

LEANDRO DOS SANTOS COELHO

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA**

**IDENTIFICAÇÃO E CONTROLE DE PROCESSOS
MULTIVARIÁVEIS VIA METODOLOGIAS AVANÇADAS
E INTELIGÊNCIA COMPUTACIONAL**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

**FLORIANÓPOLIS
2000**

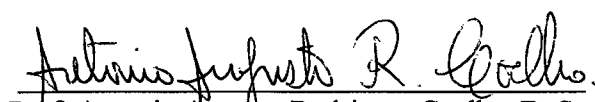
LEANDRO DOS SANTOS COELHO

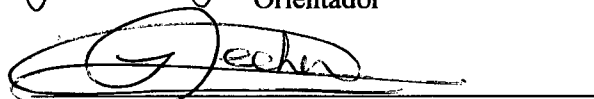
Florianópolis, Fevereiro de 2000.

IDENTIFICAÇÃO E CONTROLE DE PROCESSOS MULTIVARIÁVEIS VIA METODOLOGIAS AVANÇADAS E INTELIGÊNCIA COMPUTACIONAL

Leandro dos Santos Coelho

‘Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração em *Sistemas de Informação*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’



Prof. Antonio Augusto Rodrigues Coelho, D. Sc.
Orientador

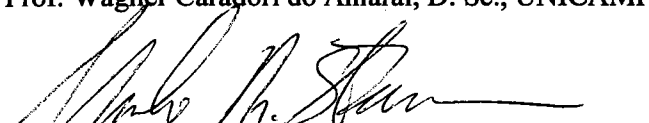

Prof. Ildemar Cassana Decker, D. Sc.

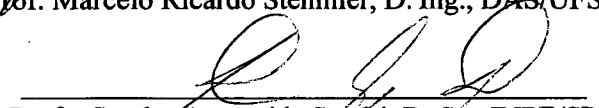
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:


Prof. Antonio Augusto Rodrigues Coelho, D. Sc., DAS/UFSC
Presidente


Prof. Wagner Caradori do Amaral, D. Se., UNICAMP


Prof. Marcelo Ricardo Stemmer, D. Ing., DAS/UFSC


Profa. Sandra Aparecida Sandri, D. Sc., INPE/SP


Prof. Fernando Mendes de Azevedo, D. Sc., GPEB/UFSC


Prof. Jorge Muniz Barreto, D. Sc., INE/UFSC

AGRADECIMENTOS

Ao professor Antonio Augusto Rodrigues Coelho, pela orientação, paciência e dedicação demonstrada nas etapas do desenvolvimento desta tese, além do incentivo e amizade ao longo da nossa convivência.

Aos colegas e professores do Departamento de Automação e Sistemas pelos laços de amizade e companheirismo.

Aos integrantes do Laboratório de Controle de Processos (*LCP*) que contribuíram para o andamento deste trabalho.

Aos revisores pelas valiosas sugestões para o aprimoramento desta tese.

Em especial, a minha esposa Viviana que sempre ajudou nos momentos difíceis com amor e incentivo, além da compreensão e a paciência que tornaram possível esta caminhada.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (*CNPq*) pelo apoio financeiro.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

IDENTIFICAÇÃO E CONTROLE DE PROCESSOS MULTIVARIÁVEIS VIA METODOLOGIAS AVANÇADAS E INTELIGÊNCIA COMPUTACIONAL

Leandro dos Santos Coelho

Fevereiro / 2000

Orientador: Antonio Augusto Rodrigues Coelho.

Área de Concentração: Sistemas de Informação.

Palavras-chave: identificação, controle avançado, controle preditivo, controle de estrutura variável, inteligência computacional, redes neurais artificiais, computação evolutiva, sistemas nebulosos, processos multivariáveis.

Número de Páginas: 342.

RESUMO: Nas últimas décadas, a identificação e o controle de processos multivariáveis são áreas de interesse crescente na comunidade de sistemas de controle. Os métodos clássicos de controle, usualmente utilizados na indústria, apresentam limitações quando aplicados a processos multivariáveis com características complexas, tais como: não-linearidades, instabilidade, atraso de transporte não-unitário, comportamento não-estacionário e acoplamento entre as variáveis. Diversas abordagens não-convencionais, destacando-se as metodologias de controle auto-ajustável, preditivo e inteligente são propostas na literatura para aplicações multivariáveis. Esta tese estuda os fundamentos e a aplicação de metodologias de controle preditivo, controle de estrutura variável e inteligência computacional em processos não-lineares multivariáveis. As metodologias da inteligência computacional abordadas são: a computação evolutiva, os sistemas nebulosos, as redes neurais artificiais e os sistemas híbridos inteligentes. Os paradigmas da computação evolutiva abordados são: algoritmos genéticos, estratégias evolutivas, programação evolutiva e algoritmos híbridos com *simulated annealing* e método simplex. Os sistemas nebulosos abordados visam a identificação de processos e na estruturação de projetos do tipo controle *PID* nebuloso. O outro paradigma inteligente apresentado são as redes neurais artificiais do tipo: *perceptron* multicamadas, parcialmente recorrente de Elman, *Group Method of Data Handling* e função de base radial. Os aspectos de projeto preditivo e de estrutura variável tratam a abordagem adaptativa, através do estimador dos mínimos quadrados recursivo, e a hibridização com as metodologias da inteligência computacional. As metodologias de controle preditivo generalizado e por modos quase-deslizantes combinado ao controle de variância mínima generalizada são descritas e analisadas. As simulações visam identificar um processo experimental não-linear balanço horizontal, em malha fechada; a dinâmica de um manipulador robótico, com dois graus de liberdade; e um processo não-linear apresentando acoplamentos entre as variáveis de saída. A configuração e a análise dos algoritmos de controle tratam o controle do manipulador robótico, utilizado no procedimento de identificação.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

MULTIVARIABLE PROCESSES CONTROL AND IDENTIFICATION VIA ADVANCED METHODOLOGIES AND COMPUTATIONAL INTELLIGENCE

Leandro dos Santos Coelho

February / 2000

Advisor: Antonio Augusto Rodrigues Coelho.

Area of Concentration: Information Systems.

Keywords: identification, advanced control, predictive control, computational intelligence, variable structure control, artificial neural networks, evolutionary computation, fuzzy systems, multivariable processes.

Number of Pages: 342.

ABSTRACT: In the last decades, identification and control of multivariable processes can be considered two important areas of interest in the control community. Classical control methods, usually utilized in industry, present limitations when applied to multivariable processes with complex features, such as: nonlinearities, unstability, non-unitary time delay, non-stationary behavior, and coupling between variables. Several non-conventional approaches have been proposed in the literature, standing out powerful tools of self-tuning, predictive and intelligent control for multivariable applications. This thesis studies concepts and application of predictive control, variable structure control and computational intelligence methodologies to multivariable nonlinear processes. The computational intelligence methodologies studied here are: evolutionary computation, fuzzy systems, artificial neural networks, and intelligent hybrid systems. The evolutionary computation paradigms are: genetic algorithms, evolution strategies, evolutionary programming, and hybrid algorithms with simulated annealing and simplex method. Fuzzy systems approaches are used in tasks of identification and fuzzy *PID* control design. Other intelligent paradigms presented are artificial neural networks, of the type multilayer perceptron, Elman partially recurrent model and radial basis function. Design aspects of predictive and structure variable controls are presented, mainly relevant topics of adaptive implementation through of recursive least-squares algorithm and computational intelligence methodologies. Generalized predictive control and quasi-sliding mode with generalized minimum variance control approaches are described and analysed. Simulation results deal with the closed-loop identification of a nonlinear experimental process denominated horizontal balance, the dynamic of a robotic manipulator with two degrees-of-freedom, and a nonlinear process, with coupling between the output variables. The analysed control configurations and algorithms are presented for the robotic manipulator control, used in the identification procedure.

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO E RELEVÂNCIA DO TRABALHO	1
1.2 INTELIGÊNCIA COMPUTACIONAL	4
1.2.1 SISTEMAS NEBULOSOS	7
1.2.2 REDES NEURAIS	8
1.2.3 COMPUTAÇÃO EVOLUTIVA	9
1.2.4 SISTEMAS HÍBRIDOS INTELIGENTES	10
1.3 IDENTIFICAÇÃO DE PROCESSOS	12
1.4 CONTROLE PREDITIVO, PREDITIVO ADAPTATIVO E BASEADO EM <i>IC</i>	12
1.5 CONTROLE DE ESTRUTURA VARIÁVEL	15
1.6 OBJETIVOS DA TESE.....	16
1.7 ORGANIZAÇÃO DA TESE.....	17
2. COMPUTAÇÃO EVOLUTIVA: FUNDAMENTOS E APLICAÇÕES	19
2.1 INTRODUÇÃO	19
2.1.1 PRINCÍPIOS DOS PROCEDIMENTOS EVOLUTIVOS NATURAIS	21
2.1.2 INSPIRAÇÃO BIOLÓGICA PARA A EVOLUÇÃO E O APRENDIZADO	22
2.2 BREVE HISTÓRICO	23
2.3 POR QUE UTILIZAR OS <i>AEs</i> ?	25
2.4 APLICAÇÕES INDUSTRIAIS	26
2.5 ALGUMAS FORMULAÇÕES E TERMINOLOGIA DOS <i>AEs</i> EM PROBLEMAS DE OTIMIZAÇÃO	27
2.5.1 ESPAÇO DE BUSCA	27
2.5.2 REPRESENTAÇÃO	28
2.6 ALGORITMOS GENÉTICOS	29
2.6.1 OPERADOR DE SELEÇÃO	32
2.6.2 TIPOS DE REPRESENTAÇÃO	36
2.6.2.1 ALGORITMO GENÉTICO CANÔNICO.....	36
2.6.2.1.1 OPERADOR DE CRUZAMENTO	38
2.6.2.1.2 OPERADOR DE MUTAÇÃO.....	39
2.6.2.2 ALGORITMO GENÉTICO COM REPRESENTAÇÃO EM PONTO FLUTUANTE.....	40
2.6.2.2.1 OPERADOR DE MUTAÇÃO.....	40
2.6.2.2.2 OPERADOR DE RECOMBINAÇÃO.....	41
2.6.3 CONFIGURAÇÃO DOS OPERADORES, SINTONIA E CONTROLE DOS PARÂMETROS NOS <i>AEs</i>	42

2.6.4	ABORDAGEM DE AJUSTE AUTOMÁTICO DAS PROBABILIDADES DE CRUZAMENTO E MUTAÇÃO..	44
2.6.5	ABORDAGEM COM PROBABILIDADE DE MUTAÇÃO VARIÁVEL	44
2.6.6	ABORDAGEM COM OPERADOR DE REPRODUÇÃO BASEADO EM GRADIENTE	45
2.7	ESTRATÉGIAS EVOLUTIVAS.....	45
2.7.1	MECANISMO DE AUTO-ADAPTAÇÃO ATRAVÉS DE MUTAÇÕES CORRELACIONADAS.....	48
2.8	PROGRAMAÇÃO EVOLUTIVA.....	53
2.8.1	ABORDAGEM DA <i>PE</i> COM O OPERADOR DE MUTAÇÃO DE CAUCHY	55
2.9	RESUMO DAS CARACTERÍSTICAS DOS <i>AEs</i>	56
2.10	<i>AEs</i> HÍBRIDOS COM METODOLOGIAS DE BUSCA LOCAL	57
2.10.1	<i>AE</i> HÍBRIDO COM <i>SIMULATED ANNEALING</i>	59
2.10.2	<i>AE</i> HÍBRIDO COM MÉTODO SIMPLEX	62
2.11	CONCLUSÃO DO CAPÍTULO.....	66
3.	SISTEMAS NEBULOSOS: CONCEITOS E PROJETOS.....	68
3.1	INTRODUÇÃO	68
3.2	BREVE HISTÓRICO	69
3.3	POR QUE UTILIZAR OS <i>SNs</i> ?.....	72
3.4	APLICAÇÕES INDUSTRIAIS	73
3.5	ESTRUTURA GERAL DOS SISTEMAS NEBULOSOS	73
3.6	MODELOS NEBULOSOS: ABORDAGENS E REVISÃO DA LITERATURA.....	74
3.6.1	TIPOS DE MODELOS NEBULOSOS.....	75
3.6.1.1	MODELO NEBULOSO LINGÜÍSTICO DO TIPO MAMDANI.....	76
3.6.1.2	MODELO NEBULOSO INTERPOLATIVO	77
3.6.1.2.1	MODELO INTERPOLATIVO DO TIPO TAKAGI-SUGENO-KANG	77
3.6.1.2.2	MODELO INTERPOLATIVO DO TIPO TSUKAMOTO	80
3.7	<i>SNs</i> EM IDENTIFICAÇÃO DE PROCESSOS.....	81
3.7.1	SISTEMA DE INFERÊNCIA NEBULOSO COM APRENDIZADO MODIFICADO DE EXEMPLOS.....	83
3.7.1.1	PASSOS DO ALGORITMO <i>MLFE</i> PARA IDENTIFICAÇÃO DE PROCESSOS	84
3.7.2	MODELO <i>TSK</i> COM <i>CLUSTERING</i> E MÍNIMOS QUADRADOS RECURSIVOS.....	85
3.8	CONTROLE LÓGICO NEBULOSO	87
3.8.1	BASE DE CONHECIMENTO.....	88
3.8.2	FASE DE NEBULIZAÇÃO (<i>FUZZIFICATION</i>)	88
3.8.3	MECANISMO DE INFERÊNCIA	89
3.8.4	FASE DE DESNEBULIZAÇÃO (<i>DEFUZZIFICATION</i>)	89
3.9	CONTROLE LÓGICO NEBULOSO MULTIVARIÁVEL	89
3.9.1	PROJETO <i>CLN-(PD+I)</i>	93
3.9.2	PROJETO <i>CLN-PI</i> NEBULOSO + (<i>D</i> CONVENCIONAL)	94

3.9.3 PROJETO <i>CLN-(PD+I)</i> INCREMENTAL	96
3.9.4 PROJETO <i>CLN-(PD+PI)</i>	96
3.10 CONCLUSÃO DO CAPÍTULO	97
4. REDES NEURAIS ARTIFICIAIS EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS	99
4.1 INTRODUÇÃO	99
4.2 BREVE HISTÓRICO	100
4.3 POR QUE UTILIZAR AS <i>RNs</i> ?	102
4.4 APLICAÇÕES INDUSTRIAIS	102
4.5 MODELO DE NEURÔNIO	103
4.6 CLASSIFICAÇÃO DAS <i>RNs</i>	105
4.7 <i>RN PERCEPTRON</i> FEEDFORWARD MULTICAMADAS	107
4.8 ALGORITMO DE TREINAMENTO <i>BP</i>	107
4.8.1 MÉTODO ADAPTATIVO DE <i>VOGL</i>	110
4.8.2 MÉTODO ADAPTATIVO DE ANGUITA-VOGL - ALGORITMO <i>YPROP</i>	111
4.8.3 MÉTODO DELTA-BAR-DELTA DE JACOBS	111
4.8.4 MÉTODOS DE TREINAMENTO DE SEGUNDA ORDEM	112
4.8.4.1 MÉTODO ADAPTATIVO DIRETO - ALGORITMO <i>RPROP</i>	114
4.8.4.2 MÉTODO ADAPTATIVO DE FAHLMAN - ALGORITMO <i>QUICKPROP</i>	117
4.9 <i>RN</i> COM <i>GROUP METHOD OF DATA HANDLING</i>	117
4.10 <i>RNs</i> RECORRENTES	120
4.10.1 <i>RN</i> PARCIALMENTE RECORRENTE DE ELMAN	121
4.11 <i>RN</i> FUNÇÃO DE BASE RADIAL	123
4.11.1 MÉTODOS DE CONFIGURAÇÃO	124
4.11.1.1 MÉTODOS CONSTRUTIVOS	125
4.11.1.1.1 APRENDIZADO VIA K-MÉDIAS E TÉCNICA DA PSEUDO-INVERSA	126
4.11.1.1.2 APRENDIZADO VIA ALGORITMO DOS MÍNIMOS QUADRADOS ORTOGONAL	128
4.11.2 DIFERENÇA ENTRE <i>RNs-RBF</i> E <i>RNs-MLP</i>	131
4.11.3 <i>RN-RBFx</i> MODELO NEBULOSO DE <i>TSK</i> : SIMILARIDADES E DIFERENÇAS	132
4.12 <i>RNs</i> EM IDENTIFICAÇÃO DE PROCESSOS MULTIVARIÁVEIS	133
4.12.1 PROCEDIMENTO PARA IDENTIFICAÇÃO DE PROCESSOS VIA <i>RNs</i>	133
4.12.1.1 SELEÇÃO DO SINAL DE TREINAMENTO	134
4.12.1.2 CÁLCULO DOS PARÂMETROS E MODELOS NEURAIS EM IDENTIFICAÇÃO	135
4.12.1.3 VALIDAÇÃO E SELEÇÃO DA CONFIGURAÇÃO DAS <i>RNs</i>	137
4.12.1.4 GENERALIZAÇÃO E SOBRE-TREINAMENTO	138
4.13 <i>RNs</i> EM CONTROLE DE PROCESSOS MULTIVARIÁVEIS	139
4.14 CONCLUSÃO DO CAPÍTULO	141

5. SISTEMAS HÍBRIDOS INTELIGENTES	143
5.1 INTRODUÇÃO	143
5.2 CLASSIFICAÇÃO DOS SISTEMAS HÍBRIDOS INTELIGENTES	145
5.3 COMBINAÇÃO DE SISTEMAS NEBULOSOS E ALGORITMOS EVOLUTIVOS	146
5.3.1 DEFINIÇÃO EVOLUTIVA DA BASE DE DADOS DOS <i>SNs</i>	148
5.3.2 DEFINIÇÃO EVOLUTIVA DA BASE DE REGRAS DOS <i>SNs</i>	149
5.3.3 APRENDIZADO EVOLUTIVO DA BASE DE CONHECIMENTO DOS <i>SNs</i>	149
5.4 COMBINAÇÃO DE REDES NEURAI E ALGORITMOS EVOLUTIVOS (<i>SHRNAE</i>)	149
5.4.1 EVOLUÇÃO DOS PESOS DAS <i>RNs</i>	150
5.4.2 EVOLUÇÃO DAS ARQUITETURAS DAS <i>RNs</i>	150
5.4.3 EVOLUÇÃO DAS REGRAS DE APRENDIZADO DAS <i>RNs</i>	151
5.4.4 OUTRAS FORMAS DE CONFIGURAÇÃO E APLICAÇÕES	152
5.5 COMBINAÇÃO DE SISTEMAS NEBULOSOS E REDES NEURAI (<i>SHSNRN</i>)	153
5.6 FORMAS ALTERNATIVAS DE COMBINAR <i>SNs</i> , <i>RNs</i> E <i>AEs</i> EM PROJETOS DE CONTROLE	154
5.7 CONCLUSÃO DO CAPÍTULO	156
6. CONTROLE PREDITIVO: ABORDAGEM ADAPTATIVA E INTELIGENTE	157
6.1 INTRODUÇÃO	157
6.2 BREVE HISTÓRICO	158
6.3 MOTIVAÇÃO À UTILIZAÇÃO DE <i>MPBC</i>	158
6.4 APLICAÇÕES INDUSTRIAIS	159
6.5 IDENTIFICAÇÃO DE PROCESSOS PELO ALGORITMO DOS MÍNIMOS QUADRADOS RECURSIVO	160
6.5.1 PROCEDIMENTOS DE APLICAÇÃO DOS <i>MQR</i>	164
6.6 CONTROLADOR PREDITIVO GENERALIZADO	164
6.6.1 <i>GPC</i> MULTIVARIÁVEL	166
6.6.1.1 ASPECTOS TEÓRICOS	166
6.7 <i>GPC</i> COM ABORDAGEM ADAPTATIVA	169
6.8 <i>GPC</i> HÍBRIDO COM INTELIGÊNCIA COMPUTACIONAL	170
6.8.1 <i>GPC</i> NEURAL	171
6.8.2 <i>GPC</i> NEBULOSO	173
6.8.3 <i>GPC</i> EVOLUTIVO	173
6.8.4 <i>GPC</i> COM SISTEMAS HÍBRIDOS INTELIGENTES	175
6.9 CONCLUSÃO DO CAPÍTULO	178

7. CONTROLE DE ESTRUTURA VARIÁVEL	180
7.1 INTRODUÇÃO	180
7.2 BREVE HISTÓRICO	181
7.3 MOTIVAÇÃO À UTILIZAÇÃO DO CONTROLE DE ESTRUTURA VARIÁVEL	182
7.4 APLICAÇÕES INDUSTRIAIS	183
7.5 CONTROLADOR <i>VSS</i> COM MODOS QUASE-DESLIZANTES (<i>QSMC</i>)	184
7.5.1 ASPECTOS TEÓRICOS	184
7.5.2 ATRIBUTOS E DEFINIÇÕES DO <i>QSMC</i>	186
7.5.3 PROBLEMA DO <i>CHATTERING</i> E SOLUÇÕES PROPOSTAS NA LITERATURA	187
7.5.4 <i>QSMC</i> COMBINADO A CONTROLE DE VARIÂNCIA MÍNIMA GENERALIZADA	188
7.5.4.1 ABORDAGEM ADAPTATIVA DO <i>SMC</i> COM A IDENTIFICAÇÃO VIA <i>MQR</i>	191
7.6 <i>SMC</i> HÍBRIDO COM INTELIGÊNCIA COMPUTACIONAL	192
7.6.1 PROJETO <i>VSS</i> NEURAL	193
7.6.2 PROJETO <i>VSS</i> NEBULOSO	194
7.6.3 PROJETO <i>VSS</i> EVOLUTIVO	195
7.7 CONCLUSÃO DO CAPÍTULO	196
8. FUNDAMENTOS PRÁTICOS E APLICAÇÕES EM IDENTIFICAÇÃO DE PROCESSOS ...	197
8.1 INTRODUÇÃO	197
8.2 IDENTIFICAÇÃO DE PROCESSOS: FUNDAMENTOS E OBJETIVOS	197
8.3 ASPECTOS RELEVANTES EM IDENTIFICAÇÃO DE PROCESSOS	201
8.4 EXPERIMENTAÇÃO DO SISTEMA E ESCOLHA DO TEMPO DE AMOSTRAGEM	202
8.5 DETECÇÃO DE NÃO-LINEARIDADES	203
8.6 PROJETO DO SINAL DE EXCITAÇÃO	204
8.7 ESCOLHA DE ESTRUTURA E FORMAS DE REPRESENTAÇÃO	205
8.8 SELEÇÃO DE ORDEM E ESTIMAÇÃO DE PARÂMETROS	206
8.9 IDENTIFICAÇÃO E VALIDAÇÃO DE MODELOS	206
8.9.1 VALIDAÇÃO ATRAVÉS DE ANÁLISE DE AUTOCORRELAÇÃO E CORRELAÇÃO CRUZADA	207
8.10 CRITÉRIOS DE DESEMPENHO	208
8.11 FUNDAMENTOS DA CONSTRUÇÃO DE MODELOS <i>NARMAX</i>	209
8.12 APLICAÇÃO DE IDENTIFICAÇÃO EM MALHA FECHADA	210
8.12.1 ESTUDO DE CASO 1: IDENTIFICAÇÃO <i>MISO</i> DO PROCESSO BALANÇO HORIZONTAL	210
8.12.1.1 IDENTIFICAÇÃO <i>MISO</i> DO <i>PBH</i> VIA <i>AEs</i>	212
8.12.1.2 IDENTIFICAÇÃO <i>MISO</i> DO <i>PBH</i> VIA <i>SNs</i>	217
8.12.1.3 IDENTIFICAÇÃO <i>MISO</i> DO <i>PBH</i> VIA <i>RNs</i>	220
8.12.1.4 ANÁLISE DOS RESULTADOS PARA O ESTUDO DE CASO 1	228
8.12.2 ESTUDO DE CASO 2: IDENTIFICAÇÃO DE UM MANIPULADOR ROBÓTICO	230

8.12.2.1 IDENTIFICAÇÃO DO MANIPULADOR ROBÓTICO	233
8.13 APLICAÇÃO DE IDENTIFICAÇÃO EM MALHA ABERTA	248
8.13.1 ESTUDO DE CASO 3: IDENTIFICAÇÃO DE PROCESSO <i>MIMO</i>	238
8.14 CONCLUSÃO DO CAPÍTULO	243
9. APLICAÇÕES EM CONTROLE DE PROCESSOS	247
9.1 INTRODUÇÃO	247
9.2 CONTROLE DE UM MANIPULADOR ROBÓTICO COM DOIS GRAUS DE LIBERDADE	248
9.3 PROJETO DE CONTROLDE <i>PD</i> E <i>PID</i> COM SINTONIA VIA <i>AEs</i>	249
9.3.1 PROJETO <i>PD</i> MULTIVARIÁVEL DESCENTRALIZADO	249
9.3.2 PROJETO <i>PID</i> MULTIVARIÁVEL DESCENTRALIZADO	250
9.3.3 PROJETOS <i>PD</i> E <i>PID</i> MULTIVARIÁVEIS CENTRALIZADOS	251
9.3.4 RESULTADOS DE CONTROLE	251
9.4 PROJETOS DE CONTROLADORES NEBULOSOS, NEURAI E HÍBRIDOS INTELIGENTES	261
9.5 PROJETOS DE <i>GPC</i> E <i>QSMC</i> COM ABORDAGENS ADAPTATIVAS E INTELIGENTES	273
9.6 CONCLUSÃO DO CAPÍTULO	282
10. CONCLUSÃO E PERSPECTIVAS	287
10.1 INTRODUÇÃO	287
10.1.1 IDENTIFICAÇÃO DE PROCESSOS MULTIVARIÁVEIS	288
10.1.2 CONTROLE DE PROCESSOS MULTIVARIÁVEIS	290
10.2 PRINCIPAIS CONTRIBUIÇÕES	292
10.3 PERSPECTIVAS	293
10.3.1 ÁREA DE ALGORITMOS EVOLUTIVOS	293
10.3.2 ÁREA DE SISTEMAS NEBULOSOS	294
10.3.3 ÁREA DE REDES NEURAI	295
10.3.4 ÁREA DE SISTEMAS HÍBRIDOS INTELIGENTES	296
10.3.5 ÁREA DE CONTROLE PREDITIVO	296
10.3.6 ÁREA DE CONTROLE DE ESTRUTURA VARIÁVEL	298
APÊNDICE A – ESTADO DA ARTE EM PROJETO INTELIGENTE E PREDITIVO	299
REFERÊNCIAS BIBLIOGRÁFICAS	307

LISTA DE FIGURAS

2.1. RELAÇÕES ENTRE C , S E R .	29
2.2. PSEUDOCÓDIGO BÁSICO DOS AGs	31
2.3. ALGORITMO DE SELEÇÃO PROPORCIONAL	34
2.4. ALGORITMO DE SELEÇÃO COM TRUNCAMENTO	35
2.5. ALGORITMO DE SELEÇÃO ATRAVÉS DE TORNEIO	35
2.6. ALGORITMO GENÉTICO CANÔNICO	37
2.7. OPERADOR DE CRUZMENTO COM UM OU VÁRIOS PONTOS DE CORTE	39
2.8. OPERADOR DE CRUZMENTO UNIFORME	39
2.9. OPERADOR DE MUTAÇÃO NOS AGs CANÔNICOS	40
2.10. EFEITO DA OPERAÇÃO DE MUTAÇÃO DESCORRELACIONADA E CORRELACIONADA EM EES	49
2.11. PSEUDOCÓDIGO BÁSICO DA PE	54
2.12. DIFERENÇA ENTRE AS FUNÇÕES DENSIDADE DE CAUCHY E GAUSSIANA	56
2.13. FLUXO BÁSICO DOS DADOS NO ALGORITMO SA	61
2.14. PONTOS GERADOS PELO SIMPLEX DE NELDER & MEAD EM DUAS DIMENSÕES	63
3.1. ESTRUTURA GERAL DE INFERÊNCIA NOS SNs	74
3.2. EXEMPLO DO MODELO NEBULOSO DE MAMDANI	77
3.3. EXEMPLO DO MODELO NEBULOSO DE TSK	79
3.4. PASSOS PARA IDENTIFICAÇÃO DE UM MODELO NEBULOSO DO TIPO TSK	80
3.5. EXEMPLO DO MODELO NEBULOSO DE TSUKAMOTO	81
3.6. ESQUEMA DE CONFIGURAÇÃO DE IDENTIFICAÇÃO NEBULOSA E SUAS ESTRUTURAS	82
3.7. CONFIGURAÇÃO BÁSICA DE CLN	88
3.8. ESQUEMA DE CLN MULTIVARIÁVEL	90
3.9. ELEMENTOS ESTRUTURAIS PARA CONFIGURAÇÃO DE CLN DO TIPO PID	92
3.10. DIAGRAMA BÁSICO DO $CLN-(PD+I)$	93
3.11. DIAGRAMA BÁSICO DO $CLN-PI$ NEBULOSO + (D CONVENCIONAL)	95
3.12. DIAGRAMA BÁSICO DO $CLN-(PD+I)$ INCREMENTAL	96
3.13. DIAGRAMA BÁSICO DO $CLN-(PD+PI)$	97
4.1. ESTRUTURA DO NEURÔNIO ARTIFICIAL	104
4.2. TAXIONOMIA DAS RNs $FEEDFORWARD$ E $FEEDBACK$	106
4.3. $RN-MLP$ COM TREINAMENTO BP .	109
4.4. ALGORITMO $YPROP$ HEURÍSTICO PARA O TREINAMENTO DE UMA RN	110
4.5. ALGORITMO $RPROP$ PARA O TREINAMENTO DE UMA RN	116
4.6. $ADALINE$ COM PRÉ-PROCESSADOR NÃO-LINEAR	119
4.7. DIAGRAMA ESQUEMÁTICO DA $RNPREM$	122

4.8. UMA <i>RN-RBF</i> TÍPICA COM UMA CAMADA INTERMEDIÁRIA	124
4.9. REPRESENTAÇÃO DE PROCESSOS <i>MIMO</i> (MODELO I)	136
4.10. REPRESENTAÇÃO DE PROCESSOS <i>MIMO</i> (MODELO II)	136
4.11. REPRESENTAÇÃO DE PROCESSOS <i>MIMO</i> (MODELO III)	136
4.12. REPRESENTAÇÃO DE PROCESSOS <i>MIMO</i> (MODELO IV)	137
4.13. CLASSIFICAÇÃO DOS ESQUEMAS DE CONTROLE NEURAL, BASEADA NA FORMA EM QUE O SINAL DE ERRO É UTILIZADO, PARA O TREINAMENTO DA <i>RN</i>	140
4.14. CONTROLE ÓTIMO NEURAL	141
5.1. SISTEMA DE DEFINIÇÃO EVOLUTIVA DA BASE DE DADOS NOS <i>SNs</i>	148
5.2. ESTRUTURA GERAL PARA SISTEMAS HÍBRIDOS DE <i>RNs</i> E <i>AEs</i>	150
5.3. PROCEDIMENTO DE PROJETO EVOLUTIVO DE <i>RNs</i>	151
5.4. POSSIBILIDADES DE PROJETO HÍBRIDO EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS ATRAVÉS DE <i>IC</i>	155
6.1. DEFINIÇÃO DO PROBLEMA DE OTIMIZAÇÃO PARA O <i>MBPC</i>	165
6.2. CONFIGURAÇÃO DE <i>GPC</i> ADAPTATIVO	170
6.3. CONTROLE PREDITIVO NEURAL (TIPO I)	172
6.4. CONTROLE PREDITIVO NEURAL (TIPO II)	172
6.5. DIAGRAMA DO ESQUEMA DE CONFIGURAÇÃO DO <i>GPC</i> BASEADO EM <i>RN-RBF</i> E OTIMIZAÇÃO EVOLUTIVO-SIMPLEX.	176
7.1. EXEMPLO DA TRAJETÓRIA DE UM SISTEMA <i>VSS</i>	185
7.2. EXEMPLOS DE FUNÇÕES DE CHAVEAMENTO	188
8.1. PROCEDIMENTO PARA IDENTIFICAÇÃO DE PROCESSOS	202
8.2. REPRESENTAÇÃO DA CONFIGURAÇÃO DO <i>PBH</i>	211
8.3. SINAIS DAS ENTRADAS, REFERÊNCIA E SAÍDA REAL DO <i>PBH</i> , EM MALHA FECHADA	212
8.4. CONFIGURAÇÃO DA IDENTIFICAÇÃO DO <i>PBH</i>	212
8.5. REPRESENTAÇÃO DA CONFIGURAÇÃO DE IDENTIFICAÇÃO DE MODELOS LINEARES <i>ARX</i> E <i>ARMAX</i> PARA O <i>PBH</i> VIA <i>AEs</i>	213
8.6. ANÁLISE COMPARATIVA DE IDENTIFICAÇÃO <i>ARMAX</i> DO <i>PBH</i> VIA <i>AEs</i>	215
8.7. RESULTADO DA IDENTIFICAÇÃO <i>ARMAX</i> DO <i>PBH</i> VIA <i>AEs</i>	217
8.8. ANÁLISE COMPARATIVA DA IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>SN-MLFE</i>	218
8.9. RESULTADOS DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>SN-MLFE</i>	220
8.10. EVOLUÇÃO DO APRENDIZADO DE MÉTODOS HEURÍSTICOS DE TREINAMENTO DE <i>RNs-MLP</i> PARA IDENTIFICAÇÃO DO <i>PBH</i>	222
8.11. RESULTADOS DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-MLP</i>	223
8.12. EVOLUÇÃO DO APRENDIZADO DE MÉTODOS DE 2 ^a ORDEM DE TREINAMENTO DE <i>RNs-MLP</i> PARA IDENTIFICAÇÃO DO <i>PBH</i>	225

8.13. RESULTADOS DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-MLP</i> (MÉTODOS DE 2A. ORDEM)	226
8.14. RESULTADOS DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-GMDH</i> E <i>RN-PREM</i>	226
8.15. ANÁLISE COMPARATIVA DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-RBF(KM)</i>	227
8.16. RESULTADO DE IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-RBF(KM)</i> E <i>RN-RBF(OLS)</i>	228
8.17. ANÁLISE DE AUTO-CORRELAÇÃO E CORRELAÇÃO CRUZADA	229
8.18. EXEMPLO DA OCORRÊNCIA DE SOBRETREINAMENTO EM <i>RN-MLP</i>	230
8.19. REPRESENTAÇÃO DO MANIPULADOR ROBÓTICO COM DOIS GRAUS DE LIBERDADE	231
8.20. TORQUES E AS SAÍDAS DE POSIÇÃO E VELOCIDADE DO MANIPULADOR ROBÓTICO	233
8.21. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>RN-MLP(QUICKPROP)</i>	234
8.22. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>RN-MLP(SGC)</i>	235
8.23. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>RN-MLP(QN)</i>	235
8.24. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>RN-RBF(KM)</i>	236
8.25. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>RN-RBF(OLS)</i>	236
8.26. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>SHSNRN(TSK)</i>	237
8.27. POSIÇÕES E VELOCIDADES ESTIMADAS DO ROBÔ VIA <i>SHRNAE</i>	237
8.28. DADOS DE ENTRADA E SAÍDA DO PROCESSO <i>MIMO</i> NÃO-LINEAR	239
8.29. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>RN-MLP(QUICKPROP)</i>	241
8.30. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>RN-MLP(SCG)</i>	241
8.31. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>RN-MLP(QN)</i>	241
8.32. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>RN-RBF(KM)</i>	242
8.33. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>RN-RBF(OLS)</i>	242
8.34. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>SHSNRN(TSK)</i>	242
8.35. RESULTADOS DE IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>SHRNAE</i>	243
9.1. ESQUEMAS DE CONTROLE <i>PD</i> E <i>PID</i> COM SINTONIA VIA <i>AEs</i>	252
9.2. ANÁLISE COMPARATIVA DA SINTONIA VIA <i>AEs</i> DOS GANHOS DE CONTROLE <i>PD</i> E <i>PID</i>	253
9.3. ANÁLISE COMPARATIVA DA FUNÇÃO OBJETIVO, <i>J</i> , DOS MELHORES RESULTADOS DOS <i>AEs</i> NA SINTONIA DOS GANHOS DE CONTROLE <i>PD</i> E <i>PID</i>	254
9.4. ANÁLISE COMPARATIVA DA VARIÂNCIA DO CONTROLE DOS MELHORES RESULTADOS NA SINTONIA VIA <i>AEs</i> DOS GANHOS DE CONTROLE <i>PD</i> E <i>PID</i>	255
9.5. CONTROLE <i>PD</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO SERVO) VIA <i>AG(8)</i>	256
9.6. CONTROLE <i>PID</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO SERVO) VIA (1+29)- <i>EEMC</i>	256
9.7. CONTROLE <i>PID-AP</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO SERVO) VIA (1+29)- <i>EEMC</i>	257
9.8. CONTROLE <i>PID-AV</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO SERVO) VIA <i>AGSA(1)</i>	257
9.9. CONTROLE <i>PID-APV</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO SERVO) VIA (1+29)- <i>EEMC</i>	258

9.10. CONTROLE <i>PD</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO) VIA <i>AG</i> (8)	258
9.11. CONTROLE <i>PID</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO) VIA (1+29)- <i>EEMC</i>	259
9.12. CONTROLE <i>PID-AP</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO) VIA (1+29)- <i>EEMC</i>	259
9.13. CONTROLE <i>PID-AV</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO) VIA <i>AGSA</i> (1)	260
9.14. CONTROLE <i>PID-APV</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO) VIA (1+29)- <i>EEMC</i>	260
9.15. ESQUEMA DE CONTROLE NEBULOSO <i>MULTILOOP</i>	262
9.16. ESQUEMA DE CONTROLE NEURAL <i>RN-RBF</i> COM OTIMIZAÇÃO ATRAVÉS DE <i>AGSA</i>	263
9.17. RESULTADOS DE CONTROLE NEBULOSO, NEURAL E HÍBRIDO INTELIGENTE	263
9.18. <i>CLN-(PD+I)</i> (COMPORTAMENTO SERVO)	264
9.19. <i>CLN-PI</i> NEBULOSO + (D CONVENCIONAL) (COMPORTAMENTO SERVO)	265
9.20. <i>CLN-(PD+I)</i> INCREMENTAL (COMPORTAMENTO SERVO)	265
9.21. <i>CLN-(PD+PI)</i> (COMPORTAMENTO SERVO)	266
9.22. CONTROLE NEURAL (COMPORTAMENTO SERVO)	266
9.23. CONTROLE ATRAVÉS DE <i>SHI</i> COMBINANDO <i>CLN-(PD+I)</i> INCREMENTAL COM AJUSTE DOS FATORES DE ESCALA VIA <i>EEMC</i> (COMPORTAMENTO SERVO)	267
9.24. CONTROLE ATRAVÉS DE <i>SHI</i> NEURAL COM OTIMIZAÇÃO EVOLUTIVA (COMPORTAMENTO SERVO)	267
9.25. CONTROLE ATRAVÉS DE <i>SHI</i> COM ESTRUTURA NEURO-NEBULOSO-EVOLUTIVA (COMPORTAMENTO SERVO).	268
9.26. <i>CLN-(PD+I)</i> DO MANIPULADOR ROBÓTICO (COMPORTAMENTO REGULATÓRIO)	268
9.27. <i>CLN-PI</i> NEBULOSO + (D CONVENCIONAL) (COMPORTAMENTO REGULATÓRIO)	269
9.28. <i>CLN-(PD+I)</i> INCREMENTAL (COMPORTAMENTO REGULATÓRIO)	269
9.29. <i>CLN-(PD+PI)</i> (COMPORTAMENTO REGULATÓRIO)	270
9.30. CONTROLE NEURAL (COMPORTAMENTO REGULATÓRIO)	270
9.31. CONTROLE ATRAVÉS DE <i>SHI</i> COMBINANDO <i>CLN-(PD+I)</i> INCREMENTAL COM AJUSTE DOS FATORES DE ESCALA VIA <i>EEMC</i> (COMPORTAMENTO REGULATÓRIO)	271
9.32. CONTROLE ATRAVÉS DE <i>SHI</i> NEURAL COM OTIMIZAÇÃO EVOLUTIVA (COMPORTAMENTO REGULATÓRIO)	271
9.33. CONTROLE ATRAVÉS DE <i>SHI</i> COM ESTRUTURA NEURO-NEBULOSO-EVOLUTIVA (COMPORTAMENTO REGULATÓRIO)	272
9.34. ANÁLISE DOS RESULTADOS DE ESTRUTURAS DE <i>GPC</i> E <i>QSMC</i>	274
9.35. COMPORTAMENTO SERVO DO <i>GPC</i> ADAPTATIVO (MODELO Nº 1)	275

9.36. COMPORTAMENTO SERVO <i>GPC</i> ADAPTATIVO (MODELO Nº 2)	275
9.37. COMPORTAMENTO SERVO DO <i>QSMC</i> ADAPTATIVO (MODELO Nº 1)	276
9.38. COMPORTAMENTO SERVO DO <i>QSMC GPC</i> ADAPTATIVO (MODELO Nº 2)	276
9.39. COMPORTAMENTO SERVO DO <i>GPC</i> BASEADO EM MODELO <i>TSK</i>	277
9.40. COMPORTAMENTO SERVO DO <i>GPC</i> BASEADO EM <i>RN-RBF(KM)</i>	277
9.41. COMPORTAMENTO SERVO DO <i>GPC</i> BASEADO EM <i>RN-RBF</i> E OTIMIZAÇÃO VIA <i>PE</i> E MÉTODO SIMPLEX	278
9.42. COMPORTAMENTO REGULATÓRIO DO <i>GPC</i> ADAPTATIVO (MODELO Nº 1)	278
9.43. COMPORTAMENTO REGULATÓRIO DO <i>GPC</i> ADAPTATIVO (MODELO Nº 2)	279
9.44. COMPORTAMENTO REGULATÓRIO DO <i>QSMC</i> (MODELO Nº 1)	279
9.45. COMPORTAMENTO REGULATÓRIO DO <i>QSMC</i> ADAPTATIVO (MODELO Nº 2)	280
9.46. COMPORTAMENTO REGULATÓRIO DO <i>GPC</i> BASEADO EM MODELO <i>TSK</i>	280
9.47. COMPORTAMENTO REGULATÓRIO DO <i>GPC</i> BASEADO EM <i>RN-RBF(KM)</i>	281
9.48. COMPORTAMENTO REGULATÓRIO DO <i>GPC</i> BASEADO EM <i>RN-RBF</i> E OTIMIZAÇÃO VIA <i>PE</i> E MÉTODO SIMPLEX	281

LISTA DE TABELAS

1.1. CLASSIFICAÇÃO DE CONTROLADORES CONVENCIONAIS E AVANÇADOS	3
2.1. CARACTERÍSTICAS DOS <i>AEs</i>	57
4.1. COMPARAÇÃO ENTE UMA <i>RN-RBF</i> E UMA <i>RN-MLP</i> CONVENCIONAL	131
5.1. AVALIAÇÃO DE ALGUMAS CARACTERÍSTICAS DOS SISTEMAS INTELIGENTES	144
8.1. RESULTADOS DA IDENTIFICAÇÃO DO <i>PBH</i> — MODELO <i>ARMAX</i> — VIA <i>AEs</i>	214
8.2. RESULTADOS DA IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-MLP</i> (MÉTODOS HEURÍSTICOS)	221
8.3. RESULTADOS DA IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-MLP</i> (MÉTODOS DE 2ª ORDEM)	224
8.4. RESULTADOS DA IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RN-PREM</i> E <i>RN-GMDH</i>	226
8.5. RESULTADOS DA IDENTIFICAÇÃO DO <i>PBH</i> VIA <i>RBF(KM)</i> E <i>RN-RBF(OLS)</i>	228
8.6. RESULTADOS DA IDENTIFICAÇÃO DO PROCESSO <i>MIMO</i> VIA <i>IC</i>	240
9.1. PARÂMETROS DE SINTONIA VIA <i>AEs</i> PARA OS PROJETOS DE CONTROLE <i>PD</i> E <i>PID</i>	254
A.1. APLICAÇÕES DOS <i>AEs</i> EM AMBIENTES INDUSTRIAL E COMERCIAL	299
A.2. <i>HARDWARE</i> E <i>SOFTWARE</i> COM A UTILIZAÇÃO DE <i>AEs</i>	300
A.3. APLICAÇÕES DOS <i>SNs</i> NA INDÚSTRIA	300
A.4. <i>HARDWARE</i> E <i>SOFTWARE</i> COM A UTILIZAÇÃO DE <i>SNs</i>	301
A.5. APLICAÇÕES DAS <i>RNs</i> NA INDÚSTRIA	303
A.6. <i>SOFTWARE</i> COM A UTILIZAÇÃO DE <i>RNs</i>	304
A.7. ÁREAS DE CLASSIFICAÇÃO DE APLICAÇÕES DA COMBINAÇÃO DE <i>SNs</i> E <i>AGs</i>	305
A.8. RESUMO DE APLICAÇÕES COMERCIAIS DE MBPC UTILIZANDO MODELOS LINEARES	305
A.9. RESUMO DE APLICAÇÕES DE MBPCs UTILIZANDO MODELOS NÃO-LINEARES	306

definição na pág.	símbolo	definição
160	<i>MQR</i>	mínimos quadrados recursivo
184	<i>MR</i>	modo de regime
158	<i>MRAC</i>	controle adaptativo por modelo de referência
158	<i>MUSHAC</i>	<i>Multipredictor Receding Horizon Adaptive Control</i>
158	<i>MUSMAR</i>	<i>Multistep Multivariable Adaptive Control</i>
134	<i>NARMAX</i>	<i>Non-linear AutoRegressive Moving Average model structure with eXogenous inputs</i>
134	<i>NARX</i>	<i>Non-linear AutoRegressive model structure eXogenous inputs</i>
134	<i>NFIR</i>	<i>Non-linear Finite Impulse Response model</i>
26	<i>NFL</i>	teorema <i>no free lunch</i>
134	<i>NOE</i>	<i>Non-linear Output Error model</i>
210	<i>PBH</i>	processo balanço horizontal
53	<i>PE</i>	programação evolutiva
55	<i>PE-L</i>	<i>PE</i> com operador de mutação lognormal
55	<i>PE-C</i>	<i>PE</i> com operador de mutação com distribuição de Cauchy
62	<i>PESA</i>	<i>PE</i> híbrida com <i>SA</i>
158	<i>PFC</i>	<i>Predictive Functional Control</i>
24	<i>PG</i>	programação genética
1	<i>PI</i>	proporcional-integral
1	<i>PID</i>	proporcional-integral-derivativo
250	<i>PID-AP</i>	controle <i>PID</i> com termos acoplados de posição
251	<i>PID-APV</i>	controle <i>PID</i> com termos acoplados de posição e velocidade
251	<i>PID-AV</i>	controle <i>PID</i> com termos acoplados de velocidade
184	<i>QSMC</i>	modos quase-deslizantes
117	<i>QuickProp</i>	método adaptativo de Fahlman
8	<i>RN</i>	rede neural
117	<i>RN-GMDH</i>	rede neural polinomial do tipo <i>Group Method of Data Handling</i>
9	<i>RN-MLP</i>	rede neural <i>perceptron feedforward</i> multicamadas
221	<i>RN-MLP(JACOBS)</i>	<i>RN-MLP</i> com método delta-bar-delta de Jacobs
221	<i>RN-MLP(VOGL)</i>	<i>RN-MLP</i> com método adaptativo de Vogl
221	<i>RN-MLP(YPROP)</i>	<i>RN-MLP</i> com método adaptativo de Anguita e Vogl
208	R_n^2	coeficiente de correlação múltipla
121	<i>RN-PRE</i>	rede neural parcialmente recorrente de Elman
122	<i>RN-PREM</i>	<i>RNPRE</i> modificada
123	<i>RN-RBF</i>	<i>RN</i> função de base radial
8	<i>RNs</i>	redes neurais artificiais
114	<i>RPROP</i>	<i>resilient propagation</i>
59	<i>SA</i>	<i>simulated annealing</i>
6	<i>SC</i>	<i>soft computing</i>
114	<i>SGC</i>	método gradiente conjugado escalar
143	<i>SHI</i>	sistema híbrido inteligente
147	<i>SHSNAE</i>	sistemas híbridos nebulosos-evolutivos
144	<i>SHSNRN</i>	combinação de sistemas nebulosos e redes neurais
149	<i>SHSNRNAE</i>	combinação de sistemas nebulosos, redes neurais e algoritmos evolutivos
180	<i>SMC</i>	controle por modos deslizantes
68	<i>SNs</i>	sistemas nebulosos
89	<i>SOM</i>	primeiro máximo
208	<i>SSE</i>	<i>Sum of Squared Error</i>
77	<i>TSK</i>	Takagi-Sugeno-Kang
158	<i>UPC</i>	<i>Unified Predictive Control</i>
181	<i>VSS</i>	estrutura variável

SIMBOLOGIA

pág. da definição	símbolo	definição
19	<i>AEs</i>	algoritmos evolutivos
48	<i>AEMAs</i>	algoritmos evolutivos com mecanismos adaptativos
29	<i>AGs</i>	algoritmos genéticos
62	<i>AGSA</i>	<i>AG</i> híbrido com <i>SA</i>
134	<i>ARMAX</i>	<i>AutoRegressive Moving Average model structure with eXogenous inputs</i>
134	<i>ARX</i>	<i>AutoRegressive model structure eXogenous inputs</i>
134	<i>BJ</i>	<i>Non-linear Box-Jenkins model structure</i>
107	<i>BP</i>	<i>back-propagation</i>
166	<i>CARIMA</i>	<i>Controlled Auto-Regressive Integrated Moving Average</i>
166	<i>CARMA</i>	<i>Controlled Auto-Regressive Moving Average</i>
19	<i>CE</i>	computação evolucionária ou evolutiva
69	<i>CLN</i>	controle lógico nebuloso
93	<i>CLN-(PD+I)</i>	controle lógico nebuloso do tipo proporcional-derivativo mais a parte integral
96	<i>CLN-(PD+I)</i> incremental	controle lógico nebuloso do tipo proporcional-derivativo mais ação integral nebulosa
96	<i>CLN-(PD+PI)</i>	controle lógico nebuloso do tipo proporcional-derivativo mais proporcional-integral
90	<i>CLN-PD</i>	controle lógico nebuloso do tipo proporcional-derivativo
90	<i>CLN-PI</i>	controle lógico nebuloso do tipo proporcional-integral
94	<i>CLN-PI</i> nebuloso + (D convencional)	controle lógico nebuloso do tipo proporcional-integral mais ação derivativa do <i>PID</i> convencional
91	<i>CLN-PID</i>	controle lógico nebuloso do tipo proporcional-integral-derivativo
89	<i>COA</i>	centro de área
158	<i>DMC</i>	<i>Dynamic Matrix Control</i>
50	<i>EEMC</i>	<i>EE</i> com tamanho de passo individual e mutações correlacionadas
45	<i>EEs</i>	estratégias evolutivas
62	<i>EESA</i>	<i>EE</i> híbrida com <i>SA</i> (<i>EESA</i>)
158	<i>EHAC</i>	<i>Extended Horizon Adaptive Control</i>
118	<i>EP</i>	<i>Annual Conference on Evolutionary Programming</i>
158	<i>EPSAC</i>	<i>Extended Prediction Self-Adaptive Control</i>
113	<i>GC</i>	método gradiente conjugado
117	<i>GMDH</i>	<i>Group Method of Data Handling</i>
165	<i>GMV</i>	controlador de variância mínima generalizada
158	<i>GPC</i>	controlador preditivo generalizado
6	<i>HC</i>	<i>hard computing</i>
4	<i>IC</i>	inteligência computacional
158	<i>IDCOM</i>	<i>Identification and COMmand</i>
227	<i>KM</i>	treinamento via <i>k</i> -médias e pseudo-inversa
113	<i>LM</i>	método de otimização de Levenberg-Marquardt
158	<i>MAC</i>	<i>Model Algorithmic Control</i>
184	<i>MD</i>	modo deslizante
184	<i>MH</i>	modo de aproximação da superfície
9	<i>MIMO</i>	múltiplas entradas e múltiplas saídas
13	<i>MISO</i>	múltiplas entradas e uma saída
83	<i>MLFE</i>	geração de regras através de aprendizado modificado
89	<i>MOM</i>	média dos máximos
157	<i>MPBC</i>	controle preditivo baseado em modelo

Capítulo 1

INTRODUÇÃO

1.1 Motivação e relevância do trabalho

A teoria convencional de controle — Nyquist, Bode, espaço de estados, controle ótimo, lugar das raízes — é adequada quando direcionada ao projeto de algoritmos para o tratamento de processos lineares. Além disto, os métodos convencionais baseiam-se na linearidade e comportamento invariante no tempo de processos a serem controlados.

Na teoria convencional de controle, os objetivos de projeto do controlador são fixos e definidos pelo projetista. Esta abordagem é limitada pelas situações do mundo real, devido ao fato que sistemas complexos requerem algumas formas de autonomia. Estas complexidades caracterizam processos com modelos matemáticos “pobres”, elevados níveis de ruído, rigorosos índices de desempenho, presença de não-linearidades, sensores distribuídos e a necessidade da tomada de decisões.

Os pesquisadores da teoria convencional de controle têm interesse em controle não linear, mas a teoria de controle não linear, na maioria das vezes mais eficiente, é considerada um tópico avançado, usualmente de difícil compreensão e com complexidades à implementação prática (SCHWARTZ *et al.*, 1994). Alguns trabalhos preliminares sobre a utilização de algoritmos de controle avançado e teoria de controle não linear são apresentados em COELHO & COELHO (1997f, 1998b), SIMAS *et al.* (1998) e BOL *et al.* (1999a, 1999b, 1999c). Neste caso são tratadas aplicações em processos não lineares monovariáveis e aspectos de educação em controle.

A grande parte dos problemas de controle no meio industrial podem ser resolvidos com controladores clássicos simples do tipo *PI* (proporcional-integral) ou *PID* (proporcional-integral-derivativo). Entretanto, algumas malhas de controle, principalmente da indústria química, podem beneficiar-se de técnicas de controle avançado, pois os métodos clássicos apresentam limitações quando aplicados a processos

apresentando características complexas, tais como: não-linearidades, instabilidade, atraso de transporte não-unitário e comportamento não-estacionário (COELHO *et al.*, 1998a, 1999d).

O termo *controle avançado* apresenta várias interpretações. Nos últimos trinta anos, muito tem-se escrito sobre controle avançado, enfatizando-se aspectos teóricos, estudos de implementações, declarações sobre os benefícios que suas aplicações trarão e as projeções de tendências futuras. Durante os anos 60, controle avançado foi vinculado ao significado de algum algoritmo ou estratégia que desvia-se do controle clássico *PID* (ÅSTRÖM, 1996, BUSHNELL, 1996).

O advento dos computadores em controle de processos implicou em algoritmos que não poderiam ser configurados e aplicados, utilizando-se a tecnologia analógica. As filosofias de controle *feedforward*, controle multivariável e controle ótimo tornaram-se alternativas praticáveis. A proliferação das denominadas metodologias de controle avançado pode ser atribuída aos avanços realizados na indústria eletrônica, especialmente no desenvolvimento de dispositivos computacionais de baixo custo (anos 70). Atualmente, controle avançado está intrinsecamente ligado a implementação de tecnologias baseadas em computadores (WILLIS & THAM, 1994).

O que os usuários denominam de “avançado” é uma estratégia de controle dependente do marco de referência adotado. Na década de 40, se consideravam controladores avançados, os controladores que hoje em dia são classificados como controladores convencionais ou clássicos, entre os quais, controle em cascata, *feedforward* e compensadores avanço-atraso. Durante a década de 60, controle avançado significava algum algoritmo ou estratégia que difere do controlador *PID* clássico.

O que é exatamente controle avançado? Esta é uma questão relevante e dependendo do conhecimento individual, controle avançado pode significar diferentes abordagens. Controle avançado pode ser a implementação de estratégias de controle *feedforward* ou em cascata, de compensadores para o tratamento de atraso de transporte, de algoritmos auto-ajustáveis, adaptativos ou a configuração de estratégias de otimização em projetos de controle. Sob este ponto, a visão de engenheiros acadêmicos e práticos pode diferir significativamente. WILLIS & THAM (1994) preferem considerar controle avançado a utilização de computadores com múltiplos processadores ou o estado da arte

de ambientes computacionais. Neste caso, nenhuma destas abordagens refere-se a utilização singular de algoritmos sofisticados de controle.

O projeto de algoritmos avançados, sob outro ponto de vista, descreve usualmente a combinação de elementos de muitas disciplinas, tais como engenharia de controle, processamento de sinais, estatística, teoria de decisão, inteligência computacional, entre outras. Muitos especialistas em controle denominam controladores avançados quando referem-se a controladores adaptativos, não lineares, preditivos e inteligentes (TOVAR, 1996). A tabela 1.1 apresenta uma classificação de controle convencional e avançado, segundo UNBEHAUEN (1996), baseado na aceitação de controladores na indústria, predominantemente vinculado à indústria química.

Tabela 1.1. Classificação de controladores convencionais e avançados.

projeto	tipo de controle
convencional	manual, <i>PID</i> , proporcional, em cascata e <i>feedforward</i>
avançado: <i>clássico</i>	ganho escalonado, compensação do atraso de transporte, controle desacoplado
avançado: <i>com aplicações</i>	preditivo, modelo interno, adaptativo
avançado: <i>poucas aplicações</i>	não linear, ótimo, robusto, inteligente

As estratégias convencionais de controle são vastamente utilizadas na indústria. Entre os motivos pode-se mencionar: estrutura simples, robustez, reduzido número de parâmetros a serem configurados, conhecimento intuitivo sobre o desempenho destas técnicas de controle, e requerimento mínimo de conhecimento matemático e técnico. As limitações destas estratégias de controle estão relacionadas com o número reduzido de graus de liberdade que oferecem.

Uma observação relevante é o aumento do número de aplicações de controle avançado, devido a diversos fatores vinculados a benefícios econômicos, tais como: projeto integrado de plantas e em sistemas de controle, flexibilidade na produção, aprimoramento do controle de qualidade, restrições ambientais, requerimentos de economia de energia, tentativas de maximização dos meios do setor produtivo (DE KEYSER, 1998), visando que os processos industriais sejam operados de maneira eficiente atendendo requisitos e normas de qualidade impostas pelo mercado, cada vez mais competitivo.

Segundo SEBORG & HENSON (1997), nos últimos anos, existe um ressurgimento de interesse no desenvolvimento de estratégias de controle aprimoradas e estratégias de identificação de sistemas não lineares motivado por diversos fatores, tais como:

- (i) avanços da teoria de sistemas não lineares, ocasionando metodologias de projeto aplicáveis a uma extensão de problemas de controle não linear;
- (ii) desenvolvimento de métodos de identificação eficientes, para modelos não lineares empíricos, e vasta aplicabilidade em pacotes computacionais, disponíveis comercialmente;
- (iii) desenvolvimento continuado das capacidades de *software* e *hardware*, tornando possível a incorporação de modelos não lineares complexos aos sistemas de controle.

As estratégias de controle avançadas permitem o aprimoramento do desempenho dos sistemas de controle, se comparada com as estratégias convencionais. Mas, usualmente, o projetista necessita configurar um grande número de parâmetros que, em alguns casos, pode dificultar o domínio do conhecimento de usuários, que não são especialistas na utilização destas técnicas de controle. Também, é necessária a utilização de diferentes pacotes ou mesmo o desenvolvimento de um sistema para a implementação de diferentes estratégias de controle (NAVARRO & ALBERTOS, 1996). Entretanto, o projeto adequado de metodologias de controle avançadas, considerando um compromisso entre desempenho e complexidade, pode oferecer uma ferramenta acessível e eficiente para a comunidade de controle atuante no meio industrial.

1.2 Inteligência computacional

Os seres humanos são fascinados com a idéia da construção de máquinas inteligentes, por duas razões: imitar o comportamento inteligente e providenciar soluções funcionais para tarefas complexas, que não podem ser tratadas por outras metodologias. A abordagem tradicional da produção de *software*, para a resolução de problemas particulares, não pode ser aplicada facilmente, em geral, em áreas tais como: reconhecimento de voz, visão, e robótica. Estas áreas são melhor resolvidas com técnicas da inteligência computacional (IC), que sejam aptas a desenvolver soluções, na forma simbólica ou numérica, para o problema particular em consideração (DRACOPOULOS & KENT, 1997).

A teoria de controle moderno, conforme mencionado anteriormente, tem sido aplicada com sucesso em sistemas bem definidos deterministicamente e/ou estocasticamente. Em ambientes práticos, contudo, encontram-se muitos problemas em aplicações da engenharia de controle, onde os sistemas a serem controlados exibem características complexas, tais como:

- (i) não-linearidades ou incertezas no modelo;
- (ii) entradas dos sensores e variáveis de controle multivariáveis complexas;
- (iii) atrasos de transporte das malhas de controle variantes no tempo;
- (iv) medidas apresentando incertezas e inadequação;
- (v) tratamento de aspectos de segurança da planta, diagnóstico de falhas e restrições ambientais;
- (vi) ambientes variantes no tempo e que apresentam incertezas;
- (vii) mudanças nas variáveis externas não medidas e não-controláveis, tais como: variação da persistência de alimentação, temperaturas externas e pressão barométrica.

O desenvolvimento da tecnologia de processamento de informação e a inteligência computacional constituem-se em uma solução alternativa para estes problemas e apresenta aspectos relativos à incorporação de características inspiradas na natureza e na inteligência humana em sistemas de controle automático. Estes sistemas de controle, ditos “inteligentes”, tentam imitar a maneira de tomada de decisão humana e a representação de conhecimento, motivando a atenção crescente da comunidade de controle (BARTOS, 1997, NG & LI, 1994). Entre as vantagens potenciais oferecidas por estas técnicas, em relação aos esquemas convencionais de controle, pode-se ressaltar (COELHO & COELHO, 1997d, 1997e):

- (i) menor dependência de modelos quantitativos (vantagem dos sistemas nebulosos);
- (ii) algoritmos estruturados de maneira simples para a tomada de decisões;
- (iii) capacidade de aprendizagem;
- (iv) maior grau de autonomia.

Os sistemas de controle inteligentes têm a habilidade de aprendizado, raciocínio e tomada de decisão. Todavia, os sistemas de controle convencionais apresentam certo grau de inteligência. As características que os diferenciam são o grau de complexidade, incerteza e o tipo de informações apresentadas para o sistema (SHOURESHI, 1993).

Em particular, os sistemas de controle inteligentes devem possuir a habilidade de tratarem um vasto conjunto de incertezas, os aspectos qualitativos da informação, as estruturas de dados complexas, a vasta quantidade de dados não-estruturados e a informação de sensores. Em termos de inteligência, é esperado que um sistema de

controle inteligente seja capaz de replicar habilidades humanas, entre as quais, criação, sensação, percepção, adaptação, inferência e desenvolvimento da experiência. Além disso, no atual estágio do desenvolvimento tecnológico, os sistemas de controle ditos inteligentes apresentam, ainda, características rudimentares quanto aos aspectos de autonomia em termos de aprendizado, raciocínio, planejamento e tomada de decisões (SHOURESHI, 1993).

Apesar do controlador clássico *PID* ter um baixo nível de inteligência, devido a sua estrutura fixa, este tipo de *PID* não tem auto-reconfiguração e a habilidade de tomada de decisões é baseada no cálculo da resposta de controle em termos da realimentação do erro. Por exemplo, um controlador *PID* adaptativo apresenta um nível médio de inteligência, porque tem um mecanismo de reconfiguração dos ganhos com uma estrutura fixa e algumas capacidades de aprendizado, em termos de identificação de sistemas, usualmente, via algoritmos recursivos do tipo mínimos quadrados.

ZADEH (1994b) propôs a denominação de *soft computing* (*SC*) para uma coleção de metodologias que visam explorar: a tolerância a falhas, a imprecisão e a incerteza proporcionando robustez e solução de baixo custo. Os principais membros deste consórcio denominado *SC*, incluem os campos das redes neurais, a computação evolutiva, os sistemas nebulosos, o raciocínio probabilístico (gerenciamento da incerteza e aprendizado de máquina) e a combinação destes (sistemas híbridos).

A *hard computing* (*HC*) representa lógica *crisp* ou binária, e o estilo tradicional de computação. Os exemplos de *HC* são: computação numérica, processamento simbólico e engenharia do conhecimento.

A *SC* é uma poderosa ferramenta para o projeto e o desenvolvimento de sistemas inteligentes e, mais especificamente, com um fundamento para a configuração de algoritmos de controle inteligente. A essência da *SC* difere da *HC*, pois a *SC* visa tratar problemas inerentemente imprecisos do mundo real. Contudo, o princípio que guia a *SC* é explorar a tolerância à imprecisão e incerteza visando aspectos de tratabilidade e melhor relação com a realidade. Em síntese, o modelo de regra para a *SC* é o pensamento humano (ZADEH, 1994b, 1996).

Apesar de ZADEH (1994b, 1996) definir a *SC* existem muitas discussões sobre a definição e os aspectos relevantes, quanto às características, deste campo emergente do

conhecimento. LI *et al.* (1998) apresenta discussões consistentes sobre a conceituação, as principais características e os tipos de problemas adequados a aplicação de SC.

1.2.1 Sistemas nebulosos

Os seres humanos, quando tomam decisões, tendem a trabalhar com conceitos vagos e imprecisos, que podem freqüentemente ser expressos lingüisticamente. Uma das maneiras de modelar o processo de tomada de decisões, proposto por ZADEH (1965) é denominado de conjuntos nebulosos (*fuzzy sets*) habilitando certas classes de declarações lingüísticas serem tratadas matematicamente. Os sistemas nebulosos são úteis para o tratamento da imprecisão, raciocínio aproximado, sistemas baseados em regras e cálculos utilizando termos lingüísticos.

A subárea de controle nebuloso é constituída de diversas metodologias baseadas na teoria dos sistemas nebulosos. Os controladores nebulosos são úteis, principalmente, no projeto e controle de sistemas complexos em que o modelo matemático é complexo e/ou não pode ser obtido facilmente. Neste caso, as regras são utilizadas para expressar a experiência do operador e/ou conhecimento do engenheiro de controle. A concepção lingüística da teoria dos conjuntos nebulosos habilita a fusão de processamento simbólico e numérico, provendo uma ferramenta promissora para o tratamento de problemas de decisão e controle (KATAYAMA *et al.*, 1993).

Recentemente, a literatura tem apresentado alguns estudos sobre a utilização de sistemas nebulosos em controle de processos multivariáveis. LING & EDGAR (1993) propõem o projeto de controle nebuloso multivariável do tipo *PI* aplicado em um processo simulado. LU *et al.* (1994) apresentam o projeto de controle nebuloso multivariável, com ajuste *on-line* dos fatores de escala, aplicado a um processo experimental *dyeing jig*.

STENZ & KUHN (1995) propõem um estudo da aplicação de controle nebuloso multivariável, combinado a técnicas de controle linear e não linear, em problema de controle de uma coluna de destilação. ORDÓÑEZ *et al.* (1996) apresentam um estudo teórico da configuração de controle nebuloso adaptativo indireto utilizando os sistemas nebulosos do tipo Takagi-Sugeno-Kang, os sistemas nebulosos convencionais e as redes neurais artificiais. VILJAMAA & KOIVO (1996) apresenta o procedimento de sintonia

de controladores nebulosos multivariáveis do tipo *PI* para o controle, em escala laboratorial, de uma máquina de papel. NIE (1997) controla servo-mecanismos através de dois projetos de controladores nebulosos visando o desacoplamento entre as malhas de controle. OLIVEIRA & LEMOS (1997) propõem um modelo relacional nebuloso simplificado relevante para aplicações em controle de processos.

1.2.2 Redes neurais

As redes neurais (*RNs*) são técnicas matemáticas inspiradas por estudos do cérebro humano e o sistema nervoso. As redes neurais podem ser consideradas como sistemas dinâmicos não lineares consistindo de elementos de processamento (neurônios), que são interconectados entre si, com ponderações ajustáveis. Entre as características relevantes das redes neurais têm-se: processamento paralelo, aprendizado, memória associativa e distribuída. Estas características são inspiradas nas redes neurais biológicas, mesmo que rudimentarmente (LU, 1992).

As redes neurais providenciam, usualmente, conhecimento quantitativo não-paramétrico e são adequadas para identificação de sistemas, aprendizado e adaptação de processos complexos. As redes neurais em aplicações de controle de processos providenciam funções versáteis, tais como reconhecimento de padrões, identificação de sistemas não lineares, diagnóstico e controle de processos. Entretanto, os aspectos relativos a convergência e as propriedades de estabilidade das redes neurais ainda é assunto de diversas pesquisas e gerador de divergências na comunidade de controle.

Alguns estudos do estado da arte de redes neurais, em identificação e controle de processos multivariáveis, têm sido apresentados. BARTO (1990) aborda um apanhado do aprendizado de redes neurais, enfatizando aspectos de projetos para o controle de sistemas dinâmicos. HUNT *et al.* (1992) apresentam um *survey* das técnicas e as aplicações de topologias de redes neurais em controle de processos. HAGAN & DEMUTH (1999) apresentam um tutorial de abordagens a aplicações de redes neurais em controle de processos.

Quanto a aplicações em identificação de processos multivariáveis, algumas abordagens são relevantes. No artigo clássico de NARENDRA & PARTHASARATHY

(1990) são apresentadas diversas estruturas de projeto neural, em problemas de identificação de processos não lineares. MISTRY & NAIR (1994) aplicam as *RNs* em identificação e controle de um processo experimental, composto de um motor *DC* e uma barra com 4 juntas. LIGHTBODY & IRWIN (1996) aplicam uma rede neural *multilayer perceptron* em identificação de um sistema composto de reator e tanque. JAGANNATHAN & LEWIS (1996) apresentam uma nova configuração de *RNs* à identificação de processos multivariáveis discretos no tempo.

Quanto às aplicações em processos multivariáveis de redes neurais, são apresentadas diversas abordagens na literatura. WILLIS *et al.* (1992) tratam as características relevantes de redes neurais em aplicações de controle preditivo. CHEN & LIU (1993) propõem um projeto de controle adaptativo, baseado em redes neurais, aplicado a um processo multivariável. KHALID *et al.* (1993) tratam as questões de configuração de controle neural, aplicado a um processo multivariável de controle de temperatura. ZBIKOWSKI *et al.* (1994) apresentam o projeto de controle adaptativo neural sob diferentes abordagens. STECK *et al.* (1996) apresentam projeto de controle neural não linear aplicado a um modelo dinâmico multivariável de um avião. SONG & GRIMBLE (1997) propõem um projeto de controlador neural multivariável para a compensação de não-linearidades.

1.2.3 Computação evolutiva

Outra metodologia da *SC* são os paradigmas da computação evolutiva denominados algoritmos evolutivos. Os algoritmos evolutivos são sistemas computacionais para resolução de problemas baseados nos princípios da teoria evolutiva e na genética. Uma variedade de algoritmos evolutivos tem sido desenvolvidos e todos dividem uma base conceitual comum, através de procedimentos de seleção, mutação e recombinação. O interesse nestes algoritmos é devido ao fato de serem técnicas robustas e proverem mecanismos de busca eficientes frente a buscas globais. Os algoritmos genéticos são os representantes da computação evolutiva com maior número de publicações e aplicações, devido, principalmente, a facilidade de implementação desta técnica.

As aplicações dos algoritmos evolutivos, em identificação de processos, propostas na literatura, têm abordado a identificação de processos lineares com estimativa do atraso de transporte, linearização de processos não lineares, incluindo-se neste contexto processos monovariáveis e multivariáveis, além de comparativos com técnicas convencionais em identificação (CHIPPERFIELD & FLEMING, 1996).

Os algoritmos evolutivos são uma ferramenta robusta à configuração e ao projeto de sistemas de controle, geralmente realizado através de procedimento *off-line*. A sintonia de parâmetros dos controladores *PID* e alocação de pólos através de algoritmos evolutivos são descritas e analisadas por diversos pesquisadores em aplicações multivariáveis (COELHO & COELHO, 2000b), aplicações a processos monovariáveis não lineares (COELHO & COELHO, 1999c) e projeto de controle robusto (KRÖHLING *et al.*, 1999).

Entre as aplicações de algoritmos evolutivos em identificação e controle multivariáveis pode-se mencionar algumas: PATTON & LIU (1994) apresentam o desenvolvimento de algoritmos de otimização com múltiplos objetivos para o controle robusto multivariável via algoritmos genéticos; ZUO (1995) propõe um método de sintonia e projeto de controle *PID* multivariável via algoritmos genéticos, aplicado a um sistema de controle de estação espacial; TAN *et al.* (1996) configuram o projeto genético de problemas de linearização, no domínio da frequência, para processos multivariáveis. Nesta tese, as aplicações dos algoritmos evolutivos objetivam a identificação de processos, otimização e sintonia de controladores. Estas aplicações são detalhadas nos capítulos 8 e 9 e podem ser divididas em quatro tópicos:

- (i) identificação de processo não linear;
- (ii) sintonia de parâmetros de controladores do tipo *PID*;
- (iii) otimização de sistemas híbridos inteligentes, combinando sistemas nebulosos e/ou redes neurais;
- (iv) otimização da lei de controlador preditivo generalizado.

1.2.4 Sistemas híbridos inteligentes

As metodologias da *SC* — sistemas nebulosos, redes neurais, computação evolutiva, *rough sets* e raciocínio probabilístico — são complementares mais que competitivas. Por esta razão é frequentemente vantajoso utilizar estas metodologias em

combinação, levando ao desenvolvimento dos denominados sistemas híbridos inteligentes. Nesta concepção, de sistemas híbridos inteligentes, têm-se os sistemas neuro-nebulosos, nebulosos-evolutivos, neuro-evolutivos e neuro-nebuloso-evolutivos, com diversas aplicações em identificação e controle de processos, e outras áreas afins (KOSLA & DILLON, 1997).

Recentemente, na literatura, são apresentados alguns estudos da aplicação de sistemas híbridos inteligentes em projetos de identificação e controle de processos multivariáveis. TAKAGI & HAYASHI (1991) propõem a idéia da utilização de redes neurais para o projeto de sistemas nebulosos e a utilização do método do gradiente descendente para a sintonia dos parâmetros que definem a forma e a posição das funções de pertinência. Um exemplo da utilização deste tipo de sistemas híbrido nebuloso-neural é o controle de uma máquina fotocopadora da Matsushita Electric.

GARIGLIO *et al.* (1994) propõem a combinação de redes neurais e algoritmos evolutivos em identificação e controle de uma coluna de destilação através de simulação. MILLS & HARRIS (1995) empregam a metodologia híbrida neuro-nebulosa em tarefas de identificação e controle multivariável de um veículo autônomo. HAYASHI *et al.* (1995) propõem um sistema de controle neuro-nebuloso de transmissão de um automóvel com carga variável projetado e testado em um sistema de transmissão experimental.

KIM *et al.* (1995) apresentam a configuração de um algoritmo neuro-nebuloso-genético em controle de um sistema clássico composto de um pêndulo invertido sob um carro. TANI *et al.* (1996) apresentam um sistema de controle híbrido neuro-nebuloso para controle de nível de um planta petrolífera. LINKENS & NYONGESA (1996b) apresentam o projeto de controle nebuloso multivariável com aprendizado por algoritmos genéticos aplicado a processo simulado de anestesia. CARBAJAL *et al.* (1998) apresentam o projeto de controle nebuloso combinado a redes neurais e algoritmos genéticos, aplicado a uma fábrica siderúrgica. SENG *et al.* (1999) propõem um controlador neuro-nebuloso com otimização via algoritmos genéticos. KASABOV (1999) apresenta uma abordagem híbrida para a tomada de decisão e controle baseado em estruturas neuro-genética e neuro-nebulosa com otimização genética.

1.3 Identificação de processos

A identificação de processos é uma área relevante em muitos campos do conhecimento. A identificação de processos é o procedimento de identificar um modelo de um processo desconhecido, para propósitos de previsão e/ou compreensão do comportamento do processo.

A complexidade inerente de muitos processos reais (multivariáveis, não lineares e variantes no tempo) dificulta a aplicação de técnicas convencionais de identificação. Este aspecto tem motivado o desenvolvimento de técnicas de identificação avançadas, baseadas em *IC*, que sejam aptas a (BARTOS, 1997; CHIU *et al.*, 1995):

- (i) gerenciar a incerteza e o conhecimento de especialistas;
- (ii) acomodar mudanças significativas no processo e seu ambiente;
- (iii) incorporar metodologias para aprendizado de informações incertas;
- (iv) combinar o conhecimento existente com um procedimento de aprendizado.

1.4 Controle preditivo, preditivo adaptativo e baseado em *IC*

O controle preditivo baseado em modelo é uma metodologia que utiliza o modelo do processo para o cálculo das previsões e otimização das ações de controle. Os dois princípios relevantes que regem estes controladores são (DE KEYSER, 1998):

- (i) utilização explícita de um modelo do processo para as previsões das saídas do processo nas amostras futuras;
- (ii) cálculo de uma ação de controle ótima, baseada na minimização de uma (ou mais) função objetivo, possivelmente incluindo-se as restrições nas variáveis do processo.

Os métodos preditivos, em controle adaptativo, referem-se a uma coleção de formulações de projeto de controle, que possuem um critério de controle dado explicitamente em termos de previsões das entradas e/ou saídas futuras do processo estimadas de forma *on-line* (BITMEAD *et al.*, 1990). Entre as vantagens da utilização de controle preditivo têm-se (COELHO *et al.*, 1999d):

- (i) a possibilidade da compensação de atraso de transporte e compensação *feedforward* de perturbações;
- (ii) o controle de processos com comportamento dinâmico não usual, tais como: fase não-mínima, oscilatórios e instáveis em malha aberta.

A aplicação de estruturas de controle convencionais em processos multivariáveis geralmente apresenta desempenho inadequado quando o processo é complexo, ou seja, apresenta não-linearidades, restrições, fase não-mínima, atraso de transporte maior que a unidade e o acoplamento entre as variáveis (AMARAL *et al.*, 1998).

Os controladores preditivos podem ser utilizados baseados em metodologias de identificação inteligentes. Um sistema preditivo híbrido incorporando metodologias inteligentes constitui-se de uma ferramenta adequada ao tratamento de processos não lineares, principalmente multivariáveis, onde os problemas e as complexidades devidos ao acoplamento entre variáveis estão, freqüentemente, presentes.

A implementação de modelos baseados em algoritmos inteligentes fornece a previsão futura da saída do processo, para um determinado horizonte de tempo. As previsões supridas por modelos com configuração via algoritmos evolutivos, sistemas nebulosos, redes neurais ou sistemas híbridos inteligentes são passadas para uma rotina de otimização numérica, que visa minimizar uma função objetivo. O controle preditivo resulta em ações de controle adequadas, se o modelo matemático estimado, o procedimento de otimização e o ajuste das ponderações e horizontes são projetados adequadamente.

A previsão do comportamento dinâmico da planta em mais de um instante futuro de amostragem — *single-step-ahead* (controle de variância mínima) ou *multi-step-ahead* (controle preditivo) — necessita a utilização dos valores preditos pela metodologia inteligente, usualmente alimentando-se o estimador com as entradas anteriores e as saídas anteriores e atuais do processo.

A utilização de *IC* tem sua principal motivação no tratamento de processos não lineares com acoplamentos entre variáveis, onde a configuração de algoritmos convencionais de identificação é uma tarefa complexa. O aumento da complexidade destes algoritmos para processos multivariáveis não lineares, com acoplamentos entre variáveis, é ainda mais proeminente.

O desacoplamento é um tópico típico da teoria de controle e engenharia. Em diversos sistemas industriais *MIMO* (múltiplas entradas e múltiplas saídas) e *MISO* (múltiplas entradas e uma saída) as interconexões existem entre as entradas e saídas, isto

é, cada entrada controla mais que uma saída e cada saída é controlada por mais que uma entrada. Os sistemas com tais interconexões são denominados de sistemas acoplados.

No projeto e implementação de um sistema de controle de um simples processo com acoplamentos, é usualmente desejável que as interconexões sejam tratadas de forma que cada entrada controle somente uma saída e cada saída seja controlada por somente uma entrada. O tratamento das interconexões é referido como projeto desacoplado na literatura de controle (XU, 1991). As técnicas de controle desacoplado apresentam limitações pela falta de robustez, já que são freqüentemente insensíveis às incertezas do modelo do sistema. As metodologias de desacoplamento, aproximadas utilizando-se *IC*, podem ser uma maneira alternativa na resolução deste problema.

Os controladores preditivos baseados em modelo são alvo de diversas pesquisas vinculadas a aplicações em processos multivariáveis. A prova disto são as publicações recentes na área. DU & DUMONT (1996) propõem um esquema de controle preditivo generalizado multivariável e demonstram as vantagens desta técnica aplicada a um processo químico-termo-mecânico. MAHFOUF & LINKENS (1997) apresentam o controle preditivo generalizado multivariável com tratamento de restrições via programação quadrática, e aplicado a um processo de anestesia. DUBAY *et al.* (1997) apresentam um estudo experimental e comentam a eficiência de controladores preditivos baseados em modelo no controle da temperatura de fusão do plástico.

AMARAL *et al.* (1998) apresentam um estudo de equivalência e desempenho de controladores de variância mínima generalizada, baseado no conhecimento da matriz *interactor* do processo. HANSEN *et al.* (1998) apresentam procedimentos de identificação e controle preditivo multivariável aplicado ao controle de um sistema de geração de potência a diesel. TSAI & LU (1998) desenvolvem um controlador preditivo generalizado multivariável para controle em tempo real, utilizando uma *DSP*, aplicado ao controle de temperatura. MAGNI *et al.* (1999) propõem um projeto de controle preditivo não linear aplicado à fabricação de cimento. VILLAMIL & RIVERA (1999) propõem o projeto de controle preditivo baseado em modelo multivariável e adaptativo para a otimização e o controle da manufatura de semicondutores.

1.5 Controle de estrutura variável

Uma das realizações na pesquisa sobre sistemas incertos é o desenvolvimento da abordagem de controle de estrutura variável, principalmente a metodologia de controle por modos deslizantes. Os sistemas de estrutura variável constituem-se em uma classe de sistemas não lineares em que a lei de controle (ou estrutura de controle) é modificada durante a atuação para obter-se um aprimoramento das características de controle do processo. O controle por modos deslizantes é eficiente para o tratamento de processos com variações paramétricas e sujeito a perturbações externas.

As origens da teoria de controle de modos deslizantes foram concentradas para o caso de sistemas contínuos. Na abordagem contínua, a regra da alteração da estrutura do controlador é regida por funções das variáveis de estado do sistema. O projeto de controle por modos deslizantes para sistemas discretos é denominado de controle de modos quase-deslizantes. Na forma discreta, o controlador é projetado para mover o estado do sistema do exterior para o interior de um setor deslizante, onde o sistema é estável. As abordagens adaptativas e incorporando metodologias da *IC* do controle de estrutura variável, na forma discreta, são descritas na tese.

Na literatura existem alguns estudos sobre o projeto de controle de estrutura variável para o tratamento de processos multivariáveis. SU *et al.* (1993) apresentam o projeto prático de controle por modos deslizantes aplicado a um manipulador robótico na presença de perturbações, variações paramétricas e dinâmicas não-modeladas. HEALAY & LIENARD (1993) propõem um projeto de controle por modos deslizantes aplicado a um veículo submarino, com 6 graus de liberdade. BARTOLINI & FERRARA (1996) abordam uma forma eficiente de configuração de controle por modos deslizantes discreto para o tratamento de processos com múltiplas entradas. IORDANOU & SURGENOR (1997) avaliam experimentalmente e comparam o controle quadrático linear e o controle por modos deslizantes, aplicados a um pêndulo invertido. ZHIHONG & YU (1997) abordam o projeto de controle por modos deslizantes aplicado a processos lineares multivariáveis. CORRADINI & ORLANDO (1997) propõem o projeto de controle de estrutura variável adaptativo discreto, aplicado a um modelo de veículo submarino. TANG & MISAKA (1998) apresentam uma estrutura de controle por modos deslizantes para o controle de processos lineares multivariáveis.

1.6 Objetivos da tese

Este trabalho objetiva apresentar metodologias de projeto em identificação e controle de processos multivariáveis, empregando metodologias avançadas em controle de processos. As áreas de controle avançado abordadas são:

- (i) inteligência computacional (algoritmos evolutivos, sistemas nebulosos, redes neurais artificiais e sistemas híbridos inteligentes);
- (ii) controle preditivo adaptativo;
- (iii) controle de estrutura variável;
- (iv) algumas configurações de controle preditivo, baseadas em modelos da inteligência computacional.

Os projetos e os resultados apresentados, em identificação de processos, são obtidos através de diversas metodologias, incluindo-se: algoritmos evolutivos, sistemas nebulosos, redes neurais e sistemas híbridos inteligentes. Espera-se avaliar as limitações e as potencialidades de cada técnica no tratamento dos processos multivariáveis não lineares.

A classe de processos avaliada em identificação multivariável é caracterizada por apresentar complexidades, tais como: comportamento não linear com parâmetros desconhecidos ou variantes no tempo e acoplamento entre as variáveis. Para avaliação experimental das técnicas, um processo experimental denominado balanço horizontal, um manipulador robótico com dois graus de liberdade e um processo não linear regido por equações a diferenças, são tratados em identificação multivariável. Esta análise auxilia e torna-se importante na solução do compromisso entre a complexidade do modelo matemático e a sua capacidade de representação das dinâmicas de um processo multivariável.

Em controle de processos apresenta-se a descrição e a análise de desempenho dos algoritmos inteligentes, preditivos e de estrutura variável, quando aplicados a um manipulador robótico, utilizado no procedimento de identificação. Um estudo comparativo e análise de desempenho são realizados para a verificação do comportamento servo e comportamento regulatório. A análise dos resultados e as discussões são apresentadas no sentido de avaliar as vantagens e desvantagens da utilização das técnicas avançadas, visando o tratamento das complexidades dos processos multivariáveis. As

contribuições, neste sentido, podem ser úteis quanto ao aproveitamento industrial ou comercial, futuramente, de algumas das metodologias de projeto em identificação e controle de processos complexos.

Os principais objetivos desta tese são, portanto,

- (i) realizar uma descrição dos aspectos relacionados à identificação multivariável não linear, tais como: tipos de modelos, sinais de excitação, seleção de estrutura, escolha de ordem, critérios de avaliação de desempenho e procedimentos de estimação e validação;
- (ii) analisar o estado-da-arte, fundamentos e potencialidades dos algoritmos evolutivos;
- (iii) avaliar configurações de aprendizado Lamarckiano, baseadas em algoritmos evolutivos combinados a *simulated annealing* e o método de otimização simplex;
- (iv) analisar o desempenho de um sistema nebuloso com geração de regras para a identificação de processos multivariáveis;
- (v) avaliar os aspectos do projeto de controle nebuloso do tipo *PID* para aplicações em processos multivariáveis;
- (vi) realizar um estudo comparativo de métodos de treinamento de redes neurais do tipo *perceptron* multicamadas;
- (vii) aplicar as redes neurais função de base radial, rede neural parcialmente recorrente de Elman, rede neural *Group Method of Data Handling* e sistemas híbridos inteligentes em identificação multivariável não linear;
- (viii) abordar os paradigmas da inteligência computacional em identificação e controle de processos multivariáveis não lineares;
- (ix) analisar o desempenho de controle preditivo generalizado, na forma adaptativa e inteligente;
- (x) descrever e aplicar controle de estrutura variável combinado a controle de variância mínima, em formulação adaptativa.

1.7 Organização da tese

A estrutura da tese é dividida nos seguintes tópicos: Este capítulo apresenta uma abordagem introdutória e a motivação à identificação e controle de processos através das metodologias ditas “avançadas”.

No capítulo 2 são discutidos os conceitos fundamentais, as definições e as características relativas aos algoritmos evolutivos. No mesmo capítulo, os paradigmas evolutivos tratados são: algoritmos genéticos, estratégias evolutivas e programação evolutiva.

No capítulo 3 são descritas as noções relativas a sistemas nebulosos, procedimentos de projeto de identificação nebulosa e controle lógico nebuloso de forma

sistemática. Os aspectos relativos a formas de projeto e estruturação de controle lógico nebuloso do tipo *PID* são enfatizados.

As diversas redes neurais e os tópicos relativos a aplicações de identificação e comparação de concepções *feedforward* e *feedback* em controle de processos, são abordados no capítulo 4. No capítulo 5 descreve-se o projeto de sistemas híbridos inteligentes, motivado por aliar as vantagens de cada paradigma da *IC* na realização de sistemas robustos e eficientes. Esta visão integrada dos sistemas híbridos é apresentada de forma a combinar algoritmos evolutivos, sistemas nebulosos e redes neurais artificiais.

O capítulo 6 descreve o projeto de controle preditivo multivariável de processos com configuração adaptativa. Algumas possibilidades da integração de controle preditivo generalizado baseado em modelos da *IC*, são também tratados neste capítulo. O capítulo 7 apresenta as características e algumas concepções de projeto de estrutura variável na forma discreta. Os resultados de simulação e experimentais em identificação e controle de processos multivariáveis são tratados nos capítulos 8 e 9, respectivamente. No capítulo 10 são apresentadas a conclusão geral da tese e as perspectivas para futuros trabalhos.

A tese, sob ponto de vista das áreas de controle avançado, está dividida nos seguintes tópicos gerais: Em síntese, os capítulos 2 a 5 são relativos a metodologias da *IC*, o capítulo 6 é relativo a controle preditivo multivariável adaptativo e inteligente, enquanto o capítulo 7 trata do controle de estrutura variável.

Capítulo 2

COMPUTAÇÃO EVOLUTIVA: FUNDAMENTOS E APLICAÇÕES

2.1 Introdução

Os métodos de otimização e busca estocástica baseados nos princípios e modelos da evolução biológica natural têm recebido crescente interesse nas últimas décadas, devido principalmente a sua versatilidade para a resolução de problemas complexos, nas áreas de otimização e aprendizado de máquina. O desenvolvimento de modelos computacionais, inspirados nos mecanismos evolutivos, caracteriza-se pela configuração de algoritmos de otimização robustos e sistemas adaptativos (CHIPPERFIELD & FLEMING, 1996).

Os algoritmos evolutivos (*AEs*) — metodologias da área computação evolucionária ou evolutiva (*CE*) — não são algoritmos computacionais em seu significado usual, mas formam uma classe de métodos regidos por princípios similares. Estes princípios oriundos do “mundo biológico” são baseados na teoria da evolução Darwiniana (DARWIN, 1872). Os *AEs* tentam abstrair e imitar alguns dos mecanismos evolutivos à resolução de problemas que requerem adaptação, busca e otimização.

Nos *AEs*, os pontos no espaço de busca são modelados através de indivíduos que interagem em um ambiente artificial. Um conjunto de soluções (população) é manipulado a cada iteração, em contraste com outros métodos de otimização, onde apenas uma solução para o problema é utilizada. A chance que um indivíduo da população seja selecionado na próxima geração depende da função de aptidão (existem outros sinônimos, utilizados na literatura, tais como: função de adequação, adequabilidade, *fitness*) do indivíduo, que consiste, geralmente, de uma função objetivo ou mesmo uma transformação simples desta para o tratamento do problema em questão.

Os *AEs* não requerem o conhecimento das características do problema de otimização e não dependem de certas propriedades da função objetivo, tais como

convexidade ou diferenciabilidade. Os *AEs* são guiados pela avaliação da função de aptidão dos indivíduos e não dependem do tipo de problema que está sendo tratado, se o problema é codificado de forma adequada:

Os *AEs* são especialmente úteis às tarefas de otimização global, onde os métodos determinísticos podem levar a soluções de mínimos locais. Por conseguinte, os *AEs* são aptos à resolução de um amplo espectro de problemas não lineares, descontínuos, discretos, multivariáveis, entre outros (MITCHELL, 1996). O ciclo básico dos dados num *AE* é baseado nos seguintes passos:

- (i) inicialização aleatória da população de soluções;
- (ii) avaliação da função de aptidão;
- (iii) seleção dos indivíduos mais aptos de acordo com uma estratégia de seleção;
- (iv) aplicação dos operadores de recombinação e mutação;
- (v) geração de uma nova população de soluções candidatas;
- (vi) repetição dos passos (ii) a (v) até que uma condição de parada seja satisfeita.

2.1.1 Princípios dos procedimentos evolutivos naturais

O propósito desta seção é apresentar uma breve abordagem dos princípios evolutivos naturais, que fundamentam os *AEs*. Os detalhes podem ser encontrados na seguinte literatura: HOLLAND (1975), GOLDBERG (1989), DAVIS (1991), MICHALEWICZ (1992), FOGEL (1995), MITCHELL (1996) e BÄCK *et al.* (1997a).

O conjunto mais aceito de teorias evolutivas é o paradigma *neo-Darwiniano*. Os argumentos deste paradigma defendem que a maior parte da história da vida animal considera a operação de procedimentos físicos em populações e espécies. Estes procedimentos abordam a evolução como uma interação de quatro procedimentos essenciais: reprodução, competição, mutação e seleção (FOGEL, 1995, BÄCK *et al.*, 1997a).

A reprodução é uma propriedade óbvia das espécies existentes. As espécies, que apresentam maior potencial reprodutivo, têm o crescimento da população com proporção exponencial, se todos os indivíduos destas espécies reproduzirem-se com sucesso. A reprodução realiza a transferência do código genético do indivíduo (assexuadamente ou sexuadamente) para a progênie. Através da reprodução, os indivíduos com maior aptidão

são copiados várias vezes na próxima geração, enquanto os indivíduos com menor aptidão recebem poucas cópias (ou mesmo nenhuma).

A mutação é responsável pelo fato que os erros de replicação, durante a transferência de código genético, necessariamente ocorrem. A competição é uma consequência de populações em expansão em um ambiente de recursos finitos. A seleção é o resultado decorrente da competição, que influencia a forma que as espécies ocupam os recursos disponíveis (BÄCK *et al.*, 1997a).

Os indivíduos e as espécies podem ser vistas como uma dualidade de seus códigos genéticos, o *genótipo*, e a expressão de suas características comportamentais, o *fenótipo*. Os resultados das variações genéticas são geralmente imprevisíveis, devido aos efeitos de *pleiotropia* e *poligenia*. A *pleiotropia* é o efeito onde um gene pode afetar simultaneamente diversas características fenotípicas. A *poligenia* é o efeito onde uma característica fenotípica simples pode ser determinada pela interação simultânea de muitos genes (FOGEL, 1994, 1995).

A evolução é um procedimento que opera em cromossomos a nível genotípico e pode ser vista, do ponto de vista matemático, como um procedimento de otimização. A seleção natural relaciona os cromossomos com o desempenho de suas estruturas codificadas, ou seja, a seleção direciona os fenótipos para um valor tão próximo, quanto possível, do ótimo, dadas as condições iniciais e as restrições do ambiente.

A seleção natural influencia os cromossomos, que codificam estruturas bem sucedidas, a reproduzirem mais frequentemente que aqueles que não as codificam. Entretanto, o ambiente natural está em mudança contínua. Nenhum indivíduo pode ser visto como perfeitamente adaptado ao seu ambiente, pois o indivíduo está em constante evolução para um “novo” ótimo.

2.1.2 Inspiração biológica para a evolução e o aprendizado

Os aspectos relacionados à inspiração biológica da evolução e o aprendizado são resumidos nesta seção. Estes aspectos servem de inspiração ao desenvolvimento dos *AEs* e suas variantes e têm muitos fundamentos baseados em estudos de Charles Darwin, Jean-

Baptiste de Lamarck, Gregor Mendel, James Baldwin, Hugo Marie De Vries, Thomas Morgan, e outros.

No século XIX, Darwin da mesma forma que Lamarck observou a adaptação e a adequação entre a forma e o desempenho dos órgãos dos seres vivos e, sua maneira de viver. A diferença é que Darwin apresentou em suas pesquisas um mecanismo de evolução baseado na seleção natural. Entretanto, a proposta de Lamarck defendia o mecanismo de evolução através da hereditariedade, das características adquiridas diretamente durante a vida, também denominada de lei do uso e desuso (DIAS *et al.*, 1998). Em síntese, Lamarck propôs que mudanças ambientais através da vida de um organismo causam mudanças estruturais, que são transmitidas ao seu descendente.

Enquanto, até hoje, os biólogos não acreditam na plausibilidade biológica desta teoria, o poder da teoria Lamarckiana pode ser ilustrado pela evolução da nossa sociedade. Neste caso, as idéias e o conhecimento são passados de geração para geração através da estruturação da linguagem e da cultura (KENNEDY, 1993; WHITLEY *et al.*, 1994).

No século XX, Mendel propôs uma teoria de hereditariedade e dominância baseada em genes. As Leis de Mendel procuram explicar a troca de material genético através de operações de cruzamento e mutação. Outra pesquisa relevante foi a de Baldwin, que acreditava no aprendizado adquirido, através das interações dos indivíduos com o meio, poderia ser transmitido a gerações futuras de forma indireta. A abordagem de Baldwin contraria a proposta de Lamarck, de aprendizado direto. Os estudos de De Vries são relevantes quanto a análise da forma como a mutação acontece na natureza. A reunião destas teorias em torno da denominação de neo-Darwinismo ou teoria sintética da evolução deu-se em uma conferência internacional em Princeton, nos Estados Unidos, em janeiro de 1947, onde estavam geneticistas, naturalistas e paleontólogos (DIAS *et al.*, 1998).

Richard Dawkins (DAWKINS, 1976) apresenta a evolução sob uma perspectiva diferente, onde a unidade fundamental da seleção natural é o gene, mais que o próprio indivíduo. Esta visão não é contrastante com alguns dos princípios da evolução natural, mas providencia uma forma de interpretação alternativa, que é formalizada na *teoria do gene egoísta*.

Outra contribuição de Dawkins é a utilização do termo *meme*, para referir-se a uma idéia. DAWKINS (1989) elaborou uma teoria para estudar e explorar os três fenômenos, que são únicos, para a evolução cultural.

O primeiro é o *operador baseado em conhecimento*: o cérebro detecta a regularidade e constrói esquemas que este operador utiliza para adaptar-se aos equivalentes mentais da mutação e da recombinação para os seus *memes* fundamentais. O segundo é a *imitação*: as idéias se propagam quando os membros de uma sociedade observam e copiam um ao outro. Isto é observado nas sociedades animais e humanas. A imitação habilita os indivíduos a compartilhar as soluções completas ou parciais dos problemas que eles defrontam-se. O terceiro é a *simulação mental*: os indivíduos podem imaginar o que poderia acontecer se um *meme* fosse implementado antes que os recursos fossem passados para ele. Este mecanismo provê os indivíduos de uma forma rudimentar de seleção (GABORA, 1995).

2.2 Breve histórico

As origens da *CE* podem ser traçadas por trabalhos pioneiros de R. M. Friedberg, H. J. Bremermann, W. Spindley, F. E. Satterthwaite, entre outros, nos anos 50. Após esta fase, este campo do conhecimento permaneceu relativamente desconhecido ou inexplorado pela maioria da comunidade científica, por mais de três décadas. Este fato deve-se, principalmente, à falta de plataformas computacionais poderosas naquela época, da formalização e caracterização deficiente de cada metodologia evolutiva nos primeiros estudos nesta área (FOGEL, 1994, 1995, BÄCK *et al.*, 1997a, 1997b).

Os trabalhos fundamentais de Holland, Rechenberg, Schwefel e Fogel, serviram à realização de mudanças neste cenário, durante a década de 70. Atualmente, observa-se um relevante e permanente crescimento do número de publicações (ALANDER, 1996), aplicações e conferências no campo da *CE*. A maioria das abordagens correntes dos *AEs* descende dos princípios de diferentes metodologias, principalmente (COELHO & COELHO, 1999c):

- (i) *algoritmos genéticos*, desenvolvidos principalmente por A. S. Fraser, H. J. Bremermann, J. Reed e J. H. Holland, entre a década de 50 e 70, com refinamentos posteriores por D. Whitley, D. E. Goldberg, K. De Jong e J. Grefenstette;
- (ii) *programação evolutiva*, desenvolvidas por L. J. Fogel, A. J. Owens e M. J. Walsh, nos Estados Unidos, na década de 60, refinada recentemente por D. B. Fogel, G. H. Burgin, P. J. Angeline, V. W. Porto e W. Atmar;
- (iii) *estratégias evolutivas*, desenvolvidas na Alemanha, por I. Rechenberg e H. P. Schwefel, na década de 60, com aprimoramentos posteriores de G. Rudolph, H. G. Beyer, F. Kursawe e T. Bäck;
- (iv) *programação genética (PG)*, abordadas pelos pesquisadores J. R. Koza, J. P. Rice, K. E. Kinneer e P. J. Angeline.

Um fato relevante é que durante muito tempo as metodologias que constituem hoje os *AEs* foram desenvolvidas independentemente uma das outras. Um esforço organizado, de forma a criar-se um fórum de interação das pesquisas entre as várias comunidades de pesquisadores de *AEs*, emergiu em 1990. O evento foi o *workshop* internacional intitulado *Parallel Problem Solving from Nature (PPSN'91)* realizado em Dortmund, na Alemanha. Após este evento, outros eventos significativos (*ICGA'91*, *Alife'91*, *EP'92*, *PPSN'92* e *WCCI'94*) e algumas publicações (*Evolutionary Computation*, da MIT Press, 1993, *BioSystems*, da Elsevier, e *IEEE Transactions on Evolutionary Computation*, 1997) levaram a consenso a denominação de *CE*, a este novo campo do conhecimento.

Entre os anos de 80 e 90, os avanços no desempenho das plataformas computacionais habilitaram a aplicação de *AEs* à resolução de problemas de otimização do mundo real e em projetos relevantes. Entre os projetos relevantes relativos a aplicação de *AEs* na indústria deve-se citar a *EvoNet*, uma rede europeia de excelência em computação evolutiva, fundada pela comissão europeia *ESPRIT IV*. A *EvoNet* é responsável pela disseminação e colaboração ativa entre pesquisadores da comunidade europeia e a transferência de tecnologia que utiliza os *AEs*, para os ambientes industrial e comercial. As áreas de aplicação abrangem: telecomunicações, escalonamento, robótica móvel, manufatura, reconhecimento de faces, identificação, previsão, controle, sistemas de potência, configuração de *hardware*, processamento de imagens, otimização de forma e processamento de sinais. Entre os membros da *EvoNet* estão: British Aerospace, Daimler-Benz, Dassault Aviation, Hewlett Packard Laboratories, Institut Français du Pétrole, Rolls-Royce, SGS-Thomson e Siemens (EVONEWS, 1996, 1999, SCHWEFEL, 1997).

2.3 Por que utilizar os AEs?

Os AEs são técnicas robustas e eficientes em espaços de procura irregulares, complexos e apresentando múltiplas dimensões. Um AE caracteriza-se por (GOLDBERG, 1994):

- (i) operar em uma população de pontos;
- (ii) não requerer cálculos de derivadas e informação sobre o gradiente da função objetivo;
- (iii) trabalhar com a codificação de seu conjunto de parâmetros, não com os próprios parâmetros (representação binária);
- (iv) realizar transições probabilísticas, em vez de regras determinísticas;
- (v) necessitar apenas da informação sobre o valor da função objetivo para cada indivíduo da população;
- (vi) apresentar simplicidade conceitual;
- (vii) ser pouco afetado, quanto à eficiência, quando descontinuidades e ruídos estão presentes nos dados do problema.

As características (iii) a (v) não são comuns a todos os AEs, mas geralmente presentes nos algoritmos genéticos. Quanto ao projeto e à configuração em sistemas de controle, os AEs são empregados com sucesso devido as seguintes características (COELHO & COELHO, 1999c):

- (i) tratarem adequadamente os sistemas sujeitos a restrições;
- (ii) não requererem as informações relativas a derivadas, estas usualmente necessárias em métodos convencionais de otimização;
- (iii) adequarem-se à implementação em paralelo e distribuídas;
- (iv) possibilitarem a utilização do conhecimento obtido *a priori* pelo projetista; e
- (v) tratarem com sistemas complexos e espaços de busca com múltiplas modas e/ou múltiplos objetivos.

Contudo, algumas limitações dos AEs devem ser mencionadas. Os AEs tratam-se de métodos estocásticos e seu desempenho varia de execução para execução (a menos que o mesmo gerador de números aleatórios com a mesma semente é utilizado). Devido a isto, a média da convergência sobre diversas execuções do AE é um indicador de desempenho mais útil que uma simples execução.

Os AEs, nas suas configurações usuais, também apresentam dificuldades para a determinação do ótimo global, sem a utilização de uma metodologia de otimização local. Outra limitação é a necessidade de análise de todas as amostras do processo a cada

avaliação da função de aptidão. Este aspecto é uma limitação relevante para aplicações de controle em tempo real.

Uma observação relevante quanto aos *AEs* é apresentada no teorema *no free lunch* (*NFL*) (WOLPERT & MACREADY, 1997): não existe algoritmo para a resolução de todos problemas de otimização que seja genericamente (em média) superior que outro algoritmo competidor. Segundo o *NFL*, a afirmação de que os *AEs* são inferiores ou superiores a algum método alternativo é “insensata”. O que pode ser afirmado somente é que *AEs* comportam-se melhor que outros métodos com respeito a resolução de uma classe específica de problemas, e como consequência comportam-se inadequadamente para outras classes de problemas.

O teorema *NFL* pode ser confirmado pela análise dos *AEs* em relação a muitos métodos clássicos de otimização. Os métodos clássicos são mais eficientes para resolução de problemas lineares, quadráticos, fortemente convexos, unimodais, separáveis, e em muitos outros problemas em especial. Por outro lado, os *AEs* têm sido utilizados nos mais diversos problemas principalmente quando estes são descontínuos, não diferenciáveis, multimodais, ruidosos, e quando superfícies de resposta não convencionais são envolvidas (BÄCK *et al.*, 1997a, WOLPERT & MACREADY, 1997, SCHWEFEL, 1997).

As consequências deste fato são a existência de uma dicotomia entre eficiência e aplicabilidade geral, entre propriedades de convergência e esforço computacional dos algoritmos para a resolução de problemas. Algum conhecimento específico sobre a situação que está sendo tratada pode ser utilizado para especificação de um algoritmo adequado a solução do problema. Entretanto, não existe um método que resolva todos os problemas efetivamente tão bem quanto eficientemente, pois estas metas são contraditórias (COELHO & COELHO, 1999c).

2.4 Aplicações industriais

As aplicações industriais de *AEs* é uma realidade na Europa e Estados Unidos, na resolução de problemas de otimização, gerenciamento, economia, projeto, roteamento, escalonamento e reconhecimento de padrões. Os exemplos da aplicação de *AEs* existem

em diversas áreas do conhecimento, contudo algumas das mais relevantes aplicações são resumidas nas Tabelas A.1 e A.2 (EVONEWS, 1996, 1999, COELHO & COELHO, 1999c), apresentadas no apêndice A.

Quanto às aplicações na área acadêmica têm-se importantes abordagens em projeto de sistemas de controle, controle robusto, identificação de sistemas, otimização com múltiplos objetivos, detecção de falhas, projeto evolutivo de *hardware* e integração de sistemas (DAVIS, 1991, ALANDER, 1995, CHIPPERFIELD & FLEMING, 1996, LINKENS & NYONGESA, 1995, HIGUCHI & KAJIHARA, 1999). Um apanhado abrangente de aplicações em identificação e controle de processos são apresentados em ALANDER (1996) e COELHO & COELHO (1999c).

Muitos dos fundamentos e características dos *AEs* para a configuração em procedimentos de identificação e controle de processos são apresentados a seguir e estão baseados em COELHO & COELHO (1997a, 1997b, 1997d, 1997e, 1997g, 1997j, 1997k, 1998c, 1999c).

2.5 Algumas formulações e terminologia dos *AEs* em problemas de otimização

2.5.1 Espaço de busca

Um problema de busca é caracterizado por um espaço de busca, S — o conjunto de objetos sobre os quais a busca é conduzida — e uma função objetivo, ou seja, o mapeamento de S para o espaço de valores da função objetivo, R , isto é

$$f: S \rightarrow R \quad (2.1)$$

Embora as situações usuais apareçam quando $R = \mathbb{R}$ (o conjunto dos números reais) e a meta é a minimização ou maximização de f . Contudo, R poderia ser um vetor dos valores das funções objetivo, isto em problemas de otimização com múltiplos objetivos. Observa-se que o conjunto de todos os mapeamentos de S em R são denotados por R^S , assim $f \in R^S$ (RADCLIFFE & SURRY, 1995).

Em outras palavras, as aplicações dos *AEs*, em tarefas de otimização (sem restrições) de parâmetros, requerem que o problema de minimização seja estabelecido por uma dada função $f(x)$ e um conjunto $D \in \mathbb{R}^{np}$, encontrando um x^* tal que:

$$f_{np}(x^*) = \inf_{x \in D} f_{np}(x) \quad (2.2)$$

onde np é o número de parâmetros do problema e $x = (x_1, x_2, \dots, x_{np})^T$ é um vetor. Para problemas de maximização usualmente substitui-se f por $-f$. Algumas mudanças podem ser realizadas na função objetivo para adequar-se ao tratamento de restrições ou à inclusão de termos de punitivos para que a eliminação de soluções não-factíveis.

2.5.2 Representação

A seleção de uma representação apropriada é essencial para a busca. A representação de S é um conjunto C (o espaço de representação ou espaço dos cromossomos) de tamanho pelo menos igual a S , junto com uma função g que mapeia C em S , tal que

$$g: C \rightarrow S \quad (2.3)$$

Assim, os valores obtidos para a função objetivo são associados com os pontos no espaço de representação C , através da composição de f com g ,

$$f \circ g: C \rightarrow R \quad (2.4)$$

que é regida por

$$f \circ g(x) \stackrel{\Delta}{=} f(g(x)) \quad (2.5)$$

As relações entre C , S e R são apresentadas na figura 2.1 (RADCLIFFE & SURRY, 1995).

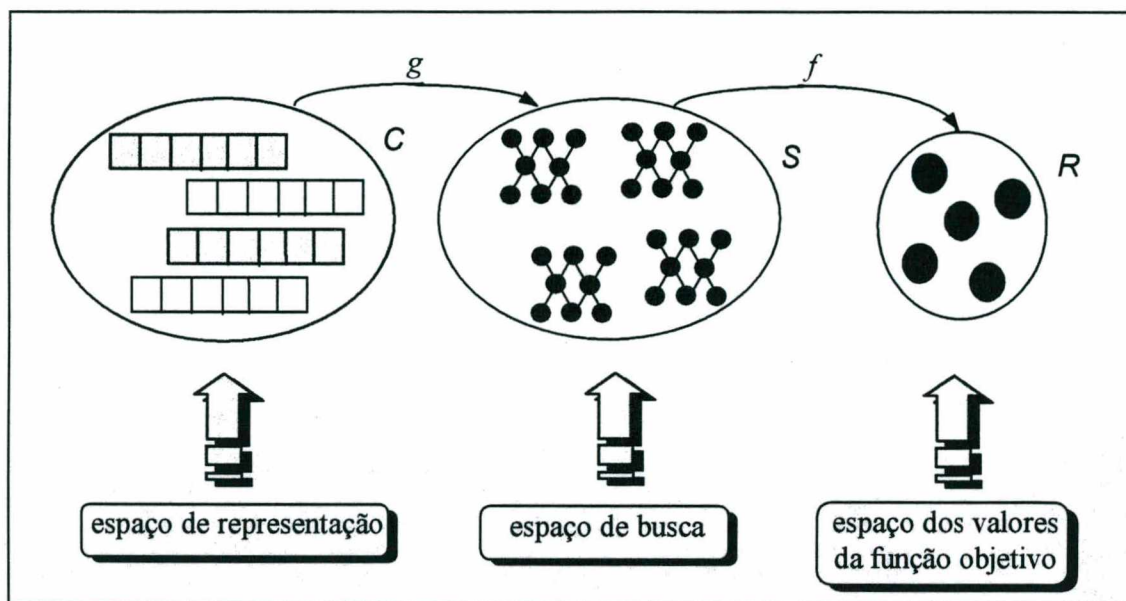


Figura 2.1. Relações entre C, S e R.

2.6 Algoritmos genéticos

John Holland e alguns de seus colaboradores da *University of Michigan* estavam interessados em sistemas complexos artificiais, capazes de adaptarem-se a mudanças de condições ambientais. Sob este ponto de vista, a necessidade da estruturação de sistemas com mecanismos de auto-adaptação é enfatizada. Uma população de indivíduos, para adaptar-se coletivamente em um ambiente, deve comportar-se como um sistema natural, onde a sobrevivência é promovida eliminando-se os comportamentos inúteis (ou prejudiciais) e recompensando-se os comportamentos úteis.

HOLLAND (1975) compreendeu que os mecanismos biológicos permitiam a adaptação do sistema de forma que poderiam ser expressas matematicamente, e simuladas em um computador. Esta abstração originou os algoritmos genéticos (AGs). A ligação entre a busca atual, o problema de otimização e o AG é o indivíduo. Cada indivíduo representa uma solução factível em um mesmo espaço de busca, que pode ser verificado através de um mapeamento apropriado. O mapeamento do espaço de busca, para os indivíduos, e o mapeamento reverso foi realizado originalmente através de dígitos binários.

Os *strings* de *bits* são formas gerais e permitem a análise de alguns resultados teóricos sobre os *AGs*. Contudo, a codificação binária não é sempre a melhor escolha e outras representações são possíveis.

Em síntese, HOLLAND formulou originalmente os *AGs* e providenciou os fundamentos teóricos para a representação binária. Trabalhos relevantes sobre outras representações também são encontrados na literatura, tais como GOLDBERG (1989), MICHALEWICZ (1992), FOGEL (1994) e BÄCK *et al.* (1997a).

Os *AGs* possuem procedimentos probabilísticos de busca, baseados nos princípios decorrentes da dinâmica das populações naturais. Nos procedimentos de busca, uma população de soluções candidatas é aleatoriamente gerada e “evoluem” para uma solução, através da aplicação de operadores genéticos. Os três operadores usualmente empregados em *AGs* são: seleção, recombinação e mutação (GOLDBERG, 1989).

Um *AG* é projetado, na sua configuração básica, conforme as seguintes etapas (BÄCK *et al.*, 1997a, FOGEL, 1995, MICHALEWICZ, 1992):

- (i) criar a população inicial de parâmetros compreendendo N_{ind} indivíduos (soluções para o problema). Cada uma das soluções consiste de vetores $x_i \in \{0,1\}$ (representação canônica) ou $x_i \in \mathbb{R}$ (representação real). Estes parâmetros são inicializados aleatoriamente, de acordo com uma distribuição uniforme;
- (ii) classificar cada solução x_i , $i=1, \dots, N_{ind}$, com relação ao cálculo da função de aptidão, ou seja, avalia-se o grau de adaptação de cada indivíduo da população em relação ao problema;
- (iii) selecionar os indivíduos mais aptos de acordo com uma estratégia de seleção;
- (iv) aplicar o operador genético de cruzamento (representação canônica) ou recombinação (representação por ponto flutuante);
- (v) aplicar o operador genético de mutação;
- (vi) gerar uma nova população; e
- (vii) repetir as etapas (ii) a (vi) até que um critério de convergência seja satisfeito.

As etapas mencionadas para os *AGs* podem ser sintetizadas pelo pseudocódigo do seu ciclo evolutivo, conforme a figura 2.2 (BÄCK & SCHWEFEL, 1993).

```

t := 0;
inicializar: P(0) := {x1(0), x2(0), ..., xα(0)}
avaliar: P(0) := {Φ(x1(0)), Φ(x2(0)), ..., Φ(xα(0))}
enquanto uma condição de parada não é satisfeita
{
  realizar recombinação: P'(t) := rΘc(P(t))
  realizar mutação: P'(t) := mΘm(P(t))
  avaliar: P'(t) := {Φ(x'1(t)), Φ(x'2(t)), ..., Φ(x'α(t))}
  selecionar: P(t+1) := sΘs(P'(t))
  t := t + 1;
}

```

Figura 2.2. Pseudocódigo básico dos AGs.

