with improved accuracy). The LM35 series is available packaged in

hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full -55° to $+150^\circ\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than $60\ \mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, $0.1\ \Omega$ for 1 mA load

Typical Applications

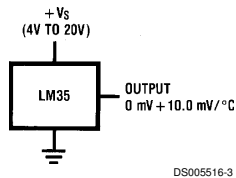
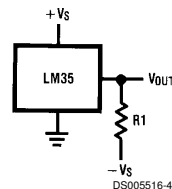


FIGURE 1. Basic Centigrade Temperature Sensor
($+2^\circ\text{C}$ to $+150^\circ\text{C}$)

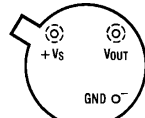


Choose $R_1 = -V_S/50\ \mu\text{A}$
 $V_{\text{OUT}} = +1,500\ \text{mV}$ at $+150^\circ\text{C}$
 $= +250\ \text{mV}$ at $+25^\circ\text{C}$
 $= -550\ \text{mV}$ at -55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

**TO-46
Metal Can Package***

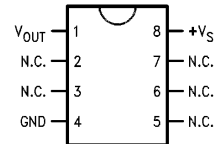


BOTTOM VIEW
DS005516-1

*Case is connected to negative pin (GND)

**Order Number LM35H, LM35AH, LM35CH, LM35CAH or
LM35DH**
See NS Package Number H03H

**SO-8
Small Outline Molded Package**

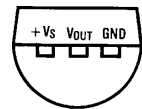


DS005516-21

N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

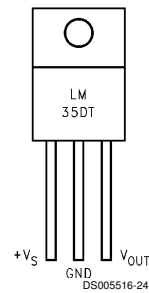
**TO-92
Plastic Package**



BOTTOM VIEW
DS005516-2

**Order Number LM35CZ,
LM35CAZ or LM35DZ**
See NS Package Number Z03A

**TO-220
Plastic Package***



DS005516-24

*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package,	
(Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5	± 1.0	$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3		± 1.0	$^\circ\text{C}$
	$T_A = T_{MAX}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = T_{MIN}$	± 0.4	± 1.0		± 0.4		± 1.5	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.18		± 0.35	± 0.15		± 0.3	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.5		± 3.0	± 0.5		± 3.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.05		± 0.01	± 0.05		mV/V
	$4V \leq V_S \leq 30V$	± 0.02		± 0.1	± 0.02		± 0.1	mV/V
Quiescent Current (Note 9)	$V_S = +5V, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5V$	105		131	91		114	μA
	$V_S = +30V, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30V$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4V \leq V_S \leq 30V$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of <i>Figure 1</i> , $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{MAX}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	±0.4	±1.0		±0.4	±1.0		°C
	$T_A = -10^\circ\text{C}$	±0.5			±0.5		±1.5	°C
	$T_A = T_{\text{MAX}}$	±0.8	±1.5		±0.8		±1.5	°C
	$T_A = T_{\text{MIN}}$	±0.8		±1.5	±0.8		±2.0	°C
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				±0.6	±1.5		°C
	$T_A = T_{\text{MAX}}$				±0.9		±2.0	°C
	$T_A = T_{\text{MIN}}$				±0.9		±2.0	°C
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	±0.3		±0.5	±0.2		±0.5	°C
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.8, +10.2		+10.0		+9.8, +10.2	mV/°C
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$	±0.4	±2.0		±0.4	±2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	±0.5		±5.0	±0.5		±5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	±0.01	±0.1		±0.01	±0.1		mV/V
	$4V \leq V_S \leq 30V$	±0.02		±0.2	±0.02		±0.2	mV/V
Quiescent Current (Note 9)	$V_S = +5V, +25^\circ\text{C}$	56	80		56	80		µA
	$V_S = +5V$	105		158	91		138	µA
	$V_S = +30V, +25^\circ\text{C}$	56.2	82		56.2	82		µA
	$V_S = +30V$	105.5		161	91.5		141	µA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		µA
	$4V \leq V_S \leq 30V$	0.5		3.0	0.5		3.0	µA
Temperature Coefficient of Quiescent Current		+0.39		+0.7	+0.39		+0.7	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of <i>Figure 1</i> , $I_L = 0$	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	±0.08			±0.08			°C

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5\text{Vdc}$ and $I_{\text{LOAD}} = 50 \mu\text{A}$, in the circuit of *Figure 2*. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mV}/^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in °C).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.