



UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

**SINTONIA AUTOMÁTICA DE
CONTROLADORES PID: UMA ABORDAGEM
NEURO-GENÉTICA**

DOUTORAMENTO EM ENGENHARIA ELECTRÓNICA E COMPUTAÇÃO
NA ESPECIALIDADE DE SISTEMAS DE CONTROLO

JOÃO MIGUEL GAGO PONTES DE BRITO LIMA

FARO

Julho de 2004



JOÃO MIGUEL GAGO PONTES DE BRITO LIMA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

Orientador: PROF. DOUTOR ANTÓNIO EDUARDO DE BARROS RUANO

Data: Julho 2004

**SINTONIA AUTOMÁTICA DE CONTROLADORES PID:
UMA ABORDAGEM NEURO-GENÉTICA**

3413T.

14 12 04: 60009

⁰⁰⁷
LIM * Ssn

1

ÍNDICE GERAL

1. INTRODUÇÃO	1
1.1 VISÃO GERAL	1
1.2 PRINCIPAIS OBJECTIVOS E ESTRUTURA DA DISSERTAÇÃO	1
1.3 O CONTROLADOR <i>PID</i>	5
1.3.1 Acção Proporcional	7
1.3.2 Acção Integral	8
1.3.3 Acção Derivativa	9
1.3.4 Implementação digital do controlador <i>PID</i>	10
1.4 ESTADO DA ARTE DE MÉTODOS DE SINTONIA AUTOMÁTICA	13
1.4.1 Técnicas de Ziegler e Nichols	15
1.4.1.1 Técnicas de Ziegler e Nichols baseadas na resposta ao degrau	16
1.4.1.2 Técnicas de Ziegler e Nichols baseadas na resposta em frequência	16
1.4.2 Novas técnicas de sintonia do controlador <i>PID</i>	17
1.4.2.1 Técnicas de sintonia baseadas na resposta ao degrau	19
1.4.2.2 Técnicas de sintonia no domínio da frequência	20
1.4.3 Técnicas de sintonia utilizando optimização	23
1.4.4 Técnicas de sintonia baseadas em modelo interno	24
1.4.5 Técnica de Cohen-Coon	26
1.5 DEFINIÇÃO DE SINTONIA AUTOMÁTICA	27
1.5.1 Sintonia automática do controlador <i>PID</i> com base na resposta em frequência	28
1.5.2 Auto-sintonia do controlador <i>PID</i> usando técnicas de <i>soft-computing</i>	31
1.5.2.1 Auto-sintonia do controlador <i>PID</i> usando Algoritmos Evolutivos	31
1.5.2.2 Auto-sintonia do controlador <i>PID</i> usando lógica difusa	36

1.5.2.3	Auto-sintonia do controlador <i>PID</i> usando redes neuronais-----	40
1.6	CONCLUSÕES -----	43
2.	FUNDAMENTOS DE TÉCNICAS DE <i>SOFTCOMPUTING</i> -----	45
2.1	INTRODUÇÃO -----	45
2.2	REDES NEURONAIS -----	45
2.2.1	Inspiração biológica das redes neuronais artificiais-----	49
2.2.1.1	O neurónio -----	50
2.2.1.2	As sinapses -----	51
2.2.2	Redes neuronais do tipo <i>MLP</i> -----	54
2.2.2.1	Algoritmo de aprendizagem <i>backpropagation</i> -----	58
2.2.2.2	Algoritmos de treino de 2ª ordem -----	62
2.2.3	Redes neuronais do tipo <i>RBF</i> -----	64
2.2.3.1	Diferentes heurísticas para a determinação dos raios -----	69
2.2.4	Redes neuronais do tipo <i>B-Spline</i> -----	71
2.2.4.1	Algoritmo de aprendizagem para redes neuronais do tipo <i>B-Spline</i> -----	77
2.2.4.2	Treino de redes neuronais do tipo <i>B-Spline</i> usando programação genética	81
2.3	ALGORITMOS EVOLUTIVOS-----	82
2.3.1	Surgimento dos algoritmos genéticos-----	82
2.3.2	Noções de algoritmos genéticos -----	83
2.3.3	Operações básicas dos algoritmos genéticos-----	87
2.3.4	Programação genética -----	90
2.4	CONCLUSÕES -----	91
3.	ARQUITECTURA ADAPTATIVA PARA AUTO-SINTONIA DO CONTROLADOR <i>PID</i> -----	93

3.1	INTRODUÇÃO	93
3.2	SISTEMA DE CONTROLO	94
3.2.1	Identificação de sistemas	94
3.3	MECANISMO DE SINTONIA DO SISTEMA DE CONTROLO	97
3.3.1	Ciclo de funcionamento do sistema adaptativo	101
3.4	OPTIMIZAÇÃO GENÉTICA DO BLOCO DE SINTONIA	103
3.4.1	Função objectivo	104
3.4.2	População inicial	106
3.4.3	Estrutura do procedimento de optimização	107
3.4.4	Comportamento do sistema face às variações das ponderações	112
3.4.5	Diferentes tipos de sintonia	113
3.5	CONCLUSÕES	116
4.	SELECÇÃO DO TIPO DE REDE NEURONAL	117
4.1	INTRODUÇÃO	117
4.2	CONJUNTOS DE APRENDIZAGEM	118
4.2.1	Normalização dos padrões de entrada	121
4.3	MODELOS NEURONAIS UTILIZANDO <i>MLPs</i>	123
4.3.1	Actualização de pesos padrão a padrão de <i>MLPs</i>	126
4.4	MODELOS NEURONAIS UTILIZANDO <i>RBFs</i>	128
4.4.1	Plano de aprendizagem	129
4.4.2	Diferentes heurísticas para a determinação dos raios	132
4.4.3	Actualização de pesos padrão a padrão de <i>RBFs</i>	136
4.5	MODELOS NEURONAIS UTILIZANDO <i>B-SPLINES</i>	137

4.5.1	Estruturação da aprendizagem -----	137
4.5.2	Modelo com boa capacidade de generalização -----	139
4.5.3	Modelo para o conjunto de treino -----	141
4.5.4	Actualização de pesos padrão a padrão de <i>B-Splines</i> -----	145
4.6	ANÁLISE COMPARATIVA DE RESULTADOS -----	147
4.7	CONCLUSÕES -----	149
5.	CONJUNTOS DE APRENDIZAGEM PARA MODELAÇÃO DE PROCESSOS	
	VARIANTES NO TEMPO -----	151
5.1	INTRODUÇÃO -----	151
5.2	CONJUNTO BASE DE SISTEMAS DE EXPERIMENTAÇÃO -----	152
5.2.1	Sistema de controlo de nível -----	152
5.2.2	Formas gerais de sistemas a controlar -----	153
5.2.3	Determinação dos valores de σ -----	155
5.2.3.1	Linearização das medidas de identificação -----	156
5.2.4	Distribuição no espaço das medidas de identificação-----	161
5.3	PADRÕES DE APRENDIZAGEM -----	165
5.3.1	Parâmetros <i>PID</i> mínimos -----	166
5.3.2	Parâmetros <i>PID</i> representativos do espaço de aprendizagem -----	168
5.3.2.1	Distribuição prismática hexagonal recta -----	168
5.3.2.2	Distribuição esférica -----	169
5.3.3	Estabelecimento dos conjuntos de aprendizagem -----	174
5.3.3.1	Conjuntos de aprendizagem para o mapeamento no critério -----	177
5.4	CONCLUSÕES -----	179

6. APRENDIZAGEM DE REDES NEURONAIS PARA PROCESSOS	
VARIANTES NO TEMPO -----	181
6.1 INTRODUÇÃO -----	181
6.2 REDES NEURONAIS PARA MAPEAMENTO DO <i>ITAE</i> -----	182
6.2.1 Determinação de uma estrutura comum -----	183
6.2.2 Modelo neuronal para um único elemento do conjunto base de experimentação -----	185
6.2.3 Modelo com estrutura parcialmente fixa -----	188
6.2.4 Modelo de 2ª ordem para aproximação nas vizinhanças do mínimo -----	191
6.2.5 Redes neurais para avaliação dos parâmetros dos modelos de 2ª ordem --	198
6.2.6 Aprendizagem de <i>B-Splines</i> utilizando <i>ASM</i> -----	199
6.2.7 Aprendizagem de <i>B-Splines</i> utilizando programação genética-----	201
6.2.8 Escolha duma estratégia de aprendizagem -----	206
6.2.9 Aprendizagem de redes neurais do tipo <i>B-Spline</i> para modelar os 10 parâmetros-----	208
6.2.10 Validação dos modelos para o conjunto base de experimentação -----	213
6.3 REDES NEURONAIS PARA MAPEAMENTO DO TEMPO DE CRESCIMENTO-----	216
6.4 REDES NEURONAIS DE SINTONIA-----	219
6.4.1 Conjuntos de aprendizagem para os mapeamentos de sintonia -----	220
6.4.2 Redes neurais de sintonia óptima para o <i>ITAE</i> -----	221
6.4.3 Redes neurais de sintonia óptima para o tempo de crescimento -----	222
6.5 CONCLUSÕES -----	226
7. SISTEMA DE SINTONIA AUTOMÁTICA -----	229
7.1 INTRODUÇÃO -----	229

7.2	SIMULADOR NEURO-GENÉTICO -----	230
7.3	SINTONIA OPTIMIZADA PARA OS CRITÉRIOS DE DESEMPENHO ISOLADOS -----	234
7.3.1	Verificação da estabilidade do sistema -----	235
7.3.2	Sintonia otimizada para cada critério de desempenho -----	238
7.4	SINTONIA OPTIMIZADA COM PONDERAÇÃO DE CRITÉRIOS -----	252
7.5	CONCLUSÕES -----	256
8.	COMPARAÇÃO DO DESEMPENHO DE DIFERENTES MÉTODOS DE SINTONIA -----	257
8.1	INTRODUÇÃO -----	257
8.2	ESTIMAÇÃO DE MODELOS -----	258
8.3	COMPARAÇÃO DE CRITÉRIOS DE DESEMPENHO PARA DIFERENTES MÉTODOS DE SINTONIA -----	261
8.4	CONCLUSÕES -----	270
9.	CONCLUSÕES -----	273
9.1	CONCLUSÕES GERAIS -----	273
9.2	SUGESTÕES DE TRABALHO FUTURO -----	277
	Referências -----	281

LISTA DE FIGURAS

Fig. 1.1 - Malha típica de controlo. -----	5
Fig. 1.2 - Malha de controlo com perturbação à entrada do processo e ruído à saída. -----	7
Fig. 1.3 - Resposta ao degrau dum processo genérico. -----	16
Fig. 1.4 - Malha de controlo <i>PID</i> com relé. -----	22
Fig. 1.5 - Diagrama de blocos para sistema sujeito ao controlador <i>IMC</i> . -----	24
Fig. 1.6 - Sistema realimentado com perturbação externa à saída. -----	34
Fig. 1.7 - Controlador <i>PID</i> com dois graus de liberdade e ponderação da referência. -----	38
Fig. 1.8 - Controlador difuso tipo <i>PID</i> . -----	39
Fig. 1.9 - Auto-sintonia do controlador <i>PID</i> usando redes neuronais. -----	41
Fig. 1.10 - Esquema geral de controlo usando redes neuronais. -----	42
Fig. 2.1 - Um neurónio típico. -----	51
Fig. 2.2 - Alguns tipos de Neurónios: A-bipolar, B-unipolar, C-multipolar. -----	52
Fig. 2.3 - Alguns exemplos de sinapses: A - neurónio sensitivo, B - neurónio de associação, C - neurónio motor. -----	53
Fig. 2.4 - Alguns pontos do neurónio onde é possível encontrar sinapses. -----	54
Fig. 2.5 - Combinador linear adaptativo (ALC). -----	55
Fig. 2.6 - Rede neuronal por camadas - <i>MLP</i> . -----	56
Fig. 2.7 - Rede neuronal com funções de base radial - <i>RBF</i> . -----	65
Fig. 2.8 - Rede neuronal do tipo <i>B-Spline</i> . -----	71
Fig. 2.9 - Distribuição de nós para a entrada x_1 . -----	74
Fig. 2.10 - Algoritmo <i>ASMOT</i> . -----	79
Fig. 2.11 - Algoritmo genético simples. -----	88

Fig. 3.1 - Processo + controlador <i>PID</i> .-----	94
Fig. 3.2 - Esquematização do processo de auto-sintonia do controlador <i>PID</i> .-----	98
Fig. 3.3 - Momentos de actuação no processo de auto-sintonia do controlador <i>PID</i> .-----	101
Fig. 3.4 - Evolução parcial da optimização, com população inicial determinística.-----	108
Fig. 3.5 - Registo final da optimização, população inicial determinística: 289 indivíduos. -----	109
Fig. 3.6 - Evolução parcial da optimização, com população inicial aleatória.-----	109
Fig. 3.7 - Registo final da optimização, população inicial aleatória: 20 indivíduos.-----	110
Fig. 3.8 - Representação dos percentis.-----	111
Fig. 3.9 - Sintonia genética: seguimento da referência.-----	113
Fig. 3.10 - Sintonia genética: rejeição de perturbações à saída.-----	114
Fig. 3.11 - Seguimento da referência: 3 sintonias distintas.-----	115
Fig. 3.12 - Rejeição de perturbações à saída: 3 sintonias distintas.-----	115
Fig. 4.1 - Sobreposição do conjunto de adaptação no espaço de trabalho.-----	121
Fig. 4.2 - Normalização do conjunto base de padrões.-----	122
Fig. 4.3 - Aprendizagem de <i>MLP</i> em diferido para o mapeamento <i>PID</i> → <i>ITAE</i> .-----	126
Fig. 4.4 - Adaptação em linha de <i>MLP</i> para $\alpha = 0.1$.-----	128
Fig. 4.5 - Evolução do <i>MSE</i> na aprendizagem de <i>RBF</i> em diferido com raio = 0.15.-----	131
Fig. 4.6 - Aprendizagem de <i>RBF</i> em diferido com raio = 0.15. Esquerda: <i>MRE</i> , direita: <i>SSRE</i> .-----	131
Fig. 4.7 - Aprendizagem de <i>RBF</i> . Heurística 2, $k=7\%$. Evolução do <i>MSE</i> .-----	134
Fig. 4.8 - Aprendizagem de <i>RBF</i> . Heurística 2, $k=7\%$. Esquerda: <i>MRE</i> , direita: <i>SSRE</i> .--	135
Fig. 4.9 - Aprendizagem de <i>RBF</i> com a heurística 4. Evolução do <i>MSE</i> .-----	135
Fig. 4.10 - Aprendizagem de <i>RBF</i> com a heurística 4. Esquerda: <i>MRE</i> , direita <i>SSRE</i> . ---	136
Fig. 4.11 - Adaptação em linha de <i>RBF</i> para $\alpha = 0.04$.-----	137

Fig. 4.12 - Aprendizagem em diferido de <i>B-Spline</i> .-----	143
Fig. 4.13 - Adaptação de <i>B-Spline</i> para $tx=4\%$.-----	148
Fig. 5.1 - Sistema de controlo de nível. -----	152
Fig. 5.2 - Evolução das medidas de identificação para $\sigma \in \{0.1, 0.5, 1, 2\}$. -----	156
Fig. 5.3 - Discretizações de x_1 e x_2 com $\alpha = 0.1$. -----	159
Fig. 5.4 - Evolução das medidas de identificação rectificadas para $\sigma = 0.1$ ($\sigma \in \{0.1, 0.5, 1, 2\}$). -----	161
Fig. 5.5 - $(F(1), F(2))$ para as funções de transferência linearizadas e não linearizadas.	162
Fig. 5.6 - $(F(1), F(2))$ para dados transformados.-----	164
Fig. 5.7 - Plano dos óptimos, pontos internos e externos. -----	169
Fig. 5.8 - Estrutura prismática hexagonal recta com plano dos óptimos a meia altura. ---	169
Fig. 5.9 - Discretização para um exemplo do conjunto base de experimentação fazendo-se $N=5$. -----	171
Fig. 5.10 - Distribuição de pontos após migração para o 1º octante. -----	173
Fig. 5.11 - $TC=f(ITAE)$, para um processo com $k_p = 1, T_1=1, T_2=0.16$ e $L=0.64$. -----	178
Fig. 6.1 - Esquerda k_c^0 constante $ITAE \equiv f(t_i, t_d)$. Direita t_i^0 constante $ITAE \equiv f(k_c, t_d)$. -----	187
Fig. 6.2 - t_d^0 constante $ITAE \equiv f(k_c, t_i)$. -----	187
Fig. 6.3 - Modelo <i>PID</i> \rightarrow <i>ITAE</i> com $\lambda = 10^{-3}$. -----	188
Fig. 6.4 - Parâmetros 1 e 6 em função das medidas de identificação. -----	197
Fig. 6.5 - Saídas de teste para o parâmetro 1. -----	202
Fig. 6.6 - Padrões de Teste: objectivo e propagado pelo modelo para o qual <i>MSE</i> teste é mínimo.-----	210

Fig. 6.7 - <i>BIC</i> , complexidade, <i>MSE</i> e <i>MSRE</i> para o mapeamento $MI + PID \rightarrow TC$ utilizando o <i>ASMOTD</i> .-----	219
Fig. 6.8 - <i>BIC</i> , Complexidade, <i>MSE</i> e <i>MSRE</i> para o mapeamento $MI \rightarrow k_c^{ITAE}$.-----	221
Fig. 6.9 - <i>BIC</i> , Complexidade, <i>MSE</i> e <i>MSRE</i> para o mapeamento $MI \rightarrow t_i^{ITAE}$.-----	221
Fig. 6.10 - <i>BIC</i> , Complexidade, <i>MSE</i> e <i>MSRE</i> para o mapeamento $MI \rightarrow t_d^{ITAE}$.-----	222
Fig. 7.1 - Modelo SIMULINK para auto-sintonia neuro-genética do controlador <i>PID</i> . --	230
Fig. 7.2 - Sintonia sem adaptação para o processo com $T1=1, T2=0.242$ e $L=0.306$. ----	235
Fig. 7.3 - Adaptação para o <i>ITAE</i> para o processo com $T1=1, T2=0$ e $L=0.198$. -----	242
Fig. 7.4 - Adaptação para o <i>ITAE</i> para o processo com $T1=1, T2=0.129$ e $L=0.627$. ----	243
Fig. 7.5 - Adaptação para o <i>ITAE</i> para o processo com $T1=1, T2=0.242$ e $L=0.399$. ----	243
Fig. 7.6 - Adaptação para o tempo de crescimento para o processo com $T1=1, T2=0$ e $L=0.198$.-----	244
Fig. 7.7 - Adaptação para o tempo de crescimento para o processo com $T1=1, T2=0.129$ e $L=0.627$.-----	244
Fig. 7.8 - Adaptação para o tempo de crescimento para o processo com $T1=1, T2=0.242$ e $L=0.399$.-----	245
Fig. 7.9 - Adaptação otimizada para o <i>ITAE</i> com o processo variante no tempo. -----	246
Fig. 7.10 - Adaptação otimizada para o tempo de crescimento considerando o processo variante no tempo ($L: 0.162 \rightarrow 0.229 \rightarrow 0.306$). -----	249
Fig. 7.11 - Adaptação otimizada para o tempo de crescimento considerando o processo variante no tempo ($L: 0.192 \rightarrow 0.269 \rightarrow 0.355 \rightarrow 0.451 \rightarrow 0.560$). -----	251
Fig. 7.12 - <i>ITAE</i> e tempo de crescimento para a adaptação referente à Fig. 7.11. -----	251
Fig. 7.13 - Adaptação para ponderação de critérios com $k=0$.-----	254
Fig. 7.14 - Adaptação para ponderação de critérios com $k=0.5$. -----	255
Fig. 7.15 - Adaptação para ponderação de critérios com $k=1$.-----	255

Fig. 8.1 - Comparação das respostas ao degrau para o processo com $T=1$ e $L=0.27$. -----	259
Fig. 8.2 - Comparação das respostas ao degrau para o processo com $T_1=1$, $T_2=0.20$ e $L=0.05$. -----	260
Fig. 8.3 - Comparação das respostas ao degrau para o processo com $T_1=1$, $T_2= 1$ e $L=0.83$. -----	260
Fig. 8.4 - Respostas ao degrau do sistema em malha fechada para o processo com $T=1$ e $L=0.27$, considerando os vários métodos de sintonia. -----	264
Fig. 8.5 - Respostas ao degrau do sistema em malha fechada para o processo com $T_1=1$, $T_2=0.20$ e $L=0.05$, considerando os vários métodos de sintonia. -----	264
Fig. 8.6 - Respostas ao degrau do sistema em malha fechada para o processo com $T_1=1$, $T_2= 1$ e $L=0.83$, considerando os vários métodos de sintonia. -----	265
Fig. 8.7 - Legenda para os gráficos de Fig. 8.8 a Fig. 8.10. -----	267
Fig. 8.8 - Evolução do <i>ITAE</i> e do <i>ITAY</i> para os exemplos em estudo. -----	267
Fig. 8.9 - Evolução do tempo de crescimento e do tempo de estabelecimento para os exemplos em estudo. -----	268
Fig. 8.10 - Evolução da percentagem de sobre-elevação para os exemplos em estudo. ---	269

LISTA DE TABELAS

Tab. 1.1 - Parâmetros <i>PID</i> avaliados pelas regras de Ziegler e Nichols. -----	17
Tab. 1.2 - Não linearidades do tipo relé. -----	22
Tab. 1.3 - Parâmetros <i>PID</i> avaliados pela técnica de Cohen-Coon. -----	27
Tab. 2.1 - Heurísticas para determinação dos raios. -----	69
Tab. 2.2 - Algoritmo de programação genética para treino de redes neuronais do tipo <i>B-Spline</i> .-----	91
Tab. 3.1 - Desempenho da aprendizagem de redes neuronais do tipo <i>MLP</i> . -----	105
Tab. 3.2 - Parâmetros do algoritmo genético. -----	106
Tab. 3.3 - Legenda para diferentes configurações de ponderação (Fig. 3.9 e Fig. 3.10). -	112
Tab. 3.4 - Legenda para diferentes tipos de sintonia (Fig. 3.11 e Fig. 3.12).-----	115
Tab. 4.1- Algoritmo para construção da matriz <i>PID</i> .-----	119
Tab. 4.2- Conjuntos de aprendizagem. -----	119
Tab. 4.3- Desempenho da aprendizagem para selecção da topologia. -----	124
Tab. 4.4 - Pontuações finais para ambos os mapeamentos e respectivas somas.-----	125
Tab. 4.5 - <i>MSE</i> e <i>MRE</i> iniciais para adaptação em linha de <i>MLPs</i> .-----	127
Tab. 4.6- Adaptação em linha para as <i>MLPs</i> . -----	128
Tab. 4.7 - Aprendizagem em diferido, valores mínimos para o <i>MRE</i> no conjunto de teste. -----	130
Tab. 4.8 - Desempenho para as 6 heurísticas. -----	134
Tab. 4.9- <i>MSE</i> e <i>MRE</i> iniciais para adaptação em linha de <i>RBFs</i> . -----	136
Tab. 4.10- Adaptação em linha para as <i>RBFs</i> . -----	136

Tab. 4.11- Algoritmo para obtenção de modelo com boa capacidade de generalização. -	139
Tab. 4.12- Mínimos de <i>BIC</i> treino e <i>MRE</i> teste tendo em vista a procura de modelo com boa capacidade de generalização.-----	140
Tab. 4.13- Mínimo de <i>MRE</i> teste após um lote de inicializações aleatórias dos conjunto de aprendizagem. -----	140
Tab. 4.14- Mínimos de <i>BIC</i> treino e <i>MRE</i> teste para <i>B-Splines</i> de 2ª ordem. -----	142
Tab. 4.15 - Diferentes estados dum processo de aprendizagem. -----	143
Tab. 4.16- Aprendizagem com vista à exactidão do modelo para o conjunto de treino.--	144
Tab. 4.17- Modelo obtido com aprendizagem em diferido de <i>B-Splines</i> . -----	146
Tab. 4.18- <i>MSE</i> e <i>MRE</i> iniciais para adaptação em linha de <i>B-Splines</i> .-----	146
Tab. 4.19- Adaptação em linha para as <i>B-Splines</i> .-----	147
Tab. 4.20- Aprendizagem em diferido para os 3 tipos de redes neuronais. -----	148
Tab. 4.21- Resultados iniciais para a adaptação em linha. -----	148
Tab. 4.22- Ganhos de adaptação do <i>MSE</i> . -----	149
Tab. 5.1 - Cociente dos valores singulares para as medidas de identificação não rectificadas. -----	156
Tab. 5.2 - Máximos valores de <i>Q</i> para rectificações com diferentes valores de σ .-----	160
Tab. 5.3 - Distribuição das medidas de identificação: não rectificadas rectificadas.-----	163
Tab. 5.4 - Dados transformados: distribuição das medidas de identificação: não rectificadas rectificadas. -----	164
Tab. 5.5 - Discretização duma esfera para um dado <i>N</i> .-----	171
Tab. 5.6 - Critérios para estabilização de pontos.-----	174
Tab. 5.7 - Cociente de aproximação. -----	174
Tab. 5.8 - Cociente de aproximação tomando o 1º e 3º critérios. -----	174
Tab. 5.9 - Aproximação linear para o <i>ITAE</i> e para o tempo de crescimento.-----	176
Tab. 5.10 - Conjuntos de aprendizagem após filtragem. -----	179

Tab. 6.1 - Modelo para o mapeamento $MI + PID \rightarrow ITAE$ utilizando o <i>ASMOD</i> . -----	182
Tab. 6.2 - Desempenho do modelo da Tab. 6.1. -----	182
Tab. 6.3 - % de <i>MRE</i> para o espaço de entrada em cada uma das 3 configurações. -----	184
Tab. 6.4 - Rede neuronal para o processo em estudo quando o espaço de entrada assume as 3 configurações. -----	184
Tab. 6.5 - Heurística para obtenção de padrões nas vizinhanças do mínimo do <i>ITAE</i> . ---	186
Tab. 6.6 - Modelo óptimo. -----	188
Tab. 6.7 - Valores de desempenho para os 110 modelos do conjunto base de experimentação. -----	196
Tab. 6.8 - Medidas de desempenho para o mapeamento $MI \rightarrow P_i$ usando-se <i>ASMOD</i> .--	200
Tab. 6.9 - Mapeamento $MI \rightarrow P_i$ usando-se programação genética uni-objectivo para o modelo com menor <i>MSE</i> teste.-----	202
Tab. 6.10 - Valores para aprendizagem de redes neuronais do tipo <i>B-Spline</i> recorrendo à programação genética multi-objectivo. -----	203
Tab. 6.11 - Parâmetros para obtenção de soluções não dominadas e soluções preferidas tomando por base as 6803 soluções da expressão (6.23). -----	205
Tab. 6.12 - Medidas de desempenho para as 3520 soluções não dominadas. -----	205
Tab. 6.13 - Soluções preferidas.-----	206
Tab. 6.14 - Modelo para a solução preferida 5. -----	206
Tab. 6.15 - Medidas de desempenho para mapeamento $MI \rightarrow P_1$ -----	207
Tab. 6.16 - Mapeamento $MI \rightarrow P_i$ usando-se algoritmos genéticos uni-objectivo para o modelo com menor <i>MSE</i> teste.-----	209
Tab. 6.17 - Valores para os 10 parâmetros da aprendizagem de redes neuronais do tipo <i>B-Spline</i> recorrendo à programação genética multi-objectivo. -----	210
Tab. 6.18 - Totais de soluções não dominadas e soluções preferidas para os 10 parâmetros. -----	211

Tab. 6.19 - Valores para a escolha da solução preferida com melhor capacidade de generalização.-----	212
Tab. 6.20 - Totais de soluções não dominadas e preferidas com as definições da Tab. 6.19.-----	212
Tab. 6.21 - Soluções preferidas para os 10 parâmetros. -----	213
Tab. 6.22 - Resumo das medidas de desempenho para a topologia parcialmente fixa. ---	215
Tab. 6.23 - Resumo das medidas de desempenho para a topologia livre. -----	216
Tab. 6.24 - Aprendizagem de redes neuronais para mapeamento do tempo de crescimento.-----	217
Tab. 6.25 - Modelo para o mapeamento $MI + PID \rightarrow TC$ utilizando o <i>ASMOD</i> . -----	219
Tab. 6.26 - Conjuntos de aprendizagem para mapeamentos de sintonia.-----	220
Tab. 6.27 - Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow k_c^{TC}$.-----	223
Tab. 6.28 - Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow t_i^{TC}$.-----	224
Tab. 6.29 - Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow t_d^{TC}$.-----	225
Tab. 6.30 - Modelos para os 6 mapeamentos de sintonia óptima para os critérios de desempenho. -----	226
Tab. 7.1 - Tarefas para auto-adaptação do controlador neuro-genético. -----	231
Tab. 7.2 - Verificação da estabilidade utilizando-se sintonia otimizada para o tempo de crescimento.-----	236
Tab. 7.3 - Aproximação dos mínimos para os sistemas estáveis. -----	238
Tab. 7.4 - Duração das simulações e períodos da onda quadrada. -----	239
Tab. 7.5 - Processos tipo para validação da adaptação, considerando-se os critérios de desempenho isolados. -----	240
Tab. 7.6 - Ganhos de adaptação para os processos da Tab. 7.5.-----	241
Tab. 7.7 - Processos seleccionados e respectivas taxas de adaptação.-----	242

Tab. 7.8 - Ganhos de adaptação obtidos nos procedimentos de auto-sintonia com ponderação de critérios. -----	253
Tab. 8.1 - Exemplos de modelos de 1ª ordem. -----	259
Tab. 8.2 - Verificação da exactidão dos modelos de 1ª ordem. -----	261
Tab. 8.3 - Parâmetros <i>PID</i> para os vários métodos de sintonia. -----	263
Tab. 8.4 - Legenda para Fig. 8.4 a Fig. 8.6. -----	263
Tab. 8.5 - Critérios de desempenho para os vários métodos de sintonia. -----	266

LISTA DE ABREVIATURAS

<i>ALC</i>	<i>Adaptive Linear Combiner</i>
<i>AMN</i>	<i>Associative Memory Networks</i>
<i>ARMA</i>	<i>Auto-Regressive Moving Average</i>
<i>ASMOD</i>	<i>Adaptive Spline Modelling of Observation Data</i>
BFGS	Broyden Fletcher Goldfarb Shanno
<i>BIC</i>	<i>Bayesian Information Criterion</i>
<i>CACSD</i>	<i>Computer Aided Control System Design</i>
CBE	Conjunto Base de Experimentação
<i>CMAC</i>	<i>Cerebellar Model Articulation Controller</i>
<i>FLN</i>	<i>Functional Link Network</i>
<i>GPC</i>	<i>Generalised Predictive Control</i>
<i>IMC</i>	<i>Internal Model Control</i>
<i>IAE</i>	<i>Integral of Absolute Error</i>
<i>ITAE</i>	<i>Integral of Time Multiplied by Absolute Error</i>
MI	Medidas de Identificação
<i>MIMO</i>	<i>Multi-Input Multi-Output</i>
<i>MLP</i>	<i>MultiLayer Perceptrons</i>
<i>NLMS</i>	<i>Normalized Least Mean Squares</i>
<i>PID</i>	Proporcional, Integral e Derivativo
PG	Programação Genética
<i>RBF</i>	<i>Radial Basis Function</i>
<i>RWS</i>	<i>Roulette Wheel Selection</i>
<i>SUS</i>	<i>Stochastic Universal Sampling</i>
TC	Tempo de crescimento

Agradecimentos

Agradeço ao meu orientador Prof. Doutor António Ruano pela orientação, incentivo e sempre proveitosa troca de ideias.

Agradeço o suporte do programa de doutoramento da acção 5.2 do Prodep, pela possibilidade que proporcionou na aquisição de material relacionado com esta dissertação. Agradeço ao projecto "Sintonia Automática de Controladores PID utilizando Redes Neurais e Algoritmos genéticos", Praxis /P/EEI/12149/1998, a todos os colegas que nele colaboraram pela sempre proveitosa troca de ideias e motivação. Agradeço ao Professor Peter John Fleming e ao Doutor Jon Love, do *Department of Automatic Control and Systems Engineering* da Universidade de Sheffield, consultores do referido projecto, pelo acolhimento que me facultaram durante as minhas deslocações à Universidade de Sheffield.

Agradeço ao Prof. Doutor Rafael Santos pela colaboração no tratamento da influência do erro de estimação de parâmetros no cálculo do ITAE. Os meus agradecimentos são extensivos aos colegas de docência da Área Departamental de Engenharia Electrónica e Computação, do Centro de Sistemas Inteligentes (CSI) pelos meios disponibilizados. Aos colegas Cristiano Cabrita, João Cardoso, José Luís Carmo, José Maria André, Maria da Graça Marques e Patrício Serendero, agradeço a colaboração no que se refere à revisão e processamento do texto, bem como pela sempre proveitosa troca de ideias sobre investigação e temas relacionados com esta dissertação.

Agradeço aos meus pais restante família e amigos, pelo constante incentivo e apoio dado ao longo destes anos.

RESUMO

Os controladores PID são extensivamente utilizados em aplicações industriais. A popularidade deste tipo de controladores advém da sua simplicidade (apenas 3 termos para sintonizar) e robusto desempenho. Dado que os processos a controlar poderão variar ao longo do tempo, levando à necessidade de uma nova sintonia do controlador, métodos de sintonia automática de controladores PID revelam-se de grande interesse tanto prático como teórico.

Na presente dissertação propõe-se uma técnica de sintonia automática do controlador PID, com base em metodologias de *soft-computing*.

A sintonia de controladores pode ser vista como um problema de optimização simultânea de vários critérios assim, o sistema proposto incorpora algoritmos genéticos para esse propósito. Destinando-se o método desenvolvido a aplicações em tempo real, houve a necessidade de se proceder à modelação dos critérios de controlo. Este procedimento tem vantagem tanto a nível de consumo de tempo, como na prevenção de situações de instabilidade sempre possíveis durante um processo de optimização, e não sujeita o processo a perturbações necessárias ao processo de optimização; estes aspectos são de fundamental importância nas aplicações em tempo real.

O problema de modelação de critérios de controlo bem como de parâmetros optimizados do controlador foi tratado utilizando-se redes neuronais. Foram testadas as capacidades de modelação e de adaptação de vários tipos de redes neuronais, nomeadamente perceptrão multi-camada, redes com funções de base radial e *B-Splines*.

Os modelos neuronais a que se chegaram, juntamente com os algoritmos genéticos, formam alguns dos blocos constituintes do modelo de auto-sintonia neuro-genética que se propõe. Foi construído um simulador que implementa a presente arquitectura e valida a metodologia proposta.

ABSTRACT

PID controllers are extensively used in industrial applications. Its popularity comes from the fact that they are very simple -just three terms to tune- as well as from its robust performance. As plants can be time-varying, therefore precluding the need for retuning, methods for PID auto-tuning exhibit great theoretical and practical interest.

In this thesis, an automatic tuning technique for PID controllers based on soft-computing methodologies is purposed.

The tuning of controllers can be interpreted as a simultaneous optimization problem of several criteria. For this purpose, the proposed system incorporates genetic algorithms. As this method aimed for real time applications, models are used to approximate the control criteria. This has the following advantages: first, there are savings in execution time; secondly, instability situations, which can always occur during an optimization process, are avoided; finally the optimization can be performed avoiding the need of applying perturbations to the actual process, which is always undesirable.

Neural networks are used to model the proposed tuning criteria, as well as to model the PID controller optimized parameters. The modeling and adaptation capacities of several neural networks, namely multilayer perceptron, radial basis function networks and B-Spline networks were tested and compared.

The chosen neural models and genetic algorithms constitute some of the building blocks of the proposed neuro-genetic auto-tuning method.

A simulator implemented this architecture, in order to validate the proposed methodology.

1. INTRODUÇÃO

1.1 VISÃO GERAL

O controlador Proporcional, Integral e Derivativo (*PID*) é ainda hoje um dos controladores mais utilizados em aplicações industriais. A grande popularidade deste tipo de controlador advém do facto de apresentar um desempenho robusto numa vasta gama de condições de operação e simultaneamente tratar-se dum controlador simples, pois apenas possui 3 parâmetros para sintonizar. Por esta razão, em boa parte das instalações industriais, a sintonia do controlador *PID* é feita manualmente. Constata-se contudo, que grande parte dos processos são variantes no tempo, as componentes envelhecem e as não linearidades são uma presença constante nos processos industriais, assim, os controladores *PID* terão que ser necessariamente sintonizados com bastante frequência. Se se atender a que numa unidade fabril poderão existir centenas destes simples controladores, conclui-se facilmente que métodos de visem a sua sintonia automática são de grande interesse económico, pelo tempo poupado relativamente à sintonia manual. O interesse dos métodos e técnicas de sintonia automática é reforçado lembrando que, adicionalmente, estes métodos conduzirão a respostas mais robustas, uma vez que se pode ter controladores sintonizados para malhas de controlo específicas.

1.2 PRINCIPAIS OBJECTIVOS E ESTRUTURA DA DISSERTAÇÃO

Na dissertação que se apresenta é feito um estudo comparativo da aplicação de recentes metodologias de *softcomputing*, tais como redes neuronais, programação genética e lógica

difusa, à sintonia automática de controladores *PID*; a partir desse estudo comparativo é proposta uma nova metodologia de sintonia de tais controladores.

As vantagens deste tipo de abordagem, relativamente a abordagens convencionais, podem-se resumir nos seguintes pontos:

- a sintonia do controlador pode ocorrer tanto numa fase inicial com o processo em malha aberta, como em pleno funcionamento do sistema mantendo-se a malha fechada.
- contrariamente ao que se passa com outros métodos, não necessita de interrupção do funcionamento do sistema de controlo, ficando assim facilitado o processo de adaptação em linha.
- são geralmente obtidas respostas com bons valores de amortecimento sem se ter que recorrer a morosos processos de optimização; esta característica viabiliza o método na perspectiva da sua utilização em tempo real.
- não recorre à obtenção dum modelo analítico para o processo; processos variantes no tempo caracterizados por diferentes funções de transferência, são adequados a esta metodologia.
- não necessita de *hardware* específico, podendo como tal, ser facilmente integrado em controladores *PID* comerciais.

Uma revisão da literatura inicial, visou o levantamento das técnicas e métodos de sintonia do controlador *PID* apresentados nos últimos anos, cujo resumo se apresenta no capítulo 1. Após esta primeira abordagem estabeleceu-se que o novo método de sintonia recorrerá a técnicas de *softcomputing*, nomeadamente: redes neuronais e programação genética; os fundamentos destas técnicas são apresentados no capítulo 2. No capítulo 3 será apresentada uma arquitectura de controlo e auto-sintonia adaptativa do controlador *PID*. De acordo com a arquitectura apresentada, será necessário levar a cabo a modelação de processos utilizando redes neuronais. A escolha do tipo de redes neuronais para a construção de tais modelos fica a cargo do capítulo 4. No capítulo 5 é feito um levantamento de funções de transferência, eventualmente mais utilizadas em processos industriais, com vista à determinação de um conjunto base de experimentação, para o qual a metodologia proposta

é preferencialmente testada. O capítulo 6 debruça-se sobre a aprendizagem de redes neuronais para o referido conjunto de processos. No capítulo 7 apresenta-se a implementação da metodologia proposta, bem como resultados ilustrativos do seu funcionamento auto-adaptativo. No capítulo 8 é comparado o desempenho obtido pela presente metodologia com o desempenho de outras técnicas de sintonia tomadas como referência. A dissertação termina com o Capítulo 9 onde se apresentam conclusões e sugestões de trabalho futuro.

O trabalho de investigação onde se enquadra a presente dissertação deu origem às seguintes contribuições:

- Sistema de auto-sintonia do controlador *PID* utilizando redes neuronais do tipo *MLP* para mapeamento de parâmetros *PID* otimizados para o *ITAE*. Os detalhes desta abordagem podem ser vistos no artigo intitulado "Comparison of Alternative Approaches to Neural Network *PID* Autotuning" [1], publicado na revista *Journal of Systems Engineering*.
- Teste da sintonia do controlador *PID* quando os parâmetros são determinados utilizando-se algoritmos genéticos juntamente com redes neuronais. Para além do critério de desempenho mencionado no ponto anterior (*ITAE*), é considerado outro critério de desempenho alternativo, resultando a função objectivo do procedimento de optimização genética numa soma ponderada de 2 critérios. A comparação deste método com técnicas comuns de optimização por métodos baseados em gradiente e com a utilização das regras de Ziegler e Nichols, resultou na comunicação intitulada "New Methods for *PID* Autotuning" [2], apresentada na 3rd Portuguese Conference on Automatic Control.
- Desenvolvimento de uma arquitectura de controlo *PID* possibilitando auto-sintonia neuro-genética. Os testes iniciais desta arquitectura compararam o desempenho da sintonia optimizada utilizando redes neuronais juntamente com algoritmos genéticos, com o desempenho da optimização utilizando técnicas de gradiente. Este estudo conduziu às publicações, na revista *IMACS Journal of Mathematics and Computers in Simulation*, do artigo intitulado "Neuro-genetic *PID* Autotuning:

Time Invariant Case" [3], e na 14th *IFAC World Congress* à comunicação intitulada "Automatic Tuning of Controllers Using a Neural-Genetic System" [4].