As convenções utilizadas na figura 2.2, são as seguintes:

x : indivíduo da população antiga;

x' : indivíduo da população atual;

α : número de indivíduos da população;

$P(t) := \{x_1(t), x_2(t), \dots, x_\alpha(t)\}$: população na geração t ;

$P'(t) := \{x'_1(t), x'_2(t), \dots, x'_\alpha(t)\}$: população atual na geração t ;

$\Phi: I \rightarrow \mathbb{R}$: mapeamento da função de aptidão;

$m_{\Theta m}$: operador de mutação com parâmetro de controle Θm ;

$m_{\Theta c}$: operador de cruzamento (ou recombinação) com parâmetro de controle Θc ;

$s_{\Theta s}$: operador de seleção $\ni s_{\Theta s}: I^{N_{ind}} \rightarrow I^{N_{ind}}$.

O operador genético de cruzamento — ou de forma análoga, o operador de recombinação — é responsável pela troca de material genético entre os indivíduos, com maior probabilidade de reproduzirem os indivíduos mais aptos ao ambiente.

O operador de mutação modifica o valor dos genes do indivíduo e visa restaurar o material genético perdido ou não explorado em uma população. Este operador, quando projetado de forma apropriada, pode prevenir a convergência prematura do AG para soluções sub-ótimas e manter a diversidade da população.

A definição dos parâmetros intrínsecos aos AGs — *parâmetros de controle do AG* — geralmente são determinados heurísticamente, tais como: tamanho da população, tamanho da estrutura dos cromossomos, probabilidade de cruzamento, probabilidade de mutação e tipos dos operadores genéticos a serem adotados (FOGEL, 1995).

2.6.1 Operador de seleção

O operador de seleção emprega o princípio de sobrevivência dos indivíduos mais aptos, através de uma metáfora aos procedimentos de reprodução assexuada e seleção natural, de acordo com o grau de adaptação do indivíduo ao ambiente. O objetivo básico do operador de seleção é enfatizar as melhores soluções que constituem uma população. O operador não cria nenhuma nova solução. Este operador seleciona as soluções relativamente aptas de uma população e remove as soluções remanescentes (BÄCK *et al.*, 1997a).

O operador de seleção é uma combinação de dois conceitos diferentes — reprodução e seleção. As múltiplas cópias de uma solução (indivíduo) são alocadas em uma população pela remoção de algumas soluções inferiores. Entretanto, alguns estudos de *AEs* utilizam ambos conceitos de reprodução e seleção, simultaneamente, em um operador, e outros os utilizam separadamente (BÄCK *et al.*, 1997a).

A verificação se uma solução é apta (ou não) em uma população é baseada no valor da função de aptidão da referida solução. Para que uma solução tenha maior aptidão deve ter uma alta probabilidade de seleção. Contudo, os operadores de seleção diferem na maneira que as cópias são designadas para serem as melhores soluções.

Os operadores de seleção incluem a reprodução e caracterizam-se por um parâmetro denominado *pressão seletiva*, que relata o tempo de “preenchimento” do operador de seleção. O tempo de “preenchimento” é definido como a velocidade com que a melhor solução, na população inicial, pode ocupar toda a população através da aplicação repetida do operador de seleção (GOLDBERG, 1989, GOLDBERG & DEB, 1991).

O operador de seleção, quando apresenta uma grande pressão seletiva faz com que a população perca a diversidade rapidamente, ocasionando uma convergência prematura, para uma solução inadequada. Contudo, um operador de seleção com pressão seletiva pequena apresenta uma baixa convergência e permite aos operadores de recombinação e mutação iterações suficientes à busca no espaço de soluções (GOLDBERG & DEB, 1991). As noções de alguns operadores de seleção são apresentadas e discutidas a seguir.

(i) *seleção proporcional*: este mecanismo utiliza uma distribuição de probabilidade tal que a probabilidade de seleção de um dado indivíduo, para reprodução, é proporcional à função de aptidão do indivíduo. Assim, obtida a função de aptidão, f_i , de cada indivíduo em uma dada geração, obtém-se

$$F_T = \sum_{i=1}^{Nind} f_i(x) \quad (2.6)$$

que representa a aptidão total da população. Após, a probabilidade de seleção, p_i , é atribuída para cada indivíduo pela equação:

$$p_i(x) = \frac{f_i(x)}{F_T} \quad (2.7)$$

Finalmente, uma probabilidade acumulada para cada indivíduo é obtida pela soma das funções de aptidão dos membros da população com classificação inferior à sua, ou seja,

$$c_i = \sum_{k=1}^i p_k, \quad i = 1, \dots, Nind \quad (2.8)$$

Um número r uniformemente distribuído em $[0,1]$ é obtido $Nind$ vezes e a cada tempo o i -ésimo *string* é selecionado tal que $c_{i-1} < r \leq c_i$. Quando $r < c_1$, o primeiro indivíduo é selecionado. Este procedimento pode ser visualizado por meio de uma roleta com $Nind$ partes, onde cada parte tem tamanho proporcional à aptidão do indivíduo. Para ilustrar este exemplo, tomam-se valores de $p_1 = 0,10$; $p_2 = 0,40$; $p_3 = 0,30$ e $p_4 = 0,20$. Assim têm-se: $c_1 = 0,10$; $c_2 = 0,50$; $c_3 = 0,80$ e $c_4 = 1,00$. Agora, seja $r = 0,07$ o número gerado. Já que $r < c_1$, o indivíduo 1 será selecionado. Se r fosse 0,93 então o indivíduo 4 seria selecionado, pois $c_3 < r \leq c_4$. Um pseudocódigo da seleção proporcional é apresentado na figura 2.3 (BLICKLE & THIELE, 1995).

```

dados de entrada: a população  $P(t)$ 
dados de saída: a população após a realização da seleção,  $P'(t)$ 
torneio ( $P_1, \dots, P_{Nind}$ )
   $c_0 \leftarrow 0$ ;
  para  $i \leftarrow 1$  até  $Nind$  faça
     $c_i \leftarrow c_{i-1} + f_i/F_i$ ;
  fim do para  $i$ 
  para  $j \leftarrow 1$  até  $Nind$  faça
     $r \leftarrow$  aleatório  $[0,1[$ 
     $P'_j \leftarrow P_l$ , tal que  $c_{l-1} \leq r < c_l$       /*  $l$ : comprimento do string */
  fim do para  $j$ 
retorna  $\{P_1, P_2, \dots, P_{Nind}\}$ 

```

Figura 2.3. Algoritmo de seleção proporcional.

Uma variante da seleção proporcional é a seleção por *ranking*. Os métodos de *ranking* requerem somente o valor da função de aptidão, para mapear as soluções em um conjunto parcialmente ordenado. Um exemplo desta forma de seleção é a seleção por *rank* geométrico normalizado (JONES & HOUCK, 1994), onde a probabilidade de seleção de um indivíduo é dada por:

$$P(x_i) = q'(1-q)^{ra-1}, \quad i = 1, \dots, Nind \quad (2.9)$$

$$q' = \frac{q}{1 - (1-q)^{Nind}} \quad (2.10)$$

onde q é probabilidade de selecionar o melhor indivíduo, $ra(i)$ é o *rank* do indivíduo, sendo $ra(\cdot)=1$ o melhor *rank*.

(ii) *seleção elitista com truncamento*: o mecanismo de seleção elitista com truncamento consiste na seleção, em cada geração, dos β melhores indivíduos da população (β é o coeficiente de truncamento $\in [0;1]$). Contudo, β tem valores típicos entre 0,1 e 0,5. Entre os β melhores indivíduos são mantidos os indivíduos mais aptos na próxima geração, e destes são selecionados os indivíduos que passarão pelas operações de recombinação e mutação (DE FALCO *et al.*, 1996). Um pseudocódigo da seleção elitista do tipo *breeder* com truncamento é apresentado na figura 2.4.

```

dados de entrada: a população  $P(t)$ ,  $\beta \in [0;1]$ 
dados de saída: a população após a realização da seleção,  $P'(t)$ 
elitista ( $\beta, P_1, \dots, P_{Nind}$ )
 $\bar{P} \leftarrow$  população classificada de acordo com a aptidão
para  $i \leftarrow 1$  até  $Nind$  faça
   $r \leftarrow$  aleatório  $\{1, \dots, \beta_{Nind}\}$ 
   $P'_i \leftarrow \bar{P}_r$ 
fim do para  $i$ 
retorna  $\{P_1, P_2, \dots, P_{Nind}\}$ 

```

Figura 2.4. Algoritmo de seleção elitista com truncamento.

(iii) *seleção por torneio*: nesta forma de seleção um grupo de q indivíduos é aleatoriamente escolhido. Esta forma de seleção pode ser projetada da população com ou sem reposição. O grupo ocupa parte de um torneio, onde o indivíduo vencedor é determinado de acordo com o valor da função de aptidão. O melhor indivíduo é escolhido deterministicamente, ainda que uma seleção estocástica possa ser realizada. Em ambos os casos, somente o vencedor é inserido na população da próxima geração e o procedimento é repetido γ vezes para formar a nova população. Os torneios são realizados, freqüentemente, entre dois indivíduos (torneio binário). Contudo, isto pode ser generalizado para um grupo com tamanho arbitrário, q , denominado tamanho do torneio. Um pseudocódigo da seleção através de torneio é apresentado na figura 2.5 (BLICKLE & THIELE, 1995).

```

dados de entrada: a população  $P(t)$ ;  $q \in \{1, 2, \dots, Nind\}$ 
dados de saída: a população após a realização da seleção,  $P'(t)$ 
torneio ( $q, P_1, \dots, P_{Nind}$ )
para  $k \leftarrow 1$  até  $Nind$  faça
   $P(x_k) \leftarrow$  indivíduo mais apto escolhido aleatoriamente de  $q$  indivíduos de  $\{P_1, \dots, P_{Nind}\}$ ;
fim do para  $k$ 
retorna  $\{P_1, P_2, \dots, P_{Nind}\}$ 

```

Figura 2.5. Algoritmo de seleção através de torneio.

Alguns operadores de seleção foram mencionados, mas outros tipos de seleção podem ser utilizados em AEs. Outros exemplos de seleção são extensões da seleção por

roleta, *steady-state*, amostragem estocástica, Boltzmann, formas de truncamento e modelos elitistas (GOLDBERG, 1989, BÄCK *et al.*, 1997a). Os operadores de seleção por roleta e *breeder* são utilizados na tese. Os operadores de recombinação e mutação diferem quanto à estrutura e implementação, de acordo com o tipo de *AG* adotado, seja este com representação inteira, canônica ou real. As duas últimas representações são tratadas e descritas a seguir.

2.6.2 Tipos de representação

Os *AGs*, com a representação inteira, são usualmente projetados para problemas de escalonamento e do tipo caixeiro viajante (MICHALEWICZ, 1992). Os *AG* com a representação canônica e representação real têm sido aplicados nos mais diversos problemas de identificação e controle de processos. Os tradicionais *AGs*, denominados *AGs* canônicos, baseiam-se em noções do teorema de esquemas (*schema theorem*) e construção de blocos (*building blocks*). Os indivíduos são representados por vetores binários (HOLLAND, 1975). Entretanto, esta representação não é universalmente aceita na literatura. Alguns pesquisadores indicam que a representação de ponto flutuante (números reais) apresenta melhor desempenho em relação a representação binária em aplicações que necessitam do tratamento de valores reais, pois apresenta maior compatibilidade, precisão e rapidez de execução. A representação binária é apropriada em aplicações que requeiram o tratamento de valores discretos (DAVIS, 1991, MICHALEWICZ, 1992). Contudo, o procedimento básico de otimização por *AGs*, seja este com representação canônica ou real, utiliza os três operadores básicos.

2.6.2.1 Algoritmo genético canônico

Os *AGs* canônicos constituem-se de uma população dos *strings* de *bits* modificadas por operadores genéticos. Cada cromossomo é composto de N_d *substrings* constituídos de valores 0s e 1s. Os *strings* de comprimento $N_{bit(s)}$ (comprimento igual a 1), $N_{bit(i)}$ e $N_{bit(f)}$, são

utilizados para codificar o *bit* de sinal, valores inteiros e fracionários das variáveis, x , da população, respectivamente. A figura 2.6 ilustra um cromossomo que abrange duas variáveis com representação binária e a decodificação de seu valor da base binária para a base decimal.

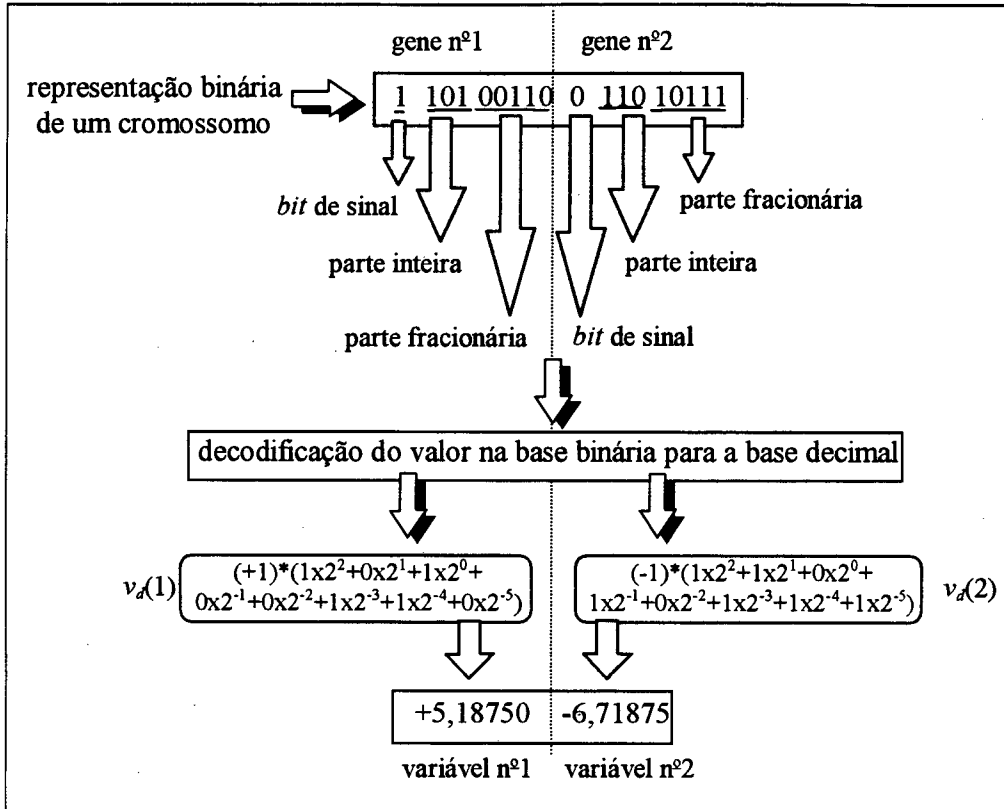


Figura 2.6. Representação de um cromossomo nos AGs canônicos.

O cromossomo da figura 2.6 tem $N_{df} = 2$, $N_{bit(s)} = 1$, $N_{bit(i)} = 3$, e $N_{bit(f)} = 5$. Cada *substring* binário de comprimento $(N_{bit(s)} + N_{bit(i)} + N_{bit(f)})$ representa um número com a representação em ponto flutuante, que pode ser descrito por

$$v_{d(k)} = sn(k) * \left[\sum_{j=1}^{N_{bit(i)}} \delta_j^{(k)} 2^{N_{bit(i)}-j} + \sum_{i=1}^{N_{bit(f)}} \delta_{i+N_{bit(i)}}^{(k)} 2^{-N_{bit(i)}-N_{bit(f)}+i+2} \right], \quad (2.11)$$

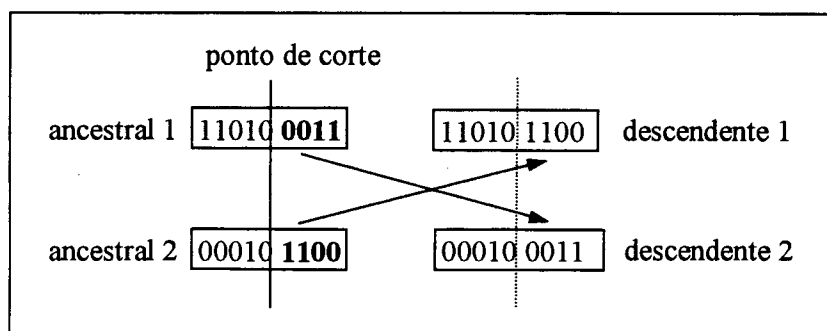
$k = 1, N_{df}$

onde $\delta_j^{(k)}$ (0 ou 1) representa o j -ésimo *locus* (ou posição) do k -ésimo *substring* binário, $sn(k)$ representa um indicador do *bit* de sinal. Este *bit* pode ser igual 1 ($sn(k)=+1$), ou caso contrário, quando o *bit* de sinal é igual 0 ($sn(k)=-1$).

Cada *bit* (ou mesmo grupos de *bits*) representa um valor da mesma variável do problema (*gene*). No caso do exemplo, ilustrado na figura 2.2, os *strings* (000000000) e (111111111) representam os limites inferiores e superiores do intervalo de busca. Necessitando aumentar a precisão requerida, um *string* N_{bit} com comprimento maior apresenta maior precisão para representação dos cromossomos. Um caso particular da representação binária é a representação inteira que pode ser configurada quando $N_{bit} = 0$.

2.6.2.1.1 Operador de cruzamento

O operador de cruzamento é o operador principal dos AGs, e consiste na troca de partes dos cromossomos entre os indivíduos. O(s) ponto(s) de cruzamento são escolhidos aleatoriamente. As partes de dois cromossomos ancestrais, à direita do ponto de cruzamento são trocadas, para a formação dos cromossomos descendentes. O cruzamento não é realizado em cada par de indivíduos, pois sua frequência de utilização é controlada através da sua probabilidade (geralmente tem valor próximo de 1, ou seja, 100%). O operador de cruzamento com um e dois pontos de corte são representados na figura 2.7(a) e 2.7(b), respectivamente.



(a) um ponto de corte

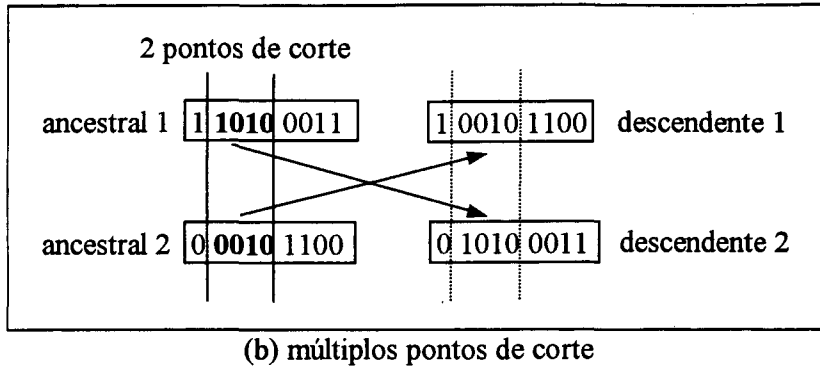


Figura 2.7. Operador de cruzamento com um ou vários pontos de corte.

Outro operador de cruzamento é o cruzamento uniforme. No cruzamento uniforme não seleciona-se um conjunto de pontos de corte, mas considera-se cada posição dos *bits* dos dois ancestrais e troca-se dois *bits* com uma probabilidade de 50% (podem ser considerados outros valores). Assim, supondo-se que as posições 2 e 5 dos ancestrais são trocadas, obtém a figura 2.8.

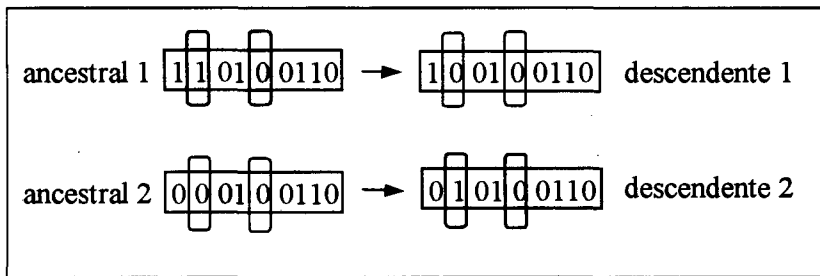


Figura 2.8. Operador de cruzamento uniforme.

2.6.2.1.2 Operador de mutação

O operador de mutação nos *AGs* é introduzido para prevenir a convergência prematura para ótimos locais, através da amostragem de novos pontos no espaço de busca. O operador de mutação é aplicado, a cada descendente individualmente, após a aplicação do operador de cruzamento. Consiste na mudança aleatória de uma parte do *string*, que representa o indivíduo (normalmente um *bit*), apresentando pequena probabilidade (MAN *et al.*, 1996). O operador de mutação é representado na figura 2.9.

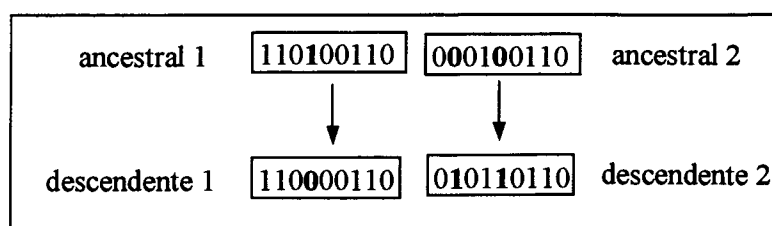


Figura 2.9. Operador de mutação nos *AGs* canônicos.

2.6.2.2 Algoritmo genético com representação em ponto flutuante

A seguir, apresenta-se uma descrição dos operadores de mutação e recombinação para configuração de *AGs* com representação em ponto flutuante.

2.6.2.2.1 Operador de mutação

Os tipos de mutação, propostos por MICHALEWICZ (1992), são os seguintes:

i) *mutação uniforme*: seja $x = (x_1, \dots, x_k, \dots, x_q)$, um indivíduo, a mutação uniforme gera um novo indivíduo a partir de outro. Após ser selecionado um indivíduo x , escolhe-se aleatoriamente um componente k , deste indivíduo para obtenção de um novo valor, tal que:

$$x' = (x_1, \dots, x'_k, \dots, x_q) \quad (2.12)$$

onde x'_k é um valor aleatório selecionado dentro dos limites do parâmetro k (FARDIN *et al.*, 1997).

(ii) *mutação não-uniforme*: o componente x_k é selecionado para mutação, resultando num vetor $x' = (x_1, \dots, x'_k, \dots, x_q)$, tal que:

$$x'_k = \begin{cases} x_k + \Delta(t, \max \lim(k) - x_k), & \text{se } z = 0, \\ x_k + \Delta(t, x_k - \min \lim(k)), & \text{se } z = 1 \end{cases} \quad (2.13)$$

onde z é um dígito binário aleatório (0 ou 1), $\text{minlim}(k)$ e $\text{maxlim}(k)$ são os valores mínimo e máximo dos limites do parâmetro x'_k , respectivamente. A função $\Delta(t,y)$ retorna um valor no intervalo $[0,y]$ tal que a probabilidade de $\Delta(t,y)$ inicia em zero e é incrementada de acordo com o número de gerações t , tal que:

$$\Delta(t,y) = y \cdot ra \cdot \left[1 - \frac{t}{Nger} \right]^b \quad (2.14)$$

onde ra é um número gerado aleatoriamente no intervalo $[0;1]$, $Nger$ é o número máximo de gerações e b é um parâmetro escolhido pelo usuário, que determina o grau de dependência com o número de gerações. Esta propriedade leva o operador a efetuar uma busca uniforme no espaço inicial, quando t é pequeno e, mais localmente nas gerações posteriores.

iii) *mutação de contorno*: nesta mutação seleciona-se aleatoriamente uma variável j com distribuição uniforme, em $[0;1]$, e após atribui-se a variável j o limite máximo ou mínimo do parâmetro. O procedimento é resumido em:

$$x'_k = \begin{cases} \text{maxlim}(k), & \text{se } k = j \text{ e } ra < 0,5 \\ \text{minlim}(k), & \text{se } k = j \text{ e } ra \geq 0,5 \\ x_k, & \text{para os outros valores} \end{cases} \quad (2.15)$$

2.6.2.2.2 Operador de recombinação

Os tipos de operadores de recombinação, para a representação em ponto flutuante, proposto por MICHALEWICZ (1992), são os seguintes:

(i) *recombinação simples*: é selecionado um par de indivíduos $x = (x_1, \dots, x_k, x_{k+1}, \dots, x_q)$ e $y = (y_1, \dots, y_k, y_{k+1}, \dots, y_q)$ recombinados a partir do k -ésimo parâmetro. Os indivíduos descendentes resultantes desta recombinação são:

$$\begin{cases} x = (x_1, \dots, x_k, y_{k+1}, \dots, y_q) \\ y = (y_1, \dots, y_k, x_{k+1}, \dots, x_q) \end{cases} \quad (2.16)$$

(ii) *recombinação aritmética*: define a recombinação como uma combinação linear de dois vetores, $x = (x_1, \dots, x_k, x_{k+1}, \dots, x_q)$ e $y = (y_1, \dots, y_k, y_{k+1}, \dots, y_q)$. Após esta operação tem-se os descendentes descritos por:

$$\begin{cases} x = \alpha_{ra} \cdot x + (1 - \alpha_{ra}) \cdot y \\ y = \alpha_{ra} \cdot y + (1 - \alpha_{ra}) \cdot x \end{cases} \quad (2.17)$$

Neste operador pode-se ter um parâmetro $\alpha_{ra} \in [0;1]$, que é uma constante (recombinação aritmética uniforme) ou uma variável que depende do tamanho da população (recombinação aritmética não-uniforme).

(iii) *recombinação heurística*: gera apenas um (ou mesmo nenhum) novo indivíduo, através da extrapolação linear de dois indivíduos. Considerando-se $x = (x_1, \dots, x_k, x_{k+1}, \dots, x_q)$ e $y = (y_1, \dots, y_k, y_{k+1}, \dots, y_q)$, após esta operação tem-se um indivíduo resultante factível regido pela expressão:

$$z = ra \cdot (y - x) + y \quad (2.18)$$

onde ra é um número aleatório, tal que $ra \in [0;1]$. Caso a solução gerada não seja factível, deve-se gerar um novo número aleatório para a obtenção de um novo indivíduo.

2.6.3 Configuração dos operadores, sintonia e controle dos parâmetros nos AEs

EIBEN *et al.* (1999) definem duas formas de configurar os valores dos parâmetros dos AEs:

- (i) *sintonia de parâmetros*: parâmetros configurados *antes* de executar o AE; e
- (ii) *controle de parâmetros*: modificação dos valores dos parâmetros *durante* a execução do AE.

As técnicas de controle de parâmetros levam em conta diversos questionamentos, tais como (EIBEN *et al.*, 1999):

- (i) *o que é modificado ?* (exemplo: representação, operadores, procedimento de seleção, mutação, ...);
- (ii) *de que forma a modificação é realizada ?* (exemplo: heurística determinística, heurística baseada em realimentação ou auto-adaptativa);
- (iii) *qual o escopo da modificação ?* (exemplo: nível populacional, nível individual, ...);
- (iv) *qual a evidência de que a modificação foi realizada ?* (exemplo: monitoramento do desempenho dos operadores, diversidade da população, ...).

Apesar de intensas pesquisas na área, não existe uma regra única para estipular o tamanho da população, nem mesmo para a probabilidade de aplicação dos operadores genéticos. A literatura menciona que as configurações adotadas em AGs (representação binária) utilizam usualmente a seguinte sintonia de parâmetros do AG: tamanho da população entre 30 e 200, probabilidade de cruzamento entre 0,5 e 1,0 e probabilidade de mutação entre 0,001 e 0,05 (DAVIS, 1991, SRINIVAS & PATNAIK, 1994b).

Numa população pequena é relativamente fácil a ocorrência de uma convergência prematura, devido à ocorrência de cromossomos muito dominantes ou recessivos, na busca pelo espaço de solução. Muitas vezes, diversos experimentos ou mesmo procedimentos de tentativa e erro, são realizados para obtenção de valores adequados para os parâmetros de probabilidade e tamanho de população.

Em síntese, pode-se dizer que a sintonia de parâmetros, através de tentativa e erro, é uma prática usual em AEs. Um parâmetro de controle sintonizado pode ocasionar escolhas sub-ótimas, já que os parâmetros interagem freqüentemente de uma forma complexa. A sintonia simultânea de mais parâmetros, contudo, leva a uma quantidade excessiva de experimentos para obter uma relação adequada entre eles. Segundo EIBEN *et al.* (1999), as desvantagens da sintonia de parâmetros baseada em experimentação são resumidas pelos seguintes aspectos:

- (i) os parâmetros de controle não são independentes, mas a obtenção sistemática de todas as combinações é praticamente impossível;
- (ii) o procedimento de sintonia de parâmetros consome tempo;
- (iii) os valores dos parâmetros selecionados não são necessariamente ótimos.

A seguir, visando-se a obtenção de um procedimento eficiente, propostas de mecanismos adaptativos das probabilidades de cruzamento e mutação, e um operador

baseado em gradiente são apresentadas. As abordagens apresentadas podem ser combinadas para abranger a vantagem de cada técnica em um *AG* eficiente.

2.6.4 Abordagem de ajuste automático das probabilidades de cruzamento e mutação

A configuração das probabilidades de cruzamento, p_c , e mutação, p_m , de forma adaptativa é realizada com dois objetivos: manter a diversidade da população e a capacidade de convergência apropriada dos *AGs*. A proposta de SRINIVAS & PATNAIK (1994a) é a determinação das probabilidades dos operadores de cruzamento e mutação, que variam de acordo com os valores da função de aptidão dos indivíduos da população. Neste caso, as expressões para p_c e p_m são regidas pelas seguintes equações:

$$p_c(t) = \begin{cases} k_1 \frac{f_{\max}(t) - f'(t)}{f_{\max}(t) - \bar{f}(t)}, & \text{para } f'(t) \geq \bar{f}(t) \\ k_3, & \text{senão} \end{cases} \quad (2.19)$$

$$p_m(t) = \begin{cases} k_2 \frac{f_{\max}(t) - f'(t)}{f_{\max}(t) - \bar{f}(t)}, & \text{para } f'(t) \geq \bar{f}(t) \\ k_4, & \text{senão} \end{cases} \quad (2.20)$$

onde k_1 , k_2 , k_3 e $k_4 \leq 1,0$ são parâmetros de projeto, $\bar{f}(t)$ é o valor de aptidão média da população em relação ao valor de aptidão máximo, f_{\max} , da população, $f'(t)$ é o maior dos valores de aptidão dos dois indivíduos a serem combinados ou o indivíduo a ser realizada mutação. Segundo SRINIVAS & PATNAIK (1994a), os valores de $k_1 = k_3 = 1,0$ e $k_2 = k_4 = 0,5$ são escolhas adequadas.

2.6.5 Abordagem com probabilidade de mutação variável

A abordagem de KIM *et al.* (1997a) adota uma probabilidade de mutação variável, durante o ciclo evolutivo regida pela função:

$$p_m(t) = \alpha_{p_m} * e^{(-\beta_{p_m} t)}, \text{ onde } \alpha_{p_m} \leq 0,1 \text{ e } \beta_{p_m} \leq 0,1 \quad (2.21)$$

onde α_{p_m} e β_{p_m} são parâmetros de projeto. Esta abordagem visa um aprimoramento da qualidade da solução, através um operador de mutação mais eficiente.

2.6.6 Abordagem com operador de reprodução baseado em gradiente

Os pesquisadores PHAM & JIN (1995) propuseram um novo operador, denominado pelos autores, de operador de reprodução baseado em gradiente. Este operador visa acrescentar aos AGs uma natureza combinando características determinísticas e estocásticas. O novo operador é empregado para superar as limitações usualmente associadas ao método de seleção por roleta. Estas limitações incluem: perda do indivíduo mais apto da população, dominância excessiva de indivíduos muito aptos (elite dominante) e a habilidade limitada de explorar novos pontos no espaço de busca. Para a implementação deste operador um novo vetor de soluções é calculado pela equação:

$$x'(t) = x(t) + \psi_i \frac{[f_{max}(t) - f_i(t)]}{f_{max}(t)} [x_b(t) - x_i(t)], \quad i = 1, \dots, N_{ind} \quad (2.22)$$

onde ψ_i constitui-se de um coeficiente positivo, $x_b(t)$ é o vetor de parâmetros com maior aptidão, $f_{max}(t)$. PHAM & JIN (1995) recomendam ψ_i entre 0 e 2 para um comportamento adequado do operador de reprodução. Este operador pode ser utilizado para aprimorar a qualidade da solução obtida, a cada geração, após aplicar-se os operadores de cruzamento e mutação.

2.7 Estratégias evolutivas

As estratégias evolutivas (EEs), inicialmente, foram desenvolvidas para a resolução de problemas de otimização em engenharia. A primeira EE desenvolvida foi a EE-(1+1),

proposta por I. Rechenberg e H. P. Schwefel, nos anos 60, no *Hermann Föttinger Institute for Hydrodynamics* (Universidade Técnica de Berlin, Alemanha), em experimentos com um processo túnel de vento. A $EE-(1+1)$ original utiliza somente o operador de mutação, onde apenas um ancestral produz um único descendente (BÄCK *et al.*, 1997a).

A $EE-(1+1)$ foi progressivamente generalizada em variantes do número de ancestrais (pais), $\mu > 1$, e número de descendentes (filhos), $\lambda > 1$, por geração. As EEs com múltiplos membros têm o embasamento biológico relacionado às características de poligenia e pleiotropia. Estas EEs são divididas de acordo com o mecanismo de seleção em:

- (i) estratégia soma (*plus strategy*) — $EE-(\mu+\lambda)$ — Os μ ancestrais geram λ descendentes. Após os μ ancestrais e os λ descendentes competem pela sobrevivência;
- (ii) estratégia vírgula (*comma strategy*) — $EE-(\mu,\lambda)$ — Os λ descendentes competem para sobreviver e os ancestrais são completamente substituídos a cada geração (SOUCEK & IRIS GROUP, 1992).

A $EE-(\mu,\lambda)$ tem a tendência de manter uma maior diversidade de indivíduos na população, o que pode ser uma vantagem de forma a evitar-se mínimos locais. A desvantagem da $EE-(\mu,\lambda)$ é a possibilidade de uma solução “ótima” obtida, em uma dada geração, não “sobreviver” até o final do procedimento evolutivo.

A $EE-(\mu+\lambda)$, ao contrário, da $EE-(\mu,\lambda)$, dependendo de sua configuração pode ser mais susceptível a mínimos locais, ou seja, o domínio de uma elite de ancestrais (soluções similares) antes da obtenção de um valor adequado. Entretanto, uma solução ótima obtida, durante o procedimento evolutivo, não é perdida e “sobrevive” até que o final do procedimento evolutivo.

Os indivíduos (soluções), (x_i, σ_{xi}) , são diretamente representados por vetores de valores reais, $x_i \in \mathbb{R}^n$ e vetores de desvio padrão, $\sigma_{xi} \in \mathbb{R}^n$. Na criação da população inicial os vetores x_i e σ_{xi} são gerados aleatoriamente dentro de intervalos definidos pelo usuário.

O operador de mutação atua em cada *variável objeto*, x_i , pela adição de números aleatórios normalmente distribuídos com média zero e variância σ_{xi}^2 , regidos pela notação $N(0, \sigma_{xi}^2)$. Um novo vetor solução (x'_i, σ'_{xi}) pode ser criado através de uma regra de atualização com distribuição lognormal, tal que:

$$x_i'(t) = x_i(t) + \sigma_{xi} N(0,1), \quad i=1, \dots, Nind \quad (2.23)$$

$$\sigma_{xi}'(t) = \sigma_{xi}(t) \cdot \exp\left[\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)\right], \quad i=1, \dots, Nind \quad (2.24)$$

onde $N_i(0,1)$ é uma distribuição variável Gaussiana aleatória para cada componente i . A mutação de σ_{xi} é baseada em um fator de busca global $\tau' \cdot N(0,1)$ e um fator de busca local $\tau \cdot N_i(0,1)$. No caso do fator de busca global é gerado apenas um valor $N(0,1)$, a ser utilizado por todos os σ_{xi}^2 dos indivíduos na geração atual. Entretanto, no caso do fator de busca local, $N_i(0,1)$, é gerado um valor para cada indivíduo da população, com distribuição normal, média zero e variância σ_{xi}^2 . Estes fatores são regidos pelas seguintes equações:

$$\tau = \frac{1}{\sqrt{2 \cdot \sqrt{np}}} \quad (2.25)$$

$$\tau' = \frac{1}{\sqrt{2 * np}} \quad (2.26)$$

onde np é o número de dimensões da função que está sendo otimizada. Os operadores de recombinação são similares aos implementados em representação por ponto flutuante nos AGs (MICHALEWICZ, 1992). Entre as opções têm-se: a recombinação discreta, a intermediária (local e global) ou mesmo a não realização da operação de recombinação (BÄCK & SCHWEFEL, 1993).

A implementação de *EEs* apresenta diversas variantes, destacando-se as *EEs* contemporâneas e as *EEs* com mecanismos de auto-adaptação, com a realização de mutações correlacionadas (BÄCK & SCHWEFEL, 1993, BEYER, 1995). A seguir é descrita a implementação de *EEs* com mecanismos de auto-adaptação e mutações correlacionadas propostas por OSTERMEIER *et al.* (1994, 1995). Algumas aplicações em identificação e controle da proposta de OSTERMEIER *et al.* (1994,1995) foram apresentadas em COELHO & COELHO (1998a, 1998c, 1998d, 1999f) e KRÖHLING *et al.* (1999).

2.7.1 Mecanismo de auto-adaptação através de mutações correlacionadas

O desempenho das *EEs*, comparadas com outras técnicas de otimização, depende da configuração adequada dos seus parâmetros internos de controle. As *EEs* apresentam facilidades no ajuste de tais parâmetros através da utilização de auto-adaptação. Nos *AGs*, os parâmetros de controle são ajustados, usualmente, através de métodos heurísticos de tentativa e erro.

O princípio da auto-adaptação é facilitar o controle implícito dos parâmetros da *EE*, pela sua incorporação na representação do indivíduo com a evolução usual das variáveis objeto. O termo parâmetros da estratégia (ou parâmetros de controle) refere-se aos parâmetros que controlam o procedimento de busca evolutiva, tais como: taxa de mutação, variância da mutação e taxa de recombinação (BÄCK *et al.*, 1997a).

Muitas das pesquisas dos princípios de auto-adaptação em *AEs* tratam de parâmetros relacionados com operador de mutação. A técnica de auto-adaptação é muito utilizada para as variâncias e as covariâncias de uma distribuição normal n -dimensional (FOGEL, 1995).

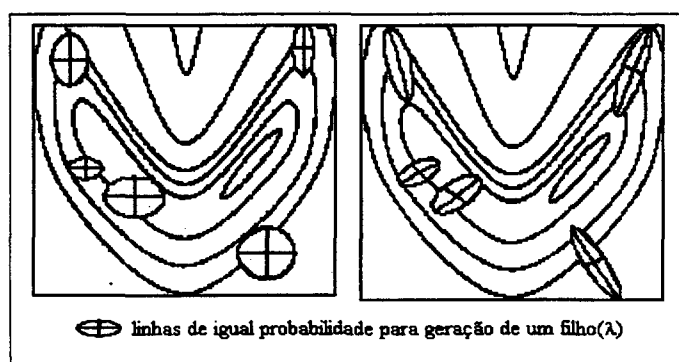
Segundo ANGELINE (1995) é possível adaptar dinamicamente os aspectos de processamento de um *AE*, antecipando as regularidades do ambiente, aprimorando o procedimento de otimização e enfatizando a rapidez na busca dos parâmetros. Os *AEs* que apresentam mecanismos adaptativos (*AEMAs*) distinguem-se pela configuração dinâmica dos parâmetros selecionados ou operadores durante o ciclo evolutivo.

Os *AEMAs* têm uma vantagem sobre os *AEs* básicos, são mais reativos em antecipar as particularidades do problema e, em algumas formulações, podem dinamicamente adquirir informação sobre as regularidades no problema e explorá-las. Segundo ANGELINE (1995), os *AEMAs* podem ser separados em três níveis, em que os parâmetros adaptativos atuam:

- (i) *nível populacional*: os métodos adaptativos ajustam dinamicamente os parâmetros, que são globais à população inteira;
- (ii) *nível individual*: os métodos adaptativos modificam a maneira que um indivíduo da população é afetado pelos operadores de mutação;

(iii) *nível de componente*: os métodos adaptativos alteram a forma pela qual os componentes de cada indivíduo são manipulados, independentemente dos outros indivíduos.

Os mecanismos de auto-adaptação, no nível de componente dos parâmetros da estratégia, providenciam uma das características principais do sucesso das *EEs*. As *EEs* utilizam princípios de busca no espaço de variáveis objeto e estratégia interna de controle dos parâmetros, simultaneamente (BEYER, 1995). A figura 2.10 apresenta o efeito da distribuição de tentativas com mutações Gaussianas descorrelacionadas (esquerda) e correlacionadas (direita).



Fonte: BÄCK & HOFFMEISTER (1991)

Figura 2.10. Efeito da operação de mutação descorrelacionada e correlacionada em *EEs*.

As mutações Gaussianas geram contornos provavelmente elípticos em cada dimensão. Se as mutações são independentes através de cada dimensão, estas elipses estão alinhadas com os eixos coordenados. Se as mutações são correlacionadas, então as elipses podem realizar rotações arbitrárias. Desta forma as curvas subjacentes indicam contornos de erro igual na superfície de resposta. A figura 2.10 indica a utilidade potencial da incorporação de mutações correlacionadas devido ao contorno da superfície não estar alinhado com os eixos coordenados.

O poder de uma *EE* é baseado principalmente na habilidade de executar uma otimização de parâmetros de segundo nível. Este procedimento adapta o poder de mutação de tal forma que a *EE* apresente desempenho “próximo” do ótimo. Existem diferentes possibilidades de construir tais estratégias. A regra mais simples é a de *1/5 de*

probabilidade de sucesso (RECHENBERG, 1973), onde a proporção de mutações bem sucedidas de todas mutações deve ser de 1/5.

Esta forma de configuração da mutação opera satisfatoriamente em muitos problemas de otimização, mas depende da aplicabilidade de um modelo externo da topologia do espaço de parâmetros e isto somente habilita a tamanho de passo geral, não um tamanho de passo individual (OSTERMEIER *et al.*, 1995).

O operador de mutação em uma *EE* realiza um tipo de procedimento de subida de encosta, quando considera sua combinação com o operador de seleção. Com o desvio padrão, σ , para cada variável objeto configura-se as direções de busca, que podem ser estabelecidas somente ao longo dos eixos do sistema de coordenadas, adotado. Em geral, a melhor direção de busca, ou seja, o gradiente, não é alinhado com estes eixos. Caso contrário, a trajetória da população, através do espaço de busca apresenta, um comportamento do tipo zigzague ao longo do gradiente (BÄCK *et al.*, 1991; DAVIDOR & SCHWEFEL, 1992).

Para evitar a redução da taxa de convergência, o operador de mutação pode ser estendido para manipular mutações correlacionadas. SCHWEFEL (1981) propôs uma extensão do operador de mutação deste tipo, que requer um vetor de estratégia adicional, aproximando a inversa da Hessiana de forma probabilística, simultaneamente, ao procedimento de otimização. RUDOLPH (1992) mostrou que o procedimento de aproximação probabilística de SCHWEFEL (1981) pode ser utilizado para a construção de um vetor aleatório correlacionado multinormal válido. Contudo, os resultados numéricos indicam que a convergência da aproximação ainda não é satisfatória. O motivo principal é que o procedimento de adaptação do tamanho de passo afeta o procedimento de adaptação do ângulo.

Nesta tese, adota-se uma *derandomized* $(1+\lambda)$ -*EE* com tamanho de passo individual e mutações correlacionadas — $(1+\lambda)$ -*EEMC* proposta por OSTERMEIER *et al.* (1994, 1995). A $(1+\lambda)$ -*EEMC* objetiva a configuração de um esquema de controle do tamanho de passo da mutação. O conceito de adaptação, deste algoritmo, utiliza a informação acumulada das gerações (iterações) precedentes. Esta forma de adaptação utiliza pontos

selecionados (passos de mutação selecionados) no espaço de parâmetros objeto para o ajuste dos parâmetros da *EE* (COELHO & COELHO, 1999e).

Esta abordagem está relacionada aos métodos da variável métrica (quase-Newton), onde a matriz Hessiana é aproximada iterativamente por regras de atualização. Assim, esta forma de otimização deve ser utilizada com cuidado para evitar que *EEs* sejam atraídas para mínimos locais (OREN & LUENBERGER, 1974; FLETCHER, 1987).

A $(1+\lambda)$ -*EEMC* apresenta como passo inicial a inicialização aleatória das variáveis objeto com distribuição uniforme. A $(1+\lambda)$ -*EEMC* emprega apenas dois operadores: seleção e mutação, estes descritos a seguir. O operador de seleção (adaptação) é determinístico e seleciona o indivíduo μ mais apto da população, podendo-se adotar $1 \leq \mu < \lambda$. Neste caso, apenas o indivíduo mais apto ($\mu=1$), ou seja, com maior aptidão, avança para a próxima geração, isto é,

$$\vec{K}_{\mu}(t+1) = \vec{K}_{\lambda sel}(t) \quad (2.27)$$

onde $K_{\mu}(t)$ é o vetor parâmetro da geração t , λsel denota a quantidade de descendentes selecionados da geração t . O operador de mutação é o operador principal e introduz modificações aleatórias na população. Este algoritmo realiza mutações elipsóides na direção dos descendentes selecionados. A criação dos descendentes, λ , é dada por (OSTERMEIER *et al.*, 1994):

$$\vec{K}_{\lambda i}(t) = \vec{K}_{\mu}(t) + \delta(t) \cdot \vec{\delta}_{scal}(t) \cdot \vec{Z}_i \quad (2.28)$$

onde $\delta(t)$ é o tamanho de passo da geração t , $\vec{\delta}_{scal}(t)$ é o tamanho de passo individual da geração t ($\vec{\delta}_{scal}(0) = (1, \dots, 1)$), $\vec{Z} = (z_1, \dots, z_{np})$, com $z_i(0,1)$ apresentando distribuição uniforme. A equação adotada para modelar a informação acumulada das mutações selecionadas é um modelo de primeira ordem e é descrita por

$$\vec{Z}(t) = (1-c)\vec{Z}(t-1) + c\vec{Z}_{sel}, \quad \vec{Z}(0) = \vec{0} \quad (2.29)$$

onde $c = \sqrt{1/np}$ é um fator constante, que determina quão rapidamente a contribuição das gerações anteriores é decrementada. O esquema de adaptação, do tamanho de passo geral, utiliza a convergência de distribuição χ , $|\bar{Z}| = \sqrt{\sum(z_i)^2} \xrightarrow{n \rightarrow \infty} N(\sqrt{n}, 0.5)$, e é regida por:

$$\delta(t+1) = \delta(t) \cdot \left(\exp\left(\frac{|\bar{Z}(t)|}{\sqrt{np} \cdot \sqrt{c/(2-c)}} - 1 + \frac{1}{5np} \right) \right)^\beta \quad (2.30)$$

$$\bar{\delta}_{scal}(t+1) = \bar{\delta}_{scal}(t) \cdot \left(\exp\left(\frac{|\bar{Z}(t)|}{\sqrt{c/(2-c)}} + 0.35 \right) \right)^{\beta_{scal}} \quad (2.31)$$

onde $\beta = \sqrt{1/np}$ e $\beta_{scal} = 1/np$. Os expoentes β e β_{scal} são fatores importantes para a velocidade de adaptação e precisão, e usualmente estão no intervalo [0;1]. Valores pequenos para os expoentes facilitam a precisão, mas consomem um número maior de gerações para a adaptação e vice-versa. O fator $\sqrt{c/(2-c)}$ normaliza as variâncias médias das distribuições, resultando em uma série geométrica de variações médias das mutações, ou seja,

$$\lim_{m \rightarrow \infty} \frac{\sqrt{c^2 + [c \cdot (1-c)]^2 + [c \cdot (1-c)^2]^2 + [c \cdot (1-c)^3]^2 + \dots + [c \cdot (1-c)^m]^2}}{\sqrt{c/(2-c)}} = \quad (2.32)$$

O controle do tamanho de passo da mutação, através da configuração da $(1+\lambda)$ -*EEMC*, habilita uma adaptação confiável do tamanho de passo e a utilização da informação acumulada. O potencial desta metodologia é a utilização implícita de correlações das mutações selecionadas, na sequência de gerações do procedimento evolutivo. Conseqüentemente, os tamanhos dos passos são adaptados para tais valores, de forma que as mutações selecionadas sucessivamente tendem a serem em média ortogonais (OSTERMEIER *et al.*, 1994, 1995).

2.8 Programação evolutiva

A programação evolutiva (*PE*) foi desenvolvida inicialmente por L. J. Fogel, na década de 1960, teve enfoque na evolução de máquinas de estado finito. A transformação de seqüências de símbolos de entrada em seqüências de símbolos de saída, pelas máquinas de estado finito, visava a previsão de séries temporais baseada nas informações disponíveis *a priori*. A *PE* foi estendida, posteriormente, a problemas de otimização de parâmetros.

A *PE*, de forma análoga aos outros *AEs* apresentados, utiliza os conceitos de evolução, para gerar progressivamente soluções apropriadas para ambientes estáticos ou que mudam dinamicamente. A *PE*, de forma similar as *EEs*, difere dos *AGs*, pois são metodologias que simulam a evolução, enfatizando mais a ligação comportamental (relação fenotípica) entre as populações geradas (ancestrais e descendentes) que a ligação genética.

Entretanto, existem algumas diferenças entre as características da *EE* em relação a *PE*. A seleção na *PE* é tipicamente probabilística, usualmente, realizada através de torneio (*tournament*), enquanto na *EE* é determinística. A operação de mutação na *PE*, usualmente, utiliza a distribuição Gaussiana, enquanto a *EE* utiliza o operador de mutação com distribuição lognormal. Atualmente, existem alguns estudos comparativos da adoção de outras distribuições de probabilidade nos operadores de mutação de ambas, *PE* e *EE* (BÄCK *et al.*, 1997a, 1997b).

A operação de seleção na *PE* caracteriza-se por abstrair a evolução da população, onde diversas espécies competem para obter parte dos recursos, enquanto na *EE* a reprodução realiza-se no nível de comportamento individual (BÄCK & SCHWEFEL, 1993, HEITKOETTER & BEASLEY, 1996). O procedimento de otimização via *PE* é implementado conforme os seguintes passos (FOGEL, 1994, 1995):

- (i) a criação da população inicial de parâmetros compreendendo N_{ind} soluções. Cada um dos indivíduos (x_i, σ_{x_i}) , onde $x_i \in \mathbb{R}^n$ consiste de vetores de soluções, e $\sigma_{x_i} \in \mathbb{R}^n$ são desvios padrões, $i=1, \dots, N_{ind}$, com as dimensões correspondendo ao número de parâmetros a serem otimizados. Os componentes de cada x_i e σ_{x_i} são gerados aleatoriamente de acordo com uma distribuição uniforme em um intervalo especificado, *a priori*;
- (ii) cada solução x_i é classificada com relação a sua aptidão;

(iii) cada vetor de solução ancestral (x_i, σ_{xi}) gera somente um vetor solução descendente (x'_i, σ'_{xi}) , de acordo com as seguintes equações:

$$x'_i(t) = x_i(t) + \sigma_{xi} N(0,1), \quad i=1, \dots, N_{ind} \quad (2.33)$$

$$\sigma'_{xi}(t) = \sigma_{xi}(t) + \alpha \sigma_{xi} N(0,1), \quad i=1, \dots, N_{ind} \quad (2.34)$$

onde α é um fator de escala, $x_i(t)$, $x'_i(t)$, $\sigma_{xi}(t)$, e $\sigma'_{xi}(t)$ denotam o i -ésimo componente dos vetores x_i , x'_i , σ_{xi} e σ'_{xi} , respectivamente. O termo $N(0,1)$ é uma distribuição variável Gaussiana aleatória com média 0 e desvio padrão 1;

- (iv) cada vetor solução descendente x'_i é avaliado com relação à função de aptidão;
- (v) comparações são conduzidas sobre todas as x_i e x'_i soluções. Para cada solução, k oponentes são selecionados, aleatoriamente, e após são escolhidos dos vetores solução ancestrais e descendentes, com igual probabilidade. A seleção realizada, neste caso, é através de competição por meio da seleção por torneio. Na seleção por torneio, em cada comparação, se a solução considerada oferece pelo menos um desempenho tão adequado quanto o oponente selecionado aleatoriamente, ela recebe uma “vitória” (detalhes no item 2.6.1 (iii));
- (vi) os vetores soluções, x_i e x'_i que apresentam mais “vitórias” são selecionados para serem ancestrais na próxima geração, sendo que os vetores σ'_x e σ_x a elas associados são também incluídos;
- (vii) repetir os passos (ii) a (vi) até que uma condição de parada seja satisfeita.

O procedimento para o paradigma *PE* pode ser sintetizado pelo pseudocódigo apresentado na figura 2.11 (BÄCK & SCHWEFEL, 1993, BÄCK *et al.*, 1997a).

```

t := 0;
inicializar: P(0) := {x1(0), x2(0), ..., xμ(0)}
avaliar: P(0) := {Φ(x1(0)), Φ(x2(0)), ..., Φ(xμ(0))}
enquanto uma condição de parada não é satisfeita
{
  realizar mutação: P'(t) := mem(P(t))
  avaliar: P'(t) := {Φ(x'1(t)), Φ(x'2(t)), ..., Φ(x'λ(t))}
  selecionar: P(t+1) := ses(P'(t) ∪ Q)
  t := t + 1;
}

```

Figura 2.11. Pseudocódigo básico da *PE*.

As convenções utilizadas na figura 2.11 são as seguintes:

$\mu \geq 1$: tamanho da população ancestral;

$\lambda \geq 1$: tamanho da população descendente;

m_{Θ_m} : operador de mutação com parâmetros de controle Θ_m ;

s_{Θ_s} : operador de seleção $\ni s_{\Theta_s}: (I^\lambda \cup I^{\mu+\lambda}) \rightarrow I^\mu$;

$Q \in \{\emptyset, P(t)\}$: conjunto de indivíduos considerados pela adição de passos de seleção.

2.3.1 Abordagem da PE com o operador de mutação de Cauchy

Os descendentes são gerados de seus ancestrais, através de mutações aleatórias. Tipicamente, cada indivíduo, composto de uma variável objeto (vetor solução) acompanhado do seu respectivo desvio padrão, tem seu valor modificado por uma variável aleatória com distribuição Gaussiana, com média zero e valor de variância adaptável. A técnica de PE utilizada tem operador de mutação lognormal (PE-L) e obedece ao método de Box-Muller (PRESS *et al.*, 1994), para a geração de valores aleatórios com uma distribuição normal.

No presente estudo aborda-se também uma variante do operador de mutação. Esta técnica utiliza o operador de mutação com distribuição de Cauchy (PE-C) e apresenta uma função densidade de probabilidade centrada na origem, definida por (YAO & LIU, 1996a):

$$f_t = \frac{1}{\pi} \frac{t}{(t^2 + z^2)}, \quad \text{para } -\infty < z < \infty \quad (2.35)$$

onde $t > 0$ é um parâmetro de escala. A função de distribuição correspondente é:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \quad (2.36)$$

A forma de $f_t(x)$ parece-se com a função de densidade Gaussiana, mas nas proximidades o eixo $f_t(x)$ decresce mais vagarosamente. Como um resultado disto, a

variância da distribuição de Cauchy é infinita. A figura 2.12 mostra a diferença entre as funções densidade de Cauchy e Gaussiana.

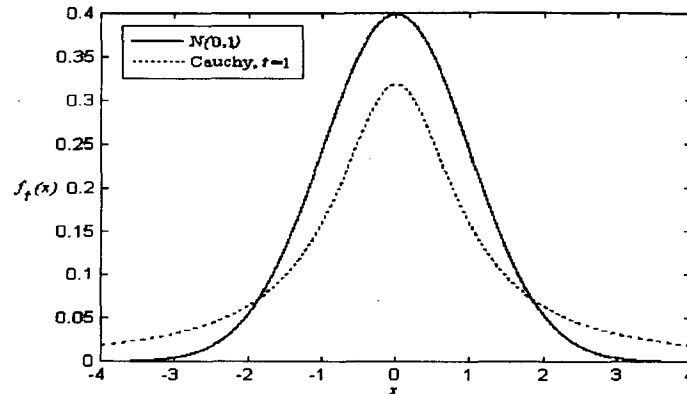


Figura 2.12. Diferença entre as funções densidade de Cauchy e Gaussiana.

Alguns estudos têm indicado o benefício do incremento da variância em algoritmos de Monte Carlo. Um exemplo é a implementação de algoritmos rápidos de *simulated annealing* (INGBER, 1989).

O operador de mutação, com distribuição de Cauchy (CHELLAPILLA, 1998), é útil para escapar de ótimos locais, enquanto o operador lognormal providencia a convergência local mais rápida em funções convexas. A estratégia de mutação, combinando estes dois operadores de mutação, pode explorar as propriedades desejadas de convergência da *PE*.

2.9 Resumo das características dos *AEs*

Os *AEs* apresentam similaridades no processo evolutivo, contudo algumas características diferem de uma técnica para outra e estão resumidas na tabela 2.1. É necessário ressaltar que existem variantes dos *AEs* e na literatura são propostas abordagens com características distintas ou mesmo combinação das apresentadas na tabela 2.1.

Tabela 2.1. Características dos AEs abordados.

<i>(i) representação</i>		
	<i>usual</i>	<i>variante</i>
<i>AG</i>	binária	inteira / real ¹
<i>PE</i>	real	híbrida inteira / real ²
<i>EE</i>	real	híbrida inteira / real ³
<i>(ii) auto-adaptação</i>		
<i>AG</i>	existem algumas abordagens que permitem auto-adaptação ⁴	
<i>PE</i>	utiliza desvio-padrão e covariância; necessita do conhecimento da dimensão do problema (nº de variáveis) ³	
<i>EE</i>	utiliza desvio-padrão e covariância; necessita do conhecimento da dimensão do problema (nº de variáveis) ⁵	
<i>(iii) seleção</i>		
<i>AG</i>	probabilística, preservativa	
<i>PE</i>	probabilística, competitiva	
<i>EE</i>	determinística, extintiva	
<i>(iv) operação</i>		
	<i>recombinação</i>	<i>mutação</i>
<i>AG</i>	operador principal	operador secundário
<i>PE</i>	operador secundário	operador principal (Gaussiano)
<i>EE</i>	operador secundário	operador principal (lognormal)
<i>convenções (referências bibliográficas)</i>		
¹	MICHALEWICZ (1992)	
²	BÄCK <i>et al.</i> (1997a, 1997b)	
³	BÄCK <i>et al.</i> (1997a), FOGEL (1995)	
⁴	SALOMON (1996)	
⁵	BÄCK & SCHWEFEL (1993); BEYER (1995)	

2.10 AEs híbridos com metodologias de busca local

Os métodos de otimização têm duas formas de configuração: os métodos determinísticos e os métodos estocásticos. As técnicas determinísticas buscam um ponto de mínimo, baseadas na informação dada pelo negativo do gradiente da função objetivo. A eficiência destas técnicas depende de diversos fatores, tais como: o ponto inicial, a precisão da avaliação da direção descendente e o método utilizado para executar a busca em linha, além do critério de parada. A solução obtida é geralmente um ponto de mínimo local, que pode ser mínimo global se a função apresentar apenas uma moda. As duas desvantagens

principais são a necessidade de avaliações do gradiente e falta da garantia do mínimo global. Os métodos estocásticos, dos quais os *AEs* fazem parte, não necessitam do cálculo do gradiente e são aptos a encontrar a solução global. Contudo, o número de avaliações da função objetivo, necessárias para encontrar a solução, é normalmente maior que o número requerido pelos métodos determinísticos (VASCONCELOS *et al.*, 1999).

A solução de problemas com muitos ótimos locais defronta-se com o conflito fundamental entre precisão, integridade e esforço computacional. O compromisso entre *exploitation* (convergência) e *exploration* (diversidade da população) é uma constante no *AE* e deve ser considerada na configuração de uma metodologia de otimização eficiente.

A configuração de abordagens compostas por técnicas determinísticas, estas hibridizadas, com técnicas estocásticas, é uma alternativa. Para obter os benefícios da configuração híbrida, uma forma eficiente é executar, inicialmente, um *AE* para localizar a região de ótimo global e após aplicar-se outra metodologia de otimização para a busca local. A vantagem da utilização de um método de busca direto em relação a busca local está na melhoria da velocidade de convergência pela avaliação da função objetivo. O valor final obtido pelo método de busca direto provavelmente será mais preciso que o obtido pelo *AE*.

Segundo VASCONCELOS *et al.* (1999), a maior dificuldade no uso de metodologias híbridadas, também conhecidos por *algoritmos meméticos*, de natureza determinística e estocástica, é encontrar o momento de parar o procedimento de busca estocástico para iniciar o procedimento de busca determinística. A idéia principal é começar o método determinístico quando a região de mínimo global foi obtida. Contudo, não se sabe quando esta região é obtida. Assim, é impossível garantir que a solução é a global. VASCONCELOS *et al.* (1999) propõem os seguintes critérios para comutação entre *AE* e métodos de busca local que podem ser definidos pelo(a):

- (i) *número de gerações*: No caso mais simples, o procedimento de otimização do *AE* é finalizado em um número especificado de gerações e o melhor resultado é transferido para o método determinístico. A principal desvantagem deste critério é que nem todas as funções objetivo apresentam o mesmo comportamento;
- (ii) *diferença entre os melhores valores da função objetivo em um conjunto de gerações*: Este procedimento é menos sensível ao tipo de problema que a do item (i);
- (iii) *diferença entre os melhores valores da função objetivo em uma mesma geração*: Neste caso analisa-se a similaridade dos indivíduos que compõem a população.

A seguir são apresentadas duas abordagens de *AE* híbrido. Estas podem ser consideradas formas de aprendizado Lamarckiano (KENNEDY, 1993, WHITLEY, 1994). As abordagens são do *AE* híbrido com o algoritmo *simulated annealing* e *AE* híbrido com método *simplex* de NELDER & MEAD (1965).

2.10.1 *AE* híbrido com *simulated annealing*

O algoritmo *simulated annealing* (*SA*) — ou têmpera simulada — é uma variação de algoritmos de subida de encosta, onde o objetivo é a minimização de uma função objetivo (nível de energia). Segundo SUN (1995), o *SA* difere dos algoritmos de otimização convencionais devido ao fato que ele pode:

- (i) processar funções objetivo que possuem graus arbitrários de não-linearidades, descontinuidades e estocasticidade;
- (ii) tratar restrições impostas pela função objetivo;
- (iii) ser implementado facilmente (poucas linhas de código) em relação a outros algoritmos de otimização não-linear;
- (iv) garantir estatisticamente que encontra a solução ótima.

O *SA* baseia-se em uma analogia com a mecânica estatística de materiais com resfriamento, onde substâncias físicas tais como os metais são levados a altos níveis de temperatura e posteriormente são gradualmente resfriadas até alcançar um estado mínimo de energia. Sob outras condições, menos cuidadosas de resfriamento, o material se cristalizaria com uma energia “localmente mínima”, o que frequentemente se traduz em imperfeições estruturais. A esse processo cuidadoso de resfriamento denomina-se *annealing*.

A abordagem do *SA* é probabilística. O *SA* não requer informação de derivadas e não é afetado por descontinuidades e não-linearidades. O *SA* é um procedimento de descida de encosta, mas configurado de forma modificada. O *SA* realiza pequenos passos de busca em uma topografia local; se o passo resulta em uma melhora na solução, a nova solução é aceita, caso contrário, esta nova solução é aceita com probabilidade que inicialmente é configurada para 1. Contudo, com o progresso das iterações o *SA* é reduzida

a probabilidade de aceitar uma nova solução que não apresenta aprimoramento em relação àquela obtida na iteração anterior.

A idéia do *SA* origina-se numa combinação das observações sobre a física dos materiais, com procedimento computacional tendo por finalidade a simulação do comportamento de uma coleção de átomos (variáveis do problema) em condições de temperatura fixa. KIRKPATRICK *et al.* (1983) foram os primeiros pesquisadores a propor e demonstrar a aplicação de técnicas de simulação da física dos materiais para problemas de otimização, especificamente em problemas de projeto de *hardware*.

A simulação a uma temperatura fixa, T , consiste em uma seqüência de passos. A cada passo, é dado um pequeno deslocamento a um dos átomos e é calculada a variação da energia, ΔE , que o sistema sofre com aquele deslocamento. Se $\Delta E \leq 0$, o deslocamento é incorporado ao estado do sistema, que é utilizado para o passo seguinte, contudo se $\Delta E \geq 0$, a aceitação ou não da transição para um estado maior de energia tem uma probabilidade dada pela função:

$$p = e^{-\Delta E * k_b * T} \quad (2.37)$$

onde k_b é a constante da distribuição de Boltzmann, T é a temperatura e ΔE é a mudança no nível de energia. Com a finalidade de simplificar a implementação do *SA* pode-se ignorar a constante k_b e reescrever a probabilidade de aceitação de uma configuração de energia mais alta como

$$p = e^{-\left(\frac{\Delta E}{T}\right)} \quad (2.38)$$

A figura 2.13 apresenta um procedimento genérico para aplicação do algoritmo *SA*. As principais características em consideração no desenvolvimento de um esquema de *SA* são:

- (i) a descrição da configuração do sistema (representam o espaço dos projetos possíveis);
- (ii) o mecanismo de perturbação do sistema (usualmente geração de um valor aleatório);
- (iii) o critério de aceitação de uma solução;
- (iv) o esquema de resfriamento, com a definição das temperatura inicial, T_0 , e temperatura final, $T_{\text{mínima}}$;

- (v) a definição da função energia, $r(k)$, que determina a redução gradual de temperatura em função de um fator de *annealing*, $\alpha_{SA} \in [0;1]$, que determina a velocidade com que o sistema é resfriado (cronograma de t mpera). Neste caso adota-se a seguinte equa o:

$$r(k) = \alpha_{SA}^{(k+1)} T_k, \text{ para todo } k \geq 0 \quad (2.39)$$

```

inicializar os par metros  $T_0$ ,  $T_{minima}$  e  $\alpha_{SA}$ 
gerar aleatoriamente um estado inicial,  $x_0$ , como estado corrente
atribuir  $k = 1$ ;  $x_k = x_0$ ;  $T_k = T_0$ ;
fazer enquanto ( $T_k \geq T_{minima}$ ) /* enquanto o crit rio de parada n o   obtido */
{
  fazer enquanto "o sistema n o est  em equil brio na temperatura  $T_k$ "
  {
    selecionar um ponto  $x_{SA}$  de forma aleat ria;
     $\Delta E = f(x_{SA}) - f(x_{k-1})$ ;    % ( $\Delta$ );
    se  $\Delta E \leq 0$  ent o
       $x_k = x_{SA}$ ;
    sen o
      se m nimo ( $1, e^{-\Delta E/T}$ ) > (n mero aleat rio entre [0;1]) ent o
         $x_k = x_{SA}$ ;
  }
   $T_{k+1} = r(k) = \alpha_{SA}^{(k+1)} T_k$ ;
   $x_{k+1} = x_k$ ;
   $k = k + 1$ ;
}
o estado corrente,  $x_k$ ,   a solu o.

```

Figura 2.13. Fluxo b sico dos dados no algoritmo SA.

O SA   aplicado principalmente em problemas de otimiza o combinat ria, em sistemas h bridos com outros m todos num ricos de otimiza o e t cnicas de IC. Neste trabalho, tem-se como objetivo a aplica o de SA conjuntamente a AGs.

Os AEs caracterizam-se pela robustez em aplica es complexas e pela efici ncia na busca global. Entretanto, para buscas locais geralmente n o apresentam um comportamento adequado. A converg ncia do AE pode tornar-se mais r pida e precisa,

implementando um *AE* com ajuste fino (busca local) através de *SA*. Os casos abordados são:

- (i) *AG* híbrido com *SA* (*AGSA*);
- (ii) *EE* híbrida com *SA* (*EESA*); e
- (iii) *PE* híbrida com *SA* (*PESA*).

O procedimento de *annealing* é aplicado ao *AE*, a cada geração, após as operações de recombinação e mutação, para sintonização fina dos valores dos indivíduos do *AE*. O *SA* é aplicado a cada geração após as operações de recombinação e mutação, através de uma pequena perturbação no valor dos indivíduos. Um baixo fator de temperatura, T , e um rápido fator de *annealing*, α_{SA} , são utilizados nos *AEs* híbridos com *SA* (TAN *et al.*, 1995).

2.10.2 *AE* híbrido com método *simplex*

O método *simplex* de NELDER & MEAD (1965) — também conhecido como método do politopo ou método ameba — é uma técnica simples de busca direta que é utilizada em problemas de otimização. Uma busca direta significa que o método é guiado somente pelo cálculo do valor da função em vários pontos e não necessita da avaliação da primeira e segunda derivadas (parcial) da função a ser otimizada. O método *simplex* “mantém” diversos pontos diferentes. Esta é uma característica similar aos *AEs* que, no caso, mantêm uma população de pontos.

A maior diferença entre os *AEs* e o método *simplex*, é que o método *simplex* tem escolha de pontos de forma determinística, configurando um politopo de forma a “repelir” soluções inadequadas. Em contrapartida, os *AEs* têm desempenho vinculado à geração de novos pontos, através dos indivíduos com função de aptidão mais apta ao ambiente, durante o ciclo evolutivo.

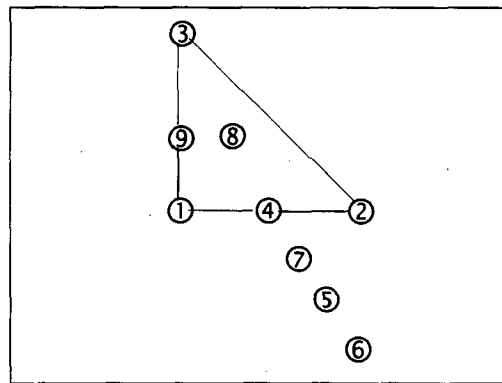
O método *simplex* tem sua configuração básica na definição de um *simplex*. Um *simplex* é uma estrutura formada por $(n_{ptos} + 1)$ pontos, não no mesmo plano, em um espaço multidimensional. O método adotado aqui apresenta algumas modificações do método original de NELDER & MEAD (1965). Segundo NASH (1990), a essência do algoritmo é a seguinte: a função é avaliada em cada ponto (vértice) do *simplex*. Em um

problema de minimização, o vértice que têm maior valor de função objetivo é substituído por um novo ponto com valor da função objetivo menor. Existem quatro operações que são realizadas em um simplex: reflexão, expansão, redução e contração.

O simplex inicial se move, expande e contrai, de tal maneira que se adapta ao panorama da função e, finalmente, aproxima-se do ótimo. Para determinar a transformação apropriada, o método usa só a ordem relativa entre os desempenhos (valor da função a ser otimizada) do ponto considerado. Depois de cada transformação, o pior ponto é trocado pelo melhor dos já existentes, deste modo o algoritmo sempre força a convergência da seqüência de iterações. De forma a operar-se o simplex é necessário ordenar os pontos de mais alto valor, x_H , o próximo mais alto, x_N , e o mais baixo, x_L . Assim o valor da função objetivo associado às soluções (pontos) do problema obedecem a relação:

$$J(x_H) \geq J(x_N) \geq J(x_i) \geq J(x_L) \quad (2.40)$$

para todo $i \neq H, N$ ou L , onde x são vetores. A figura 2.14 ilustra a situação do simplex, para um politopo com três lados.



Fonte: NASH (1990)

Figura 2.14. Pontos gerados pelo simplex de NELDER & MEAD em duas dimensões

Na figura 2.14 adota-se as seguintes definições:

- (i) o ponto ①, x_L , é o vértice inferior no simplex;
- (ii) o ponto ②, x_N , é o outro vértice, diferente do superior;
- (iii) o ponto ③, x_H , é o vértice superior;
- (iv) o ponto ④, x_C , é o centróide de todos os vértices, exceto ③(x_H), isto é,

$$x_c = \frac{\sum_{\substack{j=1 \\ j \neq H}}^{nptos+1} x_j}{nptos} \quad (2.41)$$

- (v) o ponto ⑤, x_R , é a reflexão de ③(x_H) através de ④(x_C);
- (vi) o ponto ⑥, x_E , é o resultado da extensão da linha (④, ⑤), isto é, (x_C, x_R), com comprimento regido por um fator de expansão;
- (vii) o ponto ⑦ é o resultado da redução da linha (④, ⑤), que ocorre quando x_R é inferior a x_H , mas superior a x_N ;
- (viii) o ponto ⑧ é o resultado da redução da linha (④, ③), isto é, (x_C, x_H);
- (ix) o ponto ⑨ é gerado por uma contração geral do simplex, realizada pelos vértices ①, ② e ③ através de ①(x_L).

A operação de reflexão reflete ③(x_H) através de ④(x_C) usando um fator de reflexão α_R , para encontrar o ponto ⑤, ou seja,

$$x_R = x_C + \alpha_R(x_C - x_H) \quad (2.42)$$

Se $J(x_R)$ é menor que $J(x_L)$ um novo ponto mais inferior é encontrado, e o simplex pode ser expandido estendendo a linha (x_R, x_C) para encontrar o ponto ⑥, ou seja,

$$x_E = x_R + (\gamma_E - 1)(x_R - x_C) \quad (2.43)$$

onde γ_E é o fator de expansão, é maior que a unidade, caso contrário x_E representa uma contração. Se $J(x_E) < J(x_R)$ então x_H é repassado por x_E e o procedimento repetido encontrando um novo ponto superior e um novo centróide de $nptos$, x_C . Em outro caso, x_R é o novo ponto mais inferior e é repassado por x_H . Quando x_R não é o ponto inferior, mas é menor que x_N , tem-se

$$J(x_L) \leq J(x_R) < J(x_N) \quad (2.44)$$

e x_H é repassado por x_R e o procedimento é repetido. A situação permanecendo, tem-se $J(x_R)$ no mínimo maior que $J(x_N)$ e deve-se reduzir o simplex. Existem duas possibilidades:

- (i) Se $J(x_N) \leq J(x_R) < J(x_H)$ então a redução é feita repassando x_H por x_R e, acha-se um novo vértice entre x_C e x_R (atualmente x_H). Esta é uma redução no lado da reflexão (posição inferior).
- (ii) Se $J(x_R) > J(x_H)$ a redução é feita encontrando um novo vértice entre x_C e x_H (posição superior).

Em um ou outro dos casos acima a redução é controlada pelo fator β_{NM} com valor entre 0 e 1. Visto que o caso (a) repassa x_H por x_R , a mesma fórmula aplicada para o novo ponto x_S (S denota que o simplex é menor) em ambos os casos. O x_H é utilizado para denotar-se ambos x_R e x_H , desde que no caso (i) x_R tornou-se o novo ponto superior no simplex, ou seja,

$$x_S = x_C + \beta_{NM}(x_H - x_C) \quad (2.45)$$

O novo ponto x_S repassa o corrente x_H , que no caso (i) é, de fato, x_R , a menos que

$$J(x_S) > \text{mínimo} \{J(x_H), J(x_R)\} \quad (2.46)$$

A substituição de x_H por x_R no caso (a) pode, em uma implementação, medir que este mínimo já foi salvo com seu ponto associado. Quando a equação (2.46) é satisfeita uma redução dá um ponto maior que $J(x_N)$, assim uma contração geral do simplex sobre o ponto inferior x_L , é sugerida, ou seja,

$$x_i' = x_L + \beta_{NM}'(x_i - x_L) \quad (2.47)$$

para todo $i \neq L$. Em aritmética exata, a equação (2.47) é aceitável para todos os pontos, contudo NASH (1990) faz alguns testes omitindo $i = L$. Alguns cuidados devem ser observados pois o ponto x_L pode ser alterado nas operações da equação (2.47). Diferentes fatores de contração β_{NM} e β_{NM}' podem ser utilizados. Na prática estes fatores, assim como os fatores α_R e γ_E podem ser escolhidos para tentar melhorar a taxa de convergência deste procedimento para uma classe específica ou um alcance vasto de problemas. Segundo NELDER & MEAD (1965), uma escolha adequada é

$$\begin{cases} \alpha_R = 1 \\ \gamma_E = 2 \\ \beta_{NM} = \beta_{NM}' = 0,5 \end{cases} \quad (2.48)$$

Todos os aspectos do método foram desenvolvidos heurísticamente, baseados no conceito intuitivo do problema de minimização. O critério de parada, geralmente, depende da diferença em magnitude entre a performance do melhor ponto e do pior. O ciclo de

otimização termina quando a diferença é menor que um valor pré-definido. Quando o ciclo termina, é comum recomeçar o procedimento com um novo simplex, onde o ponto ótimo, obtido anteriormente, é mantido como um dos vértices do novo simplex. Os demais pontos são escolhidos aleatoriamente em uma direção ortogonal. Este procedimento converge quando a solução começar a se repetir.

O método simplex é facilmente hibridizado com os *AEs*. Recentemente, na literatura são apresentadas algumas abordagens interessantes. YEN *et al.* (1998) apresentam um estudo de diversas configurações possíveis de *AGs* híbridos com o método simplex aplicado a um problema de identificação. YANG & DOUGLAS (1998) apresentam um estudo de *AGs* híbridos com o método simplex, abordando questões de convergência, eficiência e robustez.

Nesta tese são apresentadas duas abordagens de hibridização de *AEs* com o método simplex de NELDER & MEAD (1965). Uma primeira abordagem de hibridização é rodar o *AE* de forma usual. Contudo, a cada geração, após a aplicação dos operadores de recombinação e mutação, aplica-se o método simplex ao(s) melhor(es) indivíduo visando aprimorar o valor da função de aptidão. Neste caso, o método simplex é utilizado na forma de uma técnica de busca local do *AE* e esta combinação é usualmente referida como um *AE* com mecanismo de subida de encosta. A segunda abordagem não é implementada nesta tese, e constitui-se do método simplex ser um operador evolutivo.

2.11 Conclusão do capítulo

Este capítulo apresentou uma visão geral das características, pseudocódigos e aplicabilidade dos *AEs* em problemas de otimização. Os *AEs* descritos e analisados foram os *AGs*, *EEs* e *PE*. O *AG* canônico apresenta vantagens, pois tem uma forma de representação flexível, que pode ser utilizada no tratamento de problemas com variáveis que apresentam valores reais, inteiros e/ou binários. Os *AGs* com representação em ponto flutuante, *EEs* e *PE*, em contrapartida, apresentam a desvantagem que somente podem ser utilizados no tratamento de variáveis com valores reais. Entretanto, apresentam maior precisão e menor complexidade de implementação, pois não necessitam da fase de decodificação, esta usual nos *AGs* canônicos.

Os *AGs* convencionais têm a desvantagem de não apresentar nenhum mecanismo de auto-adaptação, dificuldades estas que podem ser razoavelmente minimizadas pela configuração adaptativa de parâmetros, tais como: probabilidade de mutação e recombinação. A *EE* e *PE* apresentam a vantagem de permitirem a configuração de mecanismos de auto-adaptação para aprimoramento da metodologia de busca, mas têm a desvantagem de necessitarem conhecer o nº de variáveis para o ajuste das variáveis a cada geração e para o cálculo da função de aptidão.

Os *AGs* apresentam operações de recombinação (ou cruzamento), como operador principal e o operador de mutação, como operador secundário. Entretanto, a *PE* e *EE* têm a operação de mutação como essencial para a configuração, mas podem ou não apresentar a operação de recombinação.

Algumas questões de projeto dos *AEs* permanecem em aberto e são alvo de pesquisas na comunidade de *CE*. As limitações e questões relevantes englobam fatores, que devem ser analisados mais profundamente, tais como:

- (i) regras determinísticas na configuração dos parâmetros de configuração dos algoritmos;
- (ii) estudo das propriedades de convergência, no sentido de melhorar a eficiência e a robustez dos *AEs* para aplicações em tempo real;
- (iii) sedimentação da teoria matemática, fundamentos e características;
- (iv) estudo comparativo de mecanismos de auto-adaptação e auto-organização;
- (v) escolha da representação e conjunto de operadores adequados ao domínio do problema em questão, podendo fazer uma considerável diferença na eficiência e na possibilidade da procura;
- (vi) hibridização eficiente dos *AEs* com outras metodologias de otimização.

Os *AEs* apresentam vantagens e desvantagens em relação aos métodos tradicionais e este aspecto serve para enfatizar a necessidade de não abandonar os métodos convencionais de otimização. Porém, o projetista pode utilizar métodos evolutivos como uma alternativa no auxílio a resolução de problemas onde os métodos convencionais podem falhar ou não terem desempenho adequado. Os *AEs* convencionais geralmente são eficientes para a busca global, mas são relativamente lentos em sintonia local, o que conduz à natural integração com métodos numéricos eficientes de descida de encosta, em buscas locais.

Capítulo 3

SISTEMAS NEBULOSOS: CONCEITOS E PROJETOS

3.1 Introdução

A teoria de controle clássica permite projetar controladores tendo informação imprecisa sobre o processo a ser tratado. Esta teoria, no entanto, apresenta limitações no tratamento de processos complexos, devido a não-linearidades e comportamentos variantes no tempo. Contudo, os operadores humanos controlam, freqüentemente, com sucesso, diversos processos complexos. Os operadores humanos são capazes de controlar baseados em informações imprecisas e/ou incertas a respeito do funcionamento de processos. A estratégia de controle, adotada pelos operadores humanos, é de natureza imprecisa e, geralmente, pode de ser expressa em termos lingüísticos.

Os sistemas nebulosos (*SNs*) possui um formalismo para a representação do conhecimento e inferência de novos conhecimentos que é similar à maneira utilizada pelos seres humanos para expressarem o conhecimento e raciocínio, ao contrário dos sistemas baseados em lógica clássica (JAGER, 1995; ZADEH, 1973).

Os *SNs* têm, recentemente, emergido como uma abordagem atrativa para o controle de processos complexos. A motivação deve-se ao conhecimento do sistema e ao comportamento dinâmico qualitativo e com incertezas. A teoria dos *SNs* provê uma representação adequada para tal conhecimento.

Quanto às suas potencialidades, os *SNs*, de maneira similar a algumas arquiteturas de redes neurais, são aproximadores universais de uma dada função com precisão arbitrária (BUCKLEY & HAYASHI, 1993a; YING, 1998). Os *SNs* são capazes de manipular simultaneamente tanto os dados numéricos quanto o conhecimento lingüístico (MENDEL, 1995). Os *SNs* podem ser vistos como membro de um consórcio de metodologias, que são coletivamente referidas como *soft computing* (ZADEH, 1994b, 1996).

A literatura apresenta diversas aplicações, bem sucedidas, de controle lógico nebuloso (*CLN*), mas paradoxalmente existe uma discussão acadêmica sobre a aplicação dos *SNs* em controle de processos ou mesmo uma rejeição. Segundo BARTOS (1996), isto pode ser causado por alguns equívocos e uso indiscriminado dos *CLNs*.

3.2 Breve histórico

Os fundamentos teóricos dos conjuntos nebulosos foram propostos por ZADEH (1965), professor da Universidade de Califórnia (Berkeley, Estados Unidos), como uma forma alternativa de modelar os sistemas complexos, se diferenciando das técnicas convencionais.

Alguns trabalhos precursores aos de Zadeh foram de suma importância para o desenvolvimento dos *SNs*, entre os quais:

- (i) década de 20: os desenvolvimentos da lógica multivalorada pelo matemático polonês J. Lukasiewicz;
- (ii) década de 30: os estudos do filósofo quântico Max Black que apresentou as noções de funções de pertinência e as definições de estruturas vagas;
- (iii) década de 50: Menger, um matemático francês, propôs o termo *ensemble flou*, uma contrapartida ao termo *fuzzy set*, que seria proposto posteriormente por ZADEH (1965).

Os conceitos dos conjuntos nebulosos e da lógica nebulosa encontram algumas críticas da comunidade acadêmica, entretanto, Zadeh teve seus seguidores. Um dos precursores das pesquisas sobre *SNs* foi e continua sendo o Japão. As pesquisas, no Japão, começaram com dois pequenos grupos, estabelecidos no começo da década de 1970: um deles liderado por T. Terano e H. Shibata, em Tóquio, e outro grupo liderado por K. Tanaka e K. Asai, em Kanasai. Estes pesquisadores, de forma similar aos pesquisadores nos Estados Unidos, encontraram uma atmosfera com certo ceticismo, no Japão, durante os primeiros resultados de seus trabalhos. Contudo, a persistência e o trabalho árduo destes pesquisadores e seus colaboradores acarretaram contribuições relevantes a teoria e as aplicações de *SNs* nos meios acadêmico e industrial (YEN, 1999).

Os pesquisadores MAMDANI & ASSILIAN (1975), do *Queen Mary College*, de Londres, motivados pelo potencial dos *SNs*, utilizaram estratégias de controle baseadas

em informações qualitativas de operadores humanos. Desta forma, surge o primeiro *CLN* aplicado ao controle, de um gerador a vapor, em escala de laboratório.

Este trabalho encorajou outras aplicações. KICKERT & VAN NAUKA LEMKE (1976) conceberam regras de controle do tipo *PI* para o controle de um processo de aquecimento de água. RUTHERFORD & BLOORE (1976) produziram um controlador para um processo de sinterização de minério de ferro.

Os fornos para a produção de cimento caracterizam-se por serem não lineares e difíceis de modelar. O controle de fornos por operadores humanos é uma tarefa complexa. A metodologia de projeto baseada em *SNs* objetiva a aquisição de experiência dos operadores na implementação de um *CLN*, que apresente uma resposta mais rápida que a dos operadores humanos e consumo reduzido de combustível. A primeira aplicação industrial de *CLN* foi o controle de um forno de secagem para a produção de cimento (HOLMBLAD & OSTERGAARD, 1982) pela Smith Co. Ltda, na Dinamarca.

Na década de 80, outras aplicações bem sucedidas dos *SNs* foram apresentadas. Entre as quais, pode-se mencionar o sistema de controle para operação de trens da cidade de Sendai (Japão), este em operação desde 1987. Este sistema foi desenvolvido por S. Yasonobu e seus colegas da empresa Hitachi. Uma outra aplicação relevante é o sistema de tratamento de água desenvolvido pela Fuji Electric (YEN, 1999). Em 1990, a Matsushita Electric Industry produziu o primeiro eletrodoméstico incorporando o *CLN*. Este produto era uma máquina de lavar (RALESCU, 1996).

Estes trabalhos, pioneiros, motivaram o emergente desenvolvimento acadêmico e comercial de ambientes de *software* e *hardware*, com a utilização da tecnologia dos *SNs* (BARTOS, 1997, BONISSONE, 1994, SCHWARTZ *et al.*, 1994, ZADEH, 1994a, KRÜGER *et al.*, 1998; LUIZ *et al.*, 1998, SILVA *et al.*, 1997).

Um destaque é dado ao projeto *LIFE* (*Laboratory for International Fuzzy Engineering Research*), fundado em abril de 1989, por iniciativa do governo japonês, através do *MITI* (*Ministry for International Trade and Industry*). A motivação eram os diversos sistemas de controle nebuloso apresentados no congresso da *IFSA* (*International Fuzzy Systems Association*), realizado em Tóquio, no ano de 1987.

As pesquisas do *LIFE* — nas áreas de *interface* inteligente, controle, memória nebulosa associativa, robótica e processamento de imagens — criaram uma “ponte” entre

a indústria, as agências governamentais japonesas e as universidades. Algumas universidades japonesas tornaram-se centros de excelência de estudo e aplicações de *SNs*. Algumas empresas, fora do Japão, são membros do *LIFE*, tais como: a *Thompson CFS* da França e a *Volkswagen* da Alemanha (RALESCU, 1996).

O resultado da colaboração e a transferência de tecnologia entre as universidades e as indústrias na América Latina também é um tópico importante de investigação. Segundo GOMIDE (1995), os *SNs* são tema de pesquisa nas principais universidades brasileiras desde a década de 70. TANSCHWEIT & BARRETO (1978) apresentaram um artigo pioneiro no Brasil sobre os *SNs* na área de controle de processos no II Congresso Brasileiro de Automática.

GOMIDE (1995) menciona as pesquisas sobre *SNs*, a partir desta época, o qual enfatiza os estudos na Universidade Estadual de Campinas (*UNICAMP*). Os primeiros estudos foram realizados pelo Prof. Armando Rocha, do Departamento de Biologia e o Prof. Engell, do Departamento de Matemática da *UNICAMP*.

Na *UNICAMP*, as pesquisas na década de 80, seguiram-se no Departamento de Engenharia de Computação e Automação Industrial com os Prof. Fernando Gomide e Prof. Márcio L. Andrade Netto. Os primeiros trabalhos abrangeram as idéias dos *SNs* em processamento de linguagem natural e aquisição automática de conhecimento em bases de dados, no Departamento de Biologia, da *UNICAMP*. Quanto ao Departamento de Matemática, da *UNICAMP*, as pesquisas iniciadas na década de 70, e que são tratadas até os dias atuais abrangem a matemática nebulosa, as medidas nebulosas, as integrais nebulosas e a previsão de séries temporais.

Da década de 80 em diante, as pesquisas na *UNICAMP* abrangem as áreas de controle de processos, automação industrial, visão computacional, projeto de sistemas e técnicas de implementação, escalonamento de sistemas de manufatura, planejamento de produção, manufatura e lógica, principalmente nas Faculdades de Engenharia Elétrica e de Computação, Faculdade de Engenharia Mecânica e Departamento de Lógica.

Atualmente, muitas outras instituições brasileiras têm pesquisado os *SNs*. Alguns exemplos são: Universidade de São Paulo, Pontifícia Universidade Católica do Rio de Janeiro, Universidade Federal de Minas Gerais, Universidade Federal de São Carlos, Universidade Federal de Santa Catarina, Instituto Nacional de Pesquisas Espaciais, para mencionar umas poucas. As pesquisas realizadas na área de controle de processos têm

destaque em eventos e revistas, principalmente da Sociedade Brasileira de Automática e Sociedade Brasileira de Computação.

Outras instituições da América Latina estão envolvidas com pesquisas sobre os *SNs*, destacando-se a Universidade Católica do Chile (com o Prof. Aldo Cipriano e seus colaboradores) e a Universidade Católica de Rosário, na Argentina. Do ponto de vista de aplicações industriais, os principais projetos têm abordado sistemas de controle, refinarias petrolíferas, planejamento de produção e sistemas de escalonamento à manufatura de plástico.

3.3 Por que utilizar os *SNs*?

As características dos *SNs*, que os tornam uma alternativa atraente de projeto, nos mais variados campos do conhecimento são (COELHO & COELHO, 1997c, 1997h, 1999a):

- (i) a habilidade de tratar processos não lineares: característica relevante à aplicação em identificação e controle de sistemas dinâmicos complexos, com ordem elevada, atraso de transporte e sistemas “mal” definidos;
- (ii) a possibilidade da utilização de termos e expressões da linguagem natural;
- (iii) em controle de processos quando o modelo matemático do processo não é avaliado ou é complexo em relação à aplicação das metodologias da teoria de controle convencional;
- (iv) a possibilidade da implementação do conhecimento, dos aspectos intuitivos e da experiência do especialista em controle, utilizando regras lingüísticas (conhecimento qualitativo e estruturado) e entradas imprecisas (NAVARRO & ALBERTOS, 1996);
- (v) a facilidade de entendimento, a simplicidade de implementação, a rápida prototipação e o custo inexpressivo de desenvolvimento (WANG, 1994).

Contudo, os *SNs* apresentam limitações, tais como:

- (i) alguns autores têm discutido que os *CLNs* são adequados onde o modelo matemático preciso do processo a ser controlado não é avaliado, mas é uma tarefa desafiadora construir-se um controlador sem fazer considerações sobre o ambiente (GODJEVAC, 1993);
- (ii) a quantidade de parâmetros a serem configurados. Exemplos: o número de subconjuntos de cada variável, o número de regras, a seleção do método à realização de operações lógicas, o método de desnebulização e os parâmetros das funções de pertinência. A determinação e a sintonia destes parâmetros não é sempre trivial. Um exemplo é o desenvolvimento de sistemas especialistas nebulosos, onde a configuração (adição e remoção) de regras é uma tarefa complexa;

- (iii) a dificuldade em analisar os aspectos de otimalidade e robustez do controlador;
- (iv) a falta de capacidades de aprendizado e memorização. Este aspecto motiva o desenvolvimento de sistemas híbridos neural-nebuloso, que estão se tornando populares para algumas aplicações;
- (v) a inexistência de uma metodologia analítica consolidada para o projeto de *SNs*. Algumas tendências são a utilização de redes neurais (HAYASHI *et al.*, 1995, NAUCK, 1995) e algoritmos evolutivos (LINKENS & NYONGESA, 1995, 1996b, TZES *et al.*, 1998, COELHO & COELHO, 1998b).

3.4 Aplicações industriais

Nos últimos anos, nota-se um rápido crescimento no número de aplicações e sofisticação dos sistemas de controle que empregam conceitos dos *SNs*. As áreas de aplicações dos *SNs* abrangem diversos campos do conhecimento tais como: controle de processos, identificação, análise quantitativa e qualitativa, reconhecimento de padrões e recuperação de informações. Nestas áreas têm-se aplicações em transportes, produtos eletrônicos de consumo, robótica, indústrias (química, geração de energia, construção, nuclear, aeroespacial), engenharia, gerenciamento, agricultura, aplicações em modelagem de sistemas, predição, controle ótimo, controle robusto e adaptativo (SCHWARTZ *et al.*, 1994, LU, 1992).

Um compêndio abrangente de produtos, aplicações, *hardware* e ferramentas de *software* disponíveis no mercado com utilização de técnicas baseadas em *SNs*, são apresentadas nas tabelas A.3 e A.4 do apêndice A.

3.5 Estrutura geral dos sistemas nebulosos

Os *SNs* baseados em regras são, usualmente, compostos por quatro blocos principais de acordo com a figura 3.1 (GODJEVAC, 1993):

- (i) *base de conhecimento* (as regras e os parâmetros das funções de pertinência);
- (ii) *unidade de inferência* (as operações de inferência sobre as regras);
- (iii) *procedimento de nebulização* (a transformação das entradas *crisp* em conjuntos nebulosos). Uma observação relevante é que nem todos os *SNs* tem esta fase;
- (iv) *procedimento de desnebulização* (a transformação dos resultados nebulosos de inferência em saídas da dados *crisp*).

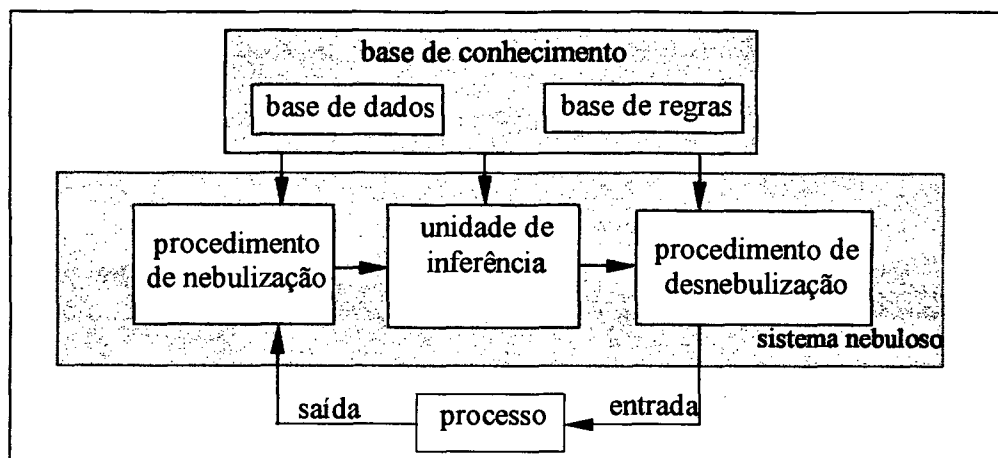


Figura 3.1. Estrutura geral de inferência nos SNs.

3.6 Modelos nebulosos: abordagens e revisão da literatura

Existem dois tipos de conhecimento, que podem ser utilizados à resolução de problemas reais, que são:

- (i) *conhecimento objetivo* (representado por modelos matemáticos), e
- (ii) *conhecimento subjetivo* (representado por informação lingüística, que é, praticamente impossível de quantificar pela matemática convencional).

Estas duas formas de conhecimento podem ser coordenadas de maneira eficiente através da utilização dos SNs. As formas de tratar o conhecimento via SNs são: abordagem baseada em modelo e livre de modelo.

A abordagem baseada em modelo nebuloso é obtida através de métodos de identificação que levam em conta as medidas de entrada e saída do processo. Esta abordagem oferece uma alternativa aos modelos convencionais no tratamento de processos complexos. Os modelos nebulosos podem ser utilizados para projeto de um controlador ou sendo parte de uma estrutura de controle baseada em modelo. Alguns exemplos são:

- (i) as formas de controle baseadas em mecanismos de ganho escalonado,
- (ii) os sistemas de controle para o tratamento de processos de fase mínima projetados pela inversão de um modelo nebuloso do processo,

- (iii) a configuração de algoritmos de controle preditivo, utilizando o modelo nebuloso na função de um previsor, enquanto as ações de controle são calculadas através de um algoritmo de otimização (KIRIAKIDIS, 1998).

A literatura têm apresentado diversos estudos de *SNs* em identificação e controle de processos, destacando-se as seguintes arquiteturas de *SNs*: modelo de Takagi-Sugeno-Kang (TAKAGI & SUGENO, 1985, SUGENO & KANG, 1986); modelo baseado em regras de Mamdani (MAMDANI & ASSILIAN, 1975); e modelo relacional nebuloso (NAZETTA *et al.*, 1998, PEDRYCZ, 1989). Entre os algoritmos que utilizam estas arquiteturas destacam-se estudos relativos a:

- (i) combinação de redes neurais (LINKENS & NYONGESA, 1996a) e *AEs* (CÓRDON *et al.*, 1997) em projeto de *SNs*;
- (ii) projeto de *SNs* via filtro de Kalman (KOSKO, 1992);
- (iii) métodos de *mountain-clustering* (YAGER & FILEV, 1994);
- (iv) método *C-médias* nebuloso (KIM *et al.*, 1997b);
- (v) métodos baseados em gradiente (WANG, 1994);
- (vi) regra Widrow-Hoff, para o aprendizado de conseqüentes (YAGER & FILEV, 1994);
- (vii) aprendizado por reforço (BERENJI, 1992).

3.6.1 Tipos de modelos nebulosos

A estrutura básica de um modelo nebuloso pode ser vista com constituindo-se de:

- (ii) *base de regras*, que contém um conjunto de regras nebulosas;
- (iii) *base de dados*, que define as funções de pertinência das regras nebulosas;
- (iv) *mecanismo de inferência*, que executa os procedimentos de inferência sobre as regras e condição para obtenção de uma saída.

Nesta tese aborda-se os modelos baseados em regras. Conforme JANG & SUN (1995), a essência dos modelos nebulosos é similar ao princípio de “dividir e conquistar”. Os antecedentes das regras nebulosas dividem o espaço de entrada em um número de regiões nebulosas locais, enquanto os conseqüentes descrevem o comportamento dentro de uma região através dos seus constituintes.

Os componentes dos conseqüentes resultam em tipos diferentes de modelos nebulosos, mas seus antecedentes são essencialmente os mesmos. Nesta tese adota-se uma classificação baseada nos componentes classificados por apresentar (YAGER & FILEV, 1994; YEN & WANG, 1999):

- (i) *modelos lingüísticos*: são baseados em regras *se-então* apresentando predicados vagos e utilizando raciocínio nebuloso. Nestes modelos, as quantidades nebulosas são associadas aos termos lingüísticos, e o modelo nebuloso é essencialmente uma expressão qualitativa do sistema. Os modelos, deste tipo, formam a base de modelagem qualitativa, que descreve o comportamento do sistema através da utilização da linguagem natural. O representante mais significativo desta classe é o modelo do tipo Mamdani (MAMDANI & ASSILIAN, 1975);
- (ii) *modelos interpolativos*: são formados por regras lógicas por uma parte nebulosa antecedente e um conseqüente funcional. Estes modelos são, essencialmente, uma combinação de modelos nebulosos e não-nebulosos. Os representantes mais significativos dos modelos interpolativos são os modelos do tipo Takagi-Sugeno-Kang e do tipo Tsukamoto. A representação do conhecimento, nestes modelos, caracterizam-se pelo potencial de expressar informações quantitativas. Entretanto, nos modelos interpolativos perde-se a informação qualitativa.

As principais características destes modelos nebulosos são apresentadas nas próximas seções.

3.6.1.1 Modelo nebuloso do tipo Mamdani

O modelo de Mamdani (MAMDANI & ASSILIAN, 1975) caracteriza-se por utilizar os conjuntos nebulosos como conseqüentes das regras de produção, ou seja,

$$R_i: \text{ se } x_1 \text{ é } A_{i,1} \text{ e } \dots \text{ e } x_m \text{ é } A_{i,m} \text{ então } y \text{ é } C_i \quad (3.1)$$

onde x_i são as variáveis de entrada, y é a variável de saída, A_i e C_i são conjuntos nebulosos. Um exemplo, apresentado na figura 3.2, ilustra o sistema de inferência do tipo Mamdani com duas regras, que apresentam duas entradas *crisp*, x_1 e x_2 , e uma saída y . A composição dos operadores de mínimo e máximo para esta representação são adotadas. Contudo, existem estudos com a composição de operadores com *t-normas* e *t-conormas* (JANG & SUN, 1995).

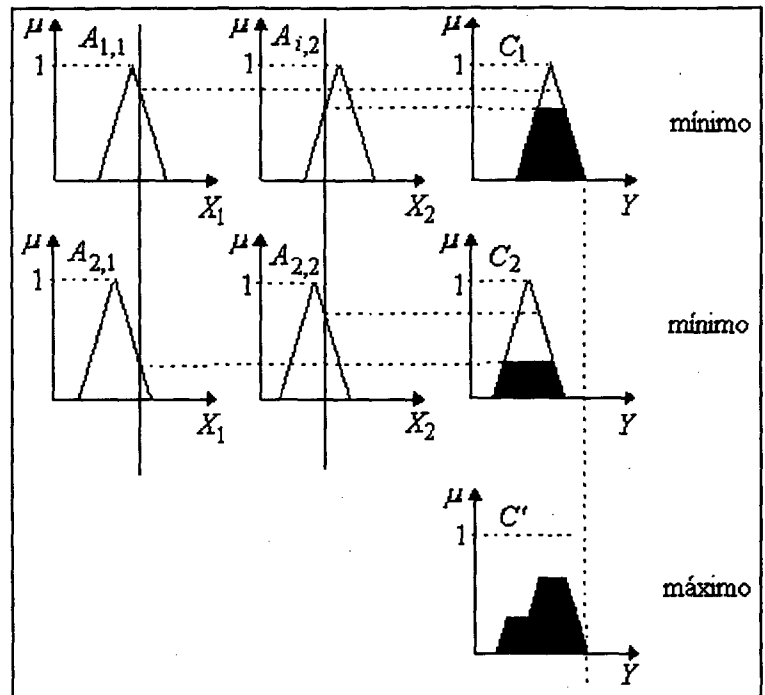


Figura 3.2. Exemplo do modelo nebuloso de Mamdani.

3.6.1.2 Modelo nebuloso interpolativo

Nesta forma de modelo, cada conclusão de uma regra é apresentada através de uma função estritamente monotônica. No modelo de Takagi-Sugeno-Kang (TSK), a função é uma combinação linear das entradas. No modelo de Tsukamoto, a função é, usualmente, não linear, tendo como domínio os possíveis graus de compatibilidade entre as premissas e as entradas.

3.6.1.2.1 Modelo interpolativo do tipo Takagi-Sugeno-Kang

Os trabalhos em identificação de parâmetros de modelos nebulosos relevantes seguiram a publicação de TAKAGI & SUGENO (1985) e SUGENO & KANG (1986), que introduziram um novo método de tratamento nebuloso, denominado atualmente de

método *TSK*. A abordagem *TSK* possibilitou o desenvolvimento de modelos nebulosos através da simplificação e da representação formal do modelo de Mamdani.

Os modelos de *TSK* empregam funções lineares das entradas como conseqüentes. Devido a esta diferença, as regras do modelo de Mamdani são, significativamente, mais intuitivas lingüisticamente, enquanto as regras do modelo *TSK* apresentam maior poder de interpolação, com um número reduzido de regras de produção (YING, 1998). As regras do modelo *TSK* apresentam a seguinte forma:

$$R_i: \text{ se } x_1 \text{ é } A_{i,1} \text{ e } \dots \text{ e } x_m \text{ é } A_{i,m} \text{ então } y \text{ é } f_i(x_1, x_2, \dots, x_m) \quad (3.2)$$

onde $f_i(x_1, x_2, \dots, x_m)$ é uma função *crisp* no conseqüente. A função $f_i(x_1, x_2, \dots, x_m)$ é um polinômio, usualmente, com as variáveis de entrada x . Quando $f_i(x_1, x_2, \dots, x_m)$ é um polinômio de primeira ordem, o sistema de inferência nebuloso é denominado de *modelo nebuloso TSK de primeira ordem*. Quando $f_i(x_1, x_2, \dots, x_m)$ é uma constante, tem-se um *modelo nebuloso TSK de ordem zero*, que pode ser visto como um caso especial do sistema de inferência de Mamdani, em que cada regra do conseqüente é especificada por um *singleton* nebuloso (ou conseqüente previamente desnebulizado). Também, pode configurar-se em um caso especial do modelo nebuloso de Tsukamoto.

Além disso, um *modelo nebuloso TSK de ordem zero* é funcionalmente equivalente a uma rede neural de base radial, sob certas restrições (JANG & SUN, 1993, BERSINI *et al.*, 1995). Um exemplo, na figura 3.3, ilustra o sistema de inferência do tipo *TSK* com duas regras.

A vantagem do modelo *TSK* consiste do seu poder de representação, especialmente para a descrição de processos complexos. Este modelo nebuloso permite a decomposição de um sistema complexo em sub-sistemas simples (YAGER & FILEV, 1994).

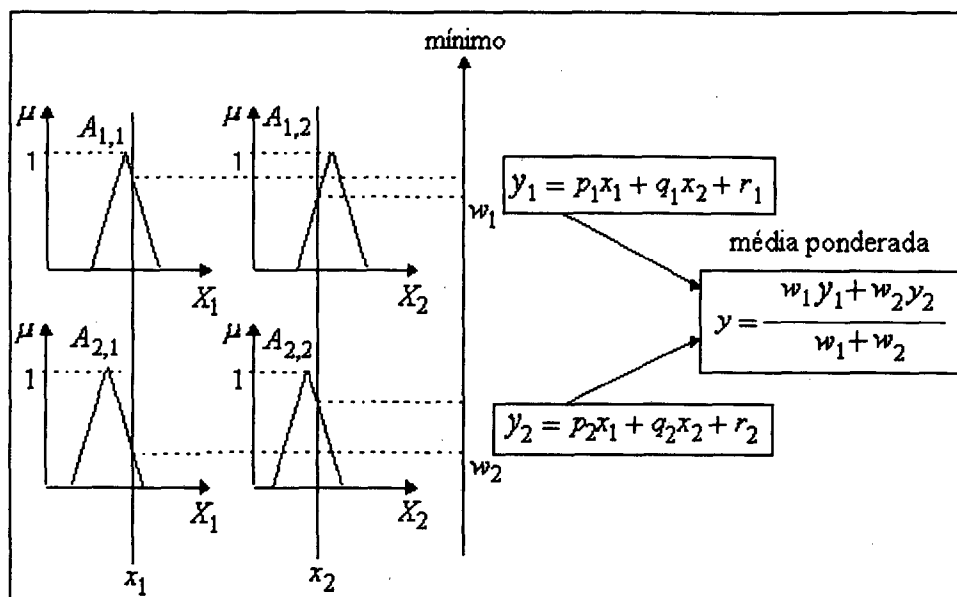


Figura 3.3. Exemplo do modelo nebuloso de TSK.

A identificação de um modelo nebuloso TSK utilizando-se os dados de entrada e saída é composto de duas partes: identificação da estrutura e dos parâmetros. Na identificação da estrutura identifica-se a estrutura do antecedente (premissa) e do conseqüente. Os parâmetros do conseqüente são coeficientes de equações lineares. Esta forma de desenvolvimento de SNs é inspirada na teoria clássica de sistemas e, alguns desenvolvimentos no campo das redes neurais artificiais. Na figura 3.4 são apresentados os passos para identificação de um modelo nebuloso do tipo TSK (SUGENO & KANG, 1988).

Os modelos de TSK apresentam algumas outras variantes. O modelo nebuloso denominado *quase linear* consiste de uma interpretação dinâmica para o modelo TSK. Este modelo é obtido pela substituição das variáveis de entrada no modelo TSK por valores das saídas passadas do processo e os valores atuais e passados dos valores de entrada. O modelo nebuloso *quase não linear* é uma generalização do conceito de *quase linear* e pode ser obtido se os sub-sistemas não lineares são considerados (YAGER & FILEV, 1994).

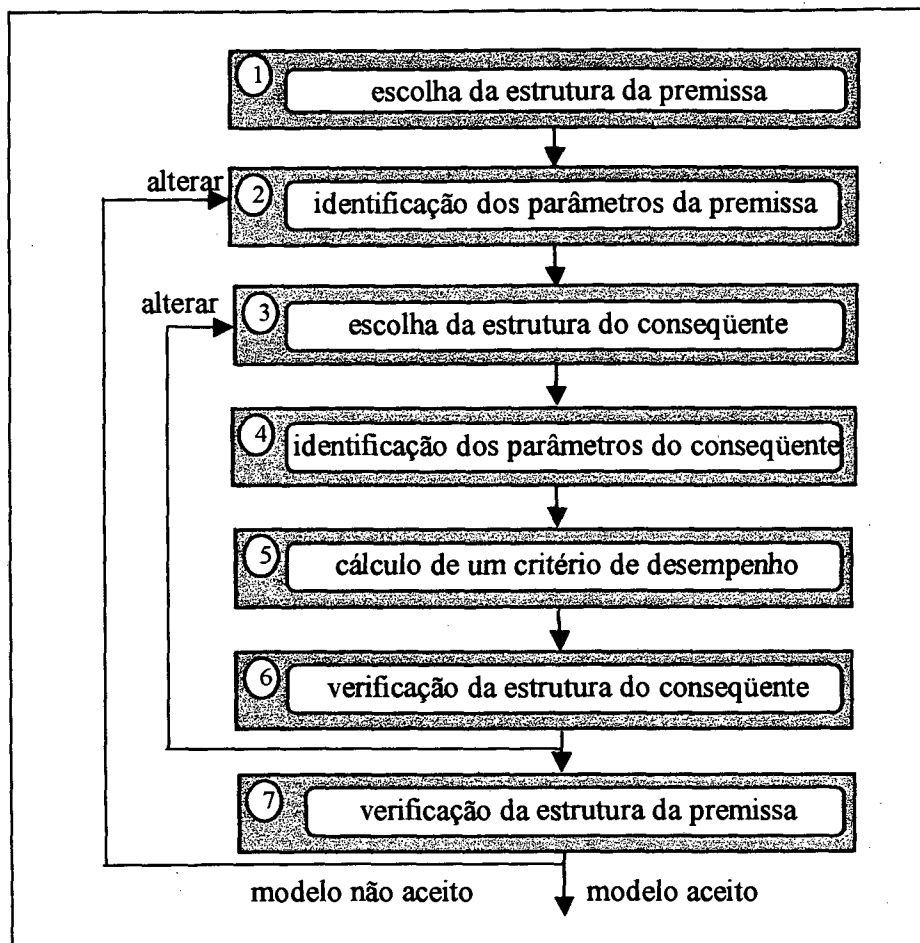


Figura 3.4. Passos para identificação de um modelo nebuloso do tipo *TSK*.

3.6.1.2.2 Modelo interpolativo do tipo Tsukamoto

No modelo de Tsukamoto, cada conseqüente de uma regra é especificado por um grau de pertinência de uma função degrau que é combinada a uma constante (JANG & SUN, 1995). O modelo de TSUKAMOTO (1979) exige que pelo menos os conjuntos nebulosos C_j , que estão associados com os conseqüentes das regras, sejam funções monotônicas. O método de interpolação obtém um valor y_j^s relativo à ação da regra R_j , dado por:

$$y_j^s = \mu_{C_j}(\alpha_j) \tag{3.3}$$

onde α_j são constantes de ponderação. As regras são agregadas e os valores da conclusão são agregados em uma única ação de controle y^s , através de um média ponderada, pela equação:

$$y^s = \frac{\sum_{j=1}^n (\alpha_j \cdot y_j^s)}{\sum_{j=1}^n \alpha_j} \tag{3.4}$$

Na figura 3.8 é apresentado um exemplo do modelo nebuloso de Tsukamoto.

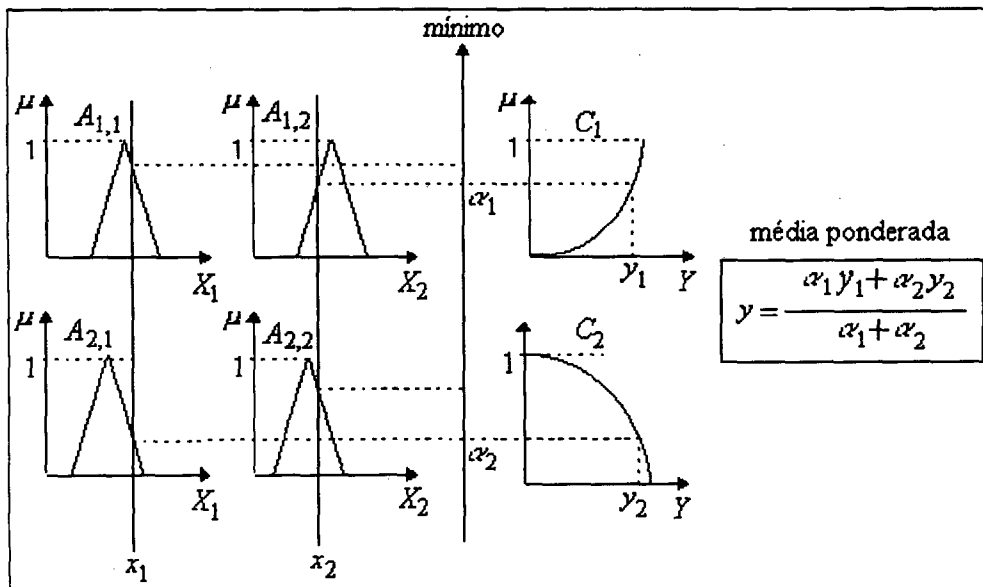


Figura 3.5. Exemplo do modelo nebuloso de Tsukamoto.

3.7 SNs em identificação de processos

A configuração de modelos nebulosos, usualmente, envolve três estruturas, ilustradas pela figura 3.6. A primeira estrutura é relacionada com a informação do processo contida no conjunto de treinamento. A informação é providenciada pelo conhecimento sobre as características físicas do processo, por alguns dados numéricos adquiridos dele, e pelo conhecimento lingüístico, expresso através das regras. Esta

estrutura envolve o *conjunto das variáveis*, que regem o comportamento do processo, e a *partição nebulosa* atribuída a cada uma destas variáveis.

A segunda estrutura é o algoritmo de aprendizado, baseado nas características da primeira estrutura. Nesta estrutura, a informação adquirida é interpretada e codificada em um conjunto de regras. A terceira estrutura executa o procedimento de inferência considerando as regras extraídas e realiza as ações sobre o processo, no caso de sistemas dinâmicos, completando o ciclo de identificação nebulosa (BRANCO & DENTE, 1997).