- Estudo comparativo dos desempenhos de redes neuronais *MLP* e *RBF*, tanto no que se refere à aprendizagem em diferido como no que respeita à adaptação em linha. Estes aspectos são fundamentais para a construção tanto dos blocos de modelação dos critérios, quanto dos blocos de sintonia, referentes à arquitectura do ponto anterior. Os resultados provenientes deste estudo estão resumidos numa comunicação intitulada "On-line Adaptation of Neural Networks" [5], apresentada na *International Conference on Engineering Applications of Neural Networks (EANN'99)*.
- Construção do bloco de modelação do *ITAE* no âmbito da arquitectura adoptada. O modelo para este critério de desempenho recorreu a redes neuronais do tipo *MLP*, *RBF* e *B-Spline* treinadas em diferido e adaptadas em linha. O estudo comparativo do desempenho dos modelos assim obtidos constam da comunicação intitulada "Comparison of Off-line and On-line Performance of Alternative Neural Network Models" [6], apresentada na 8th *Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000)*.
- Discriminação de processos e de parâmetros *PID*, para que o sistema em desenvolvimento acomode variação no tempo da dinâmica do processo. Mais detalhes sobre este aspecto podem ser consultados na comunicação intitulada "Neuro-Genetic *PID* Autotuning" [7], apresentada na *European Control Conference (ECC'2001)*.

Posteriormente, a investigação continuada na presente área tem-se centrado na avaliação do desempenho da metodologia proposta, considerando-se variação no tempo da dinâmica do processo bem como a acomodação simultânea e ponderada de 2 critérios de desempenho. Os resultados que se têm obtido serão publicados em breve.

1.3 O CONTROLADOR *PID*

O controlador *PID* [8] é um dos algoritmos de controlo mais utilizados nos mais diversos ramos da industria. Estes simples controladores são compostos por 3 termos:

- proporcional *P*, corresponde à acção de controlo proporcional,
- integral *I*, corresponde à acção de controlo proporcional ao integral no tempo do erro,
- derivativo *D*, corresponde à acção de controlo proporcional à derivada do erro.

A presente secção debruça-se sobre a topologia da malha de controlo típica onde se insere este controlador; serão apresentadas considerações sobre cada uma das acções de controlo separadamente. Os métodos de sintonia do controlador *PID* serão predominantemente tratados na secção seguinte.

Considere-se a malha de controlo com realimentação negativa da Fig. 1.1.

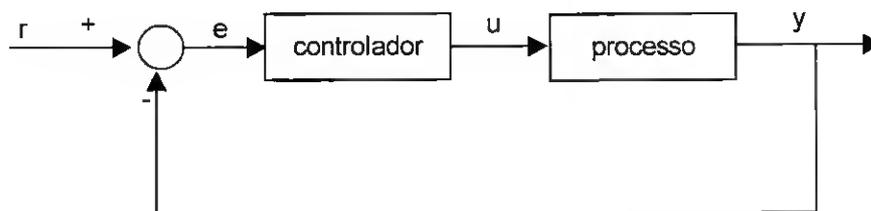


Fig. 1.1 – Malha típica de controlo.

em que se define:

y – sinal de saída,

r – sinal de referência,

e – sinal de erro,

u – sinal de controlo,

e cujas transformadas de Laplace representam-se por $Y(s)$, $R(s)$, $E(s)$ e $U(s)$.

O mecanismo de controlo por realimentação negativa da saída pode assumir inúmeras configurações. A configuração mais simples para esse propósito estipula que o sinal de

controlo u assume apenas um de dois valores possíveis: u_{max} , ou u_{min} , sendo a lei de controlo expressa por (1.1).

$$u = \begin{cases} u_{max}, & e > 0 \\ u_{min}, & e < 0 \end{cases} \quad (1.1)$$

Este tipo de realimentação denomina-se controlo *on-off* e tem a vantagem de não possuir parâmetros para sintonizar. Contudo, atendendo à forma com está definido, faz com que o sinal de controlo possa oscilar, pois a grandeza do erro não é tida em conta na reacção do controlador. Para se evitar este inconveniente, pode-se recorrer à utilização duma lei de controlo em que o sinal de controlo é proporcional ao erro, para valores de erro até certos limiares que definem a banda proporcional (P_b); obtém-se então controlo proporcional. Verifica-se que para valores elevados de erro, o controlo proporcional funciona com controlo *on-off*.

A lei de controlo proporcional fica definida à custa do ganho proporcional k_p de acordo com (1.2):

$$u_{max} - u_{min} = k_p P_b \quad (1.2)$$

Sistemas cuja transformada de Laplace não possua pólo na origem, quando submetidos ao controlo proporcional não conduz a erro nulo em regime estacionário; verifica-se contudo que quanto maior for o ganho de malha¹, menor o erro em regime estacionário.

As desvantagens apontadas até agora podem ser transpostas recorrendo ao controlador *PID*, cuja lei de controlo, na banda proporcional pode ser descrita por (1.3):

$$u(t) = k_c \left(e(t) + \frac{1}{t_i} \int_0^t e(\tau) d\tau + t_d \frac{d}{dt} e(t) \right) \quad (1.3)$$

Como referido no início, este controlador de 3 termos fica definido pelos seus 3 parâmetros:

¹ Ganho de malha é o produto do ganho do controlador pelo ganho do processo.

P – ganho proporcional,

I – constante de tempo integral e

D – constante de tempo derivativo, respectivamente designados por k_c , t_i e t_d .

Nas secções que se seguem vão ser descritas as 3 acções que compõem o controlador.

1.3.1 Acção Proporcional

Na situação em que o controlo é proporcional a lei (1.3) reduz-se a (1.4).

$$u(t) = k_c e(t) + u_b \quad (1.4)$$

A constante u_b é frequentemente fixada a $(u_{\max} + u_{\min})/2$, ou ajustada de tal forma que o erro em regime estacionário seja nulo.

Suponha-se que na malha de controlo típica apresentada na Fig. 1.1 se considera adicionalmente, uma perturbação à entrada do processo l e ruído aditivo n na sua saída, conforme a Fig. 1.2.

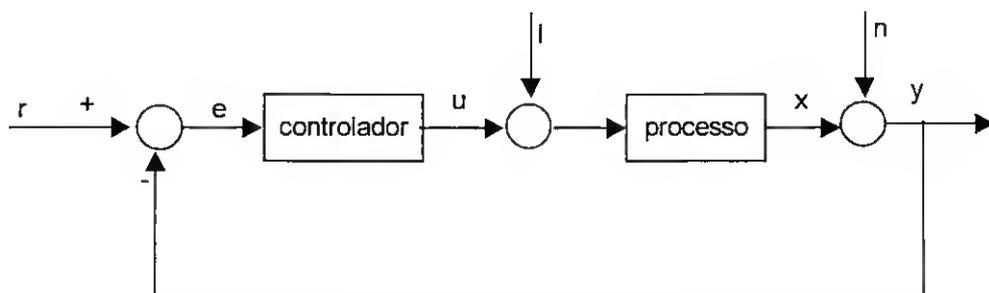


Fig. 1.2 – Malha de controlo com perturbação à entrada do processo e ruído à saída.

Admitindo que o processo tem ganho estático K_p então o seu modelo estático será dado por (1.5).

$$x = K_p (u + l) \quad (1.5)$$

Tomando por base a Fig. 1.2, pode-se estabelecer uma relação entre a saída do processo x , a referencia, a perturbação e o ruído, dada por (1.6).

$$x = \frac{k_c K_p}{1 + k_c K_p} (r - n) + \frac{K_p}{1 + k_c K_p} (l + u_b) \quad (1.6)$$

Tomando por base a expressão (1.6), algumas leituras podem ser feitas:

- sejam n e u_b iguais a zero: para que a saída do processo x se aproxime da referência r (erro nulo), o ganho de malha deverá ser elevado.
- para que o sistema seja insensível à perturbação l , o ganho de malha deverá ser elevado.
- na presença de ruído ($n \neq 0$), a saída do processo é de igual forma influenciada por este, e pela referência.
- a perturbação l e a polarização do sistema u_b influenciam o sistema da mesma forma.
- para que o sistema reduza a sensibilidade ao ruído, o ganho de malha deverá ser baixo.

Assim, pode-se dizer que a escolha do ganho de malha deverá ter em conta o compromisso entre os vários critérios de desempenho. A resposta sobre qual o melhor ganho de malha não é óbvia, mas sim, depende de qual o mais importante critério de desempenho a cumprir.

Conclui-se também que o erro em regime estacionário não é nulo, a menos que u_b seja escolhida tal que $u_b = u$ em regime estacionário.

A análise levada a cabo até agora pressupõe que o processo pode ser descrito apenas por um modelo estático; se se tiver em conta a dinâmica do sistema em malha fechada, atinge-se frequentemente situações de instabilidade para ganhos elevados.

1.3.2 Acção Integral

Tomando por base a expressão (1.3), o algoritmo de controlo para o controlador PI pode ser representado por (1.7).

$$u(t) = k_c \left(e(t) + \frac{1}{t_i} \int_0^t e(\tau) d\tau \right) \quad (1.7)$$

A principal função da acção integral do controlador é a possibilidade de se obter erro nulo em regime estacionário.

Esta constatação pode ser provada com o seguinte argumento:

Considere-se a hipótese de que em regime estacionário o sinal de controlo e o erro são constantes e iguais a u_0 e e_0 , respectivamente. Nestas circunstâncias, de acordo com (1.7) deduz-se a relação (1.8).

$$u_0 = k_c \left(e_0 + \frac{e_0}{t_i} t \right) \quad (1.8)$$

Se $e_0 \neq 0$, de acordo com (1.8) u_0 não é constante, o que contradiz a hipótese, ficando provada a sua negação: $e_0=0$ ou u_0 não é constante. Estando em regime estacionário conclui-se que $e_0=0$.

Quando se tratou o controlador proporcional, mostrou-se que existia uma constante de polarização u_b que deveria ser ajustada para que o erro em regime estacionário fosse nulo. Então, o controlador *PI* pode ser visto como um método segundo o qual esse reajuste é feito automaticamente.

As respostas de sistemas sujeitos ao controlador *PI* são mais oscilatórias e aproximam-se mais rapidamente do erro nulo, quanto menor for a constante de tempo integral t_i .

1.3.3 Acção Derivativa

O principal objectivo do controlador *PD* é o aumento da estabilidade do sistema em malha fechada. A situação de potencial instabilidade verificada até agora pode ser interpretada como sendo causada pelo atraso, imposto pela dinâmica do sistema, entre a correcção do sinal de controlo e a sua propagação até à saída do processo; este facto implica atraso na obtenção do erro. O controlador *PD* pode ser visto como um controlador proporcional que

funciona com uma predição da saída do processo. A predição ou estimativa é tomada pela tangente à curva de erro.

Seja o sinal de controlo proporcional ao erro com avanço de t_d :

$$u(t) = k_c e(t + t_d) \quad (1.9)$$

A expansão em série de Taylor, desprezando os termos de ordem superior à 1ª, conduz à expressão (1.10).

$$e(t + t_d) \approx e(t) + t_d \frac{d}{dt} e(t) \quad (1.10)$$

Tomando em conta a expressão (1.9), verifica-se (1.11).

$$u(t) = k_c \left(e(t) + t_d \frac{d}{dt} e(t) \right) \quad (1.11)$$

A expressão (1.11) é idêntica à que se obtém da lei de controlo *PID* geral (1.3), quando se considera apenas a acção proporcional e derivativa.

Dum modo geral, as respostas obtidas sob controlo *PD* apresentam aumento de amortecimento com o aumento da constante de tempo derivativa t_d , contudo, um aumento excessivo deste parâmetro faz com que o amortecimento volte a diminuir.

1.3.4 Implementação digital do controlador *PID*

Originalmente [8] as implementações do controlador *PID* utilizavam técnicas analógicas. Recentemente, com a vulgarização do uso de microprocessadores, tornou-se necessário o estudo de vários aspectos que se prendem com a implementação digital do controlador *PID*, tais como: amostragem, discretização e quantificação.

A implementação do algoritmo de controlo definido por (1.3) em computador digital, exige que os termos diferencial e integral sejam aproximados; existem vários métodos que levam a tais aproximações. Designe-se por t_k os instantes em que o computador lê a entrada analógica (instantes de amostragem), e por h o período de amostragem.

- Termo Proporcional: Considere-se o termo proporcional, dado por (1.4), com um factor de ponderação b para a referência e para uma banda proporcional tal que $u_b=0$. A versão discreta do termo proporcional exprime-se por (1.12).

$$P(t_k) = k_c (br(t_k) - y(t_k)) \quad (1.12)$$

- Termo Integral: A partir de (1.3), pode-se estabelecer que o termo integral é dado por (1.13).

$$I(t) = \frac{k_c}{t_i} \int_0^t e(\tau) d\tau \quad (1.13)$$

Derivando em ordem ao tempo ambos os membros, pode-se relacionar a derivada do termo integral com o sinal de erro (1.14):

$$\frac{d}{dt} I(t) = \frac{k_c}{t_i} e(t) \quad (1.14)$$

A implementação computacional da equação (1.14) consiste no cálculo aproximado da respectiva derivada, tendo em conta a sua interpretação geométrica. Os métodos considerados para implementação dessa aproximação são: diferenças para a frente (*forward differences*), diferenças para trás (*backward differences*) e aproximação de Tustin. Cada uma destas aproximações conduz a uma expressão recursiva para o termo integral num determinado instante em função de: termo integral no instante anterior, erro no próprio instante e erro no instante anterior; a expressão recursiva genérica apresenta-se em (1.15).

$$I(t_{k+1}) = I(t_k) + b_{i1}e(t_{k+1}) + b_{i2}e(t_k) \quad (1.15)$$

Os coeficientes b_{i1} e b_{i2} , assumem diferentes valores [8], consoante a aproximação em uso.

- Termo derivativo: Este termo faz com que no caso de existir ruído de alta frequência, este seja amplificado de tal forma que a amplitude é proporcional à sua frequência. Para se evitar este inconveniente, é comum associar ao termo derivativo um filtro de 1ª

ordem passa baixo com constante de tempo t_d/N , tipicamente com $8 \leq N \leq 20$. Assim, o termo derivativo em tempo contínuo cumpre a seguinte equação diferencial (1.16):

$$D(t) = -\frac{t_d}{N} \frac{d}{dt} D(t) - k_c t_d \frac{d}{dt} y(t) \quad (1.16)$$

Tal como aconteceu com o tratamento do termo integral, a expressão discreta recursiva que se obtém para o termo derivativo depende de qual das aproximações que se adopte. Todas as expressões discretas que daí resultam assumem a mesma forma (1.17), diferindo apenas nos parâmetros a_d e b_d [8]:

$$D(t_k) = a_d D(t_{k-1}) - b_d (y(t_k) - y(t_{k-1})) \quad (1.17)$$

Os algoritmos de cálculo computacional de cada termo, que compõe o controlador *PID* vistos até agora [8], são conhecidos por algoritmos de posição pois fornecem directamente a saída do controlador. Nas implementações digitais é comum implementar os algoritmos de velocidade ou algoritmos incrementais; com esta abordagem bastará avaliar as diferenças discretas num dado instante e adicionar sucessivamente os incrementos conforme se ilustra na expressão (1.18).

$$\Delta u(t_k) = u(t_k) - u(t_{k-1}) = \Delta P(t_k) + \Delta I(t_k) + \Delta D(t_k) \quad (1.18)$$

Partindo das expressões (1.12), (1.15) e (1.17), os incrementos $\Delta P(t_k)$, $\Delta I(t_k)$ e $\Delta D(t_k)$ são dados por respectivamente (1.19), (1.20) e (1.21):

$$\Delta P(t_k) = P(t_k) - P(t_{k-1}) = k_c (br(t_k) - y(t_k) - br(t_{k-1}) + y(t_{k-1})) \quad (1.19)$$

$$\Delta I(t_k) = I(t_k) - I(t_{k-1}) = b_{i1} e(t_k) + b_{i2} e(t_{k-1}) \quad (1.20)$$

$$\Delta D(t_k) = D(t_k) - D(t_{k-1}) = a_d \Delta D(t_{k-1}) - b_d (y(t_k) - 2y(t_{k-1}) + y(t_{k-2})) \quad (1.21)$$

1.4 ESTADO DA ARTE DE MÉTODOS DE SINTONIA AUTOMÁTICA

Os controladores *PID* têm sido abundantemente utilizados na indústria nos últimos pelo menos 60 anos [9]; no que respeita o controlo de processos estima-se que, actualmente, pelo menos 95% das malhas de controlo sejam do tipo *PID* [8]. Durante a década passada, foram publicados em revista mais de 300 artigos sobre teoria e aplicações do controlador *PID*; só em 1998, o número de publicações excedeu 60. Ainda em finais da última década, na *American Control Conference* de 1999 [10], foram apresentados cerca de 20 artigos sobre este tipo de controladores. No mesmo ano na *IFAC World Congress* [11], foram apresentados quase 90 artigos sobre os controladores *PID*. Face a este panorama, podemos dizer que se está a assistir a uma verdadeira explosão de novos resultados teóricos e novas aplicações práticas deste tipo de controladores. Apesar do grande número de novos resultados que têm surgido nesta área, os trabalhos de Ziegler e Nichols de 1942, são ainda amplamente usados em aplicações industriais. São também apontados [12] como outros exemplos de metodologias de referência, as técnicas desenvolvidas em 1952 por Cohen e Coon, ou o método *feedback relay* de Åström e Hägglund. Os algoritmos desenvolvidos nesses trabalhos recorrem ou à resposta temporal, ou à resposta na frequência, do processo que se pretende controlar. Tais técnicas, têm servido com bastante frequência, como termo de comparação relativamente às eficiências de técnicas recentemente desenvolvidas.

Existe [12] grande volume de literatura que se debruça sobre a sintonia de controladores *PID*. Como exemplos de boas revisões bibliográficas sobre esta matéria, podem-se apontar, por ordem cronológica, os trabalhos desenvolvidos por: Åström *et al.* em 1993 [13], Gorez e Calcev em 1997 [14], Tan *et al.* em 1999 [15], e mais recentemente por Cominos e Munro em 2002 [16].

O trabalho desenvolvido por Moradi e Johnson [12] apresenta uma nova classe de métodos de sintonia do controlador *PID*. O surgimento dessa nova classe de métodos está na sequência da investigação que se desenrolou no decurso dos últimos anos; nomeadamente, em 1989, Morari e Zafiriou [17] mostraram que tecnologias que utilizam Controlo de Modelo Interno (*IMC*) conduzem-nos ao controlador *PID*, para a grande maioria dos modelos dos processos utilizados na indústria. O projecto de controladores *PID* baseado no método *IMC* foi utilizado para processos de 1ª ordem em 1986 por Rivera *et al.* [18], e

posteriormente estendido a processos de 2ª ordem por Chien [19] em 1988. Em 1991 Grimble [20] utilizou a técnica de controlo H_∞ , tendo obtido controladores com estrutura semelhante à dos controladores *PID*. Wang et al. [21], desenvolveu em 2000, uma abordagem que utiliza a resposta na frequência e o algoritmo dos mínimos quadrados, para dimensionar o controlador *PID* equivalente ao projectado de acordo com IMC. Em 1999 e 2000, Rusnak [22] [23] utilizou a teoria dos controladores quadráticos lineares para projecto do controlador *PID*, tendo obtido resultados para processos até à 5ª ordem. Tan *et al.* [24] apresentou em 2000, para processos de 2ª ordem com tempo de atraso, uma metodologia de projecto do controlador *PID* baseada no Controlo Preditivo Generalizado (GPC). Katebi e Moradi [25] introduziram em 2001 o controlador *PID* preditivo para processos *SISO*, baseado no método GPC; em 2002 foi feita uma extensão [26] aos processos *MIMO* com representações polinomiais. A sequência directa destas metodologias envolvendo o controlador *PID* preditivo [25], [26], conduz aos trabalhos de Moradi e Johnson [12].

O guia de referência elaborado por Muhidin Lelic e Zoran Gajic [9] extrai o essencial dos mais recentes desenvolvimentos e inovações em termos de projecto, sintonia, implementação e considerações teóricas relativas a este tipo de controladores. Os autores tomaram como base várias centenas de artigos de revista publicados na década de 90 em áreas que abrangem o Controlo Teórico e Aplicado, Engenharia Electrotécnica, Engenharia Química e Engenharia Mecânica.

Cada uma das muitas técnicas de sintonia que têm vindo a surgir, apresenta-se como sendo a mais apropriada para um dado problema específico, verificando-se contudo, alguma escassez de análise comparativa entre diferentes técnicas. Assim sendo, será útil que nas futuras linhas de investigação na área dos controladores *PID*, sejam incluídos campos que integrem medições de desempenho comparativo face a diferentes estratégias de sintonia.

Muhidin Lelic e Zoran Gajic, no seu guia de referência [9], dividem as técnicas de sintonia do controlador *PID* em 6 grupos:

- De Ziegler e Nichols
- Baseadas na resposta ao degrau

- No domínio da frequência
- Utilizando optimização
- Baseadas em Modelo interno
- Outras técnicas de sintonia que não se incluem em nenhum dos pontos anteriores.

No que toca o último ponto [9], podem-se englobar como exemplo, técnicas de sintonia adaptativa baseada no reconhecimento de padrões desenvolvidas por Cao e McAvoy em 1990, técnicas de sintonia baseadas na identificação no domínio do tempo de parâmetros da resposta transitória de processos de 2ª ordem com tempo de atraso, desenvolvida por Hwang em 1995. Como últimos exemplos, em finais da década de 90, Scali e Rachid utilizaram em 1998 uma associação do controlador *PID* com um filtro (*PID+*) capaz de tratar a resposta do processo inverso; em 1999, Brey e Ralston realizaram trabalhos que consistem em desenvolvimentos do controlador *Q-PID* que usa o preditor de Smith.

A técnica de projecto de controladores por posicionamento de pólos dominantes tem como vantagem em relação à técnica de posicionamento arbitrário de pólos, o facto de conduzir a controladores mais simples [8]. Como exemplo de técnica de sintonia por posicionamento de pólos dominantes pode-se apresentar o método de Cohen-Coon; este método bem como as técnicas enumeradas nos pontos acima referidos, são desenvolvidas nas secções seguintes.

1.4.1 Técnicas de Ziegler e Nichols

Os métodos de sintonia baseados nas técnicas de Ziegler e Nichols, tiveram origem nos trabalhos originais [27] destes autores, e em respectivas modificações que se têm verificado ao longo dos tempos. Estes métodos empíricos, estabelecem fórmulas simples para cada parâmetro do controlador *PID* em função de aspectos ligados à dinâmica do processo que se pretende controlar. O estabelecimento de tais fórmulas recorre ou à resposta ao degrau, ou à resposta na frequência; estas duas variantes serão tratadas nas secções seguintes.

1.4.1.1 Técnicas de Ziegler e Nichols baseadas na resposta ao degrau

As técnicas de sintonia do controlador *PID* baseadas na resposta ao degrau partem da análise desta resposta, para o processo que se pretende controlar, como se exemplifica na Fig. 1.3.

A partir da curva da resposta ao degrau é determinado o ponto onde o declive da tangente é máximo; a recta tangente nesse ponto (traço interrompido oblíquo na Fig. 1.3) intercepta os eixos coordenados em dois pontos: *a* e *L*. Os parâmetros *PID* são estabelecidos empiricamente em função das distâncias destes 2 pontos à origem do referencial. Na Tab. 1.1 apresenta-se, para cada uma das 3 configurações, *P*, *PI* e *PID*, as expressões empíricas dos parâmetros k_c , t_i e t_d ; tanto para a técnica baseada na resposta ao degrau, como para a técnica baseada na resposta em frequência, que se apresenta de seguida.

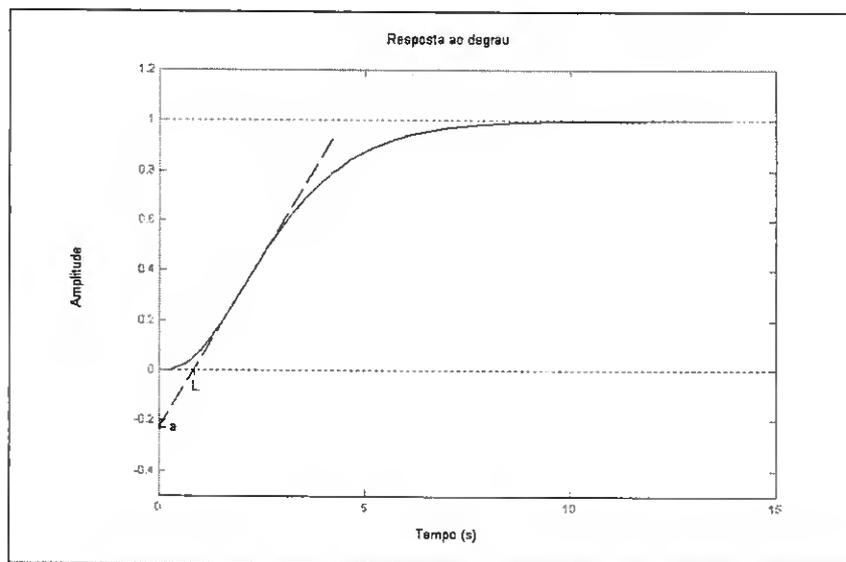


Fig. 1.3 – Resposta ao degrau dum processo genérico.

1.4.1.2 Técnicas de Ziegler e Nichols baseadas na resposta em frequência

De acordo com este método, o processo sob controlo proporcional é levado ao limiar da estabilidade; nessas circunstâncias, os pólos do sistema em malha fechada situam-se sobre o eixo imaginário.

Esta situação pode ser tratada em termos da resposta na frequência, por exemplo recorrendo ao diagrama de Nyquist. Assim, será medida a frequência para a qual o traçado de Nyquist intersecta o semi-eixo real negativo, essa frequência denomina-se frequência

crítica ω_u . O ganho crítico K_u , é o ganho que leva o sistema ao limiar da instabilidade e é o inverso da distância do ponto de cruzamento do semi-eixo real negativo ao ponto -1.

Sob o ponto de vista operacional, esta técnica pode ser implementada submetendo o processo ao controlo proporcional com ganhos baixos. Começando a aumentar o ganho, a partir de valores baixos, atinge-se um ponto em que a resposta ao degrau é oscilatória, o ganho correspondente é o ganho crítico K_u , e o período crítico será o período das oscilações

$$T_u = \frac{2\pi}{\omega_u}.$$

Tal como aconteceu com o método anterior, os parâmetros *PID* são estabelecidos empiricamente em função de K_u e T_u . Na Tab. 1.1 apresentam-se adicionalmente as expressões empíricas para os parâmetros *PID*, referentes a esta técnica.

Configuração →		<i>P</i>	<i>PI</i>	<i>PID</i>
k_c	Degrau	$1/a$	$0.9/a$	$1.2/a$
	Frequência	$0.5K_u$	$0.4K_u$	$0.6K_u$
t_i	Degrau		$3L$	$2L$
	Frequência		$0.8T_u$	$0.5T_u$
t_d	Degrau			$L/2$
	Frequência			$0.125T_u$

Tab. 1.1 – Parâmetros *PID* avaliados pelas regras de Ziegler e Nichols.

1.4.2 Novas técnicas de sintonia do controlador *PID*

As novas técnicas de sintonia do controlador *PID*, de Åström e Hägglund [8], surgem após estes autores terem apresentado uma série de métodos de sintonia e a partir daí, vários ensinamentos iniciais se terem deduzido.

Resumidamente, pode-se dizer [8] que as novas técnicas de sintonia surgem duma forma empírica que consiste, em primeiro lugar, na utilização de uma técnica de sintonia previamente estudada. Seguidamente, é escolhido um conjunto base alargado de funções de transferência representativas de processos industriais. A aplicação da técnica de sintonia seleccionada aos processos do conjunto base conduz a conjuntos de parâmetros do

controlador. De acordo com as novas técnicas de sintonia, o passo seguinte é correlacionar os parâmetros do controlador com algumas medidas características das dinâmicas dos processos.

As regras de Ziegler e Nichols tratadas na secção 1.4.1 baseiam-se em parâmetros dos processos que se pretendem controlar: a , L e T_u , K_u ; respectivamente obtidos pelos métodos da resposta ao degrau e da resposta na frequência.

A partir da análise dos resultados obtidos por este método de sintonia, podem-se resumir [8] as seguintes conclusões:

- Obtenção de respostas bastante oscilatórias
- Regras de sintonia diferentes para respostas para seguimento da entrada e para respostas a perturbações de carga.
- Os resultados obtidos são pobres para sistemas com grandes tempos de atraso normalizados.
- Não está definido nenhum índice de sintonia.

Para os métodos de sintonia que se apresentam de seguida, vai ser utilizado como índice de sintonia, a máxima sensibilidade M_s dado por (1.22), em que $G_p(s)$ e $G_c(s)$ são as funções de transferência do processo e do controlador, respectivamente.

$$M_s = \max_{\omega} \left| \frac{1}{1 + G_p(j\omega)G_c(j\omega)} \right| \quad (1.22)$$

Este índice de desempenho de sintonia é tal que, a menor distância do diagrama de Nyquist de $G_p(s)G_c(s)$ ao ponto -1 , é dada por $1/M_s$. Assim, M_s dá-nos uma medida da robustez do sistema, ou seja, quanto o processo pode ser alterado mantendo-se dentro da estabilidade; valores típicos [8] são $1.2 \leq M_s \leq 2$. Valores elevados de M_s tornam o sistema mais rápido mas menos robusto. No âmbito das novas técnicas de sintonia acabadas de introduzir nesta secção, apresentam-se nas secções seguintes respectivamente técnicas baseadas na resposta ao degrau, e técnicas baseadas na resposta em frequência.

1.4.2.1 Técnicas de sintonia baseadas na resposta ao degrau

Considere-se uma situação em que o processo é estável e cuja resposta ao degrau se assemelha com a que está representada na Fig. 1.3. A dinâmica do processo pode ser caracterizada por 3 parâmetros:

- K_p – ganho estático
- T – atraso aparente
- L – tempo morto aparente

O atraso aparente T corresponde ao instante para o qual a resposta ao degrau atinge 63% do seu valor final. Alternativamente, T pode ser determinado como $T = T_{ar} - L$. T_{ar} designa-se por tempo médio de residência e pode ser determinado [28] recorrendo ao método dos momentos. Tomando por base a Fig. 1.3, designa-se por A_0 a área limitada pela curva da resposta, e a recta horizontal que marca o seu valor final.

$$A_0 = \int_0^{+\infty} y(\infty) - y(\tau) d\tau \quad (1.23)$$

A partir da área A_0 assim definida, deduz-se [28] que $T_{ar} = \frac{A_0}{K_p}$.

Para a aplicação destas novas técnicas de sintonia, deverá ser feita nova definição de parâmetros, ou seja, em vez de se utilizar K_p e T , serão utilizados a e τ , dados por (1.24).

$$a = K_p \frac{L}{T}, \quad \tau = \frac{L}{L+T} = \frac{L}{T_{ar}} \quad (1.24)$$

Se o processo $G_p(s)$ for do tipo I deverá ser tido em conta o seguinte aspecto: seja $G_p(s) = H(s)/s$; designando por X' o parâmetro obtido para $H(s)$, provam-se [8] as expressões (1.25).

$$a = K_p' T_{ar}' = K_p' (L' + T'), \quad L = T_{ar}' = L' + T' \quad (1.25)$$

As regras de sintonia para processos do tipo I são definidas à custa de a , L e τ' , sendo este último parâmetro o tempo morto associado a $H(s)$.

Os 3 parâmetros do controlador *PID* deverão vir na forma adimensional, aplicando-se a correspondente normalização resulta:

1. ak_c – ganho normalizado
2. t_i/L – constante de tempo integral normalizada
3. t_d/L – constante de tempo derivativa normalizada

Como referido anteriormente, na perspectiva do novo método empírico para avaliação dos parâmetros do controlador, deverão ser cumpridos os seguintes pontos:

- utilização duma técnica de projecto para obtenção dos parâmetros do controlador, como por exemplo [8], posicionamento dos pólos.
- aplicação da técnica do ponto anterior a todos os processos do conjunto base de experimentação
- representação gráfica da evolução de cada parâmetro normalizado apresentado nos pontos de 1. a 3., e da ponderação da referência b , em função de τ .
- pesquisar uma função que aproxime cada uma das evoluções gráficas a que se refere o ponto anterior.
- conclui-se [8] que cada parâmetro normalizado pode ser aproximado por uma função de τ genericamente do tipo:

$$f(\tau) = a_0 e^{a_1 \tau + a_2 \tau^2} \quad (1.26)$$

Os valores de a_0 , a_1 e a_2 dependem do parâmetro normalizado que se pretende determinar, e do valor da sensibilidade máxima M_s que se pretende para o sistema. Åström e Hägglund [8] têm esses valores tabelados para $M_s=1.4$ e $M_s=2$.

1.4.2.2 Técnicas de sintonia no domínio da frequência

O novo método de sintonia que utiliza técnicas de frequência segue de muito perto aquele que recorre à resposta ao degrau, e que foi tratado na secção anterior 1.4.2.1.

Com esta abordagem, para cada processo é determinado o ganho crítico K_u e o período crítico T_u .

Os 3 parâmetros do controlador *PID* deverão vir na forma adimensional:

- a) k_c/K_u – ganho normalizado
- b) t_i/T_u – constante de tempo integral normalizada
- c) t_d/T_u – constante de tempo derivativa normalizada

Cada um dos parâmetros a) a c) e ponderação da referência avaliados para o conjunto base de experimentação, será representado graficamente em função da razão de ganhos:

$$\kappa = \frac{1}{K_p K_u}$$

De forma análoga ao que se passa com o método da resposta ao degrau, as evoluções gráfica referidas podem ser aproximadas pela função (1.26), e os valores de a_0 , a_1 e a_2 estão tabelados [8] para $M_s=1.4$ e $M_s=2$.

O método do relé é frequentemente utilizado nas técnicas que recorrem à determinação de ganhos e frequências críticas [8]. Nos métodos que utilizam a resposta na frequência os parâmetros do processo podem ser estimados observando respostas a sinusóides de diferentes frequências. De acordo com estes métodos, o sinal sinusoidal é obtido através duma não linearidade do tipo relé na malha de controlo; introduzindo essa alteração na Fig. 1.1, obtém-se a topologia que se apresenta na Fig. 1.4. Nesta figura $N(a)$ representa a função descritiva da não linearidade (relé) em causa. Na Tab. 1.2 apresentam-se os dois tipos de relé e respectivas funções descritivas. A utilização alternativa do relé com histerese evita que a introdução de algum ruído, mesmo que de baixa amplitude, faça com que o relé comute aleatoriamente como aconteceria com o relé puro.

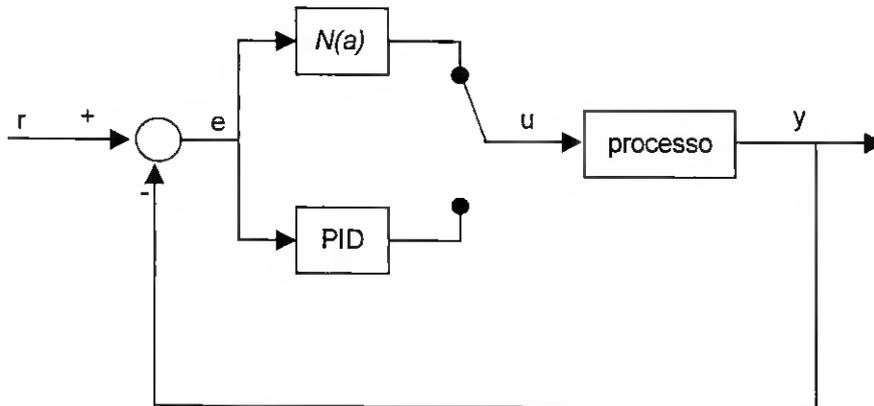


Fig. 1.4 – Malha de controlo *PID* com relé.

	Relé puro	Relé com histerese
Resposta no tempo		
Função descritiva $N(a)$	$N(a) = \frac{4d}{\pi a}$	$N(a) = \frac{4d}{\pi(\sqrt{a^2 - \epsilon^2} + \epsilon j)}$

Tab. 1.2 – Não linearidades do tipo relé.

Quando o controlador necessita de ser sintonizado o comutador que se apresenta na Fig. 1.4 deverá desligar o bloco *PID* e simultaneamente ligar a não linearidade, passando o sistema a oscilar. Quando se atinge o regime estacionário deverão ser medidas a amplitude e o período de oscilação, avaliando-se o ganho crítico K_u e a frequência crítica ω_u . Com estes dois parâmetros da dinâmica do processo pode-se, por exemplo, utilizar as regras de Ziegler e Nichols para se determinar os parâmetros do controlador. Nessa altura o comutador deverá comutar, para que o bloco *PID* seja de novo introduzido na malha.

1.4.3 Técnicas de sintonia utilizando otimização

Os métodos de sintonia que utilizam otimização são bastante poderosos e conceptualmente simples [8]. A estrutura do controlador deverá ser conhecida e fica definida à custa de um vector de parâmetros. As especificações que o sistema controlado deverá satisfazer, assumem o papel de restrições, e deverão ser expressas por inequações onde intervenham os parâmetros do controlador. A especificação mais importante será escolhida como função a otimizar de acordo com um processo de otimização uni-objectivo. Se mais do que uma especificação for considerada no processo de otimização, seremos conduzidos a um processo de otimização multi-objectivo.

Dadas estas características, os métodos de sintonia que utilizam otimização apresentam-se bastante adequados à sintonia do controlador *PID*, dado que, tanto a estrutura deste controlador, como os parâmetros que se pretendem determinar, são conhecidos.

As principais desvantagens desta metodologia de controlo, prendem-se com as limitações inerentes ao próprio processo de otimização, conforme se descreve nos pontos seguinte:

- escolha do critério e verificação das restrições: frequentemente o processo de otimização conduz a uma situação em que o critério é óptimo mas algumas restrições são violadas por não terem sido tidas em conta; o controlador assim obtido torna-se desadequado.
- existência de óptimos locais: na maior parte das vezes a geometria da função a otimizar é desconhecida, consoante o método de otimização que se utiliza poderá ocorrer paragem do processo de otimização sem que a solução encontrada seja um óptimo global.
- peso computacional: dependendo do problema que esteja a ser tratado, processos de otimização conduzem frequentemente a grande peso computacional.
- análise numérica: os processos de otimização deparam-se com alguma frequência com os problemas que surgem da análise numérica.

1.4.4 Técnicas de sintonia baseadas em modelo interno

De acordo com a abordagem apresentada por Åström e Hägglund [8], o Controlador baseado no Modelo Interno (*IMC*) pode ser visto como uma estratégia de controlo que se insere dentro dos métodos analíticos de sintonia. Este tipo de métodos de sintonia, assentam sobre o princípio de que a função de transferência em malha fechada $G_0(s)$ é determinada a partir das especificações de funcionamento. Então, tomando por base a Fig. 1.1 em que o controlador e o processo têm como função de transferência $G_c(s)$ e $G_p(s)$ respectivamente, pode-se deduzir (1.27) $G_c(s)$, em função de $G_p(s)$ e de $G_0(s)$.

$$G_c(s) = \frac{1}{G_p(s)} \frac{G_0(s)}{1 - G_0(s)} \quad (1.27)$$

A abordagem referida [8] regista que esta classe de métodos de sintonia traz alguns problemas, dado que, de acordo com (1.27) os pólos e zeros do processo são cancelados pelo controlador.

O princípio geral do *IMC* pode ser aplicado para sintonia do controlador *PID*. Na Fig. 1.5 apresenta-se um diagrama de blocos dum sistema sujeito a este controlador.

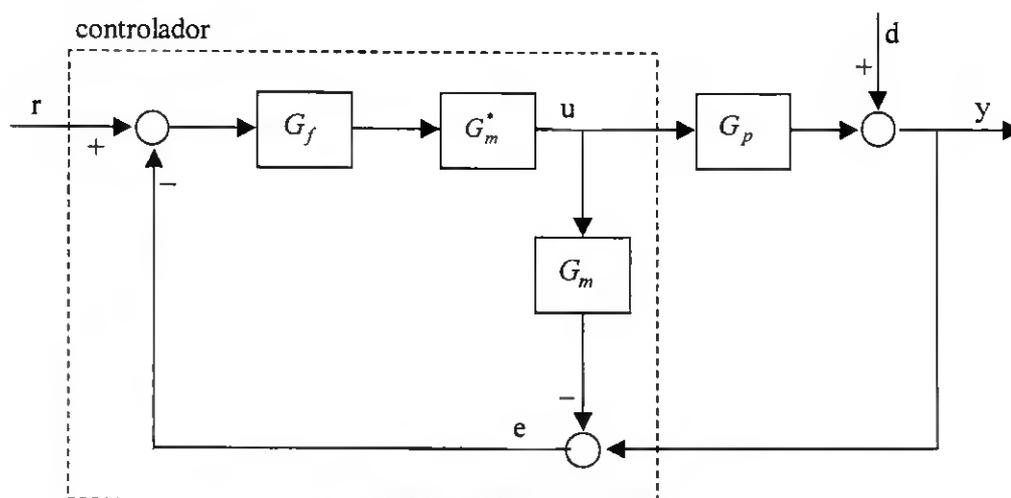


Fig. 1.5 – Diagrama de blocos para sistema sujeito ao controlador *IMC*.

Nesta figura G_m é um modelo do processo G_p e G_m^* deverá ser aproximadamente inverso de G_m ; G_f é um filtro passa baixo utilizado para que o sistema seja menos sensível a erros de modelação: $G_f(s) = 1/(1+sT_f)$, com T_f dado pelo projecto.