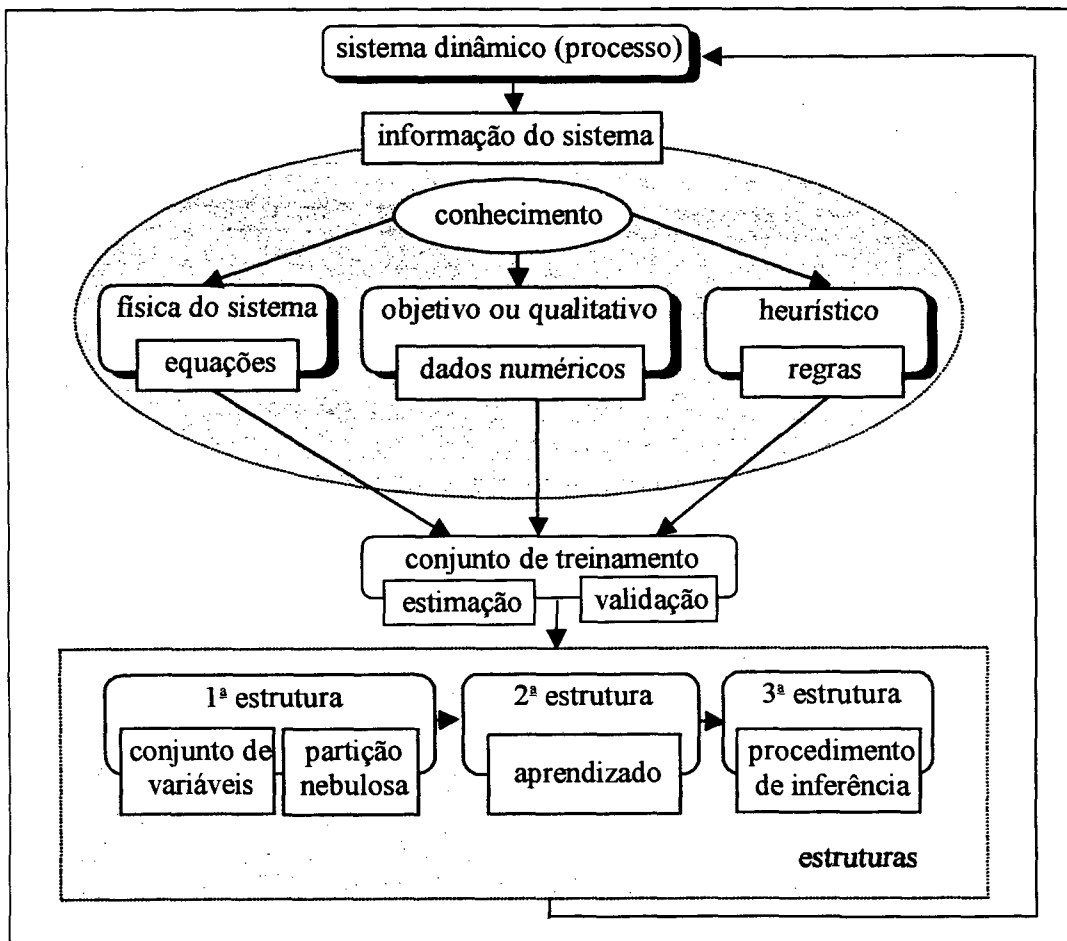


Figura 3.6. Esquema de configuração de identificação nebulosa e suas estruturas.

Entre as características de aproximação, as seguintes são relevantes à utilização dos SNs em identificação de processos (ZENG & SINGH, 1995):

- (i) *o mecanismo de aproximação nebuloso*: As regras linguísticas são utilizadas para aproximar a saída desejada. Para uma dada entrada, as saídas correspondentes do SN

resultam da combinação de diversas regras e consistem numa aproximação da saída desejada;

- (ii) *a aproximação global limitada*: a aproximação dos SNs é baseada em informação local, pois a base de regras é projetada de acordo com o conhecimento local;
- (iii) *propriedade de convergência*: quando um SN não satisfaz a uma precisão de aproximação definida, uma idéia intuitiva, para modificar o sistema e melhorar o desempenho, é dividir o espaço de entrada em mais regiões (modificação da granularidade).

A seguir são apresentadas duas abordagens para o treinamento de modelos nebulosos do tipo TSK.

3.7.1 Sistema de Inferência Nebuloso com Aprendizado Modificado de Exemplos

Um método geral, para geração de regras nebulosas, de dados numéricos denominado *geração de regras através de aprendizado* foi proposto por WANG & MENDEL (1992). Um aprimoramento posterior desta metodologia — *geração de regras através de aprendizado modificado (MLFE)* — foi proposto por LAUKONEN & PASSINO (1995) e é utilizado nesta tese. Dado um sistema nebuloso parametrizado (equação 3.2),

$$f_i(x, c, \sigma) = \frac{\prod_{j=1}^{ne} \exp\left(-\frac{(x_j - c_{i,j})^2}{2\sigma_{i,j}^2}\right)}{\sum_{i=1}^{Nr} \prod_{j=1}^{ne} \exp\left(-\frac{(x_j - c_{i,j})^2}{2\sigma_{i,j}^2}\right)} \quad (3.5)$$

$$y = \sum_{i=1}^{Nr} f_i(x, c, \sigma) y_i \quad (3.6)$$

onde y_i são os conseqüentes, $c_{i,j}$ e $\sigma_{i,j}$ são as aberturas e desvio padrão das funções Gaussianas, respectivamente. Os parâmetros $c_{i,j}$, $\sigma_{i,j}$ e o número de regras, Nr , são escolhidos para aproximar uma função não linear baseado nos conjuntos de dados de entrada e saída, onde ne é o número de entradas.

A primeira questão na construção de um sistema nebuloso é a escolha do número de regras nebulosas. Muitas regras significam aumento da complexidade computacional na implementação. Em contrapartida, a utilização de muitas regras nebulosas pode produzir um sistema nebuloso com maior capacidade funcional. Entretanto, existe um compromisso, entre a complexidade computacional e a capacidade funcional, baseado no número de regras.

O valor de ξ caracteriza a precisão desejada com que o sistema nebuloso executa a aproximação de funções. Especificamente, as saídas do sistema nebuloso são comparadas com a saída do conjunto de treinamento. Se esta comparação é maior que ξ , isto é, se $|f(x_i, c, \sigma) - y_i(k)| < \xi, \forall i$, o sistema nebuloso atual é ampliado com a adição de uma nova regra.

A escolha do ξ , pelo projetista, determina o número de regras nebulosas. Um valor menor para ξ , significa, geralmente, mais regras nebulosas para um conjunto de treinamento e vice-versa. A utilização de ξ no algoritmo para a identificação de processos é esboçado a seguir.

3.7.1.1 Passos do algoritmo *MLFE* para identificação de processos

O método *MLFE* consiste das seguintes etapas:

(i) *construir um sistema nebuloso inicial*

A primeira etapa do algoritmo é formar uma regra nebulosa de acordo com o primeiro vetor dos dados de treinamento.

(ii) *avaliar os novos dados de treinamento*

A cada novo vetor de dados de entrada(s) e saída(s) avalia-se o valor de ξ , equação (3.6), em relação ao sistema nebuloso atual. Se uma regra adicional é necessária, então seguir para a etapa (iii), caso contrário, retorna-se ao início da etapa (ii), e avalia-se os próximos dados do conjunto de treinamento.

(iii) *modificar o sistema nebuloso atual*

Neste caso, modifica-se alguns parâmetros do sistema nebuloso original, isto é, $\sigma_{i,j}$ é modificado para considerar a nova informação contida no conjunto de entrada e saída. Uma nova regra é adicionada para o sistema nebuloso, ou seja, $Nr=Nr+1$.

A modificação de $\sigma_{i,j}$ ajusta o espaço entre as funções de pertinência, para as regras no sistema nebuloso aumentado, de forma que: a adição de uma regra não altere o que tinha sido aprendido *a priori* pelo sistema nebuloso e a interpolação entre os pontos seja mantida. A modificação do valor de $\sigma_{i,j}$ é realizada para cada regra do vizinho mais próximo, denotado pelo índice i^* , tal que,

$$i^* = \arg \min \left\{ \left\| c_i^e - c_i \right\|_2 : i^e = 1, \dots, N, i^e \neq i_j \right\} \quad (3.7)$$

onde $c_i = [c_i^1, \dots, c_i^n]^T$. Atualizado para $j=1, \dots, ne$, sendo:

$$\sigma_{i,j} = \frac{1}{W^{sf}} |c_{i,j} - c_{i^*,j}| \quad (3.8)$$

onde W^{sf} é um termo ponderado que rege a sobreposição entre as regras da vizinhança mais próxima. Um valor maior de W^{sf} significa uma menor sobreposição das funções de pertinência e vice-versa. Uma vez finalizada a etapa (iii) deve-se retorna para a etapa (ii), à análise do vetor do próximo vetor de dados de entrada(s) e saída(s), até que o número de dados de todo conjunto de treinamento seja verificado.

3.7.2 Modelo TSK com *clustering* e mínimos quadrados recursivo

O modelo nebuloso do tipo *TSK* é descrito por regras de produção *se-então*, que representam relações lineares de entrada e saída de um sistema. O modelo nebuloso é descrito pela forma:

$$R_i: \text{ se } x_1 \text{ é } A_{i,1} \text{ e } \dots \text{ e } x_m \text{ é } A_{i,m} \text{ então } y = b_{i0} + b_{i1}x_1 + \dots + b_{im}x_m \quad (3.9)$$

O modelo nebuloso, empregando funções de base, caracteriza-se por apresentar as componente do conseqüente determinadas por uma constante, sendo um caso particular do modelo nebuloso de *TSK*. Este tipo de modelo nebuloso tem a seguinte forma:

$$R_i: \text{ se } x_1 \text{ é } A_{i,1} \text{ e } \dots \text{ e } x_m \text{ é } A_{i,m} \text{ então } y_i = b_i \quad (3.10)$$

onde m e Nr são o número de variáveis de entrada e as regras, respectivamente; x_i e y_i são as variáveis de entrada e saída, respectivamente, $A_{i,j}$ são as funções de pertinência das variáveis de entrada; e b_i não é um conjunto nebuloso e são as componentes constantes do conseqüente. As funções de pertinência apresentam a forma Gaussiana,

$$A_{i,j}(x_j) = \exp\left(-\frac{(x_j - c_{i,j})^2}{2\sigma_{i,j}^2}\right) \quad (3.11)$$

onde $c_{i,j}$ e $\sigma_{i,j}$ são os centros e os comprimentos das funções Gaussianas, respectivamente. Assim, utilizando-se a desnebulização média dos centros, a saída total dos modelos nebulosos *MISO* podem ser calculadas por (WANG, 1994):

$$y = \sum_{i=1}^{Nr} v_i b_i \quad (3.12)$$

onde v_i é o valor que informa a importância da i -ésima regra, definido da mesma forma que a equação (3.5) que pode ser vista como um caso especial do modelo de regressão linear,

$$y(k) = \sum_{i=1}^{Nr} p_i(k)\theta_i + e(k) \quad (3.13)$$

onde $p_i(k)$ e θ_i são dados por

$$p_i(k) \equiv v_i \quad \theta_i \equiv b_i \quad (3.14)$$

onde $p_i(k)$ são conhecidos como os regressores, θ_i são os parâmetros, $e(k)$ é um sinal de erro, que assume-se ser descorrelacionado como os regressores $p_i(k)$.

Neste trabalho, os centros $c_{i,j}$ e os comprimentos $\sigma_{i,j}$ das funções de pertinência Gaussianas são determinados pelo algoritmo de *clustering k-médias* e uma heurística de

nearest-neighbor (MOODY & DARKEN, 1989). Isto significa que os antecedentes das regras nebulosas são conhecidos *a priori* e, somente os parâmetros desconhecidos em (3.14) são θ_i e o problema é linear nos parâmetros. Os algoritmos adotados para esta finalidade foram a equação normal dos mínimos quadrados (JOHANSSON, 1993) ou o algoritmo dos mínimos quadrados recursivo (WATANABE & IMAIZUMI, 1997).

3.8 Controle lógico nebuloso

O *CLN* é geralmente, um projeto de controle não linear, sendo uma tarefa difícil examinar a influência de cada parâmetro no desempenho e na robustez do projeto. Segundo REZNIK (1997), o projeto do *CLN* pode ser sintetizado pelos seguintes procedimentos:

- (i) selecionar a estrutura e as entradas do *CLN*;
- (ii) escolher os fatores de escala e definir o número e a forma de sintonia das funções de pertinência das entradas e das saídas;
- (iii) especificar as regras associadas às funções de pertinência;
- (iv) selecionar o método de inferência associado às regras de produção;
- (v) selecionar o método de desnebulização e os seus operadores;
- (vi) avaliar o *CLN*.

A filosofia do projeto de *CLNs* caracteriza-se pelo tratamento do conhecimento decorrente do procedimento de aquisição do conhecimento sobre o processo ou são sintetizados automaticamente por meio de algoritmos de controle auto-ajustáveis. Existem diversos métodos para o projeto de *CLN*, tais como:

- (i) a modelagem do conhecimento do engenheiro de controle;
- (ii) a modelagem das ações dos operadores humanos e sua experiência;
- (iii) a identificação nebulosa do processo a ser controlado, ou seja, desenvolvimento de controle baseado em modelo.

O resultado da configuração de um *CLN* do tipo Mamdani pode ser interpretado como uma superfície de controle refletindo o conhecimento *a priori* do operador (ou engenheiro de controle). A superfície de controle é codificada em uma base de conhecimento, sob forma de um conjunto de regras, termos lingüísticos e fatores de escala. As regras são usualmente executadas por um controlador dedicado, interpretador ou compilador. O projeto dos *CLNs* apresentam um fluxo de dados, representado na figura

3.7, passando pelas fases de nebulização (*fuzzification*), análise e execução de regras, e desnebulização (*defuzzification*).

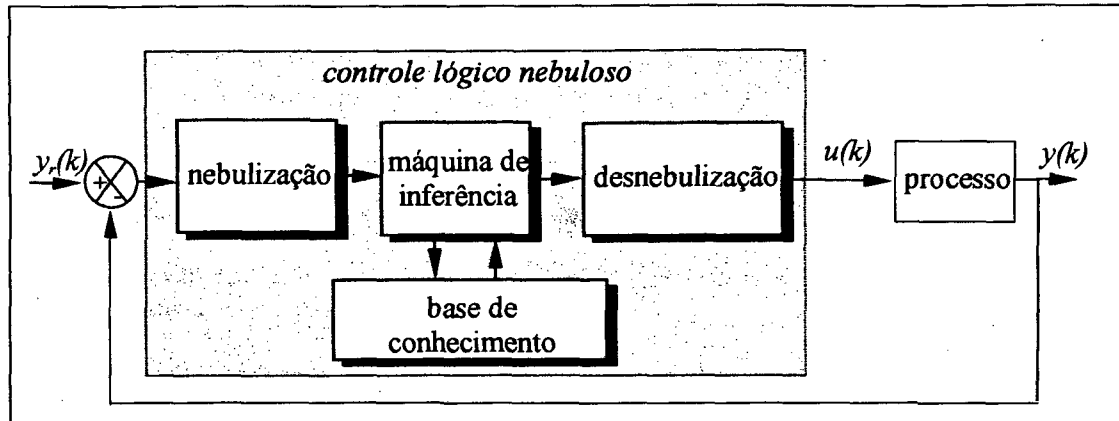


Figura 3.7. Configuração básica de CLN do tipo Mamdani.

3.8.1 Base de conhecimento

A base de conhecimento abrange a base de regras e a base de dados do CLN. A base de regras é projetada pela escolha das variáveis de entrada, saída e regras de controle. A base de dados está vinculada à definição do universo de discurso de cada variável de estado, dependendo da escolha das funções de pertinência e dos parâmetros e das funções, que as regem.

3.8.2 Fase de nebulização (*fuzzification*)

A fase de nebulização converte os valores numéricos (números reais) em conjuntos nebulosos, ou seja, transforma os dados *crisp* em um conjunto nebuloso correspondente.

3.8.3 Mecanismo de inferência

A fase de análise e execução de regras é responsável pela avaliação das regras de produção. Quando uma regra é ativada efetua-se um procedimento de cálculo baseado nos valores dos antecedentes e, então é obtida a saída da regra.

3.8.4 Fase de desnebulização (*defuzzification*)

A fase de desnebulização visa a transformação das variáveis linguísticas da saída do *CLN* em saídas *crisp*. A desnebulização descreve o mapeamento de um espaço de ações de controle nebuloso em ações de controle não nebuloso. Os métodos de desnebulização mais utilizados são os seguintes (SANDRI & CORREA, 1999):

- (i) *primeiro máximo (SOM)*: o valor de saída é obtido através do ponto em que o grau de pertinência, da distribuição da ação de controle atinge o primeiro valor máximo;
- (ii) *média dos máximos (MOM)*: o valor de saída é obtido através do ponto médio entre os valores que têm o maior grau de inferência pela utilização das regras;
- (iii) *centro da área (COA)*: o valor de saída é obtido através do centro de gravidade da função de distribuição de possibilidade da ação de controle.

3.9 Controle lógico nebuloso multivariável

O projeto de *CLN* para uma ou duas dimensões é relativamente simples, mas existem problemas quanto ao projeto de *CLNs* para aplicações multivariáveis, principalmente devido:

- (i) ao limite da intuição do especialista para as dimensões elevadas;
- (ii) as complicações no tratamento matemático dos conceitos dos *SNs* em problemas multivariáveis, decorrentes da possível interação entre as malhas de controle;
- (iii) ao crescimento exponencial do número de regras com o número de estados; e
- (iv) ao fato de que a elucidação simplificada da base de regras para um operador ser uma tarefa complexa.

Um *SN* tipo *MIMO* pode ser sempre separado em um grupo de *SNs* tipo *MISO*, sem perda de generalidade. Assume-se, neste trabalho, que os *CLN* são *CLN-MISO*. Os esquemas de *CLN* multivariável utilizados são baseados no esquema apresentado na

figura 3.6, aplicado a processos *MISO* (figura 3.8(a)) e *MIMO* (figura 3.8(b)). O acoplamento entre as variáveis não está representado na figura 3.8.

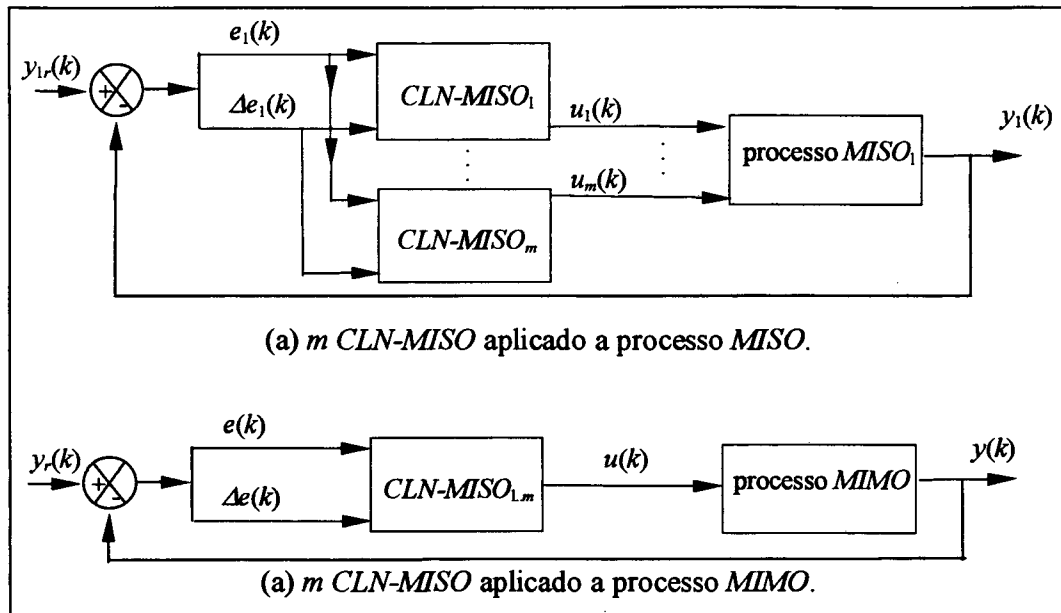


Figura 3.8. Esquema de CLN multivariável.

Uma metodologia usualmente utilizada nas aplicações de controle *MIMO*, é a utilização de múltiplos *CLN-PI* (ou *CLN-PD*) com estrutura *MISO*. Entretanto, a não consideração do acoplamento entre as variáveis das malhas de controle pode causar um desempenho em malha fechada com característica oscilatória e também ampliar dificuldades na sintonia do *CLN*.

Nos últimos anos, os *CLNs* combinados a estruturas de controle *PID* são utilizados em controle de processos industriais devido a sua natureza heurística associada com a simplicidade e a eficiência de ambos sistemas de controle linear e não linear. Muitos dos *CLNs*, empregados em processos monovariáveis, são essencialmente do tipo *PD*, *PI* ou *PID* com ganhos não lineares. Devido a não-linearidade dos ganhos de controle, os *CLNs* do tipo *PID* possuem desempenho superior aos controladores *PID* convencionais fornecendo o tratamento de não-linearidades, quando sintonizados de forma adequada. A presença de não-linearidades dificulta as análises teóricas para explicar todos os mecanismos que fazem com que os *CLNs* tipo *PID* possam apresentar melhor desempenho (LIU *et al.*, 1997).

As estruturas e os princípios de projeto de alguns *CLNs* do tipo *PID*, são apresentados. Os *CLN-PID* apresentados configuraram-se em alternativas ao projeto de *CLN-PID* completo com três entradas, que expande a base de regras substancialmente e implica em diversas dificuldades no controle (LI & GATLAND, 1995; LI & NG, 1997).

A ênfase desta seção é ao projeto e controle de processos *MIMO*. Uma análise detalhada da base teórica dos *CLNs-PID* é tratada por QIAO & MIZOMOTO (1996) e MISIR *et al.* (1996). Os *CLNs-PID* objetivam um compromisso entre um controlador *PID* convencional e um sistema de *CLN*. Devido a sua flexibilidade de projeto, os *CLNs-PID* têm a vantagem dos sólidos fundamentos teóricos do controle *PID* combinados com a flexibilidade dos *CLNs* (PEREIRA & BOWLES, 1996). Segundo MANN *et al.* (1999), existem diferentes estruturas de *CLN* do tipo *PID* combinando os elementos estruturais:

- (i) erro, $e(k) = y(k) - y_r(k)$;
- (ii) variação do erro, $\Delta e(k) = e(k) - e(k-1)$;
- (iii) taxa de variação do erro, $\Delta^2 e(k) = \Delta e(k) - \Delta e(k-1)$; e
- (iv) somatório do erro, $\sum e(k) = \sum_{k=0}^n e(k)$.

Os elementos utilizados para a configuração de um controlador *PID* nebuloso (posicional ou incremental) podem ser constituídos de elementos de uma, duas ou três entradas, conforme as opções apresentadas na figura 3.9. Além disso, estes elementos estruturais podem ser combinados a estruturas *P*, *PD*, *PI*, *I* e *D* convencionais, não representados na figura 3.9.

O controle *PID* é muito utilizado em sistemas de controle, principalmente devido a sua simplicidade e eficiência. Entretanto, o *PID* apresenta dificuldades no controle de processos complexos que são, usualmente, não lineares, especialmente quando a dinâmica do sistema apresenta incertezas estruturais e paramétricas.

A sensibilidade ou robustez de um *CLN* do tipo *PID* pode ser investigada pela mudança de parâmetros do modelo do processo. O *CLN* do tipo *PID*, quando projetado de forma adequada, é menos sensível a grandes mudanças paramétricas do processo e é comparável em desempenho ao controlador *PID* convencional para pequenas mudanças paramétricas.

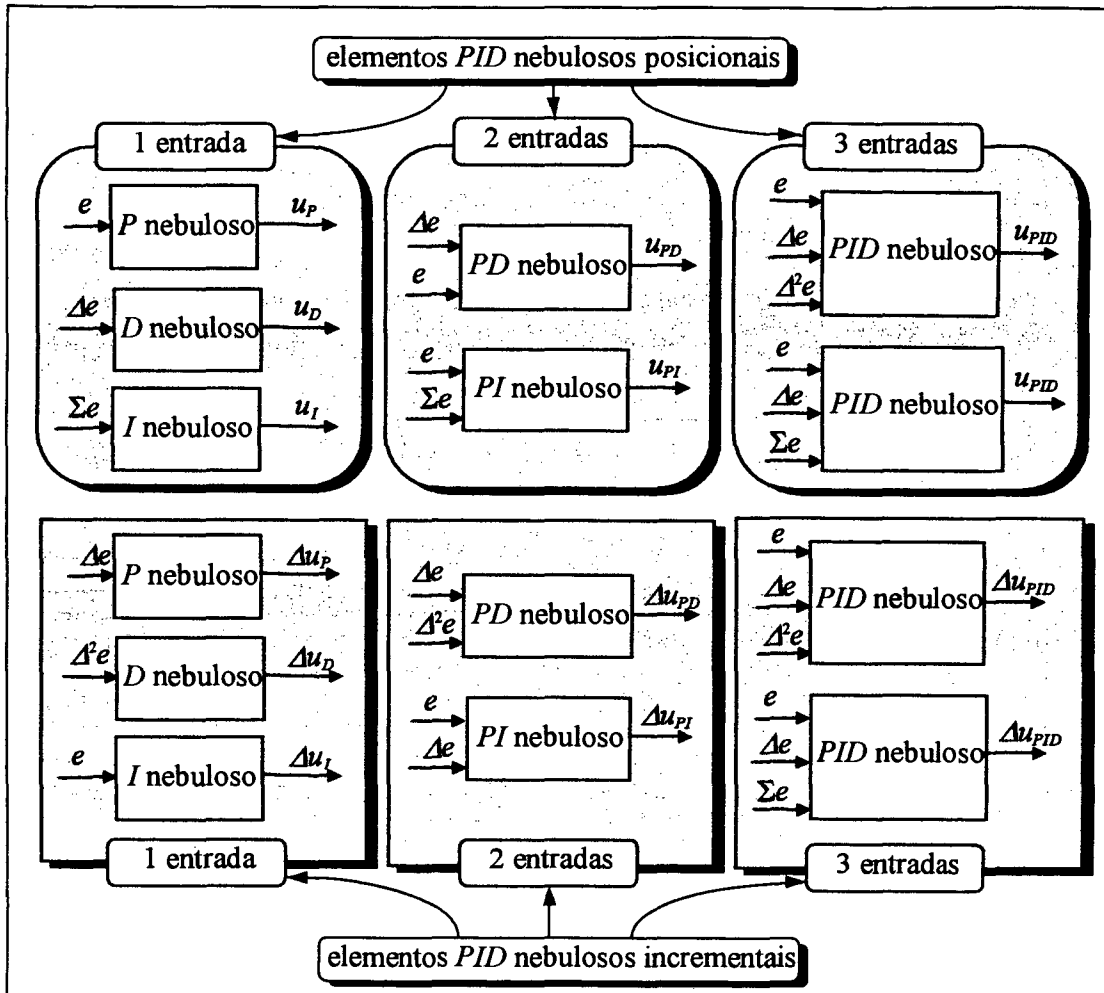


Figura 3.9. Elementos estruturais para configuração de CLN do tipo PID.

Os projetos de CLN, apresentados nas próximas subseções, de forma sintetizada, constam em COELHO & COELHO (1999a) e foram aplicados em um estudo comparativo de CLN do tipo PID, no controle de um processo experimental balanço horizontal. As abordagens dos CLNs foram apresentadas para o caso monovariável em COELHO & COELHO (1999a). Entretanto, nesta tese são estendidas para o caso de processos multivariáveis. Além disso, o projeto de CLN-PD ou CLN-PI pode ser efetuado a partir das estruturas de CLN-PID apresentadas (figura 3.9). Um detalhe relevante é que as estruturas de CLN analisadas são as do tipo Mamdani. A contribuição da tese é a verificação de aspectos de desempenho de cada projeto de CLN para o controle de processos multivariáveis.

3.9.1 Projeto $CLN-(PD+I)$

O projeto de controle $CLN-(PD+I)$ foi proposto por LI & NG (1997). O $CLN-(PD+I)$ é regido pela seguinte lei de controle:

$$u_{PID}(k) = u_{PD}(k) + \phi T_s u_I(k) \quad (3.15)$$

$$u_{PD}(k) = K_p e(k) + K_D \Delta e(k) \quad (3.16)$$

$$u_I(k) = u_I(k-1) + u_p(k) \quad (3.17)$$

$$u_p(k) = K_p e(k) \quad (3.18)$$

onde $e(k)$ é um vetor de erros entre o vetor de referência e o vetor de saídas obtidas, $y_r(k) - y(k)$; $\Delta e(k)$ é o vetor da variação do erro; n é o número de saídas do processo; m é o número de entradas, $u_{PID}(k)$ é o vetor que representa a ação de $CLN-(PD+I)$, constituída pela ação proporcional e derivativa do erro, $u_{PD}(k)$, e a ação integral do erro $u_{PID}(k)$ regida pelo vetor de controle proporcional ao erro, $u_p(k)$.

O ganho proporcional e o derivativo são multiplicados por K_p e K_D , respectivamente; T_s é o período de amostragem e ϕ é um vetor que atua como um parâmetro de seleção estrutural, sintonizando a ação integral de acordo com a natureza do processo. A característica básica do $CLN-(PD+I)$ é a necessidade da configuração de uma base de regras de duas dimensões, para o controle PD nebuloso, e uma base de regras de uma dimensão, para controle I nebuloso. A ação de controle I nebulosa utiliza a ação de controle incremental proporcional ao erro, $e(k)$, que é outra característica deste projeto. O diagrama básico do $CLN-(PD+I)$ é mostrado na figura 3.10.

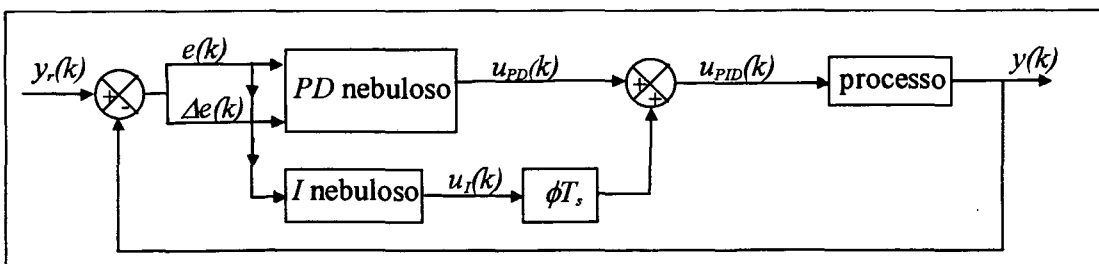


Figura 3.10. Diagrama básico do $CLN-(PD+I)$.

O $CLN-(PD+I)$ tem a vantagem de ser aplicável para processos em que o ganho em regime é conhecido ou medido facilmente, pois, neste caso, não é necessária a ação integral. Esta configuração também tem a vantagem de permitir a redução do número de regras necessárias ao projeto de um controle nebuloso PID , com três dimensões de entrada para cada controlador (estrutura *multiloop*). Esta característica tem por consequência facilitar a configuração do controlador nebuloso, por apresentar menor número de parâmetros de projeto a serem ajustados.

Uma das desvantagens deste projeto surge quando o ganho, K_p , não é conhecido e assim necessita-se da ação integral para o projeto ser eficiente e reduzir (ou eliminar) o erro de seguimento de referência em regime permanente. A parte de controle nebuloso proporcional do $CLN-(PD+I)$ é utilizada para reduzir os erros de seguimento de referência. Entretanto, o incremento do termo proporcional ao erro leva a redução do tempo de subida, pois o erro é amplificado e, conseqüentemente, o controlador apresenta uma ação de controle com valor elevado. O problema do incremento do ganho proporcional é proporcional a um comportamento oscilatório maior na saída do processo.

O projetista quando ajusta o ganho proporcional nebuloso (funções de pertinência e fator de escala) deve levar em conta um compromisso entre o tempo de subida e o comportamento oscilatório do processo. Outro aspecto relevante a ser mencionado é que o controle $u_p(k)=0$ é uma condição necessária para erro zero, em regime, na saída do processo.

3.9.2 Projeto $CLN-PI$ nebuloso + (D convencional)

O projeto $CLN-PI$ nebuloso + (D convencional), proposto por LI & GATLAND (1995) e QIN (1994), consiste de um $CLN-PI$, adicionado à ação de controle derivativa da saída do processo, composto pelo vetor $u_D(k)$, onde:

$$u_{PID}(k) = u_{PI}(k) + u_D(k) \quad (3.19)$$

sendo que a ação de controle derivativa tem a seguinte função de transferência:

$$\frac{y_D(s)}{y(s)} = \frac{T_D s + 1}{T_D s / K_D + 1} \quad (3.20)$$

onde T_D e K_D são os vetores da constante de tempo derivativa e o ganho derivativo, respectivamente. O princípio deste projeto é o algoritmo do tipo *PI* nebuloso, que tem a vantagem da implementação de controle derivativo na saída, evitando picos derivativos frente a mudanças de referência. O diagrama básico do *CLN-PI* + (*D* convencional) é mostrado na figura 3.11 (LI & GATLAND, 1995).

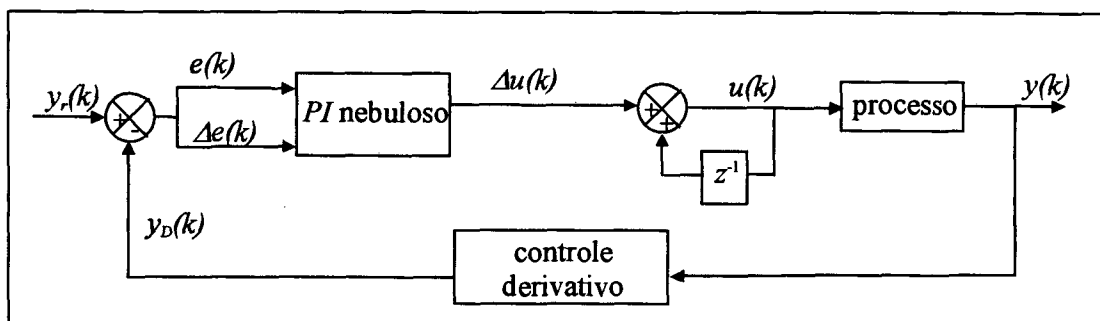


Figura 3.11. Diagrama básico do *CLN-PI* nebuloso + (*D* convencional).

O *CLN-PI* nebuloso + (*D* convencional) apresenta uma vantagem em relação ao projeto do *PID* convencional, pois um controlador *PID* convencional tem um comportamento aceitável quando atua em torno de um determinado ponto de operação, mas, freqüentemente, apresenta um desempenho pobre em regime transiente quando mudanças entre pontos diferentes de operação ocorrem devido a mudanças na dinâmica do processo (comportamento regulatório deficiente) e o controlador não é sintonizado adequadamente.

Para obter-se um desempenho adequado no projeto *CLN-PI* nebuloso + (*D* convencional), principalmente quanto aos aspectos relacionados a diminuição do *overshoot* e do tempo de subida, deve-se considerar que: (i) para diminuir o *overshoot* deve-se aumentar o fator de escala da ação de controle *I* (do controle *PI*) quando a saída está próxima da referência desejada; (ii) para diminuir o tempo de subida, a ação integral deve ser diminuída (menor fator de escala) durante o regime transiente; (iii) para diminuir o tempo de subida, o ganho do controle *PI* deve ser aumentado durante o transiente e diminuído quando a saída está próxima da referência.

3.9.3 Projeto $CLN-(PD+I)$ incremental

O projeto de controle $CLN-(PD+I)$ incremental foi proposto por MALKI *et al.* (1997). O $CLN-(PD+I)$ incremental é regido pela lei de controle:

$$u_{PID}(k) = u_{PD}(k) + u_I(k) \quad (3.21)$$

onde

$$u_{PD}(k) = -u_{PD}(k-1) + Ku_{PD} \Delta u_{PD}(k) \quad (3.22)$$

$$u_I(k) = u_I(k-1) + Ku_I \Delta u_I(k) \quad (3.23)$$

onde Ku_{PD} e Ku_I são as matrizes dos ganhos dos controles incrementais proporcional mais derivativo, e integral, respectivamente. O diagrama básico do $CLN-(PD+I)$ incremental é apresentado na figura 3.12.

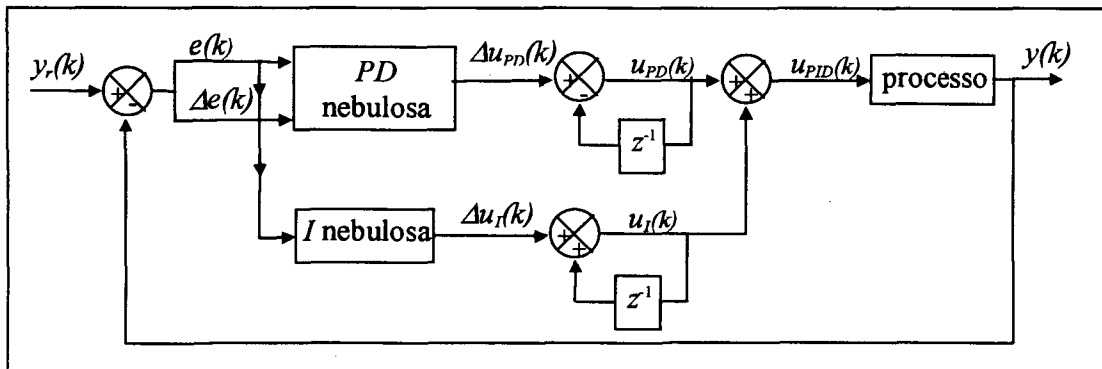


Figura 3.12. Diagrama básico do $CLN-(PD+I)$ incremental.

O projeto $CLN-(PD+I)$ incremental visa a configuração de um controlador mais robusto que o $CLN-(PD+I)$ proposto na seção 3.9.1. A vantagem da configuração incremental é maior flexibilidade no projeto para evitar-se erro em regime permanente que o $CLN-(PD+I)$ posicional.

3.9.4 Projeto $CLN-(PD+PI)$

O projeto $CLN-(PD+PI)$ consiste de um $CLN-PD$ em paralelo com um $CLN-PI$, sendo esta configuração apresentada por KWOK (1990) e LI & GATLAND (1995), onde:

$$u_{PID}(k) = u_{PD}(k) + u_{PI}(k) \quad (3.24)$$

O projeto *CLN-(PD+PI)* apresenta uma base de regras, tanto para o controle *PD* nebuloso quanto para o *CLN-PI*, e o projeto utiliza a composição das ações de *CLN-PD* e *CLN-PI*. O diagrama básico do *CLN-(PD+PI)* é mostrado na figura 3.13.

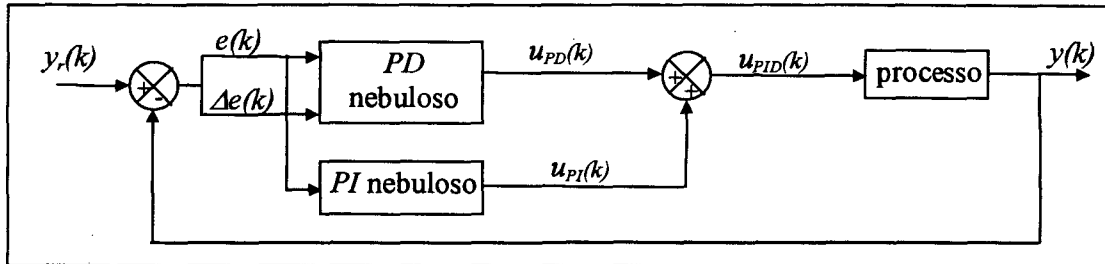


Figura 3.13. Diagrama básico do *CLN-(PD+PI)*

No projeto *CLN-(PD+PI)* existe um conflito de interesses entre os requerimentos de resposta rápida e de sobre-elevação mínima. Deve ser enfatizado, que na maioria dos processos, um aumento no ganho proporcional do controle *PI* nebuloso leva ao aumento na velocidade de resposta e redução da estabilidade, enquanto um aumento na ação integral do controle *PI* nebuloso reduz a velocidade de resposta, mas aumenta a estabilidade.

A desvantagem do projeto *CLN-(PD+PI)* é a necessidade da ação de controle *PD* nebuloso dever reduzir a sobre-elevação máxima, além da existência de pequenas oscilações que, usualmente, ocorrem em regime permanente. Entretanto, se o controle *PD* não é projetado pode-se ter como consequência uma excessiva variância na ação de controle.

3.10 Conclusão do capítulo

Neste capítulo foram introduzidos os *SNs* através da descrição de aspectos teóricos dos conjuntos nebulosos, aplicados a identificação e controle de processos. Os *SNs* não substituem nem mesmo competem com as técnicas de controle convencionais. Os *SNs*,

antes disto, estendem o caminho na obtenção de técnicas de controle utilizadas em aplicações práticas, pela adição de capacidades de controle supervisorio.

Os *SNs* são metodologias eficientes, que proporcionam uma estrutura robusta na representação de conhecimento em ambientes com incertezas e de processos complexos. A construção de *SNs* é direcionada à utilização de conectivos lógicos nebulosos e a implementação de bases de regras, que podem ser empregadas para aproximação de um sistema de entrada(s) e saída(s). Os tópicos abordados do projeto de *SNs*, em identificação e controle de processos multivariáveis são relativos:

- (i) a um breve histórico da metodologia;
- (ii) aos conceitos básicos e características relevantes;
- (iii) as pesquisas acadêmicas e aplicações industriais;
- (iv) as vantagens e as desvantagens;
- (v) a identificação nebulosa;
- (vi) a configuração de *CLNs* do tipo *PID*.

Capítulo 4

REDES NEURAIS ARTIFICIAIS EM IDENTIFICAÇÃO E CONTROLE DE PROCESSOS

4.1 Introdução

A base da teoria clássica de controle foi desenvolvida assumindo a linearidade dos processos a serem analisados e controlados. A maioria dos processos é não linear, e muitos deles são representados de forma linear, sem perda significativa da precisão. Conseqüentemente, os projetos em controle de processos com técnicas de linearização são, usualmente, adotados na indústria (GUPTA & RAO, 1994, RONCO & GAWTHROP, 1997).

Nas últimas décadas, os engenheiros de controle têm realizado esforços para explorar outras direções da teoria de controle e suas aplicações. O incremento da complexidade dos sistemas industriais é caracterizado principalmente por processos variantes no tempo, dinâmicas não lineares complexas (acentuada não-linearidade, variação rápida de parâmetros, perturbações externas) e presença de incertezas nos parâmetros. Estas características motivam o desenvolvimento de tecnologia avançada em controle de processos, destacando-se algoritmos de controle adaptativo e de controle inteligente. Alguns destes aspectos são detalhados e discutidos em ÅSTRÖM (1991), LU (1992) e COELHO & COELHO (1997d).

A área denominada controle neural é um subconjunto das áreas de *RNs* e as técnicas de controle. A essência das *RNs* pode ser vista, para um sistema de controle, como um mapeamento entre o estado do processo e os comandos de atuação. O aprendizado através de *RNs* é realizado através da modificação deste mapeamento aprimorando o desempenho do sistema de controle.

Com poucas exceções, o propósito de uma *RN* em um controlador neural é executar uma tarefa de aproximação de funções. Nas *RNs* treinadas por métodos de

gradiente descendente, esta tarefa é realizada através da associação dos sinais de erro da saída com os sinais de entrada. O objetivo é obter um erro aceitável, e os sinais de entrada são selecionados com vistas à realização deste objetivo. A função que uma *RN* executa é, entretanto, dependente da forma que o sinal de erro é calculado. Este capítulo descreve os aspectos relacionados as *RNs* em identificação e controle de processos.

4.2 Breve histórico

A idéia da utilização das *RNs* como uma ferramenta potencial à resolução de problemas não é nova, e tem origem, em 3000 a.C., com os trabalhos de Hipócrates. O pesquisador William James, *Psychology (Briefer Course)*, em 1890, tratou os tópicos relativos à atividade do cérebro, mas foi Alan Turing o primeiro pesquisador a inspirar-se no cérebro para configurar um paradigma computacional, em 1936 (NELSON & ILLINGWORTH, 1991).

Os fisiologistas têm desenvolvido modelos de aprendizado humano devido aos progressos da neuroanatomia e neurofisiologia. Um modelo, que possui aplicações no aprendizado das *RNs*, foi proposto por Donald Hebb, no trabalho *The Organization of Behavior*, em 1949. Neste modelo propõe-se uma regra de aprendizado, sendo este o marco inicial de algoritmos de treinamento das *RNs*.

Um dos trabalhos pioneiros na área foi o de Warren S. McCulloch e Walter Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, em 1943, quando propõem um modelo para o neurônio, demonstrando que as associações de neurônios artificiais implementam qualquer função lógica finita. Este foi o primeiro sucesso teórico do conexionismo. O neurônio tinha um número finito de entradas e uma saída. As entradas eram caracterizadas pelos estados excitatório (+1) ou inibitório (-1).

A primeira onda de entusiasmo com as *RNs* surgiu com o *perceptron* de Frank Rosenblatt, publicado no estudo *The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain*, em 1958. O *perceptron* consegue aprender a classificação de padrões a partir de exemplos. Assim, pela primeira vez, tinha-se a configuração de um modelo de aprendizado e percepção com resultados concretos.

Nas décadas de 50 e 60 foram publicados trabalhos relevantes quanto a utilização de *RNs* em computadores. Nathaniel Rochester e outros pesquisadores dos laboratórios da *IBM* implementam um *software* de simulação de *RNs*, baseados no trabalho de Donald Hebb. John Von Neumann, com *The Computer and the Brain*, em 1958, sugere a imitação de funções de neurônios simples para utilização em relés de telégrafos e tubos à vácuo.

Em 1959, Bernard Widrow e Marcian Hoff desenvolvem os modelos de *RNs* denominados *ADALINE* (*ADaptive LINear Elements*) e *MADALINE* (*Multiple ADALINE*). Esta foi a primeira aplicação de *RNs*, em problemas do mundo real, constituindo-se de filtros adaptativos para eliminação de ruídos em linhas telefônicas.

Ao entusiasmo seguiu-se uma grande crise com a descoberta das limitações do *perceptron*. No livro *Perceptrons*, MINSKY & PAPERT (1969) mostram as deficiências do *perceptron* e provam que as *RNs* de uma camada, então em utilização, são incapazes de resolver muitos problemas simples, incluindo a execução de uma função *ou exclusivo*. Em síntese, o *perceptron* de uma única camada é incapaz de resolver problemas linearmente não-separáveis. Neste trabalho onde todas estas dificuldades são apontadas, acompanhadas da declaração de uma crença pessoal (e errônea) dos autores, de que a extensão do modelo neural seria inútil, marca o ocaso das pesquisas em *RNs* na década de 70.

Entretanto, deve ser mencionado, que MINSKY & PAPERT (1969) apontam a solução do *ou exclusivo*, através de alguma transformação pela adição de uma camada ao *perceptron* conectando todas as entradas.

As conseqüências desta publicação foram o desencorajamento de pesquisadores e o redirecionamento dos fundos de agências governamentais. Alguns poucos cientistas, entre os quais, Teuvo Kohonen, Stephen Grossberg e James Anderson continuaram suas pesquisas. Na década de 70 e início dos 80, alguns trabalhos isolados foram publicados em jornais. Nos anos 80, gradualmente os fundamentos teóricos emergem quanto as *RNs* com múltiplas camadas.

O ressurgimento do interesse em *RNs* ocorre com a utilização do algoritmo de treinamento por retropropagação de erro, por um grupo de pesquisadores denominado *PDP* (*Parallel Distributed Processing*), em 1986. Suas pesquisas resultam na extensão do *perceptron* para várias camadas de neurônios, e assim, superam as dificuldades daquele modelo (RUMELHART *et al.*, 1986). A partir deste marco, observa-se uma explosão de

aplicações de *RNs*, nos mais variados campos do conhecimento. Os detalhes sobre as origens das *RNs* e sua evolução cronológica são encontrados em WASSERMAN (1993), HAYKIN (1994) e GUPTA & RAO (1994).

4.3 Por que utilizar as *RNs*?

As características que tornam as *RNs* atraentes são (NELSON & ILLINGWORTH, 1991, WASSERMAN, 1993, HAYKIN, 1994):

- (i) *habilidade de tratar sistemas não lineares*: relevante nas aplicações em identificação de sistemas dinâmicos e classificação de padrões;
- (ii) *tolerância a falhas*: o conhecimento é distribuído pela *RN*, mais que em uma simples localização de memória. Uma parte das conexões pode estar inoperante, sem mudanças significativas no desempenho de toda a *RN*;
- (iii) *adaptabilidade*: capacidade da *RN* em auto-ajustar-se. Os aspectos de aprendizado, auto-organização, generalização e treinamento estão intrinsecamente ligados a esta característica;
- (iv) *aprendizado*: uma *RN* pode modificar seu comportamento em resposta ao ambiente. Quando é apresentado um conjunto de entradas, as *RNs* se ajustam para gerar as respostas apropriadas;
- (v) *generalização*: consiste na *RN* mapear entradas similares em saídas similares;
- (vi) *treinamento*: é a forma pela qual a *RN* aprende;
- (vii) *processamento paralelo*: as *RNs* são estruturalmente paralelas. A seqüência de processamento das *RNs* é realizada em paralelo e simultaneamente;
- (viii) *abstração*: muitas *RNs* são capazes de abstrair a essência de um conjunto de entradas;

4.4 Aplicações industriais

As áreas de aplicações das *RNs* são as mais diversas, tais como: biologia, finanças, meio ambiente, inteligência artificial, identificação, controle, mercado financeiro, ciência da computação, telecomunicações, manufatura, medicina, e outras (WIDROW & LEHR, 1990, WIDROW *et al.*, 1994).

As aplicações usuais são em: reconhecimento e classificação de padrões, *clustering*, previsão de séries temporais (com aplicações em economia e finanças), aproximação de funções, predição, otimização, processamento de sinais para identificação de alvos e análise de imagens, sistemas especialistas, processamento de sinais (imagens, sensores, voz, caracteres, visão, compressão de dados, filtragem de sinais), telecomunicações, manufatura, monitoramento de processos e robótica (HAYKIN, 1994, WIDROW *et al.*, 1994, STEMMER *et al.*, 1997). Nas tabelas A.5 e A.6 (apêndice A) são apresentadas algumas aplicações e ferramentas de projetos com *RNs*.

Os projetos, empresas e uma revisão, do estado da arte, quanto à implementação de *RNs* em *hardware* na forma digital, analógica e híbrida (neuroprocessadores) não são apresentados na tabela A.6. Os detalhes desta área de aplicação podem ser encontrados em LINDSEY & LINDBLAD (1994) e DUONG *et al.* (1996).

Quanto aos propósitos de identificação e controle de processos industriais, podem-se destacar alguns produtos. O *NeuCOP*, da *NeuralWare*, é uma ferramenta para os projetos industriais de algoritmos de identificação e controle de processos não lineares (VAN DOREN, 1995). O sistema *NeurOn-Line*, da *Gensym*, é um conjunto de ferramentas para a construção e a aplicação de *RNs*. Este *software* permite gerar aplicações de controle em tempo real.

Uma aplicação em controle, com sucesso comercial, é o *Intelligent Arc Furnace*, desenvolvido pela *Neural Applications Corporation* e *Milltech-HOH*, que constitui em utilizar as *RNs* para regular a posição de eletrodos em fornos elétricos de arco (SAMAD, 1995).

4.5 Modelo de neurônio

Uma *RN* constitui-se de modelos de processamento paralelo e distribuído. A unidade básica de uma *RN* é o neurônio. Os neurônios são capazes de modificar o seu comportamento após a realização de um treinamento dinâmico. Os neurônios são interconectados por conexões (sinapses) com valores variáveis denominados pesos.

Em essência, um conjunto de entradas (vetor X) é aplicado a um neurônio artificial, cada uma representando uma saída de outro neurônio. Cada entrada é

multiplicada por um peso correspondente (vetor W), analogamente, ao poder da sinapse. Os pesos das entradas são somados para determinar o nível de ativação do neurônio, este adicionado a um valor limite, θ_p , conhecido por polarização ou *bias*, valor abaixo do qual, a saída é nula. Em outras palavras, um neurônio corresponde a uma soma ponderada de entradas, soma esta aplicada a uma função de ativação não linear (ou mesmo linear), contínua e diferenciável, conforme apresentado na figura 4.1.

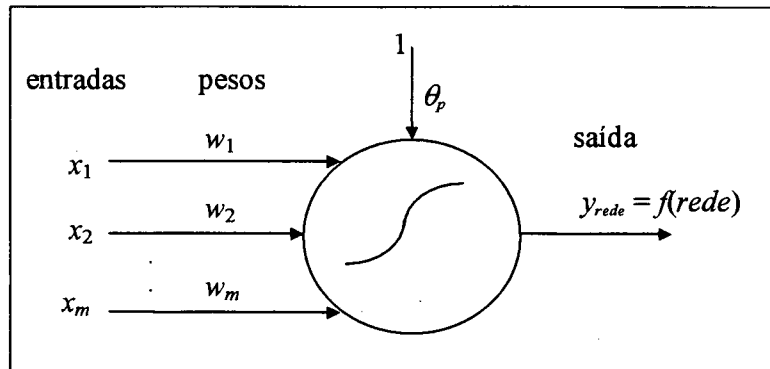


Figura 4.1. Estrutura do neurônio artificial.

A estrutura do neurônio apresentada na figura 4.1 pode ser expressa na seguinte forma:

$$s = f\left(\sum_{i=1}^n w_i x_i + \theta_p\right) \tag{4.1}$$

onde f é a função de ativação, x_i é o conjunto de entradas e w_i são os pesos do neurônio. Algumas das funções de ativação utilizadas são:

(i) linear: $idn(x) = x$ (4.2)

(ii) saturação: $sat(x) = \begin{cases} +1, & \text{se } x > 1 \\ x, & \text{se } x \in [-1, 1] \\ -1, & \text{se } x < -1 \end{cases}$ (4.3)

(iii) sinal: $sign(x) = \begin{cases} +1, & \text{se } x > 1 \\ -1, & \text{se } x < -1 \end{cases}$ (4.4)

(iv) sigmóide: $sig(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$ (4.5)

$$(v) \text{ tangente hiperbólica: } \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.6)$$

O modelo de neurônio, apresentado aqui, consiste no modelo baseado nos estudos de McCulloch e Pitts. Entretanto, existem outros modelos, por exemplo, os modelos de neurônios booleanos (REDGERS & ALEKSANDER, 1995).

Outra forma de apresentar-se os modelos formais de neurônios e as *RNs* é através de conceitos da teoria de sistemas e de grafos, propostos por BARRETO (1996), DE AZEVEDO (1997), OLIVEIRA (1998) e PODOLAK (1998). As conexões sinápticas no modelo são consideradas como externas ao modelo do neurônio, tal como ocorre no sistema nervoso biológico e não fazendo parte do neurônio, conforme formalizado por alguns autores. Esta observação proporciona a possibilidade de interpretar a matriz de conexões, denominada matriz de conectividade, como a matriz de pesos de um grafo, o grafo representativo de uma *RN* (BARRETO, 1997).

4.6 Classificação das *RNs*

Uma definição de *RN* como “um sistema interconectado de funções parametrizadas” cobre muitos tipos de *RNs* e modelos de neurônios. A classificação das *RNs* é importante no contexto de comparar estruturas e melhor diferenciá-las. Um esquema de classificação simples pode ser realizado com algumas informações que descrevem uma *RN*, tais como (REDGERS & ALEKSANDER, 1995):

- (i) *topologia*: como as funções são interconectadas;
- (ii) *arquitetura*: o tipo e a utilização da *RN*;
- (iii) *modelo do neurônio*: o que as funções são;
- (iv) *algoritmo de treinamento*: como os parâmetros são configurados;
- (v) *escalonamento da operação*: a sincronização das interações entre as funções.

A topologia de uma *RN* é um conjunto das conexões entre as entradas, as saídas e os nodos (nós). Todas topologias são decorrentes de subconjuntos da topologia totalmente conectada. As *RNs* podem ser classificadas, quanto à conectividade interna, em *feedforward* e *feedback* (recorrentes), conforme apresentado na figura 4.2. Na literatura, as diversas estruturas de *RNs* são mencionadas em aplicação na área de identificação e

controle de processos, mas as *RNs perceptron* multicamadas, Hopfield, função de base (radial, *wavelet* e polinomial) e híbridas com técnicas de controle adaptativo têm polarizado a maioria das aplicações.

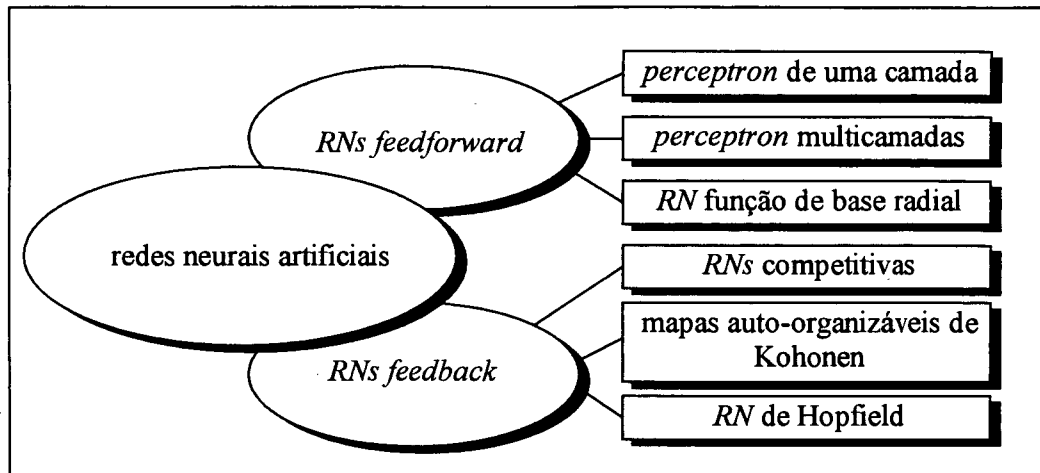


Figura 4.2. Taxionomia das *RNs feedforward* e *feedback*.

Quanto aos diferentes tipos de conexões, as *RNs* utilizam conexões entre as camadas e entre as conexões na mesma camada. As *RNs* podem ser classificadas sob outro ponto de vista, pela diferenciação entre as que generalizam globalmente e localmente. A generalização da *RN* é global se um (ou mais) dos seus parâmetros adaptativos (os pesos) podem afetar a saída da *RN*, para um (ou todo) ponto no espaço de entrada. O *perceptron* multicamadas é um exemplo de *RN* que generaliza globalmente. A generalização local ocorre em *RNs* quando somente poucos pesos afetam a resposta da saída da *RN*, para os “pontos locais” no espaço de entrada. As *RNs* função de base radial e *B-Splines* (HARRIS *et al.*, 1996) são exemplos de *RNs*, com generalização local, em que a interferência no aprendizado é minimizada e o aprendizado é relativamente rápido, devido ao número mínimo de pesos ajustados a cada par de treinamento apresentado à *RN* (HARRIS *et al.*, 1993). Outra classificação das *RNs* é quanto à forma de aprendizado. Neste caso, as *RNs* são classificadas por apresentar:

- (i) *aprendizado supervisionado* (ou aprendizado associativo): A *RN* é treinada para prover às entradas o casamento de padrões de saída desejados. Estes pares de entrada(s) e saída(s) podem ser gerados por um “professor externo” (usuário) ou por um sistema que contenha uma *RN* (*self-supervised*).

- (ii) *aprendizado não-supervisionado* (ou aprendizado com auto-organização): Neste caso, a *RN* é treinada para responder a *clusters* de padrões relativos a entrada. Nesta forma de aprendizado, não existe um conjunto de categorias *a priori* em que os padrões são classificados, sendo que a *RN* deve desenvolver sua própria representação baseada em estímulos das entradas.
- (iii) *aprendizado híbrido*: combina a aprendizagem supervisionada e a não-supervisionada. Uma *RN* função de base radial admite esta forma de aprendizado, onde a primeira camada de conexões da *RN* é treinada de forma não-supervisionada e a segunda de forma supervisionada.

4.7 *RN perceptron feedforward* multicamadas

A *RN perceptron feedforward* multicamadas (*RN-MLP*) é aplicada para resolver os mais diversos e complexos problemas, devido a suas capacidades. Tratam-se de “aproximadores universais” de uma função com uma dada precisão (CYBENKO, 1989, RAO & GUPTA, 1993), proporcionando uma ferramenta promissora à identificação, e ao controle inteligente e adaptativo. A *RN-MLP* é uma generalização do *perceptron* de uma camada (MINSKY & PAPERT, 1969). A *RN-MLP* contém três tipos de camadas: a camada de entrada, camada(s) oculta(s) e a camada de saída. Qualquer neurônio de uma camada pode interligar-se com outro neurônio da camada seguinte. Os neurônios da camada de entrada recebem os sinais do mundo externo e os transmitem para os neurônios da próxima camada. Enquanto isso, os neurônios da camada de saída enviam a informação dos neurônios da(s) camada(s) oculta(s) para o mundo externo.

O treinamento da *RN-MLP* é realizado de maneira supervisionada com o algoritmo *back-propagation* (*BP*) ou retropropagação do erro. O termo *BP* apareceu por volta de 1985, com os trabalhos de D. Parker, Y. Le Cun, D. E. Rumelhart, G. E. Hinton, R. J. Williams, contudo a idéia básica do *BP* foi apresentada primeiro por P. J. Werbos, na sua tese de doutorado, em 1974 (HAYKIN, 1994; RUMELHART *et al.*, 1986).

4.8 Algoritmo de treinamento *BP*

O algoritmo *BP* é baseado na regra de aprendizado da correção do erro, que pode ser vista como uma generalização do algoritmo da filtragem adaptativa ou mesmo um

caso especial do algoritmo dos mínimos quadrados. O algoritmo *BP* é um método iterativo de gradiente projetado para minimizar a soma do erro quadrático, entre a saída atual e a saída desejada. Para minimizar esta função objetivo, o algoritmo *BP* utiliza uma técnica de busca baseado em gradiente, a regra delta generalizada.

É relevante mencionar que o algoritmo *BP* não tem, aparentemente, nenhuma relação com sistemas de aprendizado biológico. Não parece haver nenhum indício de retropropagação do erro no sistema nervoso animal. Entretanto, se um algoritmo de aprendizado revela resultados promissores, a inexistência de um paralelo natural (plausibilidade biológica) conhecido, não costuma ser levada em consideração em aplicações de controle e em outras aplicações industriais de engenharia.

O *BP* consiste basicamente de duas fases através das diferentes camadas da *RN*, que são: a fase *forward* e a fase *backward*. Na fase *forward*, um padrão de atividade (vetor entrada) é aplicado a *RN* e o seu efeito é propagado, camada por camada. Finalizando, um conjunto de saídas produz a resposta atual da *RN*. Durante a fase *forward*, os pesos da *RN* são fixos (HAYKIN, 1994).

Durante a fase *backward*, os pesos são ajustados de acordo com uma regra de correção do erro. Especificamente, a saída atual da *RN* é subtraída da saída desejada para o cálculo do erro. O erro é propagado para trás (*backward*) em direção às entradas, através da *RN*, atualizando os pesos da *RN*. Os pesos são ajustados de forma que a saída atual da *RN* aproxime-se da saída desejada. As etapas que regem o *BP* são sintetizadas por:

- (i) inicializar aleatoriamente os pesos das conexões da *RN*;
- (ii) aplicar o conjunto de treinamento constituído das entradas e das saídas desejadas (dados do processo) à *RN*;
- (iii) calcular a(s) saída(s) da *RN*, propagando as saídas dos neurônios de cada camada da *RN* para os neurônios da próxima camada, passando pela função de ativação dos neurônios (passo *forward*);
- (iv) especificar a saída desejada e calcular os erros das camadas;
- (v) ajustar os pesos pelos “gradientes locais” da *RN* (passo *backward*) utilizando o algoritmo recursivo, começando das unidades de saída e propagando-se em direção à primeira camada oculta, utilizando as equações:

$$w_{j,i}(k+1) = w_{j,i}(k) + \Delta w_{j,i}(k) \quad (4.7)$$

$$\Delta_p w_{j,i} = \eta \delta_{p,j} o_{p,i} \quad (4.8)$$

onde $w_{j,i}(k)$ é o peso, η é ganho que representa o coeficiente de aprendizado, $\delta_{p,j}$ é o termo de erro da unidade j ; e $\Delta_p w_{j,i}$ é a mudança realizada no peso da i -ésima para a j -ésima unidade seguindo a apresentação do padrão p ;

(vi) repetir os passos (iv) e (v), enquanto a função do erro da saída não apresentar um valor de tolerância aceitável (critério de parada do treinamento);

(vii) utilizar conjunto de teste para analisar o desempenho do treinamento.

Após a conclusão do treinamento, a RN atua como uma RN *feedforward*, pois os pesos das conexões permanecem constantes, exceto no caso da necessidade de um novo treinamento com um novo conjunto de dados. A figura 4.3 ilustra um exemplo de uma RN-MLP com treinamento BP. Os detalhes da formulação matemática da regra delta generalizada e do algoritmo BP podem ser encontrados em OLIVEIRA (1998).

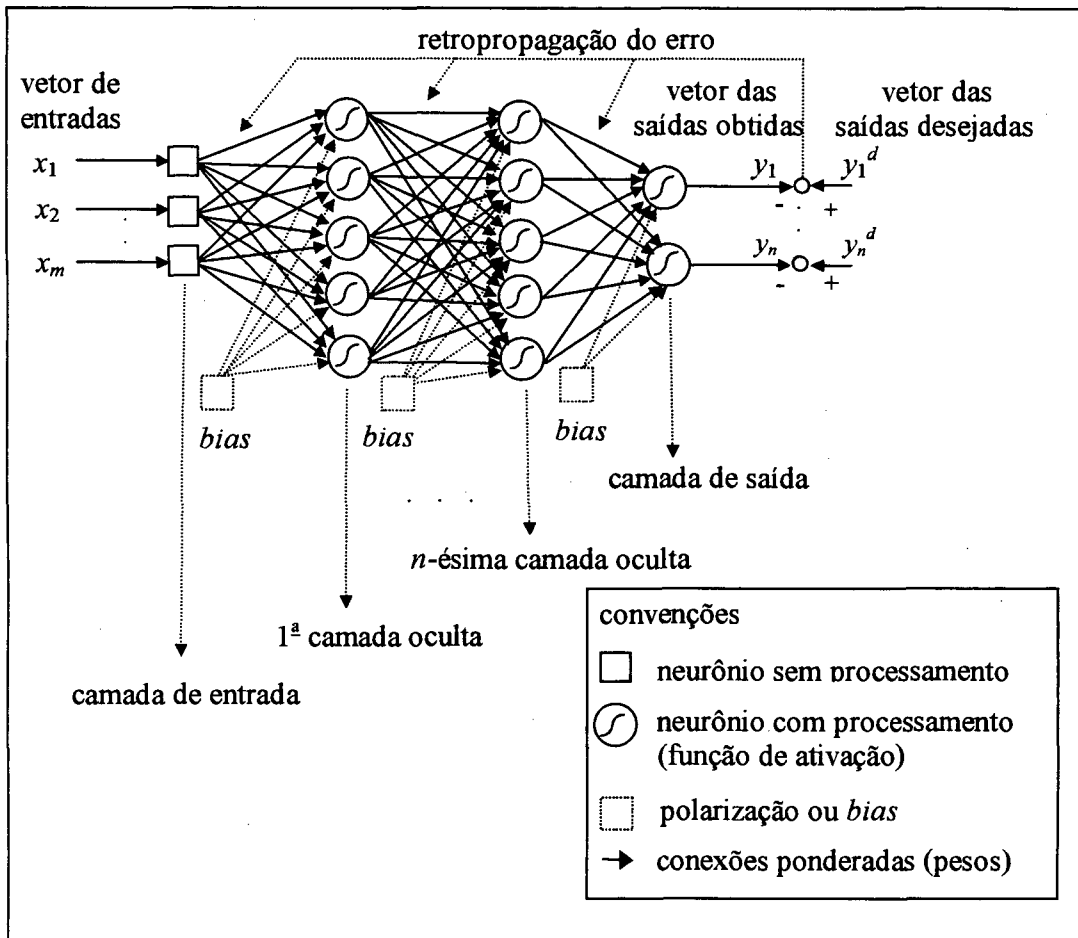


Figura 4.3. RN-MLP com treinamento BP.

O *BP* é uma das mais populares técnicas de aprendizado supervisionado de *RNs*. É uma metodologia simples de implementar e sua capacidade de generalização é atraente para um grande espectro de aplicações em problemas de reconhecimento e classificação de padrões. Apesar do sucesso aparente do *BP*, existem alguns aspectos que fazem com que este algoritmo apresente algumas deficiências, tais como: (i) justificar suas respostas é praticamente impossível, (ii) custo computacional significativo, e (iii) baixa velocidade de aprendizado.

O aumento da velocidade de aprendizado, através de procedimentos heurísticos, pela modificação adaptativa dos parâmetros e prescrição de uma seleção específica dos pesos iniciais de uma *RN-MLP* são alternativas viáveis. A seguir são descritas algumas metodologias para aceleração do *BP*.

4.8.1 Método adaptativo de VOGL

O método de VOGL *et al.* (1988) tem atualização dos pesos, $\Delta w_{i,j}$, regida por

$$\Delta w_{i,j}(k) = -\eta(k) \sum_{p=1}^N \frac{\partial E_p}{\partial w_{i,j}} + \alpha(k) \Delta w_{i,j}(k-1) \quad (4.9)$$

O pseudocódigo deste método é explicitado na figura 4.4.

<p>(i) inicializar $\eta(0) = \eta_0$ e $\alpha(0) = \alpha_0$; (ii) se $E(k) < E(k-1)$ então $\eta(k) = \phi \cdot \eta(k-1)$ e $\alpha(k) = \alpha_0$; se $E(k) < (1+\varepsilon)E(k-1)$ então $\eta(k) = \phi \cdot \eta(k-1)$ e $\alpha(k) = \alpha_0$; se $E(k) > E(k-1)$ então descartar o último passo de aprendizado e $\eta(k) = \beta \cdot \eta(k-1)$ e $\alpha(k) = 0$;</p>
--

Figura 4.4. Algoritmo *YPROP* heurístico para o treinamento de uma *RN-MLP*.

Na figura 4.4, o termo ε é um fator de ajuste de “sensibilidade”, η é o coeficiente de aprendizado, α é o fator de momento, E é a soma do erro médio quadrático da(s) saída(s) da *RN-MLP*, ϕ e β são fatores de aceleração e desaceleração do coeficiente de aprendizado, respectivamente.

4.8.2 Método adaptativo de Anguita-Vogl — algoritmo *YPROP*

A abordagem *YPROP* (ANGUITA *et al.*, 1993) é um aprimoramento do método proposto por VOGL *et al.* (1988). O *YPROP* emprega uma heurística adaptativa para a variação do coeficiente de aprendizado, através dos fatores de aceleração, ϕ_{YPROP} , e desaceleração, β_{YPROP} . Estes fatores são regidos por

$$\phi_{YPROP}(k) = 1 + \frac{K_a}{K_a + \eta(k-1)} \quad (4.10)$$

$$\beta_{YPROP}(k) = 1 + \frac{K_d}{K_d + \eta(k-1)} \quad (4.11)$$

onde K_a e K_d são constantes de aceleração e desaceleração, respectivamente.

4.8.3 Método *delta-bar-delta* de Jacobs

No *delta-bar-delta* de JACOBS (1988), cada peso da *RN-MLP* tem o seu próprio coeficiente de aprendizado, que é modificado visando evitar as oscilações no aprendizado. A modificação do coeficiente de aprendizado é realizada da seguinte forma:

$$\Delta\eta(k) = \begin{cases} \kappa, & \text{se } \bar{\delta}(k-1)\delta(k) > 0 \\ -\phi_{JAC} * \eta(k), & \text{se } \bar{\delta}(k-1)\delta(k) < 0 \\ 0, & \text{nos outros casos} \end{cases} \quad (4.12)$$

onde

$$\delta(k) = \frac{\partial E(k)}{\partial w(k)} \quad (4.13)$$

$$\bar{\delta}(k) = (1 - \theta_{JAC})\delta(k) + \theta_{JAC}\bar{\delta}(k-1) \quad (4.14)$$

A derivada parcial atual de um peso é comparada com uma média ponderada exponencial, $\bar{\delta}(k)$, das derivadas anteriores. Se o sinal é o mesmo em uma época (iteração) consecutiva, o coeficiente de aprendizado é incrementado de κ . Se o sinal é contrário (indício de oscilação), o valor do coeficiente de aprendizado é decrementado de uma proporção ϕ_{IAC} . A regra *delta-bar-delta* incrementa os coeficientes de aprendizado linearmente e decrementa-os exponencialmente.

4.8.4 Métodos de treinamento de segunda ordem

Existem casos em que a velocidade de aprendizado é um limitador às aplicações práticas da *RN-MLP* em problemas que requerem precisão nos resultados obtidos. O desempenho do *BP* é sensível à configuração do coeficiente de aprendizado e utiliza somente a primeira derivada (a curvatura) da superfície de erro. Com a utilização da segunda derivada (a taxa de mudança da curvatura), o tempo de treinamento pode ser reduzido significativamente. A função custo, E , é minimizada através de um procedimento iterativo pela geração de pontos de busca onde a matriz de pesos, $W(k)$, é atualizada por:

$$W(k+1) = W(k) + \eta(k)d(k) \quad (4.15)$$

onde uma matriz de pesos iniciais, $W(0)$, é determinada através do conhecimento *a priori* ou gerada aleatoriamente. O termo $d(k)$ e $\eta(k)$ indicam a direção de busca e o tamanho do passo, respectivamente. Através da equação (4.19), nota-se que quando $\eta(k)$ é muito pequeno, um pequeno aprendizado ocorre de forma de $W(k+1)$ apresenta uma mudança pouco significativa de $W(k)$. Em contrapartida, quando $\eta(k)$ tem valores grandes, para um dado $d(k)$, nota-se que os valores de $W(k+1)$ são consideravelmente diferentes de $W(k)$. Para o método gradiente descendente, a direção de busca é dada por:

$$d(k) = -\nabla E(W(k)) \quad (4.16)$$

onde ∇ indica a derivada (ou o gradiente) da função erro. Para o método de Newton, a direção é obtida por:

$$d(k) = -\left[\nabla^2 E(W(k))\right]^{-1} \nabla E(W(k)) \quad (4.17)$$

Contudo, a avaliação da inversa da matriz Hessiana $\nabla^2 E(W(k))$ pode ser complicada e envolver elevada complexidade computacional. Assim, uma aproximação da matriz, $H(k)$, é freqüentemente utilizada. Quando utilizada em um algoritmo de aprendizado tem-se um procedimento do tipo quase-Newton com:

$$d(k) = -H(k) \nabla E(W(k)) \quad (4.18)$$

As técnicas de quase-Newton estimam a inversa da matriz Hessiana da função custo iterativamente, e atualizam o vetor de pesos, assumindo que a função custo é aproximadamente localmente quadrática. Os métodos de quase-Newton são baseados em uma aproximação da curvatura de uma função não linear calculada sem a formação explícita da matriz Hessiana (SETIONO & HUI, 1995).

O método de Newton tem diversas desvantagens causadas por dificuldades numéricas. Por exemplo, se a Hessiana, H , da função custo não é definida positiva, o cálculo da direção pode não decrescer a função custo a ser minimizada. Neste caso, o passo não pode ser executado ao longo desta direção. O algoritmo *LM* (LEVENBERG, 1944, MARQUARDT, 1963) é uma aproximação do método de Newton e freqüentemente constitui-se de uma alternativa eficiente aos algoritmos do tipo *BP*.

O *LM* considera a curvatura da função para estabelecer o tamanho de passo ao longo da sua curvatura. A estratégia consiste na expansão da função $f(x, w) = t$ em uma série de Taylor. O propósito é minimizar a função custo, $J(w)$, com respeito ao peso w . O *LM* apresenta custo computacional elevado, em memória e no número de operações requeridas. Os detalhes do *LM* para o treinamento de *RN-MLP* são encontrados em KOLLIAS & ANASTASSIOU (1989) e HAGAN & MEHNAJ (1994).

Outra técnica, o método gradiente conjugado (*GC*), emprega uma forma recursiva da direção de busca dada pela equação:

$$d(k) = \alpha(k) d(k-1) - \nabla E(W(k)) \quad (4.19)$$

O *GC* não segue a direção do gradiente, mas, em vez disso, a direção que é “conjugada” à direção do passo anterior. O gradiente permanece, no passo atual, perpendicular a direção do passo anterior. Entretanto, a mudança no gradiente deve ser perpendicular a direção do passo anterior (WASSERMAN, 1993).

O algoritmo do *GC* converge adequadamente em funções em que a matriz Hessiana é positiva definida. Entretanto, esta condição não é sempre encontrada, especialmente em regiões afastadas do erro mínimo. Um método desenvolvido por MØLLER (1994) resolve este problema, determinando quando a Hessiana não é definida positiva e ajustando um parâmetro escalar para isto. Esta abordagem, denominada *GC* escalar (*SGC*), evita o tempo computacional, no procedimento de busca unidimensional, pela utilização do algoritmo de *LM* para determinar o tamanho de passo. Os detalhes e o pseudocódigo do *SGC* para o treinamento de *RN-MLP* são encontrados em MØLLER (1994). A seguir são descritos dois métodos baseados em informações de segunda ordem, o *RPROP* e o *Quickprop*.

4.8.4.1 Método adaptativo direto — algoritmo *RPROP*

A abordagem *RPROP* (*resilient propagation*) é realizada por RIEDMILLER & BRAUN (1993) e executa uma adaptação local da atualização de pesos de acordo com o comportamento da função de erro, equação (4.9). O *RPROP* utiliza o sinal da derivada para determinar a direção de atualização dos parâmetros de configuração da *RN-MLP*.

O *RPROP* visa eliminar a influência negativa do valor da derivada parcial na definição do ajuste dos pesos. Esta influência negativa ocorre quando a saída de um neurônio for próxima de 0 (ou 1) e a saída desejada for 1 (ou 0), a derivada é próxima de 0, fazendo com que os pesos recebam um ajuste mínimo, próximo de 0. O *RPROP* elimina este problema utilizando apenas o sinal da derivada, e não o seu valor (BRAGA *et al.*, 1998).

Este esquema de aprendizado introduz para cada peso um valor individual de atualização de seu valor, ou seja, determina o tamanho de atualização do peso, de acordo com a regra de aprendizado:

$$\Delta_{j,i} = \begin{cases} \eta^+ * \Delta_{j,i}(k-1), & \text{se } \left(\frac{\partial \mathcal{E}(k-1)}{\partial w_{j,i}(k-1)} \right) * \left(\frac{\partial \mathcal{E}(k)}{\partial w_{j,i}(k)} \right) > 0 \\ \eta^- * \Delta_{j,i}(k-1), & \text{se } \left(\frac{\partial \mathcal{E}(k-1)}{\partial w_{j,i}(k-1)} \right) * \left(\frac{\partial \mathcal{E}(k)}{\partial w_{j,i}(k)} \right) < 0 \\ \Delta_{j,i}, & \text{nos outros casos} \end{cases} \quad (4.20)$$

A cada iteração a derivada parcial da mudança de pesos, $w_{j,i}$, correspondente, muda o sinal, indicando que a última atualização é grande e o algoritmo passa do mínimo local. Assim, o valor de $\Delta_{j,i}$, é decrementado de um fator, η^- . Se a derivada mantém o sinal, o valor de $\Delta_{j,i}$ é incrementado de η^+ , visando acelerar a convergência (RIEDMILLER, 1994). Os valores dos parâmetros de incremento, η^+ , e do decremento, η^- , são parâmetros de projeto. Uma vez que o valor de atualização de cada peso é adaptado, a atualização dos pesos segue a regra: se a derivada é positiva (incremento do erro) o peso é decrementado pelo valor de atualização, equação (4.20), se a derivada é negativa o valor de atualização é expresso por (RIEDMILLER & BRAUN, 1993):

$$\Delta w_{j,i} = \begin{cases} -\Delta_{j,i}(k), & \text{se } \left(\frac{\partial \mathcal{E}(k)}{\partial w_{j,i}(k)} \right) > 0 \\ +\Delta_{j,i}(k), & \text{se } \left(\frac{\partial \mathcal{E}(k)}{\partial w_{j,i}(k)} \right) < 0 \\ 0, & \text{nos outros casos} \end{cases} \quad (4.21)$$

Contudo, existe uma exceção, se a derivada parcial muda de sinal, isto é, se o passo anterior é grande e o mínimo é perdido, a atualização dos pesos é restabelecida, ou seja:

$$\Delta w_{j,i}(k) = -\Delta w_{j,i}(k-1), \text{ se } \left(\frac{\partial \mathcal{E}(k-1)}{\partial w_{j,i}(k-1)} \right) * \left(\frac{\partial \mathcal{E}(k)}{\partial w_{j,i}(k)} \right) < 0 \quad (4.22)$$

O procedimento *RPROP* é resumido pelo pseudocódigo da figura 4.5 (RIEDMILLER, 1994).

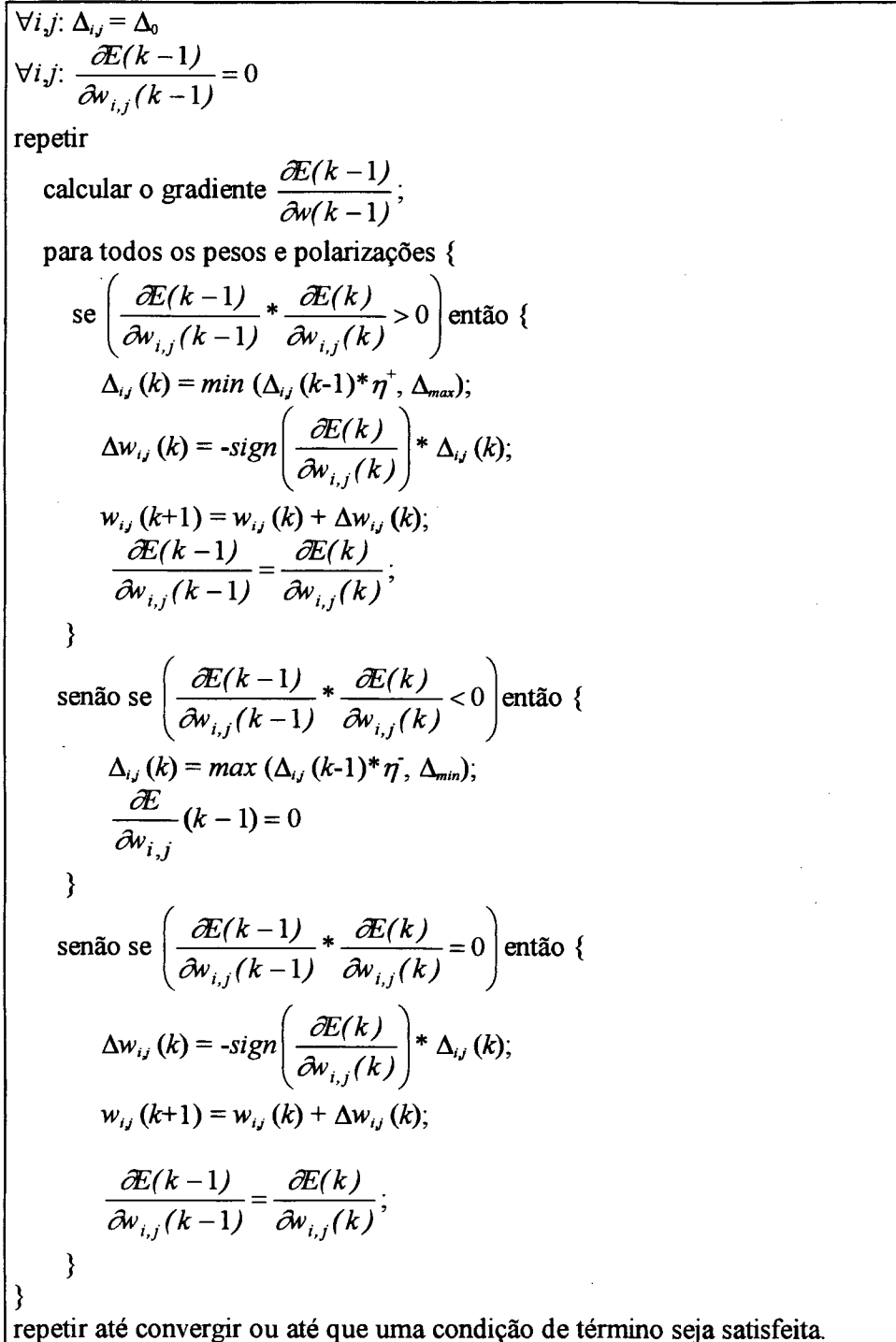


Figura 4.5. Algoritmo *RPROP* para o treinamento de uma *RN-MLP*.

4.8.4.2 Método adaptativo de Fahlman — algoritmo *QuickProp*

O treinamento de uma *RN-MLP* através *BP* emprega a primeira derivada parcial do erro referente a cada peso para ajustá-lo. O treinamento é acelerado pela utilização da informação da curvatura da superfície de erro (BRAGA *et al.*, 1998).

A abordagem *QuickProp* (*quick propagation*), proposta por FAHLMAN (1988), é um método que utiliza uma heurística baseada no método de Newton. O *Quickprop* assume que a superfície do erro é localmente quadrática (formando uma parábola) e procura “saltar” da posição atual na superfície para o ponto de mínimo da parábola. A atualização dos pesos é realizada pela equação:

$$\Delta w_{j,i}(k) = \frac{\frac{\partial E(k)}{\partial w_{j,i}(k)}}{\frac{\partial E(k-1)}{\partial w_{j,i}(k-1)} - \frac{\partial E(k)}{\partial w_{j,i}(k)}} \Delta w_{j,i}(k-1) \quad (4.23)$$

4.9 *RN* com *Group Method of Data Handling*

O pesquisador Gregory Alexeyevich Ivakhnenko (IVAKHNENKO, 1970, 1971), do Glushkov Institute of Cybernetics de Kiev, propôs o *Group Method of Data Handling* (*GMDH*). Este método é baseado no princípio de auto-organização heurística e é utilizado para construção de um modelo hierárquico com múltiplos níveis, que é transformado em uma descrição de entrada(s) e saída(s).

O *GMDH* é uma técnica baseada em mecanismos que servem ao aprimoramento gradual de modelos. Esta metodologia indutiva é diferente das técnicas dedutivas, utilizadas em outras *RNs*.

O *GMDH* resolve problemas complexos de interpolação, tais como: modelagem de sistemas complexos, predição, identificação e aproximação de processos, suporte à decisão, diagnóstico, prognóstico financeiro, reconhecimento de padrões e *clustering* de dados (DOLENKO *et al.*, 1996; IVAKHNENKO *et al.*, 1994).

O *GMDH* é também utilizado para a configuração de *RNs*. A *RN-GMDH* constitui-se de uma *RN* polinomial *GMDH* (ou *RN* auto-organizável). A *RN-GMDH* utiliza

ADALINES com pré-processadores não lineares (*N-ADALINES*). O número de camadas e o número de *ADALINES* em cada camada são determinados durante o treinamento, sendo que o treinamento é realizado pela aplicação da regra de aprendizado Widrow-Hoff (WIDROW & LEHR, 1990).

As *RNs-GMDH* são configuradas durante o treinamento. Os elementos de processamento nestas *RNs* têm, usualmente, duas entradas. A saída de cada elemento de processamento (*EP*) é uma combinação quadrática das duas entradas. As funções de ativação dos *EPs* podem ser consideradas polinômios de segunda ordem.

Durante o treinamento, o número de camadas de uma *RN-GMDH* aumenta. A cada época (iteração), uma nova camada é adicionada e *EPs* são gerados. Os *EPs* com desempenho inadequado são descartados e os que apresentam desempenho adequado são preservados na construção de novas camadas. Deste modo, o número de *EPs* em uma camada são determinados de forma adaptativa (ICHIHASHI *et al.*, 1995).

A técnica empregada na *RN-GMDH* divide os dados experimentais em três conjuntos, que são: um conjunto de treinamento, um conjunto de seleção de *EPs* promissores e um conjunto de testes. Os coeficientes dos polinômios em uma *RN-GMDH* são encontrados por regressão linear. A desvantagem de um esquema *GMDH*, baseado em regressão linear, é envolver um número acentuado de cálculos matriciais de alta dimensão.

A *RN-GMDH* é representada por uma construção de *ADALINES* com pré-processadores não lineares de segundo grau (*N-ADALINE*), como pesos, que correspondem aos coeficientes polinomiais da *RN-GMDH* original. O número de camadas e de *ADALINES* em cada camada é determinado durante o treinamento, que é realizado com aplicação da regra de aprendizado Widrow-Hoff (WIDROW & LEHR, 1990) de forma recursiva. As unidades *N-ADALINES* são apresentadas na figura 4.6.

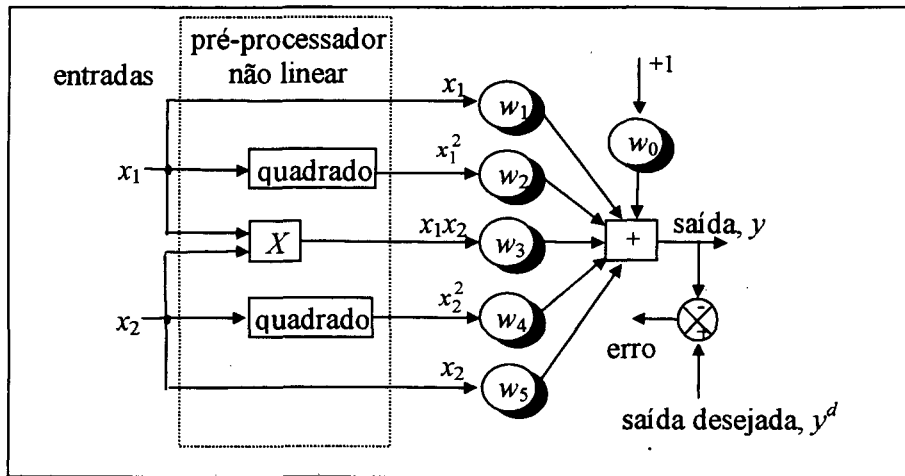


Figura 4.6. ADALINE com pré-processador não linear.

A partir disto, pode-se deduzir que a saída, y , da unidade é dada por :

$$y = w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1 x_2 + w_4 x_2^2 + w_5 x_2 \quad (4.24)$$

onde $w_i, i=[0;5]$, são os pesos da unidade, x_1 e x_2 são entradas da unidade, y é uma combinação linear de pesos e um polinômio de grau dois em termos das entradas. Se as unidades N -ADALINES são agrupadas em camadas, uma RN polinomial é obtida para aproximar mapeamentos complexos de alto grau. Seja $W = [w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ w_5]^T$ e $X = [1 \ x_1 \ x_1^2 \ x_1 x_2 \ x_2^2 \ x_2]^T$. A regra delta de Widrow-Hoff, para o treinamento da RN-GMDH, é regida por:

$$W(k+1) = W(k) + \eta \frac{X(k)}{X^2(k)} (y^d(k) - W^T(k)X(k)) \quad (4.25)$$

onde $y^d(k)$ é a saída desejada. A aplicação da equação (4.25) modifica a matriz de pesos, W , para reduzir a diferença entre a saída desejada e a saída atual da RN-GMDH. A RN-GMDH utilizada, no procedimento de identificação de processos, tem as seguintes etapas:

- (i) normalizar os dados e eliminar os componentes estacionários DC. Os dados de entrada-saída avaliados são processados antes do treinamento pelas equações (DUFFY & FRANKLIN, 1975):

$$u(k)_{\text{normalizado}} = \frac{u(k) - \bar{u}}{\sigma_u} \quad (4.26)$$

$$y(k)_{\text{normalizado}} = \frac{y(k) - \bar{y}}{\sigma_y} \quad (4.27)$$

onde $u(k)$ e $y(k)$, $k=1, \dots, N$, são vetores de entrada e saída das N amostras coletadas, \bar{u} e \bar{y} são a média de $u(k)$ e $y(k)$, e σ_u e σ_y são os desvios padrão de $u(k)$ e $y(k)$, respectivamente.

- (i) escolher as entradas externas à *RN-GMDH*;
- (ii) separar os dados experimentais em conjuntos de treinamento, seleção e validação;
- (iii) criar uma camada de *N-ADALINES* baseado no número de entradas, onde cada par produz uma *N-ADALINES*;
- (iv) inicializar os pesos dos *EPs* provisórios (*N-ADALINES*) em zero;
- (v) calcular os erros de saída dos *EPs* e modificar seus pesos.
- (vi) introduzir à *RN-GMDH* os dados do conjunto de seleção.
- (vii) o menor erro médio quadrático, obtido ao longo de todos os *EPs* durante o procedimento de seleção, é armazenado e utilizado como um critério de parada do treinamento. Assumindo que o menor erro médio quadrático para a camada atual é menor que o erro médio quadrático armazenado para a camada anterior, uma nova camada da *RN-GMDH* é criada. O tamanho desta camada é determinado pelo número de *EPs* selecionados. Os procedimentos de seleção e treinamento são repetidos para esta nova camada. As camadas existentes treinadas são insensíveis a este procedimento;
- (viii) testar o desempenho da *RN-GMDH* treinada, com um conjunto de dados de validação.

4.10 RNs recorrentes

As *RN feedforward* requerem ajustes à aplicações em identificação e controle de processos, pois estas *RNs* apresentam apenas unidades estáticas e podem somente identificar sistemas sem memória (SAKAWA *et al.*, 1995). As *RNs feedforward* não têm memória dinâmica e o método *tapped-delay-line* é, usualmente, adotado para uma *RN* representar sistemas dinâmicos. Este método apresenta desvantagens quanto à velocidade de aprendizado, devido ao número acentuado de neurônios na camada de entrada (HAYKIN, 1994).

Estas desvantagens estão presentes se a ordem do processo, a ser identificado, é desconhecida. Neste caso, pode-se sobre-estimar a ordem do processo e, por consequência, o número de neurônios na camada de entrada é considerado para tratar os sistemas de ordem desconhecida. Este aspecto conduz, usualmente, a uma *RN feedforward* com dimensão elevada e baixa velocidade de aprendizado.

Devido a sua estrutura, as *RNs* recorrentes (*RNRs*) não apresentam as desvantagens das *RNs feedforward*. As *RNRs* podem ser classificadas em totalmente ou parcialmente recorrentes. As *RNRs* totalmente recorrentes possuem as conexões *feedforward* e *feedback*, isto é, possuem todo tipo de conexões recorrentes e ajustáveis. Em *RNs* parcialmente recorrentes, a estrutura principal é não-recorrente. As conexões *feedforward* são treináveis e as conexões realimentadas são formadas através de um conjunto de unidades de contexto não treináveis.

JORDAN (1988) propôs um modelo de *RN* com conexões das unidades de saída para as unidades de entrada. Em contrapartida, ELMAN (1990) propôs um modelo em que as conexões *feedback* são das unidades da camada oculta às unidades de entrada. Estas *RNs* são denominadas *RNs* parcialmente recorrentes.

WILLIAMS & ZIPSER (1989) propõem um algoritmo de aprendizado para as *RNs* inteiramente conectadas. Esta *RN* é similar à *RN* de Hopfield (Hopfield, 1982, 1984), quanto à conectividade da *RN*, mas existem diferenças. Os pesos das conexões da *RN* de Hopfield são simétricos, enquanto os da *RN* de WILLIAMS & ZIPSER (1989) são não-simétricos (SAKAWA *et al.*, 1995, ISHII & KYUMA, 1996).

4.10.1 *RN* parcialmente recorrente de Elman

A *RN* parcialmente recorrente de Elman (*RNPRE*) utiliza o algoritmo *BP*. A *RNPRE* é, originalmente, projetada para aplicações de processamento de linguagem, mas é aplicada em reconhecimento de voz, predição de séries temporais, identificação de sistemas dinâmicos, entre outras aplicações (ELMAN, 1989, WILSON, 1995).

A utilização de coeficientes de aprendizado com valores elevados levam a oscilações e instabilidades no processo de treinamento da *RNPRE*. Para obter um desempenho adequado é necessário introduzir auto-conexões nas unidades de contexto

proporcionando a *RNP*RE de unidades de inércia, configurando uma *RNP*RE na (*RNP*REM). Deste modo, melhora-se a capacidade de memorização da *RNP*RE tanto à utilização de funções de ativação tanto lineares quanto não lineares. Um diagrama da *RNP*REM é apresentado na figura 4.7.

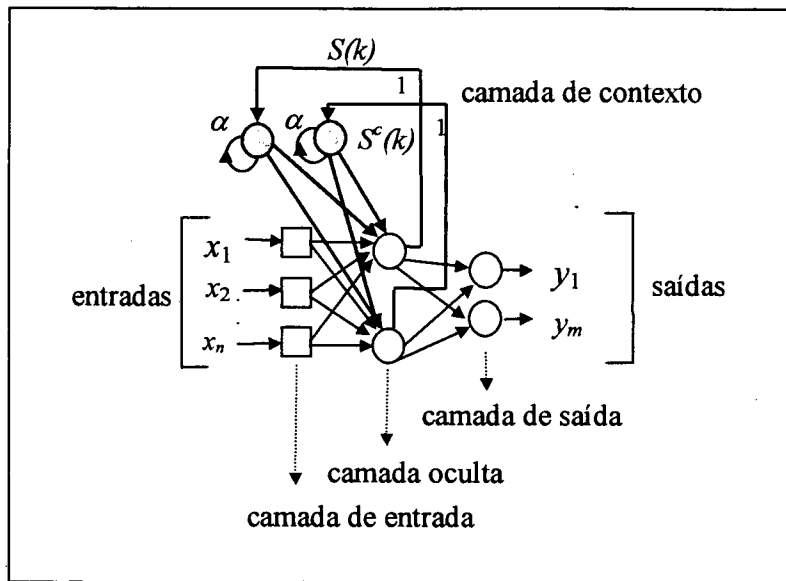


Figura 4.7. Diagrama esquemático da *RNP*REM.

Na figura 4.7, as entradas externas da *RNP*REM são representadas por X , as saídas por Y , as ativações das unidades ocultas por $S(k)$, e as saídas da camada de contexto são $S^c(k)$. As equações que caracterizam a *RNP*REM são (PHAM & LIU, 1992):

$$S(k) = f\{W(s,c)S^c(k), W(s,x)X(k-1)\} \tag{4.28}$$

$$S^c(k) = S^c(k-1) + \alpha S^c(k-1) \tag{4.29}$$

$$Y(k) = W(y,s)S(k) \tag{4.30}$$

onde $W(s,c)$, $W(s,x)$, $W(y,s)$ são matrizes de pesos e f é uma função vetor. A modificação na *RNP*RE tem a característica da j -ésima unidade de contexto na *RNP*REM ser regida por:

$$s_j^c(k) = s_j^c(k-1) + \alpha s_j^c(k-2) + \alpha^2 s_j^c(k-3) + \dots \tag{4.31}$$

O valor de α adotado, no caso $\alpha = 0.6$, é o mesmo para todas as auto-conexões e não é modificado durante o algoritmo de treinamento. Usualmente, $\alpha \in [0;1]$, onde o α próximo de 1 habilita as unidades da camada de contexto em agregar mais saídas passadas. A introdução de auto-conexões, nas unidades de contexto, aumenta a possibilidade da *RNPRE* modelar sistemas de ordem elevada.

4.11 RN função de base radial

A característica essencial da RN função de base radial (*RN-RBF*), conforme apresentada pela figura 4.8, é que a camada de saída é uma combinação linear dos sinais da camada oculta, sendo que a *RN-RBF* utilizada tem somente uma camada oculta (ou intermediária). As *RNs-RBF* permitem um procedimento de atualização de pesos, e posteriormente, possibilidade de provas de estabilidade e robustez (WARWICK, 1995).

Uma *RN-RBF* com m entradas e n saídas implementa o mapeamento $f_r: \mathbb{R}^m \rightarrow \mathbb{R}^n$ representado por (XIAOHONG *et al.*, 1996):

$$f_r(x) = A_0 + \sum_{j=1}^{Me} A_j \Phi(\|x - c_j\|) \quad (4.32)$$

onde Me é o número de funções de base radial; $x \in \mathbb{R}^m$; $\Phi(\bullet)$ é uma função de \mathbb{R}^+ para \mathbb{R}^- ; $\|\bullet\|$ denota uma norma Euclidiana, neste caso, a distância Euclidiana dos vetores de entrada, x , para os centros c_j ; $A_j \in \mathbb{R}^m, j=1, \dots, Me$, são vetores ponderados; $A_0 \in \mathbb{R}^m$ é um vetor constante e $c_j \in \mathbb{R}^m, j=1, \dots, Me$, são centros da função de base radial, onde

$$A_j = [\lambda_{mj}]^T, j=1, \dots, Me \quad (4.33)$$

A equação (4.32) é escrita na forma decomposta, onde

$$f_{r_i}(x) = A_{i0} + \sum_{j=1}^{Me} A_{ij} \Phi(\|x - c_j\|), \quad i = 1, \dots, m \quad (4.34)$$

Muitas funções são utilizadas como funções de base radial, $\Phi(\bullet)$, na camada intermediária. Uma condição suficiente para uma função ser candidata à função de base

radial é a de ser radialmente simétrica e a primeira derivada ser monotônica (POGGIO & GIROSI, 1990). As funções de base radial usualmente empregadas são:

$$(i) \text{ função Gaussiana: } \Phi(v) = \exp\left(-\frac{v^2}{\beta}\right) \quad (4.35)$$

$$(ii) \text{ função aproximação localmente linear: } \Phi(v) = v \quad (4.36)$$

$$(iii) \text{ função aproximação cúbica: } \Phi(v) = r^3 \quad (4.37)$$

$$(iv) \text{ função multiquadrática: } \Phi(v) = \sqrt{r^2 + \sigma^2} \quad (4.38)$$

$$(v) \text{ função multiquadrática inversa: } \Phi(v) = \frac{1}{\sqrt{r^2 + \sigma^2}} \quad (4.39)$$

$$(vi) \text{ função thin plate spline: } \Phi(v) = r^2 \log(r) \quad (4.40)$$

A figura 4.8 apresenta uma *RN-RBF* típica com uma camada intermediária.

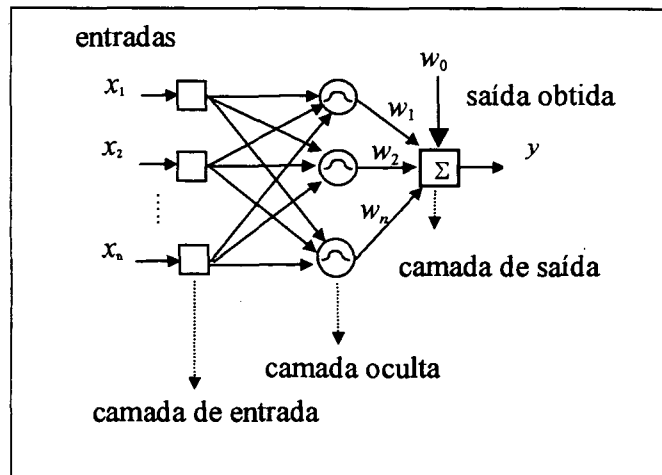


Figura 4.8. Uma *RN-RBF* típica com uma camada intermediária.

4.11.1 Métodos de configuração

O procedimento de aprendizado de uma *RN-RBF* é sintetizado em três etapas:

- (i) selecionar o número de centros;
- (ii) escolher os valores dos centros;
- (iii) ajustar os pesos.

Existe uma variedade de algoritmos para a configuração de *RNs-RBF*. Os algoritmos podem ser agrupados em duas categorias (SJÖBERG & LJUNG, 1995):

- (i) métodos construtivos;
- (ii) métodos de otimização não linear.

Nos métodos de otimização não linear, os parâmetros da *RN-RBF* (pesos, centros e variâncias) são otimizados através de métodos do tipo gradiente conjugado, Quase-Newton, Levenberg-Marquardt e outros (HARRIS *et al.*, 1993). A seguir são apresentados os detalhes relativos aos métodos construtivos.

4.11.1.1 Métodos construtivos

Nos métodos construtivos, os parâmetros não lineares (centros e variâncias) da *RN-RBF* são fixos e técnicas de estimação lineares determinam os valores de polarização e dos pesos (MOODY & DARKEN, 1989, CHEN *et al.*, 1990).

O treinamento de uma *RN-RBF* através de métodos construtivos inicia pela determinação dos centros da função base e pela escolha da função base. Os pesos da *RN-RBF* são ajustados para que a *RN-RBF* aproxime o valor da resposta desejada, através da de um critério de minimização do erro. Uma relação linear existe entre os pesos e a saída da *RN-RBF*, assim, um esquema de estimação linear pode ser empregado no treinamento dos pesos, por exemplo, algoritmos de mínimos quadrados, *GC* e pseudo-inversa (GOLUB & VAN LOAN, 1983).

O procedimento de otimização linear das *RNs-RBF* apresenta uma vantagem quando comparado com *RNs-MLP* que requerem algoritmos de aprendizado não linear. De forma similar ao caso da *RN-MLP*, o problema estrutural existe na *RN-RBF*, contudo, em termos do número e posições dos centros da função base requerida para um problema em particular.

A escolha dos centros é crítica, pois afeta a qualidade da aproximação da função obtida. Um requerimento é que o número de centros seja pelo menos suficiente para cobrir todo o domínio das entradas (CHEN *et al.*, 1990).

Existe um compromisso direto entre o número de centros da *RN-RBF* selecionados e a qualidade da aproximação obtida. Para os problemas de dimensão e complexidade baixa, um pequeno número de centros é requerido, embora isto não queira dizer que não possa haver uma sobre-parametrização da solução. Entretanto, quando o problema é de

dimensão elevada, uma solução razoável é providenciada na seleção de um número de centros fixo e razoavelmente baixo. Entre as possibilidades de seleção do número de centros têm-se: (i) escolha, com distribuição uniforme, dos valores do conjunto de dados de entrada da *RN-RBF*; (ii) aplicação de um algoritmo de *clustering*; e (iii) outros métodos, tais como: *AEs* (WHITEHEAD, 1996) e *RNs* com aprendizado não-supervisionado, por exemplo, as *RNs* do tipo *SOM* ou *LVQ* (HAYKIN, 1994).

4.11.1.1.1 Aprendizado via *k-médias* e técnica da pseudo-inversa

Neste algoritmo de treinamento, a forma funcional, $\Phi(\bullet)$, é fixa, enquanto os centros, c_j , são ajustados pelo algoritmo de *clustering k-médias*. O algoritmo *k-médias* (MOODY & DARKEN, 1989) particiona o conjunto de dados em subconjuntos de *clusters*, e encontra um conjunto de centros dos *clusters*. O número de centros da *RN-RBF*, M_c , deve ser especificado. Os centros dos *clusters* são centros das funções de base radial, na camada oculta da *RN-RBF*. O conjunto de dados é particionado tal que para cada amostra no espaço de entrada, definido por $x_i(t)$, é atribuído o *cluster* com o centro mais próximo.

Existem as versões *off-line* (LEONARD & KRAMER, 1991) e *on-line* (MOODY & DARKEN, 1989) do algoritmo de *clustering*. Entretanto, as duas versões minimizam a distância quadrática total entre os pontos de dados atribuídos para cada *cluster* e os centros dos *clusters*.

Na versão *off-line*, o centro de cada *cluster* é inicializado aleatoriamente. A seguir, cada ponto é atribuído para o *cluster* com o centro mais próximo. Quando cada ponto no conjunto de treinamento já foi atribuído para um determinado *cluster*, cada centro de *cluster* é movido para a média dos dados do *cluster*. Os dados são novamente designados para o *cluster* com centro mais próximo, e o procedimento é repetido até convergir.

O algoritmo heurístico *p-nearest neighbor* (MOODY & DARKEN, 1989) é utilizado para a determinação da superposição, β_{RBF_j} , entre as funções de base radial. Esta heurística é considerada para superposição entre as funções de base radial e é utilizada após as posições dos centros da *RN-RBF* estarem fixas.

Existem modificações do algoritmo básico, propostas na literatura. CHEN *et al.* (1992) propõem a utilização de um algoritmo *k*-médias recursivo combinado com o algoritmo dos *MQR*, para a determinação dos pesos de uma *RN-RBF*. O algoritmo de *clustering k-médias* segue o procedimento:

- (i) escolher inicialmente $k=M_{ci}$ centros dos *clusters* para $j=1, \dots, M_{ci}$, aleatoriamente, no conjunto de dados de entrada(s) e saída(s), e atribuir $M=M_{ci}$;
- (ii) para $i=1, \dots, N$, onde N é o número de amostras do conjunto de dados, encontrar k , tal que a distância do vetor de entrada, x_i , para um centro do cluster é mínima

$$\|x_i - c_k\|_2 = \min_j \left\{ \|x_i - c_j\|_2 \right\}, \quad \text{para } j = 1, \dots, M_c \quad (4.41)$$

e atribuir a amostra i para o *cluster* k , $i \in C_k$.

- (iii) calcular a média dos pontos de dados pertencentes a cada *cluster*:

$$\bar{x}_j = \frac{1}{N_j} \sum_{i \in C_j} x_i, \quad \text{para } j = 1, \dots, M_c \quad (4.42)$$

onde N_j é o número de amostras pertencentes ao *cluster* j . O novo centro de cada *cluster* é a média, $c_j = \bar{x}_j$;

- (iv) remover o *cluster* j para $j=1, \dots, M_c$, se $N_j=1$, e atribuir $M_c = M_c - 1$;
- (v) repetir os passos (ii) a (iv) até que os centros dos *clusters* e a pertinência convirja.
- (vi) utilizar a heurística *p-nearest neighbour*, especificando o valor de β_{RBFj} para $j=1, \dots, M_c$:

$$\beta_{RBFj} = \frac{1}{p} \sum_{k=1}^p \|c_j - c_k\|_2^2 \quad (4.43)$$

onde c_k é o k -ésimo cluster mais próximo de c_j . Em LEONARD & KRAMER (1991) é sugerido $p=2$.

- (vii) calcular a polarização (w_0) e o vetor de pesos através de pseudo-inversa de Moore-Penrose (ou através de mínimos quadrados linear).

4.11.1.1.2 Aprendizado via algoritmo dos mínimos quadrados ortogonal

Outras formas de aprendizado construtivo podem ser determinadas, através da análise de regressão linear. Neste caso, os centros da *RN-RBF* são escolhidos iterativamente de um conjunto de possíveis centros e os pesos ajustados, através de algoritmos de mínimos quadrados. Os exemplos mais comuns são a regressão *forward* e a regressão escalonada (WARWICK, 1995, ZHU & BILLINGS, 1996).

O método dos mínimos quadrados ortogonal (CHEN *et al.*, 1991) consiste da seleção, de forma supervisionada, dos centros da *RN-RBF*. As técnicas ortogonais podem ser eficientes para o tratamento de mau condicionamento numérico e os problemas de seleção de estrutura (BILLINGS & AGUIRRE, 1995). O método dos mínimos quadrados ortogonal é robusto numericamente na resolução da equação clássica dos mínimos quadrados para um vetor ponderado. O algoritmo de estimação de parâmetros ortogonal para um modelo multivariável tem a forma (CHEN *et al.*, 1989, 1990):

$$z = P\Theta + \Xi \quad (4.44)$$

com

$$z = \begin{bmatrix} z(1) \\ \vdots \\ z(N) \end{bmatrix}, \quad P = \begin{bmatrix} p_1(1) & \cdots & p_m(1) \\ \vdots & \vdots & \vdots \\ p_1(N) & \cdots & p_m(N) \end{bmatrix}, \quad \Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \end{bmatrix}, \quad \Xi = \begin{bmatrix} \xi(1) \\ \vdots \\ \xi(N) \end{bmatrix} \quad (4.45)$$

onde z são as variáveis dependentes (saídas da *RN-RBF*), Θ são os parâmetros desconhecidos dos regressores da matriz, P , que é definida em termos de uma função linear (ou não linear) das saídas, as entradas atrasadas e os Ξ é a matriz dos resíduos.

Cada subsistema que compõe o vetor de saídas de um processo multivariável envolve somente alguns termos que caracterizam a dinâmica do sistema. Se os termos significativos são conhecidos *a priori*, somente eles são utilizados para configurar a matriz de regressão, P . Um problema de mínimos quadrados linear pode ser definido para encontrar as estimativas dos parâmetros $\hat{\Theta}$ que minimizam $\|z - P\Theta\|$, onde $\|\cdot\|$ é a norma Euclidiana (CHEN *et al.*, 1989). A solução deste problema satisfaz a equação

$$P^T P \Theta = P^T z \quad (4.46)$$

onde $P^T P$ é denominada de matriz de informação.

CHEN *et al.* (1991) tratam o problema da seleção de centros como um exemplo de escolha de um subconjunto de regressores significativos dentre um conjunto de candidatos. Dado um conjunto de candidatos para centros, a estratégia consiste em transformar este conjunto em uma base ortogonal no espaço da camada intermediária. Isto permite o cálculo da contribuição de cada vetor de base à obtenção da saída desejada pela *RN-RBF*. Assim a matriz de regressão é decomposta em (BILLINGS *et al.*, 1989, CHEN *et al.*, 1989, 1991)

$$P = WA \quad (4.47)$$

onde

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{12} & \cdots & \alpha_{12} \\ 0 & 1 & \alpha_{12} & \cdots & \alpha_{12} \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & 1 & \alpha_{12} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad (4.48)$$

é uma matriz triangular superior $M \times M$ e

$$W = [w_1 \dots w_M] \quad (4.49)$$

é uma matriz $N \times M$ com colunas ortogonais que satisfazem

$$W^T W = H \quad (4.50)$$

onde H é uma matriz diagonal positiva obtida por

$$H = \begin{bmatrix} h_1 & & 0 \\ & \ddots & \\ 0 & & h_M \end{bmatrix} \quad (4.51)$$

com

$$h_i = w_i^T w_i = \sum_{t=1}^N w_i(t) w_i(t), \quad i = 1, \dots, M \quad (4.52)$$

Assim, a equação (4.44) é reformulada de forma que

$$z = (PA^{-1})(A\Theta) + \Xi = Wg + \Xi \quad (4.53)$$

onde

$$(A\Theta) = g \quad (4.54)$$

Devido Ξ ser considerado descorrelacionado com $p_i(t)$, a solução do mínimos quadrados ortogonal \hat{g} é dada por:

$$g = H^{-1}W^T z \quad (4.55)$$

ou

$$g_i = \frac{w_i^T z}{w_i^T w_i}, \quad i=1, \dots, M \quad (4.56)$$

As técnicas para a estimação dos parâmetros, $\hat{\Theta}$, do sistema triangular (4.55) são usualmente conhecidos como métodos de mínimos quadrados ortogonal. Entre as abordagens utilizadas no cálculo dos elementos da matriz A destacam-se métodos de Gram-Schmidt, Gram-Schmidt modificado, transformações de Householder e transformações de Givens (GOLUB & VAN LOAN, 1983).