O controlador obtido pelo princípio *IMC* pode ser representado pela função de transferência (1.28).

$$G_c(s) = \frac{G_f(s)G_m^*(s)}{1 - G_f(s)G_m^*(s)G_m(s)} \quad (1.28)$$

Tal como seria de esperar dum método de sintonia analítico, este controlador cancela os pólos e zeros do processo.

Este princípio de controlo [8], dum modo geral, conduz-nos a controladores de elevada ordem; contudo, sob algumas condições, obtém-se controladores *PI* ou *PID*.

Como exemplo [8], considere-se um processo de 1ª ordem com tempo de atraso (1.29) e um filtro de 1ª ordem com constante de tempo T_f dado por (1.30).

$$G_p(s) = \frac{K_p}{1+sT} e^{-sL} \quad (1.29)$$

$$G_f(s) = \frac{1}{1+sT_f} \quad (1.30)$$

Considerando o atraso modelado pelo desenvolvimento em série de Taylor truncando todos os termos de ordem superior à 1ª, ($e^{-sL} \approx 1-sL$), a partir de (1.28) obtém-se (1.31) onde facilmente se reconhece um controlador *PI*.

$$G_c(s) = \frac{1+sT}{K_p s(L+T_f)} \quad (1.31)$$

Alternativamente, utilizando a aproximação de Padé de 1ª ordem para o atraso

$(e^{-sL} \approx \frac{1 - \frac{sL}{2}}{1 + \frac{sL}{2}})$, obtém-se um controlador *PID*, conforme se pode observar na expressão

(1.32).

$$G_c(s) = \frac{\left(1 + \frac{sL}{2}\right)(1 + sT)}{K_p s(L + T_f)} \quad (1.32)$$

1.4.5 Técnica de Cohen-Coon

A técnica proposta por estes autores considera que o processo a controlar deverá ser representado por uma função de transferência de 1ª ordem com tempo de atraso como a que se apresenta na expressão (1.33).

$$G(s) = K \frac{e^{-Ls}}{1 + Ts} \quad (1.33)$$

No que se refere ao controlador *PID*, os 3 pólos em malha fechada (2 complexos conjugados) têm igual módulo. O valor do módulo é dimensionado de tal forma que seja mínimo o integral do erro face a uma perturbação na carga.

A partir dos parâmetros da expressão (1.33), definindo-se $a = KL/T$ e $\tau = L/(L+T)$, os parâmetros *PID* são determinados de acordo com a Tab. 1.3.

A confrontação das expressões da Tab. 1.3. com as equivalentes que se obtêm pelas regras da Ziegler e Nichols utilizando-se a resposta ao degrau, permite concluir que os valores são semelhantes para pequenos valores de τ .

Configuração →	<i>P</i>	<i>PI</i>	<i>PD</i>	<i>PID</i>
k_c	$\frac{1}{a} \left(1 + \frac{0.35\tau}{1-\tau} \right)$	$\frac{0.9}{a} \left(1 + \frac{0.92\tau}{1-\tau} \right)$	$\frac{1.24}{a} \left(1 + \frac{0.13\tau}{1-\tau} \right)$	$\frac{1.35}{a} \left(1 + \frac{0.18\tau}{1-\tau} \right)$
t_i		$\frac{3.3-3.0\tau}{1+1.2\tau} L$		$\frac{2.5-2.0\tau}{1-0.39\tau} L$
t_d			$\frac{0.27-0.36\tau}{1-0.87\tau} L$	$\frac{0.37-0.37\tau}{1-0.81\tau} L$

Tab. 1.3 – Parâmetros *PID* avaliados pela técnica de Cohen-Coon.

1.5 DEFINIÇÃO DE SINTONIA AUTOMÁTICA

Entende-se por sintonia automática [29], a acção segundo a qual se determina parâmetros dum controlador ligado a um processo, sem que ocorra intervenção do operador para além da acção de inicialização do funcionamento do sistema controlado.

O sistema de sintonia automática e o controlador são duas entidades distintas, podendo inclusivamente, estar fisicamente separadas. Em qualquer instante de funcionamento da malha de controlo, a determinação do sinal de controlo não pode depender da presença do sistema de sintonia automática.

O conceito de sintonia automática difere do conceito de controlo adaptativo [30]. De acordo com o controlo adaptativo os parâmetros do controlador são avaliados sem intervenção do operador, ocorrendo a sintonia numa forma contínua. Num sistema de sintonia automática, deverá haver a capacidade deste sugerir ao operador que se proceda a uma actualização dos parâmetros do controlador. Nestas circunstâncias, a sintonia é iniciada directamente pelo operador, ou pelo desencadeamento de determinados eventos estabelecidos na documentação que levem à sintonia. Quando a sintonia ocorre automaticamente na sequência de determinadas condições se tornarem verdadeiras, (por exemplo o valor de alguma medida de erro em determinado instante) estamos perante uma situação híbrida entre a sintonia automática e a adaptação contínua. Nestas circunstâncias a

lógica que desencadeia a sintonia deverá estar devidamente documentada e configurada; deverá igualmente ser possível ligar e desligar essa funcionalidade do controlador.

A forma como um sistema de sintonia automática actua, não deverá diferir substancialmente daquela levada a cabo por um operador que estivesse incumbido da mesma tarefa. Nessas circunstâncias os seguintes pontos deverão ser observados:

- tomada de conhecimento do comportamento do processo a controlar: o operador deverá por observação ou simulação do processo, recolher medidas relevantes que permitam caracterizar o seu comportamento.
- verificação do funcionamento do sistema antes da ocorrência da sintonia: a partir do conhecimento recolhido no ponto anterior, verificar como irá funcionar a malha de controlo. Nestas circunstâncias, o funcionamento verificado deverá ser confrontado com aquele que se pretende observar. Deve ser tido em conta o facto da descrição recolhida no ponto anterior poder ser incompleta ou estar sujeita a imprecisões.
- sintonia do controlador: face à diferença apontada no ponto anterior entre o funcionamento da malha de controlo verificado e aquele que se pretende como objectivo, o operador deverá alterar, (sintonizar) os parâmetros do controlador, com vista à concretização do funcionamento pretendido.

O procedimento composto pelos três pontos acabados de descrever denomina-se metodologia de sintonia. Um sistema de sintonia automática do controlador será a implementação (concretização) da metodologia de sintonia, de tal forma que funcione automaticamente.

1.5.1 Sintonia automática do controlador *PID* com base na resposta em frequência

Pagola e Pecharromán desenvolveram um método [31] de sintonia do controlador PI que recorre à determinação de dois pontos da resposta na frequência, respectivamente, frequência crítica e frequência de corte, em dois passos, tal como se apresenta de seguida.

- identificação da frequência crítica: Para a situação em malha aberta, com o processo $P(s)$ sujeito a controlo proporcional, é identificado o ganho para o qual a fase vale -180° . O ganho crítico, e o período crítico são dados respectivamente por K_u e T_u , de acordo com as expressões (1.34).

$$P(j\omega_u) = -A_u \quad K_u = \frac{1}{A_u} \quad T_u = \frac{2\pi}{\omega_u} \quad (1.34)$$

- identificação da frequência de corte: Para vários processos utilizados pelos autores, este ponto não existe quando se usa apenas o controlo proporcional; nessas circunstâncias o controlador PI dado por (1.35) deve ser considerado.

$$C(s) = K_p \frac{1+T_i s}{T_i s} \quad K_p = \alpha K_u \quad T_i = \beta T_u \quad (1.35)$$

Os parâmetros α e β foram escolhidos [31] respectivamente $\alpha = 0.3$ e $\beta = 1.6$. Será identificada a frequência ω_0 para a qual $|C(j\omega_0)P(j\omega_0)| = 1$; então, A_0 e φ_0 serão tais que: $P(j\omega_0) = A_0 e^{j\varphi_0}$.

Existem neste momento os 4 parâmetros: ω_u , K_u , ω_0 e φ_0 ; a frequência e a amplitude são normalizadas de acordo com a expressão (1.36).

$$\omega_{0u} = \frac{\omega_0}{\omega_u} \quad A_{0u} = \frac{A_0}{A_u} \quad (1.36)$$

Tomando-se [31] por base as expressões (1.34), (1.35) e (1.36) prova-se a relação (1.37).

$$A_{0u} = \frac{1}{\alpha \left| 1 - \frac{1}{2\pi\beta\omega_{0u}} j \right|} \quad (1.37)$$

Os processos utilizados no teste desta metodologia incluem os que se apresentam em [8], bem como outros exemplos publicados na literatura da área; tratam-se de processos do tipo 0 e do tipo I com e sem tempo de atraso.

A representação gráfica de ω_{0u} em função de φ_0 , para todos os processos do conjunto de teste, permite verificar grande similaridade entre as evoluções para os processos do tipo 0 e para os processo do tipo 1. Foi possível concluir [31] que ω_{0u} e φ_0 estão fortemente correlacionados para os processos que compõem o conjunto de teste, fornecendo assim basicamente a mesma informação.

O ganho normalizado A_{0u} , dado por (1.37), permitiu [31] calcular as propriedades do modelo do processo de 2ª ordem com tempo de atraso, sem se recorrer à utilização das margens de estabilidade.

Tendo em conta as definições apresentadas na Fig. 1.1, ponderando a referência com um factor b para evitar excessiva sobre-elevação, o controlador usado tem dois graus de liberdade, sendo o sinal de controlo $u(t)$ dado pela expressão (1.38):

$$u(s) = k_c \left(br(s) - y(s) + \frac{1}{t_i s} (r(s) - y(s)) \right) \quad (1.38)$$

O processo de sintonia visa a optimização do *ITAE* segundo duas perspectivas:

- rejeição de perturbações na resposta: pretende-se determinar o máximo da função sensibilidade M_s , com a restrição $M_s \leq 2$: $M_s = \max_{\omega \geq 0} \left| \frac{1}{1 + C(j\omega)P(j\omega)} \right|$. Na maior parte dos casos esta optimização conduz a bons resultados, embora não suficientemente amortecidos. Repare-se que a ponderação da referência b não afecta a resposta face a perturbações, então, neste ponto são determinados os outros dois parâmetros do controlador *PI*: k_c e t_i .
- erro da resposta em relação à referência: são tomados os parâmetros K_p e t_i avaliados no ponto anterior; é optimizado o *ITAE* tendo em conta que se pretende que a resposta siga tão próximo quanto possível a referência; o resultado desta optimização permite a determinação de b . O conjunto de parâmetros obtido não é um óptimo absoluto para esta optimização mas não se afasta muito dessa situação.

Os resultados obtidos com esta metodologia [31] apresentam melhor desempenho do que aqueles referidos por Åström e Hägglund em 1995 [8] e por Ziegler e Nichols em 1942

[27]. A comparação com os resultados apresentados anteriormente, pelos mesmos autores [32], revelam diferenças menos significativas em grande parte dos casos.

1.5.2 Auto-sintonia do controlador *PID* usando técnicas de *soft-computing*

1.5.2.1 Auto-sintonia do controlador *PID* usando Algoritmos Evolutivos

Trabalhos desenvolvidos por A. H. Jones e P. B. de Moura Oliveira [33] apresentam um processo de auto-sintonia do controlador *PID* digital utilizando algoritmos genéticos. De acordo com a técnica apresentada, os dados gerados pelo processo tanto em malha aberta como em malha fechada, servirão para construir o correspondente modelo *ARMA* genético discreto, sendo requerido apenas que o sistema tenha atingido o regime estacionário e que os dados medidos estejam uniformemente espaçados o tempo. A função de custo a minimizar será uma medida integral da discrepância no tempo entre a saída do processo e a resposta do modelo, quando sujeitos à mesma sequência de entrada.

A partir deste modelo, os algoritmos genéticos irão ser utilizados para se efectuar uma sintonia do *PID* em diferido. Com os parâmetros obtidos com esta sintonia, o controlador é colocado em linha.

A técnica base acabada de referir, foi aplicada nos mesmos moldes e pelos mesmos autores A. H. Jones e P. B. de Moura Oliveira [34], à sintonia do controlador *PI* Preditor de Smith. De acordo com esta abordagem o modelo *ARMA* do processo é usado para que, conjuntamente com os algoritmos genéticos e técnicas de simulação, o controlador *PI* Preditor de Smith seja sintonizado em diferido. O controlador de Smith depende fortemente da adequação do modelo do processo, bem como da exactidão do tempo de atraso. A metodologia de identificação de sistemas usando algoritmos genéticos [35] provou ser bastante adequada na avaliação do tempo de atraso. Este controlador uma vez sintonizado em diferido é inserido na malha de controlo para funcionamento em linha.

Os métodos acabados de referir foram melhorados por A. H. Jones, Ying-Chih Lin, P.B. de Moura Oliveira e S.B. Kenway [36], no sentido de se estabelecer técnicas de auto-sintonia numa malha de controlo dual. Desta forma, os algoritmos genéticos são utilizados para sintonizar dois controladores, e definir os correspondentes modelos de sintonia.

De acordo com esta técnica, pretende-se otimizar dois objectivos:

- seguimento duma referência,
- rejeição de perturbações aditivas na medição da saída.

A malha de controlo dual incorpora dois controladores que actuam em sequência:

- C1 – Controlador digital *Feedforward* responsável pelo seguimento da referência
- C2 – Controlador *PID* digital responsável pela rejeição de perturbações à saída.

A sintonia individual de cada um dos referidos controladores faz-se otimizando os respectivos parâmetros², utilizando-se para tal as operações básicas dos algoritmos genéticos.

Tal como na abordagem preliminar [33] referida anteriormente, são estabelecidos modelos genéticos que simulam a sintonia do controlador *Feedforward* e a sintonia do controlador *PID*. Os controladores em modo dual são primeiramente sintonizados em diferido e posteriormente colocados em linha.

Tanto a técnica base [33], como as variantes [34] e [36] foram aplicadas ao controlo laboratorial dum permutador de calor; os resultados foram comparados com os que se obtêm usando-se o método de auto-sintonia de Åström [37]. Concretamente, a metodologia que incorpora o Preditor de Smith [34] permite uma resposta cujo transitório se extingue mais rapidamente do que a que se verifica quando se utiliza apenas o controlador *PID* [33]. Todas as técnicas apresentaram-se preferíveis quando comparadas com a técnica de Åström [37].

O projecto de Sistemas de Controlo assistido por Computador (*CACSD*) utilizando técnicas evolutivas é direccionado fortemente para a obtenção da solução, havendo o risco de se negligenciar alguns aspectos importantes dos processos evolutivos. Devido a este facto, a linha de investigação que visa a preservação da diversidade genética assume especial importância.

² De acordo com [36], são otimizados 5 parâmetros para C1, e 3 para C2.

P. B. de Moura Oliveira e A. H. Jones [38] salientam o facto de nos algoritmos genéticos simples existir o risco de acontecer convergência prematura [39]. Este fenómeno consiste [38] na tendência natural para a perda da diversidade genética, caindo-se rapidamente em populações homogéneas, estando a maioria dos indivíduos localizada nas proximidades dum óptimo local; os operadores genéticos habituais por vezes dificilmente resolvem esta situação. Uma das possibilidades de se evitar a convergência prematura, e como tal aumentar a diversidade genética, é recorrendo ao uso da co-evolução artificial [40]; obtendo-se soluções mais robustas.

Estas técnicas recorrem à definição de pelo menos duas populações, e integram as seguintes abordagens:

- Competitiva: Modelo predador presa, segundo o qual a população predadora desenvolve mecanismos ofensivos contra a presa, como consequência a população das presas desenvolve mecanismos defensivos, resultando um processo global de evolução de ambas as populações.
- Cooperativa: Cada população evolui separadamente tendo em conta as respectivas funções de aptidão específicas. As soluções encontradas são combinadas num segundo estado de evolução que se desenvolve sob controlo duma população de árbitros.

Os trabalhos realizados por P. B. de Moura Oliveira e A. H. Jones [38] exploram as técnicas co-evolutivas no projecto de controladores *PID*.

A abordagem competitiva surge no projecto do controlador *PID SISO* com incertezas nos parâmetros. Para se atingir este objectivo são considerados dois estados de evolução. O 1º estado de evolução tem como principal objectivo determinar um conjunto de processos difíceis de controlar representados no espaço de incertezas dos parâmetros. No 2º estado de evolução, uma população de controladores evolui contra uma nova população de processos que se mantém fixa e deverá cobrir os modelos no espaço das incertezas. Cada indivíduo da população dos controladores é testado com todos os indivíduos da população dos processos. Para cada par, o valor da função de custo a otimizar é guardado numa matriz bi-dimensional; em que, custo de cada controlador C_i é o máximo de todos os custos da linha i .

A abordagem cooperativa, juntamente com os algoritmos genéticos simples, é utilizada na identificação dos processos *MIMO* e auto-sintonia de controladores *PI* multivariáveis.

Os algoritmos genéticos provaram ser bastante eficazes no projecto de controladores *PID* óptimos para rejeição de perturbações aditivas à saída, conforme trabalhos de Renato A. Krohling e Joost P. Rey [41]; com esta abordagem os autores generalizam os métodos inicialmente apresentados em [42].

A Fig. 1.6 apresenta uma variante relativamente à situação referida a na Fig. 1.2; considera-se ausência de ruído e a perturbação aditiva $D_y(s)$ tomada à saída do processo é ponderada por um filtro passa baixo $W_d(s)$.

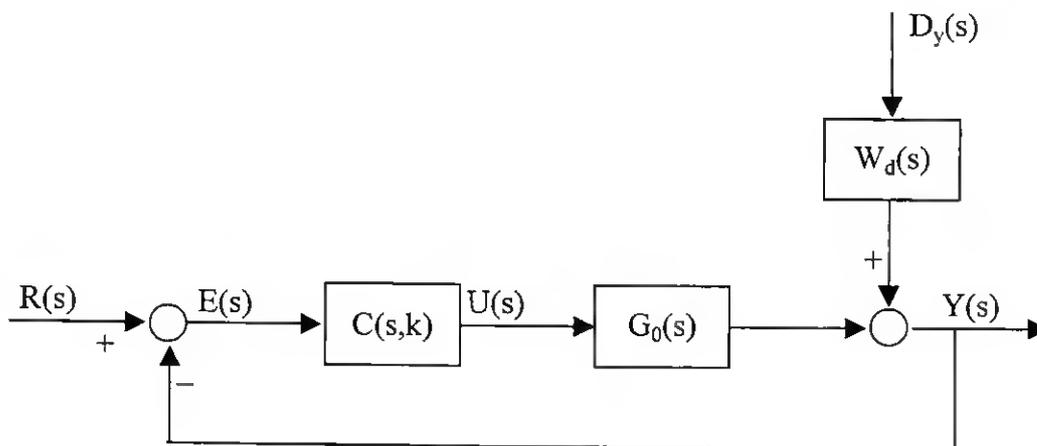


Fig. 1.6 - Sistema realimentado com perturbação externa à saída.

Na presente situação considera-se adicionalmente:

$C(s,k)$ – controlador de estrutura fixa, sendo $k = [k_1 \ k_2 \ \dots \ k_m]$ o vector de parâmetros do controlador,

$G_0(s)$ – processo a controlar.

A rejeição da perturbação pode ser descrita recorrendo à norma H_∞ como se apresenta na expressão (1.39).

$$\left\| \frac{Y(s)}{D_y(s)} \right\|_{\infty} = \left\| \frac{w_d(s)}{1 + C(s, k)G_0(s)} \right\|_{\infty} < \gamma \quad (1.39),$$

sendo $\gamma < 1$ o nível de rejeição e a norma H_{∞} definida como:

$$\|G(s)\|_{\infty} = \max_{\omega \in [0, +\infty[} |G(j\omega)| \quad (1.40).$$

Aplicando a norma H_{∞} à restrição da rejeição da perturbação, tem-se:

$$\left\| \frac{w_d(s)}{1 + C(s, k)G_0(s)} \right\|_{\infty} = \max_{\omega \in [0, +\infty[} (\alpha(\omega, k))^{0.5} \quad (1.41),$$

definindo-se:

$$\alpha(\omega, k) = \frac{w_d(j\omega)w_d(-j\omega)}{(1 + C(j\omega, k)G_0(j\omega))(1 + C(-j\omega, k)G_0(-j\omega))} \quad (1.42).$$

Desta forma, a condição de rejeição será dada por (1.43):

$$\max_{\omega \in [0, +\infty[} (\alpha(\omega, k))^{0.5} < \gamma \quad (1.43).$$

O projecto do controlador consiste na resolução dum problema de minimização com restrições. Pretende-se determinar o vector k que minimize o índice de desempenho $I=ITSE$, sujeito à restrição definida por (1.43). A solução deste problema será obtida recorrendo-se aos algoritmos genéticos com codificação real.

Serão utilizados dois algoritmos genéticos que actuam em alternância:

- GA1: responsável pela minimização do $ITSE$.
- GA2: responsável pela maximização da rejeição de perturbações $\alpha(\omega, k)$.

O método [41] foi aplicado a um processo de 2ª ordem com um pólo na origem, com dois tipos de perturbações à saída: sinusoidal e degrau unitário. As respostas a referências do tipo degrau são praticamente sobreponíveis às que se obtêm com perturbações nulas.

1.5.2.2 Auto-sintonia do controlador *PID* usando lógica difusa

Os trabalhos de A. Visioli [43] apresentam comparativamente vários métodos que integram a lógica difusa para a sintonia do controlador *PID*. Nas metodologias apresentadas foram utilizados os algoritmos genéticos [44] para a sintonia dos parâmetros dos sistemas de inferência difusos, tais como factores de escala e perfil das funções membro.

O problema de sintonia consiste na determinação dos parâmetros *PID* standard de tal forma que sejam satisfeitas especificações de controlo tais como:

- seguimento de uma referência,
- atenuação de perturbações
- robustez na modelação de incertezas
- rejeição de ruído mensurável

A função objectivo que se pretende minimizar é o integral do valor absoluto do erro; desta forma, é tido em conta simultaneamente o tempo de crescimento, a sobre-elevação e o tempo de estabelecimento.

Os parâmetros avaliados segundo as regras de Ziegler e Nichols conduzem a uma boa atenuação de perturbações, mas descuidam a margem de fase, levando a elevados valores de sobre-elevação e tempo de estabelecimento. Para se evitar estas situações, nomeadamente para se reduzir a sobre-elevação, a referência é ponderada por um factor $b < 1$. A partir da lei de controlo *PID* típica (1.3), definindo-se $k_d = k_c t_d$ e $k_i = \frac{k_c}{t_i}$, o sinal de controlo $u(t)$ é dado por (1.44).

$$u(t) = k_c (br(t) - y(t)) + k_d \frac{de(t)}{dt} + k_i \int_b e(\tau) d\tau \quad (1.44)$$

Obtém-se assim um controlador com dois graus de liberdade, havendo uma parte dedicada à atenuação de perturbações, e outra parte relacionada com a capacidade de seguimento da entrada.

A partir da transformada de Laplace da expressão (1.44), podem-se definir os blocos G_{ff} e G_c :

$$G_{ff} = k_c \left(b + \frac{1}{st_i} + st_d \right) \quad (1.45),$$

$$G_c = k_c \left(1 + \frac{1}{st_i} + st_d \right) \quad (1.46),$$

que se integram na seguinte topologia de controlo da Fig. 1.7.

Os vários métodos comparados em [43] resumem-se de seguida:

- IFE – Controlador PID incremental difuso [45] : Os valores dos parâmetros PID inicialmente determinados pelas regras de Ziegler e Nichols, seriam modificados durante a resposta transitória, em função do erro e da sua taxa de variação, de acordo com a seguinte lei:

$$X = X + k_x CV \left\{ e(t), \dot{e}(t) \right\} \quad (1.47),$$

sendo:

X – cada parâmetro PID ,

CV – saída do sistema de inferência difusa baseado na matriz de regras difusas de Macvicar-Whelan [46].

k_x – constantes definidas para cada parâmetro PID que determinam as gamas de variação.

Deve ser notado que a escolha das constantes k_x bem como dos factores de escala que multiplicam $e(t)$ e $\dot{e}(t)$, fica a cargo do utilizador o que constitui uma desvantagem do método.

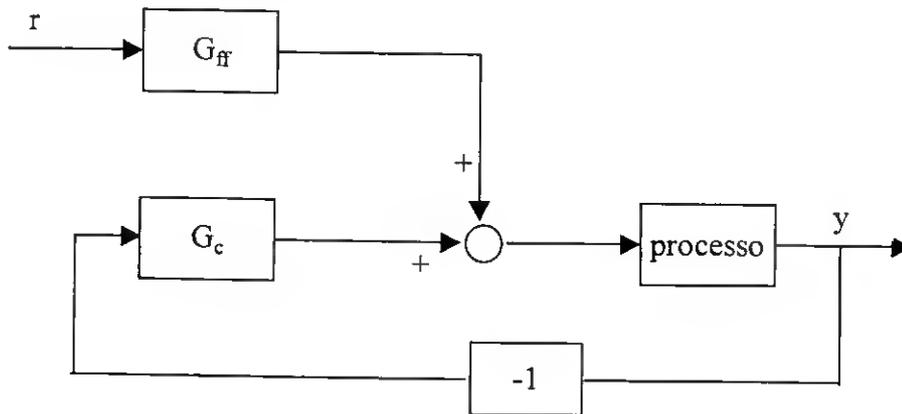


Fig. 1.7 – Controlador *PID* com dois graus de liberdade e ponderação da referência.

- SSP – Auto-sintonia de parâmetro simples [47]: Os parâmetros *PID* são determinados pelas regras de Ziegler e Nichols parametrizadas.

$$k_p = 1.2\alpha(t)k_u, \quad T_i = 0.75 \frac{1}{1+\alpha(t)}t_u, \quad T_d = 0.25T_i \quad (1.48),$$

sendo:

k_u – ganho crítico,

t_u – período crítico.

O parâmetro α é determinado recursivamente em linha, por um sistema de inferência difuso. O valor inicial será 0.5 para que os parâmetros *PID* coincidam com os de Ziegler e Nichols.

- FGS – Tabulação difusa do ganho [48] : Os 3 parâmetros são dados por:

$$\begin{aligned} k_c &= (k_{c,\max} - k_{c,\min})k'_c + k_{c,\min} \\ k_d &= (k_{d,\max} - k_{d,\min})k'_d + k_{d,\min} \\ k_i &= \frac{k_c^2}{\alpha k_d} \end{aligned} \quad (1.49),$$

sendo k'_c , k'_d e α determinados por um mecanismo difuso.

$k_{c,\max}$, $k_{c,\min}$, $k_{d,\max}$ e $k_{d,\min}$ são ajustados de tal forma que k_c e k_d se situem entre 0 e 1.

- FSW – Ponderação difusa da referência [49]: A referência é ponderada por uma função difusa. Os 3 parâmetros *PID* são determinados pelas regras de Ziegler e Nichols para garantirem uma boa rejeição a perturbações, mantendo-se fixos. O sinal de controlo é dado por (1.44) e a ponderação da referência deixa de ser constante para ser dada por (1.50).

$$b(t) = w + f(t) \quad (1.50),$$

sendo:

w – constante positiva < 1 ,

$f(t)$ – saída dum sistema de inferência difuso que consiste em 5 funções membro triangulares para as duas entradas $e(t)$ e $\dot{e}(t)$, e 9 funções membro triangulares para a saída.

As regras difusas são baseadas na matriz de Macvicar-Whelan [46], tal como acontecia com o controlador *PID* incremental difuso [45].

- Controlador difuso (tipo *PID*): Contrariamente aos métodos apresentados até agora, de acordo com esta estratégia [50], o sistema de inferência difuso não será utilizado para determinar os parâmetros *PID* mas sim para determinar directamente o sinal de controlo, conforme se esquematiza na Fig. 1.8.

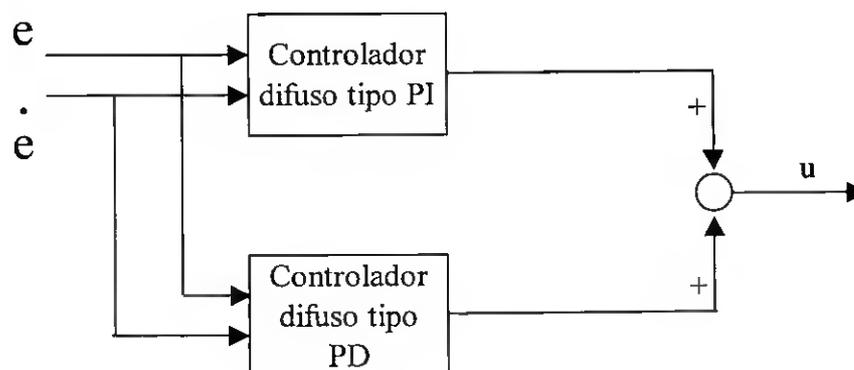


Fig. 1.8 – Controlador difuso tipo *PID*.

Os métodos anteriormente apresentados podem ser confrontados com o controlador *PID* não linear [51] no qual não é utilizada a lógica difusa. De acordo com as definições já apresentadas, o sinal de controlo é dado pela expressão (1.51).

$$u(t) = k_c \left(L + (1-L)|e(t)| \right) e(t) + k_d \frac{de(t)}{dt} + k_i \int_0^t e(\tau) d\tau \quad (1.51)$$

Com esta abordagem são sintonizados 4 parâmetros; os 3 parâmetros *PID* e adicionalmente o parâmetro $L \in [0,1]$ que representa o grau de não linearidade. $L=1$ conduz ao *PID* linear, $L=0$ conduz ao grau máximo de não linearidade.

Os métodos descritos foram testados para um lote de processos de 2ª, 3ª e 4ª ordem com e sem tempo de atraso e sem zeros. Após a fase de sintonia com utilização dos algoritmos genéticos, foi produzida a resposta ao degrau e analisados indicadores de desempenho tais como o *IAE*, sobre-elevação e tempo de crescimento. A apreciação global dos resultados permitiu concluir que o método de ponderação difusa da referência (*FSW*) é preferível em relação aos outros métodos.

1.5.2.3 Auto-sintonia do controlador *PID* usando redes neuronais

Os trabalhos de Q. Li, S. K. Tso e A. N. Poo [52], apresentam uma estratégia de auto-sintonia adaptativa do controlador *PID*, sem a necessidade de se conhecer previamente a dinâmica do processo. Para tal, são utilizadas duas redes neuronais do tipo *MLP*, treinadas com o algoritmo de retropropagação do erro com termo de momento, responsáveis respectivamente pelo mapeamento do processo, e pela implementação do controlador *PID*, conforme a Fig. 1.9.

O objectivo da sintonia do controlador será a determinação em linha dos ganhos que minimizam a função de custo dada pela expressão (1.52), cujo gradiente é dado pela expressão (1.53).

$$J = \frac{1}{2} [r(k) - y(k)]^2 \quad (1.52)$$

$$\frac{\partial J}{\partial w_{pid}} = - \frac{\partial J}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial w_{pid}} \quad (1.53)$$

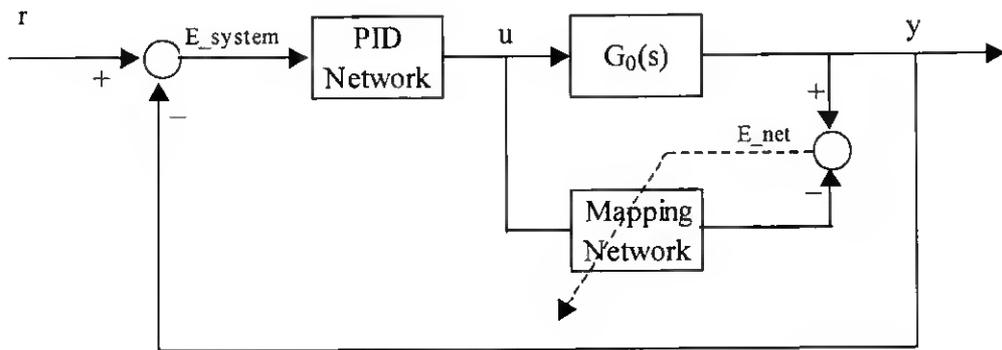


Fig. 1.9 - Auto-sintonia do controlador *PID* usando redes neuronais.

Os termos $\frac{\partial y}{\partial u}$ e $\frac{\partial u}{\partial w_{pid}}$, serão determinados pela dinâmica do processo e pelo controlador, respectivamente. Para se concretizar esse objectivo, é treinada em linha a *Mapping Network* com duas camadas, cujo nº de entradas depende da ordem do sistema, para emular o processo a controlar. Tendo em conta a Fig. 1.9, para este caso, a função de custo vem dada pela expressão (1.54) e o respectivo gradiente pela expressão (1.55).

$$J_1 = \frac{1}{2} (E_{net})^2 \tag{1.54}$$

$$\frac{\partial J_1}{\partial w_{pid}} = - \frac{\partial J_1}{\partial y_{net}} \frac{\partial y_{net}}{\partial u} \frac{\partial u}{\partial w_{pid}} \tag{1.55}$$

O controlador *PID* adaptativo (*PID network*) resume-se a uma rede com uma entrada $e(t)$ e uma saída $u(t)$, para a qual os pesos w_p , w_i e w_d representam os ganhos do controlador. A entrada e a saída relacionam-se de acordo com a expressão (1.56) do controlador *PID* típico.

$$u(k) = w_p e(t) + w_i \int e(t) dt + w_d \dot{e}(t) \tag{1.56}$$

A rede neuronal responsável pelo mapeamento mostrou-se eficaz na captura de não linearidades ou de incertezas nos parâmetros do modelo. Apenas deverá ser conhecida uma estimativa da ordem do sistema; assim, esta estratégia mostrou-se favorável relativamente à que se apresenta na referência [53]. Devido à sua simplicidade, a presente abordagem [52] mostrou-se igualmente preferível relativamente à que se apresenta em [54]. A simulação

numérica dum processo não linear, mostrou que o controlador rapidamente actualiza os ganhos face a variações do processo, mantendo um bom desempenho.

A auto-sintonia do controlador *PID* para estabilização do pêndulo invertido foi abordada por S. Omatu, T. Fujinaka, Y. Kishida, e M. Yoshika [55]. De acordo com estes autores, o esquema básico de controlo que se apresenta na Fig. 1.10 foi estendido a um sistema *SIMO* (duas saídas).

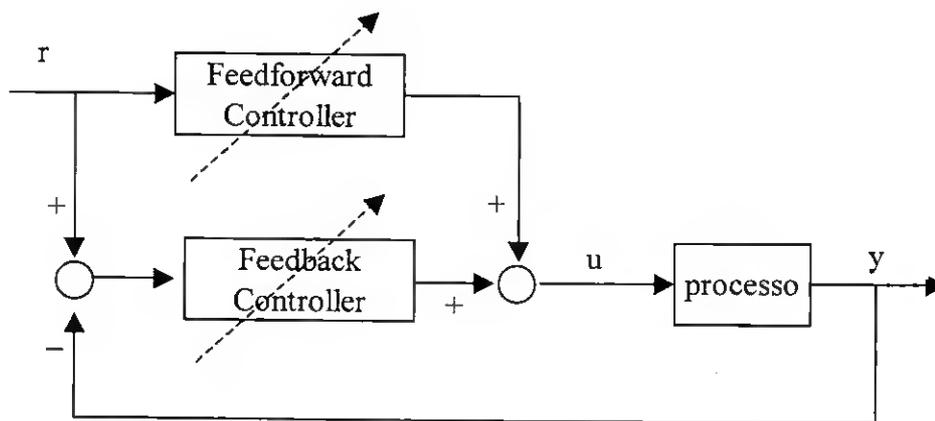


Fig. 1.10 – Esquema geral de controlo usando redes neuronais.

São utilizadas redes neuronais do tipo *MLP*, com um algoritmo de treino modificado em relação ao algoritmo da retro-propagação do erro, para se evitar a queda em mínimos locais.

O controlador neuronal de auto-sintonia apresenta 3 saídas que fornecem os 3 parâmetros *PID* e o número de entradas depende do problema específico que está sendo tratado.

O método mostrou-se robusto, face a perturbações que se verifiquem à entrada e generalizável a vários problemas de controlo que incluem o controlador *PID* usual, nomeadamente, ao problema de controlo do pêndulo duplo invertido [56].

1.6 CONCLUSÕES

Neste 1º capítulo foi enquadrada a dissertação, apresentada a motivação e as contribuições que dela resultam.

Foram descritos alguns fundamentos sobre o controlador *PID* e analisadas cada uma das suas acções. Ainda dentro da secção dedicada ao controlador *PID*, foram apresentados alguns aspectos ligados à sua implementação digital.

As restantes secções dedicaram-se ao estado da arte sobre os métodos de sintonia do controlador *PID*. É definido o conceito de sintonia automática e são apresentados alguns métodos que aplicam este tipo de sintonia ao controlador *PID*.

2. FUNDAMENTOS DE TÉCNICAS DE *SOFTCOMPUTING*

2.1 INTRODUÇÃO

No capítulo anterior foi enquadrado o trabalho que originou esta dissertação. Nessa altura, foi visto que técnicas baseadas em *softcomputing* se apresentam promissoras em campos que visam a sintonia automática de controladores *PID*. As redes neuronais artificiais têm-se revelado bastante adequadas na construção de mapeamentos; nessa perspectiva, serão utilizadas na presente dissertação como instrumentos para construção de modelos assentes sobre os processos que se pretendem controlar. Pelo facto do problema de sintonia poder estar intimamente ligado à optimização, serão igualmente exploradas as potencialidades dos algoritmos genéticos nessa área.

Na secção 2.2 serão apresentados os fundamentos de redes neuronais e a sua aplicabilidade à modelação de sistemas de controlo. Na secção 2.3 serão apresentados os fundamentos dos algoritmos genéticos, nomeadamente no que respeita à sua capacidade de optimização.

O capítulo termina com as conclusões apresentadas na secção 2.4, onde se referem os tipos de redes neuronais utilizadas e se menciona a ligação da programação genética ao treino de um dos tipos de redes neuronais.

2.2 REDES NEURONAIS

Ao longo dos últimos tempos tem-se esperado dos sistemas de controlo melhor desempenho na sua dinâmica e maior robustez no seu comportamento [58] [59]. Dum

modo geral, pode-se dizer que se tem assistido a uma necessidade crescente de que os sistemas de controlo se apresentem capazes de lidar com processos cada vez mais complexos, definidos em contextos de incerteza e com elevados graus de não linearidade.

O aumento da complexidade dos processos traz, como consequências directas, o aumento do nível de abstracção e o aparecimento de incertezas nos respectivos modelos e nas suas representações matemáticas. Neste contexto, para tratar problemas deste tipo têm sido utilizados nos últimos anos a modelação e controlo inteligentes. Os controladores inteligentes são geralmente auto-adaptativos e apresentam bons desempenhos no tratamento de alterações dos processos e meio envolvente. Deve ser realçada a importância do desenvolvimento de metodologias que permitam aferir a grandeza do erro inerente aos modelos desenvolvidos.

O Controlo inteligente foi inicialmente proposto em 1971 por Fu [60] com o objectivo de aumentar a flexibilidade e estender as possíveis aplicações dos sistemas de controlo automático existentes na altura. A abordagem de então recorria a técnicas descritas em áreas tais como Inteligência Artificial, Investigação Operacional e Controlo Dinâmico. Procuravam-se uma adequada e 'inteligente' conjugação e interacção, de acordo com trabalhos desenvolvidos em 1988 por Saridis e Valavanis [61]. Recentemente, tem-se vindo a observar igualmente o recurso às Ciências da Computação como ferramenta para tratamento de sistemas com uma complexidade crescente.

O estudo de leis físicas que permitem recorrer ao formalismo matemático para estabelecimento de modelos, torna-se cada vez menos aliciante quanto maior for o crescimento da complexidade dos processos. Alguns dos muitos aspectos que justificam tal complexidade podem ser registados nos seguintes pontos [58]:

- factores estocásticos, não lineares e variância no tempo.
- sujeição a perturbações externas próprias do meio envolvente.
- existência de variáveis difíceis de avaliar, ou tais que a sua avaliação se torna inviável em tempo real.

No enquadramento descrito compreende-se que um operador humano atinja frequentemente as suas próprias limitações.

Um dos principais requisitos dum controlador inteligente será a possibilidade deste incorporar a criatividade e os poderes de abstracção e de adaptação próprios dum operador humano, mas minimizando aspectos indesejáveis tais como a inconsistência, a fadiga, a subjectividade e a instabilidade.

A investigação em sistemas inteligentes recorre a conceitos e metodologias provenientes doutras disciplinas, de entre as quais se destacam a Neurofisiologia, a Inteligência Artificial, a Optimização, a Teoria da Aproximação, o Controlo Teórico e a Matemática. Isso contribuiu para o aparecimento duma nova disciplina, habitualmente designada por "Conexionismo" ou Sistemas Inteligentes, na qual os modelos emergentes têm tendência a ser auto-programáveis, estocásticos, difusos, heurísticos e associativos. Tratam-se de modelos definidos em espaços multi-dimensionais, finitos ou infinitos, contínuos ou discretos, genericamente representados pelo mapeamento entrada/saída do tipo $f : x \rightarrow y$.

A aproximação do mapeamento f pode ser feito por via directa, especificando-se uma transformação funcional adequada, ou por via indirecta, fazendo-se uso de exemplares de aprendizagem pertencentes aos espaços X e Y . Sistemas com capacidade de aprendizagem determinam o mapeamento f a partir de dados experimentais, pelo que será da máxima conveniência que a aprendizagem decorra com recurso à totalidade da informação disponível sobre o fenómeno que se pretende aproximar por meio de f . O termo 'inteligência', quando aplicado a sistemas, refere-se à capacidade de 'aprendizagem' e de 'adaptação' por parte destes a novas experiências.