Nesta tese, utiliza-se o método de Gram-Schmidt. O número, M , dos regressores candidatos pode ser grande, no caso da *RN-RBF*. Os regressores significativos podem ser identificados pelo algoritmo proposto por CHEN *et al.* (1989, 1990, 1991), onde a soma dos quadrados de $z(t)$ é

$$z^T z = \sum_{i=1}^M g_i^2 w_i^T w_i + \Xi^T \Xi \quad (4.57)$$

Nota-se que cada w_i consiste de uma proporção variável da variância. Entretanto, a taxa de redução do erro devido a w_i é definido através de

$$[\text{err}]_i = \frac{g_i^2 w_i^T w_i}{z^T z} \quad (4.58)$$

A taxa de redução do erro oferece uma forma simples e efetiva para a seleção de um subconjunto de regressores significativos para um número elevado de candidatos, através de uma forma de regressão *forward*. No j -ésimo passo, um regressor é

selecionado, produzindo-se o maior valor de $[err]_j$, entre os demais candidatos. O procedimento de seleção é finalizado quando

$$1 - \sum_{j=1}^{M_s} [err]_j < \rho \quad (4.59)$$

onde $0 < \rho < 1$ é a tolerância desejada. O parâmetro estimado, $\hat{\Theta}_s$, para o subconjunto resultante é então calculado como

$$A_s \Theta_s = g_s \quad (4.60)$$

onde A_s é a matriz triangular superior $M_s \times M_s$ (CHEN *et al.*, 1989, 1990, 1991, ZHU & BILLINGS, 1996). O algoritmo dos mínimos quadrados ortogonal apresentado para o aprendizado de *RNs-RBF* pode também ser utilizado na identificação de processos *MIMO* não lineares e para o aprendizado em sistemas de inferência nebulosos.

4.11.2 Diferenças entre *RNs-RBF* e *RNs-MLP*

As *RNs-RBF* e as *RNs-MLP* são aproximadores universais de funções. Existem, no entanto, várias diferenças entre estes modelos neurais. Na tabela 4.1 apresenta-se uma comparação entre uma *RN-RBF* e uma *RN-MLP* convencional.

Tabela 4.1. Comparação entre uma *RN-RBF* e uma *RN-MLP* convencional.

item	<i>RN-RBF</i>	<i>RN-MLP</i>
(i)	geralmente possui uma única camada intermediária	pode apresentar uma ou mais camadas intermediárias
(ii)	neurônios das camadas intermediárias têm funções diferentes dos neurônios da camada de saída	neurônios das camadas intermediárias e de saída têm funções semelhantes
(iii)	a entrada da função de ativação é a distância entre os vetores de entrada e de pesos	a entrada da função de ativação é o produto interno dos vetores de entrada e de pesos
(iv)	separa padrões de entrada com hiperelipsóides	separa padrões de entrada com hiperplanos
(v)	constrói aproximadores locais para o mapeamento de entrada e saída	constrói aproximadores globais para o mapeamento de entrada e saída

Fonte: BRAGA *et al.* (1998).

4.11.3 *RN-RBF* x modelo nebuloso de *TSK*: similaridades e diferenças

Diversos artigos apresentam estudos comparativos e de equivalência entre *RNs-RBF* e sistemas de inferência nebulosos do tipo *TSK*, com a parte do conseqüente regido por uma constante do tipo função de base.

HUNT *et al.* (1994) apresentam uma forma generalizada do teorema de equivalência funcional entre uma *RN-RBF* e uma classe de modelos restrita do tipo *TSK*. JANG & SUN (1993) demonstram a equivalência funcional entre uma *RN-RBF* e um sistema simplificado de inferência nebulosa. É interessante notar que as origens das *RNs-RBF* são motivadas pela fisiologia, onde a sobreposição localmente ajustada de campos receptivos é uma estrutura presente em regiões do córtex cerebral e córtex visual. MOODY & DARKEN (1989), baseados nos campos receptivos biológicos, propõem a estrutura da *RN-RBF*, que emprega campos receptivos locais para mapear funções. As origens dos sistemas de inferência nebulosos, por outro lado, são motivados pelos estudos em ciência cognitiva.

BERSINI *et al.* (1995) apresentam as formas de equivalência de mapeamentos onde o *SN* de inferência *TSK* é tratado por uma forma geral, que inclui a *RN-RBF*, como resultado de algumas opções de arquitetura e simplificações. A análise de alguns resultados em problemas de classificação e de identificação indica algumas diferenças entre estas estruturas. Para a resolução de um problema, a *RN-RBF* projeta os dados do problema em um novo espaço abstrato. Ao passo que o *SN-TSK* decompõe o problema e, assim, permite diversas operações locais. A *RN-RBF* tem uma estrutura mais simples e com menos parâmetros, enquanto o *SN-TSK*, por princípio, necessita de mais parâmetros e esta necessidade aumenta rapidamente com o aumento da dimensão dos problemas (BERSINI *et al.*, 1995). Ambas metodologias têm vantagens, tais como: (i) a funcionalidade local, (ii) o aprendizado rápido; e (iii) a sintonia ao nível de saída linear nos parâmetros.

Entretanto, ambas apresentam desvantagens comuns, sendo as mais críticas: a “maldição” da dimensionalidade e os problemas devido ao sobre-treinamento. Uma abordagem relevante é a análise e configuração de controladores hierárquicos (RAJU *et al.*, 1991) pode-se diminuir a complexidade do projeto de controle pela decomposição do

controlador em uma estrutura em cascata. Além disso, segundo WANG (1999), pode-se provar que sistemas hierárquicos são aproximadores universais.

4.12 RNs em identificação de processos multivariáveis

As metodologias de controle neural mais simples são centradas em procedimentos de inversão de processos, onde a inversa da dinâmica do processo é utilizada para o controle em malha aberta. A *RN-MLP* é muito utilizada para os propósitos de identificação e controle de processos, mas não existe garantia que a *RN-MLP* seja bem sucedida em controlar um determinado processo. O modelo matemático do processo, obtido através da *RN-MLP*, tem representação estrutural, ou seja, modelo caixa-preta. Esta característica é proibitiva quanto à análise das propriedades do modelo aprendidas.

A capacidade de identificação não linear das *RNs* pode ser explorada para aprimorar as metodologias de controle preditivo, baseadas em modelo, pois um modelo preciso do processo é parte essencial na aplicação eficiente desta metodologia (RONCO & GAWTHROP, 1997).

A literatura é extensa quanto a estudos relativos a aspectos de modelagem e identificação de processos através de *RNs*. Os estudos incluem a teoria e as aplicações em identificação caixa-preta não-paramétrica, identificação paramétrica, identificação da inversa do processo, tanto para processos *SISO* quanto para *MIMO*, principalmente não lineares (SJÖBERG, 1995, HUNT *et al.*, 1992).

4.12.1 Procedimento para identificação de processos via *RNs*

Em identificação é importante o papel das *RNs* em modelos *black-box* (identificação estrutural) e *grey-box* de sistemas dinâmicos não lineares, tratado com propriedade por SJÖBERG (1995) e NARENDRA & PARTHSARATHY (1990). As *RNs* podem ser aplicadas, com diferentes regressores, em modelos dinâmicos *black-box* não lineares, tais como:

- (i) *RN-NFIR (Non-linear Finite Impulse Response model)*: usa somente medidas passadas do vetor $u(k-i)$ como regressores;
- (ii) *RN-NARX (Non-linear AutoRegressive model structure eXogenous inputs)*: emprega o vetor das entradas $u(k-i)$ e das saídas $y(k-j)$ passadas como regressores (modelo série-paralelo);
- (iii) *RN-NOE (Non-linear Output Error model)*: utiliza o vetor das entradas passadas $u(k-i)$ e das saídas previstas passadas $\hat{y}_u(k-j|\theta)$, como entradas da *RN*. Este modelo é denominado modelo paralelo;
- (iv) *RN-NARMAX (Non-linear AutoRegressive Moving Average model structure with eXogenous inputs)*: utiliza os vetores $e(k-i)$, $u(k-i)$ e $y(t-j)$;
- (v) *RN-BJ (Non-linear Box-Jenkins model structure)*: utiliza os quatro tipos de regressores mencionados nos itens anteriores;
- (vi) modelos não lineares, representados no espaço de estados: utiliza os componentes passados “virtualmente” das saídas.

A identificação de um processo não linear, utilizando-se *RNs*, pode ser dividido em problemas básicos:

- (i) a seleção do sinal de treinamento;
- (ii) o cálculo dos parâmetros; e
- (iii) a validação e a seleção da configuração apropriada.

4.12.1.1 Seleção do sinal de treinamento

Para os sistemas lineares, existe uma base teórica para a seleção de sinais de treinamento. Entretanto, os sinais de ruído branco são freqüentemente utilizados (LJUNG, 1987). Em identificação de sistemas não lineares, o sinal ruído branco também é utilizado, porém não existe base teórica para esta escolha.

Segundo VAN CAN *et al.* (1995), a escolha do sinal de treinamento influencia, severamente, o desempenho da *RN*. Assim, o conjunto de dados de treinamento deve conter o máximo de informação relevante sobre a dinâmica do processo. Devido a falta de base teórica para a seleção de um sinal ótimo de treinamento, nenhuma especificação geral pode ser dada. Além disso, espera-se que o sinal de treinamento contenha todas as freqüências e os espectros de amplitude relevantes. Dependendo das propriedades estatísticas do estimador, existe a necessidade de obtenção dos dados de treinamento agrupados em um intervalo específico de operação.

4.12.1.2 Cálculo dos parâmetros e modelos neurais em identificação

Existem diversas formas de treinamento para as arquiteturas de *RNs* (HAYKIN, 1994). O procedimento usual na área de controle de processos é a escolha da *RN feedforward* com treinamento através do *BP*.

A *RN feedforward* básica é uma *RN* não dinâmica. Uma maneira simples e usualmente utilizada de introduzir-se dinâmica na *RN* consiste em utilizar um vetor de entradas X , composto das entradas e saídas passadas (atrasadas) do processo. Desta maneira, uma *RN feedforward* pode ser interpretada como um modelo *NARMAX* do processo, onde o modelo *NARMAX* série-paralelo para um processo *SISO* pode ser representado por:

$$\hat{y}(k) = f \{y(k-1), \dots, y(k-n); u(k-d), \dots, u(k-d-m)\} + e(k) \quad (4.61)$$

onde f é uma função não linear que descreve o comportamento do sistema. Esta maneira de introduzir as dinâmicas na *RN* estática tem a vantagem de ser simples.

NARENDRA & PARTHASARATHY (1990) apresentam quatro modelos para a representação de sistemas *SISO* discretos no tempo. Esta proposta pode ser estendida para os processos *MIMO*, conforme apresentado nas figuras 4.9 a 4.12, e equacionada pelos respectivos vetores das entradas, $u(k)$, e saídas, $y(k)$, onde:

$$\text{Modelo I: } y(k+1) = \sum_{i=0}^{m-1} A_i y(k) + G[u(k), u(k-d-1), \dots, u(k-d-m+1)] \quad (4.62)$$

$$\text{Modelo II: } y(k+1) = F[y(k), y(k-1), \dots, y(k-n+1)] + \sum_{i=0}^{m-1} B_i u(k-i) \quad (4.63)$$

$$\text{Modelo III: } y(k+1) = F[y(k), y(k-1), \dots, y(k-n+1)] + G[u(k), u(k-d-1), \dots, u(k-d-m+1)] \quad (4.64)$$

$$\text{Modelo IV: } y(k+1) = F[y(k), y(k-1), \dots, y(k-n+1), u(k), u(k-d-1), \dots, u(k-d-m+1)] \quad (4.65)$$

onde F e G são funções das saídas e entradas, respectivamente; enquanto A e B são polinômios das entradas e saídas do processo, respectivamente.

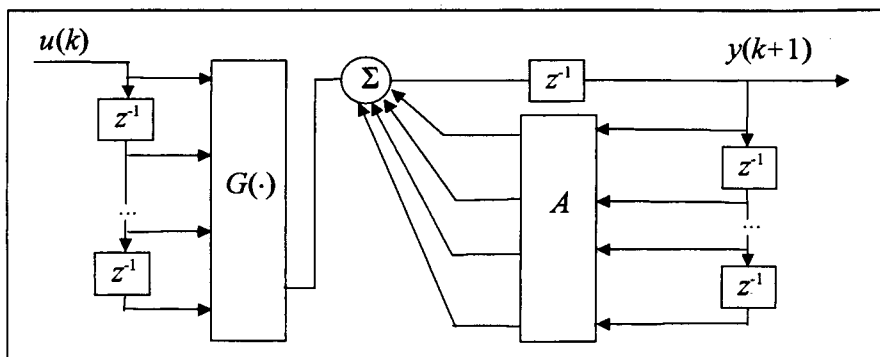


Figura 4.9. Representação de processos *MIMO* (modelo I).

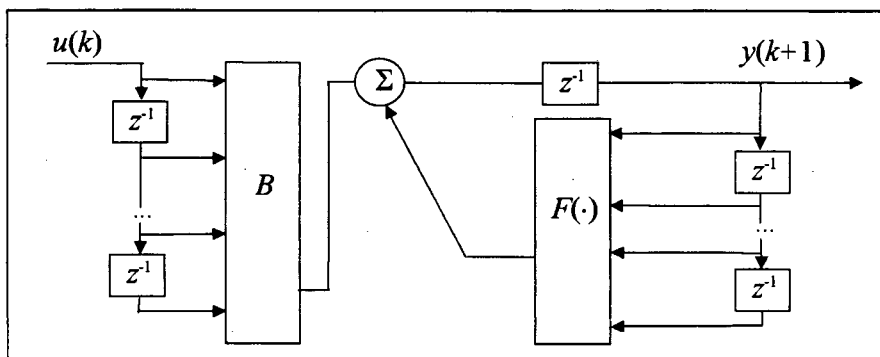


Figura 4.10. Representação de processos *MIMO* (modelo II).

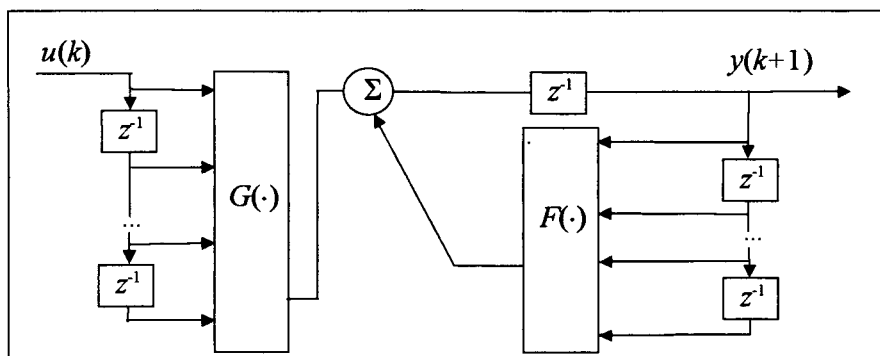


Figura 4.11. Representação de processos *MIMO* (modelo III).

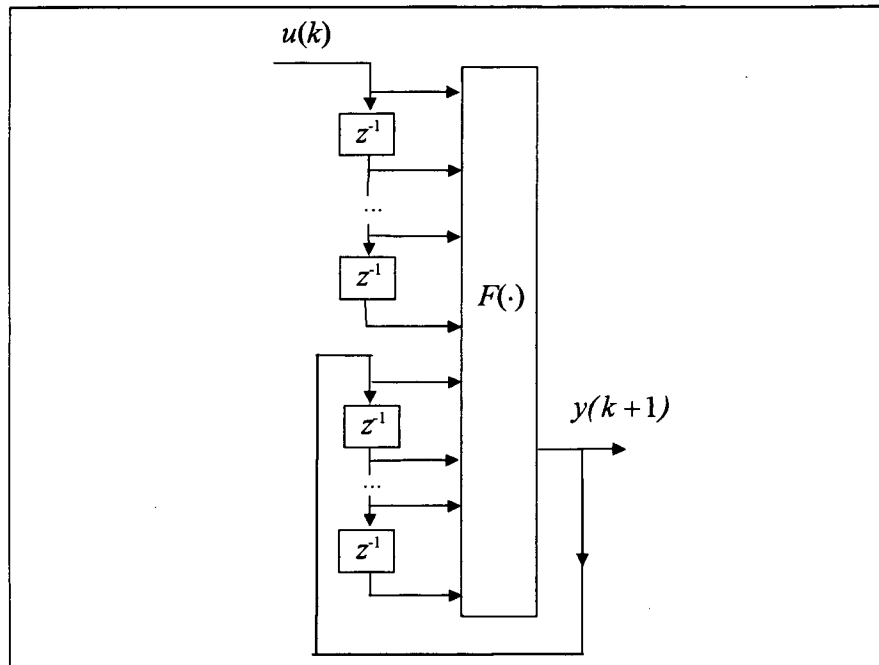


Figura 4.12. Representação de processos *MIMO* (modelo IV).

4.12.1.3 Validação e seleção da configuração das RNs

O vetor de entrada de uma RN pode conter um ou mais atrasos de tempo das entradas e saídas do processo, dependendo da ordem do mesmo. Os processos práticos e correlações não lineares apresentam ordem desconhecida.

No caso das RNs *feedforward* e *RN-RBF* necessita-se definir o número de neurônios, na camada oculta, para a definição da configuração do modelo neural. Em geral, entretanto, encontrar a configuração de RN adequada significa obter um número considerável de configurações a serem treinadas e testadas na análise do potencial de previsão. A adoção de algoritmos de treinamento eficientes diminui, razoavelmente, o tempo de treinamento e de teste dos modelos neurais.

Uma outra observação relevante é a necessidade de especificação de um critério para a análise de desempenho das configurações neurais. Isto é, usualmente, realizado pelo cálculo de uma função dos erros, usualmente, quadrática.

4.12.1.4 Generalização e sobre-treinamento

Uma *RN* generaliza (extrapola) quando calcula os mapeamentos entre as entradas e as saídas, de forma satisfatória, para um conjunto de dados de teste. Os dados de teste não são utilizados no treinamento (ou projeto) da *RN*. O conjunto de teste é utilizado na fase de validação dos modelos neurais em identificação de processos (ver capítulo 8).

Quando uma *RN* aprende muitas relações de entrada e saída ou o número de parâmetros da *RN* é grande, a *RN* pode memorizar (interpolat) os dados de treinamento, porém pode ser menos apta a generalizar. Uma forma de evitar este problema de sobre-treinamento é utilizar um conjunto de dados de treinamento amplo, que abranja toda a região de interesse e limitar o número de parâmetros, tão estritamente quanto possível.

A utilização de uma quantidade inadequada de dados de treinamento levará a um problema de estimação de parâmetros mal-condicionado. O sobre-treinamento aparece quando o erro de aprendizado é pequeno, mas o erro de teste é grande. A escolha da *RN* e do algoritmo de treinamento influencia a ocorrência do problema de sobre-treinamento.

O desempenho do algoritmo de estimação de parâmetros de uma *RN* pode ser aprimorado pela adição de um termo de regularização (SJÖBERG, 1995). Esta função tem um valor grande para mapeamentos “suaves” (modelos simples) e pequeno para outros (modelos complexos). A regularização prioriza a obtenção de modelos menos complexos, se utilizada para melhorar a capacidade de generalização de *RNs*. Uma possibilidade é adicionar um termo, que consiste do erro médio quadrático dos parâmetros da *RN* na função objetivo original. A regularização é uma característica necessária para manter a variância do erro pequena em modelos que empregam muitos parâmetros. Neste caso, a resposta da *RN* será menos sensível ao sobre-treinamento.

Uma outra alternativa, esta utilizada nos procedimentos de identificação de processos multivariáveis via *RNs* (capítulo 8), é calcular, a cada época, a capacidade de generalização da *RN*. Neste procedimento o cálculo do erro de validação é verificado na análise do comportamento do treinamento da *RN*.

Quando a *RN* apresenta um comportamento de sobre-treinamento, o erro de validação aumenta, enquanto o erro de treinamento decresce. Se o erro de validação continua crescendo para um número específico de épocas, o procedimento de treinamento é finalizado e os parâmetros da *RN*, na época, que apresentam menor erro de treinamento,

são retomados. A equação proposta, para verificação do desempenho do modelo neural, considera a função de erro médio quadrático do conjunto de validação, na época k , definida por $E_{val}(k)$. O valor ótimo, $E_{ótimo}(k)$, trata-se do menor erro, para o conjunto de validação, obtido até a época k onde

$$E_{ótimo}(k) = \min_{t' \leq t} E_{val}(k') \quad (4.66)$$

A partir disto, define-se o coeficiente de perda de generalização, para a época k , pelo incremento do erro de validação sobre o erro mínimo (em percentagem), ou seja,

$$CPG(k) = \frac{E_{val}(k)}{E_{ótimo}(k)} - 1 \quad (4.67)$$

Esta proposta se destina a permitir que a configuração de uma *RN* generalize adequadamente, mas apresenta alguns inconvenientes, pois é, muitas vezes, sobreparametrizado. Outra limitação ocorre quando o conjunto de dados de validação é extenso, o que acarreta um acentuado custo computacional.

O método de tentativa e erro é usualmente utilizado no projeto de *RNs*, mas existem outros métodos alternativos, tais como *cascade correlation*, algoritmos punitivos (REED, 1993) e *AEs* (LINKENS & NYONGESA, 1996a).

Outras abordagens possíveis no tratamento do problema do sobre-treinamento são a utilização de critérios do tipo *GMDH* (ICHIHASHI *et al.*, 1995) e análise estatística (AMARI *et al.*, 1997).

4.13 *RNs* em controle de processos multivariáveis

A funcionalidade de uma *RN* é definida através de aprendizado, por amostras que constituem o conjunto de treinamento. Isto implica que uma parte importante do projeto de um controlador neural é o mecanismo em que as amostras, utilizadas no aprendizado, são coletadas e empregadas para o ajuste dos parâmetros da *RN*. Este mecanismo é similar aos dos controladores adaptativos, contudo, os controladores neurais são adaptativos por natureza.

As aplicações em controle neural de processos são relevantes em tópicos tais como: modelagem de um controlador existente, otimização de controladores baseados ou livres de modelo, controle robusto, controle adaptativo e controladores neurais parametrizados (SAMAD, 1995). Todavia, algumas pesquisas recentes são realizadas com controladores neurais modulares do tipo *Local Model Network* e *Gated Modular Neural Networks* (RONCO & GAWTHROP, 1997). A seguir são apresentados diversos tipos de projetos que utilizam redes neurais.

AGARWAL (1997) apresenta um estudo da estruturação de controladores baseados em *RNs*. Segundo AGARWAL (1997), os esquemas de controle, que utilizam *RNs*, podem ser classificados em: (i) a *RN* atua somente como uma metodologia auxiliar; e (iii) a *RN* atua como um controlador. Outras divisões e classificações para os projetos de controle neurais são propostas na literatura (HUNT *et al.*, 1992, HAGAN & DEMUTH, 1999).

Um esquema de classificação de projetos de controle neural é baseado nesta observação foi proposto por ANDERSEN (1998), considerando-se o método pelo qual o sinal de erro para o treinamento da *RN* é obtido, conforme exposto na figura 4.13.

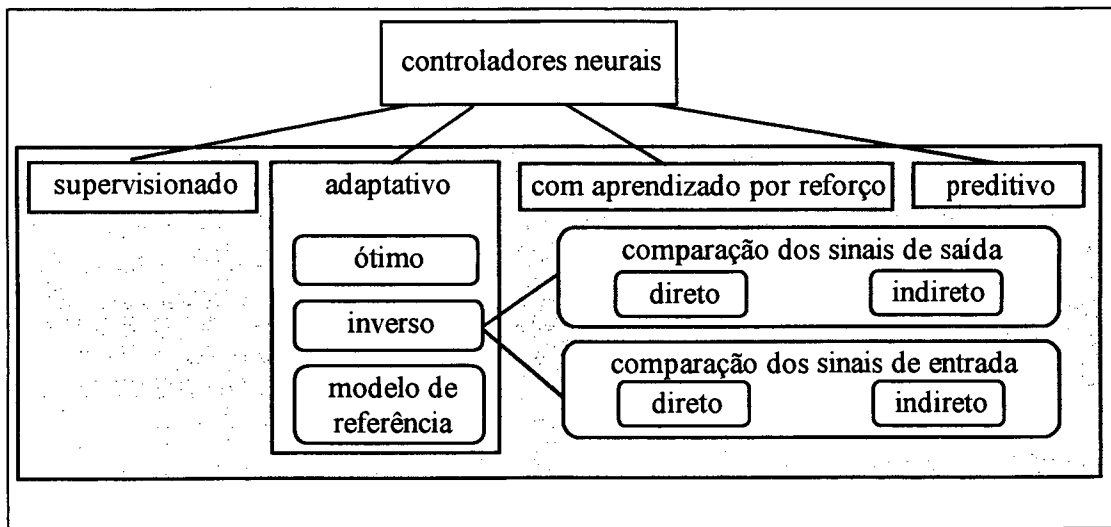


Figura 4.13. Classificação dos esquemas de controle neural, baseada na forma em que o sinal de erro é utilizado, para o treinamento da *RN*.

As formas utilizadas nesta tese de controle neural são o controle ótimo neural e controle preditivo neural. O controle ótimo neural é uma variação do controle ótimo

clássico, em que as *RNs* são utilizadas para identificação de processos, conforme apresentado na figura 4.14. No controlador ótimo neural, a cada amostra, uma função custo, baseada nas amostras coletadas até aquele instante, é minimizada, assumindo-se que o emulador neural do processo é adequado e preciso.

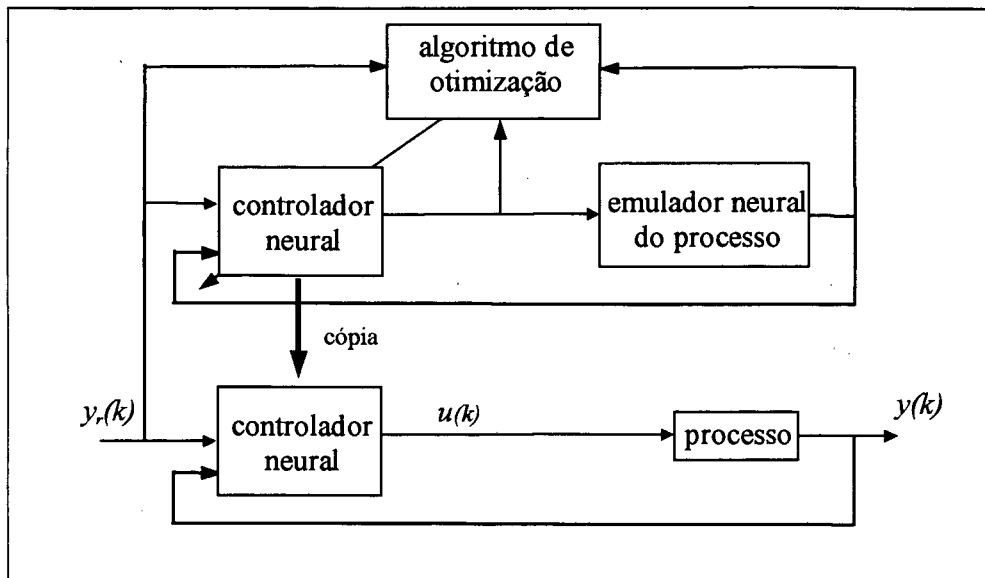


Figura 4.14. Controle ótimo neural.

Quanto ao controlador preditivo neural, um modelo neural *forward* da dinâmica do sistema é utilizado para estimar os comportamentos futuros do sistema. Os valores previstos são alimentados em um controlador, que calcula o sinal de controle ótimo de acordo com um índice de desempenho. Outra forma de controle preditivo é treinar um controlador neural para imitar um controlador ótimo. Os detalhes destes controladores são apresentados no capítulo 6.

4.14 Conclusão do capítulo

Neste capítulo foram apresentados os aspectos relativos à base teórica de *RNs* e a motivação de sua utilização, em identificação e controle de processos. Entre as características que motivam a utilização das *RNs* deve-se enfatizar: a habilidade de tratar processos não lineares, tolerância a falhas, adaptabilidade, auto-organização, aprendizado,

generalização, processamento paralelo e abstração. Alguns tipos de *RNs* foram descritos, a *RN-MLP*, a *RN-PREM*, a *RN-GMDH* e a *RN-RBF*. Todas elas constituem-se de aproximadores universais de funções com dada precisão, contudo com diferenças na estrutura e na forma de treinamento. A *RN-MLP* caracteriza-se por realizar mapeamentos globais. A *RN-PREM* apresenta recorrências, tornando-a eficiente para o tratamento de sistemas dinâmicos. A *RN-GMDH* apresenta avaliação através de três conjuntos de dados (estimação, seleção e validação) e é menos susceptível a problemas de sobre-treinamento. A *RN-RBF* separa os padrões de entrada com hiperelipsóides e apresenta uma estrutura apta à construção de mapeamentos locais dos dados de entrada e saída. Os tópicos de projeto em identificação e controle multivariáveis de processos através de *RNs* também foram abordados.

Quanto ao controle de processos apresenta-se uma classificação simplificada das configurações de controladores neurais e as características fundamentais de diversos esquemas de controle neural, entre os quais controle supervisionado, aprendizado por reforço, adaptativo, preditivo e inverso. As *RNs*, mesmo com todas as potencialidades que as tornam eficientes no tratamento de processos não lineares, têm limitações quanto a aplicações em identificação e controle de processos. Algumas delas são comuns a aplicações em outras áreas do conhecimento, entre as quais: (i) a dificuldade da análise de estabilidade e a convergência, em controle de processos; (ii) a “maldição” da dimensionalidade; (iii) a perda de generalização do modelo neural devido aos problemas de sobre-treinamento; e (iv) como o algoritmo *BP* fornece uma aproximação da trajetória no espaço de pesos, calculado por métodos de gradiente descendente, os pesos podem ficar presos a platôs (ou a mínimos locais).

Nesta tese aborda-se aspectos de identificação de processos multivariáveis via *RN-MLP*, a *RN-PREM*, a *RN-GMDH* e a *RN-RBF*. Quanto a controle processos propõe-se o projeto de controle preditivo baseado em *RN-RBF* para o tratamento de processos não lineares e é descrito no capítulo 6.

Capítulo 5

SISTEMAS HÍBRIDOS INTELIGENTES

5.1 Introdução

O progresso da tecnologia dos sistemas inteligentes é motivado pela necessidade do desenvolvimento de estratégias flexíveis e eficientes com o propósito de tratar aplicações complexas do mundo real. Cada metodologia da *IC* possui potencialidades e limitações que as tornam mais adequadas para algumas aplicações particulares. Por exemplo, enquanto as *RNs* são adequadas às tarefas de reconhecimento de padrões, não são eficientes para interpretação lingüística. Os *SNs* podem tratar informação imprecisa e são adequados em explanar suas decisões. Os *AEs* são métodos de otimização para o tratamento de projetos complexos, onde, usualmente, empregam-se procedimentos heurísticos do tipo tentativa e erro, mas demandam uma elevada complexidade computacional.

Estas limitações são a principal motivação da criação de sistemas híbridos inteligentes (*SHI*), onde duas (ou mais) técnicas são combinadas para superar as limitações das técnicas quando tratadas individualmente. Os *SHIs* são relevantes quando considera-se domínios complexos que apresentam problemas com componentes diferentes, os quais requerem diversos tipos de processamento (GOONTILAKE & TREVEAVEN, 1995).

A tabela 5.1 apresenta uma avaliação geral dos *AEs*, *SNs* e *RNs* com relação a algumas propriedades desejáveis de um sistema inteligente (FUKUDA & SHIBATA, 1994; GOONTILAKE & TREVEAVEN, 1995). Dependendo da metodologia inteligente, as características são específicas e não gerais, conforme apresentadas na tabela 5.1.

Tabela 5.1. Avaliação de algumas características dos sistemas inteligentes.

características	algoritmos evolutivos	sistemas nebulosos	redes neurais
modelo matemático	✓	✓✓	✓✓
aprendizado de dados	✓✓✓✓✓	✓	✓✓✓✓✓
conhecimento do operador	✓	✓✓✓	✓
processamento em tempo real	✓	✓✓✓✓	✓✓✓✓
representação do conhecimento	✓	✓✓	✓
não-linearidade	✓✓✓✓	✓✓✓✓✓	✓✓✓✓✓
otimização	✓✓✓✓✓	✓	✓✓
flexibilidade	✓✓✓✓	✓✓✓✓✓	✓✓✓✓✓
adaptação	✓✓✓✓	✓	✓✓✓✓✓
explicação	✓✓	✓✓✓	✓☐

convenções:

✓ - deficiente, ✓✓ - razoável, ✓✓✓ - boa, ✓✓✓✓ - muito boa, ✓✓✓✓✓ - excelente

Os pesquisadores da *IC* objetivam atender a estas necessidades com o desenvolvimento de sistemas que combinam as vantagens de algumas metodologias através da configuração de *SHIs*. Os *SHIs* combinando *SNs*, *RNs*, *AEs* e sistemas especialistas são eficientes em uma variedade de problemas reais (BONISSONE, 1994, CHIU *et al.*, 1995).

Os *SHIs* são relevantes quando se considera a natureza variada das aplicações. As formas dos sistemas híbridos inteligentes analisados incluem: (i) sistemas evolutivo-nebulosos; (ii) sistemas evolutivo-neurais; (iii) sistemas neuro-nebulosos; e (iv) sistemas neuro-nebuloso-evolutivos.

A utilização de *SHIs* está crescendo rapidamente com aplicações em áreas, tais como: controle de processos, projetos de engenharia, mercado financeiro, controle de qualidade, diagnóstico de falhas, avaliação de crédito, diagnóstico médico e simulação cognitiva (KHOSLA & DILLON, 1997).

As aplicações dos *SHI* estão presentes em diversas áreas. As aplicações comerciais usuais de *SHISNRN* são em análise de mercado financeiro (Nikko Securities), ventilador elétrico (Sanyo), máquina fotocopadora (Sanyo), máquina de lavar (Toshiba, Sanyo, Hitachi). Quanto aos *SHI*, existem alguns *softwares* comerciais. Por exemplo:

- (i) *ECANSE* (*Environment for Computer Aided Neural Software Engineering*) da Siemens da Áustria, para plataformas UNIX ou PC, que consiste de uma ferramenta para o desenvolvimento de *SHI*. Este *software* combina as abordagens de *RNs*, *SNs*, teoria do caos e *AGs*;

- (ii) *FlexTools*, da Flexible Intelligence Group, para ambiente MATLAB, permite a construção de sistemas inteligentes, utilizando *RNs*, *SNs* e *AGs*;
- (iii) *Neuframe*, da *NCS*, combina alguns tipos de *RNs* e *SNs*.

Outras aplicações incluem sistemas híbridos inteligentes em:

- (i) robótica (FUKUDA & SHIMOJIMA, 1995, COELHO & COELHO, 1997b, 1997c);
- (ii) projetos em *hardware* (LAZZERINI *et al.*, 1995);
- (iii) estabilizadores em sistemas de potência (SHARAF & LIE, 1994);
- (iv) controle de reatores de polimerização (NI *et al.*, 1995);
- (v) controle e otimização do armazenamento de frutas (MORIMOTO & HASHIMOTO, 1996) e diversas outras (COELHO & COELHO, 1998b, KHOSLA & DILLON, 1997).

A descrição e algumas formas de configuração destes *SHIs* são tratadas nas próximas seções.

5.2 Classificação dos sistemas híbridos inteligentes

Na literatura são mencionadas as formas de classificação para os *SHIs*. Quanto à funcionalidade, MEDSKER & BAILEY (1992) classifica-os em:

- (i) *modelo independente*: composto por módulos independentes, não existindo nenhuma interação entre os módulos. Este modelo não é uma proposta de integração propriamente dita, mas objetiva a comparação do desempenho entre as técnicas aplicadas a um problema específico;
- (ii) *modelo com transformação*: é similar ao modelo independente. Entretanto, a diferença é que este inicia o processamento com uma metodologia e termina com outra;
- (iii) *modelo com acoplamento livre*: é uma primeira forma de integração verdadeira. O problema é decomposto em sistemas separados que se comunicam através de arquivos de dados. Entre as variações deste modelo têm-se: pré-processadores, pós-processadores e co-processadores;
- (iv) *modelo com acoplamento rígido*: é similar ao modelo com acoplamento livre. Neste modelo os dados são transferidos de um sistema para outro e ficam residentes na memória;
- (v) *modelo de integração completa*: este modelo compartilha diferentes componentes, tais como: representação e estruturas de dados.

Outra classificação, quanto à funcionalidade, é de GOONATILAKE & KHEBBAL (1995) em relação a:

- (i) *substituição de funções*: está relacionada à composição funcional de uma única metodologia inteligente. A função principal de uma metodologia é substituída por

outra metodologia de processamento inteligente. Exemplo: otimização de um *SN* (ou uma *RN*) através de *AEs*;

- (ii) *intercomunicação*: são módulos independentes de processamento inteligente que trocam informações. Neste caso um problema pode ser decomposto em diversas tarefas menores, onde as metodologias são aplicadas independentemente da resolução de cada parte do problema;
- (iii) *polimorfismo*: são sistemas que utilizam uma arquitetura de processamento única para a realização da funcionalidade de metodologias de diferentes processamentos inteligentes. Exemplo: *RNs* funcionando como se estivessem realizando uma busca genética.

Sob outro ponto de vista, SCHAFFER (1994) classifica os *SHI* em duas classes:

- (i) *combinações de suporte*: um algoritmo ajuda o outro. Exemplo: preparação dos dados através de um algoritmo para ser utilizado por outro;
- (ii) *combinações colaborativas*: dois (ou mais) algoritmos cooperam para a obtenção de uma conclusão.

Quanto a forma de aplicação, a evolução dos *SHIs* pode ser classificada sob dois pontos de vista (KHOSLA & DILLON, 1997):

- (i) *aplicações do mundo real*: abordam os aspectos de estruturação e sintonia de *SHIs* aplicados a áreas de manufatura, planejamento, projeto, telecomunicações, diagnóstico, otimização, modelagem, identificação, controle e sistemas de potência;
- (ii) *processamento de informação humana*: tratam de aplicações vinculadas a psicologia cognitiva, ciência cognitiva, evolução, estruturação e comportamento do cérebro humano.

Os *SHIs*, em aplicações do mundo real, onde estão inseridos os problemas de identificação e controle de processos são apresentados neste capítulo.

5.3 Combinação de sistemas nebulosos e algoritmos evolutivos

Existe um crescente interesse na integração destes tópicos. A classificação bibliográfica de 562 referências tratando a combinação de *SNs* e *AGs* nas mais diversas áreas foi realizada por CORDÓN *et al.* (1995, 1997). As áreas de aplicação com o respectivo número de referências bibliográficas são apresentados na tabela A.7, no apêndice A (CORDÓN *et al.*, 1997).

Em muitos problemas de identificação e controle de processos reais, é possível encontrar um *SN* através de conhecimento heurístico e pela utilização de heurísticas de

tentativa e erro para ajustes no projeto. Entretanto, esta forma de projeto consiste indubitavelmente de uma tarefa tediosa, pois existem diversos parâmetros a serem sintonizados.

Os *SNs* e os *AES* têm diferentes vantagens. Por exemplo, os *SNs* são aptos para a representação de diferentes formas de conhecimento, modelam as interações e as relações entre as variáveis do sistema e são apropriados à representação de problemas de natureza qualitativa do conhecimento.

Sob outra visão, os *AES* apresentam características promissoras quanto à capacidade de aprendizado e otimização de busca global. Os *AES* são ferramentas de otimização de propósito geral e, desta forma, não requerem conhecimento de “como” resolver o problema. Geralmente deve ser dotado de apenas uma função de aptidão para que cada solução potencial possa ser avaliada. Os *SNs* e *AES* podem ser combinados de diversas formas à obtenção de diferentes tipos de *SHIs*, por exemplo:

- (i) *SHIs* *evolutivo-nebulosos*: visam a utilização de *SNs* para melhorar o comportamento ou mesmo a configuração de componentes dos *AES*. Trabalham com problemas de gerenciamento, em ambientes imprecisos, onde a imprecisão é modelada por *SNs* (STREIFEL *et al.*, 1999);
- (ii) *SHIs* *nebuloso-evolutivos* (*SHSNAE*): constitui-se da aplicação de *AES* em problemas envolvendo *SNs*.

O segundo tipo de sistema híbrido, denominado de *SHSNAE*, é tratado nesta tese, para a sintonia e a aplicação de *CLN* em controle de processos multivariáveis. Em particular, a aplicação no projeto, aprendizado e sintonia de regras têm apresentado resultados promissores. LINKENS & NYONGESA (1995) apresentam as características dos *SHIs* combinando os *SNs* e os *AGs* e relatam algumas configurações de projeto. KARR & GENTRY (1993) propõem o projeto de *CLN* baseado no ajuste das funções de pertinência e da base de regras via *AGs*. PARK *et al.* (1994) apresentam o projeto genético de *CLN* aplicado a um motor *DC*. TZES *et al.* (1998) apresentam algoritmo de *clustering* nebuloso baseado em *AGs* aplicado na identificação de um motor de corrente contínua. XIONG & LITZ (1998) tratam a otimização da base de regras de um *CLN* via *AGs*.

Alguns tópicos sobre os *SHSNAEs* são alvos de pesquisas, entre os quais:

- (i) a possibilidade de unificar diversas propostas em uma metodologia geral para o projeto de *SNs*;

- (ii) os modelos de codificação para o aprendizado de base de regras, com representação canônica ou ponto flutuante (ou mesmo combinação destas);
- (iii) a análise de eficiência computacional associada à determinação do número de regras;
- (iv) a necessidade do aprimoramento dos *AEs* visando acelerar a convergência.

5.3.1 Definição evolutiva da base de dados dos *SNs*

As regras apresentam alguns parâmetros e o ajuste de forma adequada permite a obtenção de um *CLN* eficiente e robusto. Além disso, a sintonia das funções de pertinência das regras nebulosas é uma tarefa relevante em *CLN*.

O método de sintonia evolutiva habilita a análise das funções de pertinência das regras nebulosas e seus parâmetros baseados na função de aptidão. Os diversos métodos propostos visam a definição da base de dados utilizando *AEs*, usualmente os *AGs*. A maioria deles é baseada na definição *a priori* de uma base de regras (KIM *et al.*, 1995, LINKENS & NYONGESA, 1995).

Cada indivíduo envolvido no processo de evolução representa as diferentes definições da base de dados, contendo uma codificação das funções de pertinência propiciando um valor numérico (representação inteira) aos termos lingüísticos. A figura 5.1 apresenta o sistema de definição evolutiva da base de dados nos *SNs*.

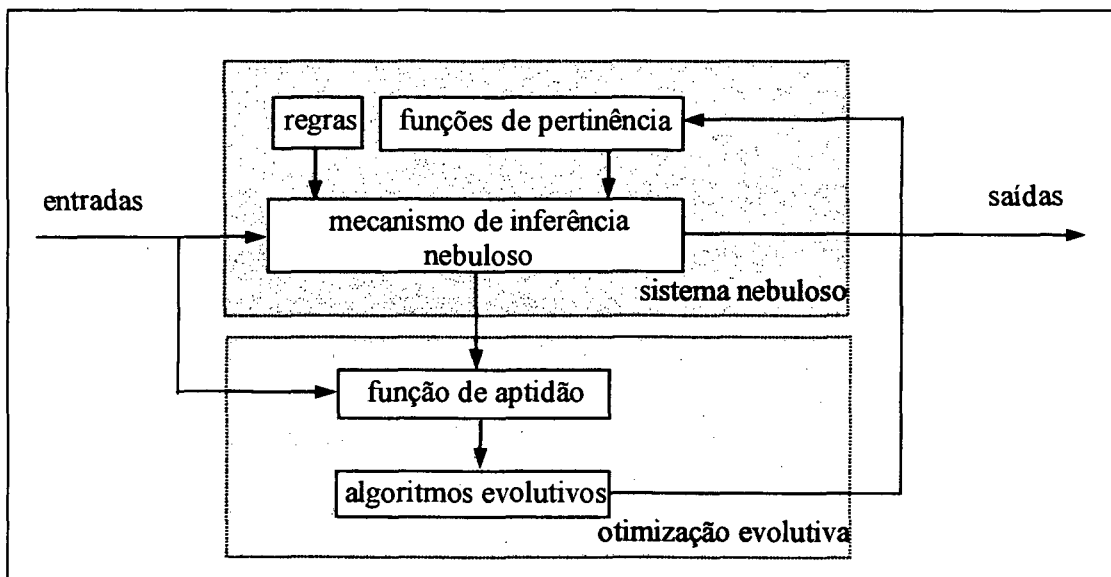


Figura 5.1. Sistema de definição evolutiva da base de dados nos *SNs*.

5.3.2 Definição evolutiva da base de regras dos *SNs*

Uma característica comum da maioria dos *SNs* consiste na representação da base de regras através de uma tabela de decisão. Uma base de regras é constituída de regras de controle apresentando variáveis de entrada e variáveis de saída. No caso de *SNs* do tipo Mamdani, a tabela de decisão contém os termos lingüísticos que formam o conjunto de valores para a variável de saída pela combinação dos antecedentes representados pelas regras. Esta estrutura é codificada pelos indivíduos que formam a população de um *AE*.

5.3.3 Aprendizado evolutivo da base de conhecimento dos *SNs*

Existem diversas formas de aplicar o aprendizado através de *AEs* em *SNs* para o projeto da base de conhecimento. Entre as características abordadas têm-se (LINKENS & NYONGESA, 1995):

- (i) o comprimento variável dos cromossomo nos *AGs*;
- (ii) as codificações com um número fixo de regras e de funções de pertinência;
- (iii) os sistemas que utilizam conhecimento do especialista para aprimoramento do procedimento de aprendizado.

5.4 Combinação de redes neurais e algoritmos evolutivos (*SHRNAE*)

Os *AEs* e as *RNs* são técnicas para o aprendizado e otimização fundamentadas em inspirações diferenciadas dos sistemas biológicos. Estas metodologias são de auto-aprendizado, mas utilizam estrutura e base matemática diferentes. As *RNs* são métodos de aprendizado indutivo, enquanto os *AEs* utilizam aprendizado dedutivo e requerem uma função de aptidão para avaliação das soluções do problema (LINKENS & NYONGESA, 1996a). O procedimento de desenvolvimento de *RNs* para uma aplicação inclui estágios de projeto, a citar:

- (i) a seleção do domínio do problema;
- (ii) a configuração da *RN* (número de unidades a serem utilizadas, conexões entre os neurônios e outros parâmetros estruturais);

- (iii) a escolha do algoritmo de treinamento da *RN*; e
- (iv) a análise das propriedades de convergência, a habilidade de resolver o problema e a capacidade de generalização (SCHIFFMANN *et al.*, 1992).

A evolução pode ser introduzida nas *RNs* em diversos níveis. Uma estrutura geral do *SHRNAE* incluindo três níveis de evolução é apresentada na figura 5.2. Existem discussões se a evolução das regras de aprendizado estariam no nível mais alto dos três níveis de evolução apresentados. Um estudo sobre as possíveis combinações é apresentado em YAO (1999). Neste artigo foram referenciados 319 artigos relevantes sobre os *SHRNAEs*.

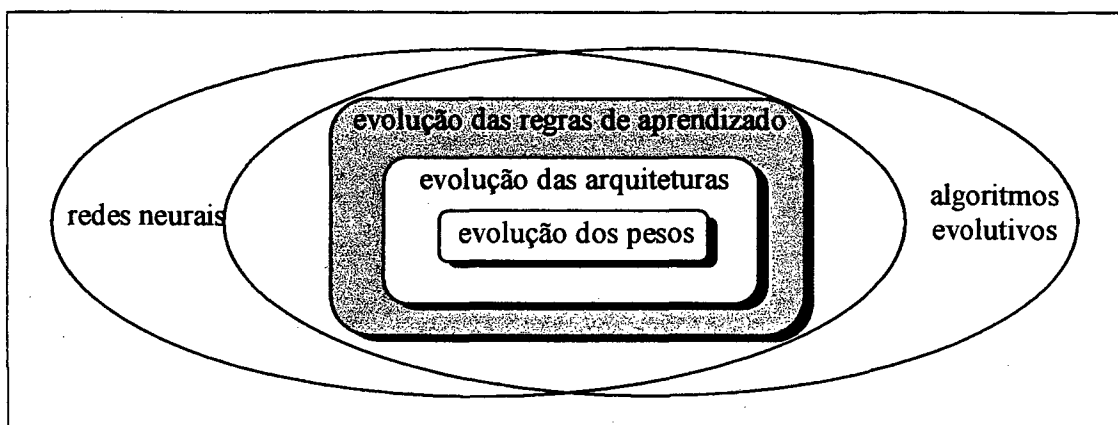


Figura 5.2. Estrutura geral para sistemas híbridos de *RNs* e *AEs*.

5.4.1 Evolução dos pesos das *RNs*

No nível inferior, a evolução pode ser introduzida no treinamento dos pesos, que são ajustados através de um ciclo evolutivo de otimização, de maneira similar ao procedimento de aprendizado nas *RNs*, onde os pesos são ajustados de forma a aprender funções.

5.4.2 Evolução das arquiteturas das *RNs*

No nível superior aos dos pesos, a evolução pode ser introduzida na adaptação da arquitetura de *RNs*. O projeto da arquitetura da *RN* é um projeto heurístico e apresenta

dificuldades para o projetista. Neste caso, visa-se um compromisso entre a complexidade da estrutura e um menor erro de treinamento da *RN*.

As metodologias para o projeto automático das arquiteturas de *RNs* têm motivado a investigação por parte de pesquisadores de algoritmos do aprendizado que aprimorem a busca de topologias apropriadas de *RNs*.

O projeto de arquiteturas é um problema desafiador de múltiplos critérios. Os *AEs* oferecem uma abordagem atraente e relativamente eficiente na busca de soluções próximas as ótimas em uma variedade de domínios de problemas. Um procedimento que pode ser adotado na abordagem evolutiva do projeto de *RNs* é apresentado na figura 5.3 (BALAKRISHNAN & HONAVAR, 1995).

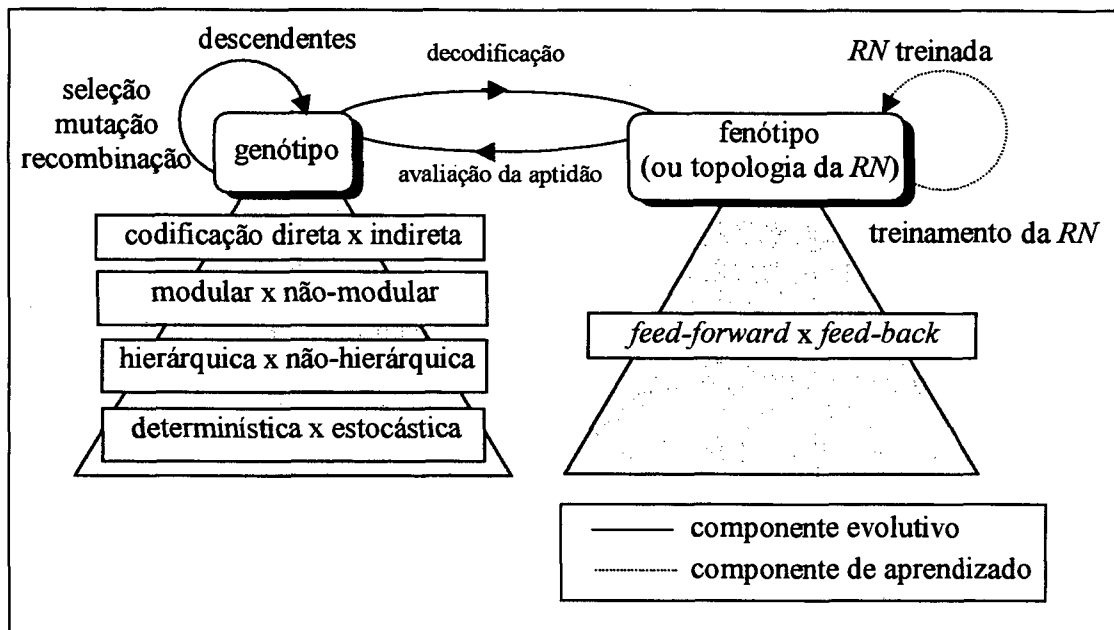


Figura 5.3. Procedimento de projeto evolutivo de *RNs*.

5.4.3 Evolução das regras de aprendizado das *RNs*

No nível superior, a evolução pode ser introduzida nas regras de aprendizado da *RN*, ou seja, as regras que especificam a forma de ajustar os pesos no treinamento da *RN*. Deste ponto de vista, o treinamento dos pesos é um procedimento de aprendizado,

enquanto a evolução das regras de aprendizado pode ser vista como um procedimento de aprendizado da tarefa de ajustar os pesos (YAO & LIU, 1996b).

Ao iniciar-se o treinamento de algumas arquiteturas de *RNs-MLP* é necessário a configuração de parâmetros de aprendizado e momento. Estes parâmetros são dependentes das características do problema a ser resolvido, e a parte crítica do procedimento de aprendizado. A busca de valores adequados para as regras de aprendizado chega a ser considerada uma “arte” (BELEW *et al.*, 1990).

5.4.4 Outras formas de configuração e aplicações

Um tópico de intensas pesquisas é o uso de *AEs* como alternativa ao treinamento de *RNs*, em substituição (ou mesmo hibridização) aos métodos baseados em gradiente. Os algoritmos de aprendizado supervisionado podem ficar retidos em mínimos locais durante o procedimento de treinamento, acarretando a obtenção de soluções sub-ótimas. Um *AE* possibilita o aprendizado de forma a escapar de mínimos locais onde os algoritmos de treinamento convergem prematuramente. Alguns estudos abordam a hibridização do *BP* com *AG* e exibem resultados promissores, quanto às propriedades de convergência, em relação ao algoritmo *BP* convencional (PACHECO & THOMÉ, 1997).

Os *SHRNAEs* demonstram eficiência em vários tipos de aplicações em engenharia, tais como: projeto de *RNs* para a classificação de padrões, controle, navegação de robôs, fusão de sensores e implementações *VLSI* de arquiteturas neurais (DAVIS, 1991, SCHIFFMANN *et al.*, 1992, BALAKRISHNAN & HONAVAR, 1995).

SAHA & CHRISTENSEN (1994) propõem um projeto de *RN* que encontra a arquitetura e os pesos da *RN* via *AGs*. KITANO (1990) discute os resultados inerentes à velocidade de convergência do treinamento de *RNs* por *AGs* e *BP*. JANSON & FRENZEL (1993) apresentam o projeto de *layout* de circuitos *VLSI* via *SHRNAE*. MANDISCHER (1993) apresenta a configuração de arquiteturas neurais através de *AGs*. BALAKRISHNAN & HONAVAR (1996) apresentam experimentos em robótica móvel com controle neuro-evolutivo.

Alguns aspectos dos *SHRNAEs* devem ser analisados para a realização de um projeto eficiente, entre os quais (KITANO, 1990, PACHECO & THOMÉ, 1997):

- (i) a dificuldade do treinamento da *RN* via *AEs*, que aumenta proporcionalmente com o número de pesos da *RN*;
- (ii) a influência acentuada da configuração dos operadores evolutivos, havendo a necessidade, muitas vezes, da criação de operadores alternativos (operadores de adição, eliminação ou substituição de pesos da *RN*) de maneira a evitar a perda de diversidade e a convergência prematura;
- (iii) diminuição da complexidade computacional dos *AEs* frente a outros métodos de otimização.

5.5 Combinação de sistemas nebulosos e redes neurais (*SHSNRN*)

As *RNs* são algoritmos paralelos, usualmente com estruturas caixa cinza ou caixa preta. Os *SNs* são também estruturas paralelas, mas apresentando estruturas paramétricas. Nas *RNs*, a extração e a representação do conhecimento são tarefas complexas. A vantagem dos *SNs* é a representação do conhecimento na forma de regras lingüísticas. Os *SNs* são aptos ao tratamento de incertezas e informações imprecisas. Entretanto, as limitações dos *SNs* são devidas, principalmente, à dificuldade da definição das funções de pertinência e da aplicação de métodos de aprendizado.

Uma das similaridades entre os *SNs* e as *RNs* é a de que ambos podem manipular as não-linearidades dos processos. A forma das funções de pertinência dos *SNs* e da função de ativação das *RNs* podem ser idênticas.

A possibilidade de integração de *SNs* e *RNs* tem ocasionado interesse e levado ao desenvolvimento crescente de metodologias combinadas, que apresentam as potencialidades de ambos (LINKENS & NYONGESA, 1996a). Entre as desvantagens dos *SNs* têm-se:

- (i) o conhecimento de especialistas é específico e difícil de generalizar;
- (ii) a sintonia manual das regras de inferência e das funções de pertinência não é uma tarefa trivial o que a leva a necessidade de um método para a sintonia das funções de pertinência pela minimização de uma função objetivo.

As *RNs* apresentam algumas limitações quanto: (i) a dificuldade de implementação direta de regras lógicas pelos especialistas; (ii) ao conhecimento adquirido que é distribuído e complexo de explicar; e (iii) a não existência de uma função simples para explicar o procedimento de inferência.

Como consequência, as *RNs* e os *SNs* podem ser integrados para combinar suas vantagens e minimizar suas deficiências individuais. Os *SNs* provêm um mecanismo de inferência sobre a incerteza cognitiva e as *RNs* oferecem vantagens, tais como: aprendizado, adaptação, tolerância a falhas, paralelismo e generalização. Em síntese, as *RNs* introduzem características computacionais ao sistema híbrido, e recebem dos *SNs* o poder de representação e inferência.

Em consequência destes aspectos, muitos pesquisadores automatizam o procedimento de configuração dos *SNs* (SUN & JANG, 1991). As tarefas podem ser divididas em duas partes: identificação da estrutura (encontrar o número adequado de regras e das partições do espaço de estados) e identificação dos parâmetros (ajustes nas funções de pertinência). Estas características e o fato que muitas arquiteturas neurais e nebulosas são aproximadores universais de sistemas motivam a combinação dos *RNs* e os *SNs* (FEURING & LIPPE, 1995).

NAUCK (1995) descreve as combinações de *SNs* e *RNs* em: redes neurais-nebulosas, modelos nebuloso-neurais concorrentes, cooperativos e híbridos. Diferentes propostas da combinação de *SNs* e *RNs* são tratadas em TING & WOLFE (1995), FÚLLER (1995) e LINKENS & NYONGESA (1996a).

Uma outra possibilidade da combinação de *SN* e *RN* é incluir um procedimento de otimização baseado em *AE*, configurando-se um sistema híbrido neuro-nebuloso-evolutivo. A literatura menciona a utilização destes sistemas em identificação e controle em NOBRE (1995), KHOSLA & DILLON (1997) e LAZZERINI *et al.* (1999).

5.6 Formas alternativas de combinar *SNs*, *RNs* e *AEs* em projetos de controle

Algumas implementações e novas formas de combinar *RNs*, *SNs* e *AEs* são propostas em aplicações de identificação e controle apresentadas nos capítulos 8 e 9, respectivamente. Na figura 5.4 são apresentadas as diversas possibilidades de projeto híbrido em identificação e controle de processos, através de *IC*.

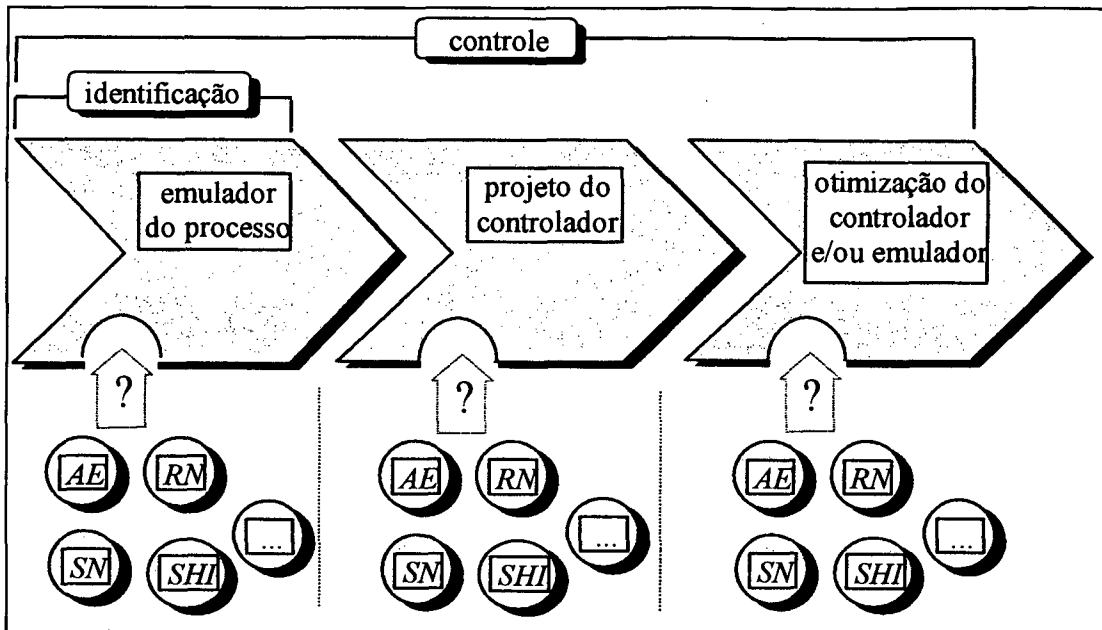


Figura 5.4. Possibilidades de projeto híbrido em identificação e controle de processos através de IC.

Os *SHIs* configurados e propostos para as aplicações em identificação de processos multivariáveis são os seguintes: *SHSNRN* do tipo Takagi-Sugeno-Kang e *SHRNAE* combinando *RN-RBF* e otimização via estratégia evolutiva. Ambos os projetos são utilizados para emular o comportamento de processos multivariáveis.

Quanto aos projetos de *SHIs*, em controle de processos multivariáveis, são propostas algumas abordagens, tais como:

- (i) combinação de *SN* e *AE*: o *CLN(PD+I)* emprega uma estratégia evolutiva para a sintonia dos fatores de escala das funções de pertinência;
- (ii) combinação de *RN* e *AE*: propõe-se um controlador do tipo *RN-RBF* com otimização dos centros e variâncias de funções Gaussianas da camada intermediária, através de um algoritmo genético híbrido com *simulated annealing*;
- (iii) combinação de *SN*, *RN* e *AE*: o projeto constitui-se de um modelo Takagi-Sugeno-Kang (emulador do processo), uma *RN-RBF* (controlador) com otimização via estratégia evolutiva.

Um projeto de controlador preditivo generalizado utilizando previsões obtidas através de modelo neural (emulador) e otimização da função custo quadrática combinando *AE* e método *simplex* também é proposto nesta tese.

5.7 Conclusão do capítulo

Neste capítulo apresentaram-se as características, motivação e projetos de *SHIs*. A meta deste capítulo foi enfatizar algumas vantagens e limitações da combinação de paradigmas da *IC*, tais como *AEs*, *SNs* e *RNs*. Os conceitos, as características e a definição de sistemas evolutivo-nebulosos, sistemas evolutivo-neurais, sistemas neuro-nebulosos e sistemas neuro-nebuloso-evolutivos foram apresentados e discutidos.

As aplicações em identificação e controle multivariáveis propostas na literatura, nos últimos anos, mostram que este campo do conhecimento emergente tem grandes perspectivas nas aplicações industriais. A combinação de técnicas, nos sistemas híbridos inteligentes, possibilita a manipulação de alguns aspectos de tomada de decisão e controle, do aprendizado do ambiente em que está inserido, de requerimentos dos usuários e de novas demandas de desempenho.

Os processos tratados, no ambiente industrial, são cada vez mais complexos motivando a utilização, no futuro, da configuração de *SHIs*, desde que estes sistemas possam adaptar-se a mudanças e prover a flexibilidade requerida em muitas aplicações.

Capítulo 6

CONTROLE PREDITIVO: ABORDAGEM ADAPTATIVA E INTELIGENTE

6.1 Introdução

A utilização de controladores multivariáveis não é recente na indústria e tem diversas aplicações, tais como: refinarias petrolíferas, petroquímicas e farmacêuticas, processamento de gás natural e indústria aeroespacial (VAN DOREN, 1997). Nestes tipos de processos complexos que, muitas vezes apresentam acoplamento entre as variáveis, a utilização de controle *PID* convencional apresenta, geralmente, desempenho inadequado do sistema em malha fechada.

A intensa competição de mercado internacional, que caracterizou a década de 1990, tem aumentado a necessidade dos processos industriais apresentarem o máximo de capacidade e eficiência. A competição motiva a introdução de novas ferramentas para o controle de processos. As diferentes técnicas propostas visam responder da melhor forma frente às necessidades do mercado. Atualmente, é plausível declarar que os algoritmos de controle, baseados em modelo, são uma poderosa ferramenta em controle avançado de processos. Alguns exemplos são apresentados em BONAVITA & TOMASI (1998) e COELHO *et al.* (1997f, 1999c, 1999d).

Para algumas aplicações industriais, a metodologia avançada adotada é o controle preditivo baseado em modelo (*MBPC*). O *MBPC* é tipicamente implementado através de um *software*, componente de um sistema de controle distribuído. Diversos ambientes computacionais com a tecnologia do *MBPC* estão disponíveis comercialmente (VAN DOREN, 1996, 1997, QIN & BADGWELL, 1997, 1998, BONAVITA & TOMASI, 1998). Os métodos de *MBPC* referem-se a uma coleção de formulações de projeto de controle, que possuem um critério de controle descrito de forma que considere as saídas

futuras do processo e, algumas vezes, as entradas futuras em um especificado instante de tempo (BITMEAD *et al.*, 1990).

O primeiro propósito deste capítulo é apresentar um breve histórico do desenvolvimento da tecnologia dos *MBPCs*. Uma visão geral do estado da arte dos *MBPCs* e as pesquisas acadêmicas quanto a processos multivariáveis são apresentadas a seguir. Nas seções seguintes do capítulo, as formulações do controlador preditivo generalizado (*GPC*), na concepção adaptativa e baseado em metodologias da *IC*, são descritas.

6.2 Breve histórico

O termo *MBPC* descreve uma classe de algoritmos computacionais de controle que controla o comportamento futuro, através de modelo explícito do processo. A cada amostra, o algoritmo *MBPC* calcula uma seqüência de ajustes das variáveis manipuladas, visando otimizar o comportamento futuro do processo. A primeira entrada de uma seqüência ótima é inserida no processo e um procedimento de otimização é repetido nos intervalos subseqüentes de controle.

Quanto ao histórico dos métodos preditivos em controle adaptativo, a literatura tem mencionado vários algoritmos. Exemplos: *IDCOM* (*IDentification and COMmand*), *DMC* (*Dynamic Matrix Control*), *Predictor-based Self-Tuning Control*, *EHAC* (*Extended Horizon Adaptive Control*), *MAC* (*Model Algorithmic Control*), *MUSMAR* (*Multistep Multivariable Adaptive Control*), *MUSHAC* (*Multipredictor Receding Horizon Adaptive Control*), *PFC* (*Predictive Functional Control*), *UPC* (*Unified Predictive Control*) e *EPSAC* (*Extended Prediction Self-Adaptive Control*) (MCINTOSH *et al.*, 1991).

6.3 Motivação à utilização de *MPBC*

O sucesso da tecnologia dos *MPBCs* como paradigma de controle de processos pode ser atribuída a três importantes fatores. O primeiro fator é a incorporação de um

modelo explícito do processo no cálculo da lei de controle, permitindo ao controlador, em princípio, tratar diretamente com as características da dinâmica do processo.

O segundo fator é que os algoritmos de *MPBC* consideram o comportamento do processo sobre um horizonte futuro no tempo. Isto significa que os efeitos de perturbações *feedforward* e *feedback* podem ser antecipados e removidos, permitindo ao controlador tratar a dinâmica do processo de forma consistente ao longo da trajetória desejada.

Outra característica de projeto dos *MBPCs* é a possibilidade de considerar o problema com restrições de entrada, estados e saída do processo diretamente no cálculo da lei de controle.

Os *MPBCs* apresentam uma série de vantagens sobre outros métodos de projeto de controle. O *MPBC* é particularmente atraente no tratamento de processos complexos, com características de atraso de transporte, fase não-mínima, instabilidade em malha aberta e apresentando restrições e incertezas (CAMACHO, 1998). Além disso, o algoritmo *MPBC* pode resolver os problemas de controle multivariável não linear de maneira eficiente (DE KEYSER, 1998).

6.4 Aplicações industriais

A tecnologia do *MBPC* foi originalmente desenvolvida para aplicações em sistemas de potência e refinarias petrolíferas. Atualmente, podem ser encontrados *MBPCs* em uma variedade de ambientes de manufatura, incluindo-se químicos, de processamento de alimentos, automotivos, aeroespacial, metalúrgico e de fabricação de papel (BONAVITA & TOMASI, 1998, VAN DOREN, 1997, QIN & BADGWELL, 1997, 1998).

Os processos de manufatura são inerentemente não lineares e a grande maioria das aplicações de *MBPC* é baseada em modelos dinâmicos lineares, os mais comuns sendo os modelos de resposta ao degrau e ao impulso. Existem diversos motivos para isto. Os modelos lineares empíricos podem ser identificados de maneira simples utilizando-se as medidas do processo.

Em muitos casos, um modelo linear provê a maioria dos benefícios possíveis para os *MBPCs*. Entretanto, existem casos onde os efeitos não lineares são significativos, o

suficiente, para justificar a não utilização da tecnologia dos *MPBC* com modelos lineares. Estes incluem, principalmente, dois tipos de aplicações:

- (i) *problemas de controle regulatório*: o processo é altamente não linear e sujeito a perturbações de altas frequências. Exemplo: controle de pH;
- (ii) *problemas de controle servo*: os pontos de operação mudam freqüentemente e a variação é suficientemente larga para as dinâmicas de processos lineares. Exemplos: manufatura de polímeros e síntese de amônia.

Em QIN & BADGWELL (1997) são apresentadas 2233 aplicações comerciais da tecnologia dos *MBPCs*. Quase todas as aplicações são implementadas com modelos lineares e são acopladas em processos de refinarias e indústria petroquímica. Entretanto, em QIN & BADGWELL (1998) são apresentadas cerca de 86 aplicações de *MBPC* baseados em modelos não lineares. Nas tabelas A.8 e A.9 são apresentadas uma síntese do estado da arte de aplicações de *MBPC* lineares e não lineares, respectivamente.

6.5 Identificação de processos pelo algoritmo dos mínimos quadrados recursivo

A aplicação de algoritmos de identificação é interessante para vários propósitos, tais como: supervisão, rastreamento de parâmetros variantes no tempo para controle adaptativo, filtragem, previsão, processamento de sinais, detecção e diagnóstico (JOHANSSON, 1993). Entretanto, diversos métodos de identificação são inadequados para aplicações em tempo real. Portanto, é desejável ter-se uma formulação adequada dos algoritmos para obter-se procedimentos eficientes.

No esquema de *MBPC* adaptativo, o estimador de parâmetros deve ser iterativo, onde o modelo do sistema é atualizado, a cada período de amostragem, quando novas medidas estão disponíveis. Nesta tese é utilizado o algoritmo dos mínimos quadrados recursivo (*MQR*). A conceituação teórica, as equações e os aspectos de configuração do *MQR* multivariável, apresentados a seguir são baseados em AMARAL (1980) e TOKARNIA (1984).

A estimação recursiva consiste na correção, a cada instante, do estimador obtido no instante anterior, através de um mecanismo de ajuste. Um esquema comum de identificação recursiva é:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)\varepsilon(k+1) \quad (6.1)$$

onde $\hat{\theta}(k)$ é o vetor dos parâmetros estimados, no instante k , $\varepsilon(k+1)$ é o vetor de erro entre a(s) saída(s) do processo e a(s) saída(s) do modelo, $K(k+1)$ é uma matriz, ajustada a cada instante, conforme as medidas realizadas. Pela análise do critério, J_1 , descrito por:

$$J_1 = \frac{1}{2} \sum_{i=1}^N \varepsilon^T V_i \varepsilon_i \quad (6.2)$$

onde V_i é a variância condicional, observa-se que o critério é quadrático quanto aos parâmetros desconhecidos, sua derivada é linear nestes parâmetros e uma equação explícita para o estimador pode ser obtida. Para obter estas condições, o erro de previsão deve ser uma função linear dos parâmetros.

Um modelo com as características mencionadas é empregado pelo estimador dos mínimos quadrados. Este modelo, muitas vezes, é referido como modelo dos mínimos quadrados e descrito por:

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + W_{BR}(k) \quad (6.3)$$

$$A(z^{-1}) = I + A_1 z^{-1} + \dots + A_{nA} z^{-nA} \quad (6.4)$$

$$B(z^{-1}) = B_0 + B_1 z^{-1} + \dots + B_{nB} z^{-nB} \quad (6.5)$$

onde $y(k) \in \mathbb{R}^n$ é o vetor de saída(s), $u(k) \in \mathbb{R}^m$ é o vetor de entrada(s), $W_{BR}(k)$ é uma seqüência branca de variáveis aleatórias com matriz de covariância V (constante), nA e nB são parâmetros inteiros que indicam a ordem das equações e, portanto, podem assumir apenas valores inteiros e positivos.

O modelo descrito pela equação (6.3) também pode ser reescrito de forma a agrupar num vetor, $\hat{\theta}$, os coeficientes das matrizes $A(z^{-1})$ e $B(z^{-1})$ a serem estimadas, obtendo-se:

$$y(k) = \psi(k)\theta(k) + W_{BR}(k) \quad (6.6)$$

onde $\psi(k) \in \mathbb{R}^{nnp}$ é um bloco diagonal definido por $[\phi(k), \dots, \phi(k)]$ com

$$\phi(k) = [-y(k-1) \dots -y(k-nA) \ u(k-1) \dots u(k-nB)]^T, \text{ tal que } \phi(k) \in \mathbb{R}^p, \text{ e} \quad (6.7)$$

$$\theta(k) = [A_1 \ A_2 \ \dots \ A_{nA} \ B_0 \ B_1 \ B_2 \ \dots \ B_{nB}]^T, \text{ tal que } \theta^T \in \mathbb{R}^m \quad (6.8)$$

onde A_i é a i -ésima linha de $[A(z^{-1}) - I]$, B_i é a i -ésima linha de $B(z^{-1})$. Utilizando-se o modelo (6.6), a equação do erro de previsão é:

$$\varepsilon(k) = y(k) - \psi(k) \hat{\theta}(k) \quad (6.9)$$

Minimizando o critério J_1 com o valor de $\varepsilon(k)$, dado pela equação (6.9), obtém-se o estimador de mínimos quadrados:

$$\hat{\theta}_i(N) = \left[\sum_{k=1}^N \psi^T(k) V^{-1} \psi(k) \right]^{-1} \left[\sum_{k=1}^N \psi^T(k) V^{-1} y(k) \right] \quad (6.10)$$

É possível estimar separadamente os parâmetros de cada linha do modelo (6.6). Esta prova é apresentada em TOKARNIA (1984) e é realizada através de propriedades do produto de Kronecker de matrizes, de onde obtém-se:

$$\hat{\theta}_i(N) = \left[\sum_{k=1}^N \phi(k) \phi^T(k) \right]^{-1} \left[\sum_{k=1}^N \phi(k) y_i(k) \right], \quad i=1, \dots, n \quad (6.11)$$

onde $y_i(k)$ é a i -ésima componente do vetor-saída e o sub-índice N indica o número de observações utilizadas no cálculo do estimador. A matriz de covariância do estimador de mínimos quadrados é regida por:

$$\text{Cov}(\hat{\theta}) = \left[\sum_{k=1}^N \psi^T(k) \psi(k) \right]^{-1} \left[\sum_{k=1}^N \psi(k)^T V \psi(k) \right] \left[\sum_{k=1}^N \psi^T(k) \psi(k) \right]^{-1} \quad (6.12)$$

Uma observação importante sobre este algoritmo é que, para um sistema descrito pelo modelo (6.3) com $W_{BR}(k)$ descrevendo uma seqüência branca e estacionária de

covariância V , o estimador dos mínimos quadrados é não polarizado e tem variância mínima entre os estimadores lineares não-polarizados. Entretanto, quando o ruído presente não é uma seqüência branca, o estimador de mínimos quadrados não converge para os valores verdadeiros dos parâmetros. Este problema é resolvido modelando-se as perturbações a partir de uma seqüência branca (LJUNG, 1987; WELLSTEAD & ZARROP, 1991).

Para obter a versão recursiva — *MQR* — considera-se a situação em que mais uma observação, entrada-saída, é realizada no instante $k+1$. Definindo-se a matriz de covariância, $P(k+1)$, de forma que:

$$P(k+1) = [P^{-1}(k) + \psi^T(k+1)\psi(k+1)]^{-1} \quad (6.13)$$

e aplicando-se a esta equação o lema de inversão matricial (WELLSTEAD & ZARROP, 1991, JOHANSSON, 1993), tem-se:

$$P(k+1) = P(k) - P(k)\psi^T(k+1)[I + \psi(k+1)P(k)\psi^T(k+1)]^{-1}\psi(k+1)P(k) \quad (6.14)$$

e, substituindo-se a equação (6.14) na equação (6.10), obtém-se:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k+1)[y(k+1) - \psi(k+1)\hat{\theta}(k)] \quad (6.15)$$

onde

$$K(k+1) = P(k+1)\psi^T(k+1) \quad (6.16)$$

As equações (6.14) a (6.16) constituem o estimador do *MQR*. A forma recursiva da equação (6.11) que calcula os valores dos parâmetros estimados de cada linha do modelo (6.3) do estimador dos *MQR* é regida por:

$$P_i(k+1) = P_i(k) - \frac{P_i(k)\phi(k+1)\phi^T(k+1)P_i(k)}{1 + \phi^T(k+1)P_i(k)\phi(k+1)} \quad (6.17)$$

e de acordo com o erro de previsão,

$$\varepsilon_i(k+1) = y_i(k+1) - \phi^T(k+1) \hat{\theta}_i(k) \quad (6.18)$$

tem-se o vetor de parâmetros estimados, $\hat{\theta}(k+1)$, calculado por

$$\hat{\theta}_i(k+1) = \hat{\theta}_i(k) + K_i(k+1) \varepsilon(k+1) \quad (6.19)$$

e o ganho do estimador por

$$K_i(k+1) = P_i(k+1) \phi^T(k+1) \quad (6.20)$$

Esta forma simplifica os cálculos ao substituir a inversão matricial por uma divisão escalar.

6.5.1 Procedimento de aplicação dos *MQR*

O procedimento à aplicação do algoritmo dos *MQR* pode ser resumido nos seguintes passos:

- (i) obter a(s) saída(s) e a(s) entrada(s) do processo a ser identificado;
- (ii) atualizar o vetor de medidas;
- (iii) calcular o erro de previsão;
- (iv) calcular o ganho do estimador;
- (v) atualizar o vetor de parâmetros estimados;
- (vi) atualizar a matriz de covariância;
- (vii) retornar para o passo (i).

6.6 Controlador preditivo generalizado

A definição de um problema de otimização para o *MPBC* apresenta um esquema funcional que pode ser resumido na figura 6.1 (CAMACHO & BORDONS, 1995).

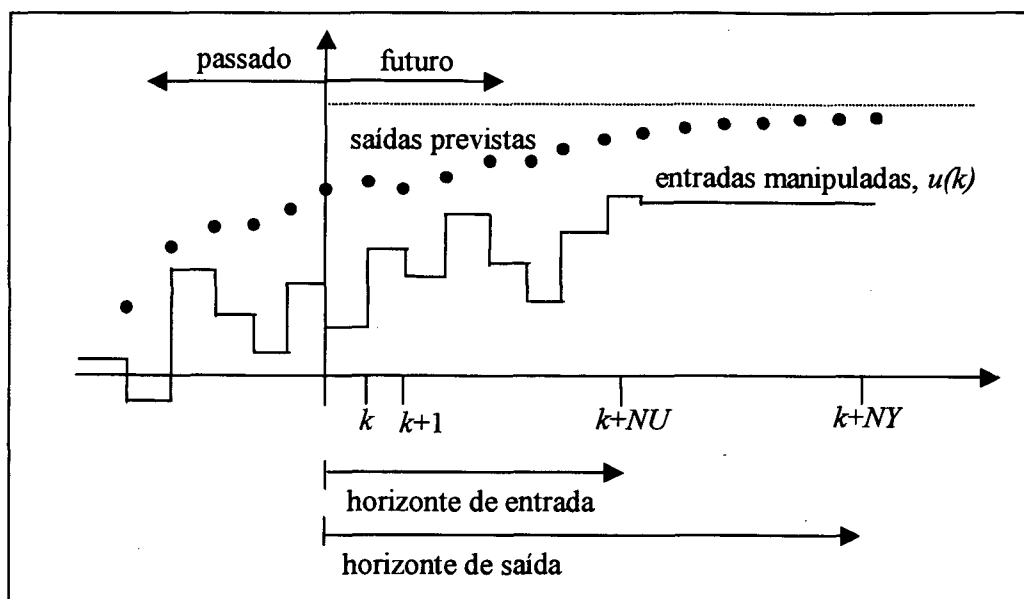


Figura 6.1. Definição do problema de otimização para o MBPC.

O *GPC* constitui-se de uma generalização dos algoritmos preditivos de múltiplos passos (*multi-step-ahead*). Além disso, o *GPC* é também uma extensão natural com horizonte estendido do controle *single-step-ahead*, denominado controlador de variância mínima generalizada (*GMV*), proposto por CLARKE & GAWTHROP (1979). O *GPC* realiza a previsão da saída do processo em um horizonte de tempo maior que o atraso de transporte do processo. O sinal de controle é calculado pela minimização de uma função custo quadrática obtida da diferença entre a saída prevista e o sinal de referência, com restrições sobre as ações de controle futuras. A maioria das estratégias de controle preditivo é basicamente uma combinação de (COELHO & COELHO, 2000c):

- (i) um modelo, que descreve o processo, necessário a previsão das saídas futuras do processo;
- (ii) uma função objetivo (custo), definida, usualmente, por uma função quadrática, considerando o controle e os erros futuros da saída do processo. Esta função é calculada para um número fixo de passos a frente para prever o comportamento do processo;
- (iii) uma trajetória conhecida das saídas de referências futuras;
- (iv) restrições do sistema e variáveis de controle (opcional);
- (v) um algoritmo de otimização para a escolha da seqüência de controle que minimiza a função custo.

No presente estudo, considera-se o *GPC* com a formulação incremental de maneira a evitar o erro em regime, muitas vezes, presente na formulação posicional.

6.6.1 *GPC* multivariável

A seguir, são apresentados aspectos de projeto do *GPC* multivariável, baseados em COELHO (1991), COELHO & AMARAL (1993) e CLARKE & MOHTADI (1987).

6.6.1.1 Aspectos teóricos

Nesta seção aborda-se o *GPC* para sistemas lineares discretos multivariáveis caracterizados por uma representação *CARIMA* (*Controlled Auto-Regressive Integrated Moving Average*) (Coelho, 1991). O procedimento que utiliza o modelo *CARIMA* em vez do *CARMA* (*Controlled Auto-Regressive Moving Average*) justifica-se para a eliminação do erro em regime. O modelo discreto linear *MIMO* é descrito pela seguinte equação:

$$A(z^{-1}) \nabla(z^{-1})y(k) = B(z^{-1}) \nabla u(z^{-1}) + C(z^{-1})\zeta(k) \quad (6.21)$$

onde $y(k)$ é um vetor ($n \times 1$) das saídas medidas, $u(k)$ é um vetor ($m \times 1$) das entradas de controle, $\zeta(k)$ é um vetor ($n \times 1$) e sendo uma seqüência descorrelacionada de variáveis aleatórias com média zero e covariância, $E\{\zeta(k)\zeta^T(k)\} = r_\zeta$. As matrizes polinomiais $A(z^{-1}) \in \mathbb{R}^{n \times n}$, $B(z^{-1}) \in \mathbb{R}^{n \times m}$, $C(z^{-1}) \in \mathbb{R}^{n \times m}$ são expressas em termos do operador atraso z^{-1} e definidas por

$$\nabla(z^{-1}) = (1 - z^{-1})I \quad (6.22)$$

$$A(z^{-1}) = I + A_1 z^{-1} + \dots + A_{n_A} z^{-n_A} \quad (6.23)$$

$$B(z^{-1}) = B_0 + B_1 z^{-1} + \dots + B_{n_B} z^{-n_B} \quad (6.24)$$

$$C(z^{-1}) = I + C_1 z^{-1} + \dots + C_{n_C} z^{-n_C} \quad (6.25)$$

As ordens das matrizes polinomiais $A(z^{-1})$, $B(z^{-1})$ e $C(z^{-1})$ são n_A , n_B , e n_C , respectivamente. Para os sistemas discretos multivariáveis, a matriz *interactor*, $\xi(z)$, representa uma estrutura de atraso de transporte, análoga ao caso monovariável (SHAH *et al.*, 1987). No caso geral, a matriz *interactor* tem forma triangular inferior e satisfaz a relação:

$$\lim_{z \rightarrow \infty} z^{-1} \xi(z) A^{-1}(z^{-1}) B(z^{-1}) = K \quad (6.26)$$

para algum finito e não-singular K . Para o cálculo do preditor j -passos a frente, o sinal $y_f(k)$ é definido como:

$$y_f(k) = \Delta(z^{-1}) \xi(z) y(k-d) \quad (6.27)$$

onde d é a maior potência em $\xi(z)$ e $\Delta(z^{-1})$ é uma matriz ajustada pelo projetista para obter o preditor j -passos a frente, $y_f(k+j/k)$ que pode ser decomposto em (COELHO & AMARAL, 1993):

$$\hat{y}_f(k+j/k) = \hat{y}_{fa}(k+j/k) + \hat{y}_{fp}(k+j/k) \quad (6.28)$$

onde

$$\hat{y}_{fa}(k+j/k) = \bar{H}_j(z^{-1}) \nabla u(k-1) - \alpha_j(z^{-1}) y(k) \quad (6.29)$$

$$\hat{y}_{fp}(k+j/k) = \bar{H}_j(z^{-1}) \nabla u(k+j-d) \quad (6.30)$$

$$\alpha_j(z^{-1}) = G_j(z^{-1}) C^{*-1}(z^{-1}) \quad (6.31)$$

$$\beta(z^{-1}) = \hat{H}_j(z^{-1}) + z^{-j+d-1} \bar{H}(z^{-1}) \quad (6.32)$$

$$\Delta(z^{-1}) \xi C^*(z^{-1}) = F_j(z^{-1}) A^*(z^{-1}) \nabla + z^{d-j} G_j(z^{-1}) \quad (6.33)$$

$$\bar{H}(z^{-1}) = H_0 + H_1 z^{-1} + \dots + H_{j-d} z^{-j+d} \quad (6.34)$$

$$H = \begin{bmatrix} H_0 & \dots & 0 \\ H_1 & H_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ H_{NY-1} & \dots & \dots & H_{NY-NU} \end{bmatrix} \quad (6.35)$$

onde H_i são elementos da matriz $\bar{H}(z^{-1})$. Considera-se a seguinte função custo para o GPC multivariável:

$$J = E \left\{ \left[\sum_{j=d}^{d+NY-1} \|e_y(k+j)\|^2 + \sum_{j=0}^{NU-1} \|e_u(k+j)\|^2 \Gamma_j \right] / I_k \right\} \quad (6.36)$$

onde

$$e_y(k+j) = \Lambda(z^{-1})\xi(z)y(k+j-d) - \Lambda(z^{-1})\xi(z)y_r(k+j-d) \quad (6.37)$$

$$e_u(k+j) = \nabla u(k+j) \quad (6.38)$$

$$\Gamma = \text{diag} [\Gamma_k] \quad (6.39)$$

sendo $E(\cdot)$ o operador esperança matemática condicionada ao conjunto, I_k , de dados disponíveis até o instante k ; Γ e $\Lambda(z^{-1})$ são matrizes polinomiais de ponderação ao controle e a saída de referência; $y_r(k+j-d)$ é a saída de referência j passos a frente, NY é o horizonte de saída e NU é o horizonte de entrada (COELHO, 1991). A lei de controle aplicada que minimiza a equação (6.36) com respeito a $E_u(k)$ é regida por (COELHO & AMARAL, 1993):

$$E_u(k) = \left[H^T H + \Gamma \right]^{-1} H^T \left[\Lambda(z^{-1})\xi(z)Y_r - Y_{\hat{F}A} \right] \quad (6.40)$$

onde

$$E_u(k) = [\nabla u^T(k) \dots \nabla u^T(k+NU+1)]^T \quad (6.41)$$

$$Y_{\hat{F}A} = [\hat{y}_a^T(k+d/k) \dots \hat{y}_a^T(k+d+NY-1)]^T \quad (6.42)$$

$$Y_r = [y_r^T(k) \dots y_r^T(k+NY+1)]^T \quad (6.43)$$

O algoritmo *GPC* multivariável é implementado de forma que, a cada amostra, um novo vetor, $E_s(k)$, é calculado utilizando a equação (6.40), mas somente a primeiro vetor de controle, $\forall u(k)$, para cada entrada é aplicado ao processo, isto é,

$$u(k) = u(k-1) + \nabla u(k) \quad (6.44)$$

Alguns aspectos importantes no projeto do *GPC* são relativos à escolha dos parâmetros NY , NU e Γ . O parâmetro NY é selecionado para englobar a resposta do processo que é efetivamente afetada pelo controle atual. É razoável que seja pelo menos do grau da matriz polinomial $B(z^{-1})$. O NY é, tipicamente, selecionado como o tempo de subida do processo (RANI & UNBEHAUEN, 1997). A escolha de NU reflete a complexidade do processo controlado. Para processos bem comportados, estáveis e de ordem baixa, NU pode assumir valor 1, enquanto para os processos complexos NU deve ser igual ou maior que 2. A determinação da ponderação Γ é usualmente heurística e depende da necessidade do projeto e das características dinâmicas do processo a ser controlado.

6.7 *GPC* com abordagem adaptativa

O projeto de um controlador preditivo adaptativo envolve os seguintes passos:

- (i) escolha da lei de controle contendo os parâmetros variáveis do sistema;
- (ii) escolha de uma lei de adaptação para o ajuste destes parâmetros;
- (iii) análise das propriedades de convergência para o sistema de controle resultante.

O projeto de controle adaptativo pode ser de duas formas, quanto ao cálculo da lei de controle: forma indireta e forma direta. No controlador adaptativo indireto, os

parâmetros do processo são estimados para o projeto da lei de controle. No direto, os parâmetros do controlador são estimados diretamente.

O projeto do *GPC* adaptativo adotado tem forma indireta, utilizando o estimador dos *MQR* descrito na seção 6.5. Na figura 6.2 é apresentada a configuração do *GPC* adaptativo com utilização do estimador *on-line* do regressor de parâmetros através do *MQR*.

A análise das propriedades de convergência para o sistema de controle resultante é um aspecto relevante no projeto de um controlador preditivo adaptativo. Vários fatores podem influenciar na convergência do sistema, em malha fechada, tais como:

- (i) conhecimento (ou mesmo adoção de uma metodologia para determinação) da ordem, atraso de transporte, acoplamentos e dinâmicas não lineares do processo;
- (ii) propriedades do ruído presente no sistema (se o ruído é branco ou colorido);
- (iii) inicialização dos vários parâmetros de projeto do algoritmo de estimação de parâmetros e do controlador preditivo. O comportamento servo e regulatório do sistema são fontes essenciais de informação para ajustes nos parâmetros do estimador, para se efetuar novos testes com outros valores de projeto do controlador e para se verificar o modelo do processo.

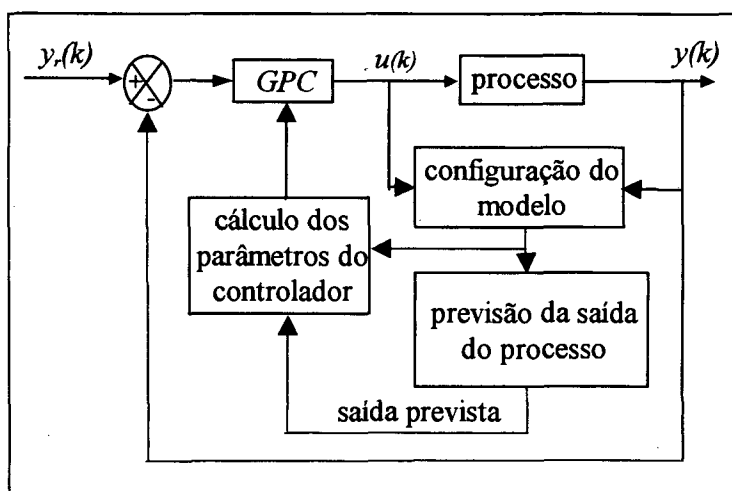


Figura 6.2. Configuração de *GPC* adaptativo.

6.8 *GPC* híbrido com inteligência computacional

Os *MBPC*, do qual o *GPC* faz parte, é uma metodologia aberta baseada em princípios que permitem extensões, contribuições, alternativas e melhoramentos de

diferentes formulações (DE KEYSER, 1998). Desde que a grande maioria de técnicas de *MBPC* é baseada em modelos lineares, o desempenho destas metodologias é afetado por erros de identificação, causados principalmente por incertezas e não-linearidades no processo. A falta de precisão dos modelos dificulta o projeto de *MBPC* lineares, principalmente no controle de processos químicos, onde as aproximações lineares resultam em perda de desempenho e limites quanto aos pontos de operação de *MBPCs*.

Entretanto, alguns esforços vem sendo realizados nos últimos anos no desenvolvimento dos *MBPCs* não lineares (MARCHI, 1999, NORQUAY *et al.*, 1999, CAMACHO & BORDONS, 1995), que frequentemente requerem técnicas de identificação não lineares. Por esta razão, alguns esforços de pesquisa são realizados quanto à aplicação de *IC* para *MBPC*, que surgem com potencial para a identificação de processos não lineares complexos. Nas próximas subseções, apresentam-se algumas propostas de estruturação e projeto de controle preditivo combinado a *IC*.

6.8.1 *GPC* neural

Recentemente, diversas abordagens apresentadas na literatura têm relatado as formas de controle preditivo baseadas em *RNs*. SAINT-DONAY *et al.* (1991) tratam o projeto de *RN-MLP* em um esquema de otimização do *MPBC* aplicado no controle de um processo de pH. EATON *et al.* (1994) apresentam estudos teóricos e analisam as questões de estabilidade de um *MBPC* neural, aplicado a processo não linear. TRAJANOSKI *et al.* (1997) desenvolvem e avaliam uma estratégia de controle preditivo neural aplicado na área de engenharia biomédica. KARAMA *et al.* (1998) apresentam o projeto de um controlador preditivo neural não linear, com treinamento realizado pelo algoritmo de otimização de Levenberg-Marquardt e aplicado ao controle de um bioreator.

Os esquemas de *GPC* neural e controle preditivo baseados em *RNs* têm, em geral, alguns atributos que o fazem uma estratégia atraente à adoção, em comparação com outros esquemas de controle neural. O controle preditivo neural não requer, usualmente, o treinamento de *RNs* em adição a um modelo *forward*, conforme requerido do controle por

modelo interno. Em esquemas de controle preditivo neural, o modelo neural *forward* do processo é configurado para atuar em paralelo com o processo (figura 6.3).

A maior vantagem do controle preditivo neural é que a entrada do processo é calculada de forma ótima para considerar as previsões futuras da resposta do processo para um horizonte de saída. Esta característica habilita o esquema a antecipar as saídas futuras do processo baseada em modelo neural, sendo este um modelo inerentemente não linear.

Outra forma possível de controle preditivo neural utiliza a saída da rotina de otimização para treinar um controlador neural, que apresenta como saída o sinal de controle ao processo, conforme apresentado na figura 6.4.

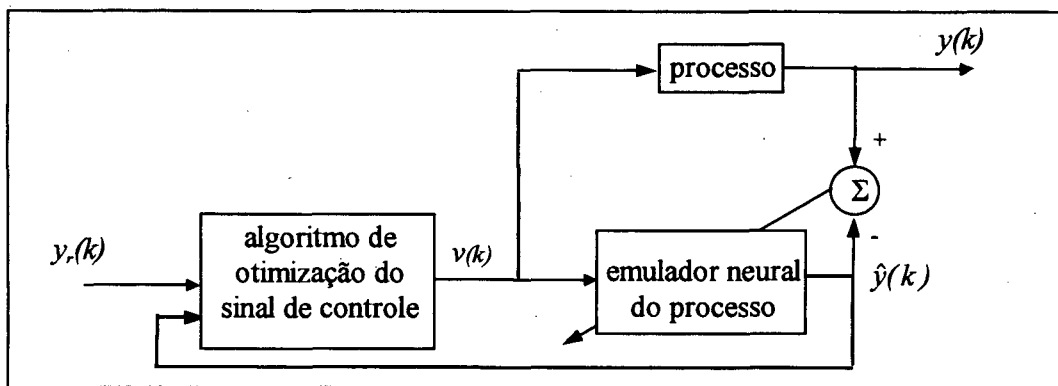


Figura 6.3. Controle preditivo neural (tipo I).

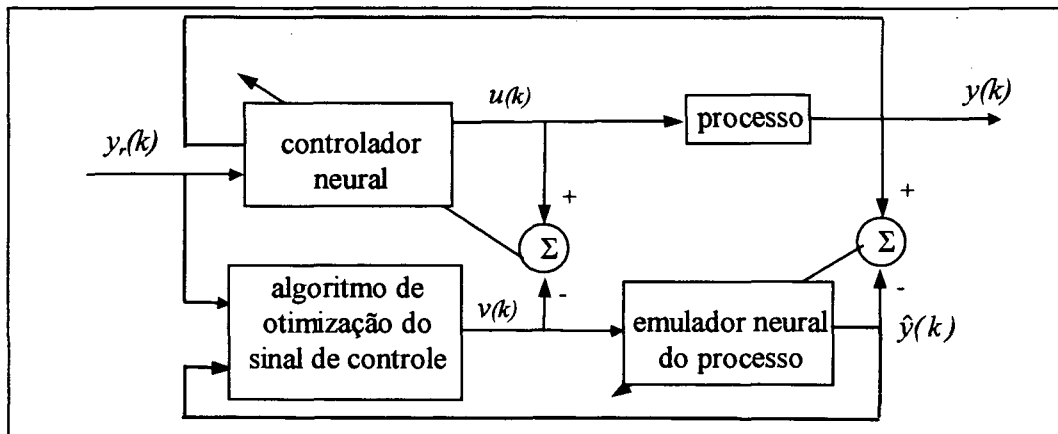


Figura 6.4. Controle preditivo neural (tipo II).

6.8.2 *GPC* nebuloso

O crescente interesse na análise e utilização de *MBPC*, baseados em modelos nebulosos, é motivada pelos *SNs* serem aproximadores universais (BUCKLEY, 1992). Os *SNs* proporcionam um mapeamento não linear adequado e genérico entre as variáveis de entrada e de saída na identificação de processos complexos (NAZZETTA *et al.*, 1998).

Os esquemas de *MBPC* nebuloso, usualmente utilizados, são similares aos apresentados nas figuras 6.3 e 6.4, para o *GPC* neural, substituindo-se as estruturas baseadas em *RNs* por *SNs*. Alguns exemplos de *MBPC* nebulosos são:

SKRJANC & MATKO (1994) propõem um algoritmo de controle preditivo baseado em *SNs*, com um mecanismo de adaptação de ganhos e o ajuste de fatores de escala das funções de pertinência. SOUZA *et al.* (1997) apresentam um método de projeto de *MBPC* baseado em modelo nebuloso do tipo *TSK* e aplicam a metodologia, com sucesso, em controle de temperatura de um sistema de ar condicionado. NAZZETTA *et al.* (1998) desenvolvem um *MBPC* baseado em modelo nebuloso relacional, utilizando um modelo nebuloso para calcular a previsão da saída do processo e na obtenção da lei de controle. OLIVEIRA & LEMOS (1997) apresentaram estudos de modelos relacionais nebulosos relevantes para controle adaptativo.

Os algoritmos de identificação nebulosos em *MBPC* são uma alternativa às técnicas de identificação convencionais. A motivação para a utilização de identificação nebulosa é a possibilidade de identificação de processos não lineares com ordem e estruturas desconhecidas (WANG, 1994).

6.8.3 *GPC* evolutivo

Os esquemas de *MBPC* envolvem especificações de parâmetros, tais como: horizontes de previsão e de controle. As pesquisas, apresentadas na literatura, sobre a seleção adequada de ponderações são escassas e arbitrárias por natureza (MCINTOSH *et al.*, 1991; RANI & UNBEHAUEN, 1997).

Entre as motivações à aplicação de *MBPC* baseado em *AEs* pode-se mencionar (FARDIN *et al.*, 1997):

- (i) a dificuldade de calcular as segundas derivadas dos previsores, para obter uma aproximação da matriz Hessiana da função custo, na aplicação de algoritmos de otimização do tipo Gauss-Newton;
- (ii) dependendo das condições iniciais, o algoritmo de otimização pode convergir para um conjunto incorreto de parâmetros;
- (iii) para os horizontes de previsão maiores que 10, o algoritmo apesar de recursivo, necessita de muito tempo e o esforço computacional é consideravelmente aumentado.

Os *AEs* podem ser utilizados para obter-se os parâmetros de projeto (horizonte de controle, horizonte de previsão, ponderações das variáveis da função custo) de um *MBPC*.

O projeto da otimização via *AE* de um *MBPC* pode ser efetivada pelos seguintes passos:

- (i) iniciar a população dos parâmetros de projeto do *MBPC* de forma aleatória. Entretanto, pode-se incluir algumas soluções (alguns indivíduos da população) que o projetista crê serem adequadas para os parâmetros do *MBPC*;
- (ii) obter as medidas do processo;
- (iii) estimar os parâmetros do modelo matemático que representa o processo (no caso adaptativo, usualmente, através do algoritmo dos mínimos quadrados recursivo);
- (iv) determinar a(s) saída(s) de referência;
- (v) calcular a saída do processo utilizando o modelo de previsão;
- (vi) determinar o valor da função custo que está sendo minimizada, com a utilização dos parâmetros do membro da população (parâmetros de projeto do *MBPC*) que está sendo avaliado;
- (vii) aplicar a entrada de controle, Δu , que minimiza a função custo;
- (viii) repetir os passos (iv) a (vii) para cada amostra, k ;
- (ix) ir para o passo (ii) até que toda a população de parâmetros de projeto do *MBPC* tenha sido avaliada;
- (x) no caso do critério de parada não ter sido alcançado aplicar os operadores evolutivos para a geração de uma nova população de parâmetros de projeto do *MBPC* e retornar ao passo (ix);
- (xi) finalizar o procedimento de otimização. Os parâmetros de projeto obtidos vêm do membro da população com menor valor de função custo.

Alguns estudos de projeto preditivo evolutivo são apresentados na literatura. FARDIN *et al.* (1997) apresentam aplicação de *AGs* na identificação paramétrica de sistemas discretos dentro de uma abordagem de identificação relevante em controle preditivo. COELHO *et al.* (1999f) tratam o projeto do *GPC* baseado em múltiplos modelos com otimização através de *AGs*. RAMÍREZ *et al.* (1999) propõem um esquema

de *MBPC* não linear com otimização por *AGs*, aplicado a uma plataforma de testes de robótica móvel.

6.8.4 *GPC* com sistemas híbridos inteligentes

Existem diversas abordagens possíveis para a configuração do *GPC* (ou mesmo outros controladores) com sistemas híbridos inteligentes, conforme pode ser verificado pela figura 5.17 no capítulo anterior. Uma nova abordagem de controle preditivo multivariável foi proposta em COELHO & COELHO (2000c). Nesta tese é apresentado o projeto de *GPC* multivariável baseado em modelo obtido via *RN-RBF* e otimização híbrida de *PE* e método *simplex*. O projeto desta abordagem de *GPC* para o tratamento de processos multivariáveis tem os seguintes passos (figura 6.5):

- (i) obter as medidas do processo (dados de entrada(s) e saída(s));
- (ii) determinar a topologia da *RN-RBF*;
- (iii) configurar as entradas e das saídas passadas do processo que compõem as entradas da *RN-RBF*;
- (iv) identificar o processo multivariável pelo treinamento da *RN-RBF*;
- (v) configurar os parâmetros de projeto do *GPC*. Neste caso foi realizada de forma heurística. Contudo, pode-se utilizar *AEs* para esta finalidade;
- (vi) determinar a(s) saída(s) de referência;
- (vii) prever o desempenho do processo utilizando o modelo *RN-RBF*. Isto é realizado para o horizonte de saída, N_2 , do modelo *RN-RBF*, e determinar o valor da função custo quadrática que está sendo minimizada;
- (viii) repetir o passo (vii), utilizando o algoritmo híbrido de *PE* e o método *simplex* até que um valor de minimização aceitável seja obtido;
- (ix) aplicar a entrada de controle, Δu , que minimize a função custo;
- (x) repetir os passos (vi) a (ix) para cada amostra, k .

A seguir são mencionadas as vantagens e desvantagens da utilização de modelos neurais e algoritmos evolutivos no projeto do controle preditivo de uma forma genérica.

As vantagens da utilização de uma *RN* em projeto de *MBPC* são relacionados à propriedade de aproximador universal das *RNs* que garante a existência de pelo menos uma *RN* que pode representar um dado processo com a precisão desejada.

As *RNs* são frequentemente mencionadas como uma panacéia para configuração de modelos não lineares necessários ao projeto de *MBPC*. Entretanto existem vários aspectos a serem considerados quando da escolha de uma *RN* para aplicações de *MBPCs*.

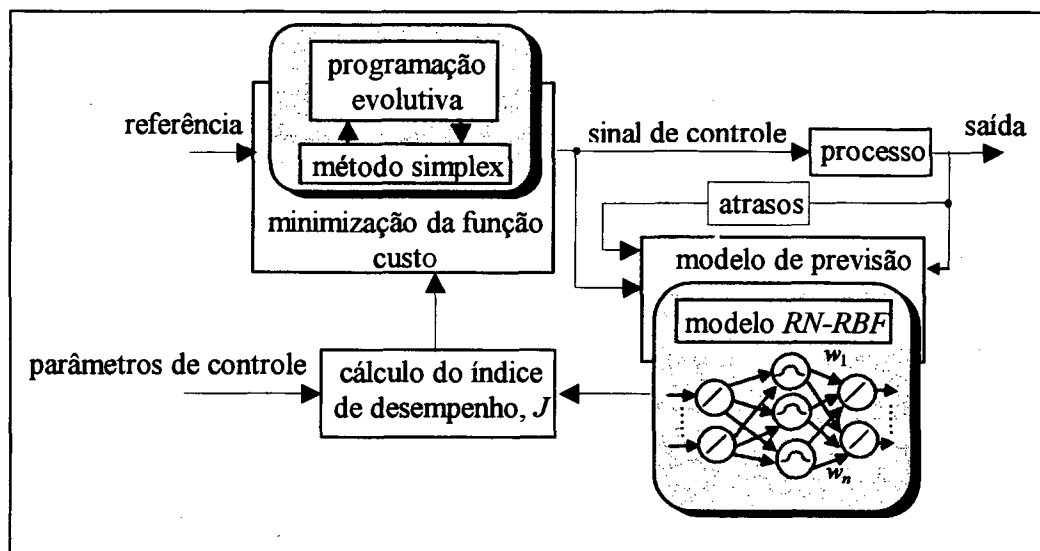


Figura 6.5. Diagrama do esquema de configuração do GPC baseado em RN-RBF e otimização evolutivo-simplex.

Entre as desvantagens do projeto de *MBPC* baseado em *RNs* enfatizam-se algumas. Em aplicações de controle de multivariável conforme considerado aqui, usualmente, não é possível o treinamento *off-line* de uma *RN* em todas as regiões de operação do processo, devido a limitações práticas. Por conseqüência, utilizando-se somente uma *RN* treinada *off-line* pode-se obter para um desempenho inferior do controlador. Por conseguinte, o treinamento *on-line* de um emulador neural pode ser necessário para aprimorar-se o desempenho do controlador preditivo baseado em modelo neural.

Outro motivo que justifica o treinamento *on-line* do emulador neural é que durante a operação de controle a dinâmica do processo multivariável pode modificar-se. O treinamento da *RN* de forma *on-line* pode ser uma necessidade para acomodar apropriadamente mudanças no processo, principalmente para propósitos de controle adaptativo. A convergência é um problema para tais aplicações, pois uma *RN* pode não estar habilitada a convergir entre duas amostragens, devido à carga computacional do treinamento de algumas topologias de *RNs*, por exemplo, a *RN-MLP*. Conseqüentemente,

deve existir um compromisso para treinamento de forma *on-line* em *RNs*. A utilização de *RN-RBF* permite uma atualização eficiente das conexões de pesos, por exemplo, com a utilização do algoritmo de *MQR*, para adaptar o modelo a mudanças nas condições de operação do processo.

Outra desvantagem da utilização de *MBPC* baseado em *RNs* ocorre quando uma *RN* não é treinada *a priori* para começar a operação de controle. Neste caso, a incerteza da identificação pode ser ilimitada e o sistema pode ficar instável até que a *RN* seja treinada *on-line* durante a operação de controle. A fase de treinamento *off-line* é realizada em malha aberta (processos estáveis em malha aberta) ou mesmo pela obtenção de dados em malha fechada com um controlador projetado *a priori* no caso de processos instáveis em malha aberta. O treinamento da *RN* de forma *on-line* é similar ao treinamento *off-line*, sendo que na forma *on-line* o processo é controlado por um *MBPC* neural e opera em malha fechada.

Um outro problema no treinamento da *RN*, quando do tratamento de processos multivariáveis, é o efeito de perturbações. Deseja-se treinar um sistema para eliminar as perturbações, contudo a *RN* aprenderá as perturbações como sendo parte da dinâmica do processo.

Quanto à otimização evolutivo-simplex da lei de controle do *MBPC*, os seguintes aspectos devem ser mencionados. O procedimento de otimização via *AEs* tem a desvantagem de ser computacionalmente intensivo, pois requer muitas avaliações da função custo, isto é, o procedimento requer a coleta de todas as amostras com o controlador, atuando em malha fechada, na avaliação da função de aptidão das ponderações do *MBPC*.

Um melhor desempenho e rapidez na obtenção de uma solução adequada para o *MBPC* depende de diversos fatores, entre os quais: (i) número de parâmetros do *MBPC* a serem encontrados via *AEs*; (ii) o intervalo de busca de cada parâmetro de projeto; (iii) período de amostragem adotado; (iii) número de gerações e tamanho da população; e (iv) propriedades de convergência do *AE*.

Os *AEs* são promissores em aplicações de otimização de *MBPCs* não lineares e que apresentem restrições. Entretanto, deve-se ter cuidado com a relação custo e benefício

quanto a precisão do controle, convergência e custos computacionais dos *AEs* em relação a outros algoritmos de otimização não linear.

Os métodos convencionais de otimização são muito sensíveis à inicialização do algoritmo e, muitas vezes, levam a soluções inaceitáveis devido à convergência para ótimos locais. A ausência de pesquisas quanto a aplicações de *AEs* para projetos de *MBPC* pode ser parcialmente justificada pela complexidade numérica dos *AEs*, estes sendo somente adequados, quando aplicados na otimização da lei de controle, para processos que apresentam dinâmica lenta. Contudo, os *AEs* podem ser uma ferramenta promissora para o projeto de *MBPCs*, especialmente para sistemas não lineares, apresentando restrições e problemas de otimização não-convexa.

Na presença de não-linearidades e restrições, usualmente, um problema de otimização não convexa deve ser resolvido a cada período de amostragem. O horizonte de controle determina diretamente a dimensão do problema de otimização, que pode chegar a ser muito complexo. Diferentes algoritmos, tais como programação quadrática seqüencial ou método *branch-and-bound* podem ser utilizados para evitar este problema. Contudo, programação quadrática seqüencial usualmente converge para mínimos locais apresentando uma solução inapropriada, e o método *branch-and-bound* requer a discretização do espaço de controle e acarreta também em uma acentuada complexidade computacional, que crescem exponencialmente com o número de ações de controle possíveis e com o horizonte de controle. A utilização de *AEs* é uma outra possibilidade pois estas técnicas têm a vantagem de tratar com problemas complexos de otimização não-convexa com características não lineares acentuadas.

6.9 Conclusão do capítulo

As estratégias de *MBPC* são caracterizadas por incluir o modelo explícito do processo, que é utilizado para prever o comportamento atual do processo, identificar funcionamento inadequado ou otimizar o desempenho, dependendo da aplicação específica. De forma que semelhante a outras tecnologias, os *MBPCs* requerem mudanças significativas na formação e atitudes de engenheiros de controle e operadores de processos

visando o aproveitamento de todos os benefícios que esta potencial metodologia oferece (BONAVITA & TOMASI, 1998).

Neste capítulo abordou-se inicialmente um breve histórico da evolução dos *MBPCs* e as diversas variantes geradas ao longo das décadas. A motivação à utilização e aplicações industriais dos *MBPCs* mostram que os algoritmos de controle preditivo são uma poderosa ferramenta para utilização no meio industrial, motivando pesquisas de formas eficientes de *MBPC*, tanto lineares quanto não lineares.

O projeto de *GPC* multivariável adaptativo, através do algoritmo *MQR*, foi apresentado. A parte final deste capítulo tratou algumas possibilidades de projeto *GPC* combinado a algoritmos inteligentes.

Capítulo 7

CONTROLE DE ESTRUTURA VARIÁVEL

7.1 Introdução

Nas últimas décadas, diversas abordagens de controle por modos deslizantes (*SMC*) têm sido estudadas sob perspectivas diferentes (DECARLO *et al.*, 1988, HUNG *et al.*, 1993). Infelizmente, apesar do *SMC* apresentar vantagens, alguns estudos questionam se o *SMC* exibe um comportamento indesejável de *chattering* no sinal de controle e erro em regime (NANDAM & SEN, 1992).

Os esforços iniciais nos estudos de controle *VSS* foram concentrados para o caso contínuo. Na abordagem contínua, a regra da alteração da estrutura do controlador é regida por funções de variáveis de estado do sistema. Quando igualadas a zero, estas funções definem uma superfície no *espaço de estados* denominada *superfície de deslizamento* (*superfície de chaveamento*), que rege o chaveamento do controlador. A lei de controle mantém o estado do sistema, movendo-se nesta superfície, até atingir o ponto de equilíbrio. A propriedade do estado do sistema de permanecer deslizando, nesta superfície, é denominada *modo deslizante*.

Um sistema de controle *VSS*, na forma contínua, é robusto para incertezas paramétricas e perturbações externas, porque o movimento é deslizante em um hiperplano pré-definido (DECARLO *et al.*, 1988). Este movimento deslizante não existe quando o controle *VSS* é implementado na forma discreta. Deste modo, a robustez do controle *VSS* tradicional não é preservada, depois de ser discretizada, para as implementações de controle digital. Em síntese, o modo deslizante projetado, na forma contínua, pode não ser razoável, pois a frequência de amostragem pode ser inadequada ao atendimento dos requisitos teóricos de um controlador *VSS*. Em consequência, o fenômeno de *chattering* ocorre em torno da superfície de deslizamento ou o controlador apresenta um

comportamento instável, quando são adotados valores de ganho elevados (FURUTA & PAN, 1996, HASKARA *et al.*, 1997).

A concepção do controle de estrutura variável (*VSS*) tem essência no controle do tipo *bang-bang* e na teoria dos relés (DECARLO *et al.*, 1988). A modificação da estrutura do controlador *VSS*, através de chaveamento, é realizada de acordo com uma regra estabelecida *a priori*, visando a obtenção de um comportamento adequado do controlador e da saída do processo.

O controle por modos deslizantes (*SMC*) é um caso especial do controle *VSS*, em que as propriedades de diferentes estruturas de controle são utilizadas na síntese de um sistema de controle estável. O *SMC* apresenta robustez, resposta rápida e precisão quando projetado adequadamente. Os méritos do *SMC* são tratamento de: variações paramétricas, perturbações externas e erros de identificação.

A teoria de estrutura variável é estudada por pesquisadores, tanto em tempo contínuo (DECARLO *et al.*, 1988, HEALAY & LIENARD, 1993), quanto em tempo discreto (FURUTA, 1990, GAO *et al.*, 1995, FURUTA & PAN, 1996, CORRADINI & ORLANDO, 1997, HASKARA *et al.*, 1997).