Uma das tecnologias frequentemente presente em sistemas de controlo integra as redes neuronais artificiais, que, uma vez treinadas, deverão ficar com a capacidade de dar resposta a sinais de entrada que não estejam contidos nos dados que serviram para a aprendizagem; esta característica é denominada generalização.

As redes neuronais artificiais são sistemas em que o processamento é distribuído e paralelo, contrastando com o processamento centralizado e sequencial de outros sistemas que compreendem poucas unidades de processamento mas nos quais cada uma tem um funcionamento complexo. As redes neuronais são modelos abstractos (simplificados em

termos do funcionamento e da estrutura) dos sistemas neuronais biológicos naturais, entre os quais se destacam, por exemplo, os tecidos de neurónios do cérebro humano.

De entre outras características das redes neuronais artificiais, podem-se apontar:

- tolerância a dados de entrada com erro;
- generalização local ou comportamento consistente: dados de entrada similares deverão conduzir a respostas semelhantes.

No seguimento, a expressão 'redes neuronais' refere-se às redes neuronais artificiais. Algumas considerações sobre o sistema nervoso que serviu de inspiração biológica para as redes neuronais artificiais serão tratadas na secção 2.2.1.

Os processos que levam à obtenção de redes neuronais integradas num sistema de controlo inteligente deverão, numa 1ª fase, determinar o tipo de rede a utilizar; posteriormente, são tidos em conta 3 elementos fundamentais [58]:

- arquitectura ou topologia: define as ligações entre os vários elementos que compõem a rede neuronal artificial.
- aprendizagem: consiste no processo de estabelecimento dos parâmetros mais convenientes da rede neuronal, implementando os algoritmos adequados que conduzam ao bom funcionamento desta tendo em conta o fim em vista. As capacidades de modelação conseguidas não deverão depender dos algoritmos utilizados, mas sim da prova de que existe um sistema óptimo, o qual deverá ser conseguido com recurso ao algoritmo utilizado. A obtenção de tal sistema num tempo razoável dependerá da arquitectura da rede neuronal. Para se decidir se um dado algoritmo é apropriado para uma determinada aprendizagem, devem-se ter em conta factores tais como a velocidade de convergência para o modelo óptimo ou a possibilidade de ocorrer uma paragem num óptimo que não seja global. Os algoritmos de aprendizagem deverão alterar a resposta da rede neuronal numa forma tão local quanto possível, de preferência. Se tal não acontecer em maior ou menor grau, a aprendizagem para alguns elementos irá 'apagar' a informação previamente aprendida para outros elementos.

- desempenho: serão utilizados vários critérios cujos valores nos dão uma medida da proximidade entre a resposta da rede neuronal e aquela que seria de esperar face aos dados experimentais que levaram à sua construção.

As redes neuronais de generalização local são esquemas de aproximação universal, uma vez que têm a possibilidade de aproximar qualquer função contínua não linear com um grau arbitrário de precisão [58]. Este conceito está na base da classe das redes neuronais de Memória Associativa (*AMN*).

As *AMN* têm a propriedade de que as suas saídas são uma soma ponderada de saídas de funções base definidas sobre uma região restrita do espaço de entrada. Alguns exemplos de redes neuronais pertencentes a esta classe são:

- *RBF* – redes neuronais com funções de base radial.
- *CMAC* – *cerebellar model articulation controller*
- *B-Spline* – *basis spline*

De entre outras topologias utilizadas para redes neuronais com capacidades de aproximação de funções, podem-se referir por exemplo [58], redes neuronais de perceptrões multi-camada (*MLP*), ou redes neuronais de ligações funcionais (*FLN*). Tal como referido anteriormente, todos os tipos de redes neuronais têm surgido directa ou indirectamente da inspiração biológica sugerida pelo sistema nervoso.

De seguida vai ser feito um breve resumo dos fundamentos biológicos das redes neuronais, após o que serão descritos os tipos de redes neuronais utilizados nesta dissertação, nomeadamente: *MLP*, *RBF* e *B-Spline*.

2.2.1 Inspiração biológica das redes neuronais artificiais

Entende-se por sistema nervoso a totalidade do tecido nervoso do corpo [62], que consiste no conjunto de unidades celulares, os neurónios, articulados entre si pelas sinapses, formando complexas cadeias ou sistemas. A célula nervosa possui a característica de ser

essencialmente excitável, sendo especializada na recepção, integração, transformação e retransmissão de informação codificada, o que torna a comunicação a sua função essencial.

2.2.1.1 O neurónio

A teoria do neurónio assenta em argumentos morfológicos, embriológicos e da fisiologia normal e patológica. Esta teoria renovou completamente a Neurologia e continua a ser instrumento imprescindível para a compreensão da forma e do funcionamento do sistema nervoso.

O neurónio é uma célula de dimensão muito variável e de morfologia complexa, constituída pelo corpo celular e todos os seus prolongamentos [63], conforme se ilustra em Fig. 2.1, Fig. 2.2 e Fig. 2.3.

Os tamanhos dos neurónios são muito variados, registando-se alguns exemplos:

- grãos do cerebelo;
- células piramidais do cérebro, ou células de Purkinje do cerebelo;
- células radiculares da medula espinal (dezenas de micra, visíveis à vista desarmada);
- nervos motores da medula (alguns decímetros de comprimento, estendem-se desde a medula até às extremidades dos membros);
- células ganglionares raquidianas (estendem-se desde os gânglios espinais até aos pontos mais distantes do corpo);
- fibras do feixe piramidal, ou cérebro-espinal (estendem-se do córtex cerebral até à medula espinal).

Os neurónios podem ser classificados consoante a sua função, implicando algumas diferenciações morfológicas (Fig. 2.3):

- neurónios sensitivos: permitem que o organismo reaja a alterações operadas no meio ambiente; genericamente, o corpo celular destas células está periféricamente situado, próximo da superfície sensitiva e o axónio estende-se para o sistema nervoso central;

- neurónios de associação: quando fazem parte de cadeias neurónicas ligando entre si grupos de células nervosas;
- neurónios motores: quando fazem parte de uma cadeia, cujos elementos terminais são axónios inervando músculos ou outros elementos efectores Fig. 2.1.

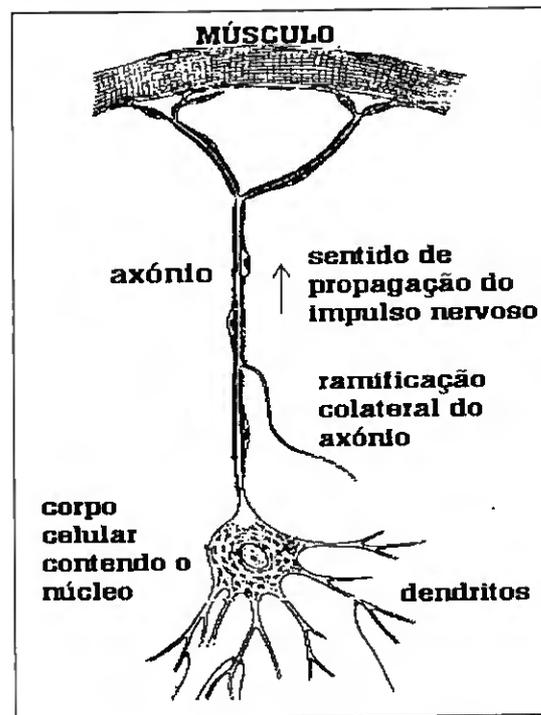


Fig. 2.1 – Um neurónio típico.

No que refere o aspecto morfológico, na célula nervosa destaca-se o corpo celular contendo o núcleo, do qual depende a integridade da célula. Do núcleo sai um ou vários prolongamentos [63], o que leva à classificação em monopolar, bipolar, ou multipolar (Fig. 2.2). Na maioria dos casos distinguem-se duas espécies de prolongamentos nervosos, os axónios que são prolongamentos cilindroaxiais e os dendritos ou prolongamentos protoplásmicos.

2.2.1.2 As sinapses

As ligações entre neurónios denominam-se sinapses (fig. 5.3). O número de sinapses dum neurónio pode ser muito reduzido ou atingir centenas de milhares, com morfologia muito variável. Estudos recentes fazem crer que a transmissão nervosa de neurónio a neurónio se faz por intermédio de substâncias químicas que se produzem nas sinapses (teoria dos intermediários químicos, acetilcolina, ou adrenalina, ou outros).

As sinapses são regiões com propriedades especiais onde o influxo nervoso passa com velocidade muito superior àquela com que percorre as fibras. Além disso, verifica-se que embora em certos casos o influxo nervoso possa transitar nas fibras nervosas em qualquer direcção, nas sinapses a transmissão faz-se sempre na mesma direcção e sentido. A condução do influxo nervoso, quando percorre o axónio, afasta-se do corpo celular, mas dirige-se para o corpo celular quando percorre os dendritos.

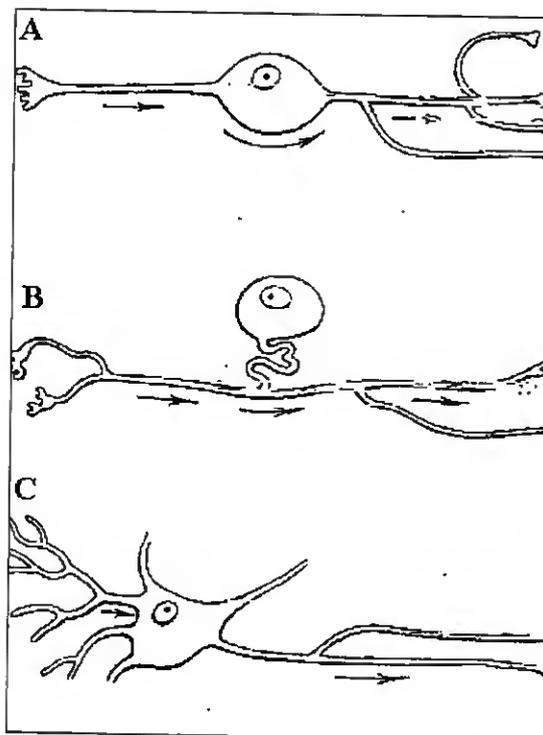


Fig. 2.2 – Alguns tipos de Neurónios: A-bipolar, B-unipolar, C-multipolar.

Embora estudos recentes apontem para que qualquer parte dum neurónio possa participar na formação duma sinapse (Fig. 2.4), de entre os tipos morfológicos mais frequentes de sinapses podem-se apontar:

- sinapses axodendríticas, isto é, entre o axónio de uma célula e dendrito ou dendritos de outras células;
- sinapses axo-somáticas, entre o axónio de uma célula e o corpo celular de outra célula;
- sinapses axoaxónicas, entre axónios.

Do ponto de vista fisiológico podem-se distinguir os seguintes tipo de sinapses:

- Sinapse eléctrica: encontrada em certos invertebrados e vertebrados inferiores, inexistente no sistema nervoso de mamíferos.
- Sinapse química: própria dos mamíferos onde se verifica transmissão química. As membranas celulares estão neste caso separadas pelas fendas sinápticas e a transmissão ocorre numa direcção apenas, isto é, a sinapse é polarizada e está limitada pelas membranas de uma célula pré-sináptica e outra pós-sináptica. À medida que o impulso nervoso se aproxima da membrana pré-sináptica, provoca a libertação de transmissor neuroquímico que rapidamente se difunde através da fenda sináptica e se aproxima da membrana pós-sináptica, onde provoca uma alteração na sua permeabilidade iónica.

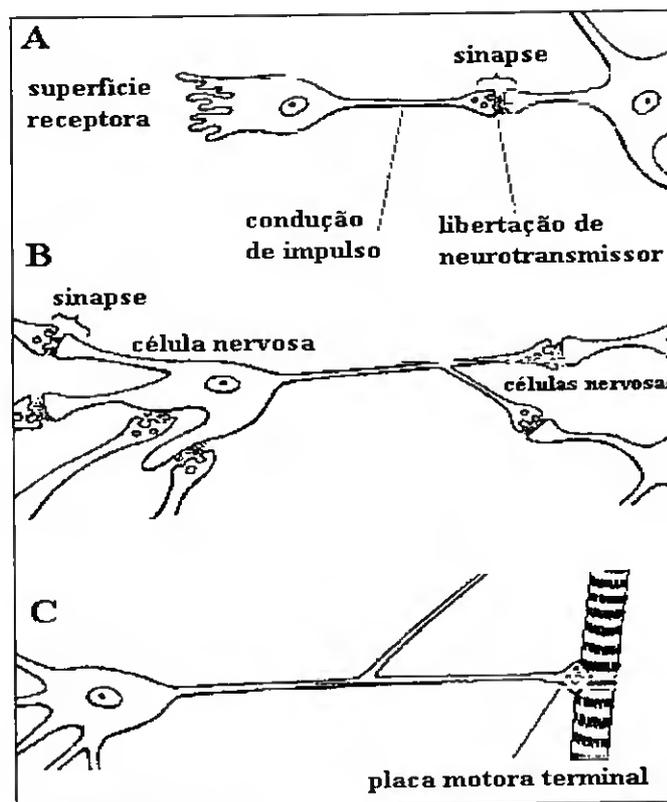


Fig. 2.3 – Alguns exemplos de sinapses: A - neurónio sensitivo, B - neurónio de associação, C - neurónio motor.

Muitos autores têm contribuído para a divulgação da teoria do neurónio; entre nós, destaca-se o Prof. Miguel Augusto Bombarda, Médico e Professor,¹ que a vulgarizou e

¹Miguel Augusto Bombarda, (1851-1910), Médico e Professor português, nasceu no Rio de Janeiro. Fez os seus estudos na Escola Médico-Cirúrgica, onde foi professor desde 1880.

defendeu com grande entusiasmo, pretendendo ver nessa teoria os fundamentos para uma nova Psicologia.

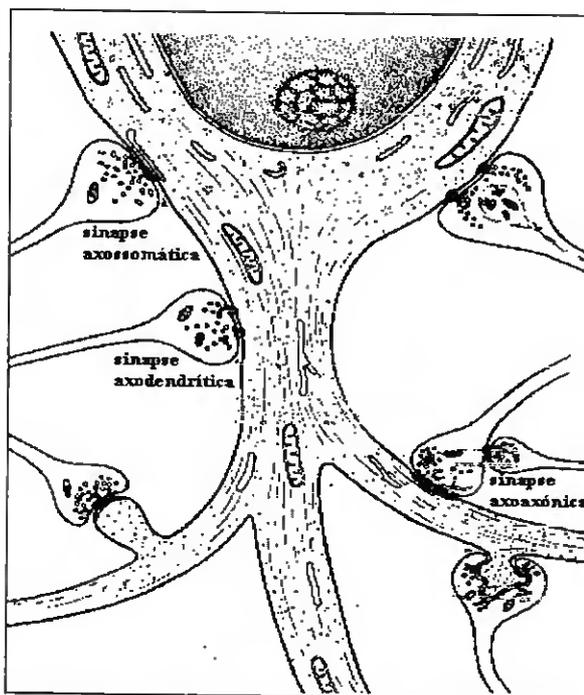


Fig. 2.4 – Alguns pontos do neurônio onde é possível encontrar sinapses.

2.2.2 Redes neuronais do tipo *MLP*

Uma das características que tem sido apontada às redes neuronais do tipo *MLP* é o facto da convergência dos seus algoritmos de aprendizagem ser lenta. Na verdade, tal facto prende-se essencialmente com o tipo de algoritmo de aprendizagem utilizado e não tanto com a topologia que está associada à própria rede, embora esta (topologia) acabe por influenciar a escolha do algoritmo de aprendizagem.

Antes de se passar à apresentação das *MLP*, vão ser feitas algumas considerações [58] sobre um elemento denominado Combinador Linear Adaptativo (*ALC*), elemento constituinte de dois tipos de redes neuronais precursoras, respectivamente: ADALINE e Perceptrão de Rosenblatt. Como ficará explícito na sequência, o *ALC* é o elemento base das *MLP*.

Em 1957 Rosenblatt [28] desenvolveu o primeiro neurocomputador – o Perceptrão; e propôs uma regra de aprendizagem para esta primeira rede neuronal. O teorema da

convergência do Perceptrão e outros resultados obtidos então foram publicados no livro "*Principles of Neurodynamics*" [64]. Por volta da mesma data, Bernard Widrow estabeleceu um processo de aprendizagem segundo o qual seria minimizado o *MSE* entre as saídas dum tipo diferente de rede neuronal ADALINE [65], e o vector de saídas objectivo, para um conjunto de padrões de aprendizagem [66]. As regras de aprendizagem de Widrow-Hoff foram aplicadas a vários problemas, entre os quais o problema de controlo do pêndulo invertido [67].

A estrutura da *ALC* apresenta-se na Fig. 2.5.

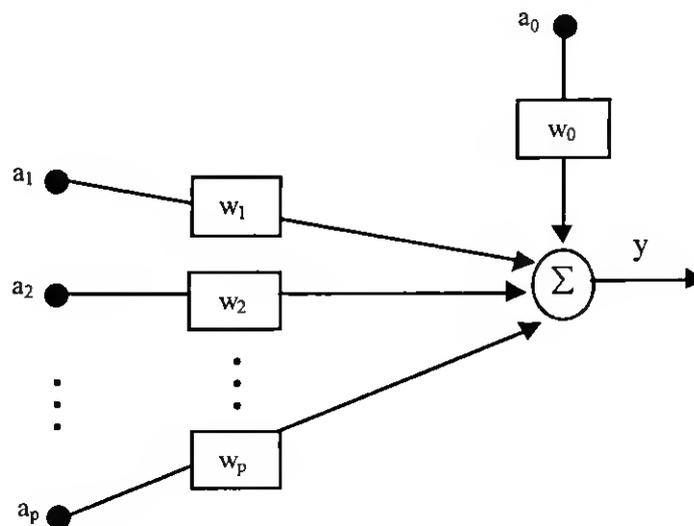


Fig. 2.5 – Combinador linear adaptativo (*ALC*).

De acordo com o *ALC*, a saída y é dada pela combinação linear das entradas:

$$y = \sum_{i=1}^p a_i w_i + a_0 w_0 \quad (2.1)$$

Por vezes é considerada uma variante de acordo com a qual:

$$y = \begin{cases} 1, & \mathbf{a}^T \cdot \mathbf{w} > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (2.2)$$

Pode-se ainda considerar uma camada adicional que corresponde a um pré-processamento dos dados originais de entrada $X (X \rightarrow A)$.

O vector de pesos w é determinado à custa duma regra iterativa de correcção do erro e o processo de aprendizagem termina quando a rede neuronal apresenta um comportamento aceitável.

As redes neuronais do tipo *MLP*, serão provavelmente as mais conhecidas e amplamente utilizadas nas mais diversas aplicações, incluindo sistemas de controlo. Este tipo de redes constitui uma extensão natural do perceptrão de Rosenblatt [64].

A topologia geral duma rede neuronal do tipo *MLP* apresenta-se na Fig. 2.6.

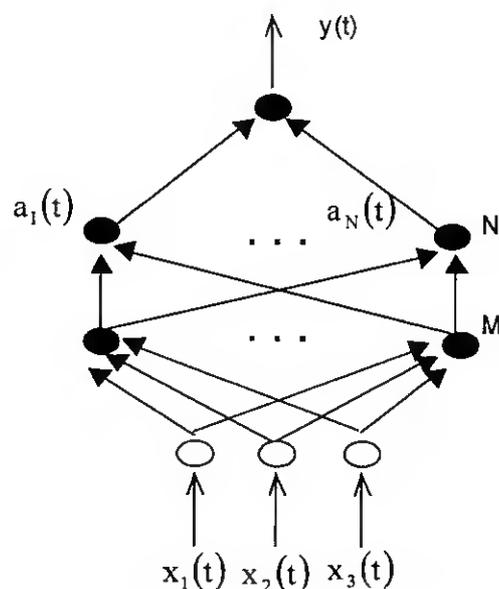


Fig. 2.6 – Rede neuronal por camadas - *MLP*.

Como o próprio nome sugere, neste tipo de redes neuronais os nós dispõem-se por camadas, sendo a 1ª camada denominada entrada e a última denominada saída. O número de entradas e de saídas é arbitrário, dependendo da aplicação em curso. No exemplo da Fig. 2.6 a rede neuronal do tipo *MLP* que se apresenta tem 3 entradas uma saída e 2 camadas intermédias (ou escondidas) com um número de nós arbitrário, designado respectivamente por M e N .

As *MLP* são redes neuronais directas onde o sinal de entrada $X(t)$ é propagado para a frente através de várias camadas de processamento, com vista ao cálculo da saída $y(t)$. O número de nós de cada camada é um parâmetro da rede neuronal, sendo cada nó um *ALC* com a função de transferência apropriada f . Neste tipo de rede neuronal, a saída de cada nó serve de entrada para todos os nós da camada seguinte, da mesma forma que cada nó recebe como entradas todas as saídas dos nós da camada anterior. Nalguns casos menos frequentes são permitidas ligações entre nós de camadas não adjacentes, bem como nós com realimentação [68] [69].

Numericamente, a saída de cada nó i é o valor de f calculado num ponto que resulta da combinação linear das saídas dos nós da camada anterior, somada com uma constante de polarização. Assim, conclui-se que, tal como acontece com a *ALC*, a cada ramo está associado um peso. A polarização de cada nó pode ser vista como o peso dum ramo que liga uma entrada fictícia x_0 , constante e igual a 1, ao respectivo nó.

A função de transferência f de cada nó é limitada, não linear e continuamente diferenciável, sendo geralmente utilizada a sigmoide (2.3).

$$f(u) = \frac{1}{1 + e^{-u}} \quad (2.3)$$

A variável independente u da expressão (2.3) é dada de forma semelhante à equação (2.1); assim, para uma camada j com N nós, designando por M o número de nós na camada anterior, para cada nó da camada j verifica-se (2.4).

$$u_j \equiv s_j(w_{ij}) = \sum_{i=0}^M w_{ij} y_i, \quad j = 1 \dots n \quad (2.4)$$

Sendo a função f contínua, significa que [58] a saída de cada nó está continuamente dependente das entradas da rede neuronal.

O primeiro algoritmo de aprendizagem utilizado para este tipo de redes neuronais é um algoritmo de gradiente descendente, tendo sido deduzido em trabalhos [70], [71], [72] e [73], desenvolvidos independentemente.

Este tipo de redes funciona adequadamente com dados de treino redundantes, como é frequente aparecer em problemas de modelação e controlo. Quando tal acontece é definido um hiper-plano paralelo às entradas da rede redundantes, os respectivos pesos tenderão para zero e não será necessário introduzir nós adicionais nas camadas intermédias.

Presentemente, o treino de redes neuronais do tipo *MLP* recorre ainda com grande frequência ao algoritmo da retropropagação do erro².

2.2.2.1 Algoritmo de aprendizagem *backpropagation*

O algoritmo de aprendizagem *backpropagation* [72] é do tipo gradiente descendente. A sua formalização ocorreu apenas nos anos 80 devido às limitações computacionais verificadas até essa data [58].

Este algoritmo de aprendizagem supervisionada pressupõe a apresentação à entrada da rede em cada instante t , do conjunto de padrões de treino X e indicação das respectivas saídas desejada Y .

A medida de desempenho do sistema (rede neuronal) será uma função de custo E definida no espaço multidimensional dos pesos.

Em geral, a função de custo E é dada por (2.5).

$$E(t) = \frac{1}{2} \sum_{k=1}^L e_k^2(t) \quad (2.5)$$

sendo:

L – número de saídas, na presente aplicação $L=1$;

$e_k(t)$ – erro no instante t , definido de acordo com (2.6)

$$e_k(t) = y_k(t) - y_k^*(t) \quad (2.6)$$

Na expressão (2.6) $y_k^*(t)$ representa o valor objectivo para $y_k(t)$.

² *error back propagation.*

O princípio base do algoritmo *backpropagation* visa a minimização da função de custo em ordem aos pesos W da rede. O procedimento de minimização utilizado, conhecido por gradiente descendente, consiste na progressão iterativa no espaço dos pesos em passos com direcção e sentido do decrescimento mais acentuado da função de custo E .

Neste contexto, a cada ponto no espaço dos pesos deverá ser aplicada uma correcção proporcional ao simétrico da respectiva componente do gradiente da função de custo, calculada no respectivo ponto. A correcção ao peso da ligação entre dois nós i e j será, então, dada por (2.7).

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}} \quad (2.7)$$

A constante de proporcionalidade positiva α , denomina-se factor de correcção, deve ser ajustada pelo utilizador de acordo com o problema em estudo.

Tendo em conta a regra da derivação de funções compostas e atendendo a (2.4), verifica-se (2.8).

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \quad (2.8)$$

A partir de (2.4) se conclui (2.9)

$$\frac{\partial s_j}{\partial w_{ij}} = y_i \quad (2.9)$$

O 1º factor do 2º membro de (2.8) designa-se por δ_j (2.10).

$$\delta_j = \frac{\partial E}{\partial s_j} \quad (2.10)$$

Considerando (2.7), pode-se concluir que as correcções aos pesos dirigidos da camada i para a camada j são dadas por (2.11).

$$\Delta w_{ij} = -\alpha y_i \delta_j \quad (2.11)$$

Para se obter a expressão final da correcção dos pesos resta determinar um processo para avaliação de $\frac{\partial E}{\partial s_j}$.

Lembrando que E é uma função composta, e seguindo a respectiva regra de derivação, verifica-se que (2.12).

$$\delta_j = \frac{\partial E}{\partial s_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \quad (2.12)$$

Atendendo às expressões (2.3) e (2.4), pode-se utilizar a notação (2.13).

$$\frac{\partial y_j}{\partial s_j} = f'_j(s_j) \quad (2.13)$$

Quanto à determinação de $\frac{\partial E}{\partial y_j}$, distinguem-se as situações em que o nó j pertence à camada de saída, ou a uma camada intermédia.

Na situação em que j pertence à camada de saída, de acordo com (2.5) e (2.6), usando, mais uma vez, a regra de derivação da função composta, verifica-se (2.14):

$$\frac{\partial E}{\partial y_j} = \frac{\partial E}{\partial e_j} \frac{\partial e_j}{\partial y} = e_j \quad (2.14)$$

Assim, a partir de (2.12), (2.13) e (2.14), lembrando (2.11), conclui-se que as correcções aos pesos da última camada são dados por (2.15).

$$\Delta w_{ij} = -\alpha y_i e_j f'_j(s_j) \quad (2.15)$$

Nas situações em que a função de transferência da camada de saída é linear, repare-se que, na expressão anterior $f'_j(s_j) = 1$.

Na situação em que o nó j é intermédio verifica-se (2.16)

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial s_k} \frac{\partial s_k}{\partial y_j} \quad (2.16)$$

Novamente utilizando (2.4), verifica-se (2.17)

$$\frac{\partial s_k}{\partial y_j} = w_{jk} \quad (2.17)$$

Entrando com (2.10) e (2.17) em (2.16), verifica-se (2.18).

$$\frac{\partial E}{\partial y_j} = \sum_k w_{jk} \delta_k \quad (2.18)$$

Substituindo (2.18) e (2.13) em (2.12), resulta (2.19).

$$\delta_j = f'_j(s_j) \sum_k w_{jk} \delta_k \quad (2.19)$$

Por último, as correcções dos pesos dadas por (2.11) assumem a forma (2.20)

$$\Delta w_{ij} = -\alpha y_i f'_j(s_j) \sum_k w_{jk} \delta_k \quad (2.20)$$

Note-se que (2.19) é uma expressão recursiva, ou seja, permite calcular as correcções aos pesos de qualquer camada intermédia a partir das correcções da última camada que serão determinadas a partir da expressão (2.15).

As duas expressões (2.15) e (2.20) traduzem a propagação para trás do erro e_j através da rede linearizada no ponto de funcionamento da rede original e transposta da rede original; sendo as saídas trocadas com as entradas. A rede assim obtida tem os mesmos nós da rede original e cada nó propaga a combinação linear das suas entradas, obtendo-se os factores de ponderação através da derivada da função de transferência calculada no valor da sua soma dada por (2.19).

O algoritmo atrás introduzido designa-se por aprendizagem estocástica, padrão-a-padrão, ou adaptativa. Normalmente a expressão (2.5) é estendida para um conjunto de padrões obtendo-se assim a versão *batch* (por lotes) deste algoritmo

2.2.2.2 Algoritmos de treino de 2ª ordem

O algoritmo de treino *backpropagation* apresenta uma convergência pobre, pois recorre a aproximações lineares para se atingir o óptimo na superfície $E(w)$ [74]. Se as direcções de procura na superfície de erro forem determinadas com base em modelos quadráticos, então será de esperar melhoria em termos da convergência para o óptimo.

Considerando como função de custo o *MSE*, então a superfície de erro para N padrões de treino será dada por (2.21).

$$E(w) = \frac{1}{N} \sum_{p=1}^N e_p^2(w) \quad (2.21)$$

Para um vector de pesos w de dimensão N , o vector gradiente e a matriz Hessiana num ponto w_k , definem-se respectivamente pelas expressões (2.22) e (2.23).

$$g_k = \nabla E(w_k) = \left[\frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \dots \quad \frac{\partial E}{\partial w_{N_k}} \right]^T \Bigg|_{w=w_k} \quad (2.22)$$

$$H_k = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_{N_k}} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \ddots & & \\ \vdots & & & \\ \frac{\partial^2 E}{\partial w_{N_k} \partial w_1} & \frac{\partial^2 E}{\partial w_{N_k} \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_{N_k}^2} \end{bmatrix} \Bigg|_{w=w_k} \quad (2.23)$$

Os três primeiros termos do desenvolvimento em série de Taylor em torno do ponto genérico w_k apresentam-se na expressão (2.24).

$$E(w) \approx E(w_k) + (w - w_k)g_k + \frac{(w - w_k)^2}{2} H_k (w - w_k) \quad (2.24)$$

Sendo H_k definida positiva, esta aproximação tem um mínimo global para $\frac{\partial E}{\partial w} = 0$, assim

$$\frac{\partial E}{\partial w} = g_k + H_k (w - w_k) = 0 \quad (2.25)$$

A partir da expressão (2.25) é possível determinar os pesos para a iteração seguinte ($k+1$) a partir dos pesos da iteração presente (k):

$$w_{k+1} = w_k - H_k^{-1} g_k \quad (2.26)$$

A expressão (2.26) implementa o método de procura de Newton; em várias situações surgem problemas no cálculo de inversa de H_k , nessas circunstâncias são utilizadas aproximações de H_k ou da sua inversa de tal forma que se mantenha definida positiva.

O algoritmo de treino de Levenberg-Marquardt é um algoritmo matricial de 2ª ordem [76] [77]. Designando o Jacobiano por $J(w) = \left[\frac{\partial e_p}{\partial w_j} \right]$ e $G(w) = \sum_{p=1}^N e_p(w) \nabla^2 e_p(w)$, então o gradiente e a matriz Hessiana da função de custo são dadas respectivamente por (2.27) e (2.28):

$$\nabla E(w) = \frac{2}{N} J^T(w) e(w) = \frac{2}{N} \sum_{p=1}^N \frac{\partial e_p}{\partial w} e_p(w) \quad (2.27)$$

$$H = \nabla^2 E(w) = \frac{2}{N} [J^T(w) J(w) + G(w)] \quad (2.28)$$

Atendendo à forma como se define $G(w)$, nas vizinhanças do mínimo esta matriz pode ser desprezada resultando a expressão (2.28) em:

$$H \approx \frac{2}{N} J^T(w) J(w) \quad (2.29)$$

Podem ocorrer situações em que a matriz Hessiana assim definida se torne singular ou mal condicionada; nessas circunstâncias, Levenberg e Marquardt consideraram a introdução do parâmetro de Marquardt μ , passando a matriz Hessiana a ser calculada de acordo com (2.30).

$$H \approx \frac{2}{N} [J^T(w)J(w) + \mu I] \quad (2.30)$$

A partir das expressões de actualização dos pesos (2.25) e (2.26) obtém-se o algoritmo de Levenberg Marquardt expresso de acordo com (2.31).

$$w_{k+1} = w_k - \frac{2}{N} [J^T(w_k)J(w_k) + \mu I]^{-1} \nabla E(w_k) \quad (2.31)$$

A expressão (2.31) mostra que o parâmetro de Marquardt interpola a direcção de pesquisa do óptimo, entre o gradiente descendente e a direcção de Newton. Para valores elevados de μ , o algoritmo é predominantemente gradiente descendente com factor de correcção de pesos $\eta = \frac{1}{\mu}$, caso contrário ($\mu \rightarrow 0$), é a direcção dada pelo método de Gauss-Newton que prevalece na evolução do algoritmo.

2.2.3 Redes neuronais do tipo *RBF*

As redes neuronais com funções de Base Radial – *RBF* foram originalmente utilizadas em problemas de interpolação em espaços de elevada dimensão. As boas propriedades numéricas reveladas, foram extensivamente investigadas por Buhmann e Powell [78], [79] e [80].

A topologia das redes neuronais com funções de base radial – *RBF* resulta da que se apresentou na Fig. 2.6, para as *MLP*, considerando-se apenas uma camada intermédia [81].

Na Fig. 2.7 apresenta-se uma topologia genérica duma rede neuronal do tipo *RBF* com n entradas e p nós intermédios.

Contrariamente ao que se passa com as *MLP*, na presente situação só a camada de saída é do tipo *ALC*; assim, a saída $y(t)$ será:

$$y(t) = \sum_{i=1}^p a_i(t) w_i \tag{2.32}$$

A saída a_i de cada nó da camada intermédia depende da estrutura do nó, definida pela respectiva função não linear f_i . De entre as várias definições possíveis para f_i , uma alternativa será considerá-la Gaussiana localizada (2.33).

$$f_i(d) = e^{-\frac{d^2}{2\sigma_i^2}} \tag{2.33}$$

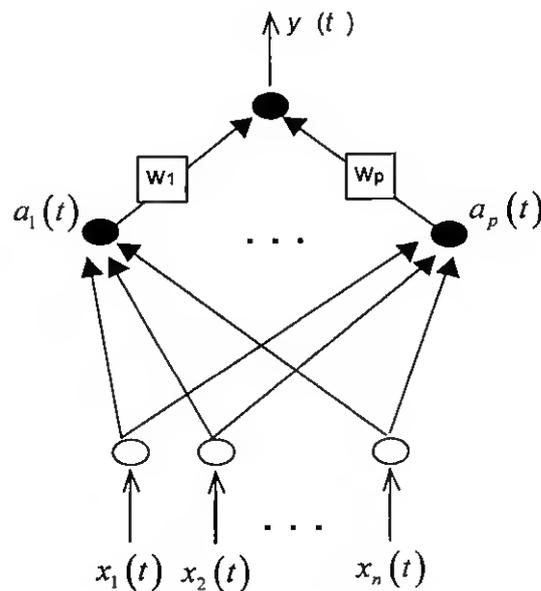


Fig. 2.7 – Rede neuronal com funções de base radial – *RBF*.

Cada função f_i tem 2 parâmetros:

- σ_i parâmetro de dispersão (ver (2.33)).
- c_i centro da Gaussiana.

A variável independente d , representa a distância do vector de entrada \mathbf{x} , ao centro da respectiva Gaussiana \mathbf{c}_i . Para n entradas, \mathbf{c}_i e \mathbf{x} serão vectores de dimensão n ,

verificando-se: $d = \|\mathbf{c}_i - \mathbf{x}\| = \sqrt{\sum_{j=1}^n (c_{ij} - x_j)^2}$; atendendo a (2.33), tem-se que:

$$f_i(\|\mathbf{c}_i - \mathbf{x}\|) = e^{-\frac{\sum_{j=1}^n (c_{ij} - x_j)^2}{2\sigma_i^2}} \quad (2.34)$$

ou ainda:

$$f_i(\|\mathbf{c}_i - \mathbf{x}\|) = \prod_{j=1}^n e^{-\frac{(c_{ij} - x_j)^2}{2\sigma_i^2}} \quad (2.35)$$

A saída da *RBF* será dada a partir de (2.32) por:

$$y = \sum_{i=1}^p w_i \prod_{j=1}^n e^{-\frac{(c_{ij} - x_j)^2}{2\sigma_i^2}} \quad (2.36)$$

As *RBF* são redes neuronais de generalização local [74]. De acordo com a expressão (2.33), as suas funções de base só apresentam valor significativamente não nulo num intervalo finito nas vizinhanças do centro ($\lim_{d \rightarrow +\infty} f_i(d) = 0$). A largura de cada função de base é regulada pelo dimensionamento do respectivo parâmetro de dispersão σ . Verifica-se então que, quanto menor for esse parâmetro, mais localizada será a resposta associada ao nó correspondente e menos elementos do espaço de entrada X irão contribuir para o cálculo da saída da rede, por via do respectivo nó. Esta propriedade é bastante importante quando de faz adaptação de redes neuronais em linha, dado que, nessas circunstâncias, se pretende que nova informação seja apreendida sem que haja deterioração da aprendizagem proveniente de outros elementos do espaço de entrada. O perfil sigmoideal (2.3) das funções de transferência utilizadas nas *MLP*, faz com que seja de esperar pior desempenho destas redes quando comparadas com as *RBF*.

O número p de nós da camada intermédia é bastante menor do que o cardinal do conjunto de treino. No entanto, consoante o posicionamento dos centros e a dimensão do conjunto de treino, pode-se tender para a obtenção de modelos extremamente complexos, sendo esta

propriedade conhecida [58], como maldição dimensional³. Padrões de entrada redundantes ou irrelevantes provocam enorme aumento do número de nós na camada intermédia, sem o correspondente aumento da flexibilidade do modelo resultante.

Em 1991, Hartman e Keeler propuseram uma variante às *RBF*, segundo a qual as entradas irrelevantes seriam rejeitadas [75]. De acordo com estes autores, em vez de ser feito o produto de Gaussianas univariáveis, como se apresenta na expressão (2.36), é utilizada a soma algébrica:

$$y = \sum_{i=1}^p w_i \sum_{j=1}^n e^{-\frac{(c_{ij}-x_j)^2}{2\sigma_i^2}} \quad (2.37)$$

O treino deste tipo de redes é feito em duas etapas que se alternam:

1. Estabelecimento da estrutura da camada intermédia: Nesta etapa é determinado o número e localização dos centros C_i , bem como as dimensões dos raios σ_i .
2. Determinação dos pesos da camada de saída: Sendo a camada de saída do tipo *ALC*, os pesos poderão ser determinados de tal forma que seja minimizado o quadrado do erro à saída da rede. Considerando a propagação dum matriz X de padrões de entrada (ver Fig. 2.7), obtém-se da correspondente matriz A das saídas da camada escondida. A matriz pseudo-inversa de A designa-se por A^+ e define-se por: $A^+ = (A^T A)^{-1} A^T$. Conhecendo-se o vector y das saídas desejadas da rede e atendendo-se à expressão (2.32), conclui-se que o vector de pesos é determinado de acordo com (2.38).

$$w = A^+ y \quad (2.38)$$

É também comum utilizar-se para determinação dos pesos lineares um factor de regularização λ . Neste caso os pesos são determinados por minimização de J dado por (2.39).

$$J = \|y - A w\|^2 + \lambda \|w\|^2 \quad (2.39)$$

³ *curse of dimensionality*

A solução dessa minimização é dada por (2.40).

$$w = (A^T A + \lambda I)^{-1} A^T y \quad (2.40)$$

No que respeita à primeira etapa descrita têm sido propostos vários algoritmos tanto para a determinação dos centros, como para a determinação dos raios.

O algoritmo para determinação de centros, *K-means Clustering* foi proposto por Duda e Hart [82]. De acordo com estes autores, seriam utilizados k centros; a disposição desses centros seria tal que a soma dos quadrados das distâncias de todos os padrões de entrada ao centro mais próximo fosse mínima.

O algoritmo *K-means Clustering* pode apresentar-se na sua versão recursiva de acordo com os seguintes pontos:

- inicializar aleatoriamente um número k de centros
- ler vector de entrada $x(t)$
- designando por $c_k(t)$ o centro mais próximo de $x(t)$, alterar $c_k(t)$ de forma a obter-se: $c_k(t+1)$: $c_k(t+1) = c_k(t) + \alpha(t)[x(t) - c_k(t)]$. O factor de aprendizagem $\alpha(t)$, ($0 < \alpha(t) < 1$) deverá diminuir ao longo das iterações t .
- parar o processo de aprendizagem quando $\alpha(t) = 0$ ou ao fim dum número fixo de iterações.

A *toolbox* de redes neuronais do MATLAB [83] implementa um algoritmo iterativo, baseado no método de *Orthogonal Least Squares* [84], no qual considera os centros como elementos extraídos do conjunto de treino. O processo inicia-se com um centro, em cada iteração o padrão que conduz ao maior valor de erro é tomado como novo centro na iteração seguinte. Este algoritmo é implementado na referida *toolbox* pela rotina *solverb*.

Nas sub-secções que se seguem apresentam-se algumas heurísticas para determinação dos raios das Gaussianas.

2.2.3.1 Diferentes heurísticas para a determinação dos raios

As heurísticas que se seguem [85] dividem-se em duas categorias (Tab. 2.1), consoante os raios associados aos vários centros sejam iguais, ou diferentes.

Raios diferentes para diferentes centros	Raios constantes para todos os centros
1. Desvio padrão empírico	4. Metade da distância máxima entre padrões
2. k vizinhos mais próximos	5. Distância máxima entre centros
3. Vizinho mais próximo	6. Distância vizinha média

Tab. 2.1 – Heurísticas para determinação dos raios.

1- Desvio padrão empírico:

Sendo n o numero de padrões que pertencem ao clustering cujo centro é c e p_i cada um desses padrões, o raio R é dado por (2.41):

$$R = \sqrt{\frac{\sum_{i=1}^n \|c - p_i\|^2}{n}} \quad (2.41)$$

2- k vizinhos mais próximos:

Considerando os k centros c_i mais próximos do centro c , o raio associado a este centro é dado por (2.42):

$$R = \frac{\sum_{i=1}^k \|c - c_i\|}{k\sqrt{2}} \quad (2.42)$$

O valor de k é dado em termos percentuais relativamente à totalidade de centros existentes em cada situação.

3- Vizinho mais próximo:

Seja c_i o centro mais próximo do centro c , então, o raio associado ao centro c é dado por (2.43):

$$R = Q \frac{\|c - c_i\|}{\sqrt{2}} \quad (2.43)$$

Q é uma constante ajustável de acordo com a aplicação; trabalhos apresentados em [85], apontam para $Q=1.25$.

4 - Metade da distância máxima entre padrões:

Sendo p o número de padrões e x_i e x_j dois padrões genéricos, o valor único do raio para todos os centros será dado por (2.44):

$$R = \frac{\max_{i,j=1\dots p} \|x_i - x_j\|}{2\sqrt{2}} \quad (2.44)$$

5 - Distância máxima entre centros

Sendo m o número de centros e c_i e c_j dois centros genéricos, o valor único do raio para todos os centros será dado por (2.45):

$$R = \frac{\max_{i,j=1\dots m} \|c_i - c_j\|}{\sqrt{2m}} \quad (2.45)$$

6 - Distância vizinha média

Sendo m o número de centros, c_i um centro genérico e c_j o seu vizinho mais próximo, o valor único do raio para todos os centros será dado por (2.46):

$$R = \frac{\sum_{i=1}^m \|c_i - c_j\|}{m\sqrt{2}} \quad (2.46)$$

2.2.4 Redes neuronais do tipo *B-Spline*

As redes neuronais do tipo *B-Spline* e do tipo CMAC fazem parte do grupo das redes de memória associativa definidas sobre grelhas⁴. A sua topologia é composta por 3 camadas conforme se apresenta na Fig. 2.8.

A camada de entrada corresponde aos padrões $X(t)$ que deverão ser normalizados num retículo ou grelha de dimensão adequada. As funções de base que compõem a camada intermédia serão definidas sobre essa grelha. A resposta obtida pela camada intermédia caracteriza-se por ser muito localizada, ou seja, as funções de base utilizadas têm a propriedade de só serem não nulas em pequenos intervalos do espaço de entrada; por esta razão, pode-se dizer que estas redes neuronais guardam informação localmente. A camada de saída é um elemento do tipo *ALC*, como já acontecia com as *RBF*; desta forma a saída $y(t)$ será determinada por (2.32).

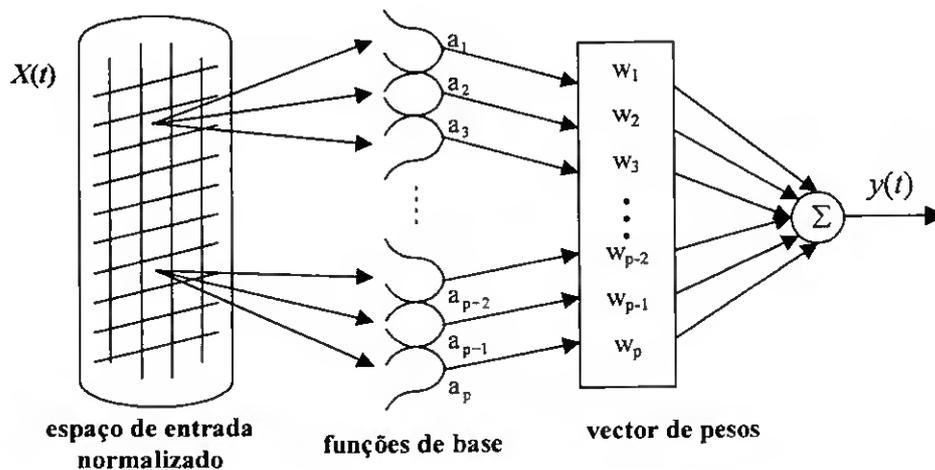


Fig. 2.8 – Rede neuronal do tipo *B-Spline*.

Ambos os tipos de redes neuronais [58], *B-Spline* e CMAC poderão apresentar problemas de maldição dimensional; tal situação poderá ser transposta recorrendo a técnicas de indexação de memória, ou à divisão da rede em sub-modelos.

⁴ *Lattice-based Associative Memory Networks (AMN)*.

As funções do tipo *B-Spline* foram utilizadas com sucesso desde a década de 70 nos algoritmos de interpolação de superfícies [86] [87]. Este tipo de funções é construído à custa de segmentos de polinómios formados por combinações lineares de funções de base; as funções de base multivariáveis são constituídas sobre uma grelha de dimensão adequada, tal como se ilustra na Fig. 2.8.

Como referido anteriormente este tipo de funções pertence à classe das *AMN*. A não inclusão das *B-Splines* na classe dos algoritmos de interpolação de superfícies deve-se à forma como o vector de pesos w é determinado. Apesar desta classificação, ambas as classes, (*AMN* e interpolação de superfícies) visam o mesmo objectivo, ou seja, construir mapeamentos não lineares a partir dos dados entrada/saída observados para o respectivo fenómeno.

O estudo deste tipo de redes neuronais permite fazer uma ligação directa entre elas e os sistemas difusos; as *B-Splines* podem ser interpretadas como um conjunto de regras difusas, conduzindo a modelos e resultados que se obteriam à custa de redes difusas.

Os algoritmos de inicialização das rede neuronais do tipo *B-Spline* partem do princípio de que a obtenção dum modelo mais flexível continua a reproduzir na totalidade o desempenho do modelo anterior [88], [89].

Como o próprio nome indica, as funções de base utilizadas para este tipo de rede são splines de ordem parametrizável. Uma função de base univariável de ordem k é composta por segmentos polinomiais da mesma ordem.

Cada entrada numa rede deste tipo está associada a um eixo coordenado, assim as n entradas definem-se num espaço de dimensão n .

A primeira camada das *B-Splines* define-se sobre uma grelha num sub-conjunto do espaço de dimensão n , podendo-se salientar as seguintes vantagens:

- fácil construção - bastará fazer uma divisão criteriosa de cada eixo para se obter a grelha de entrada.

- estabelecimento da estrutura baseado no conhecimento prévio do fenómeno: a estrutura da grelha poderá reflectir algum conhecimento prévio que possa existir sobre a relação entre os dados observados e alguma das entradas da rede.
- facilidade computacional - para cada entrada, bastará um simples algoritmo de procura para detectar qual a célula que está activada.

Como principais desvantagens podem-se registar:

- maldição dimensional - a necessidade de memória é exponencialmente dependente da dimensão do espaço de entrada.
- dificuldade de generalização - a estrutura definida para uma região do espaço de entrada poderá não ser apropriada para outra região, podendo o modelo encontrado apresentar algumas dificuldade de generalização.

A grelha da camada de entrada para um espaço de dimensão n pode ser entendida como uma extrapolação feita a partir da representação geométrica, que se obtém para as situações particulares de dimensão 2 (plano), ou 3 (espaço). Assim, independentemente da dimensão com que se esteja a trabalhar, são definidos vectores de nós, estando cada dimensão associada a um vector. Para cada eixo do espaço de entrada, a partir do conjunto de nós em causa pode-se definir hiperplanos de dimensão $n-1$, paralelos aos restantes $n-1$ eixos. Tomando em consideração a totalidade dos nós atribuídos a todos os eixos, os respectivos hiperplanos definem uma grelha sobre o espaço de entrada.

A totalidade de nós e respectivas posições para cada eixo varia e depende do algoritmo utilizado. Um conhecimento prévio sobre o problema que se pretende modelar pode servir de auxílio à definição dos nós.

Considerando uma rede do tipo *B-Spline* com n entradas tal que os padrões de entrada para a entrada i se situam-se no intervalo $[x_i^{\min} \ x_i^{\max}]$, então o espaço de entrada da rede pertence ao domínio: $[x_1^{\min} \ x_1^{\max}] \times \dots \times [x_n^{\min} \ x_n^{\max}]$. Tomando como exemplo a dimensão 1, uma possível distribuição de nós será aquela que se apresenta na Fig. 2.9.

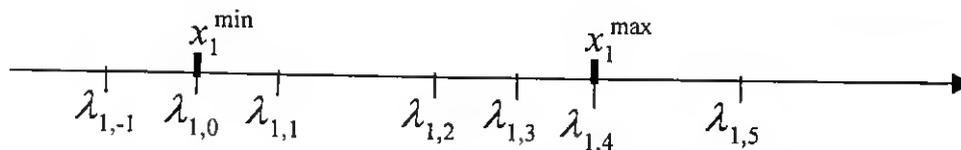


Fig. 2.9 – Distribuição de nós para a entrada x_1 .

Diz-se que um dado nó $\lambda_{1,i}$ é interior, se $\lambda_{1,i} \in]x_1^{\min} \ x_1^{\max}[$, caso contrário o nó é exterior.

Podem existir nós coincidentes que serão úteis para tratar problemas de descontinuidade no modelo. Os nós exteriores servem para gerar funções de base que estejam próximas dos limites do intervalo.

Tal como se sugere na Fig. 2.8, a saída da camada intermédia corresponde à resposta das p funções de base que a compõem definidas na grelha de dimensão n .

Define-se suporte duma função de base como sendo o domínio do espaço de entrada para o qual a sua saída é não nula. Para uma determinada rede, dado um padrão de entrada, o número de funções de base não nulas é constante e designa-se por parâmetro de generalização ρ . Assim, existem ρ funções de base que contribuem para a determinação da saída. Essas ρ funções podem ser organizadas em ρ conjuntos tais que em cada conjunto apenas uma função de base é não nula. A união dos suportes de cada conjunto forma uma cobertura⁵ completa e não sobreposta de dimensão n ; a grelha é então composta por ρ coberturas. Pode-se estabelecer que a saída $y(t)$ da rede dada por (2.32), pode alternativamente representar-se em função de ρ :

$$y(t) = \sum_{i=1}^{\rho} a_{ad(i)} w_{ad(i)}(t-1) \quad (2.47)$$

De acordo com a expressão (2.47) $ad(i)$ refere-se aos endereços das funções de base não nulas; desta forma bastariam calcular-se ρ funções de base em vez de p , com a conseqüente redução do peso computacional.

Originalmente as funções de base univariáveis do tipo *B-Spline* eram calculadas utilizando-se uma fórmula de diferenças divididas [90]. Verificou-se no entanto que para

⁵ *overlay*

funções de ordem elevada e distribuição de nós não uniforme a metodologia proposta era instável. Em 1972 Cox e DeBoor [86] [87] propuseram um método recursivo numericamente estável, computacionalmente eficiente e capaz de tratar qualquer distribuição de nós.

Considere-se o intervalo entre dois nós consecutivos: $I_j = [\lambda_{j-1} \ \lambda_j]$; a j -ésima função de base função univariável de ordem k representa-se por $N_k^j(x)$ de acordo com a expressão recursiva (2.48).

$$N_k^j(x) = \left(\frac{x - \lambda_{j-k}}{\lambda_{j-1} - \lambda_{j-k}} \right) N_{k-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-k+1}} \right) N_{k-1}^j(x) \quad (2.48)$$

$$N_k^j(x) = \begin{cases} 1, & x \in I_j \\ 0, & \text{caso contrário} \end{cases}$$

As funções de base geradas de acordo com (2.48) apresentam as seguintes propriedades:

- Suporte limitado: Verifica-se que $\begin{cases} N_k^j(x) > 0, & x \in [\lambda_{j-k} \ \lambda_j] \\ N_k^j(x) = 0, & \text{caso contrário} \end{cases}$. Assim, apenas um pequeno número de funções de base ($\rho = k$) contribui para a saída da rede. Reafirma-se então a importante característica das redes neuronais do tipo *B-Spline*, a informação é apreendida e armazenada localmente à custa das k funções de base.
- Partição da unidade: Para cada entrada da rede, verifica-se que: $\sum_j N_k^j(x) \equiv 1, \ x \in [x_i^{\min} \ x_i^{\max}]$. Alguns trabalhos experimentais [58] têm mostrado que a rede pode apresentar um comportamento com alguma instabilidade, caso a relação apresentada neste ponto não se verifique.
- Pertença à classe C^{k-2} no domínio $[x_i^{\min} \ x_i^{\max}]$: Tanto as funções de base como a saída da rede são funções de classe C^{k-2} , ou seja, as suas $k-2$ primeiras derivadas são contínuas no intervalo $[x_i^{\min} \ x_i^{\max}]$; requisito próprio das *Splines* [86].
- Saída da rede limitada aos valores dos pesos: Designado por Ω o conjunto de pesos relativos às funções de base com saída não nula, então a saída da rede $y(x)$ será tal que:

$\min\{\Omega\} \leq y(x) \leq \max\{\Omega\}$. A observação desta propriedade permite deduzir algumas características da rede a partir da observação de pesos locais; para redes com *Splines* de 1ª e 2ª ordens os limites apresentados são atingidos, tal não se verificando por vezes com *Splines* de ordem superior.

As definições feitas até agora sobre as funções de base do tipo *Spline* referem-se ao caso univariável, ou seja, são funções definidas sobre cada uma das n dimensões que compõem a entrada da rede. Em geral, para além desse tipo de funções, é frequente encontrar-se nas redes neuronais do tipo *B-Spline* funções multivariáveis. As *Splines* multivariáveis obtêm-se fazendo o produto tensorial de n funções de base univariáveis, onde, para cada eixo é tomada uma e só uma função de base univariável. A j -ésima função de base multivariável $N_k^j(x)$ é gerada multiplicando n funções de base univariáveis:

$$N_k^j(x) = \prod_{i=1}^n N_{k_i,i}^j(x_i) \quad (2.49)$$

As propriedades verificadas para as *Splines* univariáveis são estendidas de forma natural a *Splines* multivariáveis.

O treino das redes neuronais do tipo *B-Spline* compreende duas etapas fundamentais, que geralmente se alternam durante um processo de aprendizagem: definição ou alteração da estrutura da rede e determinação dos pesos da camada de saída (*ALC*).

A determinação dos pesos lineares pode ser feita minimizando o quadrado do erro da saída da rede. Tomando por base a Fig. 2.8, verifica-se que a matriz dos padrões de entrada X propagados pela rede origina a correspondente matriz A de saídas da camada intermédia. Conhecendo o correspondente vector y de saídas desejadas da rede e a matriz A^+ pseudo-inversa de A , então o vector de pesos w será dado pela expressão (2.38).

A estrutura da rede que se pretende determinar pode ser modelada por uma função f que transforma um vector de entrada na saída: $y = f(x_1, x_2, \dots, x_n)$. Dum modo geral, a função f pode ser decomposta numa soma de funções mais simples; desta forma, o modelo total é determinado pela soma de sub-modelos de complexidade menor. O estabelecimento destes sub-modelos que levam à determinação da estrutura da rede serviram de motivação para o

desenvolvimento dos algoritmos: *ASMODO* (*Adaptive Spline Modelling of Observation Data*) e *ABBMODO* (*Adaptive B-spline Basis function Modelling of Observation Data*) [88], [89].

A ideia exposta no parágrafo anterior pode ser formalizada em termos de conjuntos, da seguinte forma:

Considere-se que os padrões de treino se dispõem segundo as linhas da matriz X ; então podem-se definir sub-conjuntos de padrões tais que: $\bigcup_{u=1}^U X_u \subseteq X$. A rede neuronal completa é definida por sub-modelos S_u tais que, para cada iteração, a saída pode ser vista em função da entrada sendo dada por (2.50).

$$y(x) = \sum_{u=1}^U S_u(x_u) \quad (2.50)$$

De forma análoga, os vectores de pesos associados a cada sub-modelo e as respectivas funções de base verificam as seguintes expressões:

$$w = \bigcup_{u=1}^U w_u \text{ e } a = \bigcup_{u=1}^U a_u \quad (2.51)$$

Assim, a expressão (2.32) pode ser escrita de acordo com (2.52):

$$y(x) = \sum_{i=1}^p a_i(x) w_i \quad (2.52)$$

Verifica-se que o número de pesos necessários e o número de funções de base activas é muito menor de acordo com esta representação o que torna viável a aplicação das redes neuronais a problemas de maior dimensão.

2.2.4.1 Algoritmo de aprendizagem para redes neuronais do tipo *B-Spline*

O *ASMODO* é um algoritmo de treino construtivo que parte dum modelo muito simples; iterativamente, utilizando um processo de refinamento, esse modelo evolui levando ao crescimento da rede neuronal [91]. Tal como foi visto anteriormente, a globalidade da rede neuronal do tipo *B-Spline* é um conjunto de sub-modelos. A cada sub-modelo está associado um sub-conjunto de variáveis de entrada da rede, a cada entrada associa-se um

vector de nós e ordem das funções de base respectivas. O algoritmo de treino determina a totalidade de sub-modelos bem como as variáveis de entrada associadas e os vectores de nós associados a cada variável; o utilizador limitar-se-á a definir a ordem das funções de base. Dependendo do conhecimento que o utilizador tenha sobre os dados que vai utilizar para treino, poderá especificar uma estrutura inicial definindo-se alguns sub-modelos, uni ou multi-variáveis e respectivas entradas associadas. Se tal conhecimento não existir, é comum partir-se duma estrutura que contem um sub-modelo para cada variável de entrada com 0 ou 1 nó interior.

Na Fig. 2.10 esquematizam-se as etapas principais do algoritmo *ASMOD*. De acordo com este algoritmo, após a obtenção do modelo inicial nas condições já enunciadas, tem lugar a geração de modelos ou redes neuronais candidatas. Os modelos candidatos são gerados com recurso a duas operações fundamentais, refinamento e supressão⁶ de nós, que se descrevem de seguida.

- refinamento – esta operação é responsável pelo aumento da complexidade da rede, podendo ocorrer de acordo com três possibilidades:
 - Se os sub-modelos que constituem o presente modelo não contiverem a representação de alguma variável de entrada, então pode-se adicionar ao modelo presente um sub-modelo univariável definido na dimensão correspondente à variável não representada. A ordem e o número de nós interiores do novo sub-modelo serão especificados pelo utilizador, utilizando-se geralmente 0 ou 1 nó interior.
 - Combinação dos sub-modelos existentes através do produto tensorial entre eles, para se originar novos sub-modelos multi-variáveis. Esta operação contribui para aumentar a flexibilidade da rede, podendo-se atingir valores de complexidade elevados. Deverá haver o cuidado de que os modelos obtidos por este meio não ultrapassem a totalidade dos dados de treino.
 - Introdução de novos nós em cada dimensão de cada sub-modelo e correspondentes funções de base univariáveis associadas. Este procedimento é aplicado à totalidade

de sub-modelos aumentando a complexidade destes, devido ao produto tensorial entre as novas funções de base. Embora seja possível adicionar nós em qualquer ponto entre os limites de cada dimensão, o algoritmo prevê a adição de nós em pontos médio de segmentos definidos pelos nós já existentes.

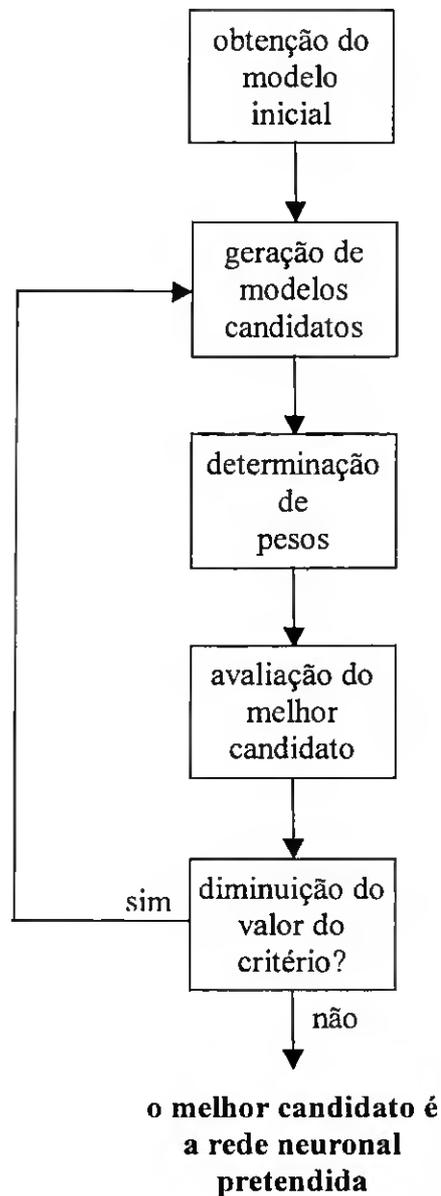


Fig. 2.10 – Algoritmo *ASMOD*.

- supressão – a operação de supressão visa regularizar a complexidade dos modelos, simplificando-os. Verifica-se contudo que a utilização desta operação, com grande

⁶ *pruning*

frequência, impede o desenvolvimento da rede e conseqüentemente o aparecimento de candidatos que possibilitem a continuação do processo iterativo. Existem três possibilidades para realizar a operação de supressão:

- Para os sub-modelos univariáveis sem nós interiores de ordem k , substituí-los por uma função de base de ordem $k-1$; se $k-1=1$, então o sub-modelo deverá ser removido.
- Substituir cada modelo multivariável de dimensão n , por n sub-modelos de dimensão $n-1$.
- Para cada dimensão de cada sub-modelo, remover sistematicamente nós interiores, obtendo-se sucessivamente modelos candidatos com complexidade decrementada de uma unidade.

Com a realização das operações acabadas de descrever obtém-se um conjunto de estruturas candidatas para a camada intermédia da rede neuronal do tipo *B-Spline* (Fig. 2.8). Para se avaliar o desempenho de cada candidato deverão ser determinados os respectivos pesos da camada de saída. De entre as várias possibilidades para determinação dos pesos regista-se a que foi apresentada pela expressão (2.36). Desta forma, propagando o conjunto de treino por cada rede candidata é possível avaliar o respectivo desempenho de acordo com algum critério previamente estabelecido. A totalidade de parâmetros necessários para se definir uma rede designa-se por complexidade dessa rede. Os critérios mais frequentemente utilizados fazem o balanceamento entre a exactidão da rede e a respectiva complexidade, apresentando-se seguidamente, pelas expressões (2.53), (2.54) e (2.55).

$$K = L \ln(J) + p \ln(L) \quad (2.53)$$

$$K(\phi) = L \ln(J) + p\phi, \quad \phi > 0 \quad (2.54)$$

$$K = L \ln(J) + p \ln\left(\frac{L+p}{L-p}\right) \quad (2.55)$$

Nas expressões anteriores p refere-se ao número de pesos, L ao número de padrões de treino e J a uma medida de erro como por exemplo o *MSE*. Os critérios apresentados por

(2.53), (2.54) e (2.55), são conhecidos por *Bayesian information criterion (BIC)*, *Akaike's information criterion* e *Final prediction error*. A utilização de uma das expressões K apresentadas permite determinar o melhor candidato.

Sendo t a iteração presente, se $K(t) < K(t-1)$, então o treino prossegue para a próxima iteração, caso contrário pára, apresentando como solução o melhor candidato da iteração anterior.

O algoritmo acabado de descrever aplica-se à aprendizagem por lotes. A adaptação em linha dos pesos para este tipo de redes neuronais [58] faz-se usualmente através da regra NLMS, segundo a qual a actualização de pesos dada pela expressão (2.7) utiliza o factor de aprendizagem α dado por (2.56).

$$0 < \alpha(t) < \frac{2}{\|\mathbf{a}(t)\|^2} \quad (2.56)$$

Considera-se uma taxa de aprendizagem percentual tx a partir da qual se define o factor de aprendizagem α (2.57).

$$\alpha(t) = \frac{2tx}{100\|\mathbf{a}(t)\|^2} \quad (2.57)$$

Assim se conclui que o valor da taxa tx da expressão anterior é um parâmetro de aprendizagem em linha que vai posicionar o factor de aprendizagem α no intervalo definido por (2.56).

2.2.4.2 Treino de redes neuronais do tipo *B-Spline* usando programação genética

Uma forma alternativa de treino de redes neuronais do tipo *B-Spline* consiste na utilização da programação genética [92].

De acordo com esta técnica, a construção da rede neuronal faz-se duma forma evolutiva utilizando-se elementos denominados funções e terminais. Através desta abordagem, a rede neuronal pode ser avaliada com base em vários critérios de desempenho recorrendo-se para tal à programação genética multi-objectivo. Na secção 2.3.4 proceder-se-á ao enquadramento deste tecnologia dentro dos algoritmos evolutivos.

2.3 ALGORITMOS EVOLUTIVOS

Desde sempre se tem verificado que fenómenos observados na natureza têm servido de fonte de inspiração a cientistas das mais diversas áreas [93]. A experiência e a observação dos resultados provenientes de tais fenómenos têm permitido a construção de dispositivos e modelos que procuram, duma forma consideravelmente mais simplificada, responder de forma 'semelhante' aquela que se verifica no correspondente sistema natural. De entre as muitas situações onde se observam tais analogias podem-se registar exemplos tais como, o olho e a máquina fotográfica, ou o sistema nervoso e as redes neuronais artificiais. De forma análoga, a Genética e o processo da evolução natural das espécies têm servido de fonte de inspiração para investigadores em sistemas artificiais; desenvolvendo-se a partir daí as técnicas que hoje se enquadram em disciplinas tais como computação evolutiva ou algoritmos genéticos.

2.3.1 Surgimento dos algoritmos genéticos

A computação evolutiva teve os seus inícios na década de 50 quando se observaram fortes evidências de que modelos computacionais poderiam ser usados para entender o processo da evolução natural [94]. As primeiras descrições de processos evolutivos surgem em 1958 e 1959 nas publicações de Friedberg, onde é descrito um algoritmo evolutivo para programação automática [95], [96]. Nos anos seguintes (1962) Bremermann aplicou técnicas de evolução simulada a problemas de optimização numérica [97].

Nos finais da década de 50 e durante a década de 60, Box apresentou os seus princípios sobre 'operações evolutivas' para projecto e análise de experiências industriais [98] [99]. Embora os princípios de Box não tenham originado um algoritmo computacional, serviram de base para Spendley [100] desenvolver o método de projecto simplex.

Apesar destes estudos preliminares terem sido vistos com algum cepticismo, foi em meados da década de 60 que ficou clarificada a distinção entre as 3 principais formas de algoritmos evolutivos:

- Programação evolutiva: apresentada na Califórnia por Lawrence Fogel [101].

- Algoritmos genéticos: desenvolvidos na Universidade de Michigan por Holland [102].
- Estratégias evolutivas: desenvolvidas em Berlim por Bienert, Rechenberg e Schwefel [103].

Estes ramos tiveram desenvolvimentos individualizados até 1990, data em que se começou a notar evidente esforço de relacionamento entre as várias comunidades de investigação em algoritmos evolutivos. Tais esforços conduziram a um *workshop* internacional em 1991 em Dortmund intitulado "*Parallel Problem Solving from Nature*" [104]. Desde essa data, a interacção entre os vários ramos não tem parado, tendo-se chegado a um consenso sobre a designação desta nova área de investigação: "Computação Evolutiva"⁷. Em 1993 surge a revista "*Journal of Evolutionary Computation*" publicada pela MIT Press. A continuação do crescente interesse por esta matéria, tornou evidente a necessidade da organização do "*Handbook of Evolutionary Computation*", que nos faculta uma coesa visão desta nova área de investigação.

2.3.2 Noções de algoritmos genéticos

As primeiras ideias subjacentes aos algoritmos genéticos encontram-se registadas nos trabalhos de Holland dos anos 60 [105]. Este autor estipulou os princípios necessários para o entendimento dos sistemas adaptativos. Tais sistemas deveriam ter a capacidade de se auto-modificar como resposta ao ambiente envolvente no qual se desenvolvem. Se por um lado o entendimento das teorias sobre os sistemas adaptativos possibilita a compressão de formas complexas de adaptação existentes nos sistemas naturais, por outro lado, fornece as capacidades para projecto de 'dispositivos' adaptativos robustos.

De acordo com Holland, serão a competição e a inovação que possibilitarão a um sistema adaptativo natural responder, com sucesso e duma forma robusta, a acontecimentos não previstos e a alterações do meio onde se inserem. A aplicação deste princípio à evolução de modelos de sistemas biológicos simples pode-se traduzir pela sobrevivência dos indivíduos melhor adaptados e contínua reprodução de descendentes.

⁷ *evolutionary computation*

Durante as décadas de 60 e 70 a investigação em algoritmos genéticos foi bastante intensa, podendo-se referir o contínuo trabalho de Holland que originou a "Análise Esquemática"⁸ de sistemas adaptativos" [106].

Nos inícios da década de 80 estavam estabelecidos vários grupos de investigação em algoritmos genéticos. Na Universidade de Michigan, investigadores tais como Bethke, Goldberg e Booker continuaram a desenvolver trabalhos em algoritmos genéticos explorando os "Sistemas de classificação de Holland"⁹, que integravam as respectivas teses de doutoramento [107] [108] [109].

A primeira *International Conference on Genetic Algorithms (ICGA)* teve lugar em Pittsburgh em 1985, onde se debateu uma gama alargada de novos desenvolvimentos tanto teóricos como a nível de aplicações dos algoritmos genéticos (Grefenstette 1985). Esta conferência é bianual e originou um grupo electrónico moderado de discussão: *GA List*.

Em 1989 é fundada a *Internal Society for Genetic Algorithms (ISGA)*, que resultou do crescimento da *ICGA* e das actividades associadas, sendo uma das suas primeiras iniciativas o apoio ao *workshop Foundations of Genetic Algorithms (FOGA)* em Bloomington.

A década de 90 foi caracterizada por um grande crescimento em termos de quantidade e diversidade da comunidade ligada aos algoritmos genéticos. Tal facto tem-se reflectido em inúmeras actividades das quais são exemplo *ICGA* e *FOGA*. Foram desenvolvidos novos paradigmas, como por exemplo, os algoritmos genéticos desregrados¹⁰ [110] e a programação genética [111].

Tem-se assistido a uma forte interacção entre os grupos de trabalho ligados às várias áreas da computação evolutiva conduzindo a novos algoritmos evolutivos híbridos. Os algoritmos genéticos têm de igual forma revelado grande adequação no campo das aplicações nas mais diversas áreas da Engenharia, Investigação Operacional e Programação Automática.

⁸ *schema analysis*

⁹ *Holland's classifier systems*

¹⁰ *messy GAs*

Dentro da área de controlo podem-se registar algumas aplicações [93]:

- Sintonia óptima de controladores: A determinação dos parâmetros de controladores é um problema clássico de optimização para o qual a aplicação dos algoritmos genéticos tem-se apresentado bastante adequada. São exemplo disso os trabalhos de Hunt levados a cabo em 1992 [112], [113] e [114], nos quais aplicou os algoritmos genéticos a várias classes de problemas. Para cada caso os parâmetros do controlador evoluíam de acordo com um algoritmo genético simples codificado em binário. As situações de instabilidade eram tratadas como elementos de reduzido valor de adaptação.
- Análise da estabilidade robusta: Uma das abordagens da análise da estabilidade debruça-se sobre a incerteza dos parâmetros do controlador. Desta forma, definem-se as "*Q-box*" como sendo vizinhanças centradas nos parâmetros. Os trabalhos desenvolvidos por Murdock apresentam um algoritmo genético para determinação de *Q-box* [115].
- Identificação de sistemas: Os algoritmos genéticos foram utilizados para identificação de sistemas a partir dos dados de entrada e correspondente saída. De acordo com os trabalhos de Kristinsson e Dumont a identificação pode ocorrer em diferido ou em linha, sendo adequada tanto para sistemas contínuos como para sistemas discretos [116].
- Sistemas de classificação: Estes sistemas codificam a criação de regras em cadeias de caracteres, denominadas classificadores. De acordo com cada regra, existe uma condição que será testada e consoante o seu valor lógico desencadear-se-á ou não uma acção. Uma das componentes do sistema de classificação poderá ser o algoritmo genético que se encarrega de determinar, eventualmente, possíveis melhores regras. Este princípio foi utilizado por Fogarty em 1990 para otimizar a combustão numa instalação com múltiplos queimadores [117].
- Controlo Difuso: De acordo com a classificação de Linkens e Nyongesa [118] podem-se destacar 4 aspectos principais no projecto de controladores difusos:
 - selecção de factores de escala;
 - obtenção de funções membro óptimas;

- dedução de regras base;
- adaptação em linha das regras base.

Karr desenvolveu trabalho no sentido da adaptação em linha das funções membro optimizadas [119], [120].

- Planeamento de trajectórias para robots: A geração automática de trajectórias para robots é em geral um problema multi-modal cujas soluções têm comprimento variável. A utilização [121] de algoritmos genéticos para tratamento deste problema requer a introdução de operadores genéticos especializados que recorrem à teoria de Lamarck¹¹.

No que se refere a problemas de optimização registam-se, de seguida, algumas das características que têm motivado o uso dos algoritmos genéticos [122]:

- Permite a optimização de parâmetros contínuos ou discretos.
- Não requer conhecimento de derivadas da superfície que se está optimizando.
- Utiliza simultaneamente vários pontos da superfície, com o objectivo de se encontrar o óptimo.
- É adequado a processamento computacional paralelo.
- Tem-se mostrado adequado a problemas de optimização de superfícies com um grau de complexidade bastante elevado, transpondo com alguma facilidade o problema da paragem antecipada em óptimos locais.
- Permite a codificação de parâmetros, desenrolando-se o processo de optimização em termos dos parâmetros codificados em vez dos próprios parâmetros.
- Apresentam bastante flexibilidade em termos do tipo de dados utilizados. Facilmente se adaptam a dados gerados numericamente, experimentais ou obtidos por via analítica.

¹¹ De acordo com Lamarck, algumas características adquiridas pelos pais passam aos filhos. Esta teoria, dum modo geral não é aceite na evolução natural das espécies.

2.3.3 Operações básicas dos algoritmos genéticos

A partir dos aspectos introdutórios descritos nas secções anteriores pode-se dizer resumidamente que os algoritmos genéticos são algoritmos estocásticos de procura baseados no mecanismo da selecção natural e na genética.

O espaço de procura dum algoritmo genético é a população de potenciais soluções onde se aplica o princípio da sobrevivência dos indivíduos mais adaptados.

O desenrolar [123] dum processo de optimização utilizando este tipo de algoritmos decorre segundo uma sucessão de gerações. Em cada geração ocorrerão uma série de operações baseadas na selecção natural e na genética que visam a preservação das soluções que melhor se adaptam à resposta para o problema em curso.

Na Fig. 2.11 apresentam-se as etapas principais dum algoritmo genético simples.

Os indivíduos que compõem a população inicial são codificados em cadeias de caracteres ou cromossomas. A numeração binária é bastante usada para codificação dos cromossomas, contudo, outras bases de numeração também têm sido usadas; têm-se registado populações em que os cromossomas são codificados com números inteiros e outras situações existem em que a codificação é real. Verifica-se então que existe um mapeamento entre os valores que codificam o cromossoma (genótipo) e os valores das correspondentes variáveis de decisão (fenótipo). Excluindo a situação em que a codificação é real o algoritmo genético opera sobre o genótipo.

A avaliação da aptidão de cada indivíduo é conseguida a partir da descodificação do cromossoma e utilização da função objectivo. Desta forma consegue-se avaliar o grau de adaptação do indivíduo para o problema que está em resolução. Se o critério de optimização for cumprido, então o melhor indivíduo será descodificado obtendo-se a solução do problema, caso contrário irão ocorrer as operações que conduzem à nova geração.

A criação da nova população começa com a operação de selecção; de acordo com este processo é determinado o número de vezes (proporcional à sua aptidão) que cada indivíduo é escolhido para reprodução, ou seja, o número de descendentes que irá produzir. Dado que este número é um inteiro então verifica-se a existência de erro de selecção. Para que o erro

de selecção seja tanto quanto possível minimizado, a operação de selecção é realizada com recurso a algoritmos estocásticos. Desta forma, para cada indivíduo, o número esperado de descendentes deverá corresponder exactamente à sua aptidão. O algoritmo proposto por Baker [124] denomina-se *Stochastic Universal Sampling (SUS)* e permite erros de selecção reduzidos. Este algoritmo aparece como alternativa ao algoritmo de Goldberg, *Roulette Wheel Selection (RWS)* [39], em que o erro de selecção é considerável.

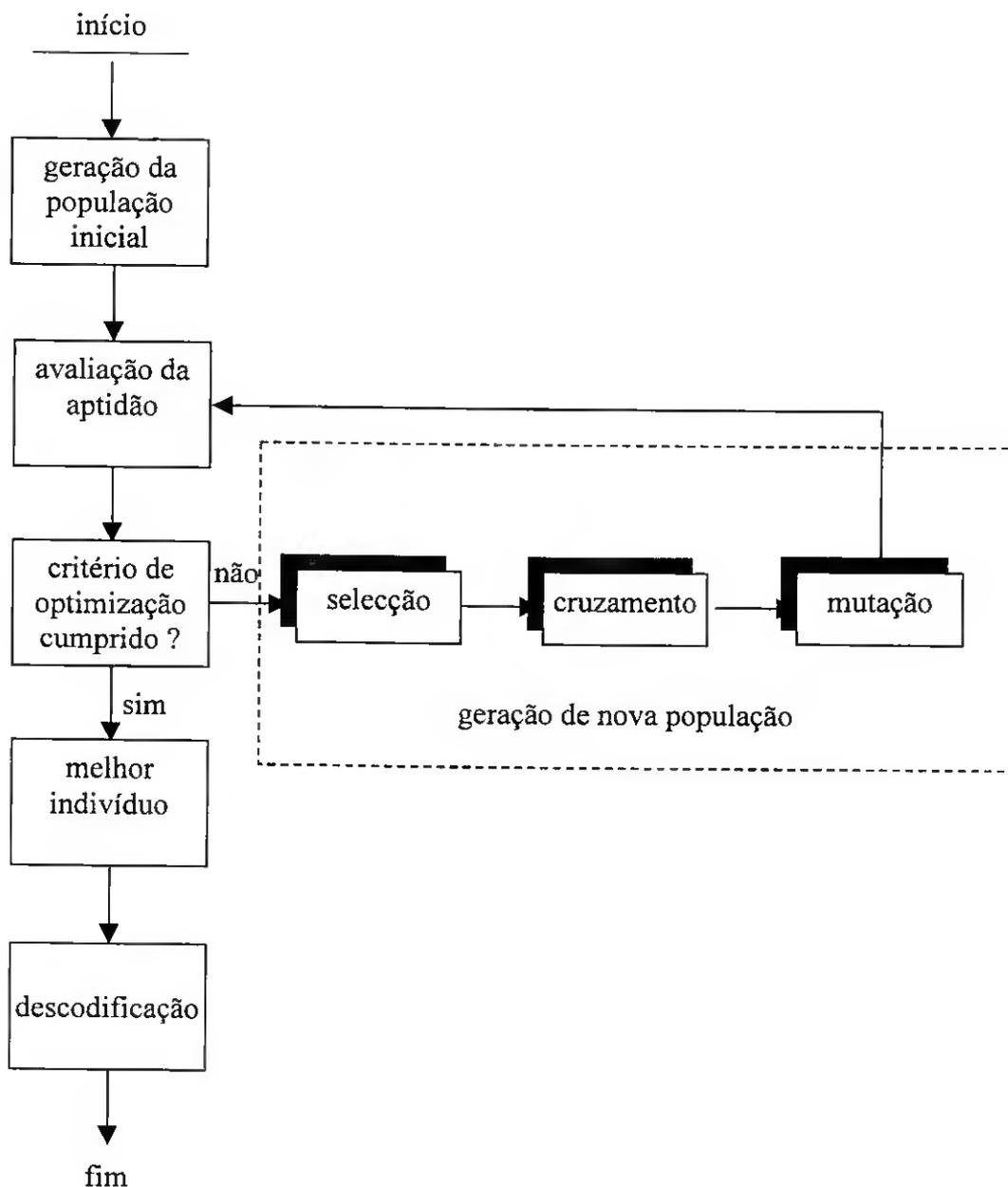


Fig. 2.11 – Algoritmo genético simples.

Concluída a selecção tem-se a população com múltiplas cópias dos indivíduos mais aptos; a manipulação do genótipo dos progenitores com vista ao aparecimento de novos elementos na população (descendentes), é feita à custa de 2 operadores genéticos: cruzamento e mutação.

O cruzamento ou recombinação é o processo pelo qual os indivíduos trocam informação genética entre eles. Quando o cruzamento ocorre num ponto simples (cruzamento de ponto simples); é escolhido um ponto aleatoriamente nos dois cromossomas e as duas partes definidas em cada cromossoma são trocadas. Podem-se encontrar situações de cruzamento em ponto duplo e cruzamento uniforme:

- cruzamento em ponto duplo: são seleccionados dois pontos de cruzamento para cada cromossoma, originando três trocas de segmentos, em vez de duas como acontecia no cruzamento de ponto simples [125].
- cruzamento uniforme: cada bit é permutado independentemente entre indivíduos, com uma dada probabilidade [126].

O operador mutação permite que o genótipo dum indivíduo seja alterado com uma determinada probabilidade. Dum modo geral, apenas uma pequena parte do cromossoma é alterada, o novo indivíduo herda a maior parte das características daquele que o originou. Este operador é mais simples do que o operador de cruzamento, no entanto, verifica-se que a aplicação da selecção e da mutação alternadamente permitem por si só manter o processo evolutivo.