O projeto de controle *VSS*, para os sistemas discretos, em vez de um modo deslizante, emprega um setor deslizante e foi proposto por FURUTA (1990). Neste caso, o controlador *VSS* discreto é projetado para mover-se do estado do sistema do exterior para o interior do setor deslizante, onde o sistema é estável.

Neste capítulo, apresenta-se o histórico e as características de projeto do controle *VSS*. Um resumo das pesquisas de controle *VSS* para as aplicações em processos multivariáveis são também apresentados. A seguir, as concepções de controle *VSS* discreto adaptativo e incorporando as metodologias da *IC*, são descritas.

7.2 Breve histórico

O termo *VSS* apareceu na década de 50. Nesta época, os projetos de controle *VSS* foram estudados para os sistemas de uma entrada, representados por equações diferenciais de ordem elevada. Para tais sistemas, ambos os modos deslizantes e o controle dos modos

de aproximação da superfície eram realizados por procedimentos de tentativa e erro. Os primeiros trabalhos, nesta área, são de pesquisadores russos, entre os quais: A. F. Filippov, A. A. Andronov, A. A. Vitt, S. E. Khaikin, V. I. Utkin, S. V. Emelyanov, A. Bakakin, M. Gritsenko, N. Kostyleva e U. Itkis (DECARLO *et al.*, 1988, GAO & HUNG, 1993).

Os controladores de estrutura variável não receberam aceitação imediata dos engenheiros de controle até a década de 70. Isto foi causado, provavelmente, pelas limitações tecnológicas, falta de um procedimento de projeto, dificuldades da obtenção de chaveamento de alta velocidade e, também, pela existência de *chattering*. O crescimento no interesse em controle *VSS* ocorreu entre os anos 70 e 80, quando a teoria geral de controle *VSS* para os sistemas lineares foi formalmente estabelecida. Neste estágio do desenvolvimento de controle *VSS*, foram tratados os sistemas lineares com múltiplas entradas. Além disso, as propriedades de robustez e invariância do controle *VSS* foram reconhecidas (GAO & HUNG, 1993).

Apesar dos desenvolvimentos nesta área, da década de 80 até a presente data, as primeiras expectativas do controle *VSS* são constantemente reavaliadas. O potencial real do controle *VSS* é revelado e as direções de pesquisa abordam o tratamento de novas classes de problemas. As contribuições realizadas referem-se aos esquemas possíveis de chaveamento, a simplificações de projeto, a redução de *chattering* e a projetos multivariáveis. Segundo HUNG *et al.* (1993), as pesquisas sobre o controle *VSS* são classificadas em cinco categorias, resumidas como: (i) extensão dos objetivos de controle; (ii) estabelecimento das leis de controle *VSS*, que possuam certas características; e (iii) aplicações em diversos problemas de engenharia.

7.3 Motivação para a utilização do controle de estrutura variável

O controle *VSS* apresenta algumas potencialidades de projeto, pois não é necessário o conhecimento do modelo exato do processo e o controlador é adequado para o tratamento de perturbações externas e limitadas, que satisfazem certas condições.

Segundo SLOTINE & LI (1991), o projeto de sistemas de controle lineares, tais como: robusto, adaptativo e por modos deslizantes permitem o controle de processos com modelos apresentando incertezas, decorrentes do desconhecimento dos parâmetros do

processo ou de uma representação simplificada das dinâmicas do sistema. Do ponto de vista de controle, as imprecisões, na identificação, são classificadas em:

- (i) *incertezas estruturadas* (paramétricas): correspondem as imprecisões nos termos incluídos no modelo;
- (ii) *incertezas não-estruturadas* (dinâmicas não-modeladas): correspondem às imprecisões (subestimação) da ordem do processo.

Entretanto, os motivos da popularidade são as propriedades que o *VSS* do tipo *SMC* apresenta, tais como: o desempenho apropriado para o tratamento de processos não lineares, a aplicabilidade em processos multivariáveis e critérios de projeto, estabelecidos para tratar os processos discretos no tempo. A propriedade mais significativa de um *VSS* do tipo *SMC* é a robustez, pois quando um sistema está em modo deslizante é pouco sensível a mudanças de parâmetros ou a perturbações externas (KAYNAK & RUDAS, 1998).

Uma desvantagem do controle *VSS* é a possibilidade de ocorrência do fenômeno de *chattering*. Este fenômeno é comum, em casos experimentais, onde as limitações físicas dos atuadores, como, por exemplo, zona morta, histerese e atraso de transporte, impossibilitam a realização de um chaveamento rápido, e impossibilitando a obtenção de um modo deslizante. Conseqüentemente, isto acarreta o surgimento de um ruído de alta frequência na ação de controle quando o estado do sistema encontra-se próximo da superfície de deslizamento, fazendo o estado do sistema oscilar em torno da superfície de deslizamento, resultando em instabilidade e na redução da vida útil dos atuadores (TANG, 1997). A solução do problema é realizada pela utilização do conceito de camada limite (DECARLO *et al.*, 1988).

No caso de controle de sistemas discretos multivariáveis, o projeto de *VSS* do tipo *SMC*, na forma discreta, apresenta dificuldades quanto:

- (i) à necessidade da satisfação das condições de aproximação da superfície;
- (ii) ao tratamento da trajetória de estado, em modo quase-deslizante.

7.4 Aplicações industriais

Existe um interesse crescente na utilização de controle *VSS* para os problemas de engenharia, principalmente em aplicações de controle de robôs, controle de motores

elétricos de alto desempenho, controle aeroespacial, servomecanismos, conversores de potência e controle de veículos remotos (HUNG *et al.*, 1993).

Em TAKATSU & ITOH (1999) é realizada uma análise da situação atual e as futuras direções da aplicação de controle nas indústrias japonesas. Nesta pesquisa é observado que o *SMC* ainda é uma metodologia em estudo e com poucas aplicações na indústria japonesa.

7.5 Controlador *VSS* com modos quase-deslizantes (*QSMC*)

A forma de controle *SMC* proporciona uma resposta rápida, com características de estabilidade, rejeição a perturbações e rastreamento de trajetória. Nesta seção é apresentado o projeto de controle *QSMC*, em abordagem adaptativa e inteligente.

7.5.1 Aspectos teóricos

O projeto de um controlador *VSS*, do tipo *QSMC*, usualmente consiste de duas etapas (GAO *et al.*, 1995):

- (i) a determinação da função de chaveamento, $s(k)$, tal que o modo deslizante no plano de chaveamento, $s(k) = 0$ seja estável;
- (ii) a determinação da lei de controle, tal que a condição de *reaching* (aproximação da superfície) seja satisfeita. Isto implica que o estado, partindo de um estado inicial move-se na direção do plano de chaveamento e atinge-o em um tempo finito. Logo, resulta que o sistema de controle, em malha fechada, é globalmente estável.

Segundo GAO *et al.* (1995), a resposta de um sistema consiste de três modos:

- (i) o de aproximação (*hitting*) da superfície (*MH*);
- (ii) o modo deslizante (*MD*);
- (iii) o *steady-state* ou de regime (*MR*).

Na figura 7.1 é representado um exemplo da trajetória de um *VSS* para o caso contínuo e discreto.

Estas condições ideais são difíceis de encontrar na prática. Entretanto, este tipo de trajetória acontece em casos raros. A trajetória P representa o movimento de estados para um VSS discreto, considerando-se que o chaveamento não ocorre no plano de chaveamento ideal e o estado apresenta um comportamento de ziguezague sobre o plano.

7.5.2 Atributos e definições do $QSMC$

Os atributos relevantes de um $QSMC$ são:

- (i) a partir de um estado inicial, a trajetória deve mover-se, monotonicamente, em direção ao plano de chaveamento e cruzá-lo em tempo finito;
- (ii) uma vez que a trajetória tenha cruzado o plano de chaveamento em primeira instância, a trajetória passa o plano novamente em cada período sucessivo de amostragem, resultando em movimento de ziguezague sobre o plano de chaveamento; e
- (iii) o tamanho de cada passo sucessivo de ziguezague não deve aumentar e a trajetória deve permanecer dentro de uma banda específica.

Algumas deduções sobre estes atributos podem ser delineadas:

- (i) um sistema discreto exibe modos quase-deslizantes se $s(k)[s(k+1)-s(k)] < 0$.
- (ii) uma condição suficiente para a existência de um modo quase-deslizante discreto é que exista um inteiro positivo k_0 , tal que $|s(k+1)| < |s(k)|$, onde $k \geq k_0$ em uma vizinhança de $s=0$. Assim, se esta condição é satisfeita, então a seqüência $\{|s(k)|\}_{k=k_0}^{\infty}$ é monotonicamente decrescente e converge para zero (SARPTÜRK *et al.*, 1987).
- (iii) deve-se assumir que $\{s^+(k)\}_{k=0}^{\infty}$ e $\{s^-(k)\}_{k=0}^{\infty}$ são sub-sequências positivas e negativas $\{|s(k)|\}_{k=0}^{\infty}$, respectivamente. A condição suficiente para a existência do modo deslizante discreto é que a sub-sequência $\{s^+(k)\}$ e a sub-sequência negativa $\{s^-(k)\}$, começando de algum inteiro $k_0 \geq 0$, sejam seqüências monotonicamente decrescentes e crescentes, respectivamente. Então $s^+(k+1) < s^+(k)$, e $s^-(k+1) > s^-(k)$, para $k \geq k_0$.

Quanto às condições de estabilidade, o projeto de controle VSS discreto é retratado em alguns estudos relevantes, na sua maioria para os processos monovariáveis. SARPTÜRK *et al.* (1987) apresentam as condições suficientes para a existência de SMC discreto e discutem os aspectos de estabilidade. SPURGEON (1992) mostra que a

condição $|s(k+1)| < |s(k)|$ não é suficiente para a existência de modos quase-deslizantes. GAO *et al.* (1995) apresentam os aspectos de controle *VSS* discreto e mencionam a dificuldade da obtenção de um método satisfatório para o tratamento dos controladores *VSS* discretos com múltiplas entradas.

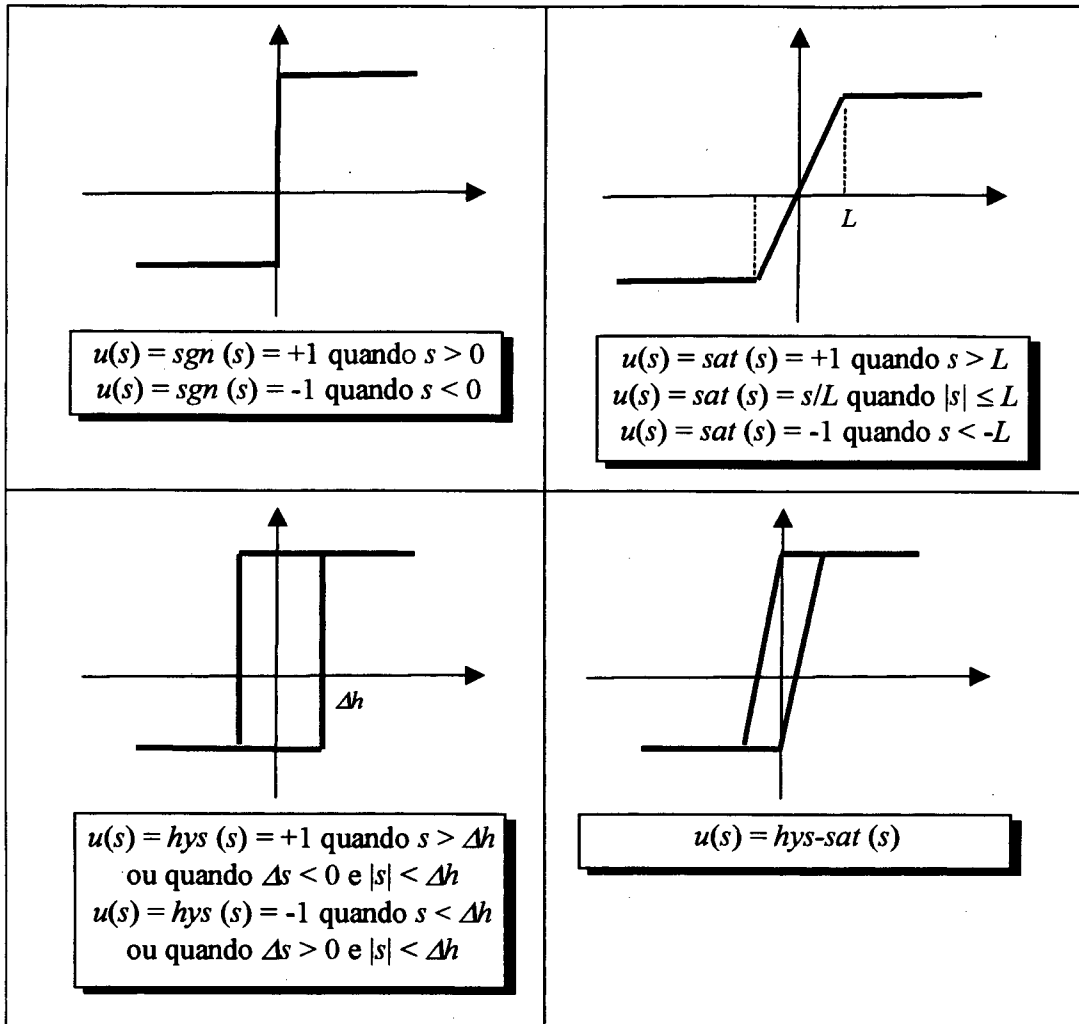
7.5.3 Problema do *chattering* e soluções propostas na literatura

Um aspecto relevante no projeto de *QSMC* é que o controle é chaveado rapidamente. Em projetos práticos, contudo, é impossível realizar um controle de chaveamento rápido, necessário em muitos projetos de controle *VSS*. Entre os motivos existe a presença de atrasos de tempo finitos para os cálculos da ação de controle e as limitações dos atuadores físicos.

É impossível em alguns casos práticos, por exemplo, o chaveamento do controle em taxa muito rápida e o *chattering* podem ocorrer, nos modos deslizantes e em regime. Em regime, o *chattering* aparece como uma oscilação, em alta frequência, sobre o ponto de equilíbrio desejado excitando as dinâmicas de alta frequência não-modeladas do processo (FURUTA, 1990, HUNG *et al.*, 1993). Considerando que o *chattering* é um efeito indesejado em projetos de *QSMC*, esforços significativos de pesquisa são direcionados à eliminação ou à redução dos seus efeitos (SU *et al.*, 1993; FURUTA & PAN, 1996).

Uma técnica para a redução deste problema é a abordagem contínua. Nos projetos de *QSMC*, a ação de controle contém termos do tipo relé. O relé ideal é praticamente impossível de ser implementado, assim uma abordagem para reduzir o *chattering* é substituir o controle do tipo relé por uma saturação, ou seja, uma aproximação contínua.

Em termos de análise, através de espaço de estados, uma camada de fronteira em torno da superfície de chaveamento é introduzida. Em síntese, é utilizada uma lei de controle contínua que aproxima o funcionamento da lei descontínua, numa região próxima da superfície de chaveamento. Algumas formas de saturação são apresentadas na figura 7.2.



Fonte: HUNG *et al.* (1993).

Figura 7.2. Exemplos de funções de chaveamento.

7.5.4 QSMC combinado a controle de variância mínima generalizada

O projeto de controle *VSS* do tipo *QSMC* discreto combinado a um controlador de variância mínima generalizada (*GMV*) é baseado nas propostas de FURUTA (1993) e CORRADINI & ORLANDO (1995, 1997). O processo é regido pela equação (6.3), enquanto as matrizes polinomiais são regidas pelas equações (6.4) a (6.5). O objetivo de controle é formalizado como a minimização do valor quadrático da equação:

$$s(k+d) = H(z^{-1})e'(k+d) + P(z^{-1})u(k) \quad (7.1)$$

onde $e'(k) = y(k) - y_r(k)$ é o negativo do erro de seguimento de referência, $H(z^{-1})$ e $P(z^{-1})$ são matrizes polinomiais de grau apropriado nH e nP regidas por:

$$H(z^{-1}) = H_0 + H_1z^{-1} + \dots + H_{nH}z^{-nH} \quad (7.2)$$

$$P(z^{-1}) = P_0 + P_1z^{-1} + \dots + P_{nP}z^{-nP} \quad (7.3)$$

O objetivo de controle é obter a condição onde a matriz $s(k+d) = 0$, que fornece a lei de controle dada pela vetor $u(k) = [u_1(k) \ u_2(k) \ \dots \ u_m(k)]$, sendo que (FURUTA, 1993):

$$u(k) = -[G(z^{-1})]^{-1} [F(z^{-1})y(k) - H(z^{-1})y_r(k+d)] \quad (7.4)$$

onde as matrizes polinomiais

$$F(z^{-1}) = F_0 + F_1z^{-1} + \dots + F_{nA-1}z^{-(nA-1)} \quad (7.5)$$

é a solução da equação polinomial

$$H(z^{-1}) = E(z^{-1})A(z^{-1}) + F(z^{-1})q^d \quad (7.6)$$

sendo

$$G(z^{-1}) = G_0 + G_1z^{-1} + \dots + G_{nB+d-1}z^{-(nB+d-1)} \quad (7.7)$$

e

$$G(z^{-1}) = P(z^{-1}) + E(z^{-1})B(z^{-1}) \quad (7.8)$$

onde $E(z^{-1})$ é uma matriz polinomial de grau apropriado. O termo $P(z^{-1})u(k)$ na equação (7.1) e a lei de controle da equação (7.4) controla processos de fase não-mínima.

O desempenho da lei de controle (7.4) é aprimorado pela adição de uma entrada auxiliar, obtida pela combinação do controlador *GMV* com uma estrutura baseada em *QSMC*. A lei de controle para dentro do setor deslizante é escolhida por

$$u(k) = -[G(z^{-1})]^{-1} [F(z^{-1})y(k) - H(z^{-1})y_r(k+d) - \Omega_k] \quad (7.9)$$

A alteração da lei de controle pela adição de um termo de chaveamento linear ponderado, $-\Omega s(k)$, serve para modificar a resposta do sinal, dimensionar o tamanho do sinal de controle e melhorar a estabilidade dentro e fora do setor deslizante, com $0 < \Omega < 1$ (LEE & OH, 1998). A lei de controle, fora do setor deslizante, apresenta a seguinte forma:

$$u(k) = -\left[G(z^{-1})\right]^{-1} \left[F(z^{-1})y(k) - H(z^{-1})y_r(k+d) - \Omega s(k) - V(k) \right] \quad (7.10)$$

onde a matriz $V(k)$ é escolhido de forma que

$$V(k) = \sum_{j=0}^{n-1} V_j e'(k-j) \quad (7.11)$$

onde V_j é uma matriz que consiste de uma função não linear do tipo *bang-bang* que depende do estado de fora do setor. Os coeficientes de V_j são chaveados fora do setor, $\delta_j(k)$, definido por um conjunto de estados de erro que tornam o sistema estável, quando os coeficientes da lei de controle são escolhidos, tal que

$$V_j = \begin{cases} V & s(k)e(k-j) < -\delta_j \\ 0 & |s(k)e(k-j)| \leq -\delta_j \\ -V & s(k)e(k-j) > -\delta_j \end{cases}, j=0,1,\dots,n-1 \quad (7.12)$$

onde δ_j é definido como:

$$\delta_j = \frac{\alpha^{n-1}}{2} \sum_{i=0}^{n-1} [|e(k-j)| |e(k-i)| V] \quad (7.13)$$

e o parâmetro $\alpha \geq 1$ (FURUTA, 1993, LEE & OH, 1998). Esta configuração de controle garante a execução de um movimento deslizante discreto estável no hiperplano $s(k+d) = 0$, ou seja, a eliminação do erro de seguimento, se os coeficientes V do polinômio $V(k)$ são escolhidos apropriadamente, tal que

$$V(k) = V_0 e'(k) + V_1 e'(k-1) + \dots + V_{n-1} e'(k-n+1) \quad (7.14)$$

Segundo CORRADINI & ORLANDO (1997), a seleção dos parâmetros de $v(k)$ afeta a duração e a forma do transiente inicial das variáveis de controle e da saída. Por

isso, as configurações devem ser realizadas de acordo com um “melhor” compromisso de aceitação dos esforços iniciais de controle e a duração do transiente. Neste caso, opta-se por uma metodologia de síntese através de algoritmos de otimização convencionais (ou mesmo através de *AEs*).

A inserção da lei de controle (7.10) na expressão de $s(k+d)$ fornece $s(k+d) = s(k) + V(k)$. Para verificar a estabilidade do controlador proposto, uma função de Lyapunov definida positiva, $L(k)$, é escolhida para o controlador discreto, tal que (FURUTA *et al.*, 1989, FURUTA, 1990):

$$L(k) = \frac{1}{2} s(k)^2 \quad (7.15)$$

e também $\Delta s(k+d)$ pela equação

$$\Delta s(k+d) = s(k+d) - s(k) \quad (7.16)$$

A partir da equação (7.16) obtém-se:

$$V(k+d) = V(k) + 2s(k)\Delta s(k+d) + \Delta s(k+d)^2 \quad (7.17)$$

O objetivo de controle é fazer $V(k)$ decrescer ao longo da hipersuperfície de chaveamento. Observando-se isto, a condição de existência do modo deslizante, estendida de FURUTA (1990, 1993) é estabelecida por

$$s(k)^T \Delta s(k+d) < -\frac{1}{2} [\Delta s(k+d)]^T [\Delta s(k+d)] \quad (7.18)$$

Os detalhes da prova de estabilidade, para o *SMC* discreto são encontrados em SARPTÜRK *et al.* (1987), FURUTA (1990, 1993) e GAO *et al.* (1995).

7.5.4.1 Abordagem adaptativa do *SMC* com a identificação via *MQR*

Na seção 6.5 do capítulo 6 foi apresentada a abordagem adaptativa do *GPC* com a utilização do estimador dos *MQR*. Nesta seção de forma análoga, utiliza-se o estimador

dos *MQR* à estimação dos parâmetros do processo, e posteriormente na configuração da lei de controle do *SMC*.

Em geral, os parâmetros de processos práticos não são precisamente conhecidos e os mecanismos de identificação de parâmetros são incluídos. A existência de incertezas, nos parâmetros, que regem o modelo do processo é um fator relevante na configuração de um controle eficiente. O desempenho do sistema de controle é aprimorado, atualizando-se os parâmetros através de estimação *on-line* (CORRADINI & ORLANDO, 1995). Um esquema de identificação implícito via *MQR* é considerado. O vetor de parâmetros, θ , a ser estimado, é constituído dos coeficientes dos polinômios F_0, \dots, F_{nA-1} e G_1, \dots, G_{nB+d-1} . Os coeficientes são identificados pela minimização da função objetivo (FURUTA, 1993):

$$J = \sum_{j=0}^k \left[s(j) + H(z^{-1})y_r(j) - (G(z^{-1})u(j-d) + F(z^{-1})y(j-d)) \right]^2 \mu(k-j) \quad (7.19)$$

onde os coeficientes de $F(z^{-1})$, equação (7.6), e $G(z^{-1})$, equação (7.8), são identificados através do *MQR*.

7.6 *SMC* híbrido com inteligência computacional

A utilização de controle *QSMC* é, muitas vezes, limitada, na prática, pois sua estrutura requer chaveamento rápido na entrada (que não pode ser providenciada por atuadores reais) e é vulnerável a medidas ruidosas do processo, além de empregar sinais de controle desnecessariamente elevados.

Para amenizar estas dificuldades, diversas modificações na lei de controle deslizante original são propostas. Algumas abordagens recentes são baseadas na utilização de metodologias emergentes da *IC*, tais como: *AEs* (LI *et al.*, 1996), *SNs* (LEE *et al.*, 1998) e *RNs* (SUNDARESHAN & ASKEW, 1997).

A combinação de metodologias da *IC* em projeto *VSS*, em geral, objetiva os problemas de implementações práticas. Estas metodologias provêm liberdade para os engenheiros de controle explorarem o conhecimento do problema no tratamento de

incertezas (ou imprecisão) e aprenderem pela experiência. Além disso, as metodologias inteligentes são candidatas ao tratamento dos problemas associados com *QSMC*, discutidos nas seções anteriores (KAYNAK & RUDAS, 1998).

Segundo KAYNAK & RUDAS (1998), as abordagens de projetos, combinando *IC* e *VSS* apresentadas, na literatura, podem ser separadas em duas classes principais: as diretas e as indiretas.

Nas abordagens indiretas, o projeto básico e a filosofia da implementação de controle *VSS* são seguidos, na maior parte das vezes, de uma metodologia da *IC*, para preencher uma função secundária. Esta abordagem é utilizada para adaptar os parâmetros do controlador e/ou para eliminar o *chattering* e/ou tratar algumas dificuldades de identificação. As propostas desta abordagem são vinculadas a *SNs* em filtros retificadores, utilização de *SNs* e *RNs* para sintonia de parâmetros de *SMC*, *RNs* e *SNs* para identificação de incertezas (LEE *et al.*, 1998, KAYNAK & RUDAS, 1998).

Na abordagem direta, ao contrário da abordagem indireta, as metodologias da *IC* têm o papel principal no projeto e configuração da lei de controle, enquanto os fundamentos de *VSS* apresentam papel secundário. Uma área promissora é a aplicação de princípios dos *SNs* para o projeto de um hiperplano deslizante (PALM, 1994, LEE *et al.*, 1998). A seguir são discutidos alguns aspectos da forma que algoritmos da *IC* são incorporados em *SMC*.

7.6.1 Projeto *VSS* neural

A abordagem de projeto *VSS* neural direto é obtida pela combinação de uma *RN* e um controlador *VSS*. Nesta abordagem o controle *VSS* apresenta vantagens de robustez, que serve à redução da complexidade do treinamento das *RNs*. Em outra direção, as capacidades das *RNs* em identificação não linear e adaptativa de *VSS*, podem ser exploradas (abordagem direta).

As possíveis combinações de *RNs* e a teoria de *VSS* podem ser abordadas nas concepções direta e indireta. Quanto à concepção direta, as propostas objetivam a

identificação neural de não-linearidades e empregam as *RNs* em auxílio à configuração de controle *VSS* (SUNDARESHAN & ASKEW, 1997, FANG *et al.*, 1999).

Alguns exemplos da abordagem indireta, onde uma *RN* tem papel essencial e a teoria de *VSS* tem papel secundário, são relatados para o treinamento eficiente e *on-line* de *RNs*. Para sanar as dificuldades do treinamento através do algoritmo *BP*, POZNIAK *et al.* (1998) e PARMA *et al.* (1998) propõem um algoritmo baseado na teoria de modos deslizantes para o treinamento de *RNs-MLP*. Este algoritmo apresenta velocidade de convergência e robustez, viabilizando algumas implementações de controle neural em tempo real.

7.6.2 Projeto *VSS* nebuloso

O potencial dos *SNs*, na área de controle de processos, consiste na habilidade de incorporar diferentes metodologias de maneira sistemática. As conexões entre os *CLNs* e o controle *VSS* por modos quase-deslizantes são realizadas com as abordagens direta e indireta, mencionadas anteriormente. Neste caso, existem duas formas a serem seguidas:

- (i) o controlador *VSS* com abordagem de projeto via *SNs*; e
- (ii) o controlador nebuloso com abordagem de projeto *VSS*.

Entre as propostas apresentadas na literatura algumas são relevantes. PALM (1994) propôs uma classe específica de *CLNs* decorrente da análise do plano de fase, similar ao controle *VSS* com uma camada limite. Em YAGER & FILEV (1994) um *CLN* é projetado utilizando-se uma base de regras do tipo MacVicar-Whelan, onde os antecedentes usuais das regras de controle (erro e variação do erro) são substituídos por termos lingüísticos, descrevendo uma função de chaveamento e sua taxa de mudança, respectivamente. WU & LIU (1996) desenvolvem um estudo de *CLN*, na forma de *VSS* com modos quase-deslizantes, e tratam aspectos relacionados à estabilidade e à robustez em relação às incertezas e perturbações. LEE *et al.* (1998) aplicam um algoritmo nebuloso do tipo *TSK* para a regulação adaptativa da superfície deslizante. Os resultados obtidos com esta abordagem apresentam melhor desempenho que o método convencional com a superfície deslizante fixa.

Os *CLNs* são adequados no tratamento de processos que apresentam complexidades, “mal-definidos”, não lineares ou variantes no tempo. Os *CLNs* são metodologias de fácil implementação e, usualmente, não necessitam do modelo matemático do sistema controlado.

A característica de ação de controle “suave” dos *CLNs* pode superar as desvantagens do *SMC* ou *QSMC*. A combinação de *CLN* com a estrutura variável do *QSMC* para configurar um controlador por modos quase-deslizantes nebuloso (*QSMC-CLN*) é uma alternativa. Nesta forma de *QSMC-CLN*, o potencial do *QSMC* na modelagem de imprecisões e de distúrbios externos é combinado ao *CLN* que provê um comportamento satisfatório de amortecimento e redução do *chattering*.

7.6.3 Projeto *VSS* evolutivo

Os projetos de *SMC* são, usualmente, baseados em simulações e/ou ajustes empíricos de tentativa e erro, pois faltam métodos de projeto adequados. Os problemas de controle não linear apresentam dificuldades na aplicação de métodos analíticos, baseados em cálculos, para a otimização de parâmetros sob restrições ou o índice de desempenho não pode ser diferenciável (LI *et al.*, 1996).

Uma forma alternativa à seleção dos ganhos do controle *SMC* é a utilização de *AEs*. A síntese e otimização de parâmetros de controle *SMC* através de *AEs* é abordada em alguns estudos aplicados a processos monovariáveis. LI *et al.* (1996) utilizam os *AGs* de projeto automático em controle *SMC*. AL-DUWAISH & AL-HAMOUIZ (1998) propõem o projeto via *AG* para determinação dos ganhos de um controle *SMC*. Os *AEs* em configurações híbridas com *RNs* e *SNs* também são alvo de estudos e aplicações (NG & LI, 1994).

7.7 Conclusão do capítulo

O conceito de modos deslizantes é originário da teoria de controle de estrutura variável. A teoria de controle de estrutura variável foi introduzida para o controle de processos sujeitos a perturbações externas e variações internas ao processo. A simplicidade e robustez do controle de estrutura variável tornam esta metodologia uma alternativa atraente em projetos em controle.

O controle de estrutura variável, na forma contínua, com modos deslizantes é conhecido por tratar-se de uma abordagem robusta com respeito a incertezas paramétricas e perturbações externas. O movimento deslizante em um processo contínuo no tempo pode não ser satisfatoriamente encontrado em controle de estrutura variável implementado em um processo discreto no tempo. Existe então a necessidade de um projeto de *QSMC* com a configuração de um setor deslizante, em vez de um modo deslizante. A lei de controle do *QSMC* transfere o estado do sistema de fora para dentro do setor deslizante onde o sistema de controle, em malha fechada, é projetado para ser estável.

A pesquisa sobre *QSMC* tem crescido consideravelmente, nos últimos anos, e a integração do *QSMC* com outras técnicas de controle é uma opção que pode oferecer o desempenho que não é possível utilizando outras técnicas individualmente.

Em particular, a integração com algoritmos de controle preditivo, adaptativo e inteligente visando o tratamento de processos complexos. Neste capítulo tratou-se inicialmente um breve histórico dos controladores *VSS*, as características e a motivação à sua utilização. O projeto de *GMV* combinado a *QSMC* multivariável adaptativo, através do algoritmo *MQR* foi descrito. Algumas possibilidades do projeto de controladores de estrutura variável combinado a algoritmos inteligentes foram apresentadas também na parte final do capítulo.

Capítulo 8

FUNDAMENTOS PRÁTICOS E APLICAÇÕES EM IDENTIFICAÇÃO DE PROCESSOS

8.1 Introdução

Neste capítulo são abordados os conceitos fundamentais e as aplicações em identificação de processos multivariáveis não-lineares, através de metodologias da *IC*. Os fundamentos de identificação de processos são descritos a seguir, incluindo-se:

- (i) os objetivos;
- (ii) os tipos de modelos;
- (iii) os sinais de excitação;
- (iv) a seleção de estrutura;
- (v) a escolha de ordem;
- (vi) os critérios de avaliação de desempenho; e
- (vii) os procedimentos de estimação e validação.

As metodologias inteligentes utilizadas nas aplicações, em identificação multivariável, são configurações de *AEs*, *SNs*, *RNs* e *SHIs*. As aplicações em identificação de processos multivariáveis abordam:

- (i) a identificação *MISO* de um processo experimental denominado balanço horizontal;
- (ii) a identificação *MIMO*, em malha fechada, de um manipulador robótico com dois graus de liberdade; e
- (iii) a identificação de processo não-linear com duas entradas e duas saídas.

8.2 Identificação de processos: fundamentos e objetivos

A tentativa de explicar ou reproduzir os comportamentos dos sistemas físicos é algo que há tempo desperta o interesse de pesquisadores. Com o desenvolvimento dos processos industriais e a necessidade de controlá-los, é preciso desenvolver modelos que reproduzam

suas características estáticas e dinâmicas. A identificação de processos tem relevância, pois pode-se prever o que acontece a um processo, conhecendo a(s) entrada(s) e a(s) saídas anteriores, disponíveis do processo.

Os modelos matemáticos podem ser de dois tipos: físicos e empíricos (PEARSON & OGUNNAIKE, 1997, HABER & UNBEHAUEN, 1990). Os modelos físicos são decorrentes do conhecimento das características químicas e físicas do processo. Devido à complexidade e às incertezas do processo, os modelos físicos precisos são, freqüentemente, complexos e custosos de se desenvolver. No entanto, são capazes de representar com fidelidade com o processo analisado. O custo pago pela fidelidade é a complexidade do modelo, e os modelos físicos, freqüentemente, não são adequados para o controle de processos, devido a dificuldades computacionais e matemáticas.

Os modelos empíricos, por outro lado, são estruturados pelos dados de entrada(s) e saída(s). Os modelos empíricos podem ser tão simples quanto desejados e podem ser projetados para auxiliar o projeto da estratégia de controle. Infelizmente, os modelos empíricos são, geralmente, menos precisos que os melhores modelos físicos, pois aproveitam a informação física. Evidentemente, muitas combinações de modelos físicos e empíricos existem, assim os dois grupos de modelos não são exclusivos.

O conhecimento matemático da dinâmica do processo é necessário na maioria dos esquemas de controle. O problema principal em identificação de processos é encontrar uma estrutura apropriada para o modelo do processo. A regra básica em estimação é “não estimar o que se conhece”. Em outras palavras, pode-se utilizar o conhecimento *a priori* do processo e suas características físicas, quando seleciona-se a estrutura do modelo. O conhecimento do processo pode ser classificado e codificado em (SJÖBERG *et al.*, 1995):

- (i) *modelo caixa-branca (white-box)*: o modelo é perfeitamente conhecido, sendo possível construí-lo a partir do conhecimento *a priori* das características físicas do sistema;
- (ii) *modelo caixa-cinza (grey-box)*: alguns conhecimentos das características físicas são avaliadas, mas diversos parâmetros necessitam ser determinados a partir dos dados observados. As duas modelagens consideradas para obtenção do modelo matemático são:
 - *modelagem física*: a estrutura do modelo é construída com base física, apresentando parâmetros a serem estimados a partir dos dados.
 - *modelagem semi-física*: o conhecimento das características físicas são utilizados para sugerir certas combinações de sinais de dados medidos.

(iii) *modelo caixa-preta (black-box)*: o conhecimento das características físicas não é avaliado (ou utilizado), mas a escolha da estrutura do modelo apresenta flexibilidade.

A análise de modelos caixa-preta é uma tarefa complexa. O motivo é que nada é excluído, e um espectro amplo das possíveis descrições do modelo é manipulado. Os estudos em identificação caixa-preta de processos não-lineares é diversificada e abrange tópicos de estimação e regressão não-paramétrica. Entre os algoritmos desta área estão os *wavelets*, modelos nebulosos e as *RNs* (SJÖBERG, 1995, SJÖBERG *et al.*, 1995).

Os métodos para identificação de processos podem ser divididos em três grupos:

- (i) os métodos paramétricos;
- (ii) os métodos não-paramétricos; e
- (iii) os métodos no domínio da frequência.

Os métodos paramétricos utilizam estruturas matemáticas parametrizadas para descrever o comportamento dinâmico, no domínio do tempo. Os parâmetros destas estruturas matemáticas são determinados através de algoritmos de estimação, a partir dos dados medidos. Através dos métodos não-paramétricos também obtém modelos no domínio do tempo e o comportamento dinâmico é determinado através de funções de correlação dos dados disponíveis. Os métodos no domínio da frequência configuram modelos utilizando ferramentas matemáticas, tais como *wavelets* e transformada de Fourier, para o cálculo da resposta em frequência do processo. Os modelos matemáticos obtidos pela identificação de processos podem ser divididos em duas classes: modelos de entrada(s) e saída(s) e modelos em espaço de estados.

A identificação de processos é, muitas vezes, como um problema de otimização, envolvendo algumas medidas para a adequação dos modelos candidatos a representar um processo. A escolha dos modelos matemáticos e o ajuste dos parâmetros são influenciados por fatores, entre os quais (COELHO & COELHO, 1999e, 1999f):

- (i) o conhecimento *a priori* do sistema (linearidade, grau de não-linearidade, atraso de transporte);
- (ii) as propriedades do modelo do sistema identificado (complexidade);
- (iii) a escolha da medida de erro a ser minimizada; e
- (iv) a presença de ruídos.

A identificação de sistemas é um exercício que envolve múltiplos e conflitantes objetivos, tipicamente a complexidade do modelo, o(s) critério(s) de desempenho e a

validação que influenciam a seleção da estrutura do modelo matemático. Existem diversas razões para manter a ordem do modelo tão baixa quanto possível. Os critérios de informação podem ser introduzidos para combinar a adequação e os princípios fundamentais de construção de modelos, tais como (COELHO & COELHO, 1999f, 2000a):

- (i) *princípio da redução de dados*: o menor número de variáveis deve ser utilizado para explicar uma quantidade máxima de informação;
- (ii) *princípio da parcimônia (ou razão de Occam)*: os melhores modelos são obtidos utilizando-se as estruturas aceitáveis simples, contendo o menor número de parâmetros.

Entre os critérios utilizados destacam-se: informação Bayesiana, Akaike ou *minimum description length*, que combinam a variância residual e a ordem do modelo. O objetivo do algoritmo de otimização é a minimização de um critério de desempenho. Se todas as restrições e as condições forem atendidas, o modelo encontrado é aceito. Caso contrário, se uma das condições impostas é violada, o procedimento de determinação do modelo, de estimação de parâmetros e diagnóstico do modelo deve ser repetido até que seja encontrado um modelo apropriado.

Nas últimas décadas, diversas configurações de algoritmos em identificação de processos não-lineares têm sido propostas, tais como: métodos freqüenciais, técnicas baseadas em estimativas de modelo Wiener-Hammerstein, Volterra e bilinear, análise de correlação, métodos de regressão, transformações *wavelets* e algoritmos de identificação recursivos (HARBER & UNBERHAUEN, 1990, LJUNG, 1996, WELLSTEAD & ZARROP, 1991).

Muitos dos métodos de identificação, tais como os baseados em mínimos quadrados ou estimativas de máxima verossimilhança, são, em essência, técnicas de busca local guiada por gradiente e necessitam de um espaço de busca regular ou uma função objetivo diferenciável. Estes métodos convencionais podem falhar na obtenção de um ótimo global, se o espaço de busca do modelo não é diferenciável. Adicionalmente, os métodos convencionais de identificação apresentam as seguintes desvantagens:

- (i) alguma informação inicial dos parâmetros do processo é necessária à convergência;
- (ii) os parâmetros estimados podem ser tendenciosos, se o ruído é correlacionado;
- (iii) a dificuldade na identificação do atraso de transporte; e
- (iv) a aplicação em sistemas não-lineares requer algumas alterações no algoritmo.

Este capítulo é dividido em duas partes. Na primeira parte, os aspectos necessários à obtenção de modelos matemáticos de processos multivariáveis não-lineares são descritos. Na segunda parte são apresentados resultados de experimentos de identificação *MISO* e *MIMO*, através de metodologias da *IC*.

8.3 Aspectos relevantes em identificação de processos

Os principais ingredientes ao problema de identificação de processos são (LJUNG, 1995):

- (i) um conjunto de dados de entrada(s) e saída(s);
- (ii) uma classe de modelos candidatos (estrutura do modelo);
- (iii) um critério de adequação entre os dados e o modelo; e
- (iv) as rotinas para a validação e aceitação dos modelos resultantes.

O desenvolvimento de metodologias em identificação de processos não-lineares é um procedimento que deve considerar as características do processo (PEARSON & OGUNNAIKE, 1997), tais como:

- (i) o grau de complexidade dinâmica;
- (ii) o grau de não-linearidade; e
- (iii) o grau de interação.

Um procedimento eficiente de identificação não-linear resolve o problema através das seguintes etapas (BILLINGS & CHEN, 1995):

- (i) *detecção de não-linearidades*: o sistema é linear ou não-linear?
- (ii) *detecção da estrutura*: quais termos estão no modelo?
- (iii) *estimação de parâmetros*: quais são os valores dos coeficientes desconhecidos?
- (iv) *validação do modelo*: o modelo está correto?
- (v) *previsão*: qual a saída do processo em um instante futuro?
- (vi) *análise*: quais são as propriedades do processo?

Estas questões constituem-se em um conjunto de ferramentas à estimação, que permitem uma descrição concisa do processo e pode ser utilizada como uma base à análise e o projeto de sistemas de identificação e controle.

Em síntese, o procedimento de identificação de processos (figura 8.1) é dividido nas seguintes etapas principais (LJUNG, 1987):

- (i) a obtenção dos dados do processo;
- (ii) a aplicação de testes aos dados obtidos à detecção de não-linearidades (as medidas têm de serem representativas do processo);
- (iii) a escolha da estrutura utilizada para representar o modelo;
- (iv) a determinação do algoritmo adequado para tratar o modelo do processo;
- (v) a estimação dos parâmetros do modelo; e
- (vi) a validação do modelo.

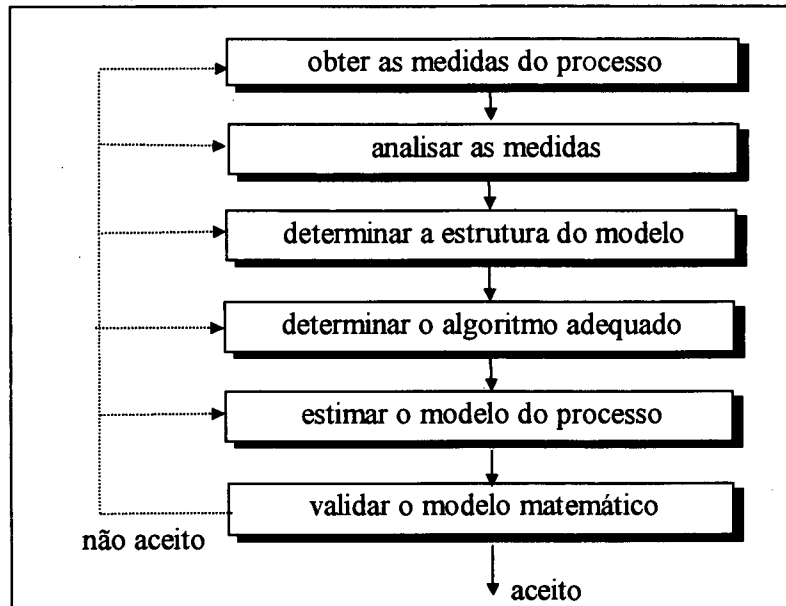


Figura 8.1. Procedimento para identificação de processos

A seguir é apresentada uma revisão dos aspectos relacionados ao procedimento de identificação. Os tópicos são relativos a representação de modelos, escolha do tempo de amostragem, escolha do sinal de excitação, detecção de estrutura, identificação e validação.

8.4 Experimentação do sistema e seleção do tempo de amostragem

Nesta etapa do procedimento de identificação, o processo é experimentado, através da aplicação de entradas pré-determinadas e da observação das saídas correspondentes. Os dados obtidos são utilizados na detecção de não-linearidades e no ajuste dos parâmetros do modelo (RODRIGUES, 1996).

O tempo de amostragem influi na seleção da estrutura, na estimação dos parâmetros do modelo e na capacidade do modelo reproduzir os diferentes regimes dinâmicos que o processo possui. A importância da seleção correta do período de amostragem é essencial e os aspectos a serem observados são (CORRÊA, 1997):

- (i) os dados devem ser amostrados, de forma suficientemente rápida, garantindo que as frequências de interesse estejam bem representadas no conjunto de dados usados para a identificação;
- (ii) o período de amostragem muito pequeno pode afetar o desempenho do algoritmo de seleção de estrutura e de estimação de parâmetros; e
- (iii) algumas interações não-lineares só aparecem e são reproduzidas, se a taxa de amostragem é suficientemente rápida.

No caso dos dados serem deliberadamente superamostrados, ou seja, um sinal amostrado mais rápido do que necessário resultam, frequentemente, em dados correlacionados e em problemas de condicionamento numérico durante a estimação de parâmetros. Por outro lado, uma subamostragem pode provocar o efeito de dados adjacentes não serem correlacionados e conduzir à perda de informação, se a dinâmica do processo for rápida, impossibilitando a identificação de forma apropriada. Alguns métodos para a escolha do tempo de amostragem utilizam funções de autocorrelação linear e não-linear (BILLINGS & TAO, 1991, BILLINGS & AGUIRRE, 1995).

8.5 Detecção de não-linearidades

Os processos reais são não-lineares. No entanto, a não-linearidade apresentada por alguns é suave, permitindo que a utilização de modelos linearizados (em torno de um ponto de operação) seja satisfatória. Quando isto não acontece, é essencial empregar um modelo não-linear para descrever a dinâmica do processo (RODRIGUES, 1996).

Os dados obtidos do processo podem ser tratados por algoritmos de detecção de não-linearidades, visando-se verificar a existência de interações não-lineares a serem modeladas (HABER & UNBEHAUEN, 1990). Os algoritmos mais utilizados na detecção de não-linearidades são:

- (i) o teste no domínio do tempo;
- (ii) o método da correlação cruzada não-linear; e

(iii) o método da autocorrelação.

O teste, no domínio do tempo, proposto por THOMSON *et al.* (1996) para caracterizar a presença do efeito não-linear no comportamento dinâmico do processo é utilizado em experimentos monovariáveis (COELHO *et al.*, 1999d). Este teste constitui-se de ensaios de resposta ao sinal degrau que pode ser estendido para processos multivariáveis.

Nesta tese, o teste de detecção de não-linearidades apresentado em BILLINGS & VOON (1986) é utilizado para detectar a presença de interações não-lineares em conjunto de dados.

8.6 Projeto do sinal de excitação

O projeto de seqüências de entrada é uma parte crítica à identificação de modelos não-lineares. As entradas para excitar o processo devem ser projetadas para satisfazer um conjunto de propriedades que, em princípio, garantem a adequação dos dados obtidos (BILLINGS & VOON, 1986). É necessário que o sinal de entrada consiga excitar o sistema no intervalo das freqüências de interesse.

Para os sistemas não-lineares, nos quais as variações da amplitude do sinal de entrada podem provocar mudanças qualitativas no comportamento do mesmo, é necessário projetar um perfil de amplitudes para os sinais de teste, de forma a garantir que todas as não linearidades presentes no processo sejam visitadas. Um projeto de seqüência de entrada apropriada deve além de apresentar excitação persistente, também refletir o comportamento do processo nas regiões de interesse, de forma que o modelo seja preciso em tarefas de previsão do comportamento do processo.

Os sinais mais utilizados para excitar os processos são ondas quadradas, seqüências binárias pseudo-aleatórias e ruídos Gaussianos, por possuírem espectro de freqüências relativamente grande. Em síntese, no projeto e na escolha dos sinais de excitação deve-se considerar alguns aspectos, tais como:

- (i) a necessidade da seleção de um sinal, que possua um espectro de freqüência, que excite todas dinâmicas de interesse na identificação do processo, de forma a obter-se um

- modelo matemático com capacidade adequada de identificação (interpolação) e previsão (generalização);
- (ii) a possibilidade de projeto de sinais compostos de diferentes sinais, explorando as características de abrangência, tanto em amplitude, quanto em frequência.

8.7 Escolha de estrutura e formas de representação

Os procedimentos de identificação lineares estão sedimentados, com relação à estimação de parâmetros e determinação da estrutura. As dificuldades existem em aplicar-se estes procedimentos de identificação em sistemas não-lineares. Nas últimas décadas, diversas representações em identificação de processos não-lineares foram propostas. Entre as abordagens mais relevantes existem as metodologias baseadas em estimativas de:

- (i) modelos de Wiener, Hammerstein, bilinear e séries de Volterra (MARCHI, 1999, MARCHI *et al.*, 1999);
- (ii) métodos freqüenciais e baseados na representação de espaço de estados (PEARSON & OGUNNAIKE, 1997);
- (iii) interpolação de modelos locais (JOHANSEN & FOSS, 1993);
- (iv) métodos de regressão e funções ortonormais (JOHANSSON, 1993, HABER & UNBEHAUEN, 1990);
- (v) modelos de blocos em cascata (JOHANSSON, 1993);
- (vi) modelos da IC (JAGANNATHAN & LEWIS, 1996, JANG, 1993);
- (vii) funções polinomiais e racionais (BILLINGS & VOON, 1986); e
- (viii) transformações *wavelets* e *splines* (SJÖBERG *et al.*, 1995).

A seleção da estrutura é também um aspecto relevante para a representação global da dinâmica de um sistema (AGUIRRE & JÁCOME, 1998). A escolha da estrutura é dividida em duas etapas:

- (i) quais agrupamentos de termos pertencem ao modelo; e
- (ii) quais termos, dentro dos agrupamentos escolhidos fazem parte do modelo.

A escolha adequada dos agrupamentos de termos evita os efeitos dinâmicos espúrios. O número de termos para representar a dinâmica do sistema deve ser suficientemente pequeno para evitar a instabilidade numérica provocada por sobreparametrização.

8.8 Seleção de ordem e estimação de parâmetros

A escolha da ordem de um modelo deve considerar um compromisso entre a capacidade de representação das dinâmicas do processo e a simplicidade do modelo, que representa um menor esforço computacional para os algoritmos de estimação e controle. Através destes métodos, é possível julgar se o incremento na ordem do modelo traz uma contribuição significativa na capacidade de representar o processo real (HABER & UNBEHAUEN, 1990). Existem diversas abordagens relevantes à seleção de ordem, tais como: métodos baseados na razão entre determinantes, métodos baseados em medidas de informação e análise de sensibilidade, através de *Lipschitz quotients* (PAN & LEVINE, 1992).

Após a determinação da estrutura do modelo, deve-se estimar seus parâmetros para aproximar o comportamento dinâmico do processo. Apesar de existirem diversas técnicas (JOHANSSON, 1993, WELLSTEAD & ZARROP, 1991), avalia-se, nesta tese, as metodologias da IC.

8.9 Identificação e validação de modelos

A validação de modelos é uma etapa importante em um procedimento de identificação. A validação de modelos visa a verificação dos méritos do modelo identificado e sua capacidade de generalização. A ausência de algum termo importante pode provocar a polarização dos parâmetros estimados. Desta forma, torna-se necessário submeter o modelo obtido a alguns testes que devem avaliar sua adequação em representar o processo original. Os testes de validação aplicados a um modelo indicam a sua capacidade de representar o processo original. A validação de modelos pode ser dividida em vertentes:

- (i) validação estatística: verificações do tipo F -teste, χ -teste e análise de correlação;
- (ii) validação dinâmica;
- (iii) pólos e zeros (sistemas lineares); e
- (iv) validação seqüencial (através de Nyquist para processos lineares)

Os testes de validação são uma forma de validação que detecta a presença de termos lineares e não-lineares nos resíduos do modelo estimado. A aplicação destes testes informa ao projetista se o modelo é adequado (ou não) dentro de um grau de confiança estipulado *a priori* (BILLINGS & VOON, 1986).

8.9.1 Validação através de análise de autocorrelação e correlação cruzada

Se o modelo matemático estimado não inclui todos os termos significativos do processo real, o modelo final selecionado não é uma representação apropriada. A validação de modelos indica a ocorrência de uma situação destas. Assim, se a estrutura e os valores dos parâmetros estão corretos, o resíduo $\varepsilon(k)$ não é previsível (descorrelacionado) de todas as combinações lineares e não-lineares das entradas e das saídas passadas. Se o modelo identificado é adequado, os erros de predição devem satisfazer as condições impostas por testes de correlação para validação de processos, que são as seguintes (BILLINGS *et al.*, 1989):

$$\Phi_{\varepsilon_i \varepsilon_j}(k) \quad k \neq 0 \text{ para } i = j \text{ (vale zero para } i \neq j) \quad (8.1)$$

$$\Phi_{u_i \varepsilon_j}(k) \quad \forall k \quad i = 1, \dots, m \text{ e } j = 1, \dots, n \quad (8.2)$$

$$\Phi_{\varepsilon_i(\varepsilon_j u_q)}(k) \quad k \geq 0 \quad i = 1, \dots, n ; j = 1, \dots, n \text{ e } q = 1, \dots, m \quad (8.3)$$

$$\Phi_{(u_i u_j) \varepsilon_q}(k) \quad \forall k \quad i = 1, \dots, m ; j = 1, \dots, m \text{ e } q = 1, \dots, n \quad (8.4)$$

$$\Phi_{(u_i u_j)(\varepsilon_q \varepsilon_t)}(k) \quad \forall k \quad i = 1, \dots, m ; j = 1, \dots, m ; q = 1, \dots, n \text{ e } t = 1, \dots, n \quad (8.5)$$

onde Φ são funções de correlação. Os testes de correlação, apresentados nas equações (8.1) a (8.5), são calculados de acordo com a equação:

$$\Phi_{(v_1)(v_2)}(k) = \frac{\frac{1}{N} \sum_{t=1}^{N-k} (v_1(t) - \bar{v}_1)(v_2(t+k) - \bar{v}_2)}{\sqrt{\left[\Phi_{(v_1)(v_1)}(0) \Phi_{(v_2)(v_2)}(0) \right]}} \quad (8.6)$$

onde os valores das funções de correlação são normalizadas de forma que $-1 \leq \Phi_{(v_1)(v_2)}(k) \leq 1$. As variáveis v_1 e v_2 podem ser substituídas pelas variáveis ε e u declaradas nas equações (8.1) a (8.5). Os intervalos de confiança são apresentados na forma de gráficos que indicam se a correlação entre as variáveis é significativa ou não. Se N tem valor acentuado, o desvio padrão a correlação estimada é $1/\sqrt{N}$, e os limites de confiança são aproximados por $\pm 1,96/\sqrt{N}$. As médias de v_1 e v_2 , representadas por \bar{v}_1 e \bar{v}_2 , respectivamente, são regidas por:

$$\bar{v}_1 = \frac{1}{N} \sum_{k=1}^N v_1(k), \quad \bar{v}_2 = \frac{1}{N} \sum_{k=1}^N v_2(k) \quad (8.7)$$

8.10 Critérios de desempenho

Os índices de desempenho adotados, para avaliação dos modelos matemáticos, apresentados, neste capítulo, são calculados pelos critérios:

(i) *SSE (Sum of Squared Error)*:

$$SSE_n = \sum_{i=1}^n \sum_{k=1}^N (y_i(k) - \hat{y}_i(k))^2 \quad (8.8)$$

(i) coeficiente de correlação múltipla, R^2 :

$$R_n^2 = 1 - \frac{\sum_{i=1}^n \sum_{k=1}^N (y_i(k) - \hat{y}_i(k))^2}{\sum_{i=1}^n \sum_{k=1}^N (y_i(k) - \bar{y}_i)^2} \quad (8.9)$$

onde i é o índice correspondente à saída i do processo, n é o número de saídas do processo, $y_i(k)$ é saída do processo, \bar{y}_i é a média das N amostras medidas para a saída i do processo. Quando o valor de R^2 é igual a 1,0 indica uma exata adequação do modelo para os dados medidos do processo. O valor de R^2 entre 0,9 e 1,0 é considerado suficiente para aplicações práticas, em sistemas de controle (SCHAIBLE *et al.*, 1997).

8.11 Fundamentos da construção de modelos NARMAX

A seguir são discutidos alguns aspectos e fundamentos de identificação não-linear do tipo NARMAX. Um processo não-linear discreto multivariável com m entradas e n saídas pode ser descrito por um modelo polinomial NARMAX com a forma geral (LEONTARITIS & BILLINGS, 1985; SARAIVA, 1999):

$$y(k) = F^l [y(k-1), y(k-2), \dots, y(k-n_y), u(k-1), u(k-2), \dots, u(k-n_u), e(k-1), e(k-2), \dots, e(k-n_e)] + \varepsilon(k) \quad (8.10)$$

onde

$$y(k) = \begin{bmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{bmatrix}, \quad u(k) = \begin{bmatrix} u_1(k) \\ \vdots \\ u_m(k) \end{bmatrix}, \quad e(k) = \begin{bmatrix} e_1(k) \\ \vdots \\ e_n(k) \end{bmatrix} \quad (8.11)$$

onde F é uma função não-linear de grau l .

A expansão de um modelo NARMAX polinomial configura um modelo linear nos parâmetros. Entre as dificuldades que a identificação NARMAX multivariável têm-se:

- (i) a determinação da estrutura, ordem e número de termos do modelo multivariável;
- (ii) a dificuldade do tratamento de acoplamentos, quando estes estão presentes, pois usualmente adota-se um projeto descentralizado; e
- (iii) a dificuldade de obter-se modelos multivariáveis simples para processos não-lineares e variantes no tempo, pois, em geral, modelos complexos apresentam dificuldades na sua manipulação e pouco valor prático.

8.12 Aplicação de identificação em malha fechada

Algumas abordagens recentes discutem que se um modelo é identificado para propósitos de projeto de um controlador, então a identificação em malha fechada é a forma de identificação preferida, em relação à identificação em malha aberta (HJALMARSSON *et al.*, 1996).

A identificação de sistemas dinâmicos baseados em dados, medidos sob condições experimentais em malha fechada, é um problema em algumas aplicações. Na prática, existem diversas situações em que a identificação em malha aberta é difícil (ou simplesmente não realizável), incluindo-se casos de processos com comportamento integrador ou instável em malha aberta. A identificação, em malha aberta, é também utilizada quando um controlador pronto está presente no sistema e a validação da sintonia atual ou um projeto novo do controlador é necessário (LANDAU & KARIMI, 1997). Os exemplos das aplicações são processos industriais, tais como, produção de papel, produção de vidro, processos de separação química, manipuladores robóticos e sistemas de movimento de alta precisão, em sistemas de controle de áudio e *drives* de *CD-ROM* (VAN DEN HOF & SCHRAMA, 1995).

8.12.1 Estudo de caso 1: Identificação *MISO* do processo balanço horizontal

O processo balanço horizontal (*PBH*) é composto por uma haste metálica, em cujas extremidades são montados dois motores de corrente contínua de 12 volts (controle u_1 e u_2) com hélices plásticas acopladas. O centro desta haste está fixo no eixo de um potenciômetro de precisão, acoplado à parte superior de outra haste metálica vertical, presa à base de apoio do protótipo. Esta base de apoio é formada por uma plataforma metálica, sobre a qual também está conectado um estojo plástico, que protege as placas de circuito impresso do processo e serve de interface entre o controlador a ser utilizado, o usuário e o protótipo, conforme apresentado na figura 8.2 (COELHO & COELHO, 1999a).

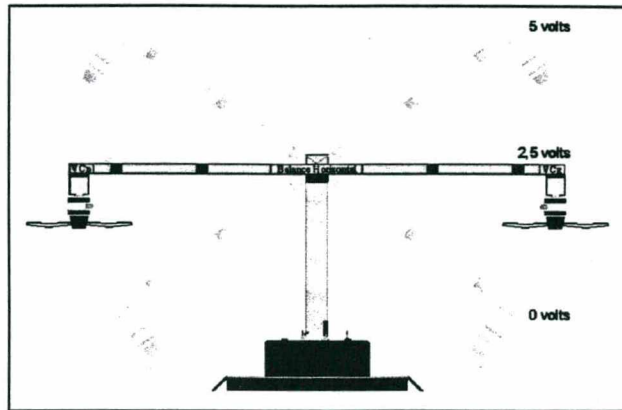
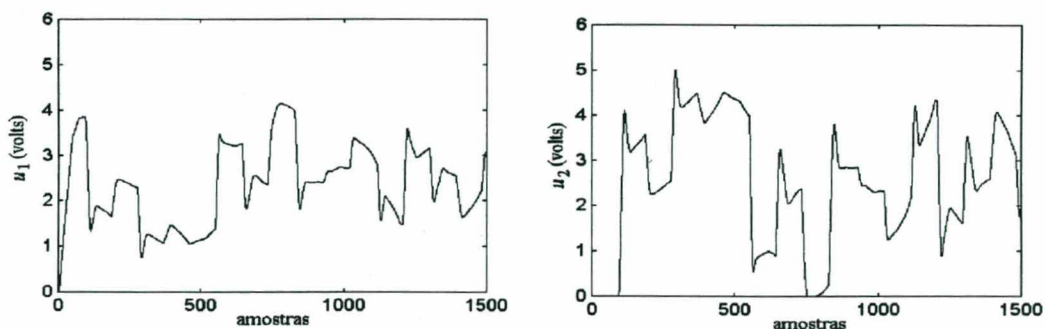


Figura 8.2. Representação da configuração do PBH.

O potenciômetro de precisão é responsável pela medição do ângulo da haste. Com o movimento da haste no plano perpendicular ao potenciômetro, este fornece variações de resistência que são transformadas, por um circuito transmissor, em valores de tensão padronizados na faixa de 0 a 5 volts, adequados ao controle analógico e digital. As saídas do controlador, também padronizadas em valores de 0 a 5 volts, são aplicadas aos motores, por intermédio de circuitos de acionamento, que transferem torque sobre o posicionamento da haste (COELHO *et al.*, 1998). Um peso preso a uma mola é inserido, no lado direito da haste do PBH, proporcionando peso diferenciado para cada lado da haste.

Os dados utilizados na fase de estimação (estimados os parâmetros do modelo matemático) são constituídos das amostras 1 a 1000, enquanto as amostras 1001 a 1500 são utilizadas na fase de validação do modelo matemático obtido para o PBH. A obtenção dos dados é realizada pela utilização de dois controladores *PID* (sintonizados por uma *EEMC*), em configuração *multiloop*, com controle de malha fechada do PBH, conforme apresentado na figura 8.3.



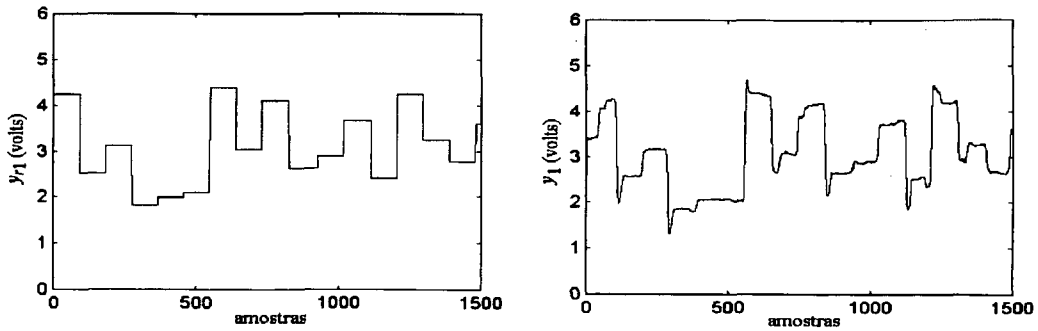


Figura 8.3. Sinais das entradas, referência e saída real do PBH, em malha fechada.

Os AEs, SNs e RNs são apresentados, a seguir, para configuração de algoritmo de identificação do PBH, conforme ilustrado na figura 8.4.

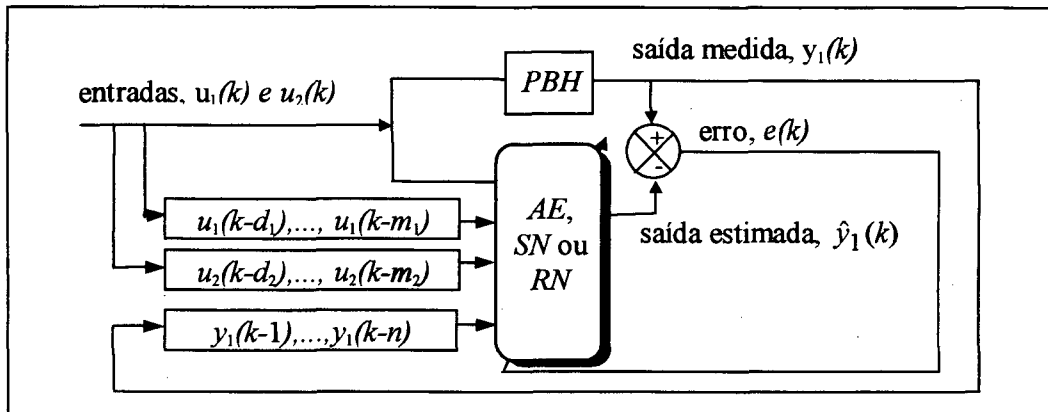


Figura 8.4. Configuração da identificação do PBH.

8.12.1.1 Identificação MISO do PBH via AEs

Na identificação do PBH via AEs obtêm-se a ordem, o atraso de transporte e os parâmetros de modelos matemáticos paramétricos lineares. Os modelos matemáticos lineares tratados são do tipo ARX (AutoRegressive model structure with eXogenous inputs) e ARMAX e regidos pelas equações:

(i) modelo ARX:

$$y_1(k) = B(z^{-1})u(k - d + 1) - A(z^{-1})y_1(k - 1) \tag{8.12}$$

(ii) modelo *ARMAX*:

$$y_1(k) = B(z^{-1})u(k - d + 1) - A(z^{-1})y_1(k - 1) + C(z - 1)e_1(k - 1) \quad (8.13)$$

A figura 8.5 mostra uma representação detalhada em forma de diagrama da identificação de modelo paramétrico linear através de *AEs*. Um detalhe relevante é que os parâmetros com valores inteiros (ordem do modelo e atraso de transporte, de cada malha), na tese, são obtidos através de *AG* canônico apresentando seleção por roleta, enquanto os parâmetros de ponto flutuante são obtidas através de diversos *AEs* descritos na tabela 8.1.

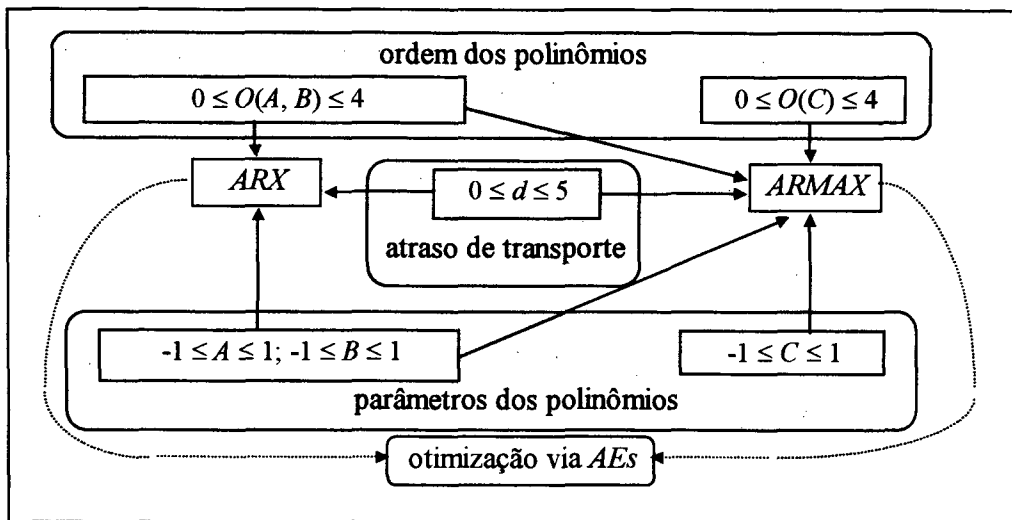


Figura 8.5. Representação da configuração de identificação de modelos lineares *ARX* e *ARMAX* para o *PBH* via *AEs*.

Os diversos experimentos realizados na identificação de um modelo *ARX* com *AEs* servem para constatar a necessidade da estimação de um polinômio $C(z^{-1})$ para representar o *PBH*. Prova disto, é que os resultados em identificação *ARX* através de *AEs* (configurações (i) a (xii) mencionadas a seguir), apresentam resultados de $0,672 \leq R^2_1 \leq 0,698$ e $243,289 \leq SSE_1 \leq 245,331$, para a fase de estimação. O modelo *ARX* é inadequado a representar o processo *PBH*. Os testes de autocorrelação e correlação cruzada são aptos a comprovar isto, onde pode-se observar a necessidade de inclusão de termos de erro para obter-se uma estimativa não-polarizada.

A capacidade de generalização do modelo *ARX* é também deficiente, pois os valores de R^2_1 e SSE_1 , para a fase de validação, são: $0,612 \leq R^2_1 \leq 0,644$ e $50,529 \leq SSE_1 \leq 54,499$.

A tabela 8.1 apresenta os melhores resultados da identificação *PBH* (modelo *ARMAX*) via *AEs*, sendo o critério de parada do ciclo evolutivo de 200 gerações e uma população de 30 indivíduos. São realizados 50 experimentos, para cada *AE*, sendo que são utilizadas sementes iguais, para a geração da população inicial, em cada experimento. As configurações dos *AEs* utilizados são:

- (i) *AG*(1): representação de ponto flutuante, método de seleção *breeder*, operador de recombinação aritmética ($p_c=0,8$), operador de mutação não-uniforme ($p_m=0,1$);
- (ii) *AG*(2): idem ao *AG*(1). A diferença é que o método de seleção é o *breeder* ($\beta=0,1$);
- (iii) *AG*(3): idem ao *AG*(1). A diferença é que o método de seleção é o *breeder* ($\beta=0,6$);
- (iv) *AGSA*(1): idem ao *AG*(1), mas utiliza *SA* para a busca local;
- (v) *AGSA*(2): idem ao *AG*(2), mas utiliza *SA* para a busca local;
- (vi) *AGSA*(3): idem ao *AG*(3), mas utiliza *SA* para a busca local;
- (vii) *EE*(1): *EE*(5+25);
- (viii) *EE*(2): *EE*(5,25);
- (ix) *EESA*(1): idem a *EE*(1), mas utiliza *SA* para a busca local;
- (x) *EESA*(2): idem a *EE*(2), mas utiliza *SA* para a busca local;
- (xi) *PE*(1): utiliza 15 ancestrais que geram 15 descendentes;
- (xii) *PESA*(1): idem a *PE*(1), mas utiliza *SA* para a busca local.

Tabela 8.1. Resultados da identificação do *PBH* — modelo *ARMAX* — via *AEs*.

modelo <i>ARMAX</i> <i>AEs</i> \ critério	fase de estimação (amostra 1 a 1000)		fase de validação (amostra 1001 a 1500)	
	R^2_1	SSE_1	R^2_1	SSE_1
<i>AG</i> (1)	0,959	22,506	0,969	25,002
<i>AG</i> (2)	0,982	13,020	0,986	13,650
<i>AG</i> (3)	0,956	26,304	0,957	27,980
<i>AGSA</i> (1)	0,982	13,061	0,986	13,812
<i>AGSA</i> (2)	0,982	12,950	0,986	13,513
<i>AGSA</i> (3)	0,982	12,642	0,986	13,201
<i>EE</i> (1)	0,982	12,630	0,986	13,148
<i>EE</i> (2)	0,982	12,690	0,986	13,254
<i>EESA</i> (1)	0,982	12,930	0,984	13,502
<i>EESA</i> (2)	0,983	12,804	0,986	13,348
<i>PE</i> (1)	0,967	24,693	0,973	26,560
<i>PESA</i> (1)	0,972	22,517	0,976	24,386

Constata-se pela tabela 8.1 que os *AEs* são adequados à estimação dos parâmetros do modelo e o atraso de transporte de um modelo *ARMAX*. Os critérios de desempenho apresentaram valores de $0,959 \leq R^2_1 \leq 0,983$ e $12,630 \leq SSE_1 \leq 26,304$, para a fase de estimação. Na fase de validação os resultados obtidos são de $0,969 \leq R^2_1 \leq 0,986$ e $13,148 \leq SSE_1 \leq 27,980$. O critério de desempenho R^2_1 para todos os *AEs* é próximo da unidade o que denota um modelo preciso para o *PBH*.

O modelo *ARMAX* mais adequado é obtido através da *EE(1)* apresentando $R^2_1=0,982$ e $SSE_1=12,630$ (fase de estimação) e $R^2_1=0,986$ e $SSE_1 = 13,148$ (fase de validação). Observa-se pela tabela 8.1, que outros *AEs* apresentaram resultados muito próximos do melhor obtido pela *EE(1)*. O modelo da *EE(1)* é regido pela equação:

$$y_1(k) = 0,9762y_1(k-1) + 0,9978u_1(k-3) - 0,9727u_1(k-4) + 0,9805u_1(k-5) - 0,116u_2(k-4) + 0,1895u_2(k-5) - 0,069u_2(k-6) + 0,9982u_2(k-7) + 0,0872e_1(k-1) \tag{8.14}$$

A análise de convergência dos *AEs* (melhor indivíduo), nos 50 experimentos, é apresentada na figura 8.6.

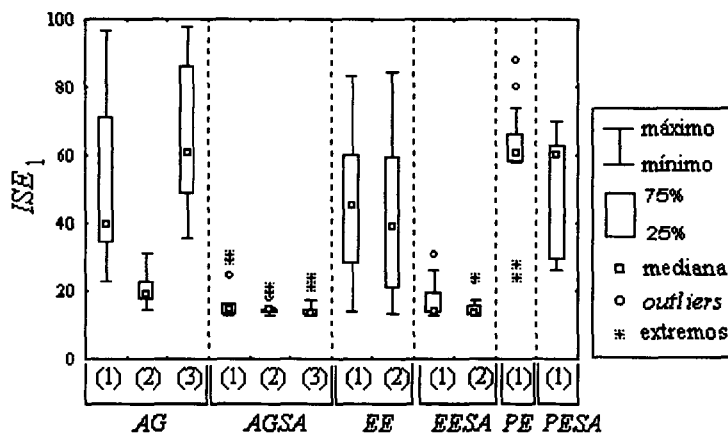
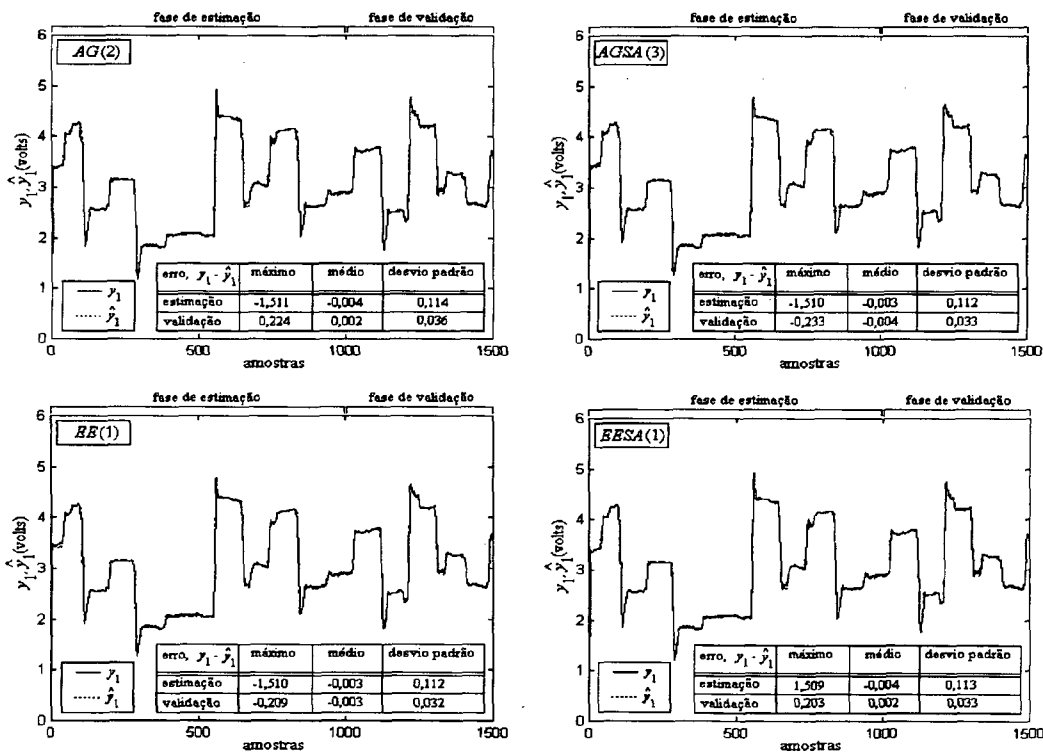


Figura 8.6. Análise comparativa de identificação *ARMAX* do *PBH* via *AEs*.

Pela análise desta figura constata-se os comportamentos dos *AEs*. Os *AG(1)*, *AG(3)*, *EE(1)*, *EE(3)*, *PE(1)*, *PESA(1)* apresentam um comportamento de variância acentuada em relação à média, com valores de máximo e mínimo muito distantes. Outra observação é que os valores dos percentis e medianas são também acentuados.

Apesar da *EE(1)* apresentar o melhor resultado, a configuração da *EE* adotada apresenta desempenho inferior aos algoritmos *AGSA(1)*, *AGSA(2)*, *AGSA(3)*, *EESA(1)*, *EESA(2)*. Os algoritmos *AGSA* e *EESA* apresentam resultados promissores, enfatizando-se o papel do algoritmo *SA* para busca locais, visando aprimorar os resultados obtidos com os *AEs*. Nota-se este fato, pois o *AG* e a *EE*, atuando isoladamente, apresentam resultados muito inferiores, quanto a capacidade de convergência.

Uma observação relevante é que algumas experimentações, para outros problemas em identificação (COELHO & COELHO, 1997e, 1997i, 1999c) e controle de processos (COELHO & COELHO, 1997b, 1998a, 1998c, 1999b) apresentam resultados de convergência similares aos da figura 8.6. Os melhores resultados da identificação *ARMAX* do *PBH* via *AEs* são apresentados na figura 8.7.



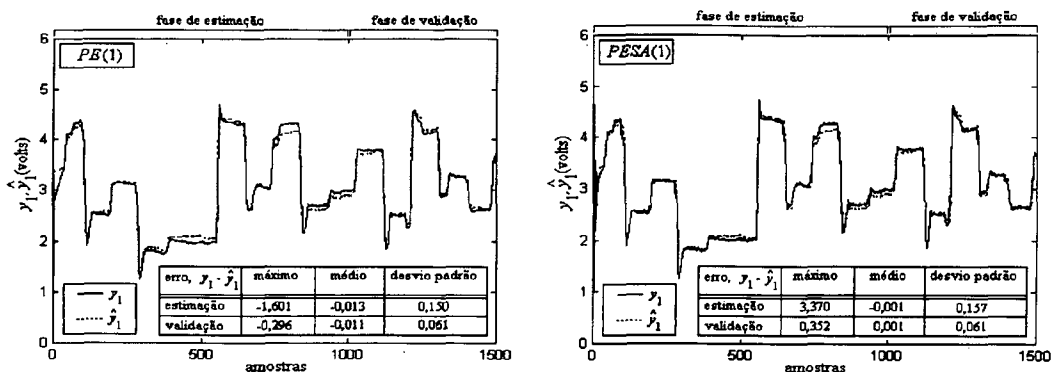
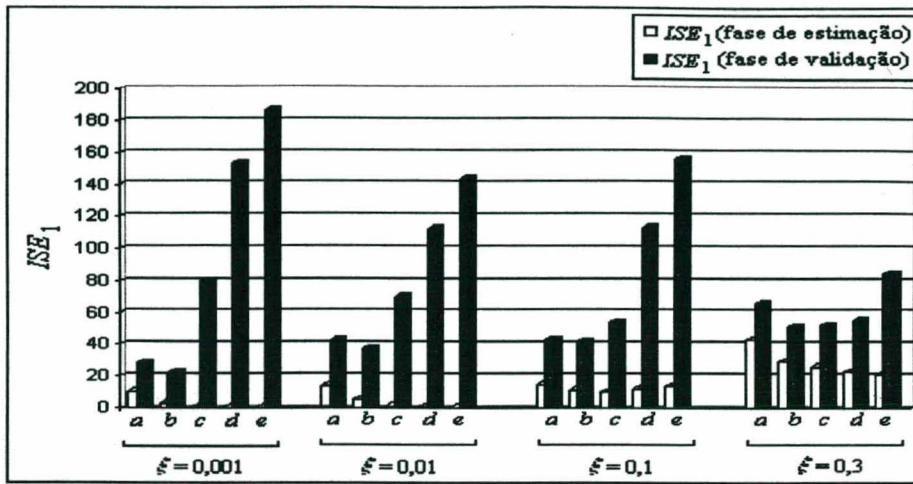


Figura 8.7. Resultado da identificação ARMAX do PBH via AEs.

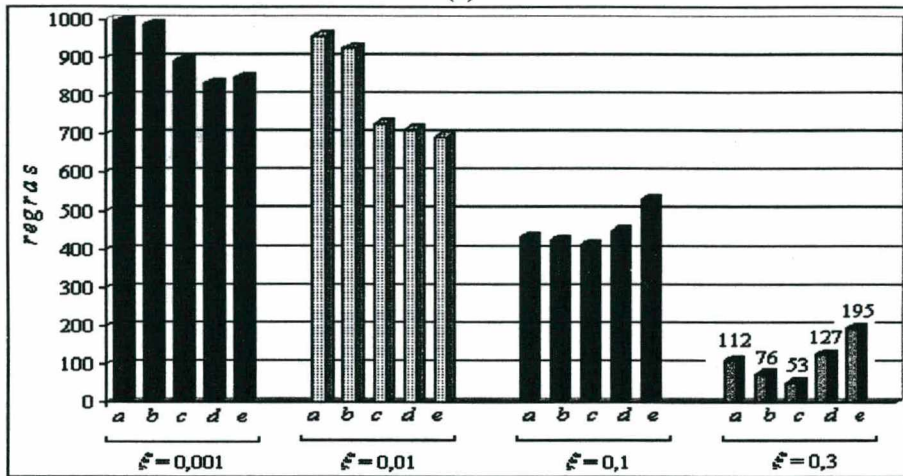
8.121.2 Identificação MISO do PBH via SNs

O SN utilizado aqui para identificação do PBH é o MLFE (LAUKONEN & PASSINO (1995). Os experimentos com o SN-MLFE servem para verificar o compromisso entre o erro de identificação e a complexidade computacional. Para o SN-MLFE são relacionadas à precisão através do ajuste dos termos W^f que regem a sobreposição entre as regras da vizinhança mais próxima. Um valor maior de W^f significa uma menor sobreposição das funções de pertinência e vice-versa.

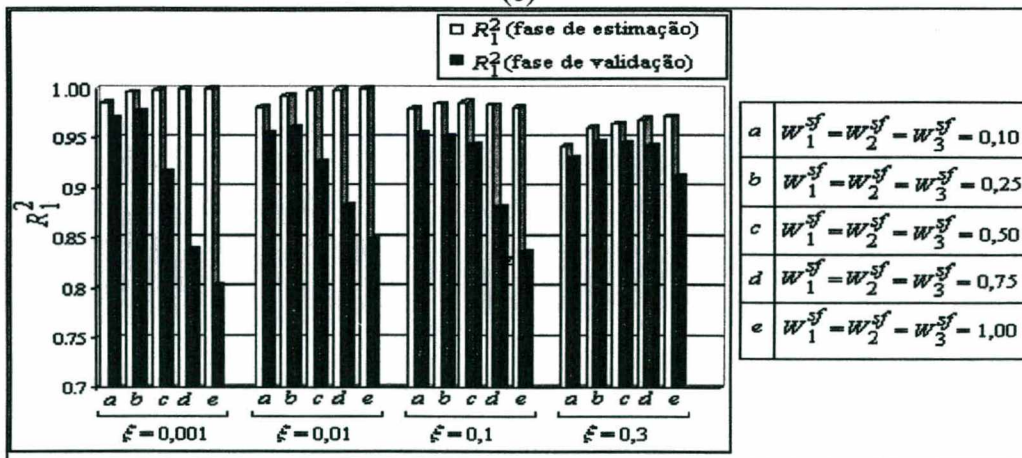
A complexidade computacional é proporcional ao número de regras nebulosas. O valor de ξ caracteriza a precisão desejada com que o SN executa a aproximação de funções, e, conseqüentemente, o número de regras, capacidade funcional e complexidade computacional do procedimento de inferência do MLFE. Os diversos experimentos realizados estão resumidos na figura 8.8.



(a)



(b)



(c)

Figura 8.8. Análise comparativa da identificação do PBH via SN-MLFE.

Os índices 1, 2 e 3 de W^{sf} apresentados na figura (8.8) significam a sobreposição das funções de pertinência para os antecedentes $u_1(k)$, $u_2(k)$ e $y_1(k-1)$, respectivamente. Analisando-se a figura 8.8(a), nota-se que quanto menor o ISE_1 , maior a capacidade do *SN-MLFE* interpolar (estimação adequada). Entretanto, a capacidade de extrapolar, dada pela fase de validação, é degradada de forma acentuada, tornando o *SN-MLFE* inadequado em generalizar e impróprio para aplicações em identificação. Nota-se pelos histogramas que uma melhor relação entre as capacidades de interpolar e validar do *SN-MLFE* é obtida com $\xi=0,3$.

A figura 8.8(b) mostra que um valor de $\xi \in [0,001; 0,1]$ demanda uma complexidade computacional elevada do procedimento de inferência do *SN-MLFE*, pois a quantidade de regras obtidas varia de 400 a 1000 regras, dependendo do ξ adotado. Note que a quantidade de amostras utilizadas, na fase de estimação, é 1000.

Pela análise da figura 8.8(c), conclui-se que o $W_1^{sf} = W_2^{sf} = W_3^{sf} = 0,50$ (para $\xi=0,3$) apresenta um compromisso adequado entre a precisão (fases de estimação e validação apresentando $R^2_1 \geq 0,900$) e a complexidade computacional (53 regras).

Os valores de ξ e W^{sf} são obtidos por heurísticas de tentativa e erro. Desta forma, não é garantida a obtenção de valores “ótimos”. Os melhores resultados da identificação do *PBH* via *SN-MLFE* para os casos de $\xi=0,03$, $\xi=0,1$, $\xi=0,01$ e $\xi=0,001$, são apresentados na figura 8.9.

Uma alternativa é a utilização de *AEs* para a obtenção de valores adequados para ξ e W^{sf} . Entretanto, no caso do *SN-MLFE*, o custo computacional é acentuado devido principalmente à necessidade de processar diversas vezes a equação (3.5) (toda a população de indivíduos, a cada geração). Esta tarefa torna-se crítica quando o número de regras é acentuado. Conseqüentemente, a utilização de *AEs* é uma alternativa a ser estudada, *a posteriori*, de forma a contornar este problema com um custo computacional menor e eficiente.

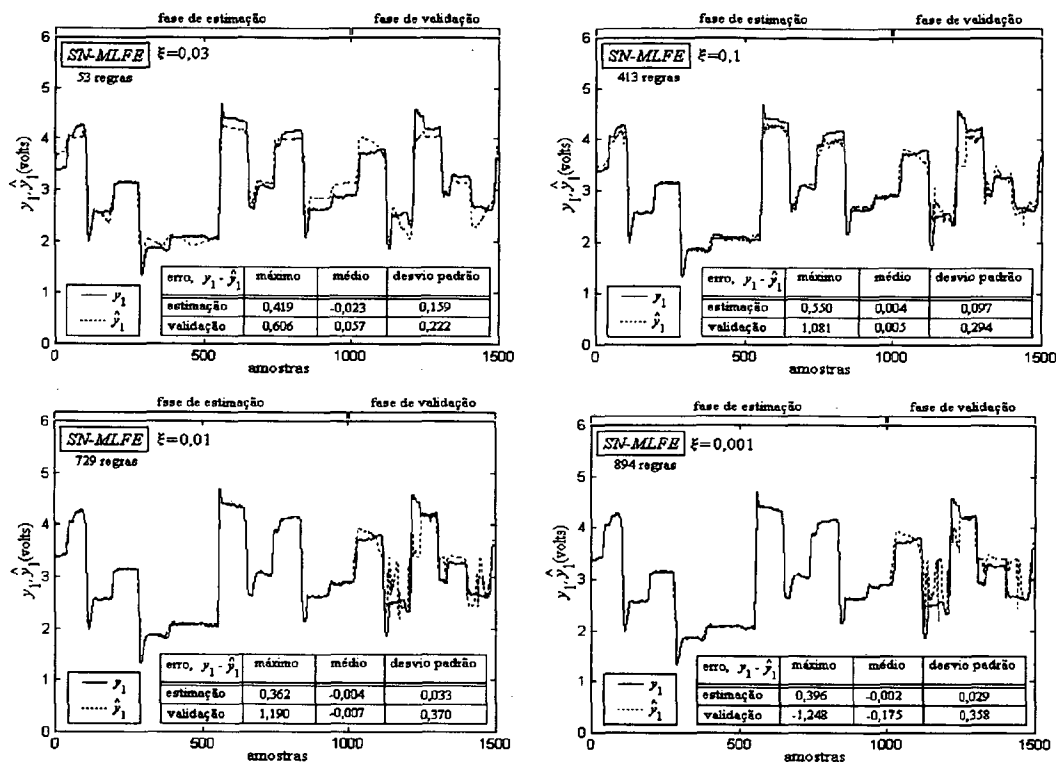


Figura 8.9. Resultados de identificação do PBH via SN-MLFE.

Pela figura 8.9, nota-se que os modelos de SN-MLFE apresentam comportamento excelente para a fase de identificação, onde a saída estimada praticamente sobrepõe a saída real do PBH. Em contrapartida, estes modelos apresentam inadequação quanto à fase de validação, caracterizando a ocorrência de sobre-treinamento.

8.12.1.3 Identificação MISO do PBH via RNs

A RN-MLP é configurada na forma de modelo série-paralelo, com 4 entradas $\{y_1(k-1), u_1(k), u_2(k), \varepsilon_1(k-1)\}$ e 1 saída $\{y_1(k)\}$. O número de neurônios testados, na camada oculta, são de 1 a 8 neurônios. Entretanto, é estipulado, heurísticamente, que 4 neurônios é um número apropriado para a obtenção de resultados em identificação do PBH via RN-MLP.

Os coeficientes de aprendizado e momento são 0,01 e 0,001 para treinamento via BP, respectivamente. Em alguns métodos heurísticos, as configurações iniciais dos

coeficientes de aprendizado e momento são alteradas no decorrer do procedimento de aprendizado, baseadas na informação do erro obtido, durante as épocas. Todas as configurações de *RNs-MLP* são treinadas adotando-se o mesmo conjunto de pesos iniciais e gerados aleatoriamente.

O número máximo de 200 épocas. Nas implementações, dos métodos heurísticos, de treinamento da *RN-MLP* foram adotadas as seguintes configurações:

- (i) *RN-MLP (JACOBS)*;
- (ii) *RN-MLP (VOGL)*;
- (iii) *RN-MLP (YPROP)*.

Os resultados da identificação do *PBH*, através de métodos heurísticos de treinamento da *RN-MLP*, são apresentados na tabela 8.2.

Tabela 8.2. Resultados da identificação do *PBH* via *RN-MLP* (métodos heurísticos).

<i>RN-MLP</i> critério	fase de estimação (amostra 1 a 1000)		fase de validação (amostra 1001 a 1500)	
	R^2_1	SSE_1	R^2_1	SSE_1
<i>RN-MLP(BP)</i>	0,937	55,628	0,715	274,273
<i>RN-MLP(JACOBS)</i>	0,907	67,782	0,888	105,664
<i>RN-MLP(VOGL)</i>	0,940	54,360	0,910	85,347
<i>RN-MLP(YPROP)</i>	0,947	44,796	0,835	113,310

Analisando a tabela 8.2, nota-se que a *RN-MLP(BP)* apresenta convergência lenta, pois com 200 épocas de treinamento não obteve valor de validação adequado para o *PBH*. As *RN-MLP(JACOBS)*, *RN-MLP(VOGL)* e *RN-MLP(YPROP)* apresentam resultados promissores na fase de estimação ($R^2_1 \geq 0,907$), todavia, apenas o modelo *RN-MLP(VOGL)* apresenta um resultado adequado, quanto à validação, com valor de $R^2_1 \geq 0,9$.

Na figura 8.10 é ilustrada a evolução, de cada algoritmo de treinamento, o *BP* e os dos métodos heurísticos de treinamento das *RNs-MLP* para a identificação do *PBH*. Melhores resultados podem ser obtidos com um projeto de *RN-MLP* apresentando um número maior de neurônios, na camada oculta e/ou a inserção de uma nova camada oculta e/ou quantidade maior de épocas para a *RN* convergir.

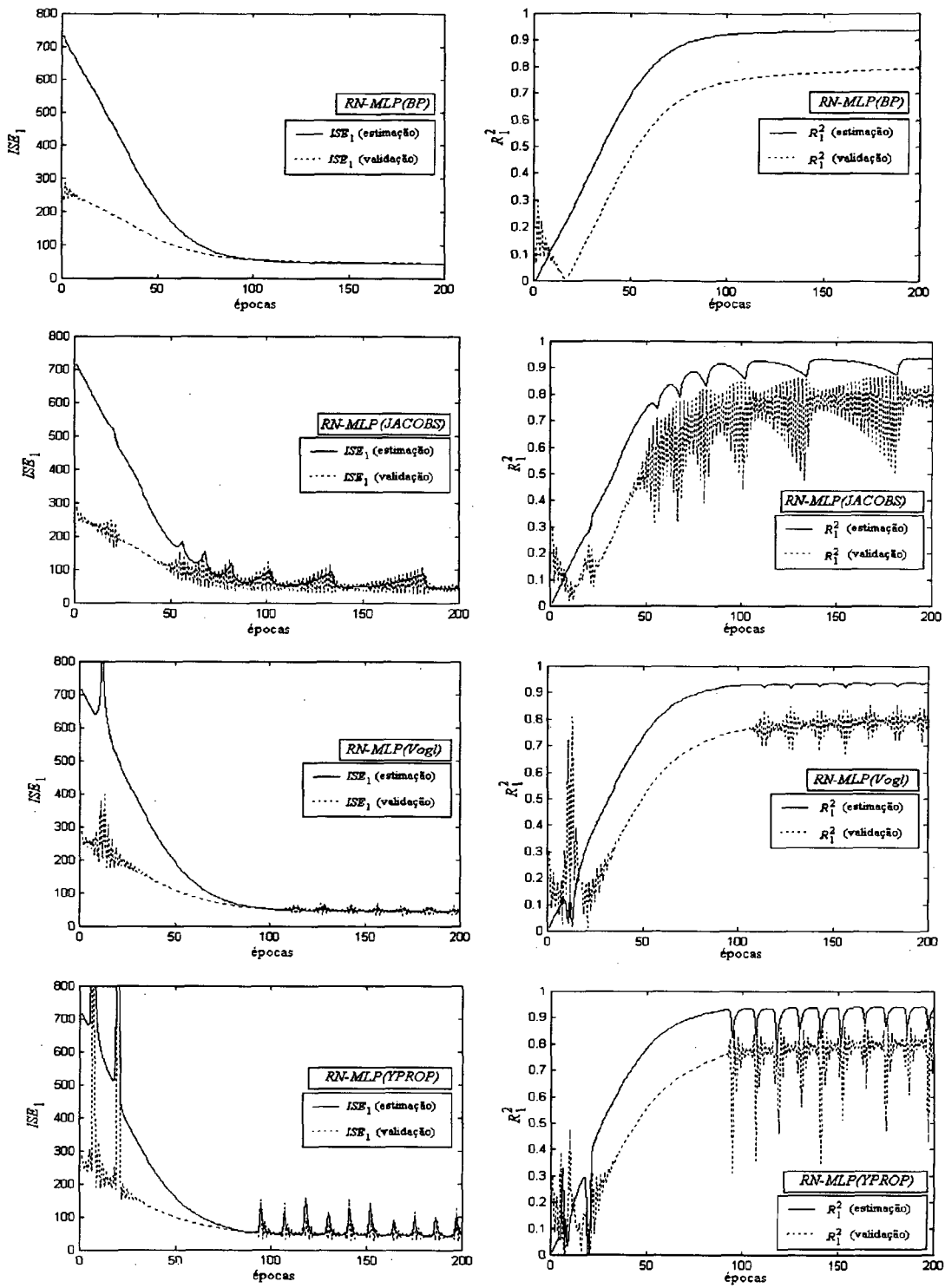


Figura 8.10. Evolução do aprendizado de métodos heurísticos de treinamento de *RNs-MLP* para identificação do *PBH*.

Pela tabela 8.2 nota-se que o algoritmo *BP* apresenta desempenho superior ao *RN-MLP(JACOBS)* e *RN-MLP(VOGL)*, e um desempenho similar ao *RN-MLP(YPROP)*. Na figura 8.10 observa-se que o desempenho do *RN-MLP(JACOBS)*, *RN-MLP(VOGL)* e *RN-MLP(YPROP)* é afetado pelas oscilações durante o aprendizado, o que requer uma análise mais detalhada e estudo heurístico para obtenção de uma configuração dos coeficientes “ideais”, vinculados ao aprendizado dos algoritmos de treinamento utilizados, sem oscilações ou pelo menos com poucas oscilações. A figura 8.11 apresenta os resultados de identificação do *PBH* via métodos heurísticos de treinamento da *RN-MLP*.

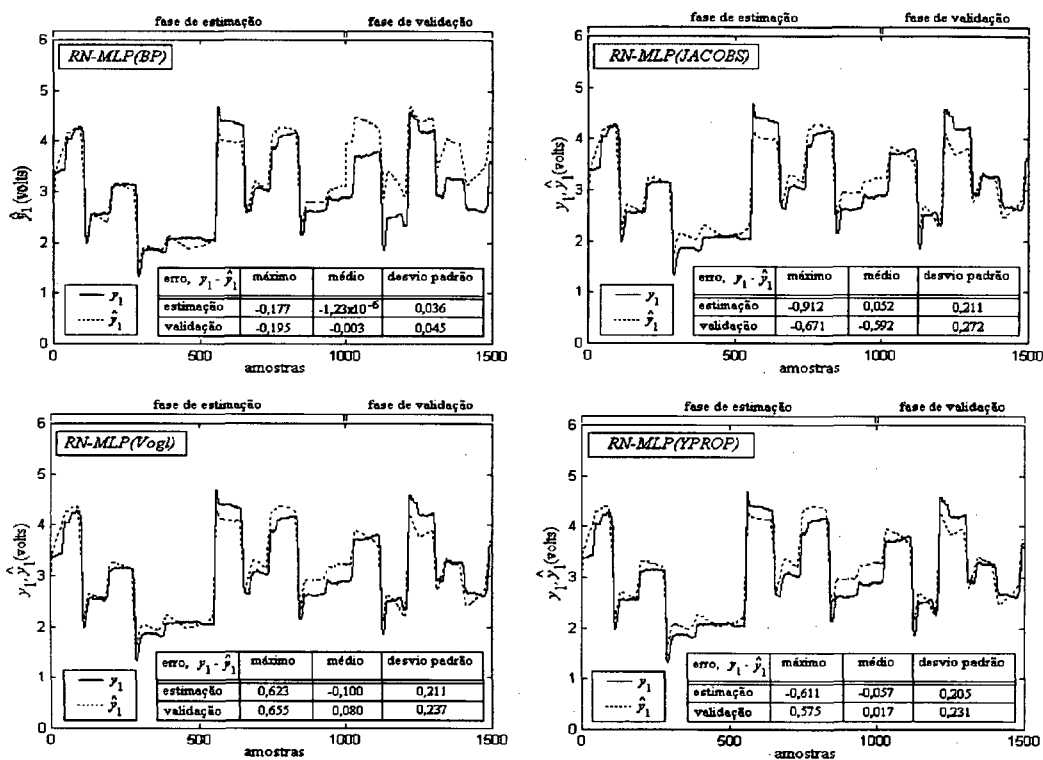


Figura 8.11. Resultados de identificação do *PBH* via *RN-MLP*.

Outra abordagem possível para o treinamento de *RNs-MLP* são os métodos de 2ª ordem. Na tabela 8.3 são apresentados os resultados da identificação do *PBH* via *RN-MLP*, através de métodos de treinamento de 2ª ordem.

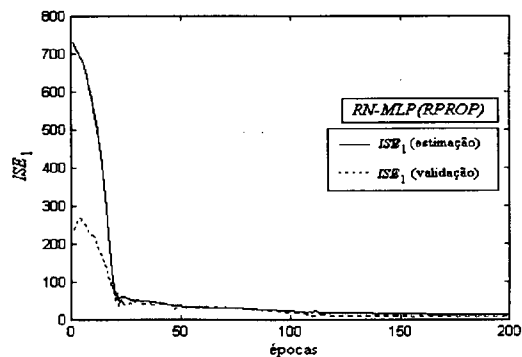
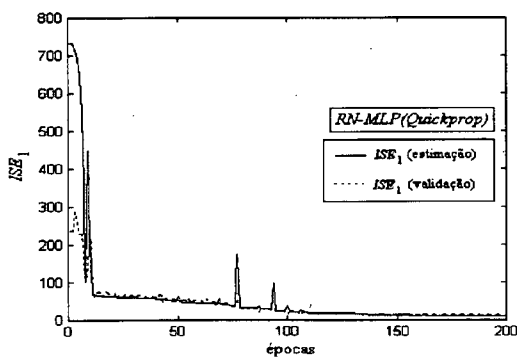
Tabela 8.3. Resultados da identificação do PBH via RN-MLP (métodos de 2ª ordem).

RN-MLP critério método	fase de estimação (amostra 1 a 1000)		fase de validação (amostra 1001 a 1500)	
	R^2_1	SSE_1	R^2_1	SSE_1
RN-MLP(Quickprop)	0,983	12,586	0,974	25,092
RN-MLP(RPROP)	0,982	13,061	0,980	18,736
RN-MLP(GC)	0,988	8,663	0,985	14,323
RN-MLP(SGC)	0,989	8,294	0,987	12,128
RN-MLP(QN)	0,990	7,165	0,989	10,894
RN-MLP(LM)	0,989	8,332	0,987	12,165

Os algoritmos de 2ª ordem considerados apresentaram desempenho excelente para identificação do PBH. O desempenho deve-se ao fato destes métodos serem robustos e eficientes na busca do menor erro que os métodos heurísticos. Nota-se, pela tabela 8.3, que:

- (i) para a fase de estimação: $0,982 \leq R^2_1 \leq 0,990$ e $7,165 \leq SSE_1 \leq 13,061$; e
- (ii) para a fase de validação: $0,974 \leq R^2_1 \leq 0,989$ e $10,894 \leq SSE_1 \leq 25,092$.

Apesar de todos os algoritmos de 2ª ordem proporcionarem resultados adequados ($R^2_1 \geq 0,974$), as RN-MLP(GC), RN-MLP(SGC), RN-MLP(QN) e RN-MLP(LM) apresentaram desempenho superior à RN-MLP(Quickprop) e RN-MLP(RPROP). Entretanto, o treinamento das RN-MLP(Quickprop) e RN-MLP(RPROP) demandam uma menor complexidade computacional. A figura 8.13 apresenta os resultados de identificação do PBH via métodos de 2ª ordem para o treinamento da RN-MLP.



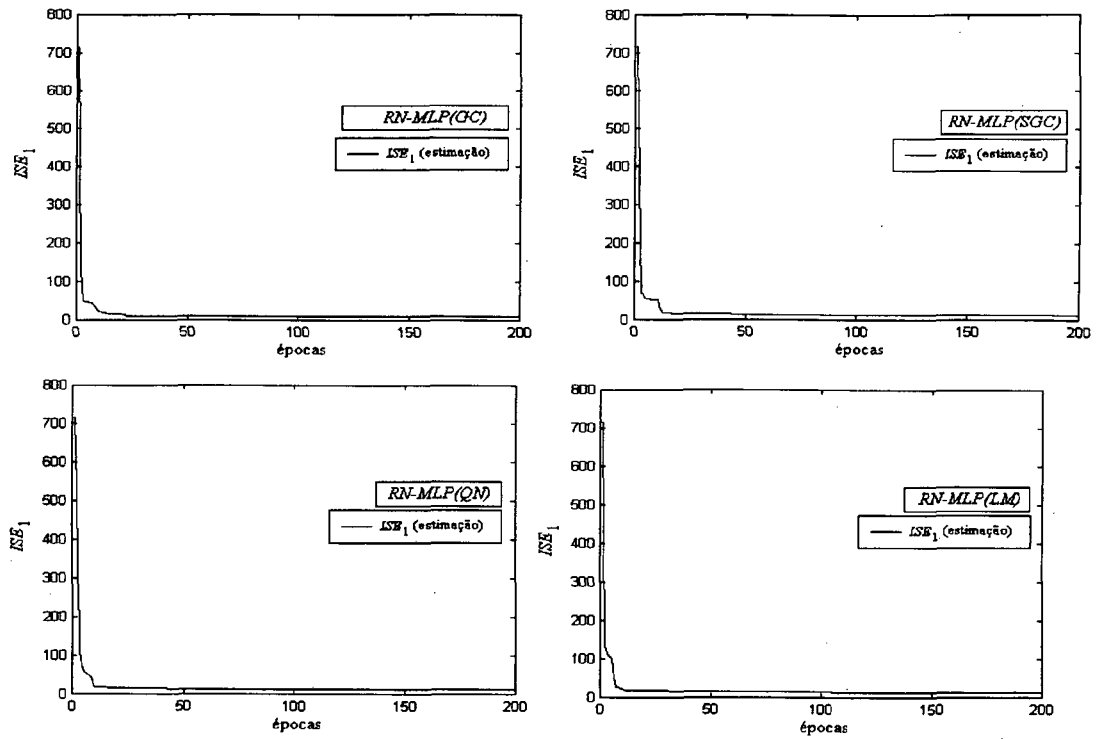
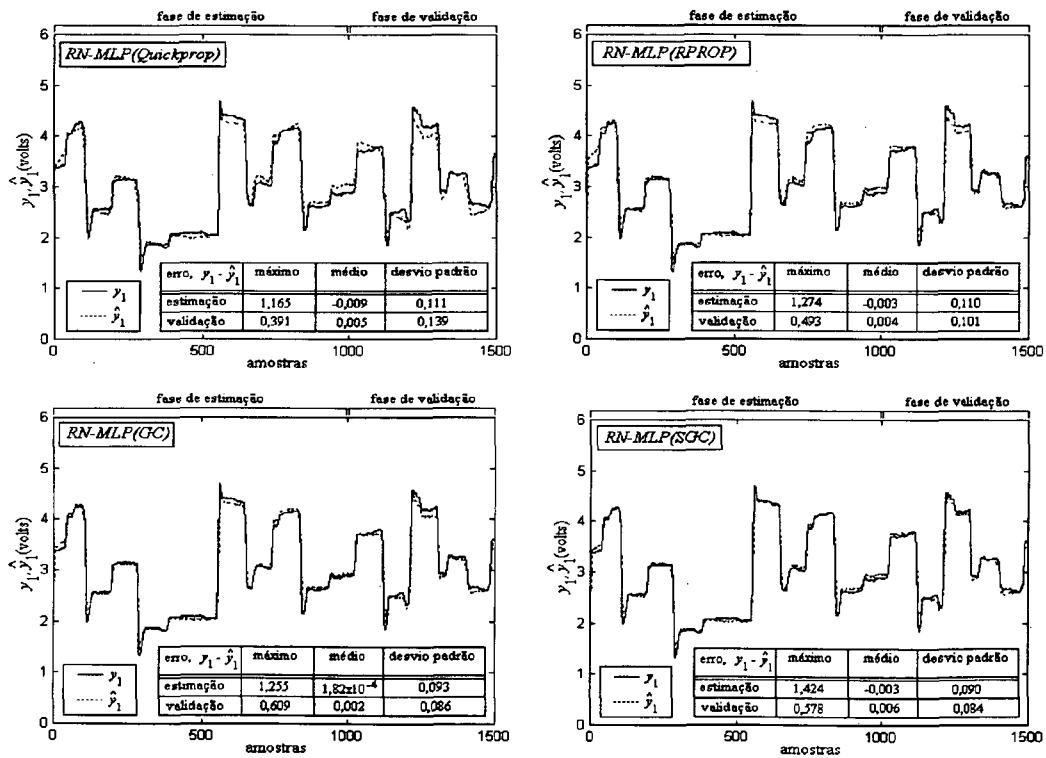


Figura 8.12. Evolução do aprendizado de métodos de 2ª ordem de treinamento de RNs-MLP para identificação do PBH.



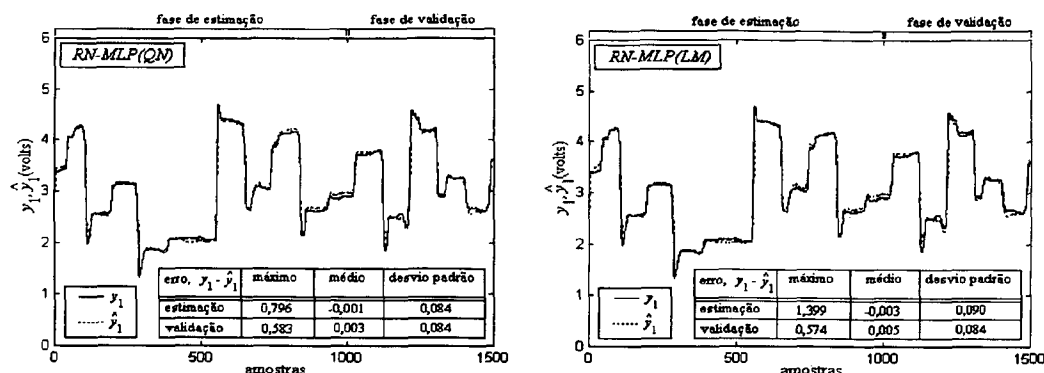


Figura 8.13. Resultados de identificação do PBH via RN-MLP (métodos de 2a. ordem).

A RN-GMDH é configurada para apresentar no máximo: 4 camadas hierárquicas, 6 EPs por camada e 2 entradas para cada Adaline. A RN-PREM tem a mesma configuração da RN-MLP, mas por tratar-se de uma RN parcialmente recorrente, apresenta o número de neurônios da camada oculta igual ao número de neurônios na camada de contexto. Os parâmetros relativos às auto-conexões, α , são iguais a 0,06. Os resultados da identificação do PBH via RN-PREM e RN-GMDH são apresentados na tabela 8.4 e na figura 8.14.

Tabela 8.4. Resultados da identificação do PBH via RN-PREM e RN-GMDH.

RNs método	fase de estimación (amostra 1 a 1000)		fase de validación (amostra 1001 a 1500)	
	R^2_1	SSE_1	R^2_1	SSE_1
RN-GMDH	0,982	13,591	0,980	17,227
RN-PREM	0,962	28,055	0,959	39,237

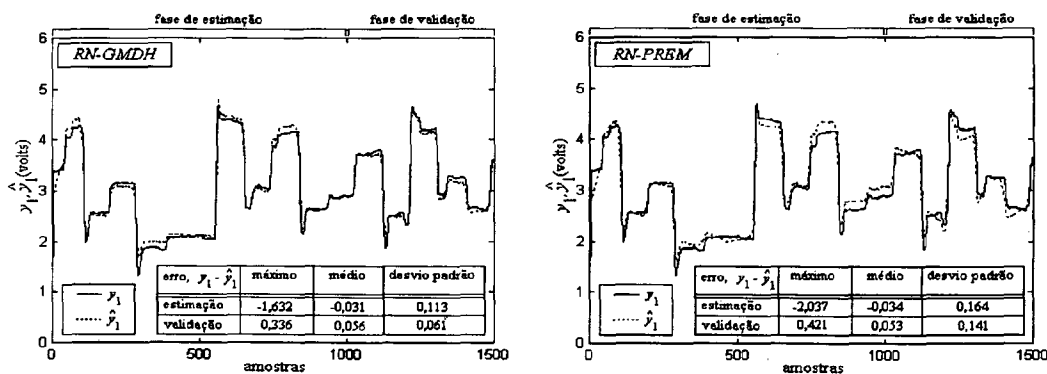


Figura 8.14. Resultados de identificação do PBH via RN-GMDH e RN-PREM.

Os resultados obtidos através das *RN-PREM* e *RN-GMDH* são adequados para a identificação do *PBH*, tanto em estimação quanto validação, pois $0,959 \leq R^2_1 \leq 0,982$ e $13,591 \leq SSE_1 \leq 39,237$.

Outra *RN* utilizada é *RN-RBF*. De forma similar à *RN-MLP*, a *RN-RBF* apresenta um modelo série-paralelo, com 4 entradas $\{y_1(k-1), u_1(k), u_2(k), \varepsilon_1(k-1)\}$, somente uma camada intermediária e 1 saída $\{y_1(k)\}$. Duas formas de treinamento são testadas. A primeira *RN-RBF*, denominada *RN-RBF(KM)*, tem o aprendizado realizado em duas etapas: i) ajuste dos centros via algoritmo *k-médias* e ajuste das variâncias via método *n-neighbor* (os valores de variâncias são multiplicados por um fator de 1,5); e ii) determinação dos pesos através do algoritmo da pseudo-inversa. A segunda *RN-RBF* é treinada através do algoritmo dos mínimos quadrados ortogonal (*RN-RBF(OLS)*). O número máximo de neurônios adotado para a camada intermediária da *RN-RBF(OLS)* é 12.

Na figura 8.15 é apresentada uma análise comparativa da identificação do *PBH* via *RN-RBF(KM)*. Nota-se que o melhor compromisso entre a precisão da identificação e a complexidade é obtido pela escolha de 4 funções Gaussianas, na camada oculta. A escolha de apenas 2 Gaussianas apresenta um valor de $R^2_1 \leq 0,870$, este um valor inadequado. No caso de 6 e 8 funções Gaussianas, na camada oculta da *RN-RBF(KM)* é obtido um valor de R^2_1 similar ao obtido com apenas 4 funções Gaussianas.

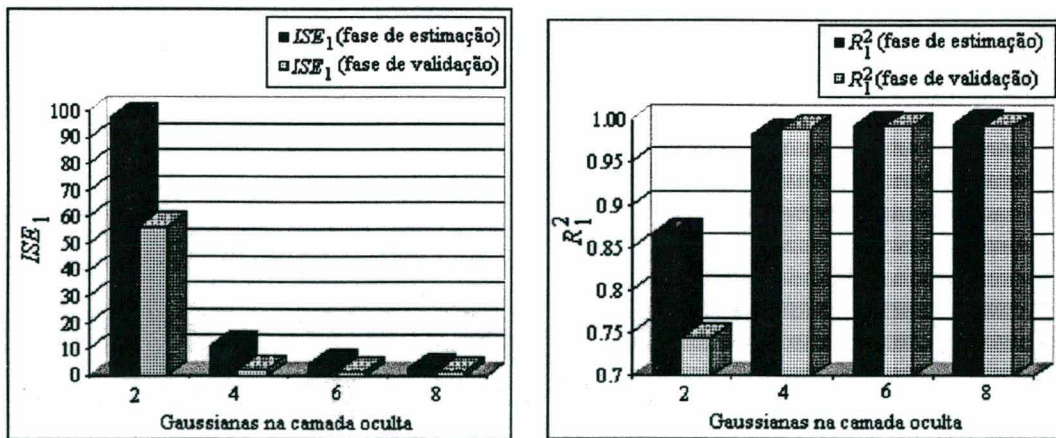


Figura 8.15. Análise comparativa de identificação do *PBH* via *RN-RBF(KM)*.