Depois de obtidos os descendentes de uma dada população, coloca-se o problema de como reinseri-los na população inicial. Existem várias alternativas em relação à coexistência de duas gerações consecutivas. Os algoritmos genéticos mais simples adoptam a substituição da população progenitora pela população de descendentes, no entanto, noutras situações é dada a possibilidade dos descendentes coexistirem com os respectivos progenitores. De um modo geral a população progenitora é parcialmente substituída pela população de descendentes, entrando-se em conta com o valor da aptidão. Estratégias de reinserção que preservam os indivíduos com maior valor de aptidão são conhecidas por estratégias elitistas.

2.3.4 Programação genética

Na presente secção apresenta-se um resumo dos fundamentos da programação genética [127] [128], no que se refere ao treino das redes neuronais do tipo *B-Spline*.

Este tipo de programação procura testar em simultâneo e duma forma iterativa um conjunto de estruturas candidatas à solução do problema, sendo como tal, um procedimento intrinsecamente paralelo.

Fazendo-se uma confrontação com a secção anterior (2.3.3), facilmente se conclui que alguns dos conceitos sobre os quais assenta a programação genética são comuns aos algoritmos genéticos, podendo-se referir os seguintes pontos:

- Definição da população de indivíduos,
- Classificação dos indivíduos de acordo com os critérios estabelecidos,
- Estabelecimento do conjunto de terminais e do conjunto de funções,
- Classificação dos indivíduos,
- Selecção dos indivíduos,
- Troca de informação entre indivíduos para criação da nova geração,
- Mutação com determinada probabilidade,
- Estabelecimento da estrutura de rede neuronal correspondente à informação genética disponível.
- Avaliação dos indivíduos com base nos critérios adequados.

Estruturalmente, as redes candidatas geradas ao longo do funcionamento do algoritmo são representadas por uma árvore constituída por funções e terminais.

O algoritmo utilizado [92] resume-se na Tab. 2.2.

Uma das vantagens da utilização de algoritmos evolutivos e nomeadamente programação genética, é a de que permite uma fácil optimização de múltiplos objectivos. Nesta

abordagem, só muito raramente existe uma única solução, dado normalmente os objectivos serem conflituosos entre si. Existe sim um conjunto de soluções de compromisso, denominado de "soluções de Pareto" ou não dominadas [93], que satisfazem objectivos pré-definidos. A técnica desenvolvida em [92] e utilizada no capítulo 5 implementa para construção de redes neuronais *B-Spline*, o conceito de ordenação de soluções de Pareto proposta em [93].

- leitura de dados e parâmetros de aprendizagem
- criação da população inicial
- para geração=1 até máximo de gerações
 - para indivíduo=1 até total de indivíduos
 - determina modelo para o indivíduo candidato
 - avalia modelo candidato
 - calcula aptidão dos candidatos
 - selecciona melhor candidato
 - renovação da população
- se geração = máximo de geração, fim do programa

Tab. 2.2 – Algoritmo de programação genética para treino de redes neuronais do tipo *B-Spline*.

2.4 CONCLUSÕES

Neste capítulo foram apresentados alguns fundamentos sobre as técnicas de *softcomputing* utilizadas no âmbito desta dissertação. A primeira parte do capítulo versa sobre vários tipos de redes neuronais. Após uma descrição sobre as origens biológicas destas tecnologias, são apresentados os principais fundamentos sobre redes neuronais do tipo *MLP*, *RBF* e *B-Spline*, no que se refere às topologias e aos algoritmos de treino.

O capítulo termina com fundamentos sobre os algoritmos genéticos; tal como aconteceu com as redes neuronais, são apresentados os fundamentos biológicos, bem como o enquadramento deste tipo de algoritmos em classes mais vastas de tecnologias relacionadas. Por último, é feita uma breve revisão sobre as principais operações que

compõem os algoritmos genéticos e sua utilização na programação genética aplicada ao treino de redes neuronais do tipo *B-Spline*.

3. ARQUITECTURA ADAPTATIVA PARA AUTO-SINTONIA DO CONTROLADOR *PID*

3.1 INTRODUÇÃO

No capítulo 1 foi feito um levantamento de algumas estratégias de sintonia do controlador *PID*. Nessa altura verificou-se que as técnicas que utilizam *softcomputing*, para a resolução de problemas que envolvem modelação e optimização de critérios, são usadas com grande frequência e apresentam-se dum modo geral bem sucedidas. Neste capítulo, apresenta-se uma proposta de arquitectura de controlo e respectivos métodos de sintonia do controlador *PID*.

Conforme ficará explícito, dentro do problema em estudo enquadram-se procedimentos de modelação e optimização. Assim, na secção 3.2 apresenta-se a malha de controlo adoptada. Na secção 3.3 o sistema controlado será enquadrado na arquitectura de sintonia adaptativa do controlador. Segue-se a secção 3.4 onde serão apresentados alguns testes iniciais da arquitectura proposta, será dado especial ênfase à parte de optimização. O capítulo termina com a secção 3.5 onde, para além das leituras de resultados e conclusões referentes ao presente capítulo, serão apresentados os capítulos seguintes que se debruçarão essencialmente sobre os blocos de modelação referentes à arquitectura proposta.

3.2 SISTEMA DE CONTROLO

A malha de controlo utilizada resulta daquela que se apresenta na Fig. 1.1 colocando-se na malha de retroacção o bloco derivativo em série com o filtro passa baixo, $D(s) = \frac{1+st_d}{1+st_f}$, com a constante de tempo do filtro escolhida tal que $t_d = 10t_f$. O bloco proporcional + integral $PI(s) = k_c \left(1 + \frac{1}{st_i} \right)$, está inserido na malha para a frente, resultando a topologia que se apresenta na Fig. 3.1 [2].

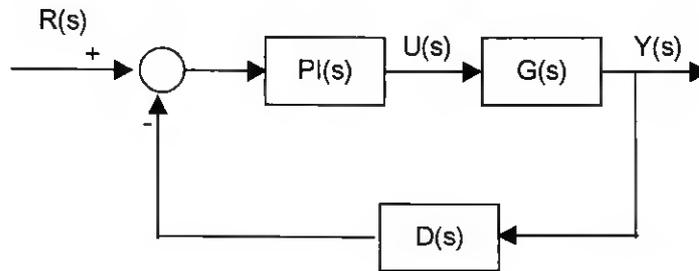


Fig. 3.1 – Processo + controlador PID.

Para que a sintonia do controlador ocorra em linha e em tempo real, é necessário um procedimento de identificação do sistema $G(s)$, que não perturbe o seu normal funcionamento. Na sub-secção seguinte apresentam-se os métodos de cálculo das medidas de identificação utilizadas nos processos em estudo.

3.2.1 Identificação de sistemas

Considere-se o processo $G(s)$ estável, com função de transferência dada por (3.1):

$$G(s) = K_p e^{-Ls} \frac{\prod_{i=1}^{n_z} (1 + T_{z_i} s)}{\prod_{i=1}^{n_p} (1 + T_{p_i} s)} \quad (3.1)$$

As medidas de identificação [28] do sistema $G(s)$, desenvolvidas a partir dos trabalhos de Nishikawa et al. [129] e posteriormente utilizadas por Lima, J. [130], representam uma extensão da técnica referida na secção 1.4.2.1. De acordo com esta metodologia, essas medidas de identificação são dadas pela função $F(\alpha)$ definida por (3.2).

$$F(\alpha) = \frac{G(\alpha)}{K_p}, \alpha \in \mathfrak{R} \quad (3.2)$$

Tomando em consideração a expressão (3.1), a função $F(\alpha)$ apresenta-se de acordo com (3.3).

$$F(\alpha) = e^{-L\alpha} \frac{\prod_{i=1}^{n_z} (1 + T_{z_i} \alpha)}{\prod_{i=1}^{n_p} (1 + T_{p_i} \alpha)} \quad (3.3)$$

Os pontos α onde F vai ser avaliada são proporcionais a σ , verificando-se a relação (3.4):

$$\alpha = \frac{\sigma}{T_T}, \sigma \in [0.1, 10] \text{ e } T_T = L + \sum_{i=1}^{n_p} T_{p_i} - \sum_{i=1}^{n_z} T_{z_i} \quad (3.4)$$

Assim, substituindo (3.4) em (3.3), obtêm-se as medidas de identificação dadas por (3.5).

$$F(\sigma) = e^{-\frac{L\sigma}{T_T}} \frac{\prod_{i=1}^{n_z} \left(1 + \frac{T_{z_i} \sigma}{T_T}\right)}{\prod_{i=1}^{n_p} \left(1 + \frac{T_{p_i} \sigma}{T_T}\right)} \quad (3.5)$$

Nas secções 5.2.3 e 5.2.4, será justificado o número de medidas de identificação a utilizar para cada sistema, bem como os valores de σ para os quais essas medidas serão avaliadas.

A expressão (3.5) permite a avaliação analítica das medidas de identificação; contudo, para que o processo de identificação seja viável em linha, estas medidas deverão ser determinadas numericamente. De acordo com trabalhos anteriores [28] e [130], a avaliação numérica das medidas de identificação pode ser feita tanto em malha aberta como em

malha fechada, recorrendo-se ao cálculo numérico de $S(\sigma)$ ou $S_u(\sigma)$ respectivamente, de acordo com as expressões (3.6).

$$S(\sigma) = \int_0^{\infty} e^{-\frac{\tau\sigma}{T_T}} [y(\infty) - y(\tau)] d\tau; S_u(\sigma) = \int_0^{\infty} e^{-\frac{\tau\sigma}{T_T}} [u(\infty) - u(\tau)] d\tau \quad (3.6)$$

A diferença essencial entre os integrais $S(\sigma)$ e $S_u(\sigma)$ reside no facto do primeiro, utilizado na identificação em malha aberta, recorrer ao sinal de saída $y(t)$; enquanto que, a identificação em malha fechada recorre ao cálculo de $S_u(\sigma)$, que utiliza o sinal de controlo $u(t)$. Para ambas as situações, trabalhos realizados anteriormente [130], provaram que a avaliação dos integrais requer que o regime transitório se encontre completamente extinto. Verificou-se igualmente nessa altura que a escolha do passo de integração é determinante na exactidão do cálculo dos respectivos integrais.

Conforme ficou provado em [28], na situação de identificação em malha aberta, o factor das constantes de tempo T_T será avaliado numericamente de acordo com (3.7):

$$T_{T_{m.a.}} = \frac{S(0)}{y(\infty)} \quad (3.7)$$

Recorrendo a este valor, juntamente com $S(\sigma)$ de (3.6), obtêm-se as medidas de identificação avaliadas em malha aberta de acordo com (3.8):

$$F_{m.a.}(\sigma) = 1 - \frac{\sigma S(\sigma)}{S(0)} \quad (3.8)$$

Para se levar a cabo a identificação do sistema em malha fechada é necessário avaliar K_p numericamente. Supondo uma referência em degrau unitário de amplitude B , verifica-se que $K_p = \frac{B}{\lim_{t \rightarrow \infty} u(t)}$.

Sabendo-se que $\eta = \frac{t_d}{T_T} = 10$, recorrendo-se aos parâmetros do controlador e ao integral $S_u(\sigma)$, o factor das constantes de tempo T_T bem como as medidas de identificação em

malha fechada podem ser determinados de acordo com as expressões (3.9) e (3.10) respectivamente.

$$T_{T_{m.f.}} = \frac{t_i}{K_p k_c} + t_d \left(1 - \frac{1}{\eta} \right) - \frac{S_u(0)}{u(\infty)} \quad (3.9)$$

$$F_{m.f.}(\sigma) = \frac{1 + \frac{\sigma t_d}{\eta T_T}}{1 + \frac{\sigma t_d}{T_T}} \left(\frac{B}{B - \frac{\sigma K_p S_u(\sigma)}{T_T}} - \frac{\sigma t_i}{k_c K_p (T_T + \sigma t_i)} \right) \quad (3.10)$$

Conforme ficou explícito nesta secção, a identificação do processo em linha necessita de aquisição e análise do sinal $y(t)$, na situação de funcionamento em malha aberta, ou do sinal de controlo $u(t)$, caso o funcionamento ocorra em malha fechada, após uma perturbação em degrau na entrada de referência. A estratégia adoptada para esse propósito bem como a sintonia do controlador serão detalhadas na secção seguinte.

3.3 MECANISMO DE SINTONIA DO SISTEMA DE CONTROLO

Pretende-se inserir a malha de controlo representada na Fig. 3.1 numa arquitectura que seja capaz de supervisionar o seu funcionamento em linha, com vista a uma sintonia optimizada e automática do controlador *PID* (ver Fig. 3.2).

A arquitectura proposta [3] evidencia dois blocos (ou sub-sistemas) principais: Sub-sistema A e Sub-sistema B. O sub-sistema B é formado pelo sistema controlado e foi objecto da secção 3.2.

O sub-sistema A ocupa-se da supervisão e sintonia optimizada do controlador *PID* para os critérios que se julgarem convenientes, distinguindo-se 3 tarefas principais:

- Identificação de sistemas.
- Determinação de valores óptimos.

- Actualização dos parâmetros do controlador *PID*.

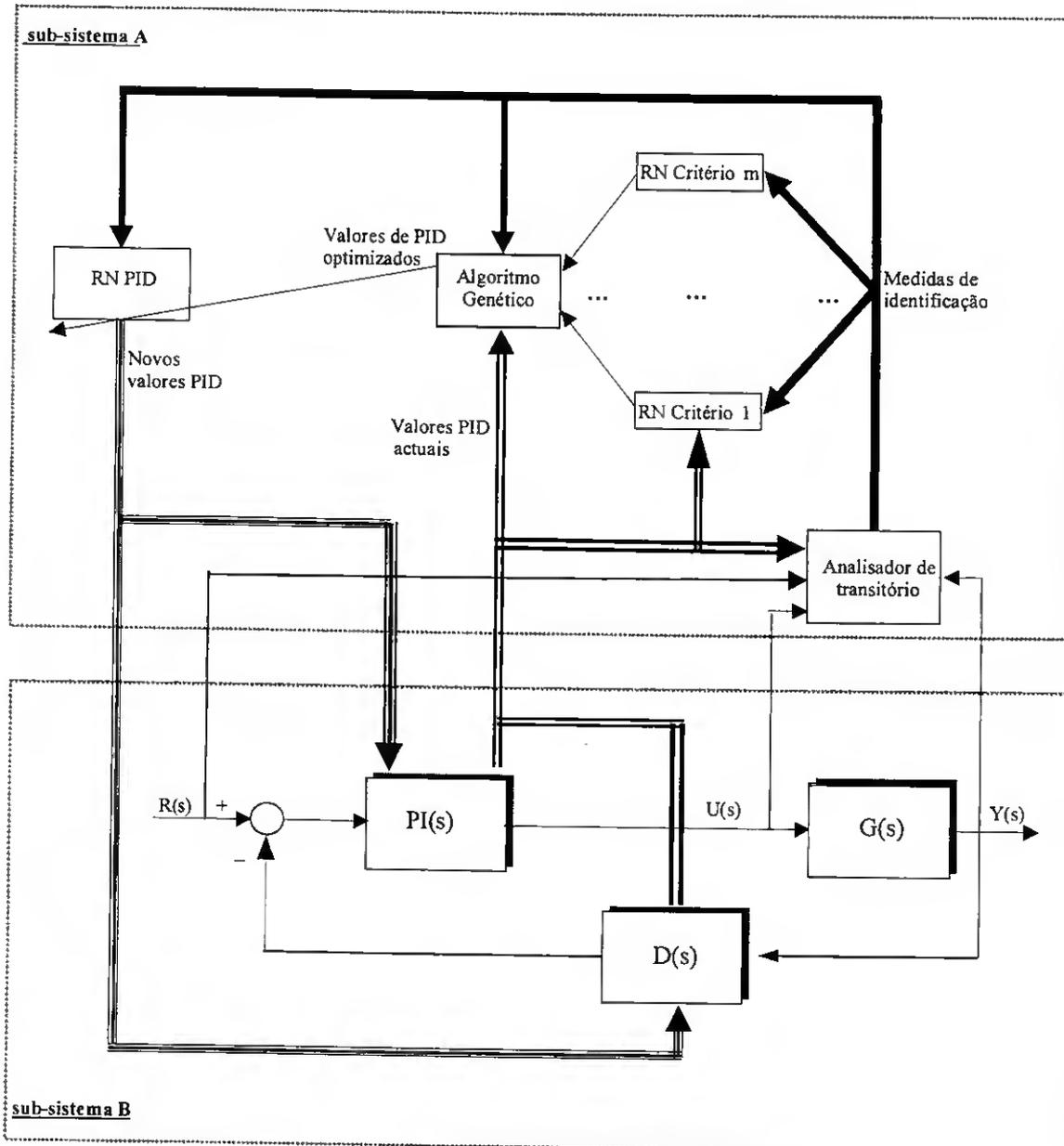


Fig. 3.2 – Esquematização do processo de auto-sintonia do controlador *PID*.

Com vista a uma leitura mais facilitada da Fig. 3.2 e antes de se passar a uma descrição mais detalhada das componentes do sub-sistema de sintonia, apresentam-se de seguida algumas das convenções adoptadas. Assim, os valores das medidas de identificação estão disponibilizados nas linhas simples a *bold*. Os parâmetros *PID* actuais encontram-se nas linhas duplas e os novos parâmetros *PID* são disponibilizados nas linhas triplas. Os blocos

referentes ao sistema controlado (sub-sistema B) estão a sombreado, enquanto que os blocos do sistema de supervisão (sub-sistema A) apresentam-se a linha simples e correspondem a algoritmos, para os quais se desenvolveu o correspondente *software*.

A tarefa central do sub-sistema A de supervisão situa-se no bloco de optimização, denominado "Algoritmo Genético". Este bloco destina-se à optimização de critérios de desempenho (p. ex. *ITAE*) em função dos parâmetros *PID*, para um dado processo $G(s)$; para tal serão utilizados algoritmos genéticos.

A estratégia apresentada deverá estar preparada para funcionar em linha em tempo real sem que o normal desempenho do controlador seja perturbado. Será, assim, de esperar que o espaço de procura utilizado pelos algoritmos genéticos compreenda alguns indivíduos que poderão, eventualmente, levar o sistema de controlo à instabilidade. Desta forma, a avaliação da função objectivo recorre a modelos dos critérios de desempenho, em vez de utilizar o funcionamento do sistema para avaliar esses mesmos critérios.

A necessidade de existência de modelos é reforçada pelo facto de se constatar que a utilização directa de medidas do processo, para a avaliação do desempenho de todos os indivíduos que integram as populações ao longo das várias gerações do algoritmo genético, implica por um lado um dispêndio de tempo incompatível com o funcionamento em tempo real, e por outro lado sujeita o sistema a um elevado número de perturbações. Em vez disso, a utilização de modelos apropriados tornará todo o processo mais célere. Os modelos utilizados são redes neuronais treinadas, tal como se esquematiza nos blocos "RN critério j ", $1 \leq j \leq m$, da parte direita do sub-sistema A da Fig. 3.2.

Serão utilizadas tantas redes neuronais para modelação dos critérios de desempenho, quantas as especificações que se pretendam ver optimizadas.

Cada bloco "RN critério j " foi abordado de duas formas alternativas:

A. Mapeamento directo no critério: De acordo com este método, as medidas de identificação do processo são concatenadas com o vector *PID*, formando-se o espaço de entrada para aprendizagem das redes neuronais. O espaço de saída é formado pelos correspondentes valores do critério de desempenho. O mapeamento conseguido é do tipo: $MI + PID \rightarrow \text{Critério}$.

B. Construção dum modelo intermédio: Para cada processo $G(s)$, são estabelecidas funções que modelam o comportamento do critério, sendo a variável independente o vector PID : $Critério \equiv f(PID)$. Estas funções f têm estruturas semelhantes para os diferentes processos e como tal, podem ser parametrizadas, fazendo-se corresponder a cada processo $G(s)$, um vector de parâmetros $P = [p_1 \cdots p_n]$. Com esta abordagem, a avaliação dos critérios faz-se em duas fases:

1. Aprendizagem de redes neuronais para obtenção dos parâmetros do modelo de f . Serão utilizadas tantas redes neuronais quantos os parâmetros da função f . O espaço de entrada é constituído pelas medidas de identificação dos processos $G(s)$; o espaço de saída, é constituído pelo correspondente parâmetro para todos os processos $G(s)$; obtendo-se assim mapeamentos do tipo: $MI \rightarrow p_i, 1 \leq i \leq n$.
2. A partir dos parâmetros avaliados pelos mapeamentos anteriores, constroi-se a função f , a partir da qual, conhecendo-se o vector PID actual se determina o critério: $Critério \equiv f(PID)$.

Como se pode concluir, ambas as abordagens levadas a cabo para cada bloco "RN critério j ", têm em comum o facto de necessitarem das medidas de identificação do processo $G(s)$ bem como do vector PID actual. De acordo com a Fig. 3.2, a avaliação das medidas de identificação está a cargo do bloco "Analisador de transitório". Numa situação inicial em que o sistema esteja em malha aberta, este bloco recolhe $y(t)$ e implementa a expressão (3.8); caso o sistema já esteja em malha fechada, será utilizado $u(t)$ e a avaliação numérica das medidas de identificação é feita pela expressão (3.10).

O bloco de optimização "Algoritmo Genético" recorre ao apoio dos blocos: "RN Critério 1", ..., "RN Critério m ", e "Analisador de transitório". Os valores de PID resultantes da optimização poderão ser directamente aplicados ao controlador, no entanto, trabalhos anteriores [131] provaram que este procedimento não conduzia a uma sintonia adequada. Assim, opta-se por se utilizar redes neuronais de sintonia esquematizadas no bloco "RN PID ", que agem como filtros. Este bloco, "RN PID ", é composto por 3 redes neuronais responsáveis pelo mapeamento das medidas de identificação do processo $G(s)$, em cada um dos 3 parâmetros PID optimizados para determinado critério: $MI \rightarrow PID_o$. O

funcionamento do bloco de optimização conduz a um novo padrão PID_o , que, juntamente com as medidas de identificação, permitirá a adaptação das redes neuronais de sintonia e a determinação de novos parâmetros PID .

3.3.1 Ciclo de funcionamento do sistema adaptativo

Após a apresentação da arquitectura proposta, descreve-se de seguida o seu funcionamento.

Antes da arquitectura de auto-sintonia adaptativa entrar em funcionamento, deverão ter sido construídos modelos (blocos "RN critério j" da Fig. 3.2), para os critérios de desempenho, tendo em conta o conjunto de processos $G(s)$ que formam o conjunto base de experimentação. Analogamente, deverá ter ocorrido um treino prévio das redes neuronais de sintonia (bloco "RN PID"), com vista ao mapeamento das medidas de identificação dos sistemas que integram o conjunto base de experimentação, nos parâmetros PID optimizados para um dado critério que se entenda conveniente.

O processo de sintonia inicia-se com o sistema de controlo em malha aberta e com a aplicação de uma onda quadrada à entrada de referência (Fig. 3.3).

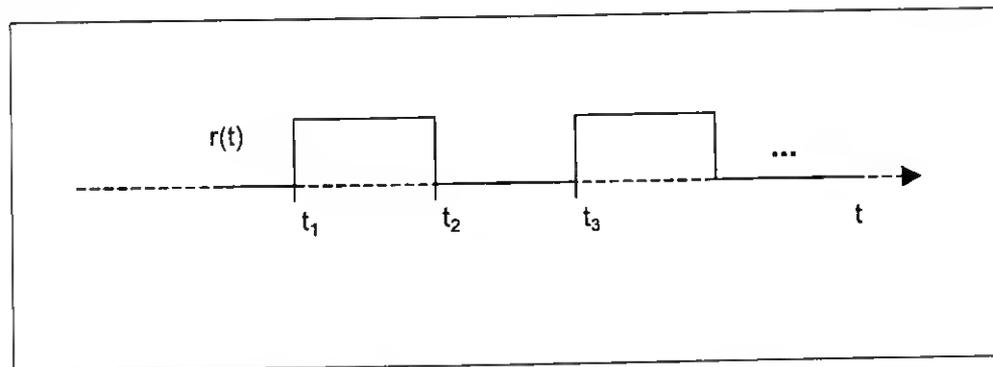


Fig. 3.3 – Momentos de actuação no processo de auto-sintonia do controlador PID .

A partir do instante t_1 em que se dá a 1ª transição de $r(t)$, o sinal $y(t)$ é recolhido pelo bloco "Analisador de transitório" até ao instante imediatamente anterior à transição seguinte (t_2). Nesse momento tem lugar a avaliação numérica das medidas de identificação em malha aberta de $G(s)$, recorrendo-se para tal às expressões (3.6), (3.7) e (3.8). As medidas de identificação acabadas de avaliar serão propagadas pelas redes neuronais de sintonia (bloco "RN PID"), que irão disponibilizar os parâmetros PID optimizados para o critério

previamente estabelecido; neste instante t_2 , o sistema é colocado em malha fechada e o bloco "Analisador de transitório" passa a recolher o sinal $u(t)$. No momento anterior à próxima transição t_3 , serão utilizadas as expressões (3.6), (3.9) e (3.10) para a avaliação das medidas de identificação em malha fechada. Durante este intervalo de tempo $[t_2 \ t_3]$ o bloco "Analisador de transitório" terá igualmente recolhido e avaliado os sinais necessários ao cálculo dos m critérios de desempenho.

Estando nesta altura o sistema a funcionar em malha fechada, ficam disponíveis os valores actuais do *PID* (linhas duplas da Fig. 3.2). Assim, a avaliação dos novos valores do *PID* vai ser feita executando-se sequencialmente as 3 etapas seguintes:

1. **Adaptação dos modelos:** As medidas de identificação juntamente com os valores actuais do *PID* e os m critérios de desempenho constituem um padrão entrada/saída de adaptação para cada uma das m redes neuronais dos blocos "RN Critério j ".
2. **Optimização:** O bloco de optimização "Algoritmo Genético" executará um procedimento de optimização, em que a avaliação das funções objectivo utiliza os modelos adaptados de acordo com o ponto anterior.
3. **Geração de novos *PID*:** Os parâmetros *PID* resultantes do processo a que se refere o ponto anterior juntamente com as medidas de identificação constituirão um padrão entrada/saída de adaptação, para cada uma das 3 redes neuronais de sintonia do bloco "RN *PID*". Depois de executada esta adaptação, os novos parâmetros *PID* resultarão da propagação das medidas de identificação pelas redes neuronais referentes ao bloco "RN *PID*". Os parâmetros assim determinados são de imediato aplicados ao controlador.

A partir desde momento, não se verificando alterações no processo, o sistema manter-se-á em malha fechada; o ciclo composto por identificação, adaptação e sintonia repete-se para cada transição de $r(t)$.

Será de esperar que o processo adaptativo subjacente à estratégia adoptada conduza a modelos do processo cada vez mais exactos, contribuindo assim para sintonias progressivamente mais precisas.

De acordo com a estratégia acabada de descrever, atendendo a que a optimização é sempre precedida da identificação do processo, concluí-se que a arquitectura proposta permite a acomodação de processos variantes no tempo. Nestas circunstâncias, ocorrendo variação do processo (por exemplo, no patamar $[t_1 t_2]$) com a conseqüentemente alteração nas medidas de identificação, deverá ser esperada a conclusão de meio período da onda quadrada, durante a qual é efectuada a aquisição do sinal. Verifica-se então que só em t_3 se volta a determinar nova identificação do processo e sintonia do controlador. Como foi já referido, para que a identificação do processo seja exacta, o tempo entre duas transições de $r(t)$ deverá ser tal que o regime transitório se veja completamente extinto, condicionando, assim, a taxa máxima de variação do processo.

3.4 OPTIMIZAÇÃO GENÉTICA DO BLOCO DE SINTONIA

Terminada a apresentação da arquitectura de auto-sintonia adaptativa, procede-se à descrição de alguns procedimentos iniciais de teste respeitantes à mesma [3]. Por se tratar de testes iniciais, algumas simplificações vão ser feitas. Assim, o processo considerado ao longo deste capítulo manter-se-á invariante no tempo, sendo a sua função de transferência

$G(s) = \frac{e^{-s}}{s+1}$. De salientar que para este caso, não há necessidade de se utilizar medidas de

identificação dado se tratar simplesmente de um mesmo processo.

Os testes agora apresentados vão incidir essencialmente sobre o bloco "Algoritmo Genético" da parte central do sub-sistema A de supervisão e sintonia.

A sintonia que se pretende realizar nesta fase inicial procura abranger dois objectivos:

1. Seguimento da referência: O cumprimento deste objectivo visa a obtenção duma sintonia tal que a saída $y(t)$ siga tão próximo quanto possível a entrada $r(t)$. Definindo-se $e(t) = y(t) - r(t)$, o desempenho deste objectivo será tanto maior quanto menor for o *ITAE*:

$$ITAE = \int t |e(t)| dt \quad (3.11)$$

2. Rejeição de perturbações à saída: O sistema deverá ser imune a perturbações que ocorram à saída. Desta forma, considerando ainda a Fig. 3.1, fazendo a referência nula e adicionando uma perturbação do tipo degrau à saída de $G(s)$, pretende-se que $y(t)$ tenda para zero tão rapidamente quanto possível. O cumprimento deste objectivo implica a minimização do $ITAY$:

$$ITAY = \int t |y(t)| dt \quad (3.12)$$

3.4.1 Função objectivo

A função objectivo, que se pretende minimizar pelo algoritmo genético representado no sub-sistema A da Fig. 3.2, é uma soma ponderada dos dois critérios de desempenho, definidos por (3.11) e (3.12):

$$f(PID) = w_1 g_1(PID) + w_2 g_2(PID) \quad (3.13)$$

sendo:

$$PID = [K_c \ T_i \ T_d],$$

$$g_1 = ITAE,$$

$$g_2 = ITAY,$$

w_1 e w_2 - ponderações que permitem definir a importância de cada critério de desempenho (3.11) e (3.12), na avaliação da função objectivo (3.13).

A Fig. 3.2 apresenta nestas circunstâncias de teste 2 blocos do tipo "RN Critério j", referentes aos dois critérios de desempenho (3.11) e (3.12). Os capítulos 4 e 5 debruçar-se-ão mais em detalhe sobre a aprendizagem de redes neuronais referentes a este tipo de blocos.

Nesta fase de teste inicial da arquitectura de auto-sintonia, apenas se resumem na Tab. 3.1 as medidas de desempenho obtidas na aprendizagem de redes neuronais do tipo *MLP* com topologia [3 9 5 1], para ambos os critérios de desempenho (*ITAE* e *ITAY*). Os valores apresentados nessa tabela referem-se à iteração que conduz ao melhor conjunto de pesos¹.

¹ O melhor conjunto de pesos da rede neuronal foi considerado como sendo aquele que conduz ao menor valor de C_1 , calculado para o conjunto de teste.

Para essa iteração, registam-se os critérios de desempenho de aprendizagem das redes neurais (C_1 , C_2 e C_3), dados por:

C_1 – razão entre a norma do vector de erro e a norma do vector de saídas desejadas;

C_2 – média das percentagens de erro;

C_3 – percentagem máxima de erro.

		$C_1(\%)$	$C_2(\%)$	$C_3(\%)$
<i>ITAE</i>	Treino	1.1	1.8	26.4
	Teste	2.2	2.7	47.7
<i>ITAY</i>	Treino	1.4	2.9	51.3
	Teste	3.2	5.0	72.0

Tab. 3.1 – Desempenho da aprendizagem de redes neurais do tipo *MLP*.

Após a avaliação da função objectivo para a população inicial ocorre o funcionamento do algoritmo genético [123] no qual têm lugar as seguintes operações:

- Avaliação da aptidão e seriação,
- Selecção,
- Cruzamento ou recombinação,
- Mutação,
- Reinserção.

Os teste realizados utilizaram os parâmetros que se apresentam na Tab. 3.2, podendo-se agrupar em 3 pontos fundamentais:

1. Decisão sobre o tipo de população inicial a utilizar (determinística/aleatória).
2. Verificação do comportamento do sistema de controlo face à variação das ponderações w_1 e w_2 na função objectivo (3.13).
3. Comparação de diferentes métodos de sintonia.

• N° de populações	1
• <i>Gap</i> de geração	1
• Taxa de cruzamento	1
• Total de gerações	60
• Taxa de inserção dos descendentes	0.9
• N° de indivíduos	20
• N° de variáveis (<i>nvar</i>)	3
• Taxa de mutação	$0.7/nvar$

Tab. 3.2 – Parâmetros do algoritmo genético.

3.4.2 População inicial

A população inicial de n indivíduos encontra-se estruturada numa matriz real $n \times 3$, sendo cada indivíduo constituído pelos 3 valores dos parâmetros *PID*: k_c , t_i e t_d .

Foram utilizados dois tipos de populações iniciais:

- Determinística:** a população inicial é a matriz *PID* usada como conjunto de treino durante a aprendizagem das redes neuronais. Esta matriz contém 289 indivíduos, entre os quais, o *PID* obtido pela regra de Ziegler & Nichols (ver secção 1.4.1.1) e valores próximos do minimizante do *ITAE*, obtidos pelo método dos gradientes.
- Aleatória:** é gerada uma matriz real com um número mínimo de indivíduos. Após algumas experiências, concluiu-se que uma população com um número de indivíduos superior a 20 não conduz a melhoria significativa dos resultados, elevando consideravelmente o peso computacional. Os limites superior e inferior para os parâmetros *PID* que integram a população coincidem, respectivamente, com o máximo e o mínimo do parâmetro correspondente, dentro do conjunto de treino utilizado na aprendizagem das redes neuronais.

Com vista à decisão quanto ao tipo de população inicial a utilizar, foram desencadeados processos de optimização utilizando-se alternativamente os dois tipos de populações iniciais, determinística e aleatória, com as características apontadas anteriormente.

Para ambos os tipos de populações, foi procurado o número mínimo de gerações para o qual o processo evolutivo se considere concluído.

3.4.3 Estrutura do procedimento de optimização

Sem perda de generalidade, nesta fase de teste a função objectivo (3.13) vai-se resumir ao *ITAY*.

Durante a execução do algoritmo genético foram feitos 2 tipos de registos: parciais (Fig. 3.4 e Fig. 3.6) e finais (Fig. 3.5, e Fig. 3.7).

Os registos parciais são produzidos no decorrer do processo cíclico e permitem-nos avaliar a evolução do mesmo. Dentro deste tipo de registos, existem quatro representações a assinalar:

- Média e mínimo dos valores objectivo: para cada geração, considerando a totalidade da população, é representado o valor mínimo e o valor médio da função objectivo.
- Melhor indivíduo: para cada geração é representado o valor das 3 variáveis que compõem o indivíduo que conduziu ao mais baixo valor da função objectivo da respectiva geração. Com esta representação gráfica, poder-se-ão observar, em função da geração, 3 curvas que representam a evolução dos 3 parâmetros k_c , t_i e t_d minimizantes.
- Todos os indivíduos: na geração em que é feita a actualização do gráfico, para cada indivíduo é representado o valor dos seus 3 parâmetros k_c , t_i e t_d , obtêm-se tantas linhas quantos os indivíduos existentes.
- Todas as funções objectivo: na geração em que é feita a actualização do gráfico são representados os valores da função objectivo para todos os indivíduos.

Os pontos que se seguem ocorrem no final do processo evolutivo, resumindo-o.

- Evolução do óptimo: mostra como o mínimo da função objectivo variou ao longo das gerações. É determinada a eficiência do processo evolutivo, ou seja, a razão entre o

mínimo da primeira geração e a solução do problema, que é considerada como sendo o mínimo absoluto tendo em conta todas as gerações.

- Todos os indivíduos nas últimas n gerações: são consideradas as últimas 5 gerações, para as quais são representados os valores das 3 variáveis que constituem cada melhor indivíduo; é adicionalmente assinalado o valor *PID* minimizante. Em ambas as figuras (Fig. 3.5 e Fig. 3.7), verifica-se que nas últimas 5 gerações os melhores indivíduos respectivos já não variam, traduzindo-se este facto pela sobreposição dos respectivos gráficos.

Na Fig. 3.4 e Fig. 3.5 apresentam-se as evoluções parciais e os registos finais referentes às optimizações efectuadas com população determinística. Os resultados análogos conseguidos com populações iniciais aleatórias, apresentam-se na Fig. 3.6 e na Fig. 3.7.

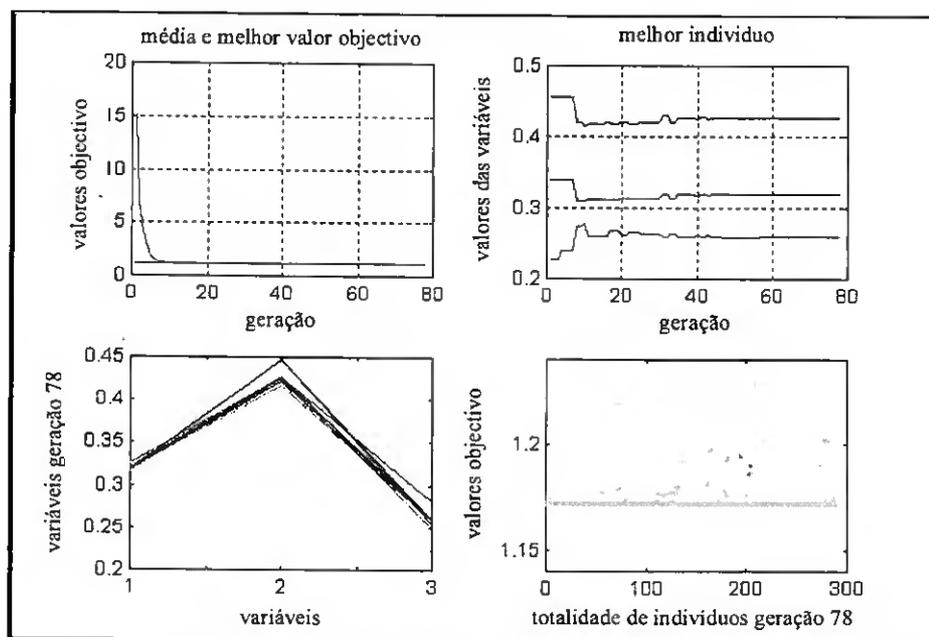


Fig. 3.4 – Evolução parcial da optimização, com população inicial determinística.

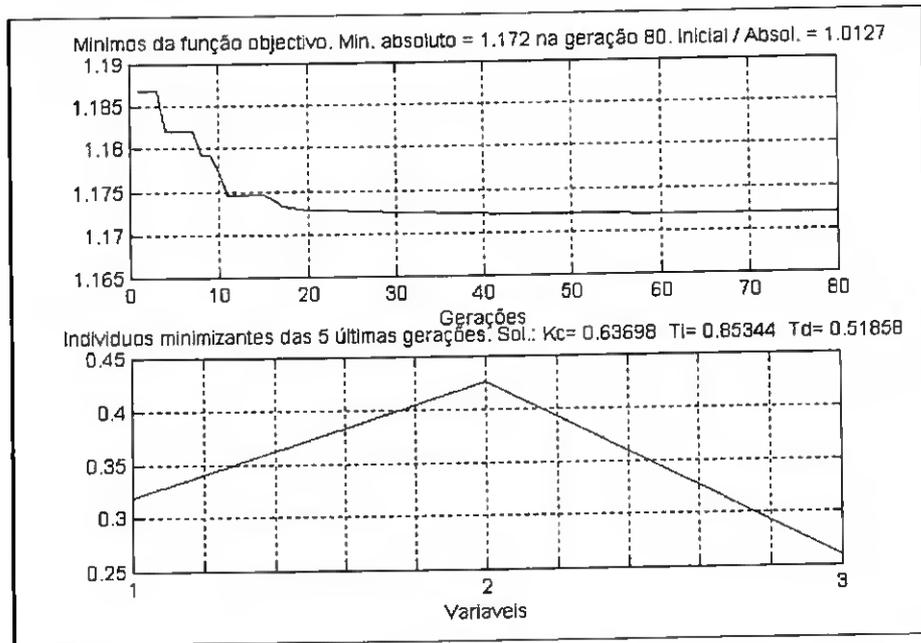


Fig. 3.5 – Registo final da optimização, população inicial determinística: 289 indivíduos.

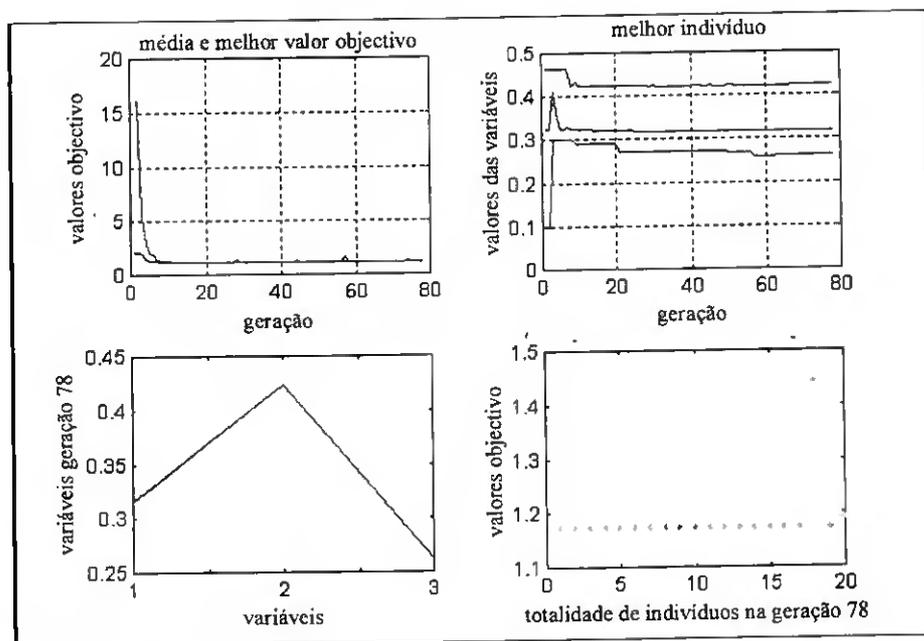


Fig. 3.6 – Evolução parcial da optimização, com população inicial aleatória.

A avaliação do processo evolutivo pode ser feita a partir duma análise dos percentis, segundo a qual se poderá deduzir, para um dado número de gerações m , qual a probabilidade do mínimo da função objectivo ser menor ou igual a determinado valor x . Para se levar a cabo esta análise, fixou-se o tipo de população inicial, o número de indivíduos (20 indivíduos) e o número de gerações necessárias para que o processo evolutivo se considere concluído (80 gerações). Utilizando estes parâmetros foram feitas

25 inicializações com os correspondentes processos evolutivos. Para cada inicialização, os mínimos da função objectivo ao longo das 80 gerações registaram-se em coluna na matriz

$$P(80 \times 25). \text{ Sendo } P = \begin{bmatrix} obj_1 & \dots & obj_{25} \end{bmatrix}, P(i, j), i \in [1 \ 80] \ j \in [1 \ 25] \text{ designa o valor}$$

da função objectivo que ocorreu na geração i da inicialização j .

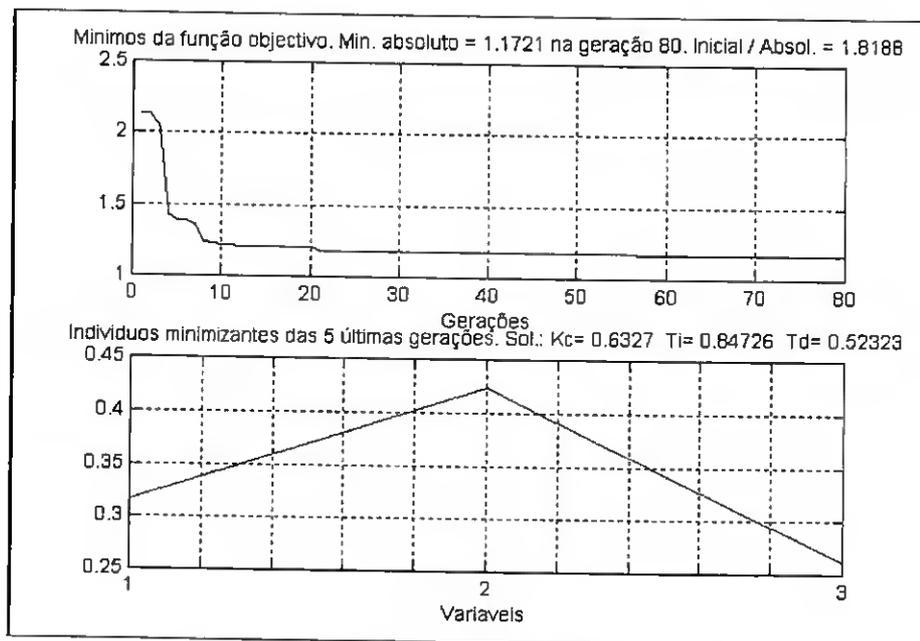


Fig. 3.7 – Registo final da optimização, população inicial aleatória: 20 indivíduos.

Cada linha da matriz P foi ordenada por ordem decrescente, originando-se a matriz P_o . Desta forma, a 1ª coluna define uma curva C_1 que representa a evolução dos valores mais altos dos mínimos da função objectivo ao longo das 80 gerações. De forma análoga, a última coluna define uma curva C_{25} , representativa dos valores mais baixos dos mínimos da função objectivo ao longo das 80 gerações. Assim, concluí-se que os pontos resultantes da representação gráfica dos mínimos da função objectivo em função da geração para a totalidade do processo evolutivo se situam entre as curvas C_1 e C_{25} , tal como se pode observar na Fig. 3.8.

Pretendendo-se uma aproximação da probabilidade do mínimo da função objectivo pertencer a um dado intervalo de percentagens, em cada geração, deve-se particionar na vertical a matriz P em tantas sub-matrizes de iguais dimensões quantos os intervalos de percentagens pretendidos. Assim, considerando os seguintes intervalos de percentagens:

$[100 \ 75[$, $[75 \ 50[$, $[50 \ 25[$, $[25 \ 0[$ o particionamento correspondente é esquematizado na expressão (3.14).

$$P_o = \begin{bmatrix} \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ c_1 & \cdots & c_7 & \cdots & c_{13} & \cdots & c_{19} & \cdots & c_{25} \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \end{bmatrix} \quad (3.14)$$

Na Fig. 3.8, para além das duas envolventes C_1 e C_{25} , destacam-se as restantes curvas C_7 , C_{13} e C_{19} , delimitando superiormente as regiões para as quais a probabilidade de se encontrar o mínimo da função objectivo, ao fim de determinado número de gerações, é de 75%, 50% e 25%, respectivamente.

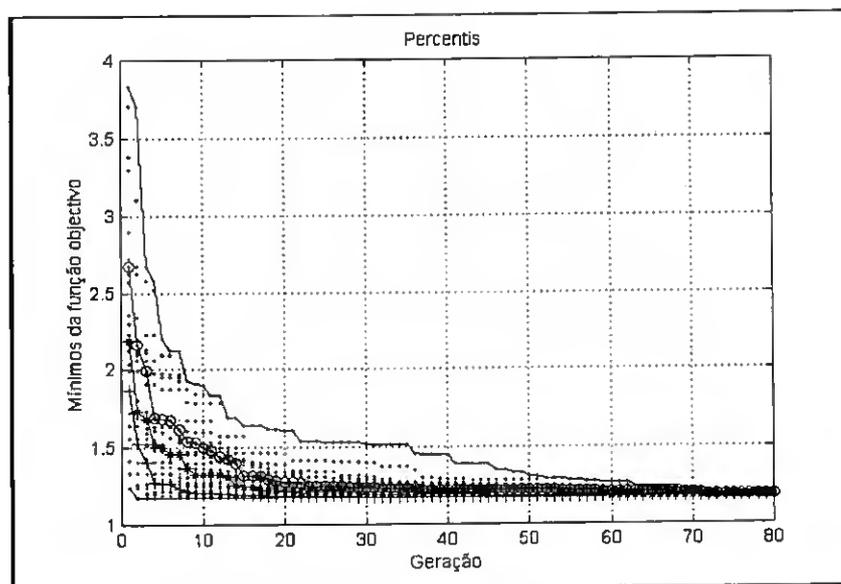


Fig. 3.8 – Representação dos percentis.

A observação da Fig. 3.8 permite concluir quanto à adequação da escolha de 80 gerações para o processo evolutivo. De facto, verifica-se que, para valores do número de gerações próximos de 80, as duas envolventes tendem a coincidir, mostrando assim, que a probabilidade de ocorrer um mínimo da função objectivo é independente do número de inicializações que se execute.

3.4.4 Comportamento do sistema face às variações das ponderações

Nesta secção apresenta-se a sintonia do controlador para as 2 configurações definidas a partir da Fig. 3.1: seguimento da entrada e rejeição de perturbações à saída.

Para cada uma destas configurações, consideram-se 3 conjuntos de ponderação diferentes para a função objectivo (3.13):

a) $w_1 = 1 \wedge w_2 = 0$

b) $w_1 = 1 \wedge w_2 = 1$

c) $w_1 = 0 \wedge w_2 = 1$,

As situações a) e c) correspondem à optimização do *ITAE* e do *ITAY* respectivamente; na situação b) está a ser optimizada a função objectivo com iguais ponderações para ambos os critérios de desempenho (*ITAE* e *ITAY*).

Para a configuração correspondente ao seguimento da entrada foram feitas 3 optimizações em que a função objectivo assumiu sucessivamente as 3 ponderações a), b) e c). Os 3 vectores *PID* resultantes das respectivas optimizações conduziram às 3 respostas sobrepostas apresentadas na parte superior da Fig. 3.9, cuja legenda se regista na Tab. 3.3. A parte inferior da mesma figura, representa a evolução do *ITAE* em função do tempo.

Legenda	
—	$w_1 = 1 \wedge w_2 = 0$
-·-·-·-·-·-·-·-·-·-	$w_1 = 1 \wedge w_2 = 1$
—	$w_1 = 0 \wedge w_2 = 1$

Tab. 3.3 – Legenda para diferentes configurações de ponderação (Fig. 3.9 e Fig. 3.10).

Foi feito um estudo semelhante para a configuração que pretende analisar a rejeição de perturbações à saída. Da mesma forma, a função objectivo foi tomada com os 3 conjuntos de ponderação descritos nos pontos a), b) e c). A sobreposição das respostas $y(t)$ para as 3 situações e as correspondentes evoluções no tempo do *ITAY* apresentam-se

respectivamente na parte superior e inferior da Fig. 3.10; esta figura utiliza a legenda apresentada na Tab. 3.3.

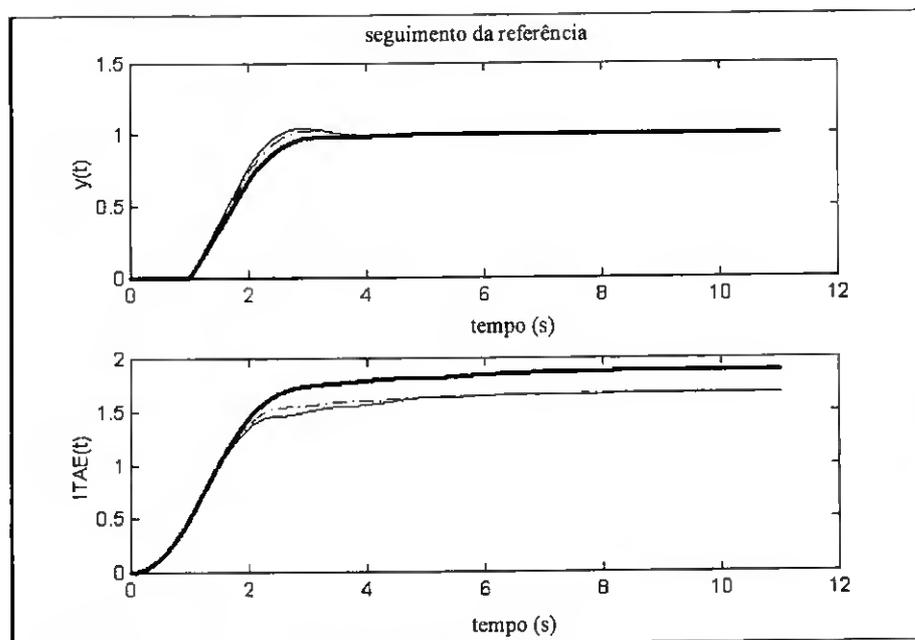


Fig. 3.9 – Sintonia genética: seguimento da referência.

A observação da Fig. 3.9 permite verificar que a optimização da função objectivo em que $w_1 = 1 \wedge w_2 = 0$ é a que conduz a um valor mais baixo do *ITAE*, embora este valor esteja muito próximo da situação em que se utiliza iguais ponderações para *ITAE* e *ITAY*. De forma análoga, quando se considera a configuração para rejeição de perturbações à saída, a sintonia que conduz a um mais baixo valor de *ITAY* é a que utiliza a função objectivo com $w_1 = 0 \wedge w_2 = 1$, tal como se pode observar na Fig. 3.10.

3.4.5 Diferentes tipos de sintonia

Os últimos testes desta fase pretendem pôr em confronto 3 métodos distintos de sintonia:

1. *PID* optimizado para o *ITAE* mínimo utilizando-se informação sobre o gradiente, de acordo com o método *quasi-Newton*, com uso da fórmula *BFGS* para cálculo das aproximações da matriz Hessiana. O procedimento de optimização recorreu à função

fminunc disponível na *toolbox* de optimização [132] do MATLAB. De acordo com a implementação utilizada os valores de *ITAE* são obtidos por simulação dinâmica².

2. *PID* optimizado para o *ITAE* mínimo utilizando-se algoritmos genéticos; os valores de *ITAE* foram obtidos utilizando-se redes neuronais treinadas.
3. Regras de sintonia de Ziegler & Nichols.

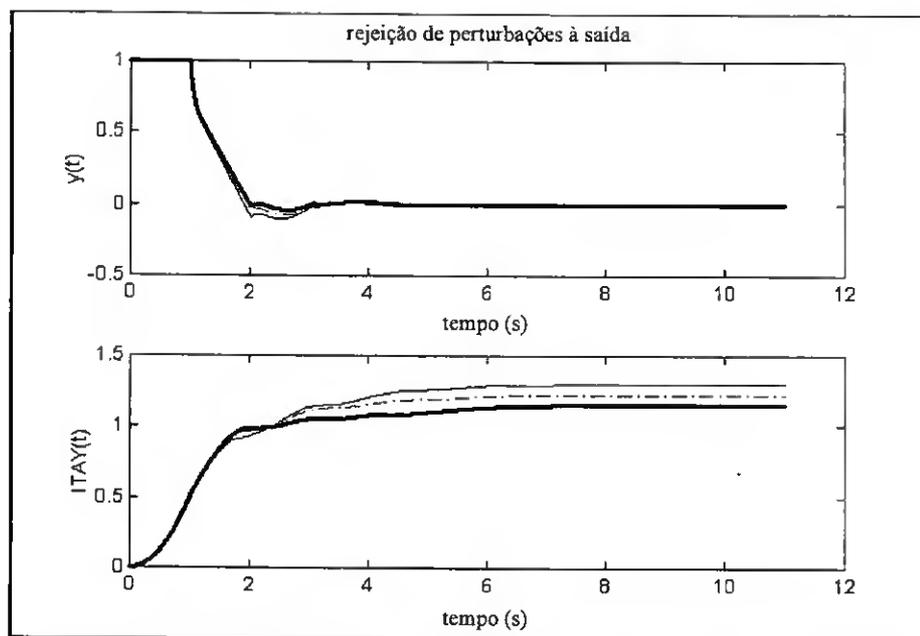


Fig. 3.10 – Sintonia genética: rejeição de perturbações à saída.

Tal como aconteceu na secção 3.4.4, o sistema controlado foi considerado nas duas configurações em estudo: seguimento da entrada e rejeição de perturbações à saída. Na Fig. 3.11, para a configuração de seguimento da entrada, e na Fig. 3.12, para a configuração de rejeição de perturbações à saída, sobrepõem-se as 3 respostas $y(t)$ referentes às regras de sintonia mencionadas. Para ambas as situações apresentam-se nas partes inferiores das figuras as evoluções temporais do *ITAE* e do *ITAY*, respectivamente, sendo a legenda apresentada na Tab. 3.4. A observação das figuras permite verificar que a optimização do *ITAE* utilizando algoritmos genéticos é aquela que conduz à sintonia que leva aos mais baixos valores tanto de *ITAE* como de *ITAY*, para a montagem de seguimento da entrada, ou rejeição de perturbações à saída, respectivamente.

² Não foram utilizadas redes neuronais.

Legenda	
—	minimização de <i>ITAE</i> usando-se algoritmos genéticos
- - - - -	minimização de <i>ITAE</i> pelo método dos gradientes
—	regras de Ziegler & Nichols

Tab. 3.4 – Legenda para diferentes tipos de sintonia (Fig. 3.11 e Fig. 3.12).

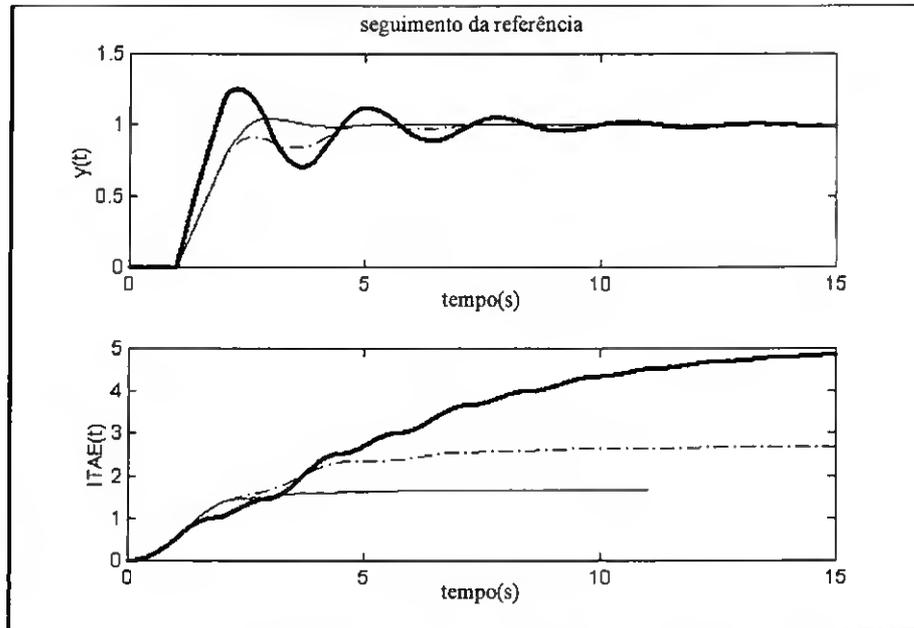


Fig. 3.11 – Seguimento da referência: 3 sintonias distintas.

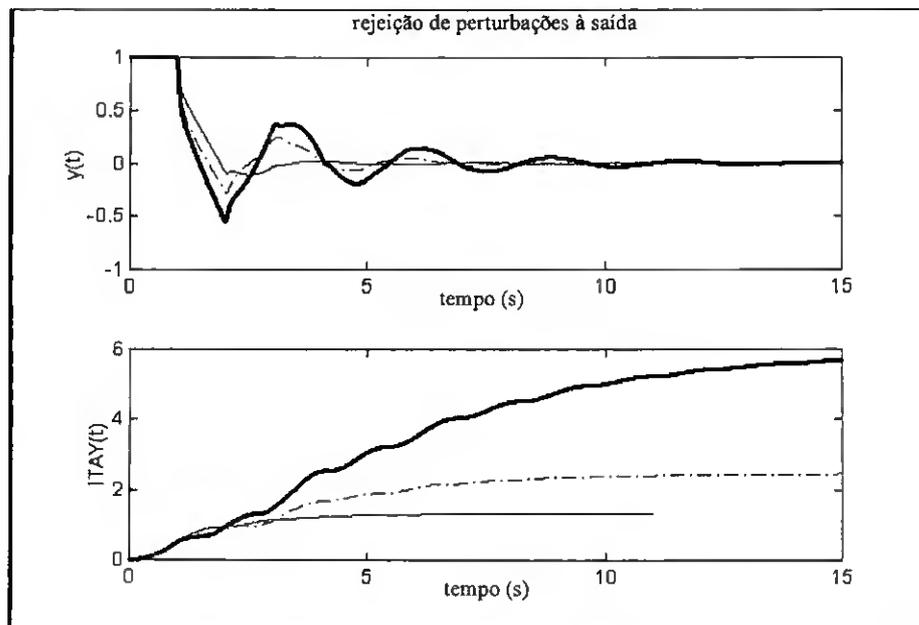


Fig. 3.12 – Rejeição de perturbações à saída: 3 sintonias distintas.

3.5 CONCLUSÕES

Neste capítulo foi feita a apresentação duma arquitectura adaptativa para auto-sintonia do controlador *PID*.

A arquitectura proposta inclui, para além do sistema controlado, um sistema de aquisição de sinal para identificação do processo e sintonia do controlador. O sistema de sintonia do controlador apresentado implica um processo de optimização que fica a cargo dos algoritmos genéticos. Os testes iniciais executados neste capítulo visaram o funcionamento dos algoritmos genéticos. Foram testadas optimizações de diferentes funções objectivo, que resultaram da atribuição de diferentes ponderações aos 2 critérios de desempenho considerados, *ITAE* e *ITAY*. Nesta fase inicial de teste foram utilizadas e comparadas várias estratégias de sintonia, tais como, optimização utilizando os algoritmos genéticos, optimização utilizando métodos de gradientes e por último, as regras de Ziegler e Nichols. Estes testes permitiram concluir que a optimização, levada a cabo com os algoritmos genéticos, conduz a resultados sensivelmente melhores do que aqueles que se conseguem com os métodos de gradientes.

A utilização dos algoritmos genéticos em que a função objectivo é avaliada utilizando redes neuronais treinadas, consome muito menos tempo do que o necessário quando se utilizam métodos de gradientes, com a função objectivo calculada recorrendo à simulação dinâmica do sistema. Este aspecto é fundamental quando se tem em vista a implementação em tempo real.

A arquitectura proposta visa a acomodação de processos variantes no tempo. Na continuação ter-se-á em conta este objectivo, utilizando-se para tal, as medidas de identificação apresentadas neste capítulo. A aprendizagem de redes neuronais deverá de igual forma ter em conta esse facto. Assim, os próximos capítulos debruçar-se-ão sobre a aprendizagem de redes neuronais com o objectivo de se obter modelos tão exactos quanto possível, que simultaneamente apresentem boa capacidade de generalização e possibilidade de adaptação.

4. SELECÇÃO DO TIPO DE REDE NEURONAL

4.1 INTRODUÇÃO

No capítulo anterior introduziu-se uma arquitectura de controlo que possibilita a sintonia do controlador *PID* através da optimização em linha dos critérios em estudo. Para que tal sintonia fosse viável em tempo real, verificou-se que seria necessária a utilização de modelos para os referidos critérios. Esses modelos para sistemas variantes no tempo, para além de necessitarem de ser tão exactos quanto possível para um espaço de trabalho previamente conhecido, devem possibilitar adaptação a variações ocorridas em relação ao espaço inicial de trabalho. Conforme referido anteriormente, nesta fase, os modelos utilizados são 2 redes neuronais treinadas, com 3 entradas $[k_c \ t_i \ t_d]$ e uma saída, respectivamente *ITAE* e *ITAY*, normalizadas de acordo com (4.1) e cujo factor das constantes de tempo T_T se define pela expressão (3.4).

$$k_c' = \frac{k_c}{T_T}, \quad t_i' = \frac{t_i}{T_T}, \quad t_d' = \frac{t_d}{T_T}, \quad ITAE' = \frac{T_T^2}{ITAE}, \quad ITAY' = \frac{T_T^2}{ITAY} \quad (4.1)$$

Estas normalizações foram feitas para que se atenda apenas aos valores relativos das constantes de tempo dos processos utilizados. As inversões efectuadas para o *ITAE* e *ITAY* devem-se ao facto dos baixos valores pretendidos para estes critérios, uma vez mapeados nos seus inversos, conduzirem a uma maior precisão absoluta.

Dada a variedade de tipos de redes neuronais disponíveis, o problema da modelação neuronal foi dividido em 2 sub-problemas:

- 1º Escolha do tipo de redes neuronais que possibilite exactidão, generalização e adaptação.
- 2º Utilização do tipo de redes neuronais determinado no ponto anterior, para construção dum modelo neuronal adequado ao espaço de trabalho em estudo.

O presente capítulo vai-se cingir ao 1º ponto; o capítulo seguinte ocupar-se-á com a matéria respeitante ao 2º ponto. Nesta fase de selecção do tipo de rede neuronal mais adequado, o processo utilizado é invariante no tempo. Assim, recorrendo-se ao processo da secção 3.4, $G(s) = \frac{e^{-s}}{s+1}$ e aos critérios de desempenho nessa altura referidos, *ITAE* e *ITAY*, decidir-se-á neste capítulo qual o tipo de rede neuronal a utilizar. Desta forma, na secção 4.2 explica-se como foram formados os conjuntos de aprendizagem e qual a normalização utilizada. Nas secções 4.3, 4.4 e 4.5 serão construídos modelos neuronais utilizando-se 3 tipos de redes neuronais: *MLP*, *RBF* e *B-Spline*, respectivamente.

Dentro da secção correspondente a cada tipo de rede neuronal, será primeiramente abordado o problema da aprendizagem em diferido e por último o problema da adaptação em linha. Segue-se a secção 4.6 onde serão confrontados em simultâneo os resultados provenientes dos 3 tipos de redes neuronais, que irão servir de base à decisão sobre o tipo de rede neuronal a utilizar nos capítulos seguintes. Este capítulo termina com a secção 4.7, onde se farão considerações sobre os 3 tipos de redes neuronais utilizados; será seleccionado um dos tipos e introduzida a matéria a tratar no capítulo seguinte.

4.2 CONJUNTOS DE APRENDIZAGEM

Pretende-se obter nesta fase exemplos de vectores *PID* baseados nos valores de Ziegler e Nichols. Assim, recorrendo-se ao algoritmo da Tab. 4.1 obtém-se a matriz de padrões *PID* nas condições pretendidas. Os correspondentes vectores *ITAE* e *ITAY* obtêm-se por simulação do sistema apresentado na Fig. 3.1, tendo em conta as expressões (3.11) e (3.12).

Dependendo do valor da percentagem p utilizada no algoritmo poderão surgir padrões que conduzem à instabilidade ou a respostas com tempos de estabelecimento muito elevados; esses padrões são excluídos dos conjuntos de aprendizagem. Nestas circunstâncias [6], fazendo, por exemplo, $p=50\%$ e $n=10$ obtém-se um conjunto base V tal que $\#V=577$. A partir desta base de padrões foram construídos os conjuntos de aprendizagem, tanto para o treino em diferido como para a adaptação em linha; o número de padrões de cada conjunto encontra-se registado na Tab. 4.2. Relativamente a esta tabela deve ser notado que, para a adaptação em linha, o conjunto de treino é utilizado como um conjunto de adaptação; ou seja, cada padrão é tomado individualmente para que seja avaliada a correspondente correcção aos pesos da rede neuronal.

- Determinar um vector PID inicial pelas regras de Ziegler e Nichols,
- Para cada componente do vector inicial, aplicar uma perturbação de $\pm p\%$, definindo-se um limite superior e um limite inferior,
- Discretizar uniformemente cada componente nesse intervalo, obtendo-se n valores distintos em torno do valor de Ziegler e Nichols.
- Cruzar as 3 discretizações correspondentes às 3 componentes, obtendo-se a matriz de $m = n^3$ vectores PID linha.

Tab. 4.1- Algoritmo para construção da matriz PID .

	em diferido	em linha
treino	$\#(R) = 381$	$\#(A) = 179$
teste	$\#(S) = 196$	$\#(C) = 398$

Tab. 4.2- Conjuntos de aprendizagem.

Para os conjuntos da Tab. 4.2 verificam-se as expressões dadas por (4.2).

$$\begin{aligned}
 R \cup S &= V, & R \cap S &= \emptyset \\
 A \cup C &= V, & A \cap C &= \emptyset
 \end{aligned}
 \tag{4.2}$$

Os conjuntos R e S são extraídos aleatoriamente de V . O domínio para extracção dos padrões para adaptação em linha (conjunto A) é obtido de acordo com a expressão (4.3).

$$A = \left\{ PID : \left(1 - \frac{r}{100} \right) x_i^* < x_i < \left(1 + \frac{r}{100} \right) x_i^*, i = 1, 2, 3 \right\} \quad (4.3),$$

Na expressão (4.3) $x_1 = k_c'$, $x_2 = t_i'$ e $x_3 = t_d'$; x_i^* e r representam respectivamente o valor central e a percentagem de desvio para cada componente i .

A adaptação em linha das redes neuronais ocorre para pequenas regiões do espaço de aprendizagem, devendo então, o valor para o raio r da vizinhança ser baixo.

De acordo com o algoritmo de geração dos vectores *PID* apresentado na Tab. 4.1 a discretização de cada componente é uniforme; contudo, dado que alguns vectores produzidos pelo algoritmo são rejeitados, dentro da base de vectores *PID* obtida, para cada componente a distribuição de pontos é não uniforme.

Sem perda de generalidade, consideremos a componente k_c do vector *PID*. Verifica-se que, para valores não muito altos de r , o número de vectores seleccionados depende de onde se centra a vizinhança para selecção de k_c . Os vectores seleccionados são ainda mais restringidos, atendendo aos limites impostos adicionalmente pelas outras duas componentes, t_i e t_d .

Pode-se finalmente concluir que a totalidade de vectores seleccionados depende do valor de r e dos centros das vizinhanças escolhidos para as três componentes, x_i^* .

No algoritmo implementado para escolha dos padrões de adaptação em linha, de acordo com a expressão (4.3), o posicionamento dos valores centrais x_i^* fez-se empiricamente. Assim, para cada uma das componentes do vector *PID* foram percorridos alguns valores de centros, dentro da gama de variação da respectiva componente. Utilizando-se valores não muito elevados para o raio r , determinaram-se, para cada concretização dos 3 centros, quantos vectores *PID* seriam seleccionados. O conjunto de adaptação a que se refere a coluna da direita da Tab. 4.2 corresponde à concretização dos 3 centros, que conduziu ao maior número de vectores seleccionados, tendo-se para tal escolhido $r=30\%$.

Na figura Fig. 4.1 apresenta-se no sistema de eixos (k_c, t_i, t_d) os pontos que constituem o conjunto de adaptação, sobrepostos com a totalidade de vectores *PID* considerados.

4.2.1 Normalização dos padrões de entrada

Os padrões de entrada das redes neuronais são normalizados de forma a se situarem entre os limites inferior e superior previamente especificados.

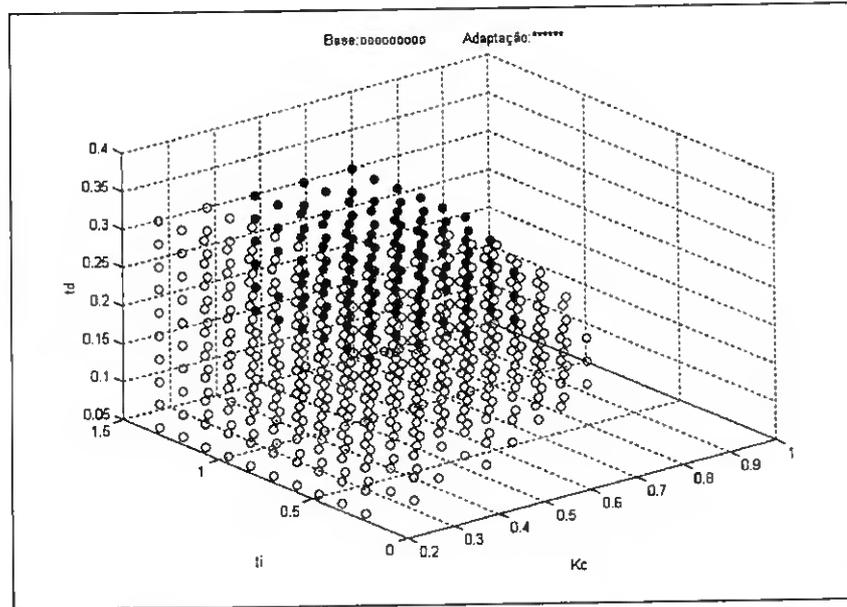


Fig. 4.1 – Sobreposição do conjunto de adaptação no espaço de trabalho.

Para se mostrar como é feita a normalização, considere-se genericamente um conjunto B , cujos elementos se pretendem colocar entre os limites l_{inf} e l_{sup} , $l_{inf} < l_{sup}$. O conjunto normalizado B_n resulta de B por aplicação da transformação linear (4.4) aos seus elementos.

$$B_n \equiv R(B) = pB + q; \quad p, q \in \mathfrak{R} \tag{4.4}$$

Para que os elementos de B_n se situem entre os limites especificados, as constantes p e q da transformação linear serão dadas pelas expressões (4.5).

$$p = \frac{l_{sup} - l_{inf}}{\max(B) - \min(B)}, \quad q = -\frac{l_{sup} - l_{inf}}{\max(B) - \min(B)} \min(B) + l_{inf} \tag{4.5}$$

A transformação linear inversa que permite obter B , a partir de B_n , é dada a partir da expressão (4.6).

$$B \equiv Q(B_n) = uB_n + v; \quad u, v \in \mathfrak{R} \tag{4.6}$$

em que se define:

$$u = \frac{\max(B) - \min(B)}{l_{\text{sup}} - l_{\text{inf}}}, \quad v = \frac{l_{\text{sup}} \cdot \min(B) - l_{\text{inf}} \cdot \max(B)}{l_{\text{sup}} - l_{\text{inf}}} \quad (4.7).$$

Os procedimentos que levam à normalização do conjunto base, com posterior extração dos conjuntos de treino e teste, estão esquematizados na Fig. 4.2.

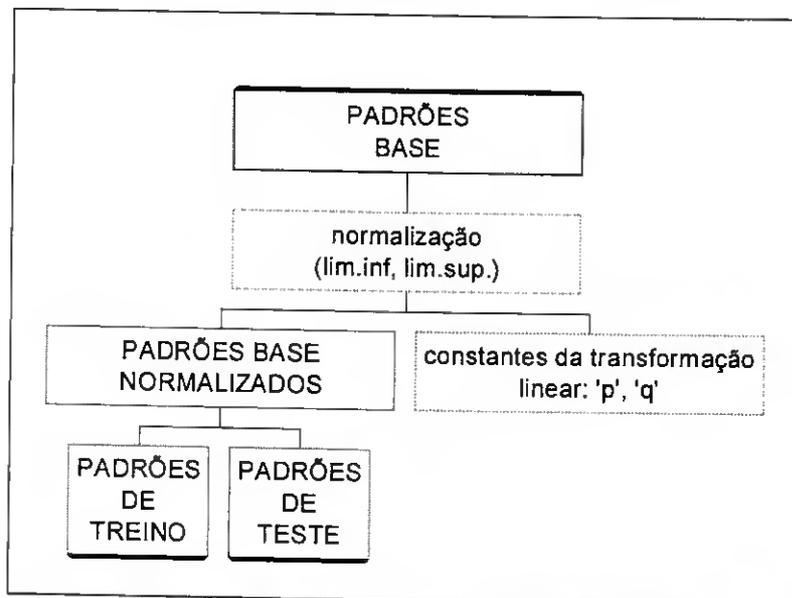


Fig. 4.2 – Normalização do conjunto base de padrões.

Nota-se que, de acordo com este processo, o conjunto de treino poderá não atingir os limites impostos para a normalização, ficando incluído no intervalo definido por estes.

Salvo indicação adicional, o desempenho dos processos de aprendizagem expostos ao longo do capítulo é avaliado pelos critérios (4.8), considerados tanto para o conjunto de treino como para o conjunto de teste.

$$MSE = \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}, \quad MRE(\%) = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right|, \quad SSRE = \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{\hat{y}_i} \right)^2 \quad (4.8)$$

4.3 MODELOS NEURONAIS UTILIZANDO MLPs

A fórmula geral adoptada [3 M N 1] permite obter 100 topologias diferentes [130] para as redes neuronais que resultam da concretização de cada parâmetro M e N no intervalo de inteiros de 1 a 10.

Tendo em conta os indicadores C_1 , C_2^\dagger e C_3 , avaliados em cada iteração para os conjuntos de treino e teste, e ambos os mapeamentos $PID \rightarrow ITAE$ e $PID \rightarrow ITAY$, pretende-se determinar a topologia mínima que conduz ao melhor desempenho.

$$C_1 = 100 \frac{\|y - y^*\|}{\|y^*\|}, \quad C_2 = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - y_i^*}{y_i^*} \right|, \quad C_3 = \frac{100}{n} \max_{i=1..n} \left\{ \left| \frac{y_i - y_i^*}{y_i^*} \right| \right\} \quad (4.9),$$

O algoritmo de treino utilizado [133], foi implementado em MATLAB [134], e baseia-se no algoritmo de Levenberg Marquadt [135] (ver secção 2.2.2.2).

A rede neuronal procurada deverá ter uma boa capacidade de generalização. Desta forma, o processo de aprendizagem faz-se com vista à obtenção dum mínimo de C_1 para o conjunto de teste. Com base neste princípio, estabeleceu-se o seguinte critério de paragem:

Considera-se terminado um processo de aprendizagem quando decorrem n iterações, sem que o último mínimo de C_1 , no conjunto de teste, seja transposto por um valor inferior.

Feitos alguns ensaios, concluiu-se que os valores de desempenho já não melhoram significativamente 15 iterações após o último mínimo de C_1 para o conjunto de teste; desta forma, foi escolhido $n=15$ no critério de paragem.

Para cada um dos mapeamentos em estudo tiveram lugar os 100 processos de aprendizagem correspondendo a todas as concretizações possíveis para os valores de M e N no intervalo considerado. Na Tab. 4.3 registam-se as topologias que levaram aos 5 menores valores dos critérios de aprendizagem em estudo (C_1 , C_2 e C_3) para ambos os mapeamentos, avaliados para o conjunto de teste. Na primeira linha apresentam-se os intervalos nos quais variam os valores do respectivo critério ao longo dos 100 ensaios.

[†] Este indicador coincide com MRE(%) de acordo com (4.8); a nova nomenclatura é adoptada por questões de uniformização.

Saliente-se que todos os valores apresentados para tais critérios se referem à iteração em que se observou o mínimo de C_1 para o conjunto de teste¹.

	Topologia	C_1 % [2.54 56.3]	Topologia	C_2 % [3.57 133.5]	Topologia	C_3 % [43.2 1560.1]
<i>PID</i> ↓ <i>ITAE</i>	[3 9 10 1]	2.54	[3 7 8 1]	3.57	[3 9 5 1]	43.2
	[3 10 10 1]	2.61	[3 9 8 1]	3.73	[3 6 7 1]	45.2
	[3 9 4 1]	2.66	[3 9 5 1]	3.73	[3 9 4 1]	46.8
	[3 9 5 1]	2.70	[3 9 10 1]	3.90	[3 9 10 1]	47.5
	[3 9 8 1]	2.83	[3 9 4 1]	4.01	[3 9 8 1]	47.5
<i>PID</i> ↓ <i>ITAY</i>	[3 7 9 1]	2.89	[3 10 8 1]	5.42	[3 7 6 1]	67.2
	[3 9 5 1]	3.23	[3 7 9 1]	5.44	[3 9 2 1]	73.0
	[3 6 10 1]	3.56	[3 9 5 1]	5.53	[3 9 6 1]	76.6
	[3 7 6 1]	3.69	[3 7 6 1]	5.53	[3 10 8 1]	78.8
	[3 9 6 1]	3.72	[3 9 6 1]	5.89	[3 8 6 1]	83.9

Tab. 4.3- Desempenho da aprendizagem para selecção da topologia.

Pretende-se, para cada mapeamento, determinar a topologia mais adequada. Por observação da Tab. 4.3 conclui-se que, para critérios diferentes, as topologias que conduzem aos valores mínimos são diferentes. Para se ter uma ideia global do desempenho de determinada topologia para cada mapeamento, estabeleceram-se pesos para os diferentes critérios, consoante a importância atribuída a cada um deles; assim, a cada topologia associa-se uma pontuação dada por (4.10).

$$P_T = 4P_1 + 2P_2 + P_3 \quad (4.10)$$

Na expressão anterior P_1 , P_2 e P_3 representam a ordem de cada topologia dentro da ordenação crescente por, respectivamente, C_1 , C_2 e C_3 . A topologia que melhor serve ambos os mapeamentos será aquela cuja soma das duas pontuações é mais baixa. Para cada

¹ Em alguns casos verificou-se que após se atingir o mínimo de C_1 teste, C_2 ainda decrescia ligeiramente.

mapeamento apresentam-se as 5 melhores topologias e respectivas pontuações² na Tab. 4.4.

Topologia	P_T		
	$PID \rightarrow ITAE$	$PID \rightarrow ITAY$	Soma
[3 9 10 1]	16	119	135
[3 9 5 1]	23	49	72
[3 9 4 1]	25	214	239
[3 9 8 1]	29	310	339
[3 7 8 1]	34	215	249
[3 7 6 1]	312	25	337
[3 7 9 1]	133	26	159
[3 9 6 1]	183	33	216
[3 10 8 1]	56	34	90
[3 6 10 1]	221	36	257

Tab. 4.4 - Pontuações finais para ambos os mapeamentos e respectivas somas.

Expandindo a Tab. 4.4 para todas as topologias, pode-se verificar que a topologia que reúne menor soma de pontos é [3 9 5 1], sendo como tal, a mais adequada.

Selecionada esta topologia pretende-se de seguida obter o melhor conjunto de pesos para cada mapeamento. Assim, para cada mapeamento, efectuaram-se sucessivas inicializações aleatórias de pesos e correspondentes processos de aprendizagem. Para cada ensaio, o mínimo de C_I no conjunto de teste era comparado com o correspondente valor do ensaio anterior; caso fosse menor, o conjunto de pesos era guardado. Totalizou-se cerca de 200 inicializações para cada mapeamento. Verificou-se que os andamentos das aprendizagens eram muito semelhantes para os mapeamentos em estudo. Na Fig. 4.3 apresenta-se a aprendizagem que conduziu ao valor mais baixo do mínimo de C_I para o conjunto de teste para o mapeamento $PID \rightarrow ITAE$.

² A pontuação soma refere-se à soma das duas pontuações referentes aos dois mapeamentos em estudo.

Nas circunstâncias experimentadas verificou-se que o processo de aprendizagem ficava concluído num número de iterações que se situa entre as 50 e 100, para ambos os mapeamentos. De forma análoga à evolução que se apresenta na Fig. 4.3, os restantes indicadores de desempenho, para todos os processos de aprendizagem, estabilizam com um número de iterações que se situa no intervalo apresentado, provando que mais iterações não seriam necessárias e que o critério de paragem utilizado é adequado.