A *RN-RBF(OLS)* apresenta melhor configuração com 6 neurônios na camada intermediária. A tabela 8.5 e a figura 8.13 mostram os resultados de identificação do *PBH* via *RN-RBF(KM)*, com 4 Gaussianas, e *RN-RBF(OLS)*.

Tabela 8.5. Resultados da identificação do *PBH* via *RBF(KM)* e *RN-RBF(OLS)*.

<i>RN-RBF</i> critério método	fase de estimação (amostra 1 a 1000)		fase de validação (amostra 1001 a 1500)	
	R^2_1	SSE_1	R^2_1	SSE_1
<i>RN-RBF(KM)</i>	0,981	12,030	0,980	2,420
<i>RN-RBF(OLS)</i>	0,984	11,737	0,974	2,474

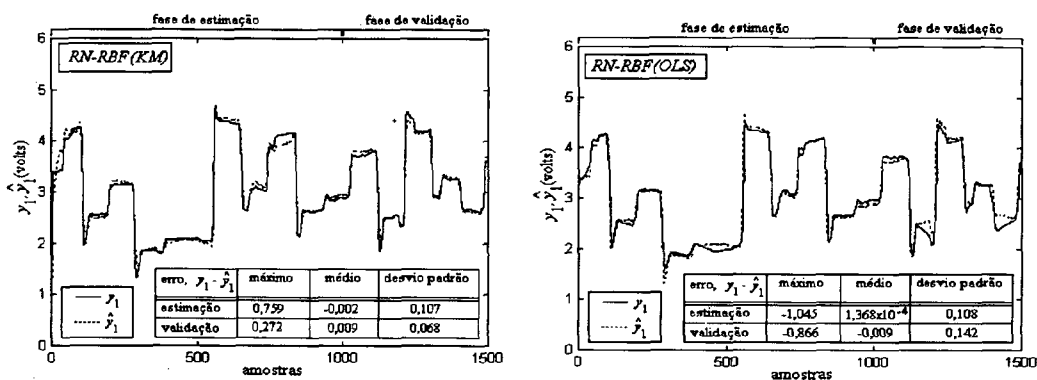


Figura 8.16. Resultado de identificação do *PBH* via *RN-RBF(KM)* e *RN-RBF(OLS)*.

8.12.1.4 Análise dos resultados para o estudo de caso 1

As metodologias da *IC* apresentaram resultados promissores da identificação, em malha fechada, do *PBH*. Isto pode ser verificado, no exemplo apresentado na figura 8.17, onde é apresentada a análise de autocorrelação e correlação cruzada, para o caso do modelo matemático *ARMAX*, obtido via *EE(1)*. As outras metodologias da *IC* apresentaram resultados similares, mas não são apresentados devido à quantidade de gráficos obtidos com estes testes. Através destes testes observa-se a necessidade de incluir-se termos de $a(k-1)$ como entradas do modelo.

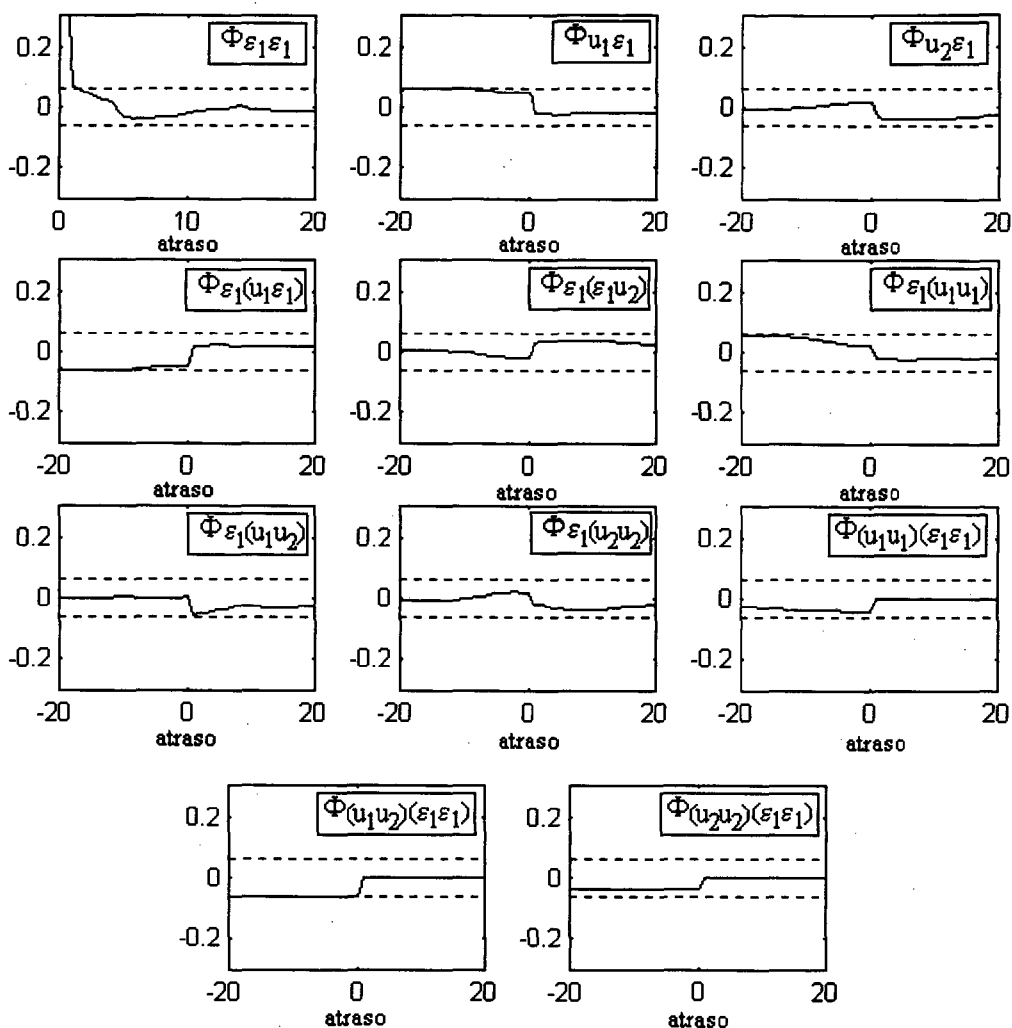


Figura 8.17. Análise de autocorrelação e correlação cruzada.

Um aspecto relevante na identificação com *RN-MLP* é a possibilidade da ocorrência de sobretreinamento. O procedimento para o tratamento deste problema, descrito no capítulo 4, foi utilizado nos algoritmos heurísticos e de 2ª ordem. Neste caso, a cada época o conjunto de dados de validação era analisado pela *RN-MLP*. Os pesos da *RN-MLP* são salvos (para posterior utilização, se necessária, quando da ocorrência de sobretreinamento), quando duas condições são satisfeitas:

- (i) na fase de estimação e validação satisfazem a inequação: $R^2_1 \geq 0,900$;
- (ii) o valor de validação obtido é melhor que o valor obtido previamente.

A ocorrência de sobre-treinamento não foi observada durante as 200 épocas estipuladas para o treinamento das *RN-MLP*. Entretanto, testes posteriores mostraram a ocorrência de sobre-treinamento quando o procedimento de treinamento é realizado para um número maior de épocas. Um exemplo, é o treinamento da *RN-MLP(Quickprop)* para 600 épocas. De acordo com a figura 8.18(a), a partir da época 440, o valor de R^2_1 , para a fase de validação, não foi melhorado até a época 600. Conseqüentemente, a *RN-MLP(Quickprop)*, após a realização de 600 épocas, apresenta ótima capacidade de interpolação, contudo, possui pobre capacidade de generalização, devido ao sobre-treinamento, conforme ilustrado pela figura 8.18(b).

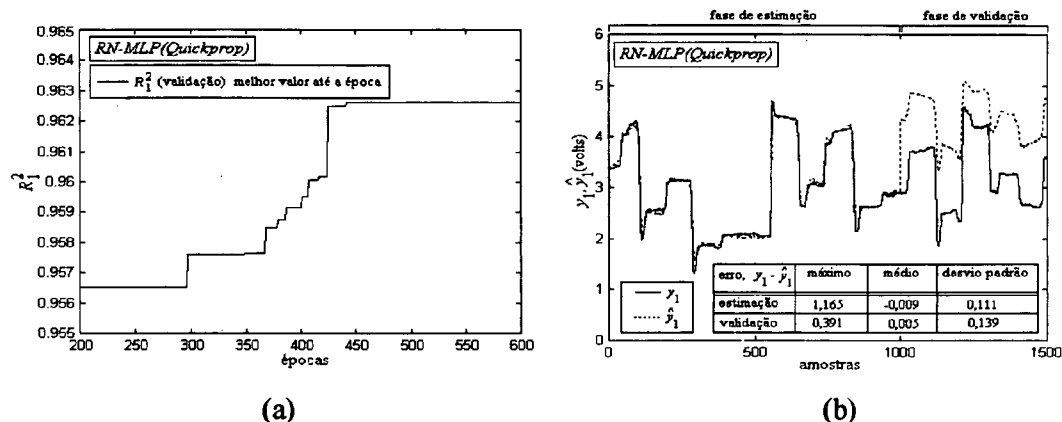


Figura 8.18. Exemplo da ocorrência de sobre-treinamento em *RN-MLP*.

8.12.2 Estudo de caso 2: Identificação de um manipulador robótico

As equações dinâmicas do robô são um conjunto de equações diferenciais acopladas contendo termos, tais como: inércia variante, torque centrífugo e Coriolis, termos de carregamento e gravidade. O movimento do efetuador final em uma trajetória particular com determinada velocidade requer um conjunto complexo de funções de torque para serem aplicadas aos atuadores nas juntas do manipulador robótico. A seguir, apresenta-se a descrição do modelo matemático do robô.

O modelo do manipulador usualmente considera o vetor de representação da dinâmica de um manipulador robótico de n -graus de liberdade (neste caso, $n=2$) regido pela equação:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (8.19)$$

onde $M(\theta) \in \mathbb{R}^{n \times n}$ é a matriz de inércia do sistema definida positiva, $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times 1}$ é um vetor que representa os efeitos do torque centrífugo e de Coriolis, $G(\theta) \in \mathbb{R}^{n \times 1}$ é o vetor relativo ao efeito dos torques gravitacionais, e $\tau \in \mathbb{R}^{n \times 1}$ é o vetor do torque da(s) junta(s). Os sinais θ , $\dot{\theta}$, $\ddot{\theta}$ são a posição angular, velocidade e aceleração da(s) junta(s). O modelo dinâmico para o manipulador robótico de dois graus de liberdade utilizado, conforme apresentado na figura 8.19, é dado por (CRAIG, 1986):

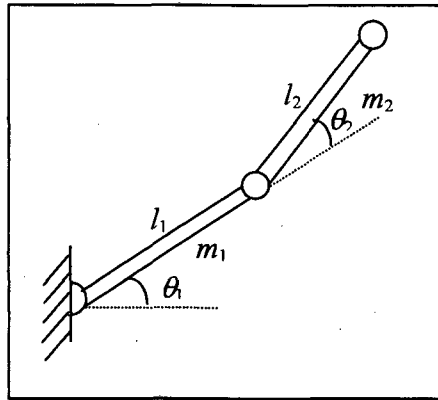


Figura 8.19. Representação do manipulador robótico com dois graus de liberdade.

$$\tau_1 = m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) + (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \quad (8.20)$$

$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \quad (8.21)$$

onde $s_1 = \text{sen}(\theta_1)$, $s_2 = \text{sen}(\theta_2)$, $c_1 = \text{cos}(\theta_1)$, $c_2 = \text{cos}(\theta_2)$, e $c_{12} = \text{cos}(\theta_1 + \theta_2)$. Os subscritos 1 e 2 denotam os parâmetros das juntas 1 e 2, respectivamente. Os parâmetros adotados nas simulações são comprimentos — $l_1 = 0,8$ m e $l_2 = 0,4$ m, massas — $m_1 = m_2 = 0,1$ kg, e aceleração da gravidade de $g = 9,81$ m/s² (MITAL & CHIN, 1995). As restrições impostas aos torques τ_1 e τ_2 são $[-1000; 1000]$ Nm.

O período de amostragem é $T_s = 10$ mseg e o tempo de simulação é de 20 seg (2000 amostras). Define-se $\theta_{d,j}$, $\dot{\theta}_{d,j}$, $\ddot{\theta}_{d,j}$ como os valores de posição angular, velocidade e

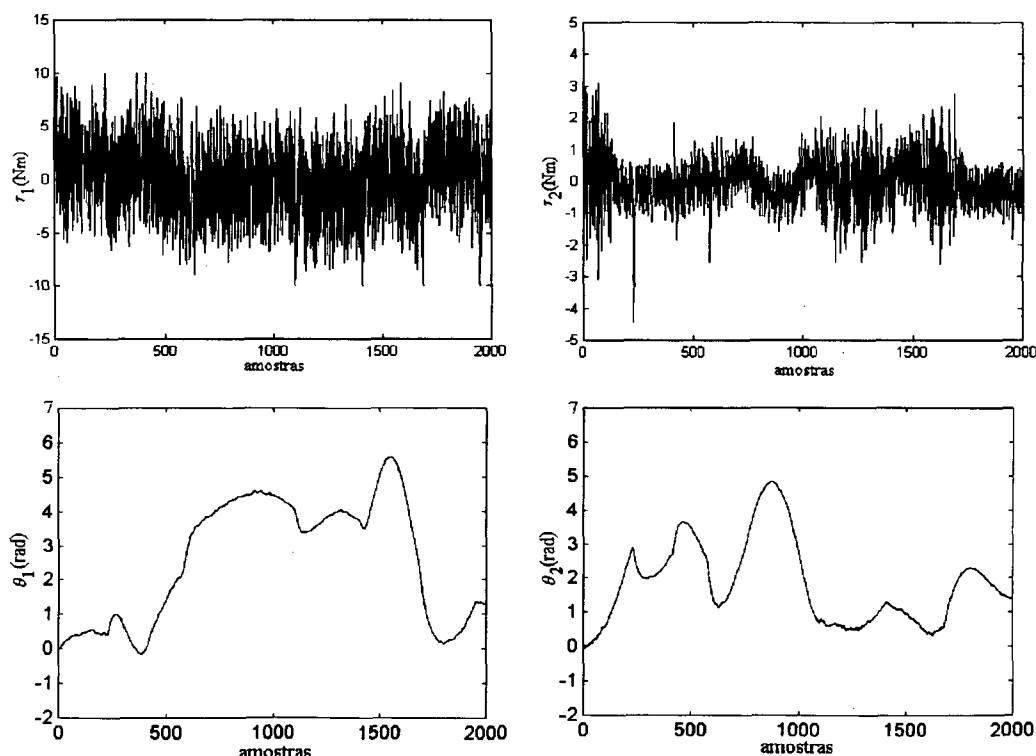
aceleração desejados para a junta robótica. Os vetores dos erros de posição, velocidade e aceleração são dados por

$$q_j(k) = \theta_j(k) - \theta_{a_j}(k), \text{ onde } j \in \mathcal{N} / j = \{1,2\} \quad (8.22)$$

$$\dot{q}_j(k) = \dot{\theta}_j(k) - \dot{\theta}_{a_j}(k), \text{ onde } j \in \mathcal{N} / j = \{1,2\} \quad (8.23)$$

A dinâmica do robô é simulada pelo método Runge-Kutta de 4ª ordem. Os torques e as saídas de posição e velocidade das juntas 1 e 2 do processo são apresentadas na figura 8.20. A obtenção dos dados é realizada pela utilização de dois controladores *PID multiloop* em malha fechada no controle do manipulador robótico. As trajetórias estipuladas para coleta dos dados foram baseadas nos estudos de BEHERA *et al.* (1996). Observação relevante é que os torques de entrada, utilizados nas simulações (figura 8.20), podem ocasionar desgaste dos atuadores em manipulador robóticos reais.

Os dados utilizados na fase de identificação são constituídos das amostras 1 a 1600, enquanto as amostras 1601 a 2000 são utilizadas na fase de validação do modelo matemático.



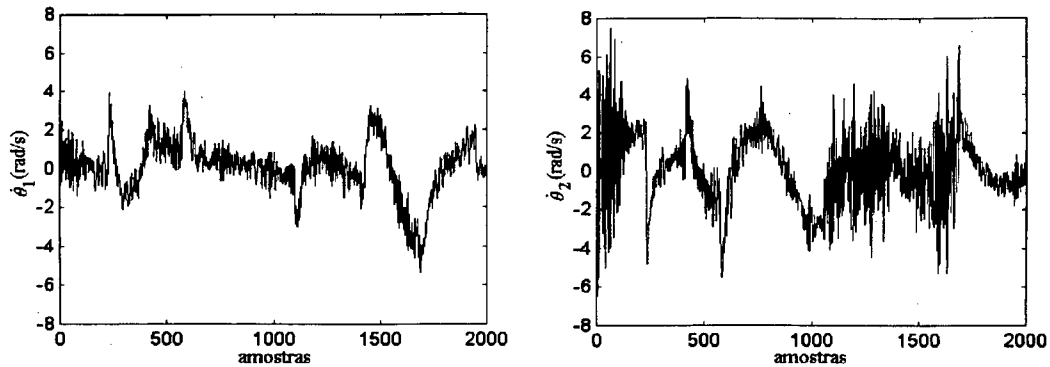


Figura 8.20. Torques e as saídas de posição e velocidade do manipulador robótico.

8.12.2.1 Identificação do manipulador robótico

O esquema de identificação, em malha fechada, da dinâmica do manipulador robótico é realizado com as seguintes metodologias da IC:

- (i) *RN-MLP(Quickprop)*;
- (ii) *RN-MLP(SCG)*;
- (iii) *RN-MLP(QN)*;
- (iv) *RN-RBF(KM)*;
- (v) *RN-RBF(OLS)*;
- (vi) *SHSNRN(TSK)*; e
- (vii) *SHRNAE*.

O algoritmos inteligentes dos itens (i) a (viii) possuem 6 entradas $\{\theta_1(k-1), \theta_2(k-1), \dot{\theta}_1(k-1), \dot{\theta}_2(k-1), \tau_1(k-1), \tau_2(k-1)\}$ e 4 saídas $\{\theta_1(k), \theta_2(k), \dot{\theta}_1(k), \dot{\theta}_2(k)\}$.

O algoritmos inteligentes dos itens (i) a (viii) possuem 6 neurônios, na camada de entrada, $\{\theta_1(k-1), \theta_2(k-1), \dot{\theta}_1(k-1), \dot{\theta}_2(k-1), \tau_1(k-1), \tau_2(k-1)\}$ e 4 neurônios na camada de saída, $\{\theta_1(k), \theta_2(k), \dot{\theta}_1(k), \dot{\theta}_2(k)\}$.

As *RN-MLP(Quickprop)*, *RN-MLP(SCG)* e *RN-MLP(QN)* são projetadas para possuírem 8 neurônios, na camada oculta, com função de ativação regidas por tangentes hiperbólicas, e coeficientes de aprendizado e momento de 0,005 e 0,001, respectivamente. Quanto as *RN-RBF(KM)*, esta apresenta 8 funções Gaussianas na camada intermediária. A *RN-RBF(OLS)* pode apresentar no máximo 20 funções Gaussianas, na camada intermediária, durante o ciclo de otimização através do *OLS*.

Os sistemas híbridos inteligentes abordados são: *SHSNRN(TSK)* e *SHRNAE*. O *SHSNRN(TSK)* utiliza treinamento das premissas realizado através do método do gradiente descendente, enquanto para o aprendizado dos consequentes é utilizado o *MQR*. O *SHRNAE* constitui-se de um algoritmo para aprimorar o desempenho obtido pela *RN-RBF(KM)* com 8 funções Gaussianas, na camada intermediária, através do ajuste das variâncias das Gaussianas por um *AE* do tipo (1,29)-*EEMC*. O ciclo evolutivo é de 50 gerações e o primeiro indivíduo da população (inicial) constitui-se de valores obtidos com o método *KM*. Os resultados com diversos algoritmos inteligentes, em identificação em malha fechada do manipulador robótico, são apresentados nas figuras 8.21 a 8.27.

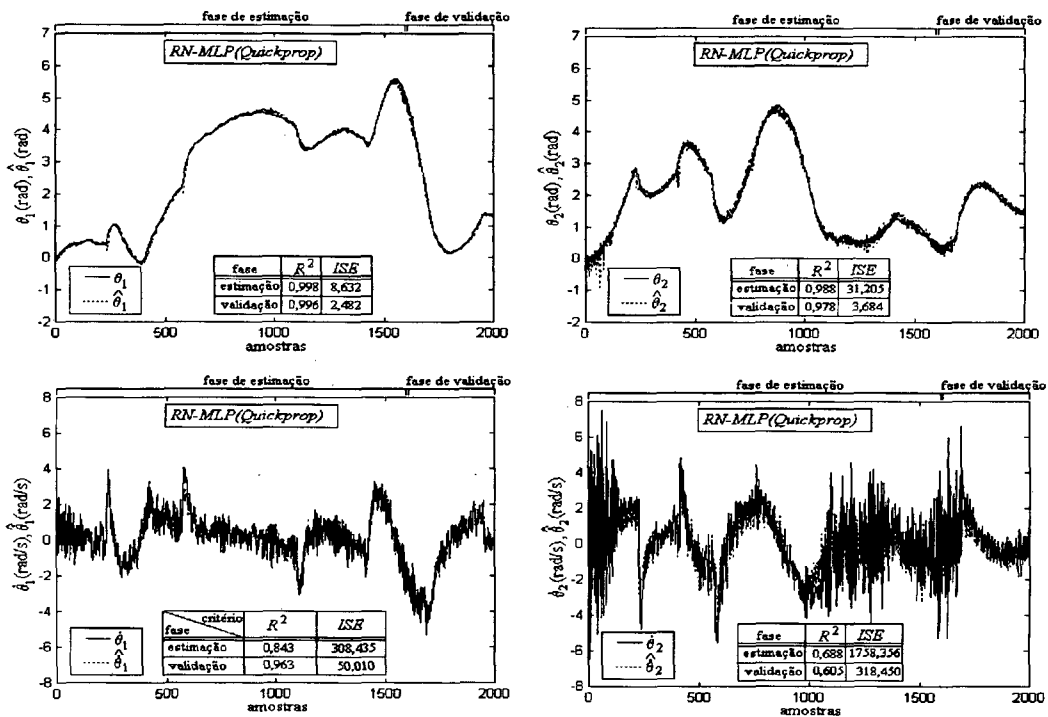


Figura 8.21. Posições e velocidades estimadas do robô via *RN-MLP(Quickprop)*.

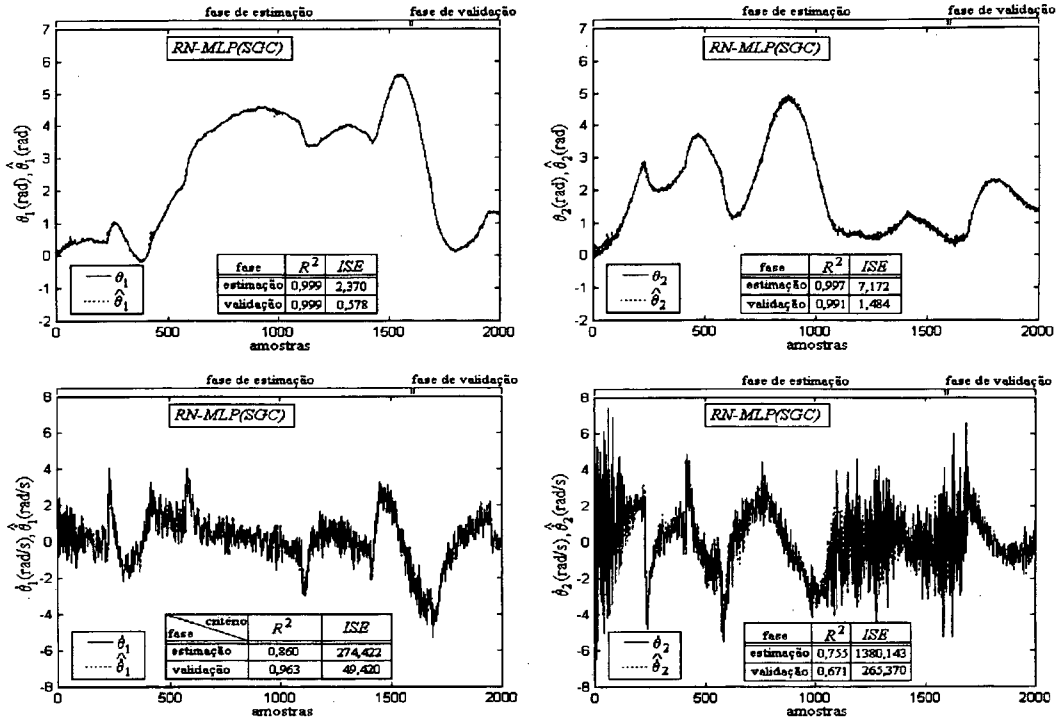


Figura 8.22. Posições e velocidades estimadas do robô via $RN-MLP(SGC)$.

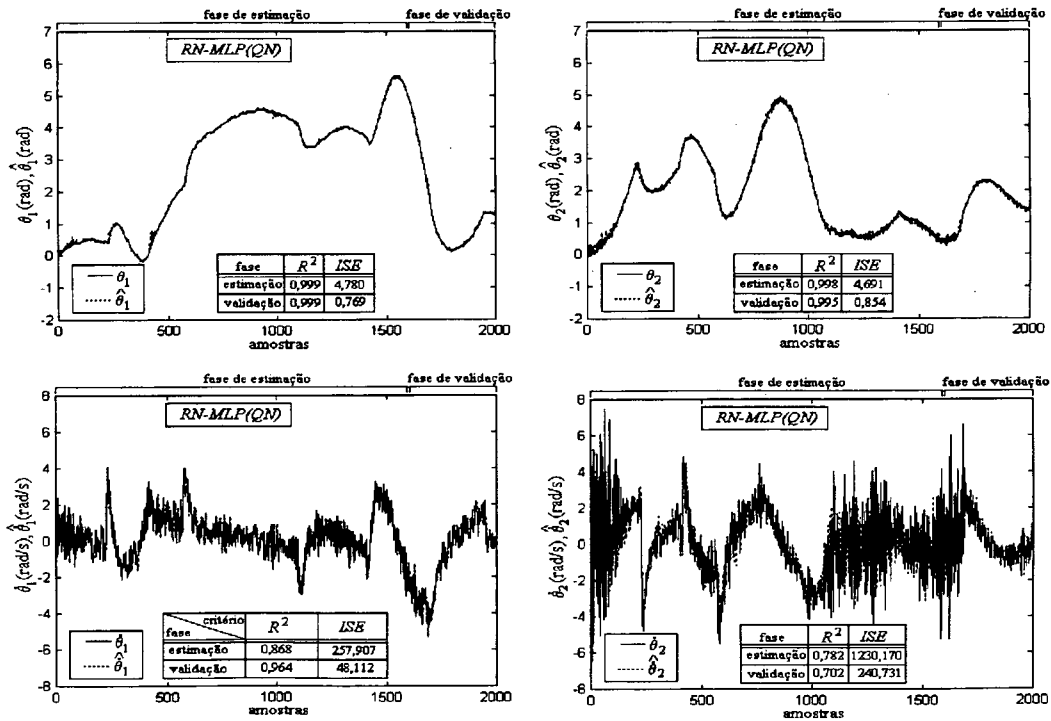


Figura 8.23. Posições e velocidades estimadas do robô via $RN-MLP(QN)$.

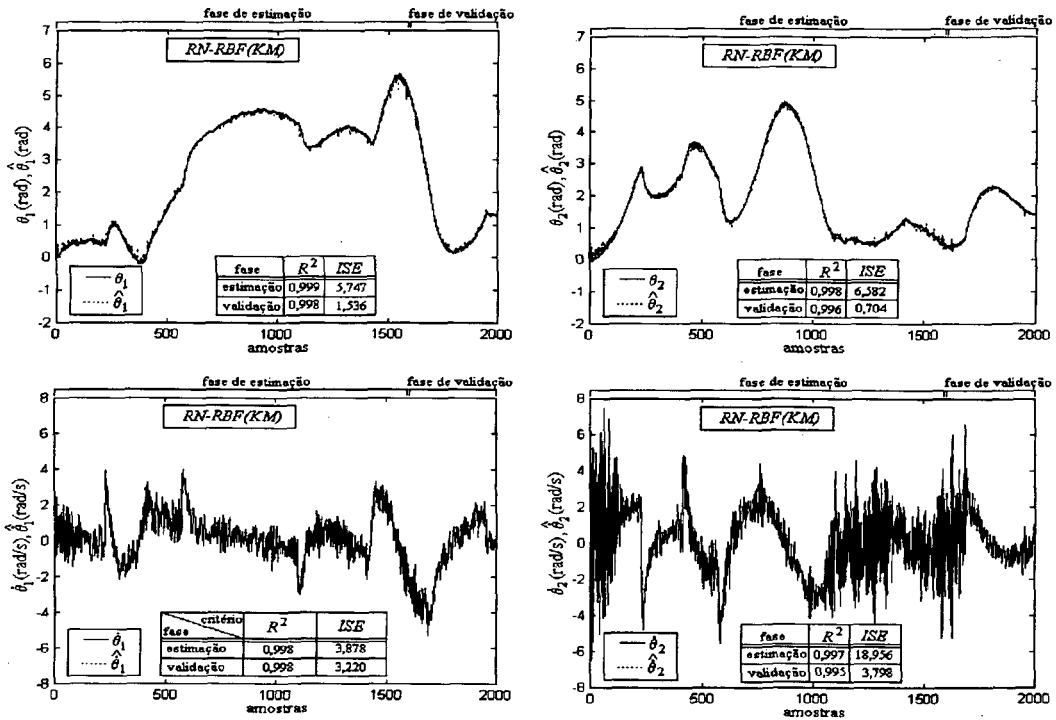


Figura 8.24. Posições e velocidades estimadas do robô via RN-RBF(KM).

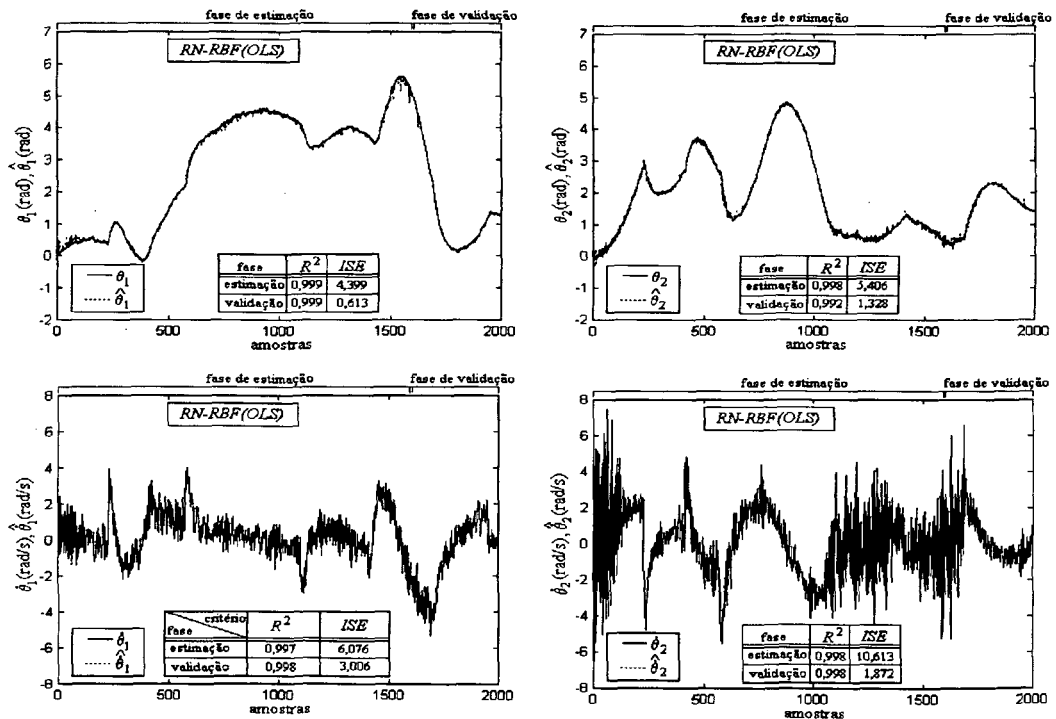


Figura 8.25. Posições e velocidades estimadas do robô via RN-RBF(OLS).

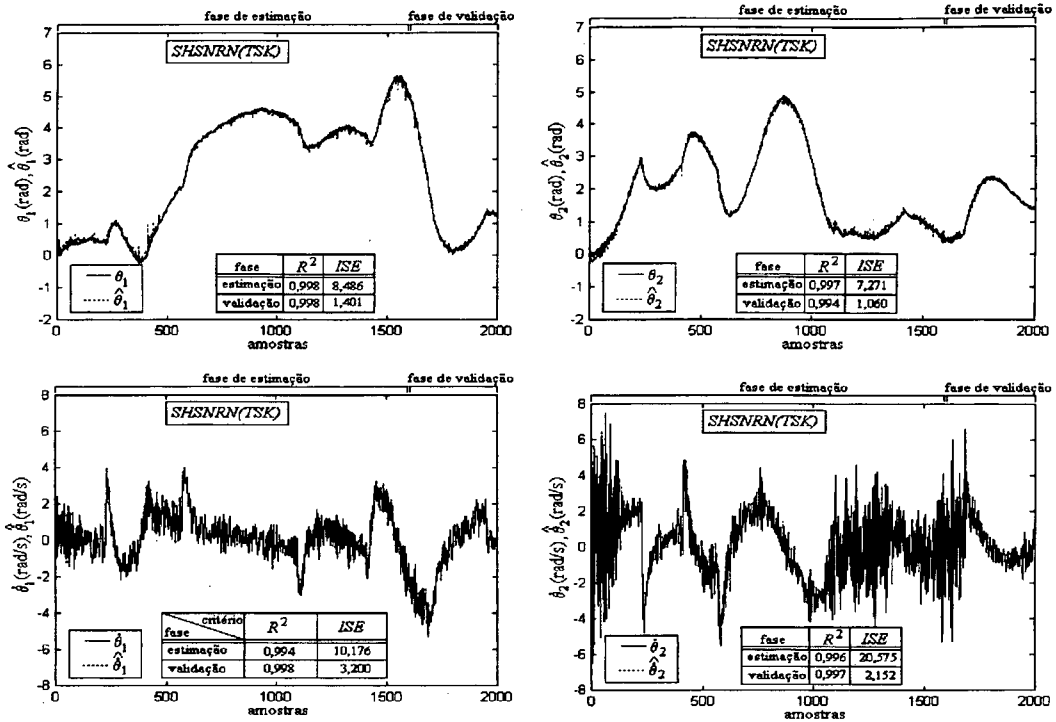


Figura 8.26. Posições e velocidades estimadas do robô via SHSNRN(TSK).

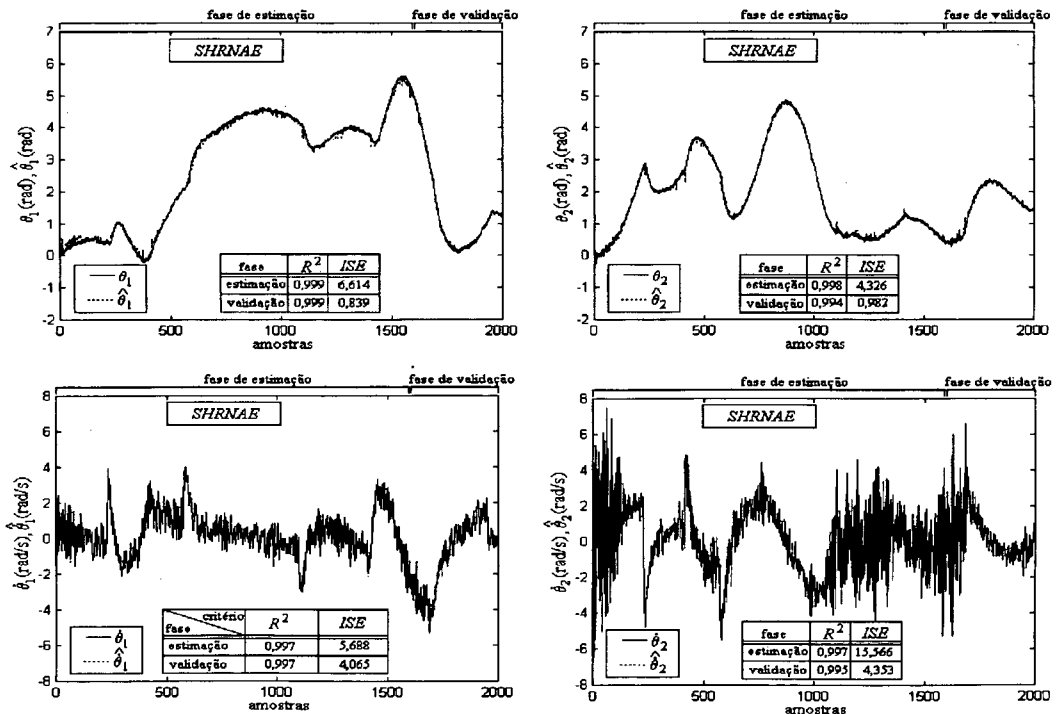


Figura 8.27. Posições e velocidades estimadas do robô via SHRNAE.

Da análise dos resultados conclui-se que as *RN-MLP(Quickprop)*, *RN-MLP(SGC)* e *RN-MLP(QN)* apresentam excelentes resultados com a identificação de três (das quatro) saídas do manipulador robótico, $\{\theta_1(k), \theta_2(k), \dot{\theta}_1(k)\}$, com valores de R^2_1 próximos de 1. Entretanto, os resultados para a saída de velocidade, $\dot{\theta}_2(k)$, os resultados não são adequados com valores de $0,688 \leq R^2_1 \leq 0,782$ (fase de estimação) e $0,605 \leq R^2_1 \leq 0,702$ (fase de validação). As *RN-RBF(KM)* e *RN-RBF(OLS)* apresentam resultados adequados para todas as saídas de posições e velocidades do manipulador robótico. Nota-se pelas figuras 8.24 e 8.25, que os valores de R^2_1 estão muito próximos da unidade. A configuração final da *RN-RBF(OLS)* para os resultados apresentados é com 8 funções Gaussianas, na camada intermediária. Os resultados com as técnicas híbridas, *SHSNRN(TSK)* e *SHRNAE*, apresentam desempenho similar aos obtidos pelas *RN-RBF(KM)* e *RN-RBF(OLS)*. Quanto à otimização, o comportamento das *RNs-RBF* e algoritmos híbridos inteligentes apresentam aprendizado rápido e maior precisão que as *RNs-MLP*.

A literatura tem mencionado a relevância de *RN-RBF* em aplicações de identificação de sistemas e robótica devido à rapidez do treinamento e a precisão apresentada (GOTO *et al.*, 1996; BEHERA *et al.*, 1996). Outra característica a ser enfatizada é o mapeamento local realizado pelas *RN-RBF(KM)*, *RN-RBF(OLS)* e *SHRNAE*, enquanto as *RN-MLP* apresentam mapeamento global são treinadas por otimização não-linear. Em estudos futuros, os resultados de $\dot{\theta}_2(k)$ devem ser analisados em detalhes visando a melhoria do desempenho das *RNs-MLP*. O projeto e precisão dos modelos obtidos pelas *RNs-MLP* pode ser aprimorado com a inserção de mais uma camada oculta, com a desvantagem de um custo computacional acentuado para o treinamento.

8.13. Identificação em malha aberta

8.13.1 Estudo de caso 1: Identificação de um processo *MIMO* não-linear

Supõe-se que um processo seja descrito por um sistema não-linear *MIMO* desconhecido, descrito pelas seguintes equações (NARENDRA & MUKHOPADHYAY, 1993):

$$x_1(k+1) = 0.9x_1(k)\sin[x_2(k)] + \left[2 + 1.5 \frac{x_1(k)u_1(k)}{1+x_1^2(k)u_1^2(k)} \right] u_1(k) + \left[x_1(k) + \frac{2x_1(k)}{1+x_1^2(k)} \right] u_2(k) \tag{8.24}$$

$$x_2(k+1) = x_3(k) \left[1 + \sin[4x_3(k)] + \frac{x_3(k)}{1+x_3^2(k)} \right] \tag{8.25}$$

$$x_3(k+1) = [3 + \sin[2x_1(k)]]u_2(k) \tag{8.26}$$

$$y_1(k) = x_1(k); \quad y_2(k) = x_2(k) \tag{8.27}$$

O sinal utilizado para excitar as não-linearidades do processo, sob a região de interesse, é gerado pela composição de entradas aleatórias com distribuição uniforme e sinais de resposta ao degrau, onde $u_1(k), u_2(k) \in [-0.8, 0.8]$. Os conjuntos de dados gerados são das amostras 1 a 1000, para a fase de estimação, e amostras 1001 a 1500, para serem empregadas na fase de validação, apresentados na figura 8.28 (COELHO & COELHO, 2000a).

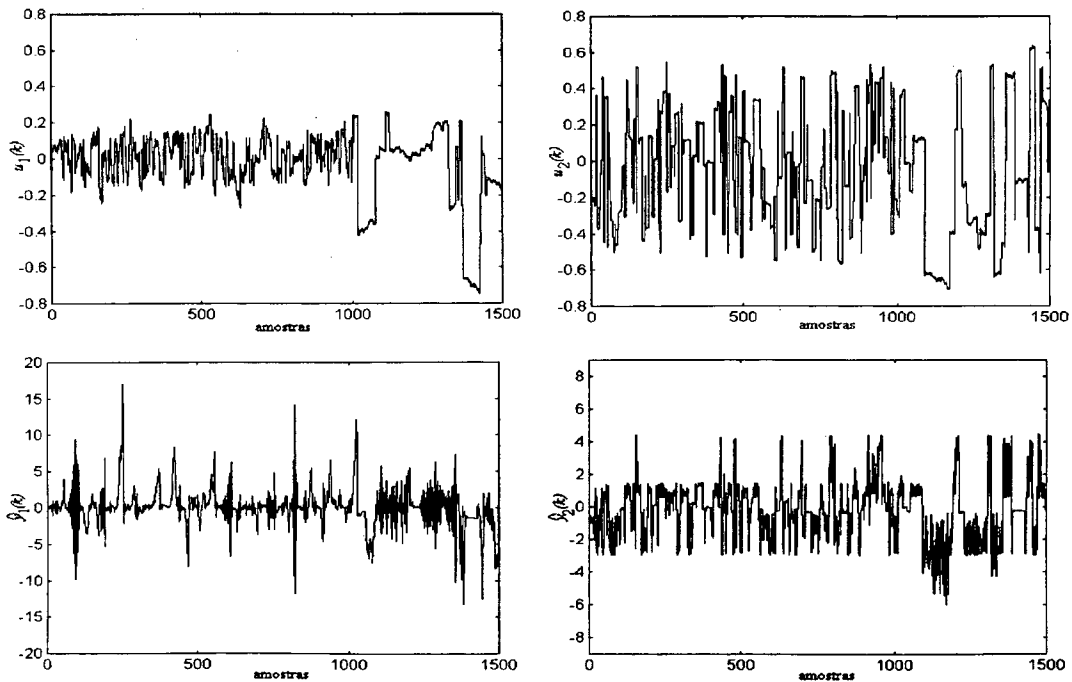


Figura 8.28. Dados de entrada e saída do processo MIMO não-linear.

Os algoritmos de identificação analisados são os mesmos do caso do manipulador robótico. A diferença é que, neste estudo de caso, os algoritmos inteligentes são configurados com 4 entradas $\{y_1(k-1), y_2(k-1), u_1(k-1), u_2(k-1)\}$ e 2 saídas $\{y_1(k), y_2(k)\}$.

As *RN-MLP(Quickprop)*, *RN-MLP(SCG)* e *RN-MLP(QN)* são projetadas com 4 neurônios, na camada oculta, e coeficientes de aprendizado e momento de 0,01 e 0,001, respectivamente. Quanto às *RN-RBF(KM)*, estas apresentam 4 funções Gaussianas, na camada intermediária. A *RN-RBF(OLS)* pode apresentar no máximo 12 funções Gaussianas, na camada intermediária.

O *SHSNRN(TSK)*, de forma similar ao caso do manipulador robótico, utiliza também treinamento das premissas realizado através do método do gradiente descendente, e dos conseqüentes via *MQR*. O *SNRHAЕ* constitui-se de um algoritmo para aprimorar o desempenho obtido pela *RN-RBF(KM)* com 4 funções Gaussianas e utilizando (1,29)-*EEMC*. Na tabela 8.5 são apresentados os resultados na identificação, em malha aberta, do processo *MIMO* não-linear.

Tabela 8.6. Resultados da identificação do processo *MIMO* via *IC*.

<i>IC</i>	fase de estimação (amostra 1 a 1000)		fase de validação (amostra 1001 a 1500)	
	<i>R</i> ² ₁	<i>SSE</i> ₁	<i>R</i> ² ₁	<i>SSE</i> ₁
<i>critério</i>				
<i>método</i>				
<i>RN-MLP(Quickprop)</i>	0,779	1236,403	0,350	3228,027
<i>RN-MLP(SCG)</i>	0,876	1036,462	0,400	3110,428
<i>RN-MLP(QN)</i>	0,930	381,909	0,905	470,168
<i>RN-RBF(KM)</i>	0,999	6,123	0,997	15,229
<i>RN-RBF(OLS)</i>	1,000	0,145	1,000	0,046
<i>SHSNRN(TSK)</i>	0,999	3,107	0,999	4,215
<i>SHRNAE</i>	1,000	0,161	1,000	0,060
<i>critério</i>				
<i>método</i>				
<i>RN-MLP(Quickprop)</i>	0,622	846,462	0,677	714,528
<i>RN-MLP(SCG)</i>	0,681	804,504	0,690	702,052
<i>RN-MLP(QN)</i>	0,600	893,364	0,625	828,726
<i>RN-RBF(KM)</i>	0,999	4,068	0,999	2,093
<i>RN-RBF(OLS)</i>	0,999	1,098	0,999	0,492
<i>SHSNRN(TSK)</i>	0,999	3,727	0,999	2,002
<i>SHRNAE</i>	0,999	2,130	0,999	1,332

Nas figuras 8.29 a 8.35, os resultados obtidos com os algoritmos inteligentes apresentados (tabela 8.6), em identificação do processo *MIMO* são ilustrados.

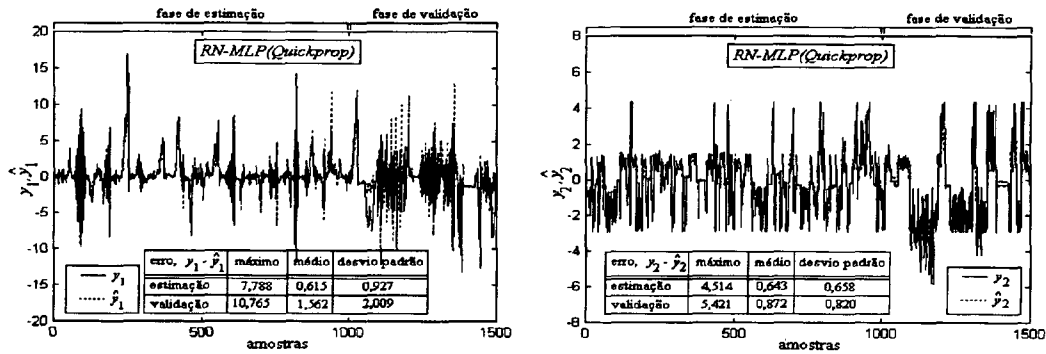


Figura 8.29. Resultados de identificação do processo *MIMO* via *RN-MLP(Quickprop)*.

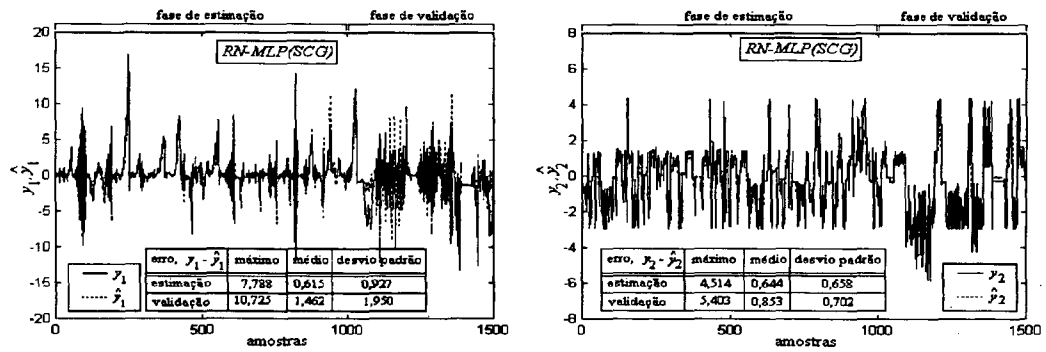


Figura 8.30. Resultados de identificação do processo *MIMO* via *RN-MLP(SCG)*.

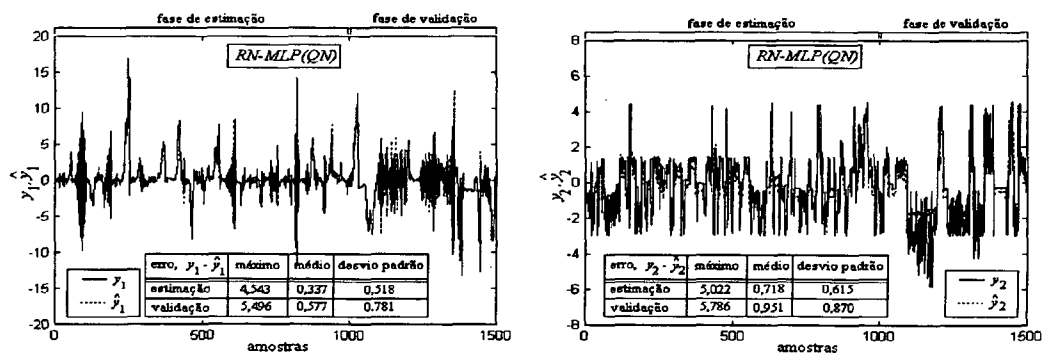


Figura 8.31. Resultados de identificação do processo *MIMO* via *RN-MLP(QN)*.

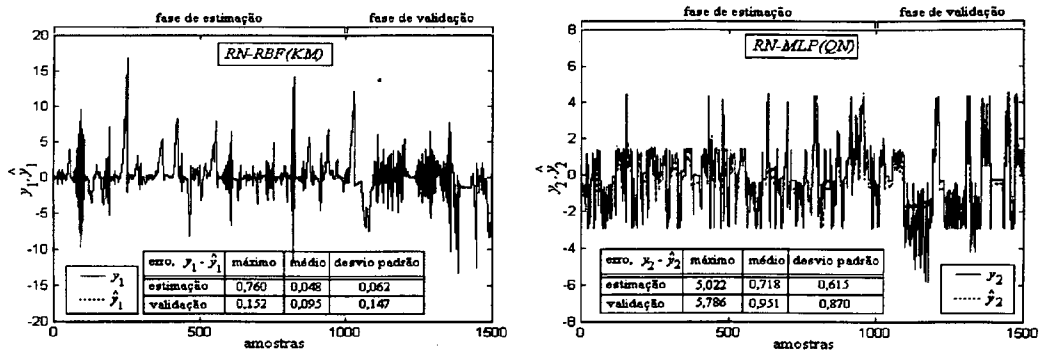


Figura 8.32. Resultados de identificação do processo MIMO via RN-RBF(KM).

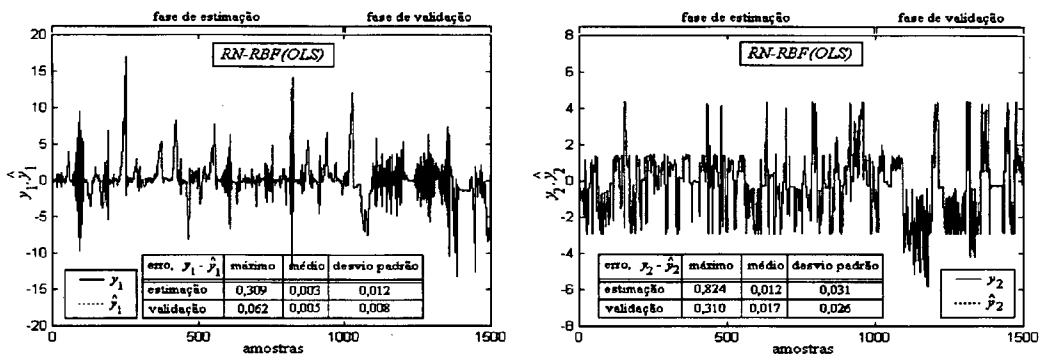


Figura 8.33. Resultados de identificação do processo MIMO via RN-RBF(OLS).

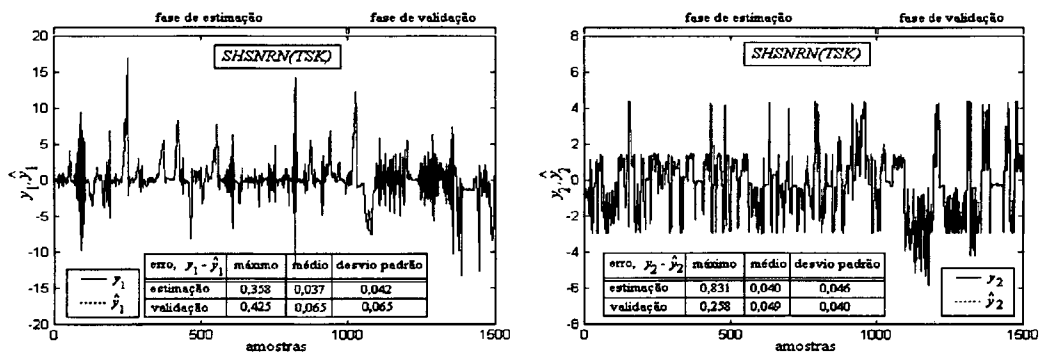


Figura 8.34. Resultados de identificação do processo MIMO via SHSNRN(TSK).

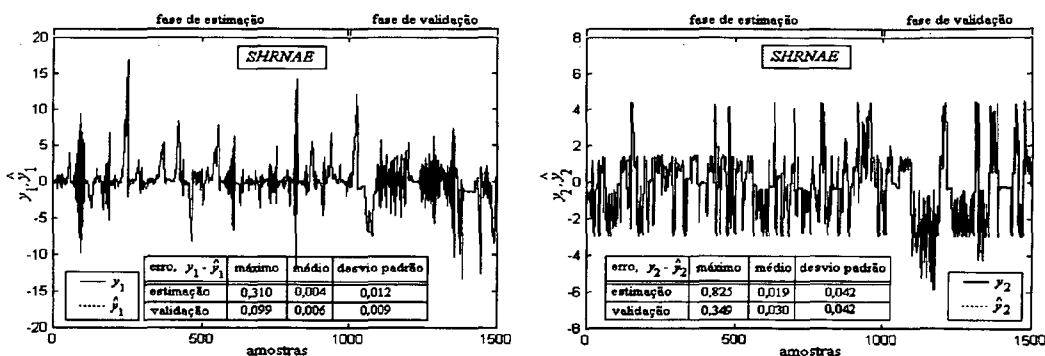


Figura 8.35. Resultados de identificação do processo *MIMO* via *SHRNAE*.

A partir da análise dos resultados obtidos pode-se concluir, de forma similar ao estudo de caso 2 da identificação do manipulador robótico que as *RN-RBF(KM)* e *RN-RBF(OLS)* tiveram desempenho adequado na tarefa de identificação. Resultados similares e também promissores são obtidos com o *SHSNRN(TSK)* e o *SHRNAE*, tanto na fase de estimação quanto validação. Entretanto, as *RNs-MLP* não apresentam resultados tão precisos, apesar que este fator pode ser aprimorado (conforme mencionado anteriormente para o estudo de caso 2, do manipulador robótico).

8.14 Conclusão do capítulo

Neste capítulo foram apresentados aspectos relativos à identificação de processos, tais como: tipos de modelos, sinais de excitação, seleção de estrutura e ordem, critérios de desempenho e procedimentos de estimação e validação.

As aplicações abordadas pelas metodologias de *IC* foram: a identificação *MISO* de um processo experimental denominado balanço horizontal, a identificação *MIMO*, em malha fechada, de um manipulador robótico com dois graus de liberdade, e a identificação de processo não-linear com duas entradas e duas saídas. Os algoritmos foram configurados adotando-se formulações de ordem reduzida para as entradas do processo. Existem várias razões para manter a ordem do modelo tão baixa quanto possível e os critérios de informação podem ser introduzidos para combinar a necessidade de adequação com o princípio da redução de dados e o princípio da parcimônia.

Quanto à identificação do *PBH*, todos os algoritmos apresentaram desempenho adequado. Um estudo comparativo entre *AEs* e *AEs* híbridos com *SA* foram apresentados, mostrando a vantagem da utilização de um método de otimização para buscas locais. A funcionalidade do *SN-MLFE* foi tratada mostrando a necessidade de um projeto apropriado que vise a satisfação de dois compromissos conflitantes, a precisão e a complexidade computacional. Em outra seção, um estudo comparativo de *RNs-MLP* mostra a importância da escolha do método de treinamento nestes tipos de *RNs* e apresenta uma forma de verificação do problema de sobre-treinamento. As *RNs-RBF* também foram analisadas em identificação do *PBH* com a obtenção de excelentes resultados com treinamento rápido.

Quanto à identificação do manipulador robótico e um processo *MIMO* não-linear, foram tratadas algumas configurações de *RNs-MLP* e *RN-RBF*, além de dois *SHIs* combinando *RNs*, *SNs* e *AEs*. A partir dos resultados obtidos nota-se a potencialidade das metodologias da *IC* utilizadas na identificação de processos multivariáveis.

Os *AEs* são ferramentas de otimização facilmente modificáveis e servem ao tratamento dos mais diversos problemas em identificação multivariável, tais como: determinação de ordem do modelo, seleção da estrutura do modelo, detecção de atraso de transporte não-unitário e sintonia de parâmetros de modelos matemáticos. Entretanto, deve ser enfatizado que os *AEs* constituem-se de metodologias mais adequadas ao tratamento de problemas de identificação *off-line*, devido à complexidade computacional necessária à obtenção de uma solução viável em identificação multivariável.

Algumas alternativas possíveis existem, visando-se minimizar estes problemas e possibilitar algumas aplicações de identificação *on-line* relevantes em aplicações de controle adaptativo e preditivo. Uma delas é a configuração hierárquica do algoritmo de identificação, onde em um nível superior se encontra um *AE* que serve para estimar a estrutura do modelo e o atraso de transporte do processo e em um nível inferior um algoritmo do tipo *MQR* para estimar os parâmetros da estrutura obtida através do *AE*.

Os *SNs* são metodologias com característica promissoras ao tratamento de processos não-lineares multivariáveis, que apresentam incertezas e/ou acoplamentos entre as variáveis. Muitas das abordagens dos *SNs*, tais como os modelos interpolativos e relacionais (não analisados nesta tese), são aproximadores universais de funções e esta é uma característica desejada de uma metodologia de identificação eficiente. Outra vantagem

dos *SNs* é a facilidade de entendimento e a simplicidade de implementação de algoritmos para identificação de processos.

Entretanto, os modelos de *SNs* interpolativos e relacionais são modelos caixa-cinza, de onde pode-se analisar claramente apenas alguns parâmetros do modelo. Outra dificuldade comum aos *SNs* e *RNs* é o tratamento de processos com dinâmica variante no tempo. Neste caso, necessita-se da configuração de alguma metodologia adaptativa baseada em algoritmos do tipo *MQR* para modificar a estrutura do modelo de forma a adequar-se à dinâmica do processo. O *MQR* é um algoritmo apropriado ao ajuste dos parâmetros de *SNs* de Takagi-Sugeno-Kang e a *RN-RBF*, pois estas técnicas requerem um problema de estimação que é linear nos parâmetros (conseqüentes do *SN* e pesos da camada intermediária para a saída da *RN-RBF*, respectivamente).

Sob outra visão, a configuração de algoritmos adaptativos em *RN-MLP* é uma tarefa complexa, pois é necessário um algoritmo de otimização não-linear para o treinamento. O motivo principal é a complexidade computacional demandada por métodos de treinamento (otimização) não-lineares, em aplicações em tempo real, é muitas vezes proibitivo.

Outra consideração importante no projeto de *SNs* e *RNs* para identificação é a necessidade de configurar-se metodologias de projeto que evitem a sobre-parametrização dos modelos inteligentes e que amenizem a falta de uma metodologia para a seleção de estrutura do modelo. A sobre-parametrização pode resultar em aprendizado lento e pobre na aproximação do processo multivariável.

As *RNs* tem sido utilizadas extensivamente de uma forma ou outra para o propósito de identificação de processos não-lineares. Muitas configurações de *RNs* são adequadas no tratamento de processos multivariáveis. Estas *RNs* podem aproximar mapeamentos não-lineares complexos com comportamento não-linear e/ou acoplamentos (sem a necessidade do cálculo da matriz *interactor*), aprender e adaptar dinâmicas de sistemas com incertezas devido as conexões dos neurônios e funções de ativação não-lineares. A *RN-RBF* é útil para a identificação de processos multivariáveis, particularmente para processos que tem pontos de operação típicos, pois estas *RNs* mapeiam aproximações locais. O aprendizado da *RN-RBF* é particularmente simples e rápido em relação à *RN-MLP*. Os principais benefícios da utilização de técnicas de aproximação local são o rápido aprendizado e a sensibilidade reduzida a ordem de apresentação dos dados de treinamento.

A estrutura da *RN-RBF* oferece uma estrutura mais propícia à análise de estabilidade e possibilita a adoção de configurações adaptativas para propósitos de controle adaptativo e preditivo. Um aspecto negativo da *RN-RBF* é a de que podem sofrer da “maldição” da dimensionalidade, pois quando a dimensão do vetor de entrada é aumentada a dimensionalidade dos centros das funções de base tem, geralmente, aumento exponencial para representar o processo multivariável adequadamente. A adição de centros pode também ser requerida para tratar com o aumento da complexidade causada por dimensões de entrada adicionais. Muitas das características da *RN-MLP* são comuns a *RN-GMDH* e *RN-PREM*. Um aspecto a ser enfatizado é que estas estruturas são aproximadores globais. A desvantagem das *RN-MLP* é a de constituírem modelos caixa-preta e apresentarem treinamento lento para aplicações de identificação multivariável.

Uma dificuldade na configuração de *RNs* para aplicações de identificação são aspectos relativos ao domínio temporal. Se a ordem de um sistema é conhecida, a utilização do método *tapped-delay-line* permite a um sistema dinâmico ser modelado com uma *RN* estática. Entretanto, quando a ordem é desconhecida, muitas aplicações podem levar o projetista a adotar um número elevado de entradas. Tal procedimento pode levar a modelos ineficientes devido ao grande número de entradas necessário. Outra dificuldade na configuração de uma *RN* é quando o processo apresenta um atraso de transporte não-unitário (valor grande). Algumas alternativas poderiam ser a combinação de algum algoritmo de estimação de atraso de transporte, do tipo preditor de Smith (MATAUSEK & MICIC, 1999), a uma *RN*.

Capítulo 9

APLICAÇÕES EM CONTROLE DE PROCESSOS MULTIVARIÁVEIS

9.1 Introdução

Neste capítulo diversas metodologias inteligentes, preditivas e de estrutura variável são configuradas e aplicadas em controle de processos multivariáveis não-lineares. Os estudos em controle multivariável são realizados no controle de um manipulador robótico, este utilizado em identificação não-linear no capítulo 8. As metodologias utilizadas são:

- (i) controle *PD* e *PID* em configurações *multiloop* e acoplada, com sintonia através de *AEs* e *AEs* combinados a *SA*;
- (ii) configurações de controle *PID* nebuloso;
- (iii) um esquema de controle neural;
- (iv) *SHIs* com abordagens evolutivas, nebulosas e neurais combinadas;
- (v) *GPC* adaptativo e combinado a *IC*;
- (vi) projeto de *QSMC* combinado a controle *GMV*;

9.2 Controle de um manipulador robótico com dois graus de liberdade

As equações que regem o manipulador robótico foram apresentadas no capítulo 8. As simulações da dinâmica do robô são resolvidas pelo método de Runge-Kutta de 4ª ordem e os controladores implementados visam satisfazer um polinômio de interpolação regido por *Splines* de quinta ordem (CRAIG, 1986).

As posições e velocidades iniciais ($k=0$) das juntas são: $\theta_1(k) = \theta_2(k) = 0$ rad, e $\dot{\theta}_1(k) = \dot{\theta}_2(k) = 0$ rad/s. A análise do comportamento servo visa verificar a capacidade do seguimento das trajetórias de posição e velocidade pelos controladores. O objetivo, neste caso, é a obtenção dos valores de posição $\theta_1(k) = 1$ rad e $\theta_2(k) = 2$ rad ($k=2$ s), e $\theta_1(k) = 0.5$ rad e $\theta_2(k) = 4$ rad ($k=4$ s), visando seguir as trajetórias estipuladas.

Os valores para a velocidade iniciais (0 e 2,01 s) e finais (2 e 4 s) são, para ambos as juntas, é de $\dot{\theta}_1(k) = \dot{\theta}_2(k) = 0$ rad/s. Os valores de massa e comprimentos de elos são os

mesmos do capítulo 8, isto é, comprimentos — $l_1=0,8$ m e $l_2=0,4$ m, e massas — $m_1=m_2=0,1$ kg. O período de amostragem é $T_s=10$ ms.

A análise do comportamento regulatório visa verificar a robustez dos projetos e controle para modificações paramétricas, ou seja, mudanças nos valores das massas e comprimentos dos elos do manipulador robótico. Neste caso, adota-se:

- (i) de 0.8 a 1.2 s: $l_1=0,8$ m, $l_2=0,4$ m, $m_1=0,05$ e $m_2=0,3$ kg;
- (ii) de 2.3 a 2.4 s: $l_1=1,3$ m, $l_2=0,6$ m e $m_1=m_2=0,1$ kg;
- (iii) de 2.7 a 3.2 s: $l_1=0,6$ m, $l_2=0,3$ m, $m_1=0,2$ e $m_2=0,3$ kg.

Os projetos de controle inteligentes tratados apresentam uma fase de sintonia (comportamento servo) e uma fase de testes (comportamento regulatório), visando a análise de robustez dos controladores frente a incertezas de mudanças de massas e comprimentos de elos diferentes dos utilizados na fase de sintonia dos controladores.

As abordagens adaptativas do *GPC* e controle de estrutura variável têm fase de sintonia, através de procedimentos heurísticos de tentativa e erro dos parâmetros de projeto. A seguir são apresentados os projetos de controle e os respectivos resultados.

9.3 Projeto de controle *PD* e *PID* com sintonia via *AEs*

Os projetos de controle evolutivo visam a sintonia de controle *PD* e *PID* em abordagens *multiloop* e acoplada. Os projetos são estruturados, na forma discreta no tempo, em controle *PD* e *PID* multivariáveis. Novas propostas são apresentadas à lei de controle pela inclusão de termos acoplados, para melhorar a robustez e o desempenho dos controladores, propostos em COELHO & COELHO (1998c). A função de adequação à otimização dos *AEs* é regida pela equação;

$$F = \frac{\xi}{1 + J} \quad (9.1)$$

onde ξ é um fator de escala e J é a função custo utilizada nos experimentos para a otimização (comportamento servo) dos controladores evolutivos configurada em:

$$J = \sum_{j=1}^2 \sum_{k=1}^N [(q_j(k))^2 + 10(\dot{q}_j(k))^2] \quad (9.2)$$

onde N é o número de amostras. O projeto dos controladores PD e PID envolve a seleção de diversos parâmetros, o que usualmente é efetuado por métodos heurísticos de tentativa e erro, motivando a utilização de AEs para sintonia adequada dos parâmetros da lei de controle. A seguir, são apresentados os projetos das leis de controle com parâmetros sintonizados através de AEs .

9.3.1 Projeto PD multivariável descentralizado

A forma de controle analógico do PD multivariável é dada por:

$$\tau_j(t) = K_{p,j}q_j(t) + K_{v,j}\dot{q}_j(t), \quad j=1,2 \quad (9.3)$$

onde $K_{p,j}$ é o ganho proporcional da lei de controle. A discretização da equação (9.3), resulta na equação discreta do controlador dada por

$$\begin{aligned} \tau_j(k) &= \left[K_{p,j} + \frac{K_{v,j}}{T_s} \right] \theta_j(k) - \frac{K_{v,j}}{T_s} \theta_j(k-1) \\ &- \left[K_{p,j} + \frac{K_{v,j}}{T_s} \right] \theta_{d,j}(k) + \frac{K_{v,j}}{T_s} \theta_{d,j}(k-1), \quad j=1,2 \end{aligned} \quad (9.4)$$

onde $K_{p,j}$ e $K_{v,j}$ são matrizes diagonais de ganhos relativos aos erros de posição e velocidade, respectivamente.

9.3.2 Projeto PID multivariável descentralizado

A forma de controle analógico do PID multivariável é dada por:

$$\tau_j(t) = K_{p,j}q_j(t) + K_{v,j}\dot{q}_j(t) + K_{i,j} \int_0^t q_j dt, \quad j=1,2 \quad (9.5)$$

onde $K_{i,j}$ é uma matriz diagonal positiva, relativa aos termos integrais da lei de controle PID . A discretização da equação (9.5) providencia uma equação discreta do controlador regida por

$$\Delta \tau_j(k) = (K_{p,j} + 0.5T_s K_{i,j}) q_j(k) - K_{p,j} - 0.5T_s K_{i,j} q_j(k-1) + K_{v,j} \dot{q}_j(k) + K_{v,j} \dot{q}_j(k-1), \quad (9.6)$$

$j=1,2$

$$\dot{q}_j = \frac{(\theta_{d,j}(k) - \theta_{d,j}(k-1))}{T_s} - \frac{(\theta_j(k) - \theta_j(k-1))}{T_s}, \quad j = 1,2 \quad (9.7)$$

9.3.3 Projetos PD e PID multivariáveis centralizados

O engenheiro de controle pode optar em projetar cada malha de controle de forma a tratar processos multivariáveis que apresentam acoplamentos. O ajuste dos parâmetros de controle de uma malha afeta o desempenho de outra, e, algumas vezes, produz um desempenho inadequado do projeto de controle.

Os controladores para robôs industriais são projetados para cada junta baseados em conceitos *SISO*, desconsiderando-se os efeitos dos acoplamentos de cada junta. Diversas estratégias de controle têm sido investigadas para tratar com não-linearidades acentuadas e acoplamentos na dinâmica do robô (CRAIG, 1986).

Geralmente, a dinâmica de robôs manipuladores é descrita por equações diferenciais não-lineares de segunda ordem. Os termos de acoplamento representam a interação entre as diversas variáveis comuns as malhas e dependem de parâmetros físicos, tais como carga, comprimento e peso dos elos.

A seguir, apresenta-se uma forma específica de controle *PID* e pode também ser adicionada ao controle *PD* de forma similar. Esta estrutura de controle consiste da adição de termos acoplados relativos a erros de posição e de velocidade, na lei de controle, obtendo-se várias formulações de controle diferentes para o *PID*. Assim, na forma de controle *PID* especificado pelas equações (9.4) e (9.6) são adicionados termos de acoplamento dados por:

(i) controle *PID* com termos acoplados de posição (*PID-AP*):

$$\Delta \tau_1(k) = (K_{p,1} + 0.5T_s K_{i,1}) q_1(k) - K_{p,1} - 0.5T_s K_{i,1} q_1(k-1) + K_{v,1} \dot{q}_1(k) + K_{v,1} \dot{q}_1(k-1) + K_1 q_2(k) \quad (9.8)$$

$$\Delta \tau_2(k) = (K_{p,2} + 0.5T_s K_{i,2}) q_2(k) - K_{p,2} - 0.5T_s K_{i,2} q_2(k-1) + K_{v,2} \dot{q}_2(k) + \quad (9.9)$$

$$K_{v,2} \dot{q}_2(k-1) + K_2 q_1(k)$$

ii) controle *PID* com termos acoplados de velocidade (*PID-AV*):

$$\Delta \tau_1(k) = (K_{p,1} + 0.5T_s K_{i,1}) q_1(k) - K_{p,1} - 0.5T_s K_{i,1} q_1(k-1) + K_{v,1} \dot{q}_1(k) + K_{v,1} \dot{q}_1(k-1) + K_3 \dot{q}_2(k) \quad (9.10)$$

$$\Delta \tau_2(k) = (K_{p,2} + 0.5T_s K_{i,2}) q_2(k) - K_{p,2} - 0.5T_s K_{i,2} q_2(k-1) + K_{v,2} \dot{q}_2(k) + K_{v,2} \dot{q}_2(k-1) + K_4 \dot{q}_1(k) \quad (9.11)$$

iii) controle *PID* com termos acoplados de posição e velocidades (*PID-APV*):

$$\Delta \tau_1(k) = (K_{p,1} + 0.5T_s K_{i,1}) q_1(k) - K_{p,1} - 0.5T_s K_{i,1} q_1(k-1) + K_{v,1} \dot{q}_1(k) + K_{v,1} \dot{q}_1(k-1) + K_1 q_2(k) + K_3 \dot{q}_2(k) \quad (9.12)$$

$$\Delta \tau_2(k) = (K_{p,2} + 0.5T_s K_{i,2}) q_2(k) - K_{p,2} - 0.5T_s K_{i,2} q_2(k-1) + K_{v,2} \dot{q}_2(k) + K_{v,2} \dot{q}_2(k-1) + K_2 q_1(k) + K_4 \dot{q}_1(k) \quad (9.13)$$

onde K_1 - K_2 , e K_3 - K_4 são ganhos positivos relativos aos erros de posição e velocidade, respectivamente.

9.3.4 Resultados de controle

A sintonia de todos parâmetros K é realizada via *AEs*. No caso, do manipulador robótico têm-se 4, 6, 8, 8 e 10 parâmetros a serem sintonizados para o *PD*, *PID*, *PID-AP*, *PID-A* e *PID-APV*, respectivamente. O intervalo de busca para $K_p \in [0;600]$, K_i , $K_d \in [0;100]$.

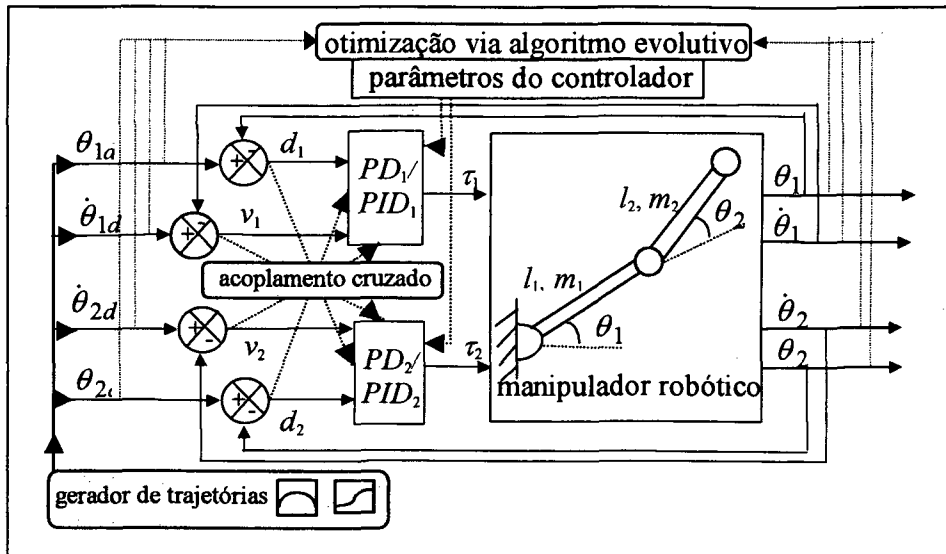


Figura 9.1. Esquemas de controle PD e PID com sintonia via AEs.

A análise de convergência dos AEs (melhor indivíduo), após 200 gerações, é apresentada na figura 9.2. O número de experimentos realizados são 50, para cada AE. Nos experimentos são utilizadas sementes iguais para o gerador de números aleatórios, para a configuração da população inicial dos AEs, em cada experimento. O critério de parada do ciclo evolutivo é de 200 gerações para uma população de 30 indivíduos. A configuração dos AEs utilizados são as seguintes:

- (i) AG(1): representação de ponto flutuante, método de seleção *breeder*, operador de recombinação aritmética ($p_c=0,8$), operador de mutação não-uniforme ($p_m=0,1$);
- (ii) AG(2): abordagem de ajuste automático das probabilidades de cruzamento e mutação (SRINIVAS & PATNAIK, 1994a);
- (iii) AG(3): abordagem com probabilidade de mutação variável (KIM *et al.*, 1997a);
- (iv) AG(4): abordagem com operador de reprodução baseado em gradiente (PHAM & JIN, 1995);
- (v) AG(5): combinando abordagem com probabilidade de mutação variável e operador de reprodução baseado em gradiente;
- (vi) AG(6): combinando abordagem com probabilidade de mutação variável e ajuste automático das probabilidades de cruzamento e mutação;
- (vii) AG(7): combina as abordagens do AG(3), AG(4) e AG(5);
- (viii) AG(8): representação canônica, método de seleção *breeder*, operador de cruzamento com um ponto de cruzamento ($p_c=0,8$), operador de mutação não-uniforme ($p_m=0,05$);
- (ix) AGSA(1): idem ao AG(4), mas utiliza SA à busca local;
- (x) AGSA(2): idem ao AG(7), contudo com SA à busca local;
- (xi) EE(1): (5+25)-EE;
- (xii) EE(2): (1+29)-EEMC;
- (xiii) EESA(1): EE(5+25) com SA à busca local;
- (xiv) EESA(2): (1+29)-EEMC com SA à busca local;

(xv) $PE(1)$;

(xvi) $PESA(1)$: PE com SA à busca local.

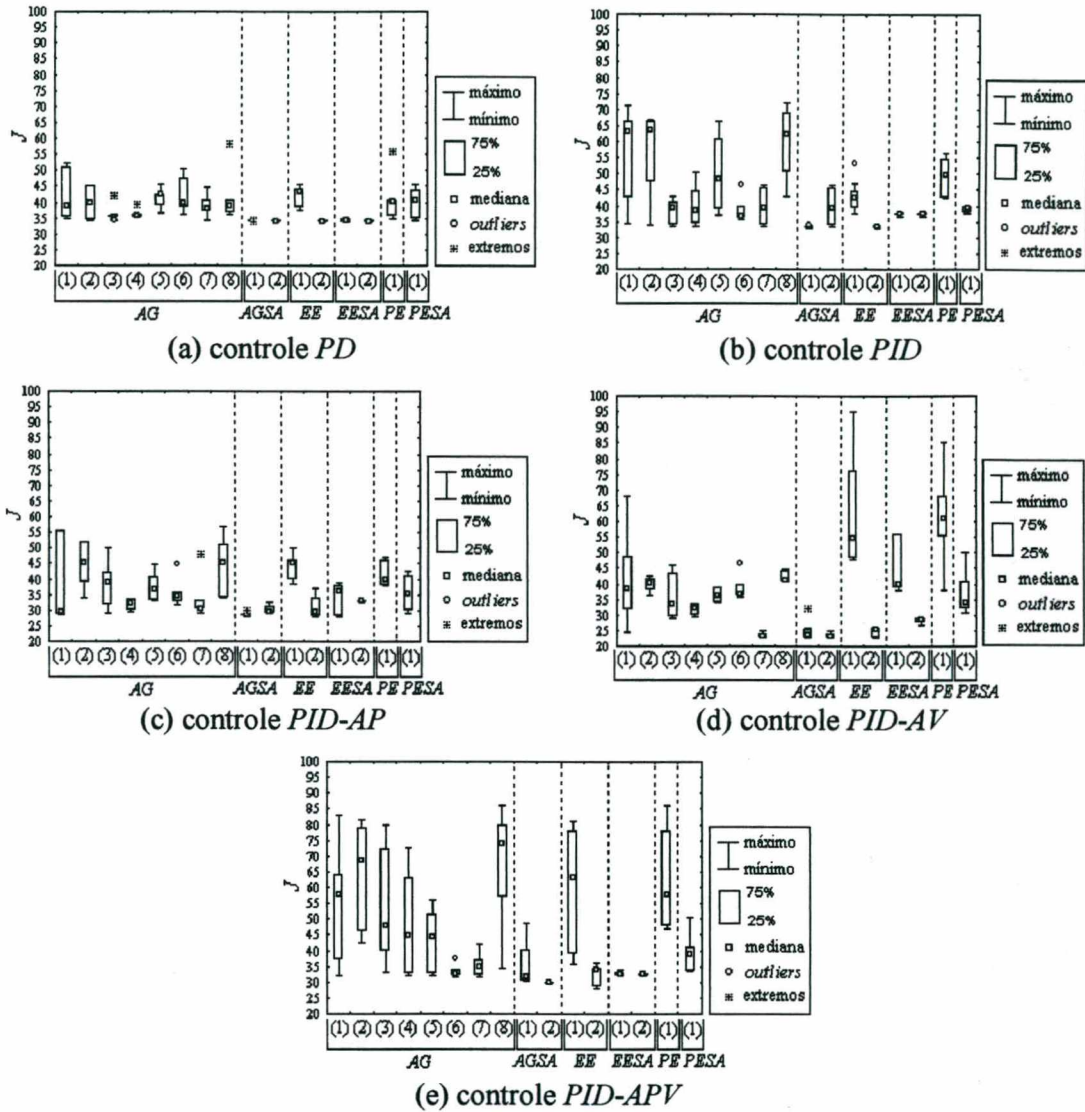


Figura 9.2. Análise comparativa da sintonia via AEs dos ganhos de controle PD e PID .

Nota-se pela análise da figura 9.2, que os AGs tem melhores resultados pela utilização de ajuste automático das probabilidades de cruzamento e mutação, abordagem com probabilidade de mutação variável e pela utilização do operador de reprodução baseado em gradiente.

Os $AGSA$ e $EESA$ apresentam resultados promissores, enfatizando-se o papel do algoritmo SA em busca locais, visando aprimorar os resultados obtidos com os AEs . Nota-

se, também, nestes casos, que existe uma pequena diferença entre os valores máximos e mínimos da função objetivo. O paradigma (1+29)-EEMC apresenta excelentes resultados, contudo, não teve aprimoramento do desempenho pela utilização do SA.

Os AGs (com representação de ponto flutuante e canônico) em configuração padrão, a EE e a PE, atuando isoladamente, apresentam resultados inferiores, quanto à capacidade de convergência que os métodos híbridos com SA.

Os AG(1), AG(8), EE(1) e PE(1) apresentam um comportamento de variância acentuada em relação à média, com valores de máximo e mínimo muito distantes. Outra observação é que os valores dos percentis e medianas são também acentuados. A tabela 9.1 apresenta os ganhos dos melhores resultados em projeto de controle PD e PID com sintonia via AEs.

Tabela 9.1. Parâmetros de sintonia via AEs para os projetos de controle PD e PID.

projeto	AE	$K_{p,1}$	$K_{p,2}$	$K_{i,1}$	$K_{i,2}$	$K_{d,1}$	$K_{d,2}$	K_1	K_2	K_3	K_4
PD	AG(8)	87,47	2,78			0,28	24,16				
PID	(1+29)-EEMC	76,24	99,04	3,54	24,33	89,22	0,23				
PID-AP	(1+29)-EEMC	57,38	18,31	5,48	27,58	32,66	0,08	4,95	0,02		
PD-AV	AGSA(1)	72,48	99,61	2,09	29,85	20,47	0,48			0,76	4,91
PD-APV	(1+29)-EEMC	56,34	53,34	5,46	27,29	34,36	0,08	4,83	0,03	0,46	2,82

Nas figuras 9.3 e 9.4 são apresentadas uma análise comparativa da função objetivo e da variância do controle dos melhores resultados na sintonia via AEs dos ganhos de controle PD e PID, respectivamente.

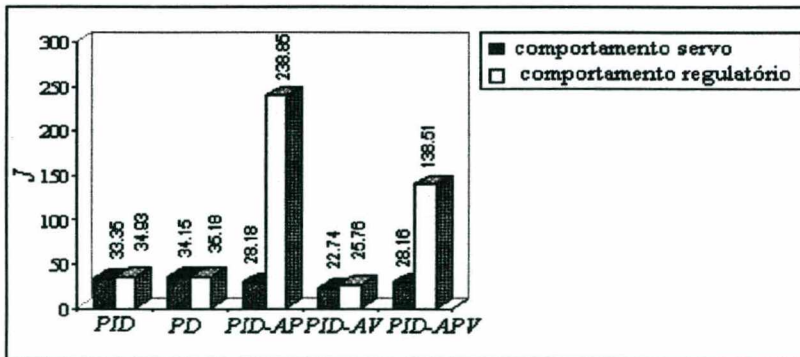


Figura 9.3. Análise comparativa da função objetivo, J, dos melhores resultados dos AEs na sintonia dos ganhos de controle PD e PID.

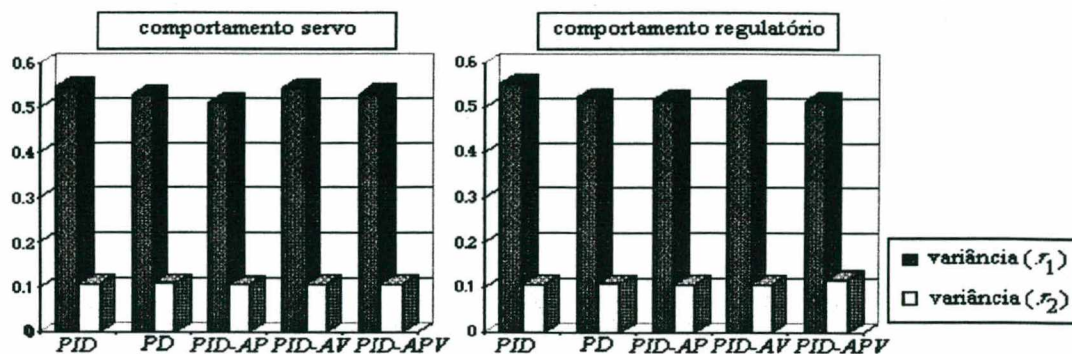


Figura 9.4. Análise comparativa da variância do controle dos melhores resultados na sintonia via *AEs* dos ganhos de controle *PD* e *PID*.

A análise apresentada na figura 9.3 indica que o desempenho quanto a função objetivo é superior para os esquemas de controle *PD*, *PID* e *PID-AV*, com sintonia evolutiva, em relação aos projetos *PID-AP* e *PID-APV*. O melhor resultado obtido é através da $(1+29)$ -*EEMC*, em projeto de controle *PID-AV*. Quanto à variância dos torques em comportamento servo e regulatório (figura 9.4), os projetos de controle apresentam valores com diferenças pouco significativas. As figuras 9.5 a 9.14 ilustram os melhores resultados de controle *PD* e *PID* via *AEs*, tanto para o comportamento servo (fase de sintonia dos controladores), quanto para o regulatório (fase de testes da robustez dos controladores).

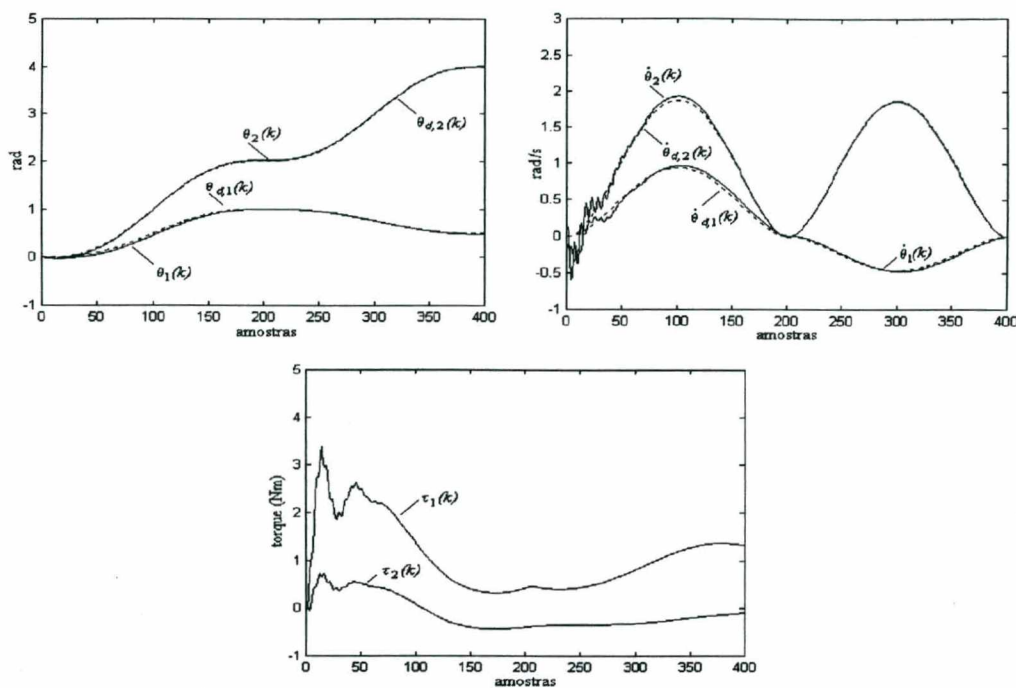


Figura 9.5. Controle PD do manipulador robótico (comportamento servo) via AG(8).

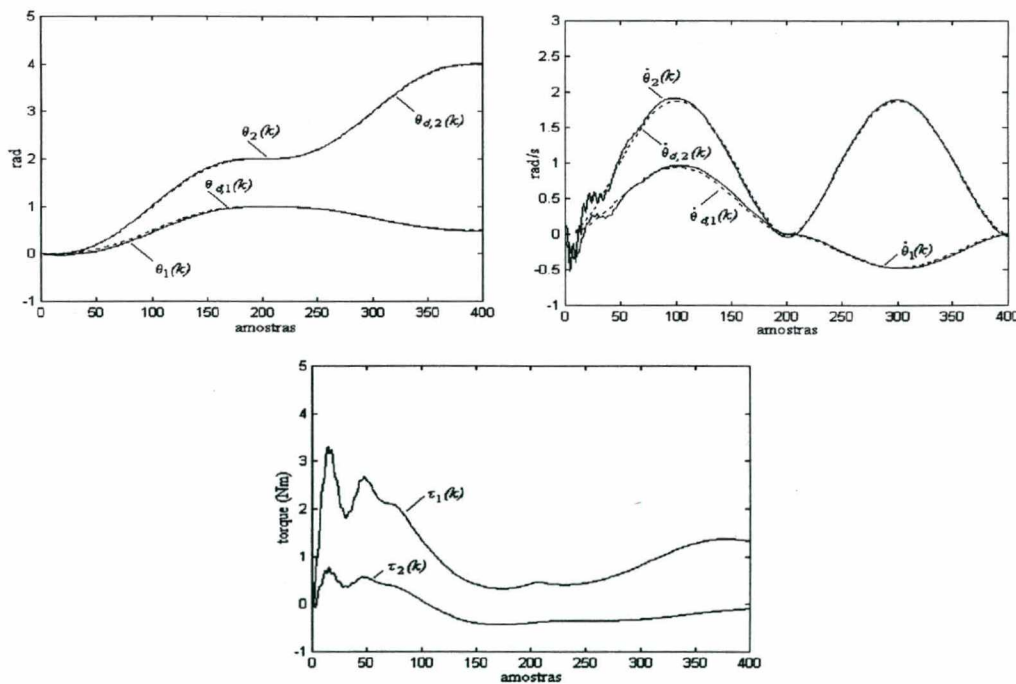


Figura 9.6. Controle PID do manipulador robótico (comportamento servo) via (1+29)-
EEMC.

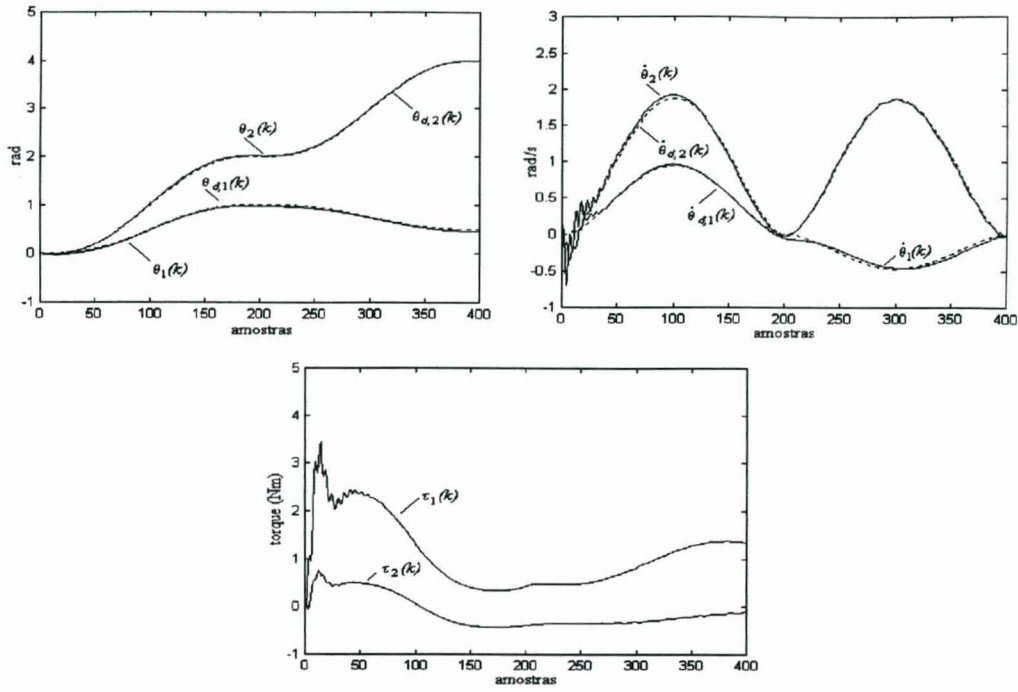


Figura 9.7. Controle *PID-AP* do manipulador robótico (comportamento servo) via $(1+29)$ -*EEMC*.

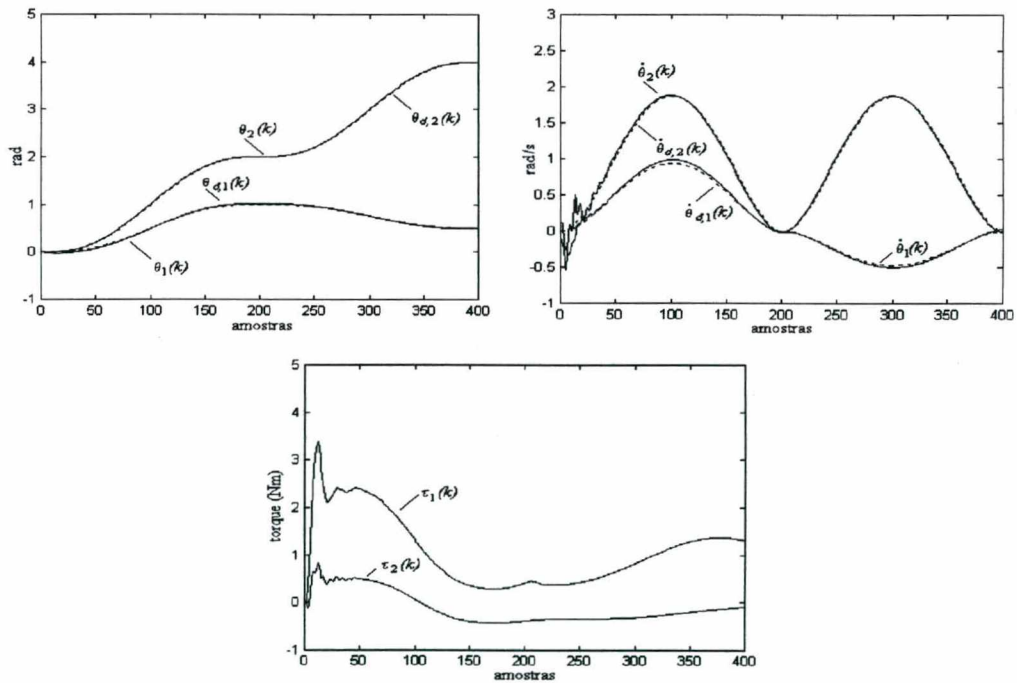


Figura 9.8. Controle *PID-AV* do manipulador robótico (comportamento servo) via *AGSA(1)*.

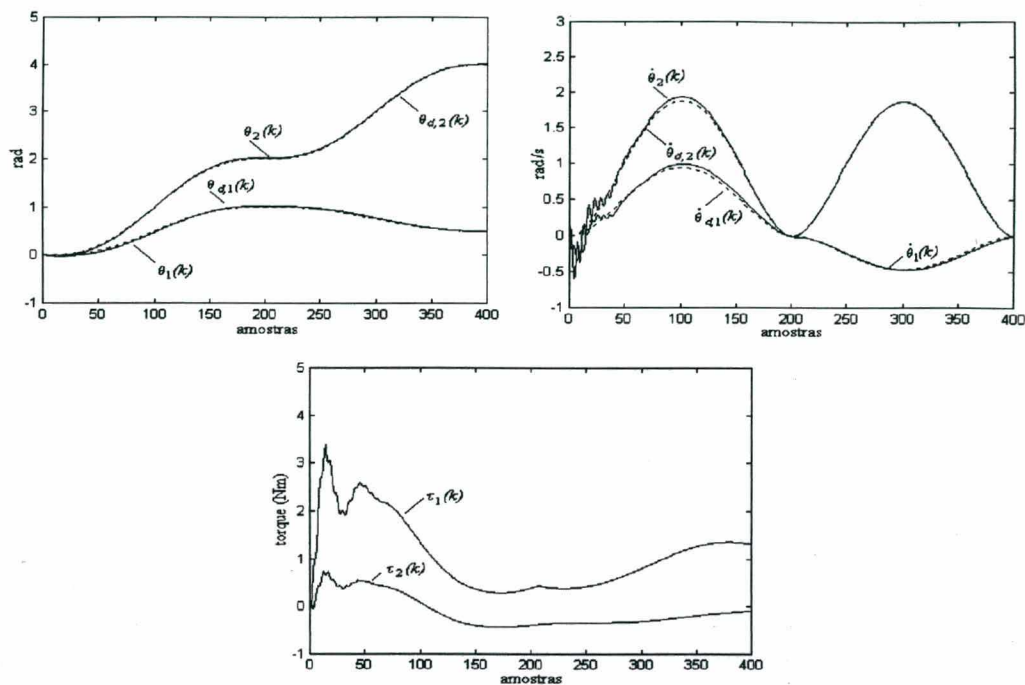


Figura 9.9. Controle *PID-APV* do manipulador robótico (comportamento servo) via $(1+29)$ -*EEMC*.

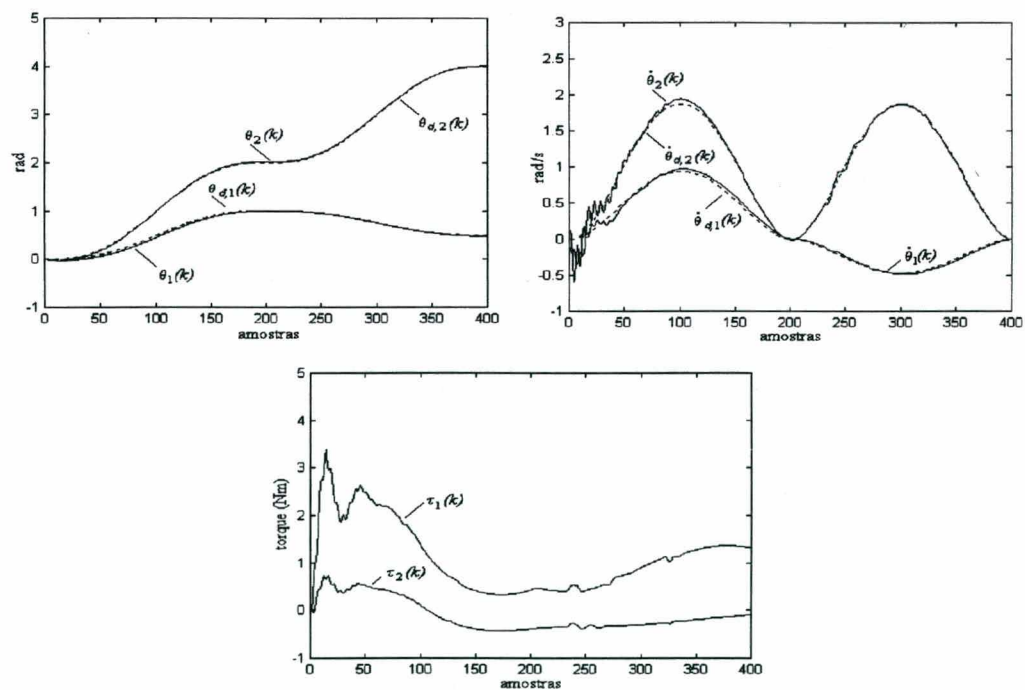


Figura 9.10. Controle *PD* do manipulador robótico (comportamento regulatório) via *AG(8)*.

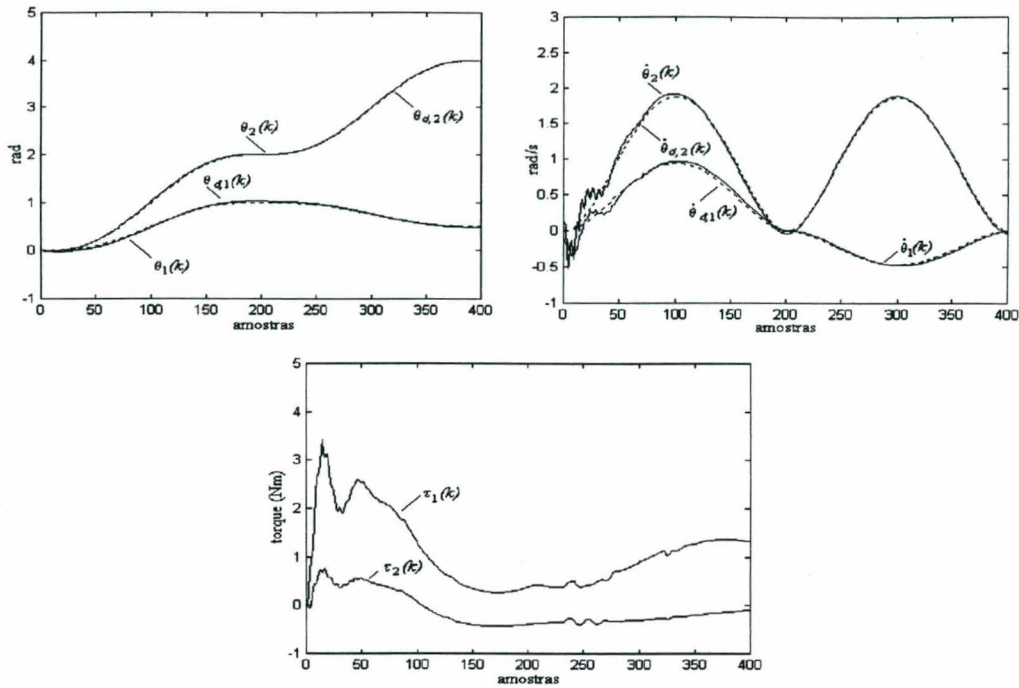


Figura 9.11. Controle *PID* do manipulador robótico (comportamento regulatório) via $(1+29)$ -*EEMC*.

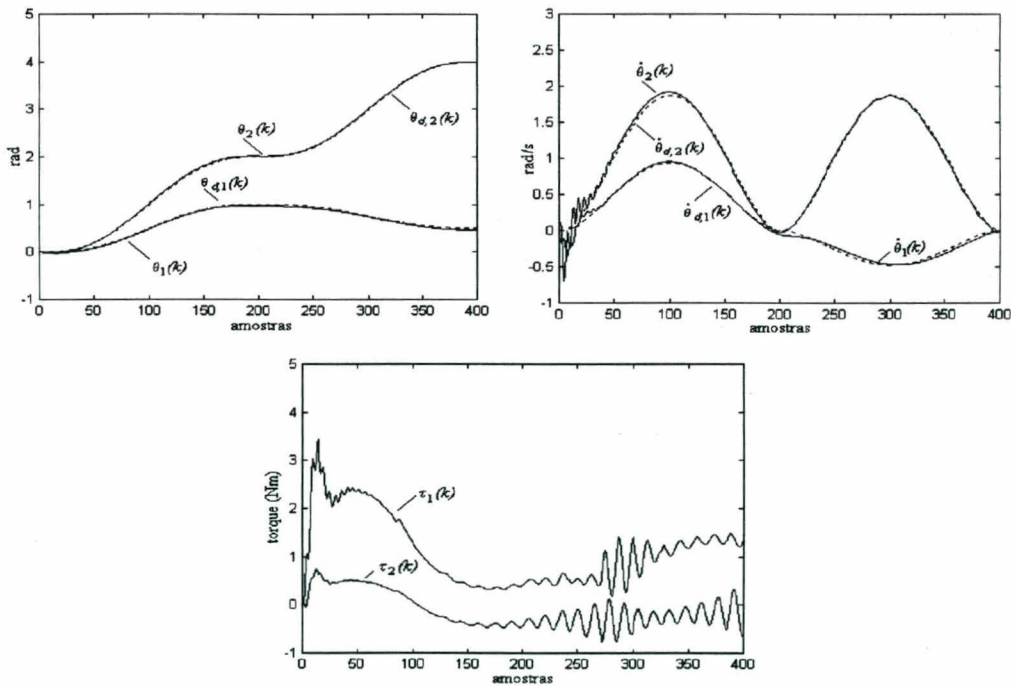


Figura 9.12. Controle *PID-AP* do manipulador robótico (comportamento regulatório) via $(1+29)$ -*EEMC*.

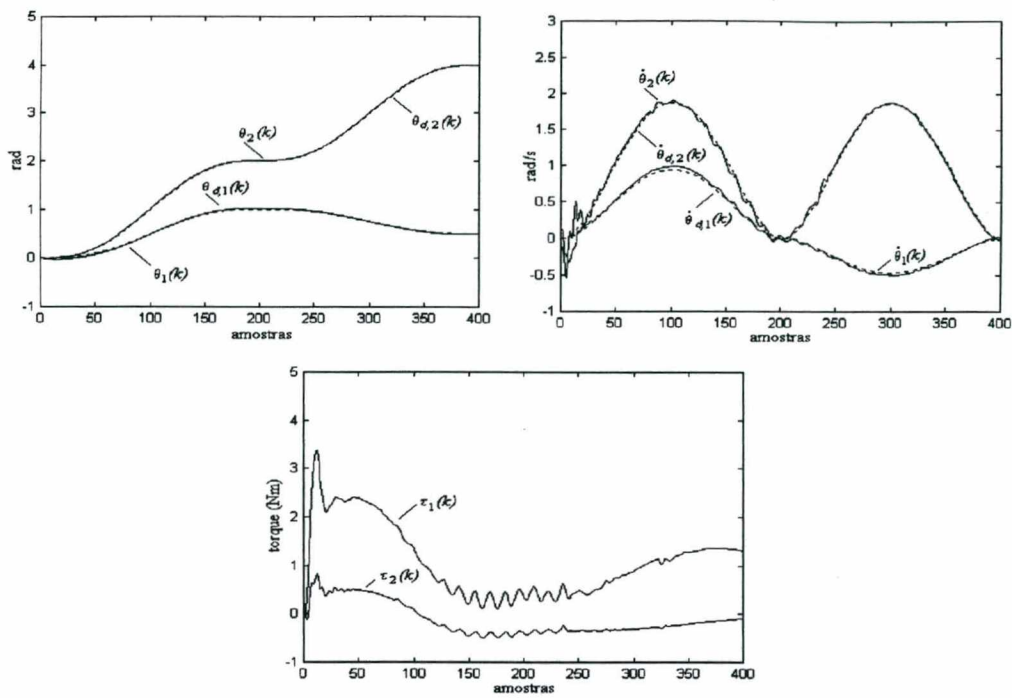


Figura 9.13. Controle *PID-AV* do manipulador robótico (comportamento regulatório) via *AGSA(1)*.

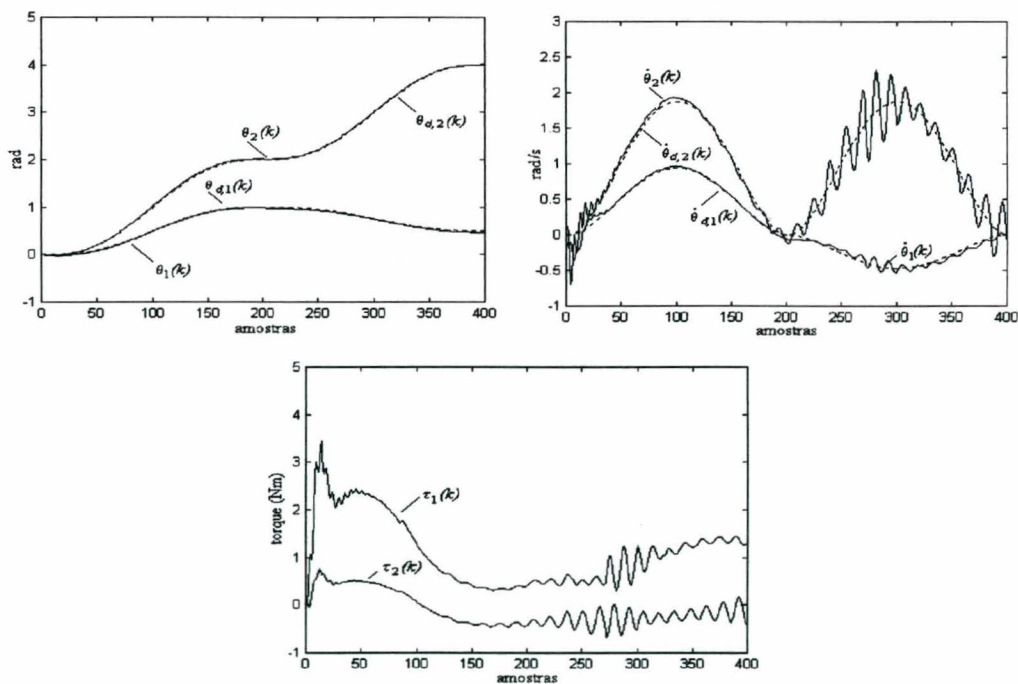


Figura 9.14. Controle *PID-APV* do manipulador robótico (comportamento regulatório) via *(1+29)-EEMC*.

Os controladores visando o comportamento servo (figuras 9.5 a 9.9), isto é, a fase de sintonia dos controladores via *AEs* apresentam um desempenho adequado no controle do manipulador robótico.

Quanto ao comportamento regulatório (figuras 9.10 a 9.14), é analisado o desempenho dos controladores frente a mudanças de massa e do comprimento de elos. Estas mudanças não são previstas na fase de sintonia. Os controladores *PD* e *PID* apresentam desempenho adequado para o seguimento das trajetórias de posição e velocidade. Nota-se, também, que a ação de controle apresenta comportamento com pequenas oscilações.

Os resultados do controlador *PID-AV* apresentam seguimento das trajetórias adequado, entretanto, apresentam oscilação significativa quanto aos torques aplicados ao robô. Os controladores *PID-AP* e *PID-APV* são pouco robustos a situações não previstas na fase de sintonia, conforme observado pelas figuras 9.13 e 9.14, onde os resultados apresentam comportamento oscilatório.

Outros estudos de diferentes configurações *AGs*, *EES* e *EEMC*, em sintonia de controle *PD* e *PID*, aplicados a um manipulador robótico são apresentados em COELHO & COELHO (1998c, 2000b).

9.3.4 Projetos de controladores nebulosos, neurais e híbridos inteligentes

Alguns projetos de controladores nebulosos, neurais e híbridos inteligentes são implementados. As configurações de projeto baseadas em *SNs* (figura 9.15) são as seguintes: *CLN-(PD+I)*, *CLN-PI* nebuloso + (*D* convencional), *CLN-(PD+I)* incremental e *CLN-(PD+PI)*.

Os controladores adotados têm configuração *multiloop*, onde cada controlador apresenta 4 entradas (erros e variação dos erros de posição e de velocidade das juntas) e 2 saídas (torques). O universo de discurso de cada variável de entrada e de saída é dividido em 5 termos lingüísticos (grande negativo, pequeno negativo, zero, pequeno positivo, grande positivo).

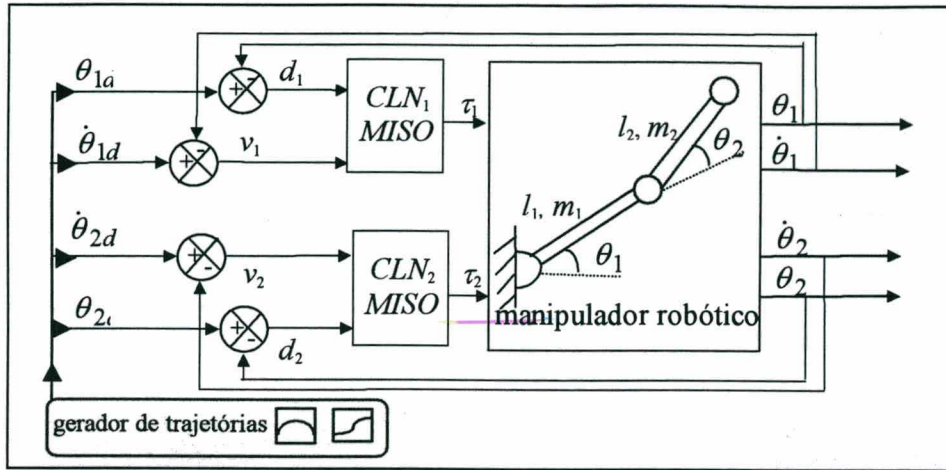


Figura 9.15. Esquema de controle nebuloso *multiloop*.

O projeto de $CLN(PD+I)$ também é utilizado em uma configuração de *SHI*, onde uma $(1+29)$ -*EEMC* é utilizada para o ajuste dos fatores de escala das funções de pertinência das variáveis de entrada e saída do *CLN*. O ciclo evolutivo para este *SHI* utilizado é de 200 gerações.

O projeto de controle neural tem a seguinte estrutura. Um controlador *RN-RBF* é configurado via pseudo-inversa, baseado em modelo neural (emulador *RN-RBF(KM)*), apresentado no capítulo 8. O controlador neural configurado é composto das entradas de posições e velocidades desejadas e obtidas, no instante $k-1$ (4 entradas com valores das posições e 4 entradas com valores de velocidades), 8 funções Gaussianas na camada oculta da *RN-RBF* controladora e 2 saídas (torques).

Outros projetos de controle via *IC* utilizados, nesta aplicação, são os *SHIs*. Um controlador neural *RN-RBF*, com estrutura *PID* e otimização dos centros e variâncias das funções Gaussianas (total de 8 funções Gaussianas), através de *AGSA* com operador de reprodução, baseado em gradiente (PHAM & JIN, 1995), é utilizado, conforme estrutura apresentada na figura 9.16. O ciclo evolutivo é de 200 gerações para uma população de 30 indivíduos.

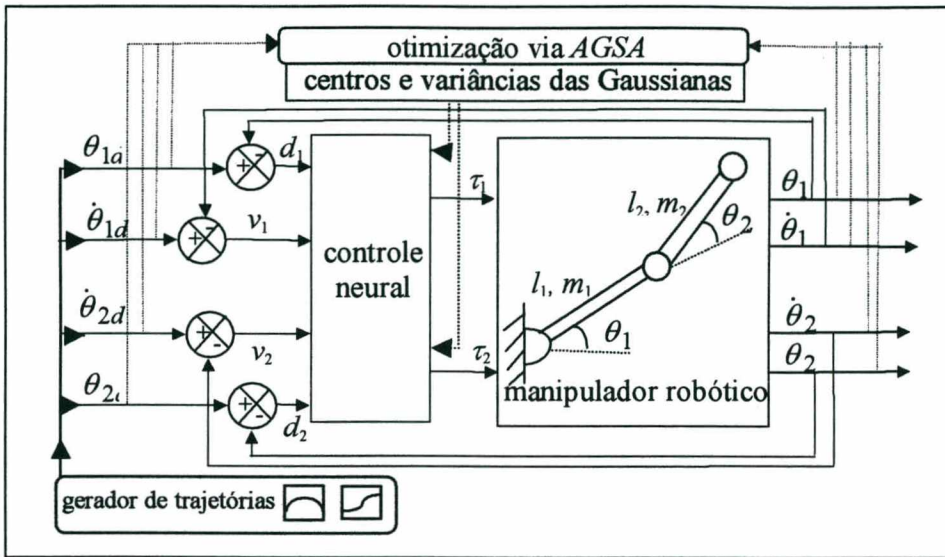


Figura 9.16. Esquema de controle neural *RN-RBF* com otimização através de *AGSA*.

Um projeto de *SHI* neuro-nebuloso-evolutivo também é configurado. O sistema neuro-nebuloso constitui-se do modelo de *TSK* (emulador), obtido em identificação do manipulador robótico (capítulo 8). A otimização do controlador, este composto por uma *RN-RBF*, é realizada através de uma $(1+29)$ -*EEMC* em 200 gerações. Os resultados de comportamento servo e regulatório dos controladores são ilustrados na figura 9.17.

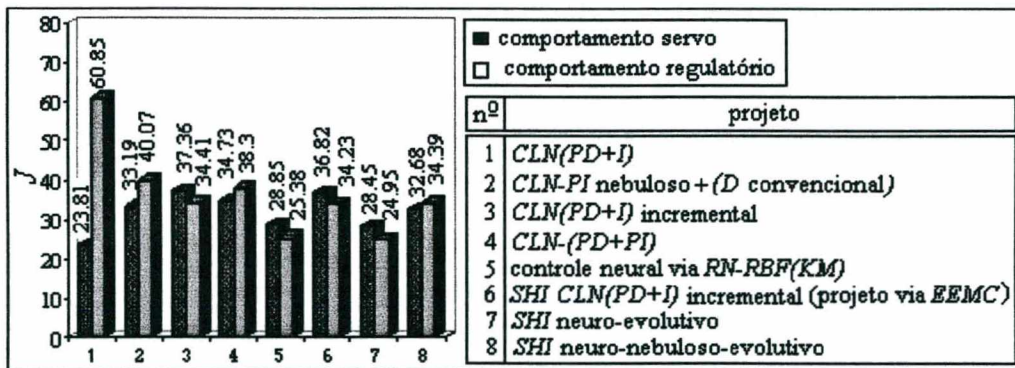


Figura 9.17. Resultados de controle nebuloso, neural e híbrido inteligente.

Os *CLNs* apresentam resultados adequados para o comportamento servo. Quanto ao comportamento regulatório, o *CLN*-(*PD+I*) não apresentou resultado apropriado, pois o valor da função objetivo excede bastante os valores dos outros três *CLNs*.

O melhor resultado dos controladores nebulosos, quanto ao comportamento servo ($J=33,19$), é obtido pelo *CLN-PI* nebuloso + (*D* convencional). Entretanto, o melhor

comportamento regulatório ($J=34,41$) dos projetos de controle nebuloso foi obtido é através do *CLN-(PD+I)* incremental.

Uma observação relevante é que estas considerações são baseadas no funcional J , equação (9.2), que considera somente os erros das trajetórias de posição e velocidade, não sendo utilizada a variância do sinal de torque aplicado ao manipulador robótico.

As figuras 9.18 a 9.27 ilustram os melhores resultados de controle *PD* e *PID* através de esquemas de controle baseados em paradigmas da *IC*.

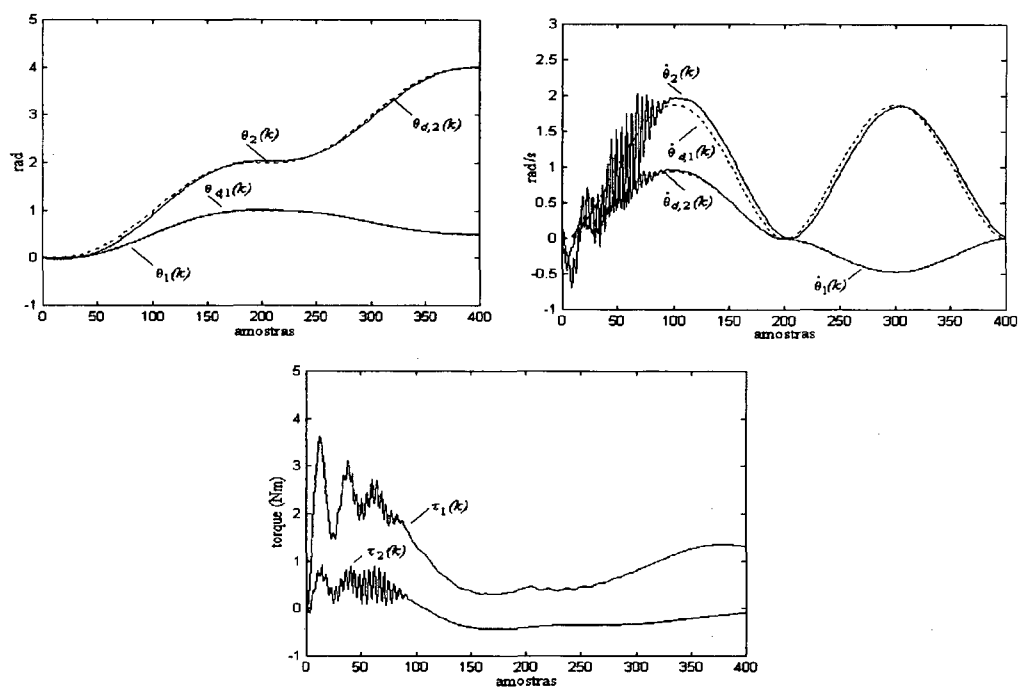


Figura 9.18. *CLN-(PD+I)* (comportamento servo).

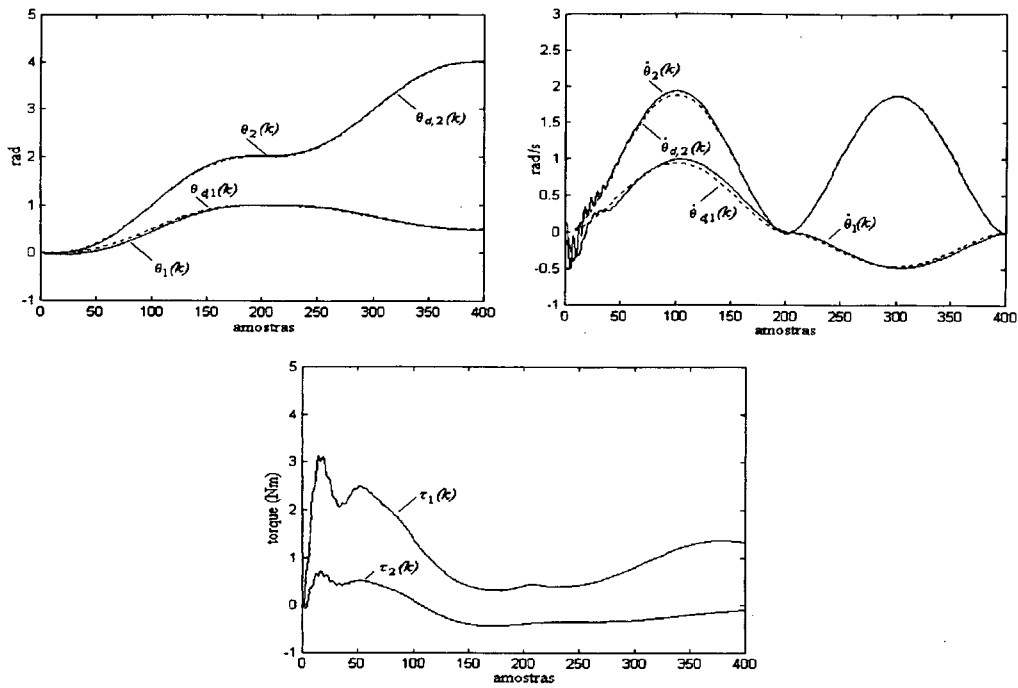


Figura 9.19. *CLN-PI* nebuloso + (D convencional) (comportamento servo).

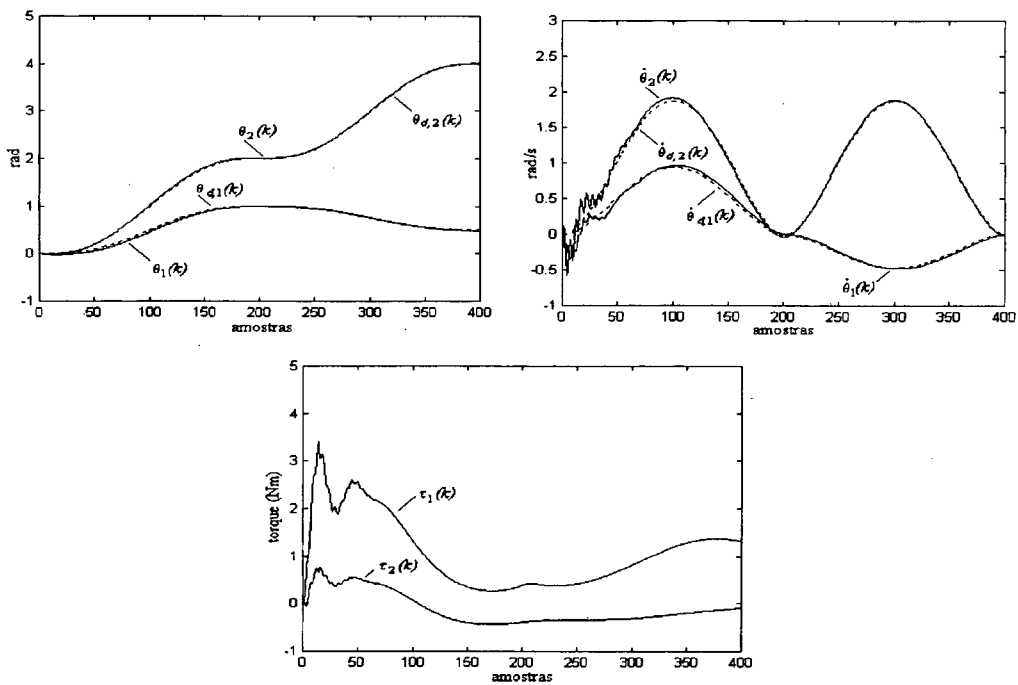


Figura 9.20. *CLN-(PD+I)* incremental (comportamento servo).

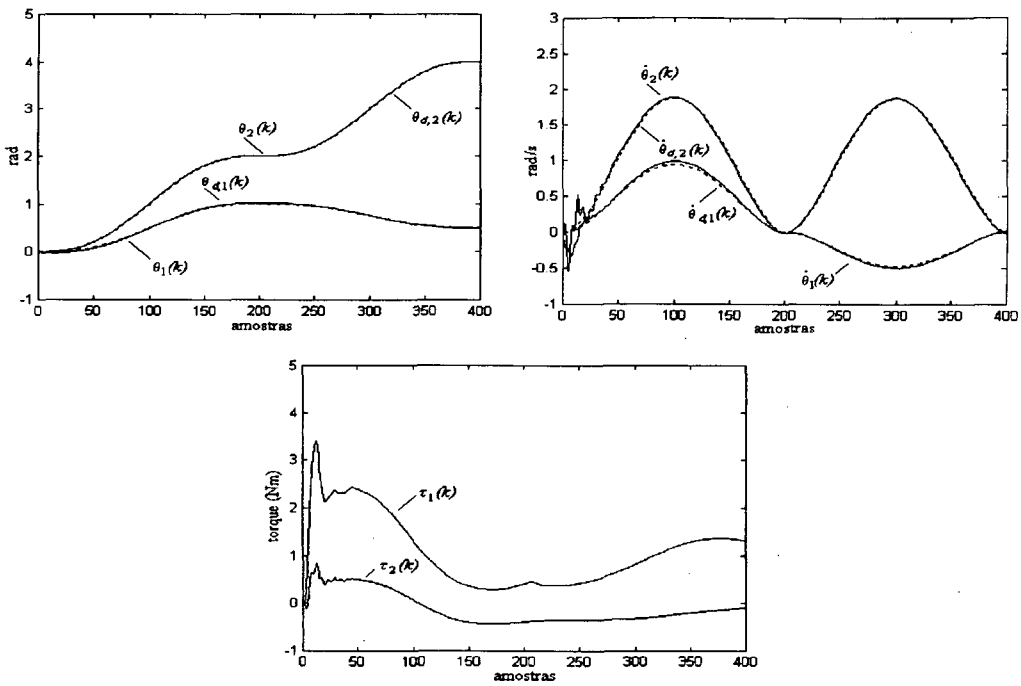


Figura 9.21. CLN-(PD+PI) (comportamento servo).

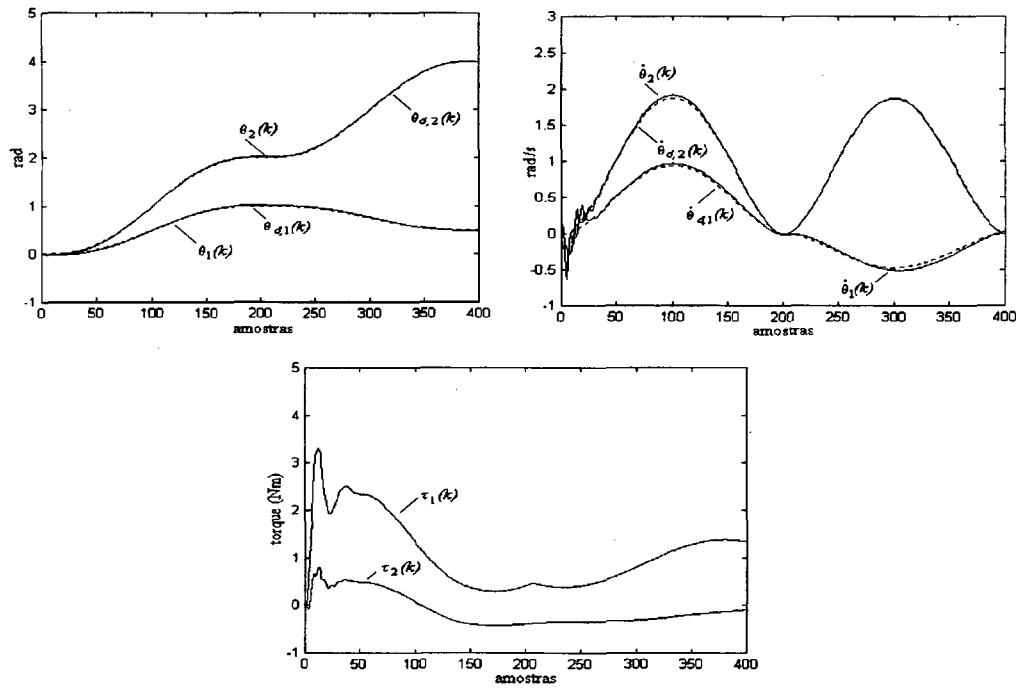


Figura 9.22. Controle neural (comportamento servo).

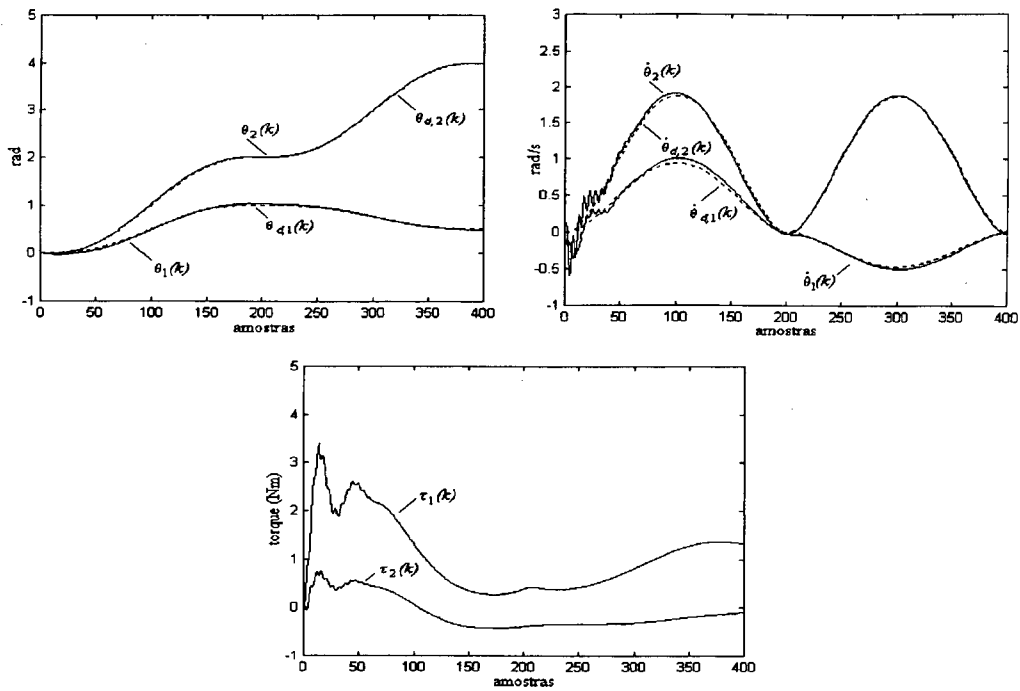


Figura 9.23. Controle através de SHI combinando CLN-(PD+I) incremental com ajuste dos fatores de escala via EEMC (comportamento servo).

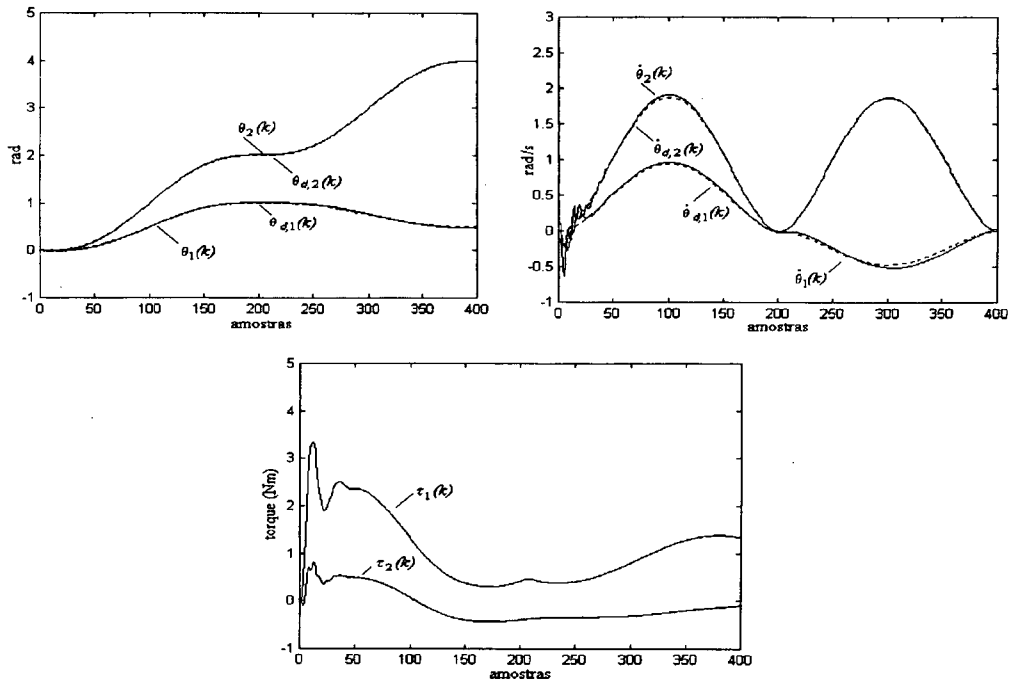


Figura 9.24. Controle através de SHI neural com otimização evolutiva (comportamento servo).

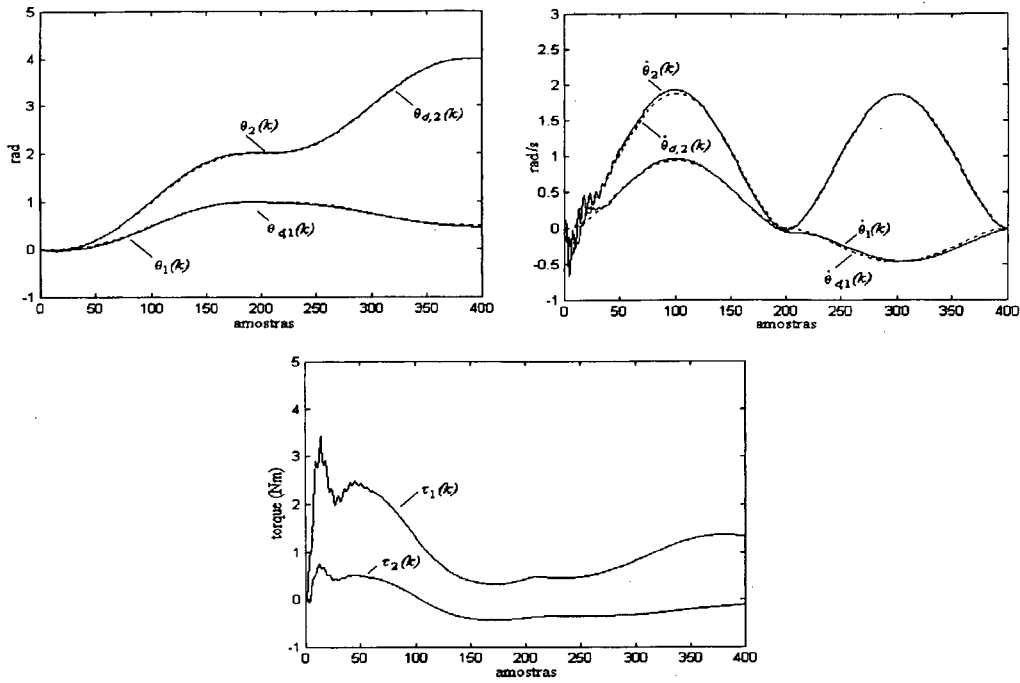


Figura 9.25. Controle através de *SHI* com estrutura neuro-nebuloso-evolutiva (comportamento servo).

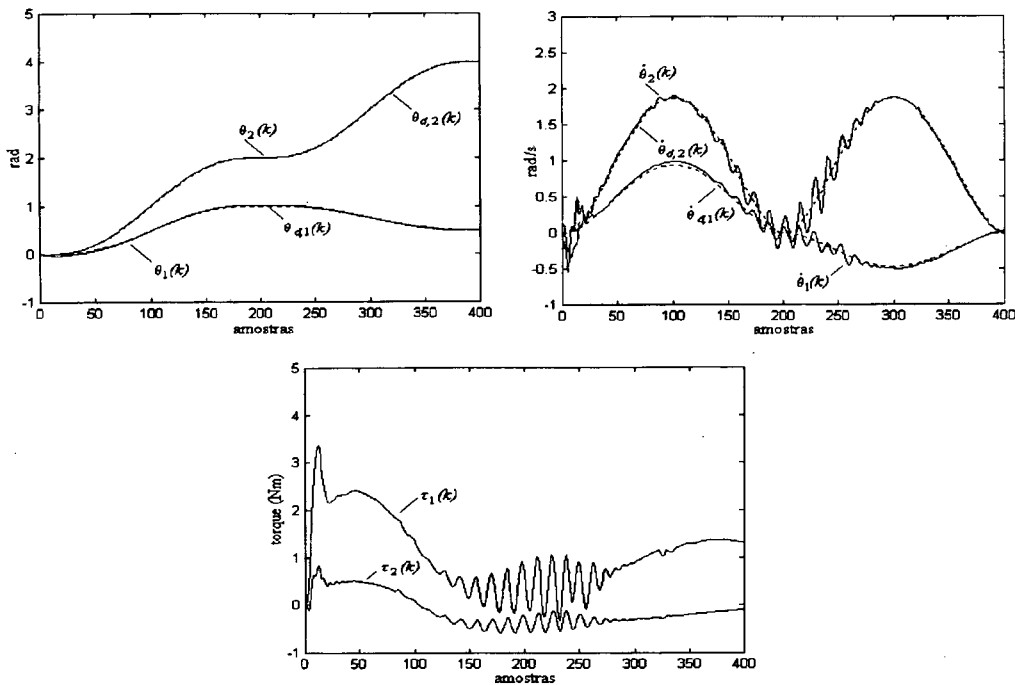


Figura 9.26. *CLN-(PD+I)* do manipulador robótico (comportamento regulatório).

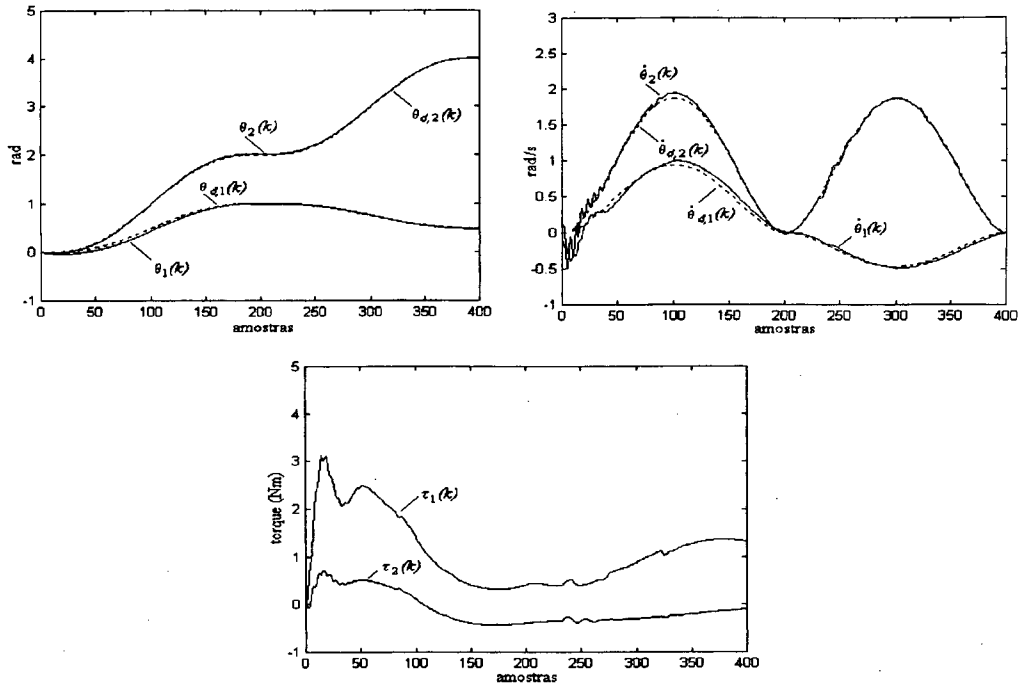


Figura 9.27. *CLN-PI* nebuloso + (D convencional) (comportamento regulatório).

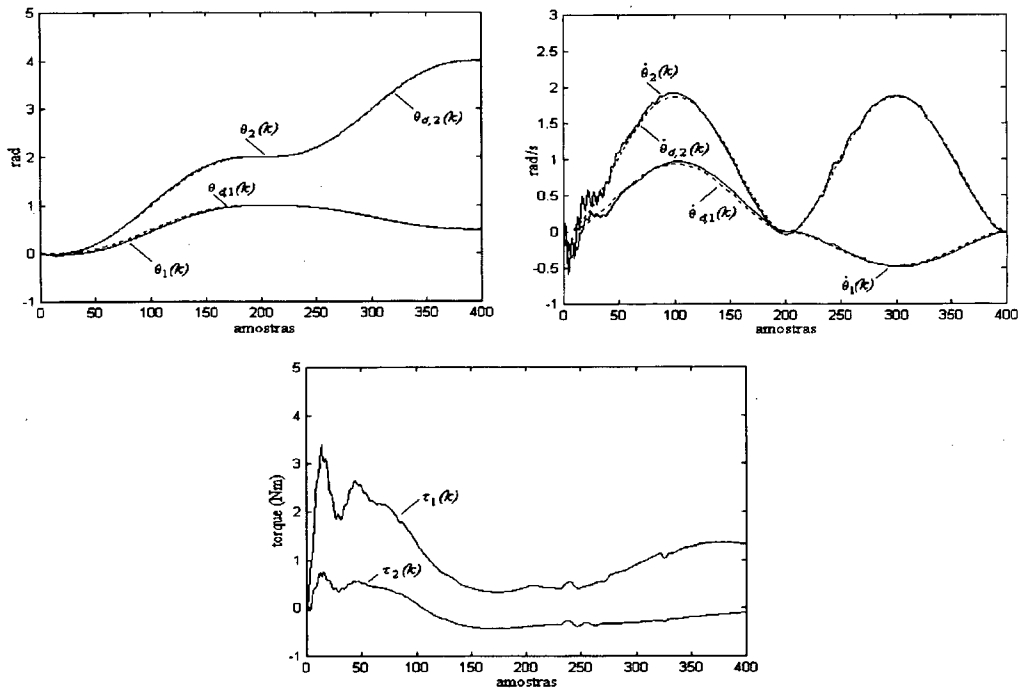


Figura 9.28. *CLN-(PD+I)* incremental (comportamento regulatório).

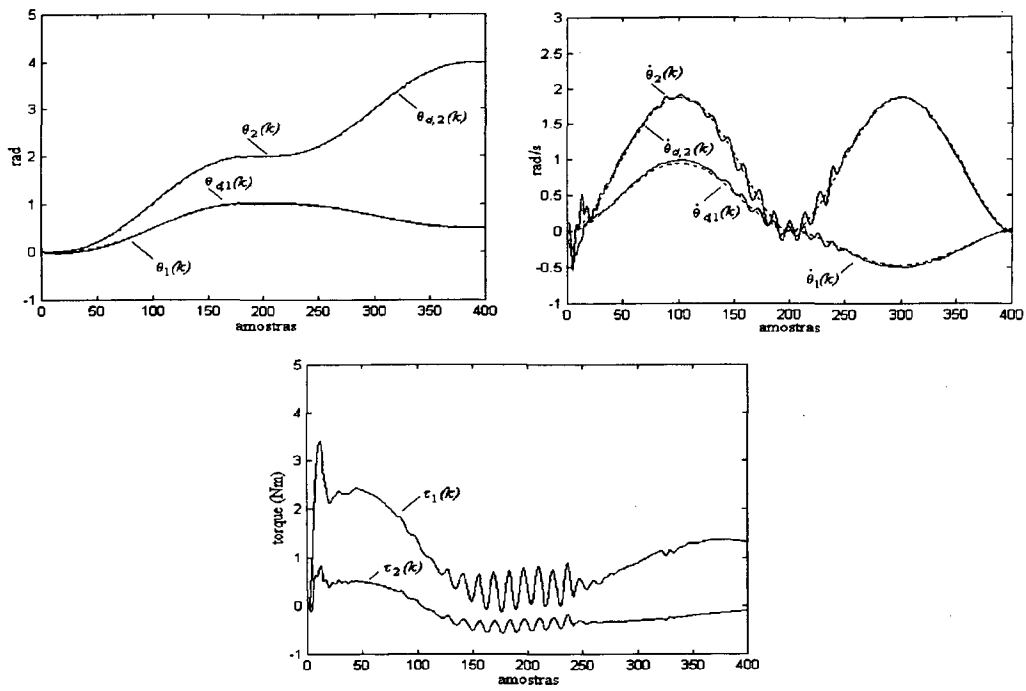


Figura 9.29. CLN-(PD+PI) (comportamento regulatório).

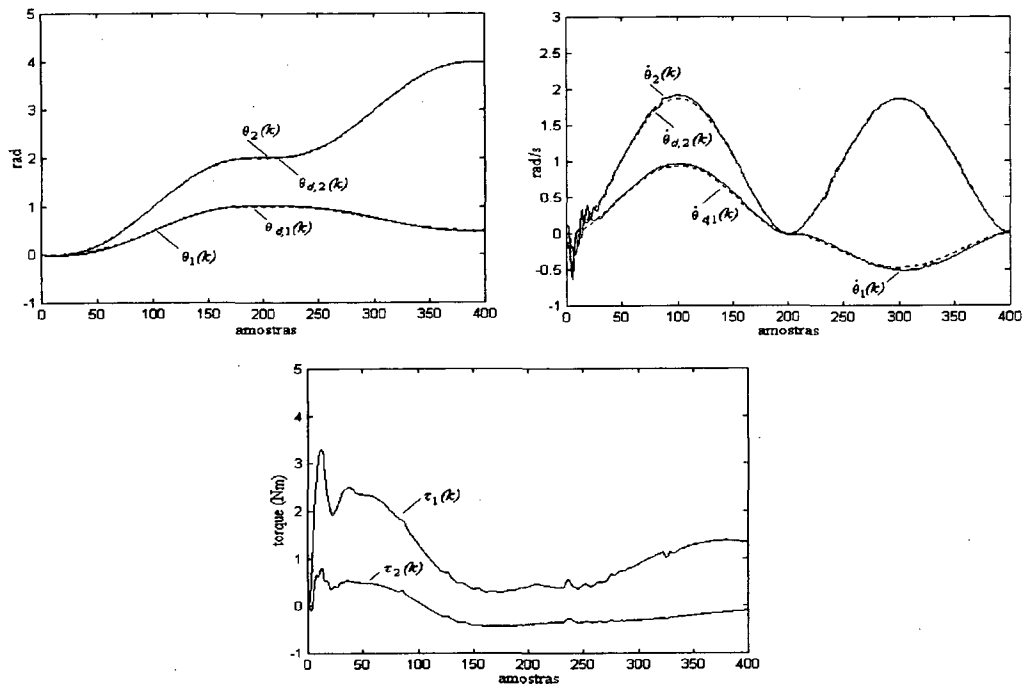


Figura 9.30. Controle neural (comportamento regulatório).

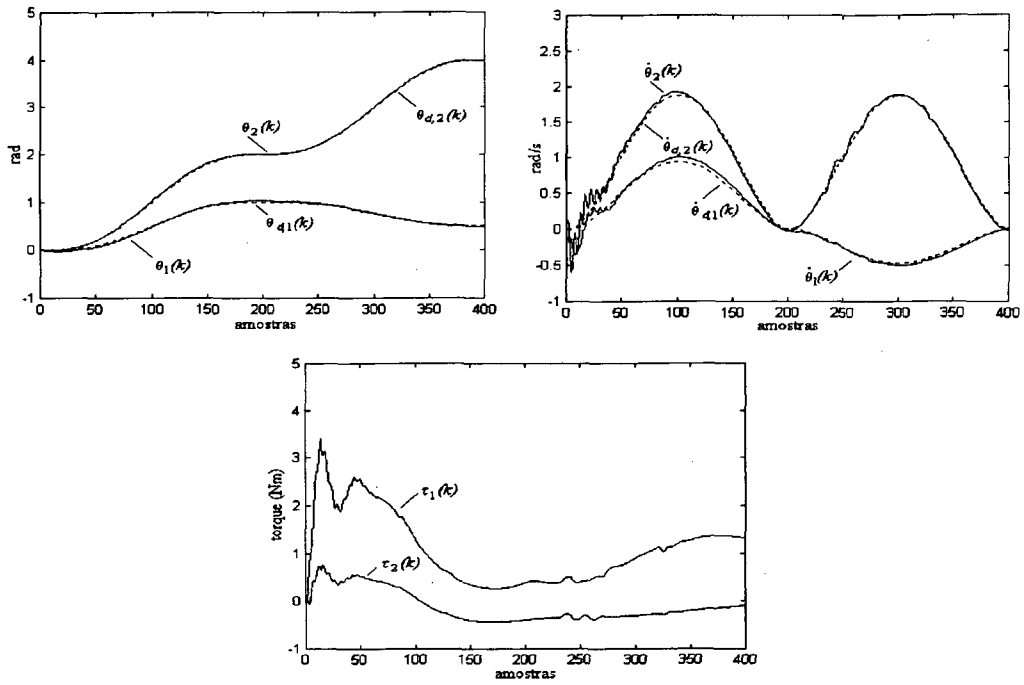


Figura 9.31. Controle através de SHI combinando CLN-(PD+I) incremental com ajuste dos fatores de escala via EEMC (comportamento regulatório).

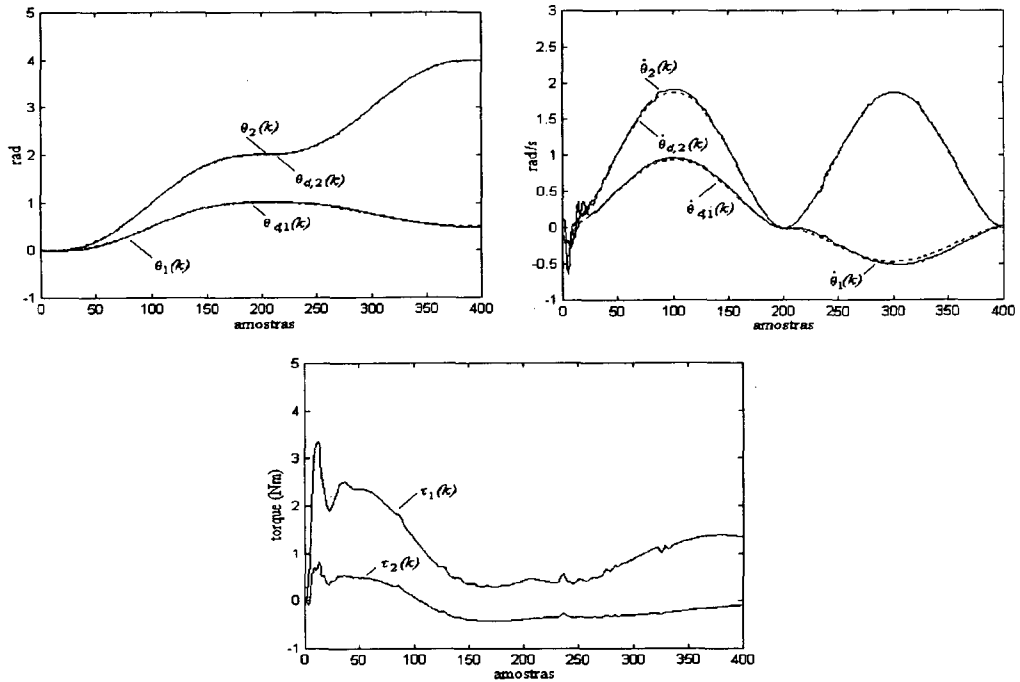


Figura 9.32. Controle através de SHI neural com otimização evolutiva (comportamento regulatório).

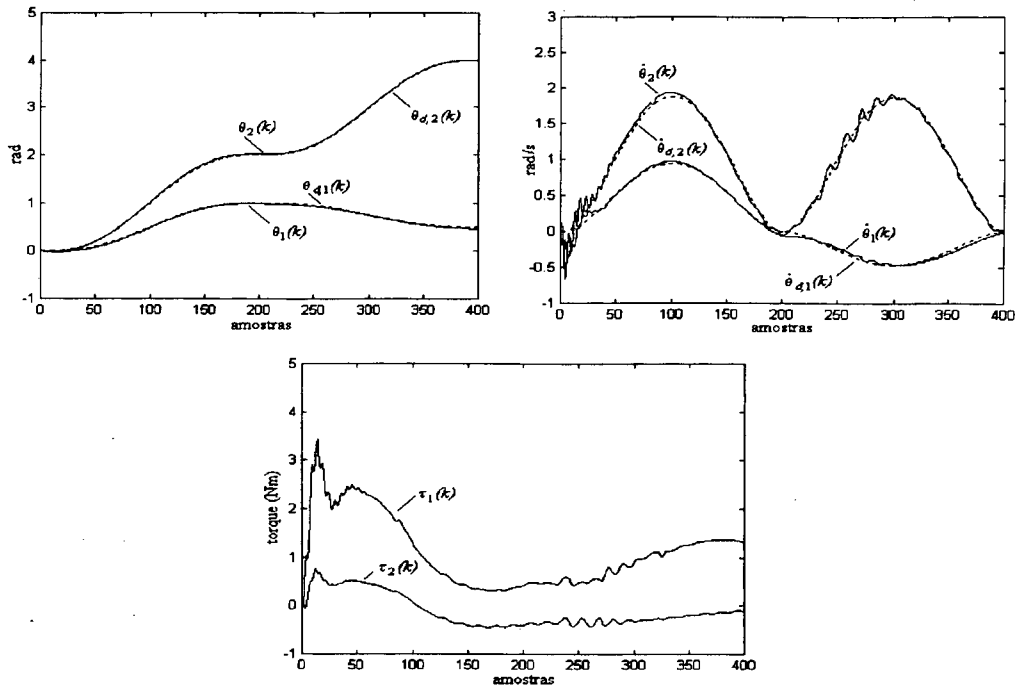


Figura 9.33. Controle através de *SHI* com estrutura neuro-nebuloso-evolutiva (comportamento regulatório).

Quanto ao comportamento servo (figuras 9.18 a 9.25), os projetos baseados em *IC* apresentam o seguimento das trajetórias de posição e velocidade de forma apropriada, exceto o *CLN-(PD+I)*, que apresenta oscilações quanto aos torques aplicados, e, conseqüentemente, na trajetória de velocidade resultante.

Pela análise do comportamento regulatório (figuras 9.26 a 9.31), nota-se que a maioria dos projetos são eficientes e apresentam robustez quando sujeitos a incertezas paramétricas. Entretanto, os projetos de *CLN-(PD+I)* e *CLN-(PD+PI)* apresentam oscilações excessivas do torque aplicado às juntas e no seguimento das trajetórias de velocidade. O *SHI* neuro-nebuloso-evolutivo apresenta algumas pequenas oscilações no seguimento de trajetória da velocidade, θ_2 . Quanto ao seguimento das trajetórias das posições todas as estruturas de projeto apresentam resultados promissores.

9.5 Projetos de GPC e QSMC adaptativos e inteligentes

Os projetos de GPC e QSMC, com abordagem adaptativa, através do algoritmo dos MQR objetivam o projeto baseado em modelos lineares acoplados. O modelo linear acoplado nº1 (AGARWAL & SEBORG, 1987) consiste de um modelo matemático para as posições e velocidades das duas juntas, onde:

$$\theta_1(k+1) = a_1\theta_1(k) + a_2\theta_1(k-1) + a_3\theta_2(k) + a_4\theta_2(k-1) + b_1\tau_1(k) + b_2\tau_2(k)$$

$$\theta_2(k+1) = a_5\theta_1(k) + a_6\theta_1(k-1) + a_7\theta_2(k) + a_8\theta_2(k-1) + b_3\tau_1(k) + b_4\tau_2(k)$$

$$\dot{\theta}_1(k+1) = a_7\dot{\theta}_1(k) + a_8\dot{\theta}_1(k-1) + a_9\dot{\theta}_2(k) + a_{10}\dot{\theta}_2(k-1) + b_5\tau_1(k) + b_6\tau_2(k)$$

$$\dot{\theta}_2(k+1) = a_{11}\dot{\theta}_1(k) + a_{12}\dot{\theta}_1(k-1) + a_{13}\dot{\theta}_2(k) + a_{14}\dot{\theta}_2(k-1) + b_7\tau_1(k) + b_8\tau_2(k)$$

O modelo linear acoplado nº2 (ALVAREZ *et al.*, 1998) consiste de um modelo de 4ª ordem para as posições e velocidades das duas juntas, onde:

$$\begin{aligned} \theta_1(k+1) = & a_1\theta_1(k) + a_2\theta_1(k-1) + a_3\theta_1(k-2) + a_4\theta_1(k-3) + \\ & a_5\theta_2(k) + a_6\theta_2(k-1) + a_7\theta_2(k-2) + a_8\theta_2(k-3) + \\ & b_1\tau_1(k) + b_2\tau_1(k-1) + b_3\tau_1(k-2) + b_4\tau_1(k-3) + \\ & b_5\tau_2(k) + b_6\tau_2(k-1) + b_7\tau_2(k-2) + b_8\tau_2(k-3) \end{aligned}$$

$$\begin{aligned} \theta_2(k+1) = & a_9\theta_1(k) + a_{10}\theta_1(k-1) + a_{11}\theta_1(k-2) + a_{12}\theta_1(k-3) + \\ & a_{13}\theta_2(k) + a_{14}\theta_2(k-1) + a_{15}\theta_2(k-2) + a_{16}\theta_2(k-3) + \\ & b_9\tau_1(k) + b_{10}\tau_1(k-1) + b_{11}\tau_1(k-2) + b_{12}\tau_1(k-3) + \\ & b_{13}\tau_2(k) + b_{14}\tau_2(k-1) + b_{15}\tau_2(k-2) + b_{16}\tau_2(k-3) \end{aligned}$$

$$\begin{aligned} \dot{\theta}_1(k+1) = & a_{17}\dot{\theta}_1(k) + a_{18}\dot{\theta}_1(k-1) + a_{19}\dot{\theta}_1(k-2) + a_{20}\dot{\theta}_1(k-3) + \\ & a_{21}\dot{\theta}_2(k) + a_{22}\dot{\theta}_2(k-1) + a_{23}\dot{\theta}_2(k-2) + a_{24}\dot{\theta}_2(k-3) + \\ & b_{17}\tau_1(k) + b_{18}\tau_1(k-1) + b_{19}\tau_1(k-2) + b_{20}\tau_1(k-3) + \\ & b_{21}\tau_2(k) + b_{22}\tau_2(k-1) + b_{23}\tau_2(k-2) + b_{24}\tau_2(k-3) \end{aligned}$$

$$\begin{aligned} \dot{\theta}_2(k+1) = & a_{25}\dot{\theta}_1(k) + a_{26}\dot{\theta}_1(k-1) + a_{27}\dot{\theta}_1(k-2) + a_{28}\dot{\theta}_1(k-3) + \\ & a_{29}\dot{\theta}_2(k) + a_{30}\dot{\theta}_2(k-1) + a_{31}\dot{\theta}_2(k-2) + a_{32}\dot{\theta}_2(k-3) + \\ & b_{25}\tau_1(k) + b_{26}\tau_1(k-1) + b_{27}\tau_1(k-2) + b_{28}\tau_1(k-3) + \\ & b_{29}\tau_2(k) + b_{30}\tau_2(k-1) + b_{31}\tau_2(k-2) + b_{32}\tau_2(k-3) \end{aligned}$$

Os projetos de controle utilizados apresentam as seguintes configurações:

- (i) GPC adaptativo tem parâmetros de projeto $NU=1$, $NY=6$ e $\Gamma \approx 0.1I_2$;
- (ii) QSMC adaptativo; e

(iii) os *GPC* baseados em modelos da *IC*, utilizam os modelos apresentados no capítulo 8 quando da identificação da dinâmica do manipulador robótico.

A figura 9.34 apresenta um resumo dos melhores resultados obtidos com os controladores preditivos e de estrutura variável aplicados ao controle de um manipulador robótico.

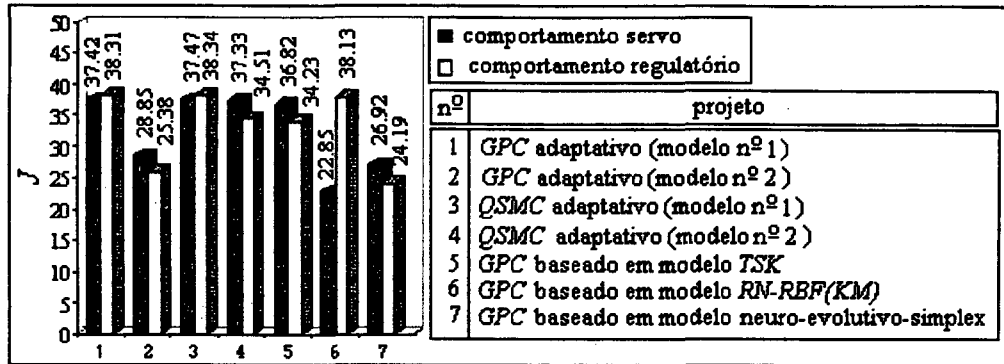


Figura 9.34. Análise dos resultados de estruturas de *GPC* e *QSMC*.

Observa-se pela figura 3.34 que o *GPC* e o *QSMC* apresentam resultados melhores com o modelo nº2 que o modelo nº1. Entretanto, o modelo nº2 tem um maior número de parâmetros a serem estimados e, conseqüentemente, maior complexidade computacional ocasionada pela número de operações do estimador dos *MQR* a cada amostra.

Os controladores preditivos baseados em *IC* apresentam resultados promissores, tanto em comportamento servo quanto regulatório. Entretanto, o *GPC* baseado em *RN-RBF(KM)* apresentou excelente resultado quanto ao comportamento servo, mas teve seu desempenho degradado quanto ao comportamento regulatório.

As figuras 9.35 a 9.48 apresentam os resultados de controle para os testes de comportamento servo e regulatório.

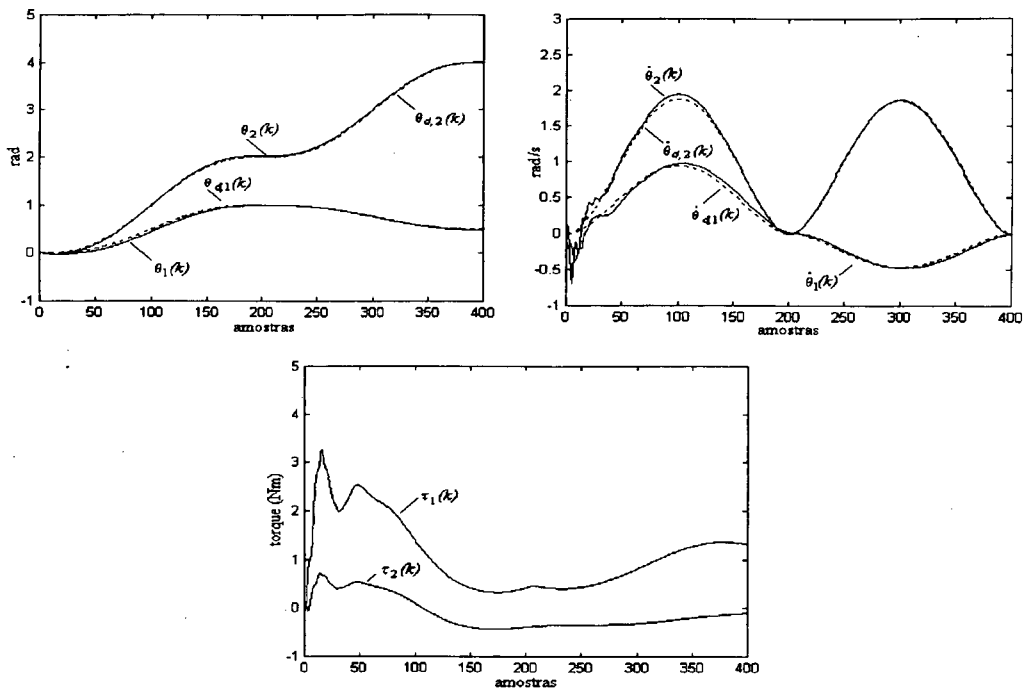


Figura 9.35. Comportamento servo do GPC adaptativo (modelo n°1).

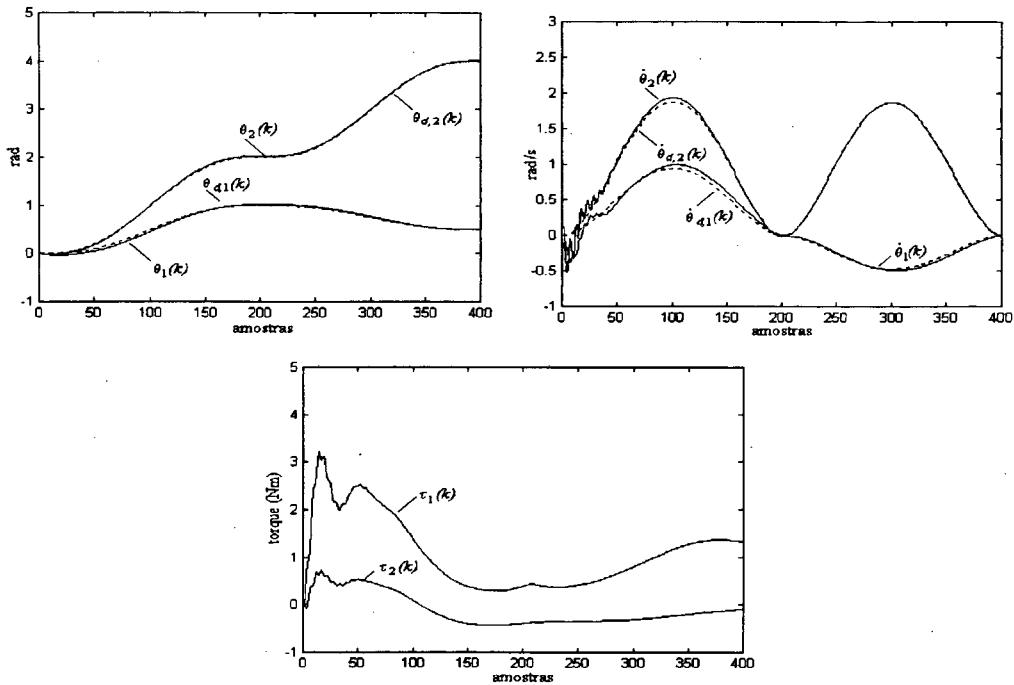


Figura 9.36. Comportamento servo GPC adaptativo (modelo n°2).

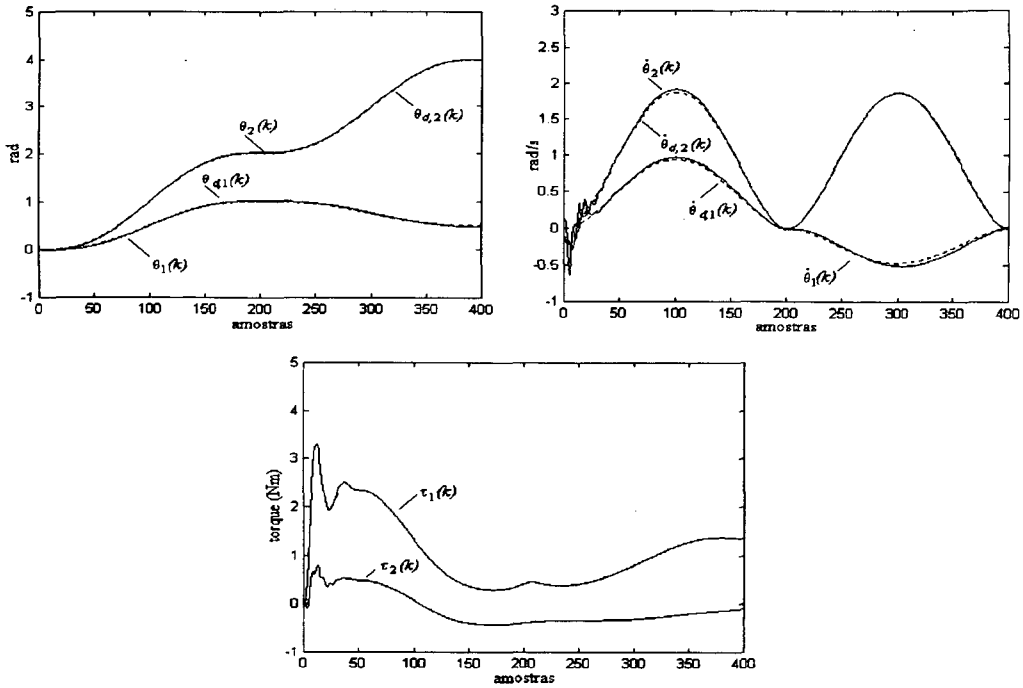


Figura 9.37. Comportamento servo do QSMC adaptativo (modelo n°1).

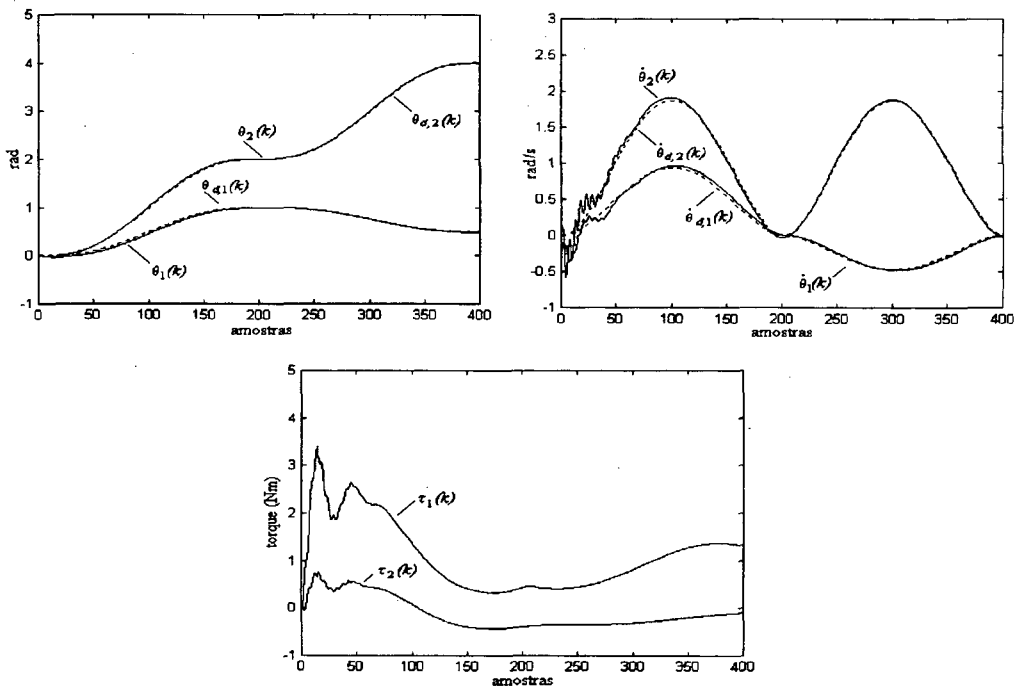


Figura 9.38. Comportamento servo do QSMC GPC adaptativo (modelo n°2).

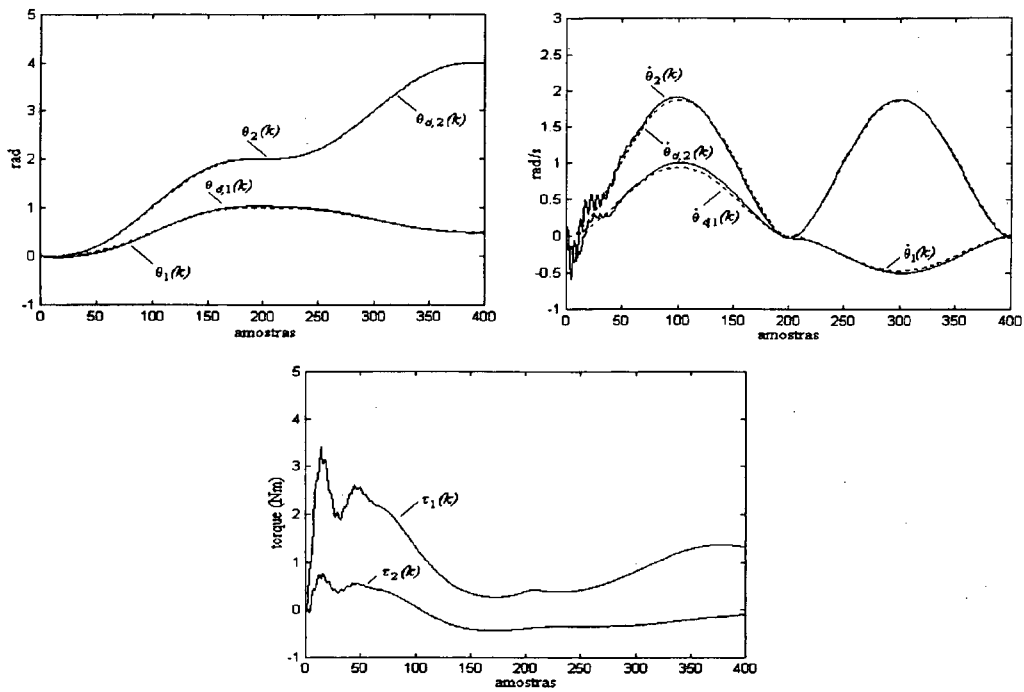


Figura 9.39. Comportamento servo do GPC baseado em modelo TSK.

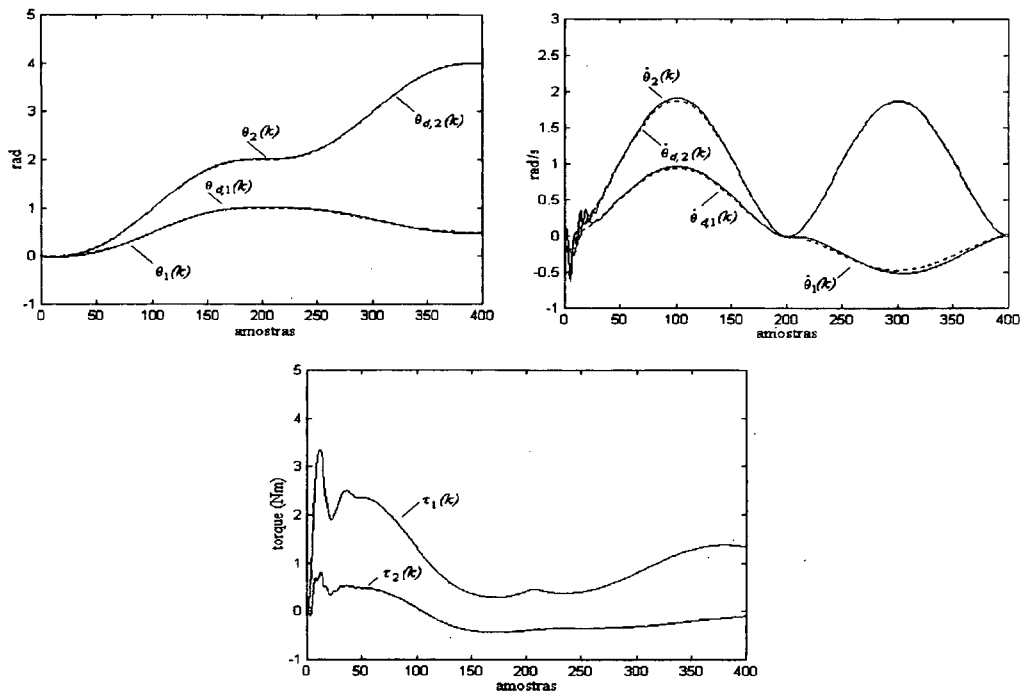


Figura 9.40. Comportamento servo do GPC baseado em RN-RBF(KM).

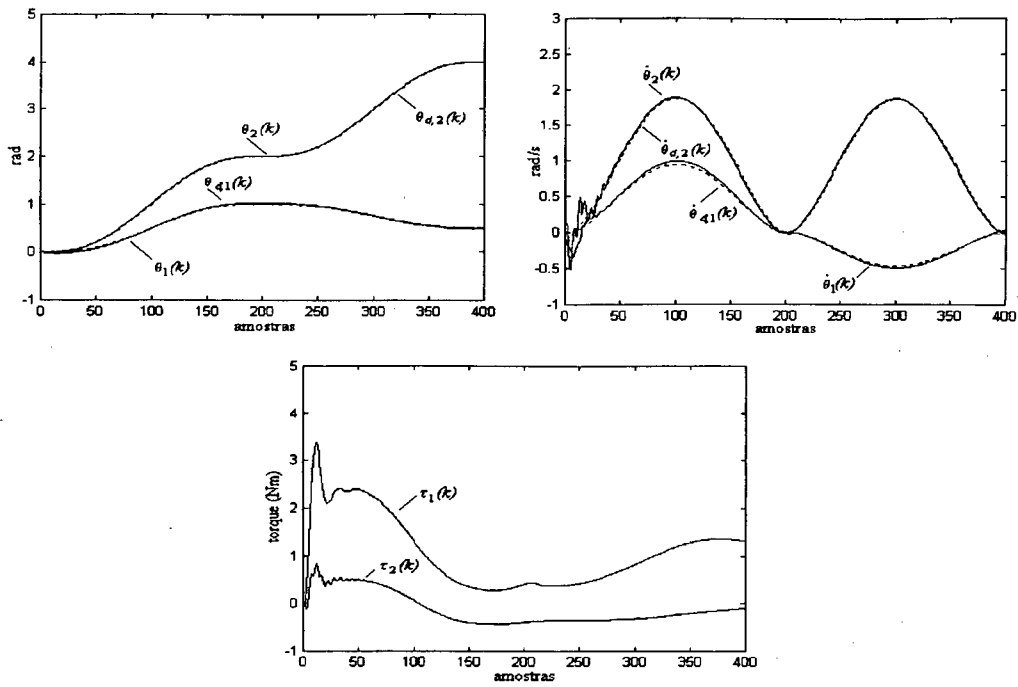


Figura 9.41. Comportamento servo do GPC baseado em RN-RBF e otimização via PE e método simplex.

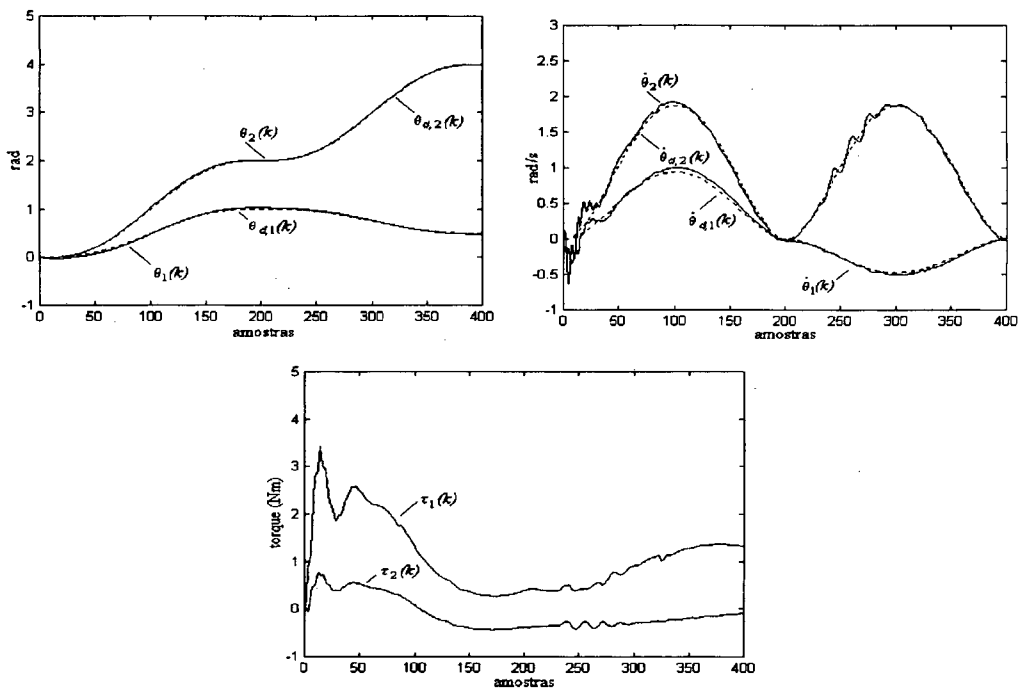


Figura 9.42. Comportamento regulatório do GPC adaptativo (modelo n°1).

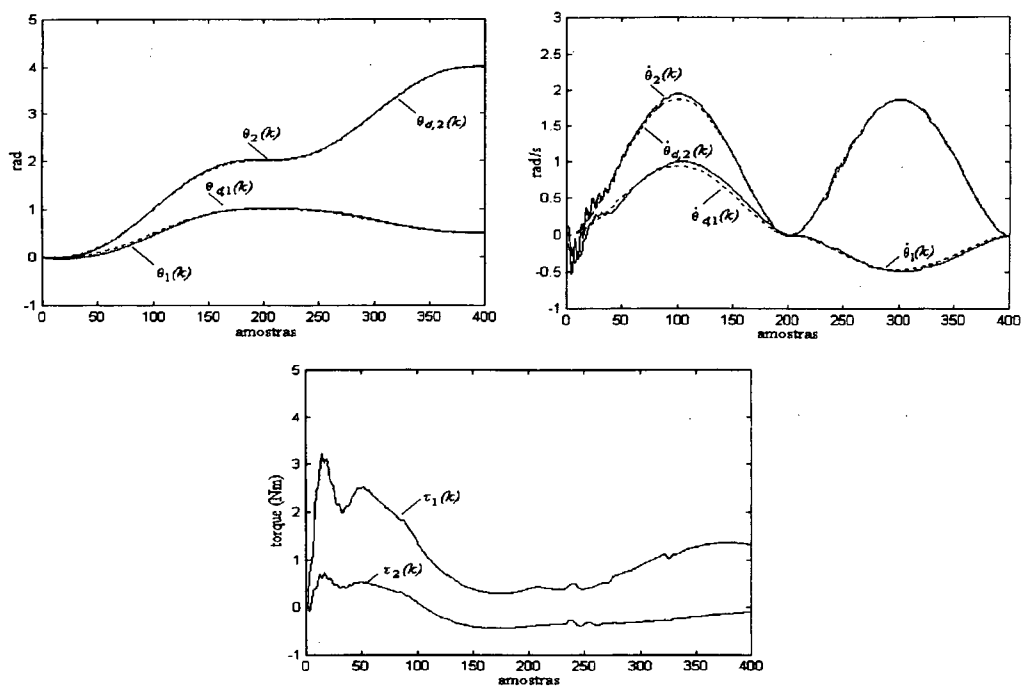


Figura 9.43. Comportamento regulatório do GPC adaptativo (modelo n°2).

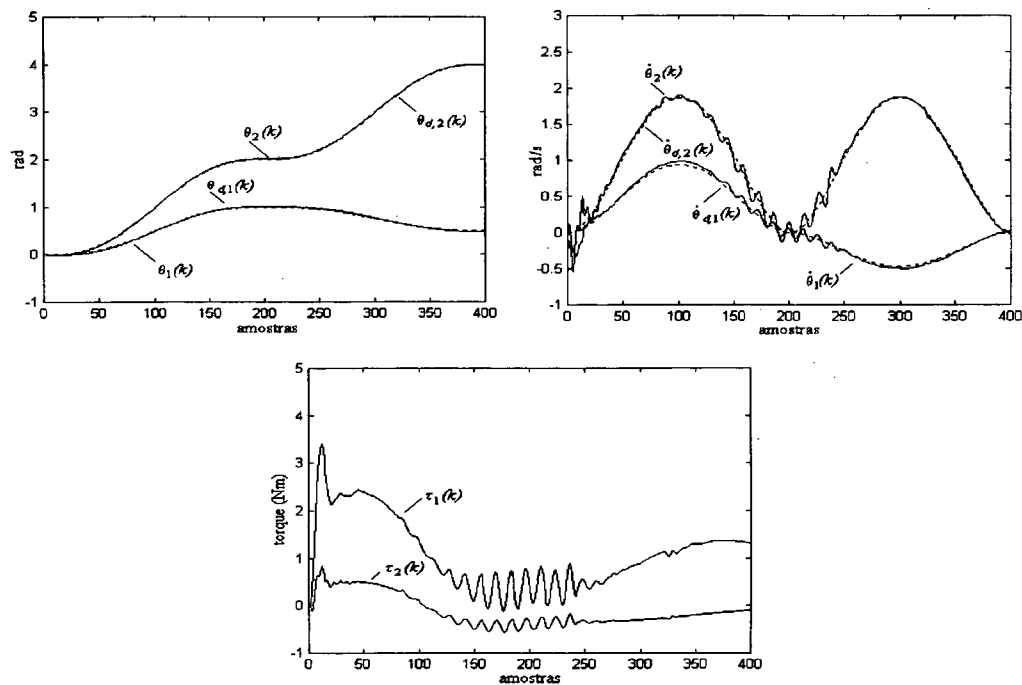


Figura 9.44. Comportamento regulatório do QSMC (modelo n°1).

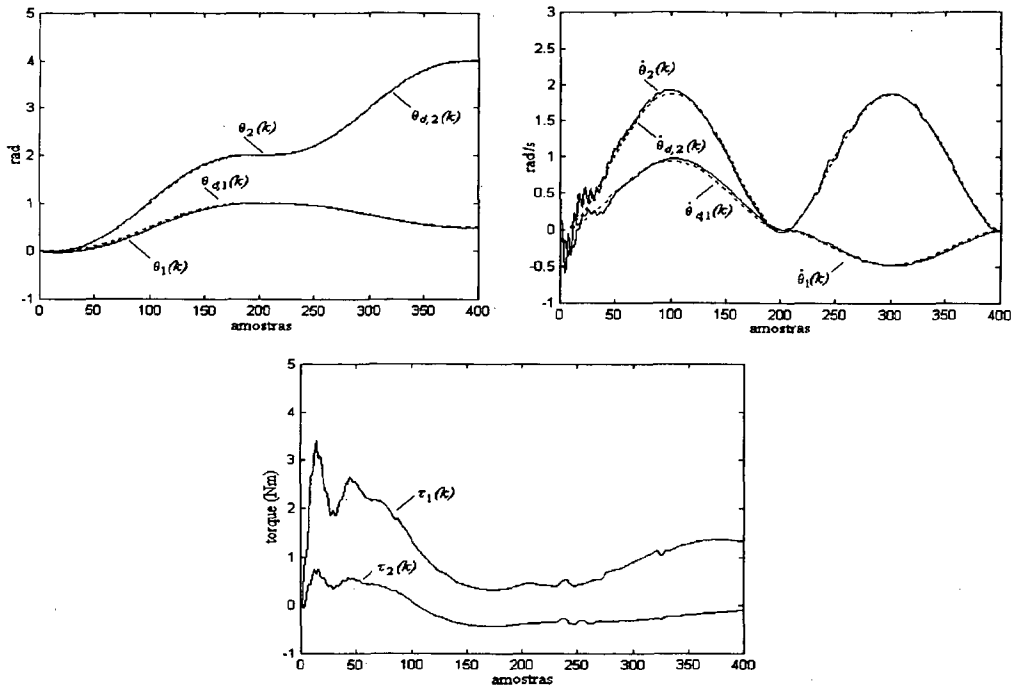


Figura 9.45. Comportamento regulatório do *QSMC* adaptativo (modelo nº2).

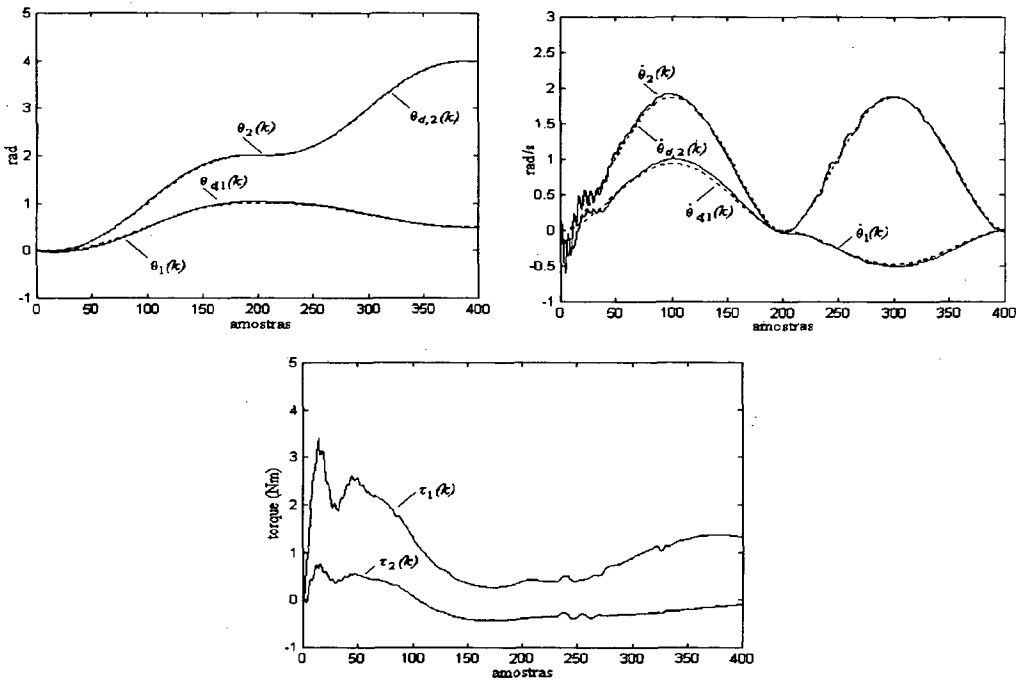


Figura 9.46. Comportamento regulatório do *GPC* baseado em modelo *TSK*.

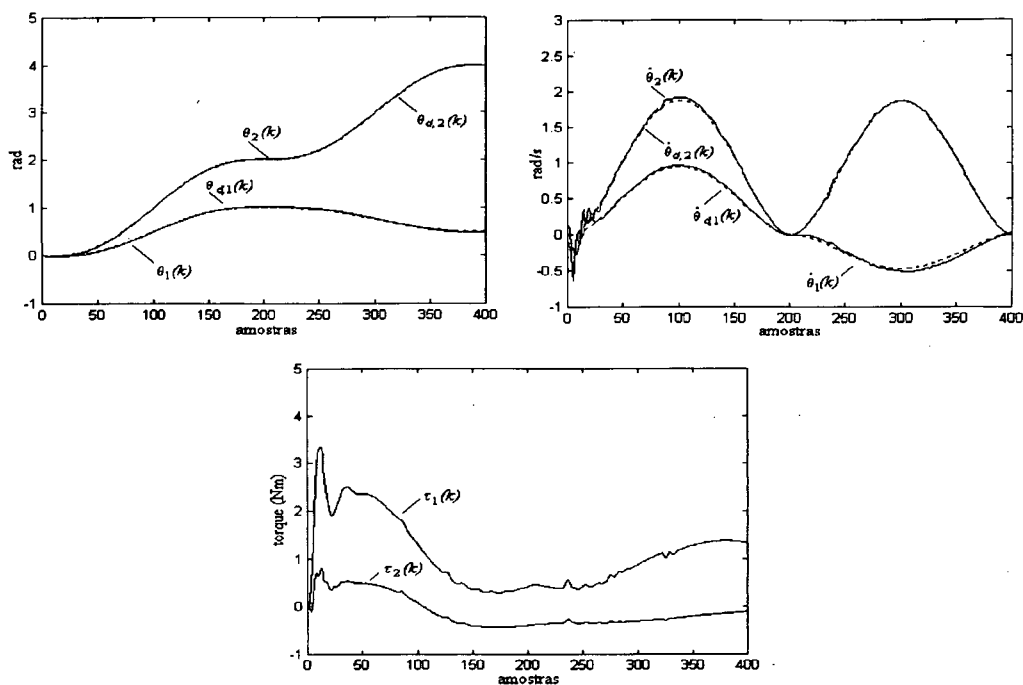


Figura 9.47. Comportamento regulatório do GPC baseado em *RN-RBF(KM)*.

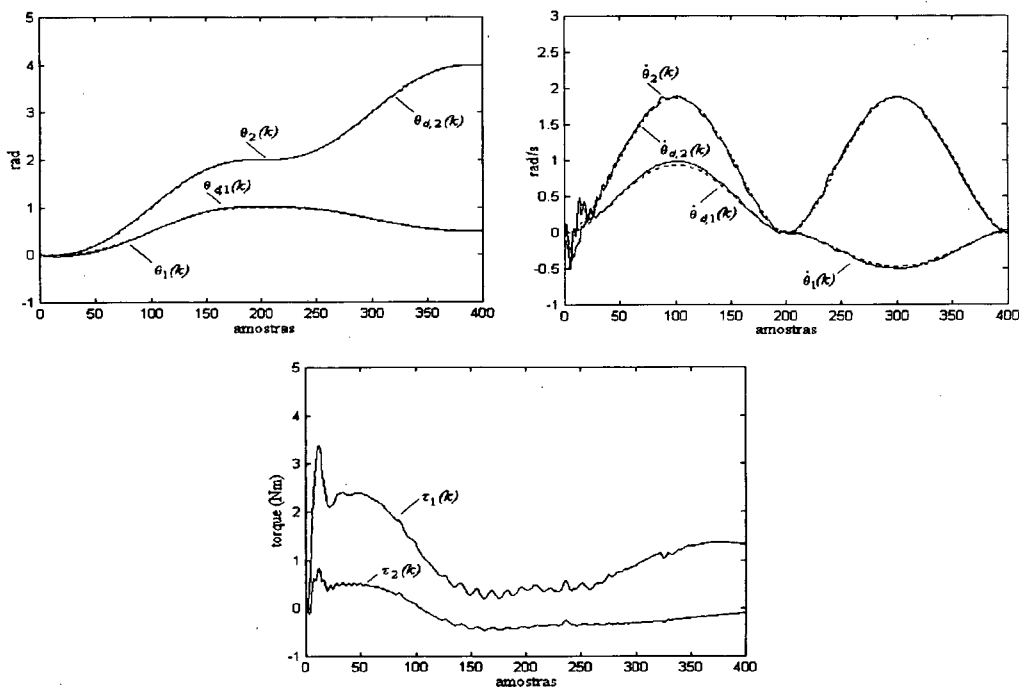


Figura 9.48. Comportamento regulatório do GPC baseado em *RN-RBF* e otimização via *PE* e método simplex.

Os resultados das figuras 9.35 a 9.48 mostram a eficiência dos controladores preditivos e *QSMC*. No entanto, o projeto do *QSMC* com modelo nº2 apresenta algumas oscilações quanto ao torques aplicados.

Os controladores preditivos mostram eficiência no controle do manipulador robótico, quando este está sujeito a variações paramétricas, onde a configuração adaptativa via estimador *MQR* tem papel relevante. Contudo, deve ser enfatizado que em controle adaptativo, o mecanismo de adaptação manipula variações paramétricas, contudo, em projetos mais elaborados para o tratamento de incertezas não-estruturadas necessita-se considerar aspectos de robustez no procedimento de controle.

9.6 Conclusão do capítulo

Neste capítulo foram tratados tópicos relativos à aplicação de estratégias de controle em um manipulador robótico. Foram realizados testes para a verificação de comportamento servo e regulatório dos projetos de controle inteligentes, preditivos e de estrutura variável.

Os resultados apresentados em controle multivariável mostraram que as técnicas de controle analisadas são adequadas para o controle de um manipulador robótico, principalmente no seguimento de trajetórias de posição. Quanto ao seguimento das trajetórias de velocidade, nos testes de comportamento regulatório, algumas metodologias apresentaram resultados inadequados.

Os estudos de convergência e análise dos resultados da sintonia através de *AEs* do controle *PD* e *PID*, na forma *multiloop* e acoplada, demonstraram a eficiência dos *AEs* na otimização de projetos de controle. A nova proposta de controle na forma acoplada mostra-se promissora para aplicações em controle de processos não-lineares e que apresentem complexidades de acoplamentos entre as variáveis. Entretanto, os resultados desta abordagem também podem ser aprimorados pela configuração de uma função objetivo que considere o torque e/ou a variação do torque.

Os *AEs* constituem-se de algoritmos de otimização de propósito geral e apresentam simplicidade de adaptação para as mais diversas classes de problemas de controle multivariável, tanto linear quanto não linear. As vantagens dos *AEs* são relacionadas a: (i)

tratarem processos sujeitos a restrições; (ii) não necessitem de informação de derivadas do critério; (iii) tratem problemas multimodais e com múltiplos objetivos, (iv) permitirem a utilização do conhecimento obtido *a priori* pelo projetista.

Entretanto, para obtenção de um desempenho adequado da sintonia evolutiva de um controlador, o projetista deve ter em mente aspectos essenciais, tais como: (i) a especificação do desempenho desejado e formulação de uma função de aptidão apropriada; (ii) a técnica de projeto do controlador deve ser compatível com o desempenho desejado e o modelo do processo; (iii) presença de um procedimento de validação do controlador sintonizado via *AEs*; (iv) os *AEs* podem se beneficiar da utilização de ambientes de processamento paralelo.

Entre as desvantagens dos *AEs* em aplicações de sintonia de controladores multivariáveis deve-se mencionar: (i) os *AEs* não garantem a obtenção de uma solução absolutamente ótima; (ii) existe dificuldade de prever quantas gerações são necessárias para produzir um resultado satisfatório. Deve ser enfatizado, mais uma vez, que a maior limitação dos *AEs* em aplicações de sintonia de controladores multivariáveis é a complexidade computacional, e conseqüentemente, dificuldade de projetar-se controladores de forma *on-line*. Uma alternativa usual é adotar-se os seguintes passos: (i) identificar um modelo que represente adequadamente o processo real, (ii) sintonizar os parâmetros do controlador utilizando o modelo que representa o processo real, e finalmente (iii) validar os parâmetros do controlador obtidos via *AEs* no processo real.

As configurações de controle *PID* nebulosos analisados apresentaram resultados de controle aceitáveis. Entretanto, o projeto automático através de *AEs* é uma alternativa relevante para projetos de controle nebulosos, conforme apresentado pelo *SHI* que consiste de um *CLN-(PD+I)* incremental com otimização via *EEMC*. A vantagem da configuração de estruturas de controle *PID* nebulosos em relação a adoção de controladores *PID* convencionais está na maior flexibilidade e eficiência do projeto nebuloso. Os controladores *PID* nebulosos são adequados ao tratamento de processos não-lineares. Entretanto, as principais desvantagens são vinculadas: maior número de parâmetros a serem ajustados que no projeto de um controlador *PID* convencional, dificuldades de prova de estabilidade e a obtenção de um projeto de *PID* nebuloso “ótimo” é uma tarefa complexa.

Os projetos de controle neural via *RN-RBF(KM)*, *SHI* neuro-nebuloso-evolutivo e *GPC* baseado em *SHI* foram adequados ao controle do manipulador robótico. Entretanto, a essência destas estruturas de projeto foi a fase de identificação da dinâmica do manipulador robótico dentro da região de interesse.

As estratégias de *GPC* e *QSMC*, na forma adaptativa, tiveram desempenho promissor quanto a análise do comportamento servo e regulatório. No entanto, a utilização do estimador do *MQR* melhora as capacidades de robustez do controlador frente a incertezas e mudanças paramétricas, mas tem a desvantagem de inserir uma significativa complexidade computacional nas aplicações em controle de processos rápidos.

A utilização de *GPC* baseado em modelo nebuloso constituiu-se da utilização de um modelo de *TSK* que é uma estrutura poderosa para representação de sistemas dinâmicos não lineares. A identificação do processo multivariável via *TSK* habilitou a incorporação do modelo nebuloso como um preditor no esquema de *GPC*. Os projetos de *MBPCs* baseados em sistemas nebulosos, em geral, têm vantagens, pois a idéia básica do controle preditivo nebuloso é simples e intuitiva.

O controle preditivo nebuloso tem a vantagem de poder detectar dinâmicas não-lineares de processos multivariáveis e tratar processos quando o conhecimento *a priori* sobre a estrutura do processo não é avaliado ou somente parcialmente conhecido. Conseqüentemente, um sistema nebuloso, não necessariamente de *TSK*, pode ser utilizado de forma a melhorar a flexibilidade do projeto e o desempenho de um controlador preditivo sujeito a diversas condições de operação.

Entretanto, quando o processo multivariável apresenta características, tais como: dinâmica variante no tempo, desconhecimento do atraso de transporte não-unitário e a ordem do processo é elevada, necessita-se fazer um estudo detalhado da viabilidade e adequação de uma determinada estrutura de *SN* para o tratamento destes processos. Possivelmente necessita-se aplicar algumas metodologias adaptativas (sintonia de parâmetros *on-line* via *MQR*) e técnicas para a detecção de ordem e atraso de transporte do processo multivariável. Contudo, a construção de um modelo nebuloso para o projeto de *MBPC* apresenta algumas dificuldades que devem ser investigadas com cuidado, principalmente o número de funções de pertinência e regras, a escolha das variáveis na premissa e/ou conseqüente, entre outros. Entre as desvantagens da utilização dos modelos

nebulosos estão a utilização de abordagens de projeto heurísticas motivadas pela inexistência de uma ferramenta eficiente e consolidada de projeto.

O projeto de *GPC* neural também é uma ferramenta que tem vantagens para o tratamento de processos multivariáveis complexos, com características de não-linearidades e acoplamentos entre as variáveis, que geralmente são difíceis de serem avaliados com as técnicas convencionais de projeto. A utilização de *RN-RBF* apresenta a vantagem de ser treinada em dois estágios, isto é, (i) aprendizado dos centros e comprimentos das funções de base da camada intermediária e (ii) aprendizado dos pesos da camada intermediária para a camada de saída. Em outras palavras, os dois estágios podem ser realizados na identificação do processo multivariável. Entretanto, para configurações de controle preditivo adaptativo pode-se utilizar o modelo neural obtido em uma identificação *a priori* e apenas realizar o estágio (ii) via algoritmo dos *MQR* (ou mesmo *OLS*, com maior complexidade computacional) por tratar de um problema linear nos parâmetros.

Algumas limitações deste projeto de controle preditivo neural estão na necessidade de um projeto adequado do sinal de entrada para excitação adequada das dinâmicas do processo não-linear em toda a região de interesse, além da determinação da quantidade de dados de treinamento necessária para esta finalidade. Uma observação relevante é que as *RNs* exibem desempenho adequado quando avaliadas para dados de treinamento, mas o desempenho da *RN* tende a ser inadequado se novos dados (acentuadamente diferentes) são avaliados. Outras limitações estão vinculadas ao projeto da *RN*, tais como: (i) a dificuldade na determinação de uma topologia de *RN* adequada da quantidade necessária de entradas e saídas (atrasadas) a serem utilizadas na entrada da *RN* para previsão; (ii) escolha do número de elementos na camada intermediária de forma a obter-se uma representação precisa das não-linearidades do processo; e (iii) determinação do critério de parada do treinamento de alguns tipos de *RNs*.

Outra forma de projeto de *GPC* é a utilização de sistemas híbridos inteligentes. Neste caso a combinação da capacidade de aprendizado das *RNs*, mecanismos de inferência e habilidade de tratamento com a incerteza dos *SNs* e as capacidades de otimização e aprimoramento do projeto através de *AEs* pode ser útil no tratamento de processos multivariáveis complexos, entretanto, os *SHIs* apresentam limitações devido a complexidade computacional e necessidade do projetista apresentar um conhecimento

amplo das vantagens e desvantagens de cada algoritmo inteligente para configurar um projeto eficiente, e, principalmente, viável para aplicações industriais.

Capítulo 10

CONCLUSÃO E PERSPECTIVAS

10.1 Conclusão

Esta tese apresentou os fundamentos e a aplicação de metodologias de controle preditivo, estrutura variável e inteligência computacional para o tratamento de processos não-lineares multivariáveis. A tese foi dividida em aspectos de projeto e aplicações em duas áreas:

- (i) identificação de processos não-lineares multivariáveis; e
- (ii) controle de processos não-lineares multivariáveis.

As potencialidades e as características do projeto multivariável, utilizando-se as metodologias da inteligência computacional, foram tratadas em detalhes nos capítulos 2 a 5. As abordagens de projeto via inteligência computacional foram relacionadas aos algoritmos evolutivos, sistemas nebulosos, redes neurais artificiais e sistemas híbridos inteligentes. Os algoritmos evolutivos abordados foram: os algoritmos genéticos, as estratégias evolutivas, a programação evolutiva e os algoritmos híbridos com *simulated annealing* e método simplex.

A configuração dos sistemas nebulosos objetivou a análise do sistema nebuloso com geração de regras, através de aprendizado modificado para identificação de processos, enquanto em controle de processos enfatizou-se a estruturação de controladores do tipo proporcional-integral-derivativo nebulosos. A outra filosofia da inteligência computacional utilizada foram as redes neurais artificiais. Foram descritas e analisadas as redes neurais artificiais: *perceptron* multicamadas, parcialmente recorrente de Elman modificada, *Group Method of Data Handling* e função de base radial.

Especificamente em controle de processos multivariáveis, as metodologias de controle preditivo generalizado e controle por modos quase-deslizantes combinado a controle de variância mínima generalizada, em configuração adaptativa, foram analisadas.

Além disso, algumas propostas de controle preditivo generalizado e controle por modos quase-deslizantes combinados a inteligência computacional foram também apresentadas.

Os procedimentos de identificação e controle multivariável foram avaliados através de simulações. As simulações realizadas visaram: a identificação de um processo experimental não-linear balanço horizontal, em malha fechada; a dinâmica de um manipulador robótico, com dois graus de liberdade; e um processo não-linear apresentando acoplamentos entre as variáveis de saída. A configuração e a análise dos algoritmos de controle trataram o controle do manipulador robótico, utilizado no procedimento de identificação. Neste estudo de caso, abordou-se a análise do comportamento servo e regulatório.

As aplicações realizadas em identificação e controle de processos multivariáveis possibilitaram a realização de alguns estudos comparativos, e a consequente obtenção de conclusões, quanto à classe de processos multivariáveis não-lineares.

10.1.1 Identificação de processos multivariáveis

As metodologias da inteligência computacional são poderosas ferramentas na configuração de procedimentos de identificação de processos multivariáveis não-lineares, conforme foi observado pelos resultados das simulações apresentados nos capítulos 8 e 9, e nas tabelas de aplicações industriais das metodologias inteligentes (Apêndice A).

Quanto a identificação de processos observou-se que as metodologias da inteligência computacional foram adequadas na identificação dos processos multivariáveis. Entretanto, um estudo comparativo “justo” das diferentes metodologias da inteligência computacional é uma tarefa complexa, pois cada abordagem tem suas potencialidades e também limitações para cada classe de problema.

Um outro agravante é que o número de parâmetros de projeto e as variáveis a serem otimizadas variam de um paradigma da inteligência computacional para outro. Apesar desta constatação algumas conclusões podem ser delineadas a partir dos resultados obtidos.

Os algoritmos evolutivos foram tratados na identificação experimental do processo balanço horizontal, em malha fechada. A abordagem proposta foi eficiente para a identificação da ordem, atraso de transporte e dos parâmetros do modelo matemático para

o processo balanço horizontal. O estudo comparativo de alguns algoritmos evolutivos serviu para a análise de diversos fatores essenciais à configuração de um algoritmo evolutivo adequado. Notou-se que a hibridização com as técnicas de *simulated annealing* e o método simplex melhorou o desempenho dos algoritmos evolutivos, comprovado pela convergência e a precisão dos resultados obtidos de um modelo matemático linearizado do processo balanço horizontal.

Um projeto de sistema nebuloso com geração de regras, através de aprendizado modificado também foi abordado em identificação do processo balanço horizontal. A análise comparativa apresentada foi relativa a alguns critérios conflitantes deste projeto. A utilização desta análise comparativa possibilitou definir-se um melhor compromisso entre: a precisão do modelo obtido, número de regras, sobreposição das funções de pertinência e a complexidade computacional do procedimento de desnebulização. Estes aspectos foram determinantes na escolha de um modelo de um sistema nebuloso adequado.

As redes neurais artificiais apresentaram resultados promissores, nos três estudos de caso, de identificação multivariável. Um estudo comparativo de métodos de treinamento heurísticos e de segunda ordem para a rede neural *perceptron* multicamadas foi apresentado. Observou-se que a eficiência de uma rede neural *perceptron* multicamadas depende dos parâmetros de projeto, usualmente determinados heurísticamente por procedimentos de tentativa e erro, tais como: o número de neurônios na camada oculta, o(s) coeficiente(s) de aprendizado, número de épocas do procedimento de treinamento. Os procedimentos de otimização de segunda ordem apresentaram desempenho superior aos procedimentos de treinamento heurísticos. Entretanto, a informação de estimativas da Hessiana, necessárias a alguns métodos de segunda ordem, apresenta um custo computacional adicional aos métodos heurísticos.

A rede neural função de base radial foi outra forma de projeto neural adotada. Deve ser enfatizado que a rede neural função de base radial apresentou resultados promissores, com aprendizado rápido e uma configuração de modelo local com estrutura caixa-cinza, ao contrário da rede neural *perceptron* multicamadas que configura um modelo global apresentando uma estrutura de modelo caixa-preta.

A rede neural parcialmente recorrente de Elman modificada e a rede neural *Group Method of Data Handling* apresentaram resultados apropriados em identificação multivariável. No entanto, limitações similares foram observadas nas redes neurais

perceptron multicamadas, pois o método de treinamento utilizado foi o algoritmo de retropropagação do erro (*backpropagation*). Outro tópico do projeto via redes neurais artificiais, este comum a outras técnicas de identificação, é relativa ao sobre-treinamento. Este tópico foi analisado e propõe-se uma alternativa de verificação da capacidade de generalização dos modelos neurais, no capítulo 8.

O sistema híbrido inteligente combinando sistemas nebulosos do tipo Takagi-Sugeno-Kang e redes neurais e sistema híbrido de rede neural função de base radial e estratégia evolutiva foram analisados em dois estudos de caso de identificação multivariável. O sistema nebuloso combinado à rede neural foi configurado para utilizar o treinamento das premissas, através do método do gradiente descendente, enquanto o aprendizado dos consequentes foi realizado pelo método dos mínimos quadrados recursivo. O sistema neural-evolutivo constitui-se de um algoritmo para aprimorar (sintonia fina) o desempenho obtido pela rede neural função de base radial com ajuste via algoritmo k -médias e pseudo-inversa, através de uma estratégia evolutiva com mutações correlacionadas. Os resultados apresentados mostraram que a integração entre os paradigmas inteligentes pode ser uma alternativa atraente de acordo com as necessidades e tratamento de complexidades que o projeto requer.

10.1.2 Controle de processos multivariáveis

O controle de processos multivariáveis foi realizado através das metodologias inteligentes — algoritmos evolutivos, sistemas nebulosos, redes neurais artificiais e sistemas híbridos inteligentes — além de estratégias de controle preditivas e controle de estrutura variável, tanto isoladas em configuração adaptativa, quanto combinadas as metodologias da inteligência computacional.

Os algoritmos de controle analisados foram aplicados em um manipulador robótico, utilizado no procedimento de identificação. Na análise do comportamento servo avaliou-se se o projeto e a sintonia de determinado controlador foi adequado para o seguimento das trajetórias das posições e velocidades. A robustez das propostas de projetos em controle foram verificadas através da análise do comportamento regulatório, com procedimentos de

testes abordando a influência da modificação de parâmetros. Diferentes valores de massa e de comprimentos dos elos foram configurados com a finalidade de testar os controladores.

Os algoritmos evolutivos foram utilizados na sintonia (otimização) dos controladores proporcional-derivativo e proporcional-integral-derivativo, em projetos descentralizados (*multiloop*) e em novas propostas de projetos centralizados (acoplados). Outras abordagens dos algoritmos evolutivos foram na configuração de sistemas híbridos inteligentes, combinados a redes neurais, sistemas nebulosos e controle preditivo com módulo de projeto evolutivo.

Algumas propostas de controle nebuloso do tipo proporcional-integral-derivativo foram comparadas. Os projetos de controle via redes neurais artificiais trataram o projeto de algumas configurações de controladores neurais multivariáveis, adotando-se os melhores modelos obtidos na identificação do manipulador robótico, que compõem o capítulo 8 desta tese.

Diversas propostas de controle preditivo generalizado foram apresentadas, sob duas óticas: a abordagem adaptativa e a abordagem inteligente. A abordagem adaptativa do controle preditivo generalizado, através do estimador dos mínimos quadrados recursivo, apresentou desempenho promissor tanto em comportamento servo quanto regulatório. O manipulador robótico com acoplamento entre as variáveis e o projeto de controle adaptativo apresentou desempenho apropriado, mesmo utilizando uma estratégia de controle preditivo baseada em modelos lineares.

O controle preditivo generalizado, na abordagem adaptativa, conforme mencionado no capítulo 6, proporcionou um controle estável para processos com pólos e zeros instáveis, ordem desconhecida do modelo e parâmetros variáveis. A aplicação do controle preditivo generalizado é apropriada também para processos de fase não-mínima e processos instáveis, em malha aberta.

O controle preditivo generalizado com a utilização de modelos da inteligência computacional também apresentou desempenho adequado e promissor. Entretanto, a utilização de algumas metodologias da inteligência computacional para a obtenção de modelos não-lineares, em projetos de controle preditivo, pode demandar uma complexidade alta em aplicações práticas, ou mesmo inadequação quando o processo é rápido e os recursos computacionais são limitados.

A implementação de controle de estrutura variável com modos quase-deslizantes apresentou resultados promissores, tanto em estrutura única, quanto quando combinado a estratégias de controle de variância mínima generalizada e inteligente. O controle por modos quase-deslizantes não requer precisão na identificação do processo, porém necessita de limites nos parâmetros do modelo. O problema no controle por modos quase-deslizantes que seja o de evitar o *chattering* causado pelo chaveamento da entrada, é resolvido pela introdução de entradas de controle, utilizando-se conceitos de camada limite.

10.2 Principais contribuições

Esta tese apresenta diversas formas de configuração de algoritmos avançados em identificação e controle de processos multivariáveis não-lineares. As principais contribuições desta tese foram relativas a:

- (i) análise dos fundamentos, potencialidades e diferenciação de diversos algoritmos evolutivos;
- (ii) propostas de configurações baseadas em concepções de aprendizado Lamarckiano, de algoritmos evolutivos combinados com *simulated annealing* e método simplex;
- (iii) nova abordagem de identificação multivariável através de algoritmos evolutivos visando a estimação de atraso de transporte e determinação da ordem dos polinômios;
- (iv) estudos comparativos de diversas configurações de algoritmos evolutivos, em problemas de identificação e sintonia de controladores;
- (v) configuração do sistema nebuloso com geração de regras, através de aprendizado modificado, para a identificação de processos, analisando aspectos relativos ao compromisso entre a precisão do modelo, o número de regras e a complexidade computacional da fase de desnebulização;
- (vi) estudo comparativo de métodos de treinamento heurísticos e de segunda ordem de redes neurais *perceptron* multicamadas;
- (vii) aplicação de redes neurais função de base radial, rede neural parcialmente recorrente de Elman modificada, rede neural *Group Method of Data Handling* e sistemas híbridos inteligentes em identificação multivariável não-linear;
- (viii) análise de sobre-treinamento baseado no conjunto de dados de validação e critério de desempenho através do coeficiente de correlação múltipla;
- (ix) estudo de abordagens evolutivas, nebulosas e neurais em identificação de processos multivariáveis não-lineares;
- (x) análise de mecanismos adaptativos para ajuste das probabilidades de cruzamento, mutação e operadores baseados em gradiente em algoritmos genéticos na sintonia de controladores *PD* e *PID* multivariáveis;
- (xi) configurações de projeto de controle preditivo generalizado na forma adaptativa e inteligente;

(xii) aplicação de controle de estrutura variável combinado a *GMV*, em formulação adaptativa.

10.3 Perspectivas

Os algoritmos avançados estão em crescimento emergente e são eficientes e robustos no tratamento de diversos problemas, na área de controle e em outras áreas. Esta tese mostrou as potencialidades de algoritmos avançados, no projeto de sistemas adaptativos, preditivos e inteligentes no tratamento de processos não-lineares.

Várias contribuições na área de controle inteligente e preditivo podem ser realizadas em estudos de identificação e controle de processos. A seguir são mencionados alguns tópicos de pesquisa a serem estendidos para o tratamento de processos multivariáveis, que apresentam características complexas.

Segundo uma pesquisa vinculada a aplicações de controle no meio industrial japonês (TAKATSU & ITOH, 1999), a situação atual e as futuras direções da tecnologia de controle, na indústria japonesa, mostram uma crescente utilização de inteligência computacional (principalmente os sistemas nebulosos e as redes neurais artificiais), de técnicas de otimização e de estratégias de controle preditivo. A pesquisa apresenta também estudos que enfatizam o grau de satisfação e as expectativas na aplicação destas metodologias no meio industrial.

10.3.1 Área de algoritmos evolutivos

A implementação de algoritmos evolutivos é uma área de intensa pesquisa. A perspectiva de contribuições, nesta área, tem um espectro amplo de abordagens evolutivas. Entre as abordagens a serem analisadas destaca-se a adoção de procedimentos de evolução Lamarckianos (WHITLEY *et al.*, 1994), efeito Baldwin (HINTON & NOWLAN, 1987), mecanismos evolutivos baseados na teoria dos memes (DAWKINS, 1977) e a configuração de metodologias combinando métodos baseados em gradiente híbridos com algoritmos evolutivos (DOZIER *et al.*, 1998).

Os métodos baseados em gradiente (*hill-climbing*) e os algoritmos evolutivos representam duas classes de técnicas de otimização que parecem ter propriedades diferenciadas. A combinação de algoritmos evolutivos híbridos com metodologias de *hill-climbing* supera, freqüentemente, o desempenho de uma metodologia tratada individualmente.

Os algoritmos evolutivos são conhecidos por serem eficientes para otimização global. Entretanto, os algoritmos evolutivos não são tão eficientes para executarem buscas locais e são propensos a convergir prematuramente, antes que a melhor solução tenha sido encontrada. Os algoritmos evolutivos podem ser lentos à sintonia fina. Assim, uma metodologia híbrida pode determinar mais rapidamente se o algoritmo evolutivo seguirá o caminho mais promissor. Se a avaliação da função do problema é ruidosa ou descontínua pode-se utilizar algoritmos evolutivos para aprimorar a capacidade de busca da solução. Uma população pequena é gerada em torno dos descendentes e após configura-se os operadores de mutação e cruzamento favorecendo mais as pequenas mudanças do que as grandes mudanças. Nesta configuração os algoritmos evolutivos para buscas locais comportam-se como algoritmos de *hill-climbing*.

A implementação de algoritmos evolutivos baseados em programação genética (MCKAY, 1996) é uma linha de pesquisa emergente para o tratamento de processos complexos, pois a programação genética permite a otimização não só paramétrica, mas também estrutural em problemas de identificação de processos.

10.3.2 Área de sistemas nebulosos

As questões de estabilidade são uma das mais importantes vertentes na análise e projeto de sistemas de controle nebulosos. A análise de estabilidade de controladores nebulosos é complexa porque estes sistemas são essencialmente não-lineares. Contudo, recentemente, alguns resultados de estabilidade baseada na teoria de estabilidade não-linear têm sido relatados, a citar, por exemplo, ORDÓÑEZ *et al.* (1996), CHENG & REES (1997), FISCHLE & SCHRÖDER (1999) e CUESTA *et al.* (1999).

As principais dificuldades no projeto de controladores para sistemas reais é o tratamento de não-linearidades e incertezas. Estes fatores motivam a abordagem robusta no

projeto de controladores nebulosos, com técnicas baseadas na teoria de controle H^∞ , utilização de algoritmos de programação convexa em abordagens via *linear matrix inequalities (LMI)* e estabilização quadrática (TANAKA *et al.*, 1996).

Outra abordagem interessante é a combinação da teoria do caos com a tecnologia dos sistemas nebulosos (KATAYAMA *et al.*, 1993) para a identificação de processos complexos e ruidosos, além da previsão das saídas futuras e controle de processos com este tipo de comportamento.

10.3.3 Área de redes neurais

Um das questões importantes em estudos de redes neurais artificiais em controle de processos é a análise de estabilidade. Algumas propostas têm sido apresentadas na literatura, mas muitos estudos são realizados quanto a critérios de estabilidade. Uma área a ser explorada é a análise de redes neurais artificiais quanto à possibilidade de aplicações da teoria de Lyapunov.

SANNER & SLOTINE (1992) propõem uma arquitetura de controle adaptativo empregando rede neural função de base radial para compensar não-linearidades e realizam um estudo de estabilidade do mecanismo de ajuste de pesos baseados na teoria de Lyapunov. Outros estudos e análises, quanto aos aspectos de estabilidade de controladores neurais, são apresentados em SUYKENS *et al.* (1995) e LEWIS & PARISINI (1998).

Quanto ao projeto de redes neurais *perceptron* multicamadas em problemas de identificação, possíveis trabalhos a serem realizados em continuação às abordagens heurísticas e de segunda ordem analisadas nesta tese, são relativas à análise de complexidade computacional de cada técnica de treinamento. O desempenho do algoritmo de treinamento da retropropagação do erro pode ser aprimorado pela utilização de algoritmos punitivos (REED, 1993) e pela adição de um termo de regularização evitando problemas devidos ao sobre-treinamento (SJÖBERG, 1995).

A configuração de rede neural *Group Method of Data Handling* visando um melhor compromisso entre a adequação da precisão, capacidade de generalização e princípio de parcimônia é outro tópico de pesquisa a ser explorado em identificação de processos.

Quanto as redes neurais artificiais para mapeamentos locais utilizando funções de base, este é um tópico emergente e de intensas pesquisas. O aprimoramento de algoritmos para aplicação de redes neurais função de base radial com treinamento via algoritmos do tipo mínimos quadrados recursivo em projetos de controle adaptativo, é uma abordagem promissora para o tratamento de processos não-lineares e variantes no tempo. Outro aspecto é o desenvolvimento de procedimentos de ortogonalização (ZHU & BILLINGS, 1997) para o tratamento de matrizes de pesos “mal condicionadas”.

10.3.4 Área de sistemas híbridos inteligentes

A configuração de sistemas híbridos inteligentes é motivada pelas limitações de algoritmos inteligentes quando projetados para operarem de forma isolada. Assim, duas (ou mais) técnicas são combinadas tentando superar as limitações das técnicas tratadas individualmente.

Quanto as perspectivas de estudos, neste tópico de pesquisa, têm-se a análise e estudo comparativo de configurações combinando algoritmos evolutivos, sistemas nebulosos, redes neurais artificiais e as metodologias de identificação não-linear e de controle adaptativo. Neste aspecto, visa-se a obtenção de sistemas híbridos inteligentes eficientes e estáveis que apresentem resultados satisfatórios e complexidade computacional apropriada ao tratamento de processos reais com características complexas que justifiquem sua aplicação.

10.3.5 Área de controle preditivo

A identificação de sistemas é um exercício que envolve múltiplos e conflitantes objetivos, tipicamente a complexidade do modelo, os critérios de desempenho e a validação que influenciam a seleção das estruturas de modelo matemático mais adequadas. A noção de um “bom modelo” é subjetiva e depende da preferência pessoal. Assim a heurística de tentativa e erro é uma regra usual em identificação de processos. A abordagem de múltiplos modelos é uma ferramenta poderosa no tratamento de problemas de

incertezas (ou mudanças) estruturais e paramétricas e para decompor um problema complexo em subproblemas simples (COELHO & COELHO, 1999f).

A estratégia de identificação de múltiplos modelos baseia-se na idéia de utilizar um conjunto de modelos para descrever o processo. Quando da aplicação de algoritmos preditivos continuamente seleciona-se o modelo com melhor aproximação. Portanto, duas questões essenciais para este tipo de identificação são: a escolha do conjunto de modelos matemáticos para descrever o processo e o esquema de chaveamento entre os modelos (NARENDRA & BALAKRISHNAN; 1994). Um estudo preliminar de controle preditivo baseado na aplicação de algoritmos genéticos em projeto com configuração de múltiplos modelos foi apresentado em COELHO *et al.* (1999a). Neste sentido a configuração de controladores preditivos baseados em múltiplos modelos (CHOW *et al.*, 1995), identificação com múltiplos objetivos (COELHO & COELHO, 1999f) e algoritmos inteligentes são tópicos pouco explorados e apresentam potencialidades para o tratamento de processos complexos.

O projeto de controle preditivo multivariável não-linear baseado em estimativas de modelos polinomiais *NARMAX*, Wiener-Hammerstein, Volterra e bilineares (MARCHI *et al.*, 1999) são abordagens para as quais podem ser realizadas contribuições: na configuração de algoritmos de otimização eficientes e adequados ao tratamento de restrições (GOSSNER *et al.*, 1998); detecção de atraso de transporte; e análise de robustez e estabilidade de controladores (AYMES & GUIBÉ, 1996) em aplicações multivariáveis.

Os aspectos de robustez dos controladores preditivos frente as incertezas paramétricas e estruturais no contexto *MIMO* deve ser enfatizado. Neste caso, alguns estudos têm sido apresentados por pesquisadores na área, por exemplo, em abordagem da parametrização de Youla no projeto de *GPC* monovariável.

Outras abordagens interessantes são o tratamento de acoplamentos entre as variáveis pelo levantamento da matriz *interactor* em projetos de controle preditivo multivariável (VAN DOREN, 1996, 1997), tratando processos lineares (SHAH *et al.*, 1987) ou não-lineares (MUTOH & ORTEGA, 1993; MUTOH, 1998).

10.3.6 Área de controle de estrutura variável

A teoria de controle de estrutura variável é uma metodologia atraente para o controle de processos sujeitos a perturbações externas e variações internas. A simplicidade e a robustez do controle de estrutura variável motivou o desenvolvimento de diversas pesquisas relativas a abordagens contínuas e ao desenvolvimento da teoria de controle por modos quase-deslizantes para a realização de projetos para aplicações de controle digital.

As possibilidades de prova de estabilidade da filosofia de projeto de controle de estrutura variável, através da teoria de Lyapunov, motivam a implementação de controladores de estrutura variável combinados a sistemas nebulosos (ALMEIDA *et al.*, 2000; LEE *et al.*, 1998) e redes neurais artificiais (FANG *et al.*, 1999).

As perspectivas de pesquisas nesta área são relativas às diversas possibilidades de combinações com algoritmos de controle preditivo e adaptativo (CHAN, 1997, CORRADINI & ORLANDO, 1997), visando o tratamento de processos não-lineares que apresentam incertezas paramétricas.

Apêndice A

ESTADO DA ARTE EM PROJETO INTELIGENTE E PREDITIVO

A seguir são apresentadas as tabelas que resumem o estado da arte quanto à utilização de metodologias de controle preditivo e inteligência computacional.

Tabela A.1. Aplicações dos *AEs* em ambientes industrial e comercial.

<i>aplicação</i>	<i>empresa</i>
projeto de bulbos de lâmpadas e turbinas de avião	General Electric
ambiente de simulação de aviões	Beranek & Newmann
propulsão de navios	Naval Surface Weapons Center
roteamento de redes de comunicação	CNET, Cap Volmac e Hewlett Packard Laboratories
escalonamento e controle	Rolls-Roice
gerenciamento	First Quadrant
aplicações comerciais	The Prediction Company
consumo de combustível de vários fornos	Intelligent Computer Systems Centre (University of the West, England) e Courtaulds Films
suprimento de energia e gerenciamento de potência	Siemens AG
controle e tarefas de otimização em aplicações aeroespaciais	Daimler-Benz Aerospace
projeto de redes de telecomunicações	Nortel Smart Network Technology Labs
gerenciamento de bancos de dados distribuídos	BT Labs
projeto de redes de fibras óticas	U S West

Tabela A.2. *Hardware e software com a utilização de AEs.*

<i>produto</i>	<i>empresa</i>	<i>descrição</i>
OMEGA	Cap Gemini e KiQ Ltd	ambiente para o desenvolvimento de modelos para previsão
Genetica, EvoSchool	Genetica Advanced Software Architecture Srl	software para aplicações de escalonamento
Evolver	Palisade Corporation	software para problemas de otimização

Tabela A.3. *Aplicações dos SNs na indústria.*

<i>aplicação</i>	<i>empresa</i>
controle de elevadores	Fujitec, Toshiba, Mitsubishi, Hitachi
auto-foco de câmera de vídeo	Sanyo & Fisher, Canon, Panasonic, Matsushita
máquina de lavar louças, aquecedor de água	Matsushita
diagnóstico, controle de motores	Rockwell, Inform GmbH, Texas
tratamento e purificação de água	Fuji Electric, American Ref-Fuel Co.
produção de celulose	Siemens
refinaria petrolífera	Seibu Oil Co.
controle de flutuações de temperatura em sistemas de ar condicionado	Mitsubishi, Sharp
<i>palmtop</i>	Sony
ajuste imagem de televisão	Sony, Sanyo
subsistemas automotivos	Subaru, Honda, Nissan, Lexus, Mitsubishi, Daihatsu, Nippon denso, Mazda, GM (Saturn)
máquina de lavar roupas	Panasonic, Sanyo, Matsushita, Toshiba, Hitachi, AEG, Goldstar, Sharp
motores, metrô e tráfego urbano	Hitachi
barbeador	Sanyo, Panasonic
aeronaves	Rockwell Corp.
fornos de microondas	Sharp, Sanyo, Matsushita, Hitachi, Toshiba, Mitsubishi
refrigerador	Sharp, Sanyo, Whirlpool
veículos auto-guiados, robôs móveis	NASA, IBM
área militar e aeroespacial	U.S. Naval Air Warfare Center, NASA
ventilador elétrico, televisão	Sanyo
otimização de aplicações de controle industrial	Apronix, Omron, Meiden, Sha, Micom, Mitsubishi, Nisshin-Denki, Oku-Electronics

(cont.)

<i>aplicação</i>	<i>empresa</i>
projeto de <i>software</i> para aplicações industriais	Aptronix, Harima, Ishikawajima-OC Engineering
controle de limpadores à vácuo	Matsushita
controle do consumo de combustível em automóveis	NOK, Nippon Denki Tools
posicionamento de pastilhas na produção de semicondutores	Canon
carregador de baterias	Bosch
câmeras de vídeo	Canon, Minolta
controle de robôs	Hirota, Fuji Electric, Toshiba, Omron
processos químicos	Asahi Chemical Industry
limpador a vácuo	Philips, Siemens
controle de temperatura em nave espacial	NASA
controle climático	Ford
sistemas de aprimoramento de segurança de reatores nucleares	Hitachi, Bernard, Nuclear Fuel div
controle de temperatura e velocidade na produção de aço	Kawasaki Steel, New-Nippon Steel, NKK
<i>camcorders</i>	Canon, Sanyo, JVC
controle automático em aplicações de geração de energia elétrica	Tokio Electric Pow.

Tabela A.4. *Hardware e software com a utilização de SNs**.

<i>produto</i>	<i>empresa</i>	<i>descrição</i>
NLX 230	American Neuralogix	microcontrolador
NLX22X	NeuroLogix	microcontrolador
FC110, FCA10VME	Togai Infralogic	microcontrolador
FP3000, FP5000	Apt Instruments/Omron	microcontrolador
WARP	SGS-Thomson	microcontrolador
CI de inferência polivalente	OKI Electric Industry	microcontrolador
UT 750/UP 750	Yokogawa	microcontrolador
H8300, H8500	Hitachi American	microcontrolador
TMS320C14	Texas	microcontrolador
A-B Flex	Allen-Bradley	<i>software</i>
UDC 2000, UDC 3000, UDC 6300	Honeywell	sintonia de controladores
Fuzzics, controlador SDC40A	Yamatate-Honeywell	módulo de aplicação
Fuzz-C Preprocessor	Byte Craft	geração de código

(cont.)

<i>produto</i>	<i>empresa</i>	<i>descrição</i>
DataEngine	Management of Intelligent Technologies (MIT) GmbH	software para análise de dados
Fuzzy Logic Designer	Byte Dynamics	ferramenta gráfica para desenvolvimento de código ANSI C
O'INCA Design Framework	Intelligent Machines	projeto e implementações
CubiCalc	HyperLogic	ferramenta de desenvolvimento
QuickFuzz Development Station	MentaLogic Systems	aplicações de CN
MATLAB (Fuzzy Logic Toolbox)	The MathWorks	toolbox
TILShell	Togai Infralogic	ferramenta CASE
KBVision	Amerinex	processamento de imagens
FIDE	Aptronix	ferramenta de desenvolvimento
Fuzzy Knowledge Builder, Fuzzy Thought Amplifier, Fuzzy Decision Maker	Fuzzy Systems Engineering	ferramentas de desenvolvimento
SDAF	HI Tecnologia	projeto e implementação de CNs
FuziCalc	FuziWare	tomada de decisões
Fuzzytech Standard Edition, Fuzzytech Explorer EWdition, Fuzzytech MCU Editors	Inform GmbH	geração e otimização de SNs
RT/Fuzzy Module	Integrated Systems	simulação e geração de código
FCM (Fuzzy Control Manager)	TransferTech GmbH	software
Metus	Metus System Group	desenvolvimento e simulação
Fuzzle	Modico	shell
Fuzzy Logic Development Environment (FLDE)	Syndesis Ltd.	tool that will allow you to develop embedded applications in C.
NeoFuz	National Semiconductor	utilização para projeto em microcontroladores
FS-Fuzzysoft	GTS Trautz GmbH	ferramenta de desenvolvimento
Fuzzy Logic Embedded Controller Development Packages	Hitchi America	ferramentas de desenvolvimento e geração de código
Hi-Flex	Hiware AG	ferramentas de desenvolvimento e geração de código

(cont.)

<i>produto</i>	<i>empresa</i>	<i>descrição</i>
Fuzzytech MCU-96 Edition, Fuzzy builder	Intel Corp.	ferramentas de desenvolvimento e geração de código
RT/Fuzzy Module	Integrated Systems	ferramentas de desenvolvimento e geração de código Ada ou C
M68HXBFIDS, FLESKTK	Motorola Inc.	kit educacional e geração de código
NewFuz4	National Semiconductor Corp.	ferramentas de desenvolvimento e geração de código Ada ou C
Insight	NeuralLogix	ferramentas de desenvolvimento
FuzzySP, FuzzyEmbed	NeuralWare Inc.	ferramentas de desenvolvimento
Fuzzytech	SGS-Thomson Microelectronics	ferramentas para aplicações em microcontroladores
Fuzzytech	Siemens Semiconductor Group, Siemens Corp	ferramentas para aplicações em microcontroladores
Fuzzytech MCU-320	Texas Instruments Inc.	ferramentas de desenvolvimento e geração de código
TILShell + fuzzy C	Togai Infralogic	sistema de desenvolvimento

* outras empresas também têm produtos e/ou aplicações de *SNs*, tais como: Klöckner-Moeller, Foxboro, Robersshaw, Fisher-Rosemount, Exsys.

Tabela A.5. Aplicações das *RNs* na indústria.

<i>aplicação</i>	<i>empresa</i>
controle de fornos de aço	Neural Application Corp.
aplicações automotivas	Ford, General Motors
controle de processo de manufatura de semicondutores	Kopin Corp.
diagnóstico de motores elétricos	Siemens
área militar e aeroespacial	U.S. Naval Air Warfare Center, NASA
controle de processos químicos	Pavilion Tech., Eastman Kodak
produção de aço	Fujitsu, Nippon Stell
controle de limpador a vácuo	Hitachi
controle da qualidade de cópias em papel	Ricoh Corp.
previsão e otimização aplicadas a colunas de destilação	Gensym
controle de impressão de tinta litográfica	Rockwell

Tabela A.6. *Software* com a utilização de *RNs**.

<i>produto</i>	<i>empresa</i>	<i>descrição</i>
ANSim	SAIC	<i>software</i> de simulação
ANSim/ANSpec	SAIC	ambientes de desenvolvimento
Axon/UISL	HNC	ambientes de desenvolvimento
BrainMaker	California Scientific Software Co.	<i>RN</i> com treinamento <i>BP</i>
Cognitron	Cognitive Software	desenvolvimento e projeto
DataEngine	Management of Intelligent Technologies GmbH	<i>software</i> para análise de dados
DESIRE/NEUNET	Granino Korn	<i>software</i>
G2 acoplado ao NeuroOn-Line Studio	Gensym	<i>software</i> para otimização, identificação e previsão
MacBrain	Neurix	desenvolvimento e projeto
Matlab: Neural Network Toolbox	The MathWorks	<i>toolbox</i>
Mimence	Mimetics	desenvolvimento e projeto
Nestor NDS	Nesotr	desenvolvimento e projeto
NeuCOP Optimizer	AspenTech	otimização/identificação em controle de processos
Neural Bench	Neural Bench Develop.	projeto e desenvolvimento
Neural Networks (Statistica)	StatSoft, Inc.	<i>software</i>
NeuralWorks Explorer	AspenTech	<i>software</i> (diversas <i>RNs</i>)
NeuralWorks Professional/Explorer	NeuralWare	ambientes de desenvolvimento
Neuro-Shell	Ward Systems Group	<i>software</i> (várias <i>RNs</i>)
NeuroCoM (Neuro Control Manager)	TransferTech GmbH	desenvolvimento e projeto
NeuroLab	Mikuni Berkeley R & D Corp.	<i>software</i> (várias <i>RNs</i>)
NeuroLution simulation and development system	GmbH Advanced Neurocomputing	projeto, análise e aplicação
NeuroSoft	HNC	<i>software</i>
NeuroSolutions	NeuroDimension Incorp.	<i>software</i>
NUERAL DESK	Neural Computer Sciences	desenvolvimento e projeto
OWL Neural Network	HyperLogic Corp.	<i>software</i>
Plexi	Lucid	<i>software</i> (várias <i>RNs</i>)
Process Perfector	Pavilion Technologies	otimização e identificação em controle de processos
Property Predictor	Pavilion	controle de processos
SN2.8	Neuristique	<i>software</i> (várias <i>RNs</i>)

Tabela A.7: Áreas de classificação de aplicações da combinação de *SNs* e *AGs* (CORDÓN *et al.*, 1995).

	área	referências
1	<i>AGs</i> nebulosos	24
2	<i>clustering</i> nebuloso	14
3	otimização nebulosa	39
4	<i>RNs</i> nebulosas	34
5	equações relacionais nebulosas	6
6	sistemas especialistas nebulosos	8
7	sistemas classificadores nebulosos	33
8	recuperação de informação e busca em base de dados	6
9	tomada de decisão nebulosa, modelos econômicos e financeiros	10
10	análise de regressão nebulosa	6
11	reconhecimento de padrões nebulosos e processamento de imagens	20
12	classificação nebulosa e aspectos de aprendizado	24
13	controladores nebulosos (projeto, aprendizado, sintonia, aplicações)	287
14	estrutura de <i>SN</i> com <i>AG</i>	13
15	diversas configurações de <i>SNs</i>	38
	total de referências combinando <i>SNs</i> e <i>AGs</i> (até 1995)	562

Tabela A.8. Resumo de aplicações comerciais de *MBPC* utilizando modelos lineares.

área	<i>DMC Corp.</i>	<i>Setpoint Inc.</i>	<i>Honeywell Profimatics</i>	<i>Adersa</i>	<i>Treiber Controls</i>	total
refinaria	360	320	290	280	250	1500
petroquímica	210	40	40	-	-	290
química	10	20	10	3	150	193
papel	10	-	30	-	5	45
gás	-	-	5	-	-	5
utilidade	-	-	2	-	-	2
separação de ar	-	-	-	-	5	5
mineração/metalurgia	-	2	-	7	6	15
processamento de alimentos	-	-	-	41	-	41
fornos	-	-	-	42	-	42
automotiva	-	-	-	13	-	13
aeroespacial/defesa	-	-	-	7	-	7
outras	10	20	-	45	-	75
total	600	402	377	438	416	2233

Tabela A.9. Resumo de aplicações comerciais de MBPC utilizando modelos não-lineares.

área	<i>Adersa</i>	<i>Aspen Tech.</i>	<i>Continental Controls</i>	<i>DOT Products</i>	<i>Pavilion Tech.</i>	total
ar e gás	-	-	18	-	-	18
química	-	-	15	-	5	20
processamento de alimentos	-	-	-	-	9	9
polímeros	-	-	-	5	15	20
papel	-	-	-	-	1	1
refinaria	-	-	-	-	13	13
utilidades	-	1	2	-	-	3
outras	1	-	1	45	-	2
total	1	1	36	5	43	86

REFERÊNCIAS BIBLIOGRÁFICAS

- AGARWAL, M.; 1997. A Systematic Classification of Neural-Network-Based Control. *IEEE Control Systems*, Special Issue on Intelligent Control, v. 17, n. 2, p. 75-93.
- AGARWAL, M., SEBORG, D. E.; 1987. A Self-Tuning Controller for MIMO Non-Linear Systems. *International Journal of Adaptive Control and Signal Processing*, v. 1, pp. 129-142.
- AGUIRRE, L. A.; JÁCOME, C. R. F.; 1998. Cluster Analysis of NARMAX models for Signal-Dependent Systems. *IEE Proceedings Control Theory and Applications*, v. 145, n. 4, p. 409-414.
- ALANDER, J. T.; 1995. *An Indexed Bibliography of Genetic Algorithms in Control*. Vaasa. Report 94-1, Department of Information Technology and Production Economics, University of Vaasa, Finland.
- _____; 1996. *Indexed Bibliography of Genetic Algorithms Papers of 1996*. Vaasa. Report 94-96-1, Department of Information Technology and Production Economics, University of Vaasa, Finland.
- AL-DUWAISH, H. N.; AL-HAMOUZ, Z. M.; 1998. A Genetic Approach to the Selection of the Variable Structure Controller Feedback Gains. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Trieste, Itália). *Proceedings*. p. 227-231.
- ALMEIDA, O. M.; COELHO, L. S.; COELHO, A. A. R.; 2000. Practical Robust Control to a Nonlinear System Using Auto-Tuning Fuzzy and Sliding-Mode Approaches. In: SUZUKI, Y.; ROY, R.; OVASKA, S. J.; FURUHASHI, T.; DOTE, Y. (eds.). *Soft Computing in Industrial Applications*. Springer-Verlag : London, 2000.
- ALVAREZ, T., TADEO, F., PRADA, C.; 1998. Constrained Predictive Control of Multivariable Robotic Systems. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Trieste, Itália). *Proceedings*. p. 588-592.
- AMARAL, W. C.; 1980. *Identificação Recursiva de Sistemas Multivariáveis*. Campinas, SP. Tese (Doutorado em Engenharia Elétrica), DCA/FEE, Universidade de Campinas, UNICAMP.
- AMARAL, W. C.; COELHO, L. S.; COELHO, A. A. R.; 1998. Equivalencia entre dos Controladores GMV y Estabilización de Procesos MIMO con Atraso de Transporte Distintos. *Revista Internacional Información Tecnológica*, Chile, v. 9, n. 6, p. 147-152.
- AMARI, S.; MURATA, N.; MÜLLER, K.-R.; FINKE, M.; YANG, H. H.; 1997. Asymptotic Statistical Theory of Overtraining and Cross-Validation. *IEEE Transactions on Neural Networks*, v. 8, n. 5, p. 985-996.

- ANDERSEN, H. C. A.; 1998. *The Controller Output Error Method*. Queensland, Australia. PhD thesis, Department of Computer Science and Electrical Engineering, University of Queensland.
- ANGELINE, P. J. Adaptive and Self-Adaptive Evolutionary Computations, In: PALNISWAMI, M.; ATTIKIIOUZEL, Y.; MARKS, R.; FOGEL, D.; FUKUDA, T. (eds.). *Computational Intelligence: A Dynamic Systems Perspective*, Piscataway, NJ, USA : IEEE Press, 1995. p. 152-163.
- ANGUITA, D.; PAMPOLINI, M.; PARODI, G.; ZUNINO, R.; 1993. YPROP: Yet Another Accelerating Technique for the Back Propagation. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS (Amsterdam, Netherlands). *Proceedings*. Amsterdam, Netherlands. p. 500.
- ÅSTRÖM, K. J.; 1991. Intelligent Control. In: EUROPEAN CONTROL CONFERENCE (Grenoble, France). *Proceedings*. p. 2328-2339.
- _____; 1996. Adaptive Control Around 1960. *IEEE Control Systems*, v. 16, n. 3, p. 44-49.
- AYMES, J. M.; GUIBÉ, J. B-; 1996. Advanced Predictive Control for Aircraft Guidance. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 249-254.
- BÄCK, T.; HOFFMEISTER, F. Adaptive Search by Evolutionary Algorithms. In: EBELING, W.; PESCHEL, M.; WEIDLICH, W. (eds.). *Models of Selforganization in Complex Systems*, v. 64, Mathematical Research. Berlin, Germany : Akademie-Verlag, 1991. p. 156-163.
- BÄCK, T.; SCHWEFEL, H. -P.; 1993. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, v. 1, n. 1, p. 1-23.
- BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z.; 1997a. *Handbook of Evolutionary Computation*. Bristol, Philadelphia : Institute of Physics Publishing. New York, Oxford : Oxford University Press.
- BÄCK, T.; HAMMEL, U.; SCHWEFEL, H.-P.; 1997b. Evolutionary Computation: Comments on the History and Current State. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 3-17.
- BALAKRISHNAN, K; HONAVAR, V.; 1995. *Evolutionary Design of Neural Architectures — A Preliminary Taxonomy and Guide to Literature*. Ames. Technical Report CS TR 9501, Department of Computer Science, Iowa State University, Ames, IA, USA.
- _____; 1996. Some Experiments in Evolutionary Synthesis of Robotic Neurocontrollers. In: WORLD CONGRESS ON NEURAL NETWORKS (San Diego, CA). *Proceedings*. San Diego, CA, USA. p. 1035-1040.

- BARRETO, J. M.; 1996. *Conexionismo e a Resolução de Problemas*. Florianópolis. Tese apresentada no Concurso para Professor Titular, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina.
- _____; 1997. Introdução às Redes Neurais Artificiais. In: V ESCOLA REGIONAL DE INFORMÁTICA DA SBC REGIONAL SUL. *Anais*. Florianópolis, SC, Maringá, PR, Santa Maria, RS. p. 41-71.
- BARTOLINI, G.; FERRARA, A.; 1996. Multi-Input Sliding Mode Control of a Class of Uncertain Nonlinear Systems. *IEEE Transactions on Automatic Control*, v. 41, n. 11, p. 1662-1666.
- BARTOS, F. J.; 1996. Fuzzy Logic Reaches Adulthood. *Control Engineering International*, July, p. 50-56.
- _____; 1997. Artificial Intelligence: Smart Thinking for Complex Control. *Control Engineering International*, v. 44, n. 10, p. 44-52.
- BEHERA, L; GOPAL, M.; CHAUDHURY, S.; 1996. On Adaptive Trajectory Tracking of a Robot Manipulator Using Inversion of Its Neural Emulator. *IEEE Transactions on Neural Networks*, v. 7, n. 6, p. 1401-1414.
- BELEW, R.; MCINERNEY, J.; SCHRAUDOLPH, N. N.; 1990. *Evolving Networks: Using the Genetic Algorithm with Connectionist Learning*. La Jolla, CA, USA. CSE Technical Report CS90-174, Computer Science & Engr. Department, University of California at San Diego.
- BERENJI, H. R.; 1992. A Reinforcement Learning-Based Architecture for Fuzzy Logic Control. *International Journal on Approximate Reasoning*, n. 6, p. 267-292.
- BERSINI, H.; BONTEMPI, G.; DECAESTECKER, C.; 1995. Comparing RBF and Fuzzy Inference Systems on Theoretical and Practical Basis. in: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETWORKS (Paris). *Proceedings*. v. 1, p. 169-174.
- BEYER, H. G.; 1995. Toward a Theory of Evolution Strategies: Self-adaptation. *Evolutionary Computation*, v. 3, n. 3, p. 311-348.
- BILLINGS, S. A.; AGUIRRE, A.; 1995. Effects of the sample time on the dynamics and identification of nonlinear models. *International Journal of Bifurcation and Chaos*, v. 5, n. 6, p. 1541-1556.
- BILLINGS, S. A.; CHEN, S. Neural Networks and System Identification, In: IRWIN, G. W.; WARWICK, K.; HUNT, K. J. (eds.). *Neural Network Applications in Control*, IEE Control Engineering Series 53, The Institution of Electrical Engineers, 1995. Chapter 11, p. 229-251.

- BILLINGS, S. A.; CHEN, S.; KORENBERG, M. J.; 1989. Identification of MIMO Non-linear Using a Forward-Regression Orthogonal Estimator. *International Journal of Control*, v. 49, n. 6, p. 2157-2189.
- BILLINGS, S. A.; TAO, Q. H.; 1991. Model Validation Tests for Nonlinear Signal Processing Applications. *International Journal of Control*, v. 54, n. 1, p. 157-194.
- BILLINGS, S. A.; VOON, W. S. F.; 1986. Correlation Based Model Validity Tests for Non-Linear Models. *International Journal of Control*, v. 44, n. 1, p. 235-244.
- BITMEAD, R. R.; GEVERS, M.; WERTZ, W.; 1990. *Adaptive Optimal Control: The Thinking Man's GPC*. Series in Systems and Control Engineering. Englewood Cliffs, NJ, USA : Prentice-Hall.
- BLICKLE, T.; THIELE, L.; 1996. *A Comparison of Selection Schemes Used in Genetic Algorithms*. Zurich, Switzerland. TIK-Report, n. 11, version 2, Computer Engineering and Communication Network Lab, Swiss Federal Institute of Technology.
- BOL, E. W.; SANTANDER, A. C. A.; COELHO, L. S.; ALMEIDA, O. M.; COELHO, A. A. R.; 1999a. A Contribution to Digital Control Education on the WinFACT. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION "TECHNOLOGY-BASED EDUCATIONS AND GLOBALIZATION" (Rio de Janeiro, RJ).
- _____; 1999b. Software WinFACT em Conexão com o Ensino Prático de Controle de Processos. In: XXVII CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA, (Natal, RN). *Anais*. p. 1251-1258.
- _____; 1999c. WinFACT — A Tool for Teaching Digital Control. In: IEEE CONFERENCE ON DECISION AND CONTROL (Phoenix, AZ, USA).
- BONAVITA, N.; TOMASI, R.; 1998. Improvements in Process Control through Model-Based Techniques: a Control System Vendor's Perspective. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Trieste, Itália). *Proceedings*. p. 298-303.
- BONISSONE, P. P.; 1994. Fuzzy Logic Controllers: An Industrial Reality. In: ZURADA, J.M.; MARKS II, R. J.; ROBINSION, C. J. (eds.). *Computational Intelligence: Imitating Life*, Piscataway, NJ, USA : IEEE Press. p. 316-327.
- BRAGA, A. P.; CARVALHO, A. P. L.; LUDERMIR, T. B.; 1998. *Fundamentos de Redes Neurais Artificiais*. 11ª Escola de Computação da Sociedade Brasileira de Computação. Rio de Janeiro, RJ : DCC/IM, COPPE/Sistemas, NCE/UFRJ.
- BRANCO, P. J. C.; DENTE, J. A.; 1997. Fuzzy Systems Modelling in Practice. In: 7TH INTERNATIONAL FUZZY SYSTEMS ASSOCIATION WORLD CONGRESS, IFSA'97 (Prague, Czech Republic), *Proceedings*. v. II, p. 360-365.

- BUSHNELL, L. G.; 1996. On the History of Control. *IEEE Control Systems*, v. 16, n. 3, p. 14-16.
- BUCKLEY, J. J.; HAYASHI, Y.; 1993a. Fuzzy Input-Output Controllers are Universal Approximators. *Fuzzy Sets and Systems*, v. 58, p. 273-278.
- CAMACHO, E. F.; 1998. Model Predictive Control: Constraints, Robustness and Predictions. In: INTERNATIONAL WORKSHOP ON CONSTRAINED CONTROL SYSTEMS (Florianópolis, SC). NECCOSYDE: Alfa Network for Constrained Control System Development.
- CAMACHO, E. F.; BORDONS, C.; 1995. *Model Predictive Control in the Process Industry. Advances in Industrial Control*. London : Springer-Verlag.
- CARBAJAL, S. G.; RODRÍGUEZ, J. O.; RAMOS, L. S.; 1998. Fuzzy Control Applied to a Gas Transport Network in a Siderurgical Environment. In: 7TH INTERNATIONAL CONFERENCE INFORMATION PROCESSING AND MANAGEMENT OF UNCERTAINTY IN KNOWLEDGE-BASED SYSTEMS, IPMU (Paris, France). *Proceedings*. v. 1, p. 483-488.
- CLARKE, D. W.; GAWTHROP, P. J.; 1975. Self-Tuning Controller. *IEE Proceedings*, v. 112, n. 9, p. 929-934.
- CHAN, C. Y.; 1997. Discrete Adaptive Sliding Mode Tracking Controller. *Automatica*, v. 33, n. 5, p. 999-1002.
- CHELLAPILLA, K.; 1998. Combining Mutation Operators in Evolutionary Programming. *IEEE Transactions on Evolutionary Computation*, v. 2, n. 3, p. 91-96.
- CHEN, F. C.; LIU, C. C.; 1993. Adaptive Control of Nonlinear Continuous-Time Systems Using Neural Networks - General Relative Degree and MIMO Cases. In: 12TH IFAC WORLD CONGRESS (Sydney, Australia). *Proceedings*. v. 10, p. 373-376.
- CHEN, S.; BILLINGS, S. A.; LUO, W.; 1989. Orthogonal Least Squares Methods and Their Application to Non-Linear System Identification. *International Journal of Control*, v. 50, n. 5, p. 1873-1896.
- CHEN, S.; BILLINGS, S. A.; COWAN, C. F.; GRANT, P. M.; 1990. Practical Identification of NARMAX Models Using Radial Basis Functions. *International Journal of Control*, v. 52, n. 6, p. 1327-1350.
- CHEN, S.; COWAN, C. F. N.; GRANT, P. M.; 1991. Orthogonal Least Squares Learning for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, v. 2, n. 2, p. 302-309.
- CHEN, S.; BILLINGS, S. A.; GRANT, P. M.; 1992. Recursive Hybrid Algorithm for Non-linear System Identification using Radial Basis Function Networks. *International Journal of Control*, v. 55, n. 5, p. 1051-1070.

- CHENG, C. M.; REES, N. W.; 1997. Stability Analysis of Fuzzy Multivariable Systems: Vector Lyapunov Function Approach. *IEE Proceedings Control Theory and Applications*, v. 144, n. 5, p. 403-412.
- CHIPPERFIELD, A.; FLEMING, P.; 1996. Evolutionary Algorithms for Control Engineering. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 181-186.
- CHIU, S.; CHENG, J. J.; SITTER, C.; FOOKS, E.; 1995. Perspectives on the Industrial Application of Intelligent Control. In: 34TH CONFERENCE ON DECISION & CONTROL (New Orleans, LA, USA). *Proceedings*. p. 757-761.
- CHOW, C. -M.; KUZNETSOV, A. G.; CLARKE, D. W.; 1995. Using Multiple Models in Predictive Control. In: 3RD EUROPEAN CONTROL CONFERENCE (Rome, Italy). *Proceedings*. p. 1732-1737.
- COELHO, A. A. R.; 1991. *Controle Adaptativo para Processos Multivariáveis: Aspectos Teóricos e Simulação*. Campinas, SP. Tese (Doutorado em Engenharia Elétrica), DCA/FEE, Universidade de Campinas, UNICAMP.
- COELHO, A. A. R.; AMARAL, W. C.; 1993. Multivariable Generalized Predictive Controller: Analysis, Tuning and Application. In: 12TH IFAC WORLD CONGRESS (Sydney, Australia). *Proceedings*. v. 2, p. 331-334.
- COELHO, A. A. R.; SIMAS, H.; ARAÚJO, F. O.; BOUZON, G.; NARDELLI, M.; SOUZA, P. F. P.; PELANDRE, G. L.; 1998. Laboratório de Controle de Processos. In: XII CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA (São Paulo, SP). *Anais. CD-ROM* (artigo completo). Resumos. p. 179.
- COELHO, L. S.; ALMEIDA, O. M.; SIMAS, H.; COELHO, A. A. R.; 1998a. Intelligent and Self-Tuning PID Controllers: Methods and Application. In: XII CONGRESSO BRASILEIRO DE AUTOMÁTICA (Uberlândia, MG). *Anais*. v. 1, p. 375-380.
- COELHO, L. S.; COELHO, A. A. R.; 1997a. Rede Parcialmente Recorrente de Elman e Algoritmo Genético Híbrido em Identificação Experimental de um Túnel de Vento. In: CALOBA, L.; BARRETO, J. M. (eds.), III CONGRESSO BRASILEIRO DE REDES NEURAS (Florianópolis, SC). *Anais*. p. 296-301.
- _____; 1997b. Algoritmos Híbridos Inteligentes Aplicados no Controle de um Veículo de Carga. In: CALOBA, L.; BARRETO, J. M. (eds.), III CONGRESSO BRASILEIRO DE REDES NEURAS (Florianópolis, SC). *Anais*. p. 419-424.
- _____; 1997c. Controlador Nebuloso com Otimização Evolutiva Aplicado em Controle de Processos: Um Estudo de Caso. In: II SIMPÓSIO NACIONAL DE INFORMÁTICA (Santa Maria, RS). *Anais*. p. 16-20.
- _____; 1997d. Rede Neurais Artificiais e Computação Evolucionária Aplicadas à Modelagem de um Turbo-Gerador. In: III CONGRESSO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, IV WORKSHOP SOBRE ASPECTOS TEÓRICOS DE LA

INTELIGENCIA ARTIFICIAL-TEORÍA (La Plata, Argentina). *Proceedings*. p. 1055-1062.

_____; 1997e. Computação Evolucionária em Identificação e Controle de Processos: Fundamentos, Análise e Aplicações, In: III CONGRESSO ARGENTINO DE CIENCIAS DE LA COMPUTACIÓN, IV WORKSHOP SOBRE ASPECTOS TEÓRICOS DE LA INTELIGENCIA ARTIFICIAL-TEORÍA (La Plata, Argentina). *Proceedings*. p. 1063-1079.

_____; 1997f. Metodologia para Ensino de Automação: Uma Visão dos Aspectos Teóricos, Práticos e Pedagógicos. In: XXV CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA (Salvador, BA). *Anais*. v. 1, p. 227-242.

_____; 1997g. Inteligência Computacional Aplicada à Modelagem de um Turbo-Gerador. In: XIV CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA (Bauru, SP). *Anais*. CD-ROM. *Abstract*, p. 142.

_____; 1997h. Controladores Nebuloso e Neural com Otimização Evolutiva: Metodologias e Aplicação, In: XIV CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA (Bauru, SP). *Anais*. CD-ROM. *Abstract*, p. 142.

_____; 1997i. Identificação e Controle de Processos através de Algoritmos Genéticos e Estratégias Evolutivas. In: XIV CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA (Bauru, SP). *Anais*. CD-ROM. *Abstract*, p. 289-290.

_____; 1997j. Inteligência Computacional em Controle de um Processo de Nível Duplo-Tanque: Abordagem Evolutiva, Nebulosa e Neural. In: XII CONGRESO CHILENO DE INGENIERÍA ELÉCTRICA (Temuco, Chile). *Proceedings*. v. 2, p. 450-458.

_____; 1997k. Identificação Experimental via Variável Instrumental com Módulo de Configuração Evolutivo. In: XII CONGRESO CHILENO DE INGENIERÍA ELÉCTRICA (Temuco, Chile). *Proceedings*. v. 2, p. 444-449.

_____. Genetic Algorithms and Evolution Strategies Applied in Identification and Control: Case Study. In: CHAUDHRY, P. K.; ROY, R.; PANT, R. K. (eds.). *Soft Computing in Engineering Design and Manufacturing*. Chapter 8: Dynamic System, Identification and Control, Springer-Verlag : London, 1998a. p. 430-438.

_____; 1998b. Computational Intelligence in Process Control: Fuzzy, Evolutionary, Neural, and Hybrid Approaches. *International Journal of Knowledge-Based Intelligent Engineering Systems*, v. 2, n. 2, p. 80-94.

_____; 1998c. Projeto e Sintonia Evolutiva de Controladores PD e PID com Aplicação em um Manipulador Robótico. In: XII CONGRESSO BRASILEIRO DE AUTOMÁTICA (Uberlândia, MG). *Anais*. v. 1, p. 381-386.

_____; 1998d. Identificação de Processos por Algoritmos Inteligentes e Variável Instrumental. In: XII CONGRESSO BRASILEIRO DE AUTOMÁTICA (Uberlândia, MG). *Anais*. v. 4, p. 1267-1272.

- _____. 1998e. Modified Elman Neural Networks Applied to Identification of a Wind Tunnel Process: Design and Experimental Results. In: V SIMPÓSIO BRASILEIRO DE REDES NEURAIIS (Belo Horizonte, MG). *Anais*. p. 165-168.
- _____. 1998f. Sintonia Evolutiva de Controladores PID Clássico e PID Ganho Escalonado Aplicado a um Processo Não-Linear Fan-and-Plate. In: V SIMPÓSIO BRASILEIRO DE REDES NEURAIIS (Belo Horizonte, MG). *Anais*. p. 181-186.
- _____. Fuzzy PID Controllers: Structures, Design Principles and Application for Nonlinear Practical Process. In: ROY, R.; FURUHASHI, T.; CHAUDHRY, P. K. (eds.). *Advances in Soft Computing - Engineering Design and Manufacturing*. Chapter 3: Intelligent Control, Springer-Verlag : London, 1999a. p. 147-159.
- _____. Experimental Evaluation of Intelligent Identification Algorithms Applied to a Wind Tunnel Process. In: ROY, R.; FURUHASHI, T.; CHAUDHRY, P. K. (eds.). *Advances in Soft Computing - Engineering Design and Manufacturing*. Chapter 4: Identification and Modelling, Springer-Verlag : London, 1999b. p. 231-247.
- _____. 1999c. Algoritmos Evolutivos em Identificação e Controle de Processos: Uma Visão Integrada e Perspectivas. *Revista Controle & Automação*, Sociedade Brasileira de Automática, v. 10, n. 1, p. 13-30.
- _____. 1999d. Automatic Tuning of PID and Gain Scheduling PID Controllers by a Derandomized Evolution Strategy. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, v. 13, n. 5, p. 365-373.
- _____. 1999e. Identificação de um Processo Multivariável Balanço Horizontal via Rede Neural Polinomial. In: ISA SHOW BRASIL'99, THE INTERNATIONAL SOCIETY FOR MEASUREMENT AND CONTROL, CONGRESSO INTERNACIONAL DISTRITO 4 (São Paulo, SP). *Anais. Revista InTech Brasil*, n. 17, p. 136-144.
- _____. 1999f. Abordagem Evolutiva e Múltiplos Objetivos em Identificação de Processo Não-Linear. In: 4º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (São Paulo, SP). *Anais*. p. 643-648.
- _____. Identification of Nonlinear Multivariable Processes by Neural Networks: Open-Loop and Closed-Loop Case Studies. In: SUZUKI, Y.; ROY, R.; OVASKA, S. J.; FURUHASHI, T.; DOTE, Y. (eds.). *Soft Computing in Industrial Applications*. Springer-Verlag : London, 2000a.
- _____. New Approaches of Multivariable Cross Coupled Control with Tuning by Evolution Strategies. In: SUZUKI, Y.; ROY, R.; OVASKA, S. J.; FURUHASHI, T.; DOTE, Y. (eds.). *Soft Computing in Industrial Applications*. Springer-Verlag : London, 2000b.
- _____. Multivariable Predictive Control Based on Neural Network Model and Simplex-Evolutionary Hybrid Optimization. In: SUZUKI, Y.; ROY, R.; OVASKA, S. J.; FURUHASHI, T.; DOTE, Y. (eds.). *Soft Computing in Industrial Applications*. Springer-Verlag : London, 2000c.

- COELHO, L. S.; MARCHI, P. A.; COELHO, A. A. R.; 1999a. Projeto de Controle Preditivo Baseado em Identificação de Múltiplos Modelos. In: ISA SHOW BRASIL'99, THE INTERNATIONAL SOCIETY FOR MEASUREMENT AND CONTROL, CONGRESSO INTERNACIONAL DISTRITO 4 (São Paulo, SP). *Anais. Revista InTech Brasil*, n. 17, p. 116-124.
- COELHO, L. S.; MARIANI, V. C.; BORGES, P. S. S.; COELHO, A. A. R.; 1999b. Avaliação da Programação Evolutiva no Projeto de Controle PID Digital: Algoritmo e Aplicação. In: 4º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (São Paulo, SP). *Anais*. p. 649-654.
- COELHO, L. S.; SILVA, A. C.; COELHO, A. A. R.; 1998b. SISCON — Ferramenta de Ensino e Aplicação em Tempo Real de Controle PID Convencional e Adaptativo. In: XII CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA (São Paulo, SP). *CD-ROM* (artigo completo). *Resumos*. p. 324.
- COELHO, L. S.; SIMAS, H.; COELHO, A. A. R.; 1999c. Design Issues and Laboratory Experiments in a Self-Tuning Control Teaching. In: 14TH IFAC WORLD CONGRESS (Beijing, China). *Proceedings*. v. M, p. 217-222.
- _____; 1999d. Inexpensive Apparatus for Control Laboratory Experiments Using Advanced Control Methodologies. In: IEEE CONFERENCE ON DECISION AND CONTROL (Phoenix, AZ, USA).
- CORDÓN, O.; HERRERA, F.; LOZANO, M.; 1995. *A Classified Review on the Combination Fuzzy Logic-Genetic Algorithms Bibliography*. Granada. Technical Report DECSAI-95129, Department of Computer Science and A.I., University of Granada.
- _____. On the Combination of Fuzzy Logic and Evolutionary Computation: A Short Review and Bibliography. In: PEDRYCZ, W. (ed.). *Fuzzy Evolutionary Computation*. Kluwer Academic : Boston, 1997. p. 57-77.
- CORRADINI, M. L.; ORLANDO, G.; 1995. Discrete Variable Structure Control for Nonlinear Systems. In: EUROPEAN CONTROL CONFERENCE (Rome, Italy). *Proceedings*. v. 2, p. 1465-1470.
- _____; 1997. A Discrete Adaptive Variable-Structure Controller for MIMO Systems, and its Application to an Underwater ROV. *IEEE Transactions on Control Systems Technology*, v. 5, n. 3, p. 349-259.
- CORRÊA, M. V.; 1997. *Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Modelos NARMAX Racionais — Aplicação a Sistemas Reais*. Belo Horizonte, MG. Dissertação (Mestrado em Engenharia Elétrica). Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, UFMG.
- CRAIG, J. J.; 1986. *Introduction to Robotics: Mechanics & Control*. Chapter 6, Addison-Wesley : Reading, MA, USA.

- CUESTA, F., GORDILLO, F., ARACIL, J., OLLERO, A.; 1999. Stability Analysis of Nonlinear Multivariable Takagi-Sugeno Fuzzy Control Systems. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 5, p. 508-520.
- CYBENKO, G.; 1989. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control Signals and Systems*, v. 2, p. 303-314.
- DARWIN, C. (1872). *Origin of Species by Means of Natural Selection, or the Preservation of Favored Races in the Struggle for Life*. 6th Edition, v. I and II. John Murray : London, Albemarle Street, 1859 (1a. ed), disponível em:
<http://honors.ccsu.ctstateu.edu/Honors/EText/Darwin/DarwinOriginContents.html>
- DAVIS, L.; 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold : New York, NY, USA.
- DAWKINS, R.; 1989. *The Selfish Gene*. Oxford University Press : UK.
- DE AZEVEDO, F. M.; 1997. Uma Proposta de Modelos Formais de Neurônios e Redes Neurais Artificiais. In: CALOBA, L.; BARRETO, J. M. (eds.). *III Congresso Brasileiro de Redes Neurais* (Florianópolis, SC). p. 503-514.
- DECARLO, R. A.; ZAK, S. H.; MATTHEWS, G. P.; 1988. Variable Structure Control of a Nonlinear Multivariable Systems: A Tutorial. *Proceedings of the IEEE*, v. 76, n. 3, p. 212-232.
- DE FALCO, I.; DEL BALIO, R.; DELLA CIOPPA, A.; TARANTINO, E.; 1996. A Comparative Analysis of Evolutionary Algorithms for Function Optimisation. In: 2ND ON-LINE WORKSHOP ON EVOLUTIONARY COMPUTATION. Hosted in Internet.
- DE KEYSER, R.; 1998. A Gentle Introduction to Model Based Predictive Control. In: INTERNATIONAL WORKSHOP ON CONSTRAINED CONTROL SYSTEMS (Florianópolis, SC). NECCOSYDE: Alfa Network for Constrained Control System Development.
- DIAS, J. S.; ZIMMERMANN, A. C.; BORGES, P. S. S.; BARRETO, J. M.; 1998. Aprendizado e Evolução: de Lamarck a Baldwin. In: V SIMPÓSIO BRASILEIRO DE REDES NEURAIIS (Belo Horizonte, MG).
- DIMEO, R. M.; LEE, K. Y.; 1994. Genetics-Based Control of MIMO Boiler-Turbine Plant. In: 33RD IEEE CONFERENCE ON DECISION AND CONTROL (Lake Buena Vista, FL, USA). *Proceedings*. p. 3512-3517.
- DOLENKO, S. A.; ORLOV, Y. V.; PERSIANTSEV, I. G.; 1996. Practical Implementation and Use of Group Method of Data Handling (GMDH): Prospects and Problems. In: 2ND INTERNATIONAL CONFERENCE ON ADAPTIVE COMPUTING IN ENGINEERING DESIGN AND CONTROL (University of Plymouth, UK). *Proceedings*. p. 291-293.

- DOZIER, G.; BOWEN, J.; HOMAIFAR, A.; 1998. Solving Constraint Satisfaction Problem Using Hybrid Evolutionary Search. *IEEE Transactions on Evolutionary Computation*, v. 2, n. 1, p. 23-33.
- DRACOPOULOS, D. C.; KENT, S.; 1997. Genetic Programming for Prediction and Control. *Neural Computing & Applications*, v. 6, n. 4.
- DU, H.; DUMONT, G. A.; 1996. Constrained Multivariable Control of a Wood Chip Refiner. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 349-354.
- DUBAY, R.; BELL, A. C.; GUPTA, Y. P.; 1997. Control of Plastic Melt Temperature: A Multiple Input Multiple Output Model Predictive Approach. *Polymer Engineering and Science*, v. 37, n. 9, p. 1550-1564.
- DUFFY, J. J.; FRANKLIN, M. A.; 1975. A Learning Identification Algorithm and its Application to an Environmental System. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 5, n. 2, p. 226-239.
- DUONG, T. A.; EBERHARDT, S. P.; DAUD, T.; THAKOOR, A.; 1996. Learning in Neural Networks: VLSI Implementation Strategies. In: CHEN, C. H. (ed.). *Fuzzy Logic and Neural Network Handbook*. Chapter 27, McGraw-Hill : New York. p. 27.1-27-48.
- EATON, J. W.; RAWLINGS, J. B.; UNGAR, L. H.; 1994. Stability of Neural Net Based Model Predictive Control. In: AMERICAN CONTROL CONFERENCE (Baltimore, Maryland, USA). *Proceedings*. p. 2481-2485.
- EIBEN, Á. E., HINTERDING, MICHALEWICZ, Z.; 1999. Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 2, p. 124-141.
- ELMAN, J. L.; 1990. Finding Structure in Time. *Cognitive Science*, v. 14, p. 179-211.
- EVONEWS; 1996. *Newsletter of the EvoNet Network of Excellence in Evolutionary Computation*. European Commission's ESPRIT IV Programme, issue 1.
- _____; 1999. *Newsletter of the EvoNet Network of Excellence in Evolutionary Computation*. European Commission's ESPRIT IV Programme, issue 9, Winter.
- FAHLMAN, S. E.; 1988. An Empirical Study of Learning Speed in Back-Propagation Networks. Pittsburgh, USA. Technical Report, CMU-CS-88-2162, Computer Science Carnegie-Mellon University.
- FANG, Y.; CHOW, T. W. S.; LI, X. D.; 1999. Use of a Recurrent Neural Network in Discrete Sliding-Mode Control. *IEE Proceedings Control Theory and Applications*, v. 146, n. 1, p. 84-90.
- FARDIN, J. F.; ARRUDA, L. V. R.; AMARAL, W. C.; 1997. Identificação Relevante para Controle Preditivo Utilizando Algoritmos Genéticos. In: 3º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (Vitória, ES). p. 295-300.

- FEURING, T.; LIPPE, W. -M.; 1995. Fuzzy Neural Networks are Universal Approximators. In: 6TH INTERNATIONAL FUZZY SYSTEMS ASSOCIATION WORLD CONGRESS (São Paulo, SP). *Proceedings*. v. II, p. 659-662.
- FISCHLE, K.; SCHRÖDER, D.; 1999. An Improved Stable Adaptive Fuzzy Control Method. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 1, p. 27-40.
- FISHER; 1991. Process Control: An Overview and Personal Perspective. *The Canadian Journal of Chemical Engineering*, v. 69, p. 5-26.
- FLEMING, P. J.; FONSECA, C. M.; 1993. Genetic Algorithms in Control Systems Engineering. In: 12TH IFAC WORLD CONGRESS (Sydney, Australia). *Proceedings*. v. 2, p. 383-390.
- FLETCHER, R.; 1987. *Practical Methods of Optimization*, 2nd edition, John Wiley & Sons : New York.
- FOGEL, D. B.; 1994. An Introduction to Simulated Evolutionary Optimization. *IEEE Transactions on Neural Networks*, v. 5, n. 1, p. 3-14.
- _____; 1995. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press : Piscataway, NJ, USA.
- FUKUDA, T.; SHIBATA, T.; 1992. Theory and Applications of Neural Networks for Industrial Control Systems. *IEEE Transactions on Industrial Electronics*, v. 39, n. 6, p. 472-489.
- _____; 1994. Fuzzy-Neuro-GA Based Intelligent Robotics. In: ZURADA, J. M.; MARKS II, R. J., ROBINSON, C. J. (eds.). *Computational Intelligence: Imitating Life*, IEEE Press : Piscataway, NJ. p. 352-363.
- FUKUDA, T.; SHIMOJIMA, K.; 1995. Fusion of Fuzzy, NN, GA to the Intelligent Robotics. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS (Vancouver, British Columbia, Canada). *Proceedings*. p. 2892-2897.
- FULLÉR, R.; 1995. *Neural Fuzzy Systems*. Åbo Akademi University, Series A: 443, Åbo, Finland.
- FURUTA, K.; KOSUGE, K.; KOBAYASHI, K.; 1989. VSS-Type Self-Tuning Control of Direct-Drive Motor. In: 15TH ANNUAL CONFERENCE OF IEEE INDUSTRIAL ELECTRONICS SOCIETY, IECON (Philadelphia, Pennsylvania, USA). *Proceedings*. v. II, p. 281-286.
- FURUTA, K.; 1990. Sliding Mode Control of a Discrete System. *Systems & Control Letters*, v. 14, p. 145-152.
- _____; 1993. VSS Type Self-Tuning Control. *IEEE Transactions on Industrial Electronics*, v. 40, n. 1, p. 37-44.

- FURUTA, K.; PAN, Y.; 1996. Design of Discrete-Time VSS Controller Based on Sliding Sector. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). p. 487-492.
- GABORA, L. M.; 1995. Meme and Variations: A Computational Model of Cultural Evolution. In: NADEL, L.; STEIN, D. L.(eds.). *Lectures in Complex Systems*. Addison Wesley : Reading, MA, USA.
- GAO, W.; HUNG, J. C.; 1993. Variable Structure Control of Nonlinear Systems: A New Approach. *IEEE Transactions on Industrial Electronics*, v. 40, n. 1, p. 45-55.
- GAO, W.; WANG, Y.; HOMAIFA, A.; 1995. Discrete-Time Variable Structure Control Systems. *IEEE Transactions on Industrial Electronics*, v. 42, n. 2, p. 117-122.
- GARIGLIO, D.; HEIDPRIEM, J.; HELGET, A.; 1994. Identification and Control of a Simulated Distillation Plant Using Connectionist and Evolutionary Techniques. *Simulation*, v. 63, n. 6, p. 393-404.
- GODJEVAC, J.; 1993. Comparison Between PID and Fuzzy Control, Lausanne, Switzerland. Internal Report no. 93.36, Département d'Informatique, Laboratoire de Microinformatique, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
- GOLDBERG, D. E.; 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley : Reading, MA, USA.
- GOLDBERG, D. E.; DEB, K.; 1991. A Comparison of Selection Schemes Used in Genetic Algorithms. In: RAWLING, G. J. E. (ed.). *Foundations of Genetic Algorithms*. Morgan Kaufmann : San Mateo, CA, USA. p. 69-93.
- GOLUB, G. H.; VAN LOAN, C. F.; 1983. *Matrix Computations*, Baltimore : The John Hopkins University Press.
- GOMIDE, F.; 1995. Research and Development on Fuzzy Systems in Brazil. *Journal of Japan Society for Fuzzy Theory and Systems*, v. 7, n. 3, p. 86-88.
- GOONATILAKE, S.; KHEBBAL, S.; 1995. *Intelligent Hybrid Systems*. Chichester : John Wiley & Sons, USA.
- GOSSNER, J. R.; KOUVARITAKIS, B.; ROSSITER, J. A.; 1998. Constrained Multivariable Cautious Stable Predictive Control. *IEE Proceedings Control Theory and Applications*, v. 145, n. 5, p. 385-391.
- GOTO, S.; NAKAMURA, M.; KYURA, N.; 1996. Accurate Contour Control of Mechatronic Servo Systems Using Gaussian Networks. *IEEE Transactions on Industrial Electronics*, v. 43, n. 4, p. 469-476.
- GUPTA, M. M.; RAO, D. H.; 1994. Neuro-Control Systems: A Tutorial. In: GUPTA, M. M.; RAO, D. H. (eds.). *Neuro-Control Systems: Theory and Applications*. IEEE Press : Piscataway, NJ, USA. p. 1-43.

- HA, Q. P.; RYE, D. C.; DURRANT-WHYTE, H. F.; 1999. Fuzz Moving Sliding Mode Control with Application to Robotic Manipulators. *Automatica*, v. 35, p. 607-616.
- HABER, R.; UNBEHAUEN, H.; 1990. Structure Identification of Nonlinear Dynamic Systems — A Survey on Input/Output Approaches. *Automatica*, v. 26, n. 4, p. 651-677.
- HAGAN, M. T.; DEMUTH, H. B.; 1999. Neural Networks for Control. In: AMERICAN CONTROL CONFERENCE (San Diego, CA, USA). *Proceedings*. p. 1642-1656.
- HAGAN, M. T.; MEHNAJ, B. M.; 1994. Training Feedforward Networks with Marquardt Algorithm. *IEEE Transactions on Neural Networks*, v. 5, n. 6, p. 989-993.
- HANSEN, J. F.; ÅDNANES, A. K.; FOSSEN, T. I.; 1998. Modelling, Simulation and Multivariable Model-Based Predictive Control of Marine Power Generation System. In: IFAC CONFERENCE ON CONTROL APPLICATIONS IN MARINE SYSTEMS (Fukuoka, Japan). *Proceedings*. p. 45-51.
- HARRIS, C. J.; BROWN, M.; BOSSLEY, K. M.; MILLS, D. J.; MING, F.; 1996. Advances in Neurofuzzy Algorithms for Real-Time Modeling and Control. *Engineering Applications of Artificial Intelligence*, v. 9, n. 1, p. 1-16.
- HARRIS, C. J.; MOORE, C. G.; BROWN, M.; 1993. *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*. World Scientific Series in Robotics and Automation Systems, v. 6, World Scientific : London.
- HASKARA, İ.; ÖZGÜNER, Ü.; UTKIN, V.; 1997. Variable Structure Control for Uncertain Sampled Data Systems. In: 36TH IEEE CONFERENCE ON DECISION & CONTROL (San Diego, CA, USA). *Proceedings*. p. 3226-3231.
- HAYASHI, K.; SHIMIZU, Y.; DOTE, Y.; TAKAYAMA, A.; HIRAKO, A.; 1995. Neuro Fuzzy Transmission Control for Automobile with Variable Loads. *IEEE Transactions on Control Systems Technology*, v. 3, n. 1, p. 49-53.
- HAYKIN, S.; 1994. *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company : Englewood Cliffs, NJ, USA.
- HEALAY, A. J.; LIENARD, D.; 1993. Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, v. 18, n. 3, p. 327-338.
- HEITKOETTER, J.; BEASLEY, D.; 1996. *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)*. Available via anonymous FTP form rtfm.mit.edu: /pub/usenet /news.answers/ai-faq/genetic.
- HENSON, M. A.; SEBORG, D. E. (eds.); 1997. *Nonlinear Process Control*, Prentice Hall PTR : Upper Saddle River, NJ, USA.
- HIGUCHI, T.; KAJIHARA, N.; 1999. Evolvable Hardware Chips for Industrial Applications. *Communications of the ACM*, v. 42 , n. 4, p. 60-66.

- HINTON, G. E., NOWLAN, S. J.; 1987. How Learning Can Guide Evolution. *Complex Systems*, v. 1, p. 495-502.
- HIROTA, K.; SUGENO, M.; 1995. *Industrial Applications of Fuzzy Technology in the World, Advances in Fuzzy Systems — Applications and Theory*, v. 2, World Scientific : Singapore.
- HJALMARSSON, H.; GEVERS, M.; DE BRUYNE, F.; 1996. For Model-Based Control Design, Closed-Loop Identification Gives Better Performance. *Automatica*, v. 32, n. 12, p. 1659-1673.
- HOLLAND, J. H.; 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press (Reprinted in 1992 by MIT Press).
- HOLMBLAD, L. P.; OSTERGAARD, J. -J.; 1982. Control of a Cement Kiln by Fuzzy Logic, in *Fuzzy Information and Decision Processes*. In: GUPTA, M. M.; SANCHEZ, E. (eds.), North-Holland : Amsterdam. p. 389-399.
- HOPFIELD, J. J.; 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, v. 79, p. 2554-2558.
- _____. 1984. Neurons with Graded Response have Collective Computational Properties like those of Two-State Neurons. *Proceedings of the National Academy of Sciences*, v. 81, pp. 3088-3092.
- HUNG, J. Y.; GAO, W. B.; HUNG, J. C.; 1993. Variable Structure Control: A Survey. *IEEE Transactions on Industrial Electronics*, v. 40, n. 1, p. 2-22.
- HUNT, K. J.; HAAS, R.; MURRAY-SMITH, R.; 1994. Direct Implementation of Fuzzy Control with Basis Function Networks. In: 33RD IEEE CONFERENCE ON DECISION AND CONTROL (Lake Buena Vista, FL, USA). *Proceedings*. p. 4146-4147.
- HUNT, K. J.; SBARBARO, D.; ZBIKOWSKI, R.; GAWTHROP, P.; 1992. Neural Networks for Control Systems — A Survey. *Automatica*, v. 28, n. 6, p. 1083-1112.
- ICHIHASHI, H.; HARADA, N.; NAGASAKA, K.; 1995. Selection of the Optimum Number of Hidden Layers in Neuro-Fuzzy GMDH. In: *FUZZ-IEEE/IFES* (Yokohama, Japan). *Proceedings*. v. 3, p. 1519-1526.
- INGBER, L.; 1989. Very Fast Simulated Re-annealing. *Mathematical Computer Modelling*, v. 12, p. 967-973.
- IORDANOU, H. N.; SURGENOR, B. W.; 1997. Experimental Evaluation of the Robustness of Discrete Sliding Mode Control Versus Linear Quadratic Control. *IEEE Transactions on Control Systems Technology*, v. 5, n. 2, p. 254-260.
- ISHII, T.; KYUMA, K.; 1996. Configuring Interaction of Memorized Patterns with an Asymmetric Hebbian Rule for Recurrent Neural Networks. *Neurocomputing*, v. 10, n. 1, p. 43-53.

- IVAKHNENKO, A. G.; 1970 Heuristic Self-Organization in Problems of Engineering Cybernetics. *Automatica*, v. 6, p. 207-219.
- _____; 1971. Polynomial Theory of Complex Systems. *IEEE Transactions Systems, Man, and Cybernetics*, v. 1, n. 4, p. 364-378.
- IVAKHNENKO, A. G.; IVAKHNENKO, G. A.; MÜLLER, J. -A.; 1994. Self-Organization of Neural Networks with Active Neurons. *Pattern Recognition and Image Analysis*, v. 4, n. 2, p. 185-196.
- JACOBS, R. A.; 1988. Increased Rates of Convergence Through Learning Rate Adaptation. *Neural Networks*, v. 1, pp. 295-307.
- JAGANNATHAN, S.; LEWIS, F. L.; 1996. Identification of Nonlinear Dynamical Systems Using Multilayered Neural Networks. *Automatica*, v. 32, n. 12, p. 1707-1712.
- JAGER, R.; 1995. *Fuzzy Logic in Control*. Delft, Netherlands. Thesis, Department of Electrical Engineering, Delft University of Technology, The Netherlands.
- JANG, J. -S. R.; SUN, C. -T.; 1993a. Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems. *IEEE Transactions on Neural Networks*, v. 4, n. 1, p. 156-159.
- _____; 1995. Neuro-Fuzzy Modeling and Control. *Proceedings of the IEEE*, v. 83, n. 3, p. 378-406.
- JASON, D. J.; FRENZEL, J. F.; 1995. Training Product Unit Neural Networks with Genetic Algorithms. *IEEE Expert*, v. 8, n. 5, p. 26-33.
- JOHNSON, M. A.; 1996. Intelligent Methods of Parameter Estimation for Adaptive Control. In: JAMSHIDI, M.; YUH, J.; DAUCHEZ, P. (eds). *Intelligent Automation and Control: Recent Trends in Development and Applications*, v. 4. Proceedings of the World Automation Congress (Montpellier, France). TSI Press Series. p. 337-342.
- JOHANSEN, T. A.; FOSS, B. A.; 1993. State-space Modeling Using Operating Regime Decomposition and Local Models. In: 12TH IFAC WORLD CONGRESS (Sydney, Australia). *Proceedings*. v. 1, p. 431-434.
- JOHANSSON, R.; 1993. *System Modeling Identification*. Prentice-Hall : Englewood Cliffs, NJ, USA.
- JONES, J.; HOUCK, C.; 1994. On the Use of Non-Stationary Penalty Functions to Solve Constrained Optimization Problems with Genetic Algorithms. In: IEEE INTERNATIONAL SYMPOSIUM EVOLUTIONARY COMPUTATION (Orlando, FL, USA). *Proceedings*. p. 579-584.
- JORDAN, J. L.; 1988. Serial Order: A Parallel Distributed Processing Approach. In: ERLBAUM, K., ELMAN, J. L.; RUMELHART, D. E. (eds.). *Advances in Connectionist Theory: Speech*.

- KARAMA, A.; BENHAMMOU, A.; VASILACHE, A.; ROUX, G.; DAHHOU, B. ; 1998. Neural Predictive Control of a Modelled Fermentation Bioreactor. In: CONFÉRENCE MEDITERRANÉENNE SUR L'ELECTRONIQUE ET L'AUTOMATIQUE (Marrakeck, Maroc). *Proceedings*. p. 228-232.
- KARR, C. L.; GENTRY, E. J.; 1993. Fuzzy Control of pH Using Genetic Algorithms. *IEEE Transactions on Fuzzy Systems*, v. 1, n. 1, p. 46-53.
- KASABOV, N.; 1999. Evolving Connectionist and Fuzzy-Connectionist Systems for On-line Adaptive Decision Making and Control. In: ROY, R.; FURUHASHI, T.; CHAUDHRY, P. K. (eds.). *Advances in Soft Computing - Engineering Design and Manufacturing*. Chapter 3: Intelligent Control, Springer-Verlag : London, 1999. p. 20-29.
- KATAYAMA, R.; KAJITANI, Y.; KUWATA, K.; NISHIDA, Y.; 1993. Developing Tools and Methods for Applications Incorporating Neuro, Fuzzy and Chaos Technology. *Computers on Industrial Engineering*, v. 24, n. 4, p. 579-592.
- KAYNAK, O.; RUDAS, I. J.; 1998. The Fusion of Computational Intelligence Methodologies in Sliding Mode Control. In: 24TH ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY, IECON (Aachen, Germany). *Tutorial*. p. T25-T34.
- KENNEDY, S. A.; 1993. Five Ways to a Smarter Genetic Algorithm. *AI Expert*, December, p. 35-38.
- KHALID, M.; OMATU, S.; YUSOF, R.; 1993. MIMO Furnace Control with Neural Networks. *IEEE Transactions on Control Systems Technology*, v. 1, n. 4, p. 238-245.
- KHOSLA, R.; DILLON, T.; 1997. *Engineering Intelligent Hybrid Multi-Agent Systems*. Kluwer Academic Publishers : Boston, USA.
- KICKERT, W.; VAN NAUTA LEMKE, H. R.; 1976. Application of a Fuzzy Controller in a Warm Water Plant. *Automatica*, v. 12, p. 301-318.
- KIM, B. M.; KIM, Y. B.; OH, C. H.; 1997a. A Study on the Convergence of Genetic Algorithms. *Computers on Industrial Engineering*, v. 33, n. 3-4, p. 581-588.
- KIM, E.; PARK, M.; JI, S.; PARK, M.; 1997b. A New Approach to Fuzzy Modeling. *IEEE Transactions on Fuzzy Systems*, v. 5, n. 3, p. 328-337.
- KIM, J.; MOON, Y.; ZIEGLER, B. P.; 1995. Designing Fuzzy Net Controllers Using Genetic Algorithms. *IEEE Control Systems*, v. 15, p. 66-72.
- KIRIAKIDIS, K.; 1998. Fuzzy Model Based Control of Complex Plants. *IEEE Transactions on Fuzzy Systems*, v. 6, n. 4, p. 517-529.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P.; 1983. Optimization by Simulated Annealing. *Science*, v. 220, p. 45-54.

- KITANO, H.; 1990. Empirical Studies on the Speed of Convergence of Neural Network Training using Genetic Algorithms. In: 8TH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (Boston, USA). *Proceedings*. v. 2, p. 789-795.
- KÓCZY, L. T.; 1994. European Applications of Fuzzy Logic with the Eyes of a European in Japan. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (Orlando, FL, USA). v. III, p. 1701-1705.
- KOLLIAS, S.; ANASTASSIOU, D.; 1989. An Adaptive Least Squares Algorithm for the Efficient Training of Artificial Neural Networks. *IEEE Transactions on Circuits and Systems*, v. 36, n. 8, p. 1092-1101.
- KORENBERG, M.; BILLINGS, S. A.; LIU, Y. P.; MCILROY, P. J.; 1988. Orthogonal Parameter Estimation Algorithm for Non-Linear Stochastic Systems. *International Journal of Control*, v. 48, n. 1, p. 193-210.
- KOSKO, B.; 1992. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall : Englewood Cliffs, NJ, USA.
- KOZA, J. R.; 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press : Cambridge, MA.
- _____; 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press : Cambridge, MA.
- KOZA, J. R.; BENNETT III, F. H.; ANDRE, D.; KEANE, M.; 1997. Automated WYWIWYG Design of Both the Topology and Component Values of Electrical Circuits Using Genetic Programming. In: KOZA, J. R.; GOLDBERG, D. E.; FOGEL, D. B.; RIOLO, R. L. (eds.). *Genetic Programming 1996: 1st Annual Conference. Proceedings*. MIT Press : Stanford University, CA, USA. p. 123-131.
- KROHLING, R. A.; COELHO, L. S.; COELHO, A. A. R.; 1999. Evolution Strategies for Synthesis of Mixed H_2/H_∞ Fixed-Structure Controllers. In: 14TH IFAC WORLD CONGRESS (Beijing, China). *Proceedings*. v. F, p. 471-476.
- KRÜGER, A. E.; COELHO, L. S.; COELHO, A. A. R.; 1997. Automatização de uma Bancada de Ensaio com Aplicação em um Pêndulo Invertido: Uma Ferramenta Educacional. In: XXV CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA (Salvador, BA). *Anais*. v. 1, p. 342-352.
- KWOK, D. P.; TAN, P.; LI, C. K.; WANG, P.; 1990. Linguistic PID Controllers. In: 11TH IFAC WORLD CONGRESS (Tallinn, Estonia, USSR). *Proceedings*. v. 7, p. 192-197.
- LANDAU, I. D.; KARIMI, A.; 1997. Recursive Algorithms for Identification in Closed Loop: A Unified Approach and Evaluation. *Automatica*, v. 33, n. 8, p. 1499-1523.
- LAUKONEN, E. G.; PASSINO, K. M.; 1995. Training Fuzzy Systems to Perform Estimation and Identification. *Engineering Applications of Artificial Intelligence*, v. 8, n. 5, p. 499-514.

- LAZZERINI, B.; REYNERI, L. M.; CHIABERGE, M.; 1999. A Neuro-Fuzzy Approach to Hybrid Intelligent Control. *IEEE Transactions on Industry Applications*, v. 35, n. 2, p. 413-425.
- LEE, H.; KI, E.; KANG, H. -J.; PARK, M.; 1998. Design of a Sliding Mode Controller with Fuzzy Sliding Surfaces, *IEE Proceedings Control Theory and Applications*, v. 145, n. 5, p. 411-418.
- LEE, P.-M.; OH, J.-H.; 1998. Improvements on VSS-Type Self-Tuning Control for a Tracking Controller. *IEEE Transactions on Industrial Electronics*, v. 45, n. 2, p. 319-325.
- LEONARD, J. A.; KRAMER, M. A.; 1991. Radial Basis Function Networks for Classifying Process Faults. *IEEE Control Systems*, v. 11, p. 31-38.
- LEONTARITIS, I. J.; BILLINGS, S. A.; 1985. Input-Output Parametric Models for Non-Linear Systems. Part I: Deterministic Non-Linear Systems, Part II: Stochastic Non-Linear Systems. *International Journal of Control*, v. 41, n. 2, p. 303-344.
- LEWIS, F. L.; PARISINI, T.; 1998. Guest editorial: Neural Network Feedback Control with Guaranteed Stability. *International Journal of Control*, v. 70, n. 3, p. 337-340.
- LEVENBERG, K.; 1944. A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quart. Applied Mathematics*, v. 2, n. 2, p. 164-168.
- LI, H. X.; GATLAND, H. B.; 1995. Enhanced Methods of Fuzzy Logic Control. In: FUZZ-IEEE/IFES (Yokohama, Japan). *Proceedings*. v. 1, p. 331-336.
- LI, X., VAN DER WAL, A. J.; RUAN, D.; 1998. Discussion on Soft Computing at FLINS'96. *International Journal of Intelligent Systems*, v. 13, n. 2-3, p. 287-300.
- LI, Y.; NG, K. C.; 1997. Reduced Rule-Base and Direct Implementation of Fuzzy Logic Control. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 85-90.
- LI, Y.; NG, K. C.; MURRAY-SMITH, D. J.; GRAY, G. J.; SHARMAN, K. C.; 1996. Genetic Algorithm Automated Approach to the Design of Sliding Mode Control Systems. *International Journal of Control*, v. 63, n. 4, p. 721-739.
- LIGHTBODY, G.; IRWIN, G. W.; 1996. Multi-Layer Perceptron Based Modelling of Nonlinear Systems. *Fuzzy Sets and Systems*, v. 79, p. 93-112.
- LIN, J. -N.; SONG, S. -M.; 1994. A Novel Fuzzy Neural Network for the Control of Complex Systems. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (Orlando, FL, USA). *Proceedings*. v. III, p. 1668-1663.
- LINDSEY, C. S.; LINDBLAD, T.; 1994. Review of Hardware Neural Networks: A User's Perspective. In: THIRD WORKSHOP ON NEURAL NETWORKS: FROM BIOLOGY TO HIGH ENERGY PHYSICS. Plenary Talk. Marciana Marina, Isola d'Elba, Italy (<http://msia02.msi.se/~lindsey/elba2html/elba2html.html>).

- LING, C.; EDGAR, T. F.; 1993. The Tuning of MIMO Fuzzy Heuristic Controllers. In: 12TH IFAC WORLD CONGRESS (Sydney, Australia). *Proceedings*. v. 6, p. 67-70.
- LINKENS, D. A.; NYONGESA, H. O.; 1995. Genetic Algorithms for Fuzzy Control, Part 1 and Part 2, *IEE Proceedings Control Theory and Applications*, v. 142, n. 3, p. 161-185.
- _____; 1996a. Learning Systems in Intelligent Control: An Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications, *IEE Proceedings Control Theory and Applications*, v. 143, n. 4, p. 367-386.
- _____; 1996b. A Hierarchical Multivariable Fuzzy Controller for Learning with Genetic Algorithms. *International Journal of Control*, v. 63, n. 5, p. 865-883.
- LIU, C.; XU, J. -X.; HANG, C. -C.; 1997. Comparison Between a Fuzzy PID Controller and a Kind of Nonlinear PID Controller. In: 36TH IEEE CONFERENCE ON DECISION & CONTROL (San Diego, CA, USA). *Proceedings*. p. 2736-2741.
- LJUNG, L.; 1987. *System Identification: Theory for the User*. Prentice-Hall : New York.
- _____; 1995. System Identification. Linköping, Sweden. Technical Report LiTH-ISY-R-1763, Department of Electrical Engineering, Linköping University.
- LU, J.; JASPER, W.; LEE, G. K.; 1994. A Multivariable Self-Learning Fuzzy Control Algorithm for Dyeing Processes. In: AMERICAN CONTROL CONFERENCE, (Baltimore, Maryland, USA).. *Proceedings*. p. 983-987.
- LU, Y. -Z.; 1992. The New Generation of Advanced Process Control. *Control Engineering International*, Mid-March, p. 21-23.
- LUIZ, C. C.; SILVA, A. C.; COELHO, L. S.; COELHO, A. A. R.; 1998. Control Adaptativo versus Control Fuzzy: un Estudio de Caso en un Proceso de Nivel. *Revista Información Tecnológica*, Chile, v. 9, n. 2, p. 279-284.
- MAGNI, L.; BASTIN, G.; WERTZ, V.; 1999. Multivariable Nonlinear Predictive Control of Cement Mills. *IEEE Transactions on Control Systems Technology*, v. 7, n. 4, p. 502-508.
- MAHFOUF, M.; LINKENS, D. A.; 1997. Constrained Multivariable Generalized Predictive Control for Anesthesia: the Quadratic-Programming Approach. *International Journal of Control*, v. 67, n. 4, p. 507-527.
- MALKI, H. A.; MISIR, D.; FEIGENSPAN, D.; CHEN, G.; 1997. Fuzzy PID Control of a Flexible-Joint Robot Arm with Uncertainties From Time-Varying Loads. *IEEE Transactions on Control Systems Technology*, v. 5, n. 3, p. 371-378.
- MAMDANI, E.; ASSILIAN, S.; 1975. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, v. 7, n. 1, p. 1-13.
- MAN, K. F.; TANG, K. S.; KWONG, S.; 1996. Genetic Algorithms: Concepts and Applications. *IEEE Transactions on Industrial Electronics*, v. 43, n. 5, p. 519-534.

- MANDISCHER, M.; 1993. Representation and Evolution of Neural Networks. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL NEURAL NETS AND GENETIC ALGORITHMS (Innsbruck, Áustria). *Proceedings*. p. 643-649.
- MANN, G. K. I.; HU, B. -G.; GOSINE, R. G.; 1999. Analysis of Direct Action Fuzzy PID Controller Structures. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 29, n. 3, p. 371-388.
- MARCHI, P. A.; 1999. *Abordagem Não-Linear para Modelagem e Controle Preditivo: Estudo de Caso*. Florianópolis, SC. Dissertação (Mestrado em Engenharia Elétrica), Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, UFSC.
- MARCHI, P. A.; COELHO, L. S.; COELHO, A. A. R.; 1999. Comparative Study of Parametric and Structural Methodologies in Identification of an Experimental Nonlinear Process. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Kohala Coast, Hawaii, USA). *Proceedings*. p. 1062-1067.
- MARQUARDT, D. W.; 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal of Applied Mathematics*, v. 11, n. 2, p. 431-441.
- MARTÍNEZ, M.; SENENT, J.; BLASCO, X.; 1996. A Comparative Study of Classical vs Genetic Algorithm Optimization Applied in GPC Controller. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 327-332
- MATAUSEK, M. R.; MICIC, A. P.; 1999. On the Modified Smith Predictor for Controlling a Porcess with an Integrator and Long Dead-Time, *IEEE Transactions on Automatic Control*, v. 44, n. 8, pp. 1603-1606.
- MCINTOSH, A. R.; SHAH, S. L.; FISHER, D. G.; 1991. Analysis and Tuning of Adaptive Generalized Predictive Control. *The Canadian Journal of Chemical Engineering*, v. 69, p. 97-110.
- MEDSKER, L.; BAILEY, D.; 1992. Models and Guidelines for Integrating Expert Systems and Neural Networks, In: KANDEL, A.; LANGHOLZ, G. (eds.). *Hybrid Architectures for Intelligent Systems*, CRC Press, Boca Raton, FL, USA, p. 154-171.
- MENDEL, J. M.; 1995. Fuzzy Logic Systems for Engineering: A Tutorial. *Proceedings of the IEEE*, v. 83, n. 3, p. 345-377.
- MICHALEWICZ, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin : Germany.
- MILLS, D.; HARRIS, C. J.; 1995. Neurofuzzy Modelling and Control of a Six Degree of Freedom AUV. Southampton, UK. Technical Report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton.
- MINSKY, M.; PAPERT, S.; 1969. *Perceptrons*. MIT Press : Cambridge, MA.
- MISIR, D.; MALKI, H. A.; CHEN, G.; 1996. Design and Analysis of a Fuzzy Proportional-Integral-Derivative Controller. *Fuzzy Sets and Systems*, v. 79, p. 297-314.

- MISTRY, S. I.; NAIR, S. S.; 1994. Identification and Control Experiments Using Neural Designs. *IEEE Control Systems*, v. 14, n. 3, p. 48-57.
- MITAL, D. P.; CHIN, L.; 1995. Intelligent Control Applications with Neural Networks. In: GUPTA, M. M.; SINHA, N. K. (eds.). *Intelligent Control Systems: Theory and Applications*, Chapter 18, Piscataway, NJ, USA, IEEE Press, p. 479-514.
- MITCHELL, M.; 1996. *An Introduction to Genetic Algorithms*. A Bradford Book, The MIT Press : Cambridge, England.
- MONTANA, D. J.; 1994. Strongly Typed Genetic Programming. BBN Technical Report 7866, Bolt Boranek and Newman, Inc., 10 Moulton Street, Cambridge, MA, 02138, USA.
- MOODY, J.; DARKEN, C. J.; 1989. Fast Learning in Networks of Locally-tuned Processing Units. *Neural Computation*, v. 1, n. 2, p. 281-294.
- MORIMOTO, T.; HASHIMOTO, Y.; 1996. Fuzzy Control for Fruit Storage Optimization Using Neural Networks and Genetic Algorithms. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 375-380.
- MUTOH, Y.; 1998. Invariant Properties of an Interactor for Nonlinear Multivariable Systems. In: IFAC CONFERENCE ON SYSTEM STRUCTURE AND CONTROL (Nantes, France). *Proceedings*. v. 3, p. 783-788.
- MUTOH, Y.; ORTEGA, R.; 1993. Interactor Structure Estimation for Adaptive Control of Discrete-Time Multivariable Nondecouplable Systems. *Automatica*, v. 29, n. 3, p. 635-647.
- MØLLER, M.; 1994. *Efficient Training of Feed-Forward Neural Networks*. Aarhus University, Denmark. PhD thesis, Computer Science Department, Aarhus University, Denmark.
- NANDAM, P. K.; SEN, P. C.; 1992. Control Laws for Sliding Mode Speed Control of Variable Speed Drives. *International Journal of Control*, v. 56, n. 5, p. 1167-1186.
- NARENDRA, K. S.; BALAKRISHNAN, J.; 1994. Improving Transient Response of Adaptive Control Systems Using Multiple Models and Switching. *IEEE Transactions on Automatic Control*, v. 39, n. 9, p. 1861-1866.
- NARENDRA, K. S.; MUKHOPADHYAY, S.; 1993. Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks. In: 32ND IEEE CONFERENCE ON DECISION AND CONTROL (San Antonio, TX, USA). *Proceedings*. p. 3066-3071.
- NARENDRA, K.; PARTHASARATHY, K.; 1990. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 4-27.
- NASH, J. C.; 1990. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*. 2nd edition. Adam Hilger: New York, USA.

- NAVARRO, J. L.; ALBERTOS, P.; 1996. Fuzzy Logic Implementation of Industrial Controllers. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 409-414.
- NAUCK, D.; 1995. Beyond Neuro-Fuzzy: Perspectives and Directions. In: 3RD EUROPEAN CONGRESS ON INTELLIGENT TECHNIQUES AND SOFT COMPUTING (Aachen, Germany). *Proceedings*. p. 1159-1164.
- NAZZETTA, R. M.; OLIVEIRA, G. H. C.; CAMPELLO, R. J. G. B.; AMARAL, W. C.; 1998. Controle Preditivo de Processos Não Lineares Utilizando Modelos Dinâmicos Nebulosos. In: XII CONGRESSO BRASILEIRO DE AUTOMÁTICA (Uberlândia, MG). *Anais*. p. 169-174.
- NELDER, J. A.; MEAD, R.; 1965. A Simplex Method for Function Minimisation. *Computer Journal*, v. 7, pp. 308-313.
- NELSON, M. M.; ILLINGWORTH, W. T.; 1991. *A Practical Guide to Neural Nets*. Addison-Wesley : Reading, MA, USA.
- NG, K. C.; LI, Y.; 1994. Design of Sophisticated Fuzzy Logic Controllers Using Genetic Algorithms. In: IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 3RD IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (Orlando, FL, USA). *Proceedings*. v. 3, p. 1708-1712.
- NI, H.; DEBELAK, K.; HUNKELER, D.; 1995. Fuzzy Logic Supervised ANN Feedforward Control For a Batch Polymerization Reactor. In: AMERICAN CONTROL CONFERENCE (Seattle, Washington, USA). *Proceedings*. p. 1777-1781.
- NIE, J.; 1997. Fuzzy Control of Multivariable Nonlinear Servomechanisms with Explicit Decoupling Scheme. *IEEE Transactions on Fuzzy Systems*, v. 5, n. 2, p. 304-311.
- NOBRE, F. S. M.; 1995. Genetic-Neuro-Fuzzy Systems: A Promising Fusion. In: FUZZ-IEEE/IFES (Yokohama, Japan). *Proceedings*. v. I, p. 259-265.
- NOBRE, F. S. M.; PALHARES, Á. G. B.; 1997. Informações Quantitativas e Qualitativas no Projeto de Controladores Nebulosos. *Revista Controle & Automação*, Sociedade Brasileira de Automática, v. 8, n. 2, p. 77-93.
- NORQUAY, S. J.; PALAZOGLU, A.; ROMAGNOLI, J. A.; 1999. Application of Wiener Model Predictive Control (WMPC) to a pH Neutralization Experiment. *IEEE Transactions on Control Systems Technology*, v. 7, n. 4, p. 437-445.
- OEI, C. K.; GOLDBERG, D. E.; CHANG, S. J.; 1991. Tournament Selection, Niching and the Preservation of Diversity. Urbana, IL, USA. Technical Report 91011. Genetic Algorithms Laboratory, University of Illinois.

- OLIVEIRA, J. V.; LEMOS, J. M., 1997. Simplifying Fuzzy Relational Structures for Adaptive Control. In: 7TH INTERNATIONAL FUZZY SYSTEMS ASSOCIATION WORLD CONGRESS, IFSA'97 (Prague, Czech Republic). *Proceedings*. p. 330-335.
- OLIVEIRA, R. C. L., 1998. *Rede Neural com Dinâmica Interna Aplicada a Problemas de Identificação e Controle Não-Linear*. Florianópolis, SC. Tese (Doutorado em Engenharia Elétrica), PPGEEL, Universidade Federal de Santa Catarina, UFSC.
- ORDÓÑEZ, R.; SPOONER, J. T.; PASSINO, K. M.; 1996. Stable Multi-Input Multi-Output Adaptive Fuzzy Control. In: 35TH IEEE CONFERENCE ON DECISION & CONTROL (Kobe, Japan). *Proceedings*. p. 610-615.
- OREN, S.; LUENBERGER, D.; 1974. Self-Scaling Variable Metric (SSVM) Algorithms, Part II: Criteria and Sufficient Conditions for Scaling a Class of Algorithms. *Management Science*, v. 20, n. 5, p. 845-862.
- OSTERMEIER, A.; GAWELCZYK, A.; HANSEN, N.; 1994. Step-size Adaptation on Non-local Use of Selection Information. In: 3RD INTERNATIONAL CONFERENCE ON PARALLEL PROBLEM FORM NATURE (Jerusalem, Israel). *Proceedings*. Springer-Verlag : Berlin, p. 189-198.
- _____; 1995. A Derandomized Approach to Self-Adaptation of Evolution Strategies. *Evolutionary Computation*, v. 2, n. 4, p. 369-380.
- PACHECO, S. S.; THOMÉ, A. G.; 1997. "Turbo-Shake" — Um Avanço "Shake" no Treinamento de Redes Neurais. In: CALOBA, L.; BARRETO, J. M. (eds.). *III Congresso Brasileiro de Redes Neurais* (Florianópolis, SC). *Anais*. p. 51-56.
- PALM, R.; 1994. Robust Control by Fuzzy Sliding Mode. *Automatica*, v. 30, n. 9, p. 1429-1437.
- PAN, J.; LEVINE, W. S.; 1992. *Theoretical and Experimental Study of Order Estimation*. Technical Report 92-31. Systems Research Center and Electrical Engineering Department. University of Maryland, College Park, USA.
- PARK, D.; KANDEL, A.; LANGHOLTZ, G.; 1994. Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 24, n. 1, p. 39-47.
- PARMA, G. G.; MENEZES, B. R.; BRAGA, A. P.; 1998. Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks. *IEE Electronics Letters*, v. 34, n. 1, p. 97-98.
- PATTON, R. J., LIU G. P.; 1994. Robust Control Design via Eigenstructure Assignment, Genetic Algorithms and Gradient-based Optimisation. *IEE Proceedings Control Theory and Applications*, v. 141, n. 3, p. 202-207.

- PEARSON, R. K.; OGUNNAIKE, B. A.; 1997. Nonlinear Process Identification. In: HENSON, M. A.; SEBORG, D. E. (eds.). *Nonlinear Process Control*, Prentice Hall PTR, Upper Saddle River : NJ, USA. Chapter 2, p.11-110.
- PEDRYCZ, W. (1989). *Fuzzy Control and Fuzzy Systems*. John Wiley & Sons : New York, USA.
- PEREIRA, J. S.; BOWLES, J. B.; 1996. Comparing Controllers with the Ball in a Tube Experiment. In: 5TH IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (New Orleans, Louisiana, USA). *Proceedings*. p. 504-510.
- PHAM, D. T.; JIN, G.; 1995. Genetic Algorithm Using Gradient-Like Reproduction Operator. *IEE Electronics Letters*, v. 31, n. 18, p.1558-1559.
- PHAM, D. T.; LIU, X.; 1992. Modelling using Partially Recurrent Neural Network. *Journal of System Engineering*, v. 2, p. 90-97.
- PIEPER, J. K.; SURGENOR, B. W.; 1993. Discrete Sliding Mode Control of a Coupled-Drives Apparatus with Optimal Sliding Surface and Switching Gain. *IEE Proceedings Control Theory and Applications*, v. 140, n. 2, p. 70-80.
- PODOLAK, I. T.; 1998. Functional Graph Model of a Neural Network. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 28, n. 6, p. 876-881.
- POGGIO, T.; GIROSI, F.; 1990. Network for Approximation and Learning. *Proceedings of the IEEE*, v. 78, n. 9, p. 1481-1497.
- POSNIAK, A. S.; YU, W.; SANCHEZ, E. N.; SIRA-RAMIREZ, H.; 1998. Robust Identification by Dynamic Neural Networks Using Sliding Mode Learning. *Applied Mathematics and Computer Science*, v. 8, n. 1, p. 135-144.
- PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P.; 1994. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge Press.
- QIAO, W. Z.; MIZUMOTO, M.; 1996. PID Type Fuzzy Controller and Parameters Adaptive Method. *Fuzzy Sets and Systems*, v. 78, p. 23-35.
- QIN, S. J. (1994). Auto-Tuned Fuzzy Logic Control. In: AMERICAN CONTROL CONFERENCE (Baltimore, Maryland, USA). *Proceedings*. p. 2465-2469.
- QIN, S. J.; BADGWELL, T. J.; 1997. An Overview of Model Predictive Control Technology. In: KANTOR, J. C.; GARCIA, C. E.; CARNAHAN, B. (eds.), 5TH INTERNATIONAL CONFERENCE ON CHEMICAL PROCESS CONTROL, AIChE and CACHE (Tahoe, CA, USA). *Proceedings*. AIChE Symposium Series 316. v. 93, p. 232-256.
- _____; 1998. An Overview of Nonlinear Model Predictive Control Applications. In: ALLGÖWER, F.; ZHENG, A. (eds), INTERNATIONAL SYMPOSIUM ON NONLINEAR MODEL PREDICTIVE CONTROL WORKSHOP — ASSESSMENT AND FUTURE DIRECTIONS (Ascona, Switzerland). *Preprints*.

- RADCLIFFE, N. J.; SURRY, P. D.; 1995. Fundamental Limitations on Search Algorithms: Evolutionary Computing in Perspective. In: VAN LEEUWEN, J. (ed.). *Computer Science Today: Recent Trends and Developments*. Springer-Verlag, LNCS 1000, p. 275-291.
- RAJAPAKSE, A.; FURUTA, K.; KONDO, S.; 1998. Adaptive Geno-Fuzzy Control of Process Plants via Perpetual Evolution. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Trieste, Itália). *Proceedings*. p. 797-801.
- RAJU, G. V. S., ZHOU, J., KISNER, R. A.; 1991. Hierarchical Fuzzy Control. *International Journal of Control*, v. 54, n. 5, p. 1201-1216.
- RALESCU, A. L. (1996). Fuzzy Engineering and its Social Impact — The Symbolic Project. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 95-100.
- RAMÍREZ, D. R.; LIMÓN, D.; GÓMEZ-ORTEGA, J.; CAMACHO, E. F.; 1999. Nonlinear MBPC for Mobile Robot Navigation Using Genetic Algorithms. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS & AUTOMATION (Detroit, Michigan, USA). *Proceedings*. p. 2452-2457.
- RANI, K. Y.; UNBEHAUEN, H.; 1997. Study of Predictive Controller Tuning Methods. *Automatica*, v. 33, n. 12, p. 2243-2248.
- RAO, D. H.; GUPTA, M. M.; 1993. Dynamic Neural Units and Function Approximation. In: IEEE CONFERENCE ON NEURAL NETWORKS (San Francisco, CA, USA). *Proceedings*. p. 743-748
- RECHENBERG, I.; 1973. Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution. Frommann-Holzboog, Stuttgart, Germany.
- REDGERS, A., ALEKSANDER, I.; 1995. Digital Neural Networks, In: IRWIN, G. W.; WARWICK, K.; HUNT, K. J. (eds.). *Neural Network Applications in Control*, IEE Control Engineering Series 53, The Institution of Electrical Engineers. Chapter 2, p. 17-32.
- REED, R.; 1993. Pruning Algorithms — A Survey. *IEEE Transactions on Neural Networks*, v. 4, n. 5, p. 740-747.
- REZNIK, L.; 1997. Fuzzy Controller Design: Methods Classification and Parameters Choice. In: 7TH INTERNATIONAL FUZZY SYSTEMS ASSOCIATION WORLD CONGRESS, IFSA'97 (Prague, Czech Republic). *Proceedings*. v. III, p. 398-403.
- RIEDMILLER, M.; BRAUN, H.; 1993. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (San Francisco, CA, USA). *Proceedings*. v. 1, p. 586-591.
- RIEDMILLER, M.; 1994. *Rprop — Description and Implementation Details*. Karlsruhe. Technical Report, Institut für Logik, Komplexität und Deduktionssysteme, University of Karlsruhe, Germany.

- RODRIGUES, G. G. (1996). *Identificação de Sistemas Dinâmicos Não-Lineares Utilizando Modelos NARMAX Polinomiais — Aplicação a Sistemas Reais*. Belo Horizonte, MG. Dissertação (Mestrado em Engenharia Elétrica). Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, UFMG.
- RONCO, E.; GAWTHROP, P. J.; 1997. Neural Networks for Modelling and Control. Glasgow, UK. Technical Report CSC97008, Centre for System and Control, Department of Mechanical Engineering, University of Glasgow.
- RUDOLPH, G.; 1992. On Correlated Mutations in Evolution Strategies. In: MÄNNER, R., MANDERICK, B. (eds.), 2ND INTERNATIONAL CONFERENCE ON PARALLEL SOLVING FROM NATURE (Brussels : Belgium). *Proceedings*. p. 105-114.
- _____; 1994. An Evolutionary Algorithm for Integer Programming. In: DAVIDOR, Y.; SCHWEFEL H. -P.; MÄNNER, R. (eds.), PARALLEL PROBLEM SOLVING FROM NATURE 3. *Proceedings*. Springer : Berlin. p. 139-148.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J.; 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. v. 1, The MIT Press : Cambridge, MA, USA.
- RUTHERFORD, D. A.; BLOORE, G. C.; 1976. The Implementation of Fuzzy Algorithms for Control. *Proceedings of the IEEE*, April, p. 572-573.
- SAHA, S.; CHRISTENSEN, J. P.; 1994. Genetic Design of Sparse Feedforward Neural Networks. *Information Sciences*, v. 79, p. 191-200.
- SAKAWA, M.; KATO, K.; MISAKA, M.; USHIRO, S.; 1995. Cooling Load Prediction through Recurrent Neural Networks. In: FUZZ-IEEE/IFES (Yokohama, Japan). *Proceedings*. v. I, p. 421-426.
- SAINT-DONAT, J.; BHAT, N.; MCAVOY, T. J.; 1991. Neural Net Based Model Predictive Control. *International Journal of Control*, v. 54, n. 6, p. 1453-1468.
- SALOMON, R., 1996. Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions. *Biosystems*, v. 39, n. 3, p. 263-278.
- SAMAD, T.; 1995. Neurocontrol: Concepts and Practical Considerations. In: GUPTA, M. M., SINHA, N. K. (eds.). *Intelligent Control Systems: Theory and Applications*. Chapter 11, IEEE Press : Piscataway, NJ, USA. p. 265-291.
- SANDRI, S.; CORREA, C.; 1999. *Lógica Nebulosa*. In: V ESCOLA DE REDES NEURAIIS (ITA, São José dos Campos, SP). p. 73-90.
- SANNER, R. M.; SLOTINE, J.-J. E.; 1992. Gaussian Networks for Direct Adaptive Control. *IEEE Transactions on Neural Networks*, v. 3, n. 6, p. 837-863.
- SARAIVA, E. T. S. (1999). *Identificação de uma Planta Piloto de Flotação em Coluna*. Belo Horizonte, MG. Dissertação (Mestrado em Engenharia Elétrica). Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, UFMG.

- SARAVANAN, N.; FOGEL, D. B.; 1996. An Empirical Comparison of Methods for Correlated Mutations under Self-Adaptation. In: FOGEL, L. J.; ANGELINE, P. J.; BÄCK, T. (eds.), 5TH ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING, EVOLUTIONARY PROGRAMMING V. MIT Press, Cambridge, Massachusetts, London, England, p. 479-485.
- SCHAIBLE, B.; XIE, H.; LEE, Y. C.; 1997. Fuzzy Logic Models for Ranking Process Effects. *IEEE Transactions on Fuzzy Systems*, v. 5, n. 4, p. 545-556.
- SCHAFFER, J. D.; 1994. Combinations of Genetic Algorithms with Neural Networks or Fuzzy Systems. In: ZURADA, J. M.; MARKS II, R. J.; ROBINSON, C. J. (eds.). *Computational Intelligence: Imitating Life*, IEEE Press : Piscataway, NJ, p. 371-382.
- SCHIFFMANN, W.; JOOST, M.; WERNER, R.; 1992. Synthesis and Performance Analysis of Multilayer Neural Network Architectures. Technical Report 16/1992, University of Koblenz, Institute für Physics.
- SCHWARTZ, D. G.; KLIR, G. J.; LEWIS III, H. W.; EZAWA; 1994. Applications of Fuzzy Sets and Approximate Reasoning. *Proceedings of the IEEE*, v. 82, n. 4, p. 482-497.
- SCHWEFEL, H. -P.; 1981. *Numerical Optimization of Computer Models*. John Wiley & Sons : NY, USA.
- _____; 1997. Advantages (and Disadvantages) of Evolutionary Computation over Other Approaches. In: BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z.; 1997a.
- SENG, T. L., KHALID, M. B., YUSOF, R.; 1999. Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 29, n. 2, p. 226-236.
- SHAH, S. L.; MOHTADI, C.; CLARKE, D. W.; 1987. Multivariable Adaptive Control without a Prior Knowledge Forms of Delay Matrix. *Systems & Control Letters*, v. 9, p. 295-306.
- SHARAF, A. M.; LIE, T. T.; 1994. A Hybrid Neuro-Fuzzy Power System Stabilizer. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (Orlando, FL, USA). *Proceedings*. v. III, p. 1760-1765.
- SHARMAN, K. C.; ALCÁZAR, A. I. E.; LI, Y.; 1995. Evolving Signal Processing Algorithms by Genetic Programming. In: 1ST INTERNACIONAL CONFERENCE ON GENETIC ALGORITHMS IN ENGINEERING SYSTEMS: INNOVATIONS AND APPLICATIONS, GALESIA (Sheffield, UK). *Proceedings*. p. 473-480.
- SHOURESHI, R. A.; 1993. Intelligent Control Systems: Are They for Real?. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, v. 115, p. 392-401.
- SHOURESHI, R. A.; BAHETI, K.; GARG, D.; MAULBETSCH, J.; ALBUS, J.; WIDROW, B.; SAMAD, T.; ANTSAKLIS, P.; MEYSTEL, A.; FURUTA, K.; 1996. Intelligent

- Control Systems. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 101-106.
- SILVA, A. C.; LUIZ, C. C.; COELHO, L. S.; COELHO, A. A. R.; 1997. Ambiente Computacional Integrado de Ensino para Supervisão, Identificação e Controle de Processos. In: XXV Congresso Brasileiro de Ensino de Engenharia (Salvador, BA). *Anais*. v. 1, p. 328-341.
- SIMAS, H.; COELHO, L. S.; BOUZON, G.; SANTOS, J. E.; COELHO, A. A. R.; 1998. Three Low-Cost Laboratory Models: Design Issues and Experiments with Advanced Control. In: 8º CONGRESSO E EXPOSIÇÃO INTERNACIONAL DE AUTOMAÇÃO, CONAI (São Paulo, SP). *Anais*. p. 156-162.
- SJÖBERG, J.; 1995. Non-Linear System Identification with Neural Networks. Linköping, Sweden. *PhD. Thesis*, Department of Electrical Engineering, Linköping University.
- SJÖBERG, J.; LJUNG, L.; 1995. Overtraining, Regularization and Searching for a Minimum with Application to Neural Networks. *International Journal of Control*, v. 62, p. 1391-1407.
- SJÖBERG, J.; ZHANG, Q.; LJUNG, L.; BENVENISTE, A.; DELYON, B.; GLORENNEC, P. -Y.; HJALMARSSON, H.; JUDITSKY, A.; 1995. Nonlinear Black-Box Modeling in System Identification: a Unified Overview. *Automatica*, v. 31, n. 12, p. 1691-1724.
- SKRJANC, I.; MATKO, D.; 1994. Fuzzy Predictive Controller with Adaptive Gain. In: CLARKE, D. (ed.). *Advances in Model-Based Predictive Control*, Oxford University Press : New York.
- SLOTINE, J. -J.; LI, W.; 1991. *Applied Nonlinear Control*, Prentice-Hall : Englewood Cliffs, NJ, USA.
- SONG, Q.; GRIMBLE, M. J.; 1997. Design of a Multivariable Neural Controller and Its Application to Gas Turbines. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, v. 119, p. 565-567.
- SOUCEK, B.; THE IRIS GROUP (eds.); 1992. *Dynamic, Genetic, and Chaotic Programming: the Sixth Generation*. John Wiley & Sons : New York, NY, USA.
- SOUZA, J. M.; BABUSKA, R.; VERBRUGGEN, H. B.; 1997. Fuzzy Predictive Control Applied to an Air-Conditioning System. *Control Engineering Practice*, v. 5, n. 10, p. 1395-1406.
- SARPTÜRK, S. Z.; ISTEFAPOULOS, Y.; KAYNAK, O.; 1987. On the Stability of Discrete-Time Sliding Mode Control Systems. *IEEE Transactions on Automatic Control*, v. 32, n. 10, p. 930-932.
- SETIONO, R.; HUI, L. C. K.; 1995. Use of a Quasi-Newton Method in a Feedforward Neural Network Construction Algorithm. *IEEE Transactions on Neural Networks*, v. 6, n. 1, p. 273-277.

- SPEARS, W. M.; DE JONG, K. A.; BÄCK, T.; FOGEL, D. B.; DE GARIS, H.; 1993. An Overview of Evolutionary Computation. In: EUROPEAN CONFERENCE ON MACHINE LEARNING. *Proceedings*. v. 667, Springer-Verlag. p. 442-459.
- SPONG, M. W.; 1995. The Swing Up Control Problem for the Acrobot. *IEEE Control Systems*, v. 15, n. 1, p. 49-55.
- SPURGEON, S. K.; 1992. Hyperplane Design Techniques for Discrete-Time Variable Structure Control Systems. *International Journal of Control*, v. 55, n. 2, p. 445-456.
- SRINIVAS, M.; PATNAIK, L. M.; 1994a. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 24, no. 4, p. 656-667.
- _____; 1994b. Genetic Algorithms: A Survey. *IEEE Computer*, v. 27, n. 6, p. 17-26.
- STECK, J. E.; ROKHSAZ, K.; SHUE, S. P.; 1996. Linear and Neural Network Feedback for Flight Control Decoupling. *IEEE Control Systems*, v. 16, n. 4, p.22-30.
- STEMMER, M. R.; DE PIERI, E. R.; BORGES, F. A. P.; 1997. Comparação de Performance entre Controladores Clássicos e um Controlador Torque Computado Neural Aplicados ao Robô PUMA 560. In: CALOBA, L.; BARRETO, J. M. (eds.), *III Congresso Brasileiro de Redes Neurais* (Florianópolis, SC). *Anais*. p. 390-395.
- STENZ, R.; KUHN, U.; 1995. Automation of a Batch Distillation Column Using Fuzzy and Conventional Control. *IEEE Transactions on Control Systems Technology*, v. 3, n. 2, p. 171-176.
- STREIFEL, R. J.; MARKS II, R. J.; REED, R.; CHOI, J. J.; HEALY, M.; 1999. Dynamic Fuzzy Control of Genetic Algorithm Parameter Coding. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 29, n. 3, p. 426-433.
- SU, C. -Y.; LEUNG, T. -P.; STEPANENKO, Y.; 1993. Real-Time Implementation of Regressor-Based Sliding Mode Control Algorithm for Robotic Manipulators. *IEEE Transactions on Industrial Electronics*, v. 40, n. 1, p. 71-79.
- SUGENO, M., KANG, G. T.; 1986. Fuzzy Modeling and Control of Multilayer Incinerator. *Fuzzy Sets and Systems*, v. 18, p. 329-346.
- _____; 1988. Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*, v. 28, p. 15-33.
- SUN, C. T.; JANG, S. J.; 1991. Fuzzy Modelling Based on Generalized Neural Networks and Fuzzy Clustering Objective Functions. In: 30TH IEEE CONFERENCE ON DECISION AND CONTROL (England, UK). *Proceedings*. p. 2924-2929.
- SUN, R. -L.; 1995. *Evolving Population-Based Search Algorithms through Thermodynamic Operation: Dynamic System Design and Integration*. Maryland, USA. Ph.D. thesis, Institute for Systems Research, University of Maryland.

- SUNDARESHAN, M. K.; ASKEW, C.; 1997. Neural Network-Assisted Variable Structure Control Scheme for Control of a Flexible Manipulator Arm. *Automatica*, v. 33, n. 9, p. 1699-1710.
- SUYKENS, J.; DE MOOR, B.; VANDERWALLE, J.; 1995. NL_q Theory: Unifications in the Theory of Neural Networks, Systems and Control. In: EUROPEAN SYMPOSIUM ARTIFICIAL NEURAL NETWORKS (Brussels, Belgium). *Proceedings*. p. 271-276.
- SWANSTON, D. J.; BISHOP, J. M.; MITCHELL, R. J.; 1994. Simple Adaptive Momentum: New Algorithm for Training Multilayer Perceptrons. *IEE Electronics Letters*, v. 30, n. 18, p. 1498-1500.
- TAKAGI, H.; HAYASHI, I.; 1991. NN-Driven Fuzzy Reasoning. *International Journal of Approximate Reasoning*, v. 3, p. 191-212.
- TAKAGI, T.; SUGENO, M.; 1985. Fuzzy Identification of Systems and its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, v. 15, n. 1, p. 116-132.
- TAKATSU, H.; ITOH, T.; 1999. Future Needs for Control Theory in Industry — Report in the Control Technology Survey in Japanese Industry. *IEEE Transactions on Control Systems Technology*, v. 7, n. 3, p. 298-305.
- TAN, K. C.; GONG, M. R.; LI, Y.; 1996. Evolutionary Linearisation in the Frequency Domain. *IEE Electronics Letters*, v. 32, n. 1, p. 74-76.
- TAN, K. C.; LI, Y.; MURRAY-SMITH, D. J.; SHARMAN, K. C.; 1995. System Identification and Linearisation Using Genetic Algorithms with Simulated Annealing. In: IEE/IEEE INTERNATIONAL CONFERENCE ON GA IN ENGINEERING SYSTEMS: INNOVATIONS AND APPLICATIONS (Sheffield, U.K.). *Proceedings*. p. 164-169.
- TANAKA, K.; IKEDA, T.; WANG, H. O.; 1996. Robust Stabilization of a Class of Uncertain Nonlinear Systems via Fuzzy Control: Quadratic Stability, H^∞ Control Theory, and Linear Matrix Inequalities. *IEEE Transactions on Fuzzy Systems*, v. 4, n. 1, p. 1-13.
- TANG, C. Y.; 1997. *Discrete Variable Structure Control for Uncertain Linear Multivariable Systems*. Stillwater, OK, USA. Master of Science thesis, School of Mechanical & Aerospace Engineering, Oklahoma State University.
- TANG, C. Y.; MISAWA, E. A.; 1998. Discrete Variable Structure Control for Linear Multivariable Systems: The State Feedback Case. In: AMERICAN CONTROL CONFERENCE (Philadelphia, Pennsylvania, USA). *Proceedings*. p. 114-118.
- TANI, T.; MURAKOSHI, S.; UMANO, M.; 1996. Neuro-Fuzzy Hybrid Control System of Tank Level in Petroleum Plant. *IEEE Transactions on Fuzzy Systems*, v. 4, n. 3, p. 360-368.

- TANSCHKEIT, R.; BARRETO, J. M.; 1978. Controle de Misturador de Fluidos Baseado na Matemática Nebulosa. In: II CONGRESSO BRASILEIRO DE AUTOMÁTICA (Florianópolis, SC). *Anais*. p. 763-770.
- TING, T. C.; WOLFE, P. M.; 1995. The Amalgamation of Neural Networks and Fuzzy Logic Systems — A Survey. *Computers on Industrial Engineering*, v. 29, n. 1, p. 193-197.
- TOKARNIA, A. M. B. H.; 1984. *Identificação Descentralizada de Sistemas Multivariáveis*. Campinas, SP. Dissertação (Mestrado em Engenharia Elétrica), DCA/FEE, Universidade de Campinas, UNICAMP.
- THOMSON, M.; SCHOOLING, S. P.; SOUFIAN, M.; 1996. The Practical Application of a Nonlinear Identification Methodology. *Control Engineering Practice*, v. 4, n. 3, p. 295-306.
- TOMASSINI, M.; 1995. A Survey of Genetic Algorithms. In: *Annual Reviews of Computational Physics*, v. III, World Scientific, p. 87-117.
- TOVAR, S. N.; 1996. El Control Avanzado en los Procesos Industriales. *Boletín Informativo de la Asociación Colombiana de Automática*, n. 3, Agosto, p. 2-3.
- TRAJANOSKI, Z.; REGITTNIG, WACH, P.; 1997. Neural Predictive Controller for Closed-Loop Control of Glucose Using the Subcutaneous Route: A Simulation Study. *Control Engineering Practice*, v. 5, n. 12, p. 1727-1730.
- TSAI, C. C.; LU, C. H.; 1998. Multivariable Self-Tuning Temperature Control for Plastic Injection Molding Process. *IEEE Transactions on Industry Applications*, v. 34, n. 2, p. 310-318.
- TZES, A.; PENG, P. -Y.; GUTHY, J.; 1998. Genetic-Based Fuzzy Clustering for DC-Motor Friction Identification and Compensation. *IEEE Transactions on Control Systems Technology*, v. 6, n. 4, p. 462-472.
- TSUKAMOTO, Y.; 1979. An Approach to Fuzzy Reasoning Method, In: GUPTA, M. M. RAGADE, R. K. YAGER, R. R. (eds.). *Advances in Fuzzy Set Theory and Applications*, North-Holland : New York.
- UNBENHAUEN (1996). Distributed Supervisory Industrial Control Systems and their Problems for Embedding Advanced Control Strategies. In: 7º CONGRESO LATINOAMERICANO DE CONTROL AUTOMÁTICO E XV SIMPOSIO NACIONAL DE CONTROL AUTOMÁTICO (Buenos Aires, Argentina). v. 1, p. I-VIII.
- VAN CAN, H. J. L.; BRAAKE, H. A. B.; HELLINGA, C.; KRIJGSMAN, A. J.; VERBRUGGEN, H. B.; LUYBEN, K. C. A. M.; HEIJNEN, J. J.; 1995. Design and Real Time Testing of a Neural Model Predictive Controller for a Nonlinear System. *Chemical Engineering Science*, v. 50, n. 15, p. 2419-2430.

- VAN DEN HOF, P. M. J.; SCHRAMA, R. J. P.; 1995. Identification and Control — Closed-Loop Issues. *Automatica*, v. 31, p. 1751-1770.
- VAN DOREN, V. J.; 1995. Neural Networks for Process Modeling and Control (Software Review). *Control Engineering International*, v. 42, p. 62.
- _____; 1996. Multivariable Controllers Balance Performance with Cost. *Control Engineering International*, v. 43, n. 17, p. 59-60.
- _____; 1997. Multivariable Controllers Enter the Mainstream. *Control Engineering International*, v. 44, n. 4, p. 107-112.
- VASCONCELOS, J. A., SALDANHA, R. R., KRÄHENBÜHL, L., NICOLAS, A.; 1999. Genetic Algorithm Coupled with a Deterministic Method for Optimization in Electromagnetics. *IEEE Transactions on Magnetics*, v. 33, n. 3, pp. 1860-1863.
- VILJAMAA, P.; KOIVO, H. N.; 1996. Multivariable Fuzzy Logic Control of Pilot Head-Box Process. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 379-384.
- VILLAMIL, F. D. V.; RIVERA, D. E.; 1999. Adaptive Model Predictive Control for Production Optimization and Inventory Control of Semiconductor Reentrant Manufacturing Lines. In: AMERICAN CONTROL CONFERENCE (San Diego, CA, USA). *Proceedings*. p. 4087-4091.
- VOGL, T. P.; MANGIS, J. K.; RIGLER, A. K.; ZINK, W. T.; ALKON, D. L.; 1988. Accelerating the Convergence of the Back-Propagation Method. *Biological Cybernetics*, v. 59, n. 4, p. 257-263.
- WANG, D.; CHAI, T.; 1994. Multivariable Adaptive Control of Unknown Nonlinear Dynamic Systems Using Neural Networks. In: 33TH IEEE CONFERENCE ON DECISION AND CONTROL (Lake Buena Vista, FL, USA). *Proceedings*. p. 2500-2505.
- WANG, L. -X.; 1994. *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*. Prentice-Hall : Englewood Cliffs, NJ, USA.
- _____; 1999. Analysis and Design of Hierarchical Fuzzy Systems. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 5, p. 617-624.
- WANG, L.; LANGARI, R.; 1995. Building Sugeno-Type Models Using Fuzzy Discretization and Orthogonal Parameter Estimation Techniques. *IEEE Transactions on Fuzzy Systems*, v. 3, n. 4, p.454-458.
- WANG, L.; MENDEL, J. M.; 1992. Generating Fuzzy Rules by Learning from Examples. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, n. 6, p. 1414-1427.
- WARWICK, K.; 1995. Neural Networks: An Introduction, In: IRWIN, G. W., WARWICK, K., HUNT, K. J. (eds.). *Neural Network Applications in Control*, IEE Control Engineering Series 53, The Institution of Electrical Engineers. Chapter 1, p. 1-16.

- _____; 1996. An Overview of Neural Networks in Control Applications, In: ZALZALA, A. M. S.; MORRIS, A. S. (eds). *Neural Networks for Robotic Control: Theory and Applications*. Chapter 1, Ellis Horwood Limited, UK, p. 1-25.
- WASSERMAN, P. D.; 1993. *Advanced Methods in Neural Computing*. Van Nostrand Reinhold : New York, NY, USA.
- WATANABE, N.; IMAIZUMI, T.; 1997. On Least Squares Methods in Fuzzy Modeling. In: 7TH INTERNATIONAL FUZZY SYSTEMS ASSOCIATION WORLD CONGRESS, IFSA'97 (Prague, Czech Republic), *Proceedings*. v. II, p. 336-341.
- WELLSTEAD, P. E.; ZARROP, M. B.; 1991. *Self-Tuning Systems: Control and Signal Processing*. John Wiley Sons : New York, NY, USA.
- WHITLEY, D.; GORDON, V. S.; MATHIAS, K.; 1994. Lamarck Evolution, The Baldwin Effect and Function Optimization. In: DAVIDOR, Y., SCHWEFEL, H. -P.; MÄNNER, R. (eds.), *PARALLEL PROBLEM SOLVING FROM NATURE*. Lecture Notes in Computer Science, v. 866, Springer-Verlag : Berlin, p. 6-15.
- WIDROW, B.; LEHR, M. A.; 1990. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE*, v. 78, p. 1415-1442.
- WIDROW, B.; RUMELHART, D. E.; LEHR, M. A.; 1994. Neural Networks: Applications in Industry, Business and Science. *Communications of the ACM*, v. 37, n. 3, p. 93-105.
- WILDBERGER, A. M.; 1997. Complex Adaptive Systems: Concepts and Power Industry Applications. *IEEE Control Systems*, v. 17, n. 6, p. 77-88.
- WILLIAMS, R. J.; ZIPSER, D.; 1989. A Learning Algorithm for Continually Fully Recurrent Neural Networks. *Neural Computing*, v. 1, n. 2, p. 270-280.
- WILLIS, M. J.; MONTAGUE, G. A.; MASSIMO, C. D.; THAM, M. T.; MORRIS, A. J.; 1992. Artificial Neural Networks in Process Estimation and Control. *Automatica*, v. 28, n. 6, p. 1181-1187.
- WILLIS, M. J.; THAM, M. T.; 1994. Advanced Process Control. Newcastle, UK. Technical Report, Department of Chemical and Process Engineering, University of Newcastle upon Tyne, UK.
- WILSON, W. H.; 1995. Stability of Learning in Classes of Recurrent and Feedforward Networks. In: AUSTRALIAN CONFERENCE ON NEURAL NETWORKS. *Proceedings*. p. 142-145.
- WHITEHEAD, B. A.; 1996. Genetic Evolution of Radial Basis Function Coverage Using Orthogonal Niches. *IEEE Transactions on Neural Networks*, v. 7, n. 6, p. 1525-1528.
- WOLPERT, D. H.; MACREADY, W. G.; 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 67-82.

- WU, J. C.; LIU, T. S.; 1996. Fuzzy Control Stabilization with Applications to Motorcycle Control. *IEEE Transactions on Systems, Man and Cybernetics — Part B: Cybernetics*, v. 26, n. 6, p. 836-847.
- XIAOHONG, C.; FENG, G.; JIXIN, Q.; YOUXIAN, S.; 1996. A Nonlinear Adaptive Controller Based on RBF Networks. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS (Beijing, China). *Proceedings*. p. 661-666.
- XIONG, N.; LITZ, L.; 1998. A New Genetic Based Approach to Fuzzy Controller Design and its Application. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS (Trieste, Itália). *Proceedings*. p. 937-941.
- XU, C. W.; 1991. Linguistic Decoupling Control of Fuzzy Multivariable Processes. *Fuzzy Sets and Systems*, v. 44, p. 209-217.
- YAGER, R. R.; FILEV, D. P.; 1994. *Essentials of Fuzzy Modeling and Control*. John Wiley & Sons : New York, NY, USA.
- YANG, R.; DOUGLAS, I.; 1998. Simple Genetic Algorithm with Local Tuning Efficient Global Optimizing Technique. *Journal of Optimization Theory and Applications*, v. 98, n. 2, p. 449-465.
- YAO, X.; 1999. Evolving Artificial Neural Network. *Proceedings of the IEEE*, v. 87, n. 9, p. 1423-1447.
- YAO, X.; LIU, Y.; 1996a. Fast Evolutionary Programming. In: 5TH ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING (San Diego, CA, USA). *Proceedings*. The MIT Press, p. 451-460.
- _____; 1996b. Towards Designing Artificial Neural Networks by Evolution. In: INTERNATIONAL SYMPOSIUM ON ARTIFICIAL LIFE AND ROBOTICS (Beppu, Oita, Japan). *Proceedings*. p. 265-268.
- YEN, J.; 1999. Fuzzy Logic – A Modern Perspective. *IEEE Transactions on Knowledge and Data Engineering*, v. 1, n. 1.
- YEN, J.; LIAO, C. J.; LEE, B.; RANDOLPH, D.; 1998. A Hybrid Approach to Modelling Metabolic Systems Using a Genetic Algorithm and Simplex Method. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 28, n. 2, p. 173-191.
- YEN, J.; WANG, L.; 1999. Simplifying Fuzzy Rule-Based Models Using Orthogonal Transformation Methods. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, v. 29, n. 1, p. 13-24.
- YING, H.; 1998. General SISO Takagi-Sugeno Fuzzy Systems with Linear Rule Consequent are Universal Approximators. *IEEE Transactions on Fuzzy Systems*, v. 6, n. 4, p. 582-587.
- ZADEH, L. A.; 1965. Fuzzy Sets. *Information and Control*, v. 8, p. 338-353.

- _____; 1973. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 3, p. 28-44.
- _____; 1994a. Why the Success of Fuzzy Logic is not Paradoxical. *IEEE Expert*, v. 9, n. 4, p. 43-46.
- _____; 1994b. Soft Computing and Fuzzy Logic. *IEEE Software*, v. 11, n. 6, p. 48-56.
- _____; 1996. Fuzzy Control: Issues, Contentions and Perspectives. In: 13TH IFAC WORLD CONGRESS (San Francisco, CA, USA). *Proceedings*. p. 35-38.
- ZENG, X. -J.; SINGH, M. G.; 1995. Approximation Theory of Fuzzy Systems — MIMO Case. *IEEE Transactions on Fuzzy Systems*, v. 3, n. 2, p. 219-235.
- ZBIKOWSKI, R. W. (1994). *Recurrent Neural Networks: Some Control Aspects*. Glasgow, UK. *PhD Thesis*, Faculty of Engineering, University of Glasgow.
- ZHIHONG, M.; YU, X. H.; 1997. Terminal Sliding Mode Control of MIMO Linear Systems. *IEEE Transactions on Circuits and Systems — I: Fundamental Theory and Applications*, v. 44, n. 11, p. 1065-1070.
- ZHU, Q. M.; BILLINGS, S. A.; 1996. Fast Orthogonal Identification of Nonlinear Stochastic Models and Radial Function Neural Networks. *International Journal of Control*, v. 64, n. 5, p. 871-886.
- ZUO, W. (1995). Multivariable Adaptive Control for a Space Station using Genetic Algorithms. *IEE Proceedings Control Theory and Applications*, v. 142, n. 2, p. 81-87.