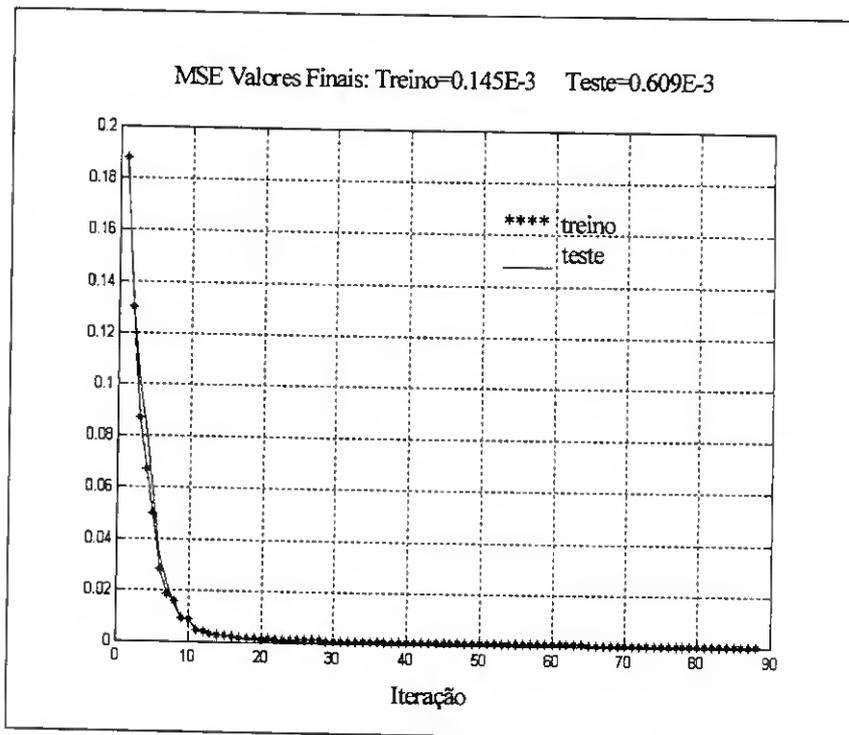


Fig. 4.3 – Aprendizagem de *MLP* em diferido para o mapeamento *PID* → *ITAE*.

O grau de generalização conseguido foi bom uma vez que a curva obtida para cada critério avaliado no conjunto de teste se aproxima da curva correspondente avaliada para o conjunto de treino. Este facto foi observado para a totalidade dos processos de aprendizagem, salientando-se o exemplo apresentado na Fig. 4.3.

4.3.1 Actualização de pesos padrão a padrão de *MLPs*

Neste capítulo de selecção do tipo de rede neuronal importa, para além de se testar qual o tipo de rede que conduz a resultados mais precisos e com melhor capacidade de generalização, verificar o comportamento de cada tipo de rede previamente treinada em diferido face à presença de novos padrões. Assim, a rede neuronal seleccionada deverá ser

adaptadas com os padrões obtidos em linha, procurando que se mantenha memorizada, tanto quanto possível, a aprendizagem realizada em diferido.

Sem perda de generalidade, a adaptação em linha incide apenas sobre o mapeamento $PID \rightarrow ITAE$. Após a realização da aprendizagem em diferido do referido mapeamento, é registado o modelo correspondente à iteração para a qual é mínimo o MSE teste (Fig. 4.3). Quando, para esse modelo, são utilizados os conjuntos de adaptação em linha (Tab. 4.2) obtêm-se os valores correspondentes ao estado inicial para o processo de adaptação (Tab. 4.5).

Iter.	$MSE (10^{-3})$		$MRE (\%)$	
	Treino	Teste	Treino	Teste
0	0.2	0.4	1.9	3.5

Tab. 4.5 - MSE e MRE iniciais para adaptação em linha de $MLPs$.

O comportamento do modelo face à adaptação em linha é deduzido a partir da análise da evolução das medidas de desempenho num horizonte de 50 iterações realizadas a partir do estado inicial que se apresenta na Tab. 4.5. Para cada critério de desempenho, define-se ganho como sendo a razão entre os respectivos valores inicial e final. Assim, ganhos superiores a 1 significam que houve melhoria durante o horizonte de adaptação, enquanto que ganhos inferiores a 1 significam que houve distorção da correspondente medida de desempenho.

Foram concluídos vários processos de adaptação para diferentes valores da taxa de aprendizagem α , utilizando-se o algoritmo *Backpropagation* aplicado padrão a padrão (ver secção 2.2.2.1). Os resultados correspondentes aos valores de α mais relevantes estão registados na Tab. 4.6.

A partir dos valores observados para as diferentes taxas de aprendizagem verifica-se que a situação para a qual existe melhoria no conjunto de treino, sem que haja distorção exagerada no conjunto de teste, ocorre para $\alpha = 0.1$, conforme assinalado na Tab. 4.6 e representado graficamente na Fig. 4.4.

α	MSE				MRE			
	TREINO		TESTE		TREINO		TESTE	
	Ganho	MIN. (10^{-3})	Ganho	MIN. (10^{-3})	Ganho	MIN. (%)	Ganho	MIN. (%)
0.01	1.10	0.2	0.82	0.4	1.00	1.9	0.89	3.5
0.1	1.20	0.2	0.29	0.4	0.99	1.8	0.46	3.5
0.5	1.26	0.2	0.14	0.4	0.90	1.9	0.27	3.5
1	1.28	0.2	0.11	0.4	0.91	1.9	0.22	3.5
1.5	1.29	0.2	0.10	0.4	0.91	1.9	0.20	3.5
2	1.28	0.2	0.09	0.4	0.90	1.9	0.19	3.5
2.5	1.09	0.2	0.08	0.4	0.85	1.9	0.08	3.5
5	0.05	0.2	0.05E-2	0.4	0.11	1.9	0.01	3.5

Tab. 4.6- Adaptação em linha para as MLPs.

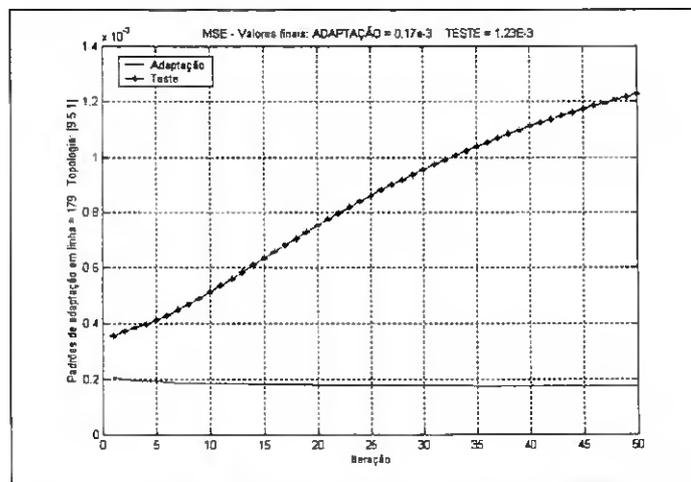


Fig. 4.4 – Adaptação em linha de MLP para $\alpha = 0.1$.

4.4 MODELOS NEURONAIS UTILIZANDO RBFs

Nesta secção vão ser realizadas aprendizagens de redes neuronais do tipo RFB com vista à obtenção de modelos. Tal como aconteceu na secção anterior, numa primeira fase a aprendizagem é feita em diferido; posteriormente, o modelo assim obtido é adaptado em linha.

De acordo com os fundamentos apresentados na secção 2.2.3, o perfil localizado das funções de activação do tipo Gaussiana utilizadas nestas redes neuronais, contrastando com o andamento mais distribuído das sigmóides utilizadas nas *MLPs*, faz com que se espere que a adaptação feita para as *RBFs* provoque menos distorção do que aquela que se verifica para as *MLPs*.

4.4.1 Plano de aprendizagem

O processo de aprendizagem levado a cabo nesta secção visa a implementação em diferido dos algoritmos adequados à determinação dos parâmetros das redes neuronais do tipo RBF, para o mapeamento $PID \rightarrow ITAE$. O conjunto de teste é utilizado simultaneamente para verificação da capacidade de generalização; os parâmetros encontrados deverão conduzir a um valor de *MRE* teste tão baixo quanto possível.

O algoritmo iterativo adoptado está implementado na *toolbox* de redes neuronais do MATLAB [83], conforme referido na secção 2.2.3.

Numa primeira fase do processo de aprendizagem, o raio mantém-se constante para todos os nós intermédios; posteriormente, as diferentes heurísticas para a determinação dos raios (ver secção 2.2.3.1) serão testadas comparativamente.

Em cada iteração, após adicionado mais um nó intermédio, o vector de pesos é determinado de acordo com (2.36).

Após a realização de várias aprendizagens, verificou-se que os modelos mais adequados eram aqueles que resultavam da procura do mínimo absoluto para o *MRE* teste. Este facto levou a que se formulasse um critério de paragem semelhante ao que foi adoptado para a aprendizagem de *MLPs*, ou seja, considera-se que certo valor de *MRE* é um mínimo absoluto numa determinada iteração se decorrerem k iterações adicionais sem que esse mínimo seja transposto por um valor inferior. Após vários ensaios concluiu-se que, para a aplicação em curso, um valor de $k=30$ seria aceitável para determinar mínimos absolutos de *MRE* teste.

Na Tab. 4.7 resumem-se alguns dos ensaios que foram feitos com diferentes valores para o raio das Gaussianas, com vista à detecção do raio que conduza a um mínimo de *MRE* teste com um número aceitável de nós na camada intermédia.

Foram feitas aprendizagens com valores para os raios fora do intervalo que se apresenta na Tab. 4.7, verificando-se que:

- Valores demasiadamente baixos para o raio conduzem a subida do *MSE* para o conjunto de teste logo desde as primeiras iterações com a consequente perda de capacidade de generalização.
- Valores mais elevados confirmam a tendência crescente, já verificada na Tab. 4.7, para as medidas de desempenho, correspondendo a uma perda de exactidão dos modelos encontrados.

RAIO	Nº Nós	TREINO		TESTE	
		<i>MSE</i>	<i>MRE</i> (%)	<i>MSE</i>	<i>MRE</i> (%)
0.07	332	4.61E-7	0.11	3.1E-3	5.14
0.08	368	1.4E-8	0.02	1.5E-3	3.13
0.1	238	1.5E-5	0.84	4.8E-4	2.90
0.15	142	3.1E-4	3.35	1.5E-3	6.35
0.17	158	2.9E-4	3.79	1.4E-3	6.13
0.2	105	1.3E-3	6.62	2.8E-3	7.65
0.3	125	1.5E-3	6.71	2.9E-3	10.15
0.5	160	1.3E-3	5.74	3.5E-3	10.50
1	163	2.8E-3	9.50	6.4E-3	12.37
8	51	3.0E-2	40.65	2.8E-2	35.43

Tab. 4.7 – Aprendizagem em diferido, valores mínimos para o *MRE* no conjunto de teste.

Na Tab. 4.7 observa-se que o *MRE* teste é mínimo (2.9%) quando se utiliza um raio de 0.1. Contudo, verifica-se uma grande redução no número de nós intermédios quando se passa do raio de 0.1 para 0.15 (238 para 142). Desta forma, decidiu-se adoptar esta topologia menos complexa, (raio = 0.15 e 142 nós intermédios) para a restante aprendizagem na expectativa de se encontrar modelos com melhor capacidade de adaptação. Na figura Fig.

4.5 apresenta-se a evolução gráfica do *MSE* para o raio considerado. Adicionalmente na Fig. 4.6, apresentam-se as correspondentes evoluções gráficas para o *MRE* e *SSRE*.

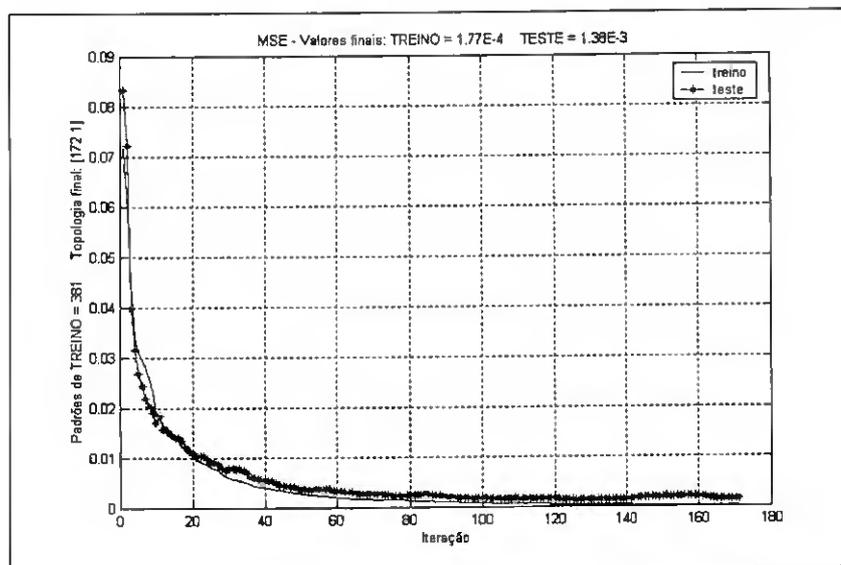


Fig. 4.5 – Evolução do *MSE* na aprendizagem de *RBF* em diferido com raio = 0.15.

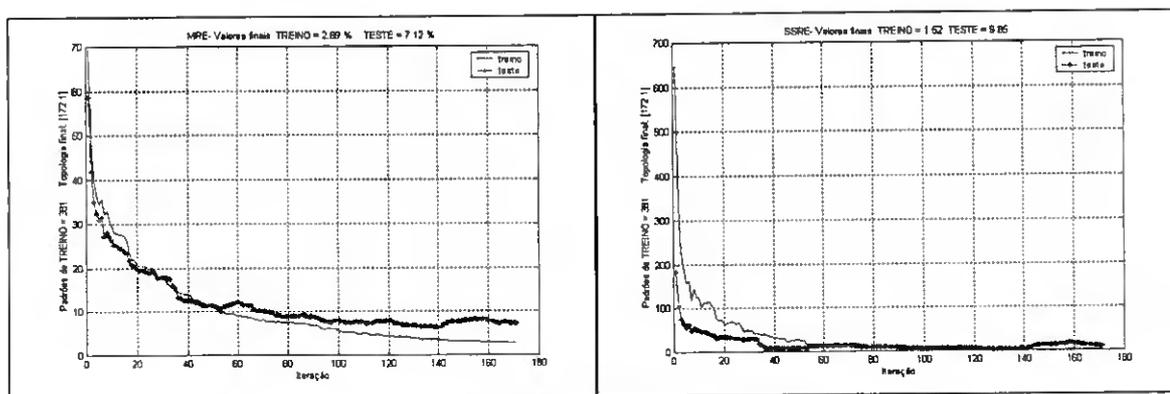


Fig. 4.6 – Aprendizagem de *RBF* em diferido com raio = 0.15. Esquerda: *MRE*, direita: *SSRE*.

Terminada esta secção, podem ser feitas as seguintes leituras:

Os resultados obtidos na aprendizagem em diferido permitem-nos concluir que os indicadores de desempenho para o conjunto de treino apresentam um andamento monótono decrescente, tal como seria de esperar. Este andamento é geralmente acompanhado pelo conjunto de teste, excepto nalguns casos tais como: *MSE* para raio = 0.001, 0.005, 0.01 e 0.02. Assim, podem-se destacar duas conclusões:

- A optimização do *MSE* é distinta da optimização de outros indicadores tais como *MRE* ou *SSRE*, daí resulta a importância do acompanhamento simultâneo de todos estes indicadores.
- A utilização de raios pequenos faz com que o perfil das funções de activação se torne mais localizado, podendo verificar-se alguma perda da capacidade de generalização, com a consequente deterioração dos indicadores para o conjunto de teste.

Embora a solução apresentada para o número de nós corresponda ao mínimo absoluto para o *MRE*, verifica-se, na maior parte dos casos, que se poderá recorrer a um número de nós muito inferior, sem que haja degradação significativa das medidas de desempenho. Este facto é apoiado pelo andamento terminal (praticamente plano) das medidas de desempenho.

Se, por um lado, valores baixos para o raio poderão dificultar a capacidade de generalização e reduzir mesmo a capacidade de aprendizagem e exactidão das redes, valores altos para este parâmetro fazem com que o andamento das medidas de desempenho apresente alguma irregularidade, tal como acontece para raios superiores a 0.5.

Alternativamente à presente situação em que o raio se mantém constante para todos os nós intermédios, na secção 4.4.2 apresentam-se resultados correspondentes à utilização de valores distintos do raio no decurso do mesmo processo de aprendizagem. Com esta abordagem pretende-se reflectir a representatividade de cada centro dentro do espaço de padrões de entrada.

4.4.2 Diferentes heurísticas para a determinação dos raios

No estudo que se segue apresentam-se resultados que permitem avaliar a influência do algoritmo *k-mean clustering* no posicionamento dos centros, bem como as diferenças de desempenho das 6 heurísticas apresentadas na secção 2.2.3.1, para determinação dos raios.

Na Tab. 4.8 reúnem-se as medidas de desempenho obtidas para as diferentes heurísticas, para os modelos que conduziram ao mínimo de *MRE* teste.

A numeração apresentada na 1ª coluna corresponde às diferentes heurísticas de acordo com a convenção apresentada na secção 2.2.3.1, que se recorda de seguida:

1. Desvio padrão empírico.
2. k vizinhos mais próximos.
3. Vizinho mais próximo.
4. Metade da distância máxima entre padrões.
5. Distância máxima entre centros.
6. Distância vizinha média.

No que se refere à heurística 2, k vizinhos mais próximos, o valor de k é dado em termos percentuais relativamente à totalidade de centros existentes em cada situação e, para este caso, foram testados exaustivamente valores de 2% a 90%. A selecção do valor mais adequado teve por base a obtenção de modelos que atendessem aos seguintes critérios:

- menor valor de MRE teste,
- menor número de nós na camada intermédia.

De entre as percentagens experimentadas, os critérios apresentados foram otimizados com respectivamente $k=7\%$ e $k=60\%$, estando as medidas de desempenho registadas nas linhas 2.a ($k=7\%$) e 2.b ($k=60\%$) da Tab. 4.8.

Para a heurística 3, vizinho mais próximo, a determinação do valor de Q foi feita otimizando os objectivos referidos na heurística anterior: MRE teste e número de nós na camada intermédia. A pesquisa destes dois objectivos é feita de forma heurística, variando-se Q em torno de 1.25 como referido na secção 2.2.3.1. Analisando as aprendizagens realizadas, verificou-se que os objectivos enunciados se concretizam respectivamente para $Q=2.1$ e $Q=2.8$, conforme as linhas 3.a e 3.b da Tab. 4.8.

Mantendo os dois objectivos já enunciados (boa capacidade de generalização traduzida por um baixo valor de MRE teste e topologia mínima que se reflecte num menor número de nós intermédios) e observando a da Tab. 4.8, conclui-se que tais objectivos são melhor conseguidos nas linhas 2.a e 4 respectivamente.

Para estas duas soluções, apresentam-se as evoluções gráficas das medidas de desempenho nas Fig. 4.7 a Fig. 4.10.

heurística	Nº Nós	TREINO			TESTE		
		MSE	MRE(%)	SSRE	MSE	MRE(%)	SSRE
1	32	3E-2	37.23	329.9	4.4E-2	36.76	108.4
2.a	232	2.0E-4	2.17	0.4	1.2E-3	5.04	2.1
2.b	89	3.1E-3	11.63	32.1	4.0E-3	9.91	4.4
3.a	163	1.1E-3	5.00	2.5	1.9E-3	7.00	3.0
3.b	73	4.5E-3	14.18	32.8	5.1E-3	11.58	8.0
4	56	9.7E-3	20.94	62.0	1.2E-2	18.17	29.5
5	105	3.8E-3	11.51	20.7	4.6E-3	12.62	14.4
6	105	8.1E-3	17.51	48.9	1.1E-2	19.23	29.4

Tab. 4.8 – Desempenho para as 6 heurísticas.

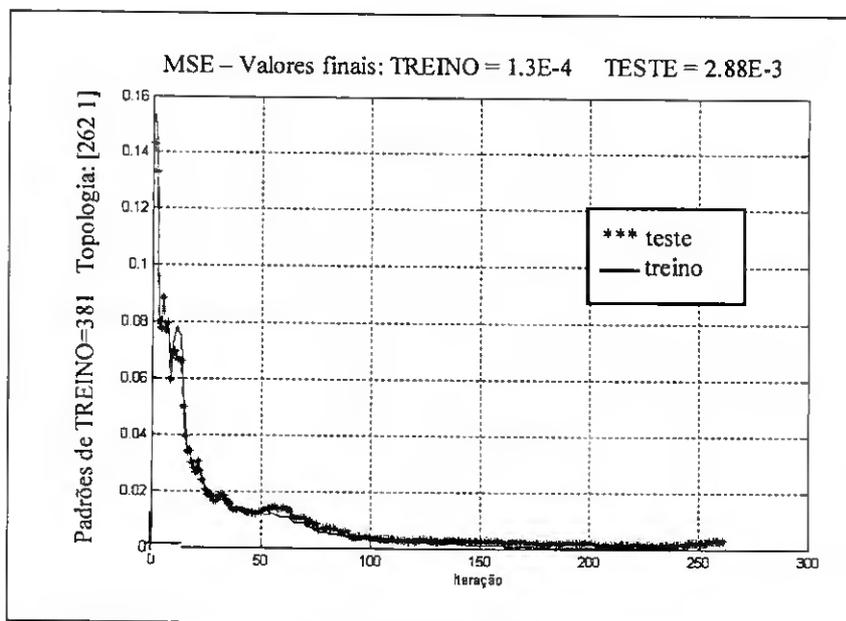


Fig. 4.7 – Aprendizagem de RBF. Heurística 2, k=7%. Evolução do MSE.

A confrontação dos resultados provenientes da utilização das diferentes heurísticas (Tab. 4.8) com os que se obtiveram na fase inicial da aprendizagem em diferido em que os raios foram mantidos constantes (Tab. 4.7), permite-nos concluir que, para a aplicação em estudo, não se verifica melhoria significativa pelo facto de se utilizar diferenciação de raios. Assim, para o prosseguimento da adaptação em linha, utiliza-se a situação apresentada na Tab. 4.7 para o raio=0.15.

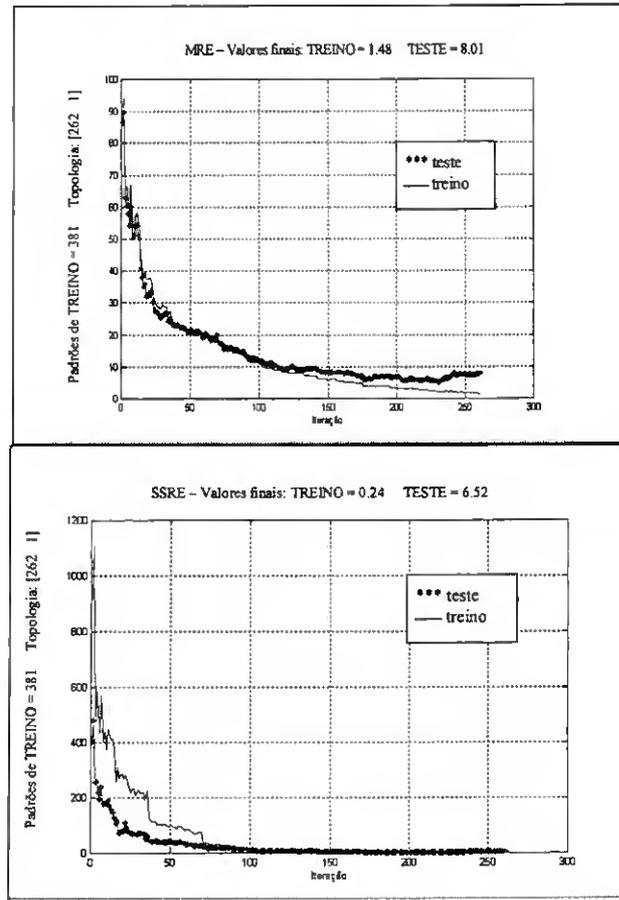


Fig. 4.8 – Aprendizagem de *RBF*. Heurística 2, $k=7\%$. Esquerda: *MRE*, direita: *SSRE*.

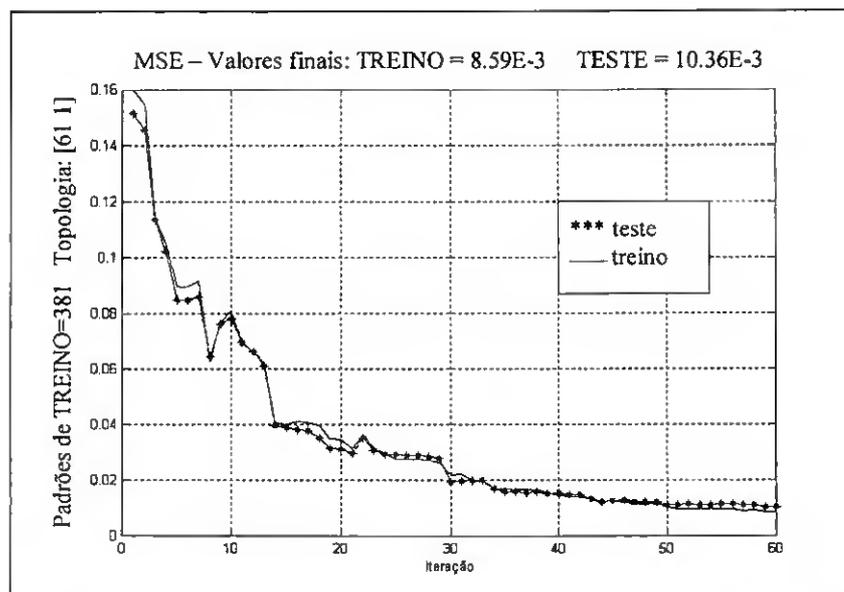


Fig. 4.9 – Aprendizagem de *RBF* com a heurística 4. Evolução do *MSE*.

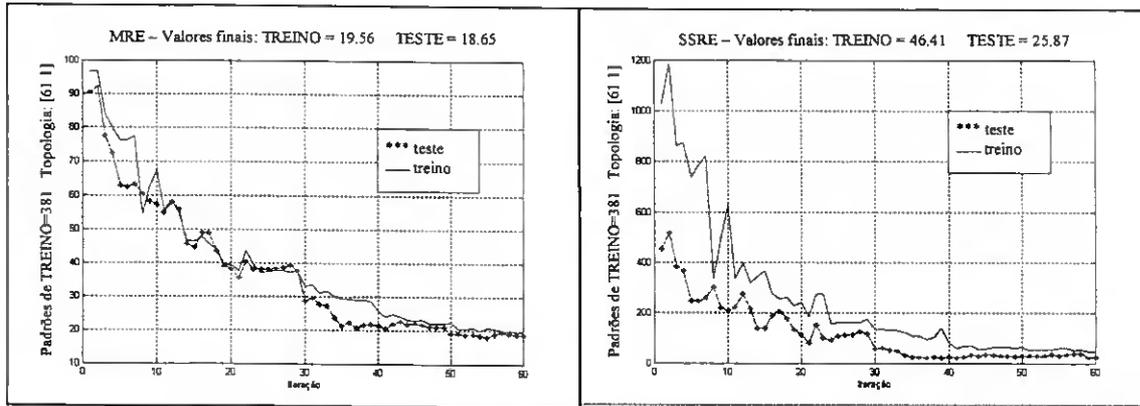


Fig. 4.10 – Aprendizagem de *RBF* com a heurística 4. Esquerda: *MRE*, direita *SSRE*.

4.4.3 Actualização de pesos padrão a padrão de *RBFs*

Durante a adaptação em linha considerou-se o modelo correspondente a $r=0.15$ da Tab. 4.7; os centros e os raios vão manter-se constantes, sendo a adaptação dos pesos feita padrão a padrão de acordo com a regra *LMS*.

Aplicado os conjuntos de adaptação (Tab. 4.2) ao modelo referido, obtém-se o estado inicial das medidas de desempenho para adaptação em linha registado na Tab. 4.9.

Iter.	<i>MSE</i> (10^{-3})		<i>MRE</i> (%)	
	Treino	Teste	Treino	Teste
0	0.9	0.6	3.2	4.9

Tab. 4.9- *MSE* e *MRE* iniciais para adaptação em linha de *RBFs*.

De acordo com a regra de adaptação em linha dos pesos, foram utilizados vários valores da taxa de aprendizagem, registando-se as situações mais relevantes na Tab. 4.10.

α	<i>MSE</i>				<i>MRE</i>			
	TREINO		TESTE		TREINO		TESTE	
	Ganho	MIN.(E-3)	Ganho	MIN.(E-3)	Ganho	MIN.(%)	Ganho	MIN.(%)
0.02	1.76	0.5	0.30	0.6	1.15	2.8	0.49	4.9
0.04	1.85	0.5	0.23	0.6	1.15	2.8	0.44	4.9
0.1	1.81	0.5	0.15	0.6	1.12	2.8	0.35	4.9
0.2	1.64	0.5	0.13	0.6	1.00	3.1	0.31	4.9

Tab. 4.10- Adaptação em linha para as *RBFs*.

A taxa de aprendizagem para a qual houve maior ganho para o conjunto de treino ocorreu para $\alpha = 0.04$, registando-se a correspondente evolução do *MSE* na Fig. 4.11.

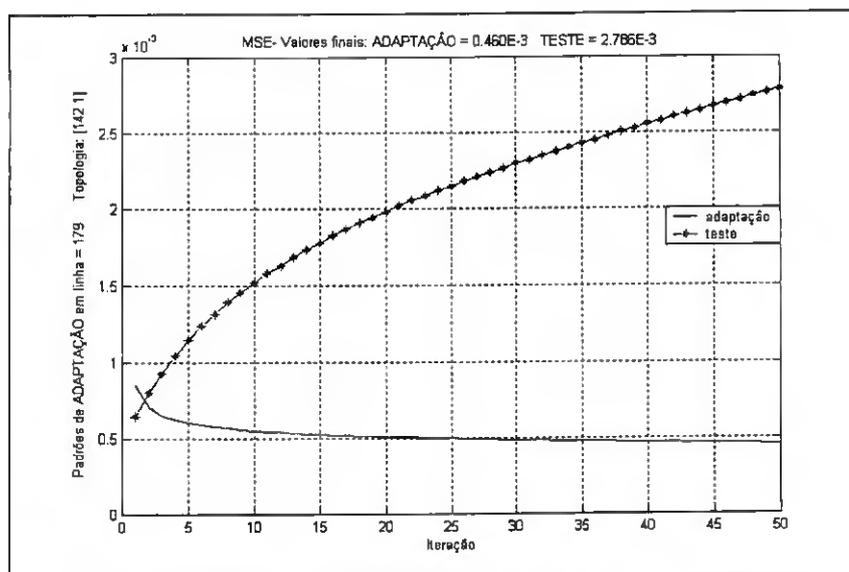


Fig. 4.11 – Adaptação em linha de *RBF* para $\alpha = 0.04$.

4.5 MODELOS NEURONAIS UTILIZANDO *B-SPLINES*

A estrutura básica das redes neuronais do tipo *B-Spline* bem como os fundamentos que serviram de base a esta fase de experimentação estão descritos na secção 2.2.4.

4.5.1 Estruturação da aprendizagem

A estrutura da rede neuronal fica definida durante a aprendizagem em diferido utilizando-se para tal o algoritmo resumido na secção 2.2.4. O critério de desempenho de treino adoptado, é o *BIC* conforme definido na expressão (2.53).

De acordo com o que ficou exposto na secção 2.2.4, o critério de paragem da aprendizagem para este tipo de rede exprime-se da seguinte forma:

A. Fim da aprendizagem: na iteração em que BIC aumenta para o conjunto de treino.

Nestas circunstâncias, dever-se-ia conseguir uma rede neuronal suficientemente exacta com uma complexidade não demasiadamente elevada. Os indicadores de desempenho dados pelas expressões (4.8) avaliados para o conjunto de teste deveriam demonstrar que o modelo assim obtido teria uma boa capacidade de generalização.

Foi efectuado um conjunto de aprendizagens prévias, que permitiram verificar-se que a utilização do critério de paragem A não conduz ao mais baixo valor dos indicadores (4.8) para o conjunto de teste. Considerou-se então conveniente considerar adicionalmente um critério de paragem em que interviesse o desempenho para o conjunto de teste (critério B):

B. Fim da aprendizagem: completadas N iterações após o mínimo absoluto de MRE teste.

Foi igualmente constatado que nem sempre o mínimo do BIC treino antes do primeiro aumento é mínimo absoluto. Tendo em vista a pesquisa do mínimo absoluto para este indicador, permitiu-se um número N de iterações após o primeiro aumento do BIC treino, formalizando-se o critério de paragem C :

C. Fim da aprendizagem: completadas N iterações após o mínimo absoluto de BIC treino.

No problema em estudo, verificou-se que a implementação do critério B ou do critério C , corresponde a um prolongamento do processo de aprendizagem para além do ponto correspondente à paragem de acordo com o critério A .

As aprendizagens realizadas orientam-se de acordo com os seguintes pontos:

- obtenção dum modelo exacto e com boa capacidade de generalização – deverá apresentar simultaneamente bom desempenho para o conjunto de treino e para o conjunto de teste.
- obtenção dum modelo o mais exacto possível – deverá apresentar um bom desempenho para o conjunto de treino, descurando-se o desempenho para o conjunto de teste.

Nas secções seguintes, para cada estratégia de aprendizagem adoptada, será explicitado o correspondente critério de paragem adoptado.

4.5.2 Modelo com boa capacidade de generalização

Algumas simulações iniciais permitiram verificar que valores elevados para o *MRE* teste se deviam a um sub-conjunto de alguns padrões; contudo, para a maioria dos padrões verificou-se que o mapeamento apresentava razoável exactidão. Com base nesta constatação, poder-se-ia pensar que os conjuntos de aprendizagem, da forma como foram gerados, não seriam os mais adequados para facultarem um treino que proporcionasse uma boa capacidade de generalização. Assim, optou-se por anteceder cada processo de aprendizagem pela geração de novos conjuntos de treino e teste.

A operação de supressão de nós deverá ser regulada de forma a permitir a evolução da rede sem que esta se torne demasiadamente complexa, pelo que se considerou um parâmetro ajustável denominado *Step Pruning (SP)*, que vai determinar de quantas em quantas iterações deverá ocorrer tal operação.

Tomando estes pressupostos, descreve-se um algoritmo (Tab. 4.11) que visa a obtenção dum modelo com boa capacidade de generalização.

1. gerar aleatoriamente conjuntos de aprendizagem disjuntos;
2. variar *SP* para alguns inteiros: $i, i+1, i+2, \dots$;
3. para cada inteiro do ponto anterior, executar um processo de aprendizagem;
4. do lote de aprendizagens do ponto anterior, seleccionar aquela que conduziu ao menor valor de *MRE* teste e registar o correspondente valor de *SP* mínimo;
5. para *SP* mínimo do ponto anterior, executar uma série de inicializações dos conjuntos de aprendizagem e correspondentes processos de aprendizagem;
6. a partir das simulações efectuadas no ponto anterior, seleccionar o par treino/teste que conduziu ao menor *MRE* teste;
7. se já tiverem ocorrido um número suficiente de ciclos, parar. Senão, para o par treino/teste do ponto anterior, passar para o ponto 2.

Tab. 4.11- Algoritmo para obtenção de modelo com boa capacidade de generalização.

A partir dos conjuntos de treino e teste iniciais gerados aleatoriamente, tiveram lugar aprendizagens de *B-Splines* de 3ª ordem, para $SP \in \{3, 4, 5, 6, 7, 8\}$. Na Tab. 4.12 regista-se,

para cada valor de SP , a iteração onde ocorreu o primeiro mínimo de BIC treino (critério A) e a iteração onde ocorreu o mínimo absoluto do MRE teste (critério B com $N=5$).

SP		Iter.	BIC treino	$MSE (10^{-3})$		$MRE (%)$	
				Treino	Teste	Treino	Teste
3	min BIC	9	-1962.7	2.3	2.5	23.1	22.1
	min MRE_{teste}	13	-1967.1	1.9	2.0	19.8	19.3
4	min BIC	9	-1962.7	2.3	2.5	23.1	22.1
	min MRE_{teste}	19	-1972.2	1.8	1.8	19.2	18.2
5	min BIC	10	-1968.6	2.3	2.5	23.1	22.1
	min MRE_{teste}	21	-1952.3	0.8	1.0	12.2	13.0
6	min BIC	9	-1962.7	2.3	2.5	23.1	22.1
	min MRE_{teste}	16	-1977.4	1.8	1.7	19.4	18.3
7	min BIC	9	-1962.7	2.3	2.5	23.1	22.1
	min MRE_{teste}	12	-1961.0	1.9	1.9	22.2	21.1
8	min BIC	9	-1962.7	2.3	2.5	23.1	22.1
	min MRE_{teste}	12	-1961.0	1.9	1.9	22.2	21.1

Tab. 4.12- Mínimos de BIC treino e MRE teste tendo em vista a procura de modelo com boa capacidade de generalização.

Sendo o objectivo desta fase de aprendizagem encontrar o modelo com melhor capacidade de generalização, tomando-se por base o algoritmo apresentado na Tab. 4.11 e observando-se a Tab. 4.12 pressupõe-se que este será obtido para $SP=5$. Desta forma procedeu-se a um lote de cerca de 100 aprendizagens com $SP=5$, sendo, no início de cada uma, gerados aleatoriamente os conjuntos de aprendizagem; o valor mais baixo para o MRE teste ocorreu na situação que se regista na Tab. 4.13.

SP	Iter.	BIC treino	$MSE (10^{-3})$		$MRE (%)$	
			Treino	Teste	Treino	Teste
5	20	-2396.0	0.093	0.88	4.4	12.4

Tab. 4.13- Mínimo de MRE teste após um lote de inicializações aleatórias dos conjunto de aprendizagem.

De acordo com o algoritmo apresentado na Tab. 4.11 seguia-se um lote de aprendizagens com os conjuntos de treino e teste constantes, variando-se o SP . Contudo, dada a

similaridade entre o *MRE* teste da Tab. 4.12 (13%), e o correspondente valor apresentado na Tab. 4.13 (12.4%), concluiu-se que a continuação do algoritmo não conduziria a melhorias significativas. Desta forma optou-se por terminar o algoritmo, apresentando-se como valor mais baixo para o *MRE* teste 12.4%, obtido para $SP = 5$, conforme Tab. 4.13.

Com o intuito de se melhorarem os resultados apresentados anteriormente procedeu-se a um lote adicional de aprendizagens utilizando-se *B-Splines* de 2ª ordem, cujos resultados se registam na Tab. 4.14.

A utilização de *B-Splines* de 2ª ordem conduz, dum modo geral, a valores mais baixos para o *MRE* teste, antevendo-se melhor capacidade de generalização.

Dado que se pretende seleccionar um modelo com boa capacidade de generalização, deverão ser tidas em conta as colunas das medidas de desempenho para o conjunto de teste. Assim, seleccionou-se como modelo de partida para a aprendizagem em linha uma das situações que conduziu aos menores valores de *MSE* e *MRE* avaliados para o conjunto de teste³, quando se procurava o mínimo de *MRE* para esse conjunto: $SP=6$, MSE teste = $4.2E-3$; MRE teste = 9%. As 17 primeiras iterações da aprendizagem que conduziram a esses resultados apresentam-se na Fig. 4.12.

4.5.3 Modelo para o conjunto de treino

Na perspectiva abordada nesta secção os processos de aprendizagem decorrem atendendo apenas ao conjunto de treino, tanto no que respeita ao algoritmo de aprendizagem como no que se refere ao critério de paragem.

Tal como já foi apontado anteriormente, a paragem precoce do processo de aprendizagem quando o *BIC* treino sobe pela primeira vez impede, em alguns casos, que o mínimo absoluto deste indicador seja encontrado; desta forma, de acordo com o critério *C* permitem-se $N=15$ iterações para além do último mínimo absoluto encontrado.

³ assinalado na Tab. 4.14.

SP		Iter.	BIC treino	MSE (10^{-3})		MRE(%)		Complex.
				Treino	Teste	Treino	Teste	
2	min BIC	16	-1814.4	0.3	13.7	2.8	10.6	205
	min MRE_{teste}	17	-1808.5	0.3	13.7	2.8	10.5	206
3	min BIC	21	-1785.1	0.3	55.4	3.4	14.2	214
	min MRE_{teste}	17	-1699.9	0.7	10.2	4.0	10.2	182
4	min BIC	16	-1814.4	0.3	13.7	2.8	10.6	205
	min MRE_{teste}	19	-1830.4	0.3	15.2	2.1	10.4	208
5	min BIC	21	-1792.4	0.3	58.3	3.1	12.5	213
	min MRE_{teste}	17	-1699.9	0.7	10.2	4.0	10.2	182
6	min BIC	20	-1765.1	0.4	92.90	3.5	76.7	205
	min MRE_{teste}	16	-1670.1	0.7	4.2	4.0	9.0	187
7	min BIC	21	-1761.1	0.3	46.3	3.4	12.8	219
	min MRE_{teste}	16	-1670.1	0.7	4.2	4.0	9.0	187
8	min BIC	16	-1814.4	0.3	13.7	2.8	10.6	205
	min MRE_{teste}	25	-1838.9	0.3	12.8	2.2	9.7	214
10	min BIC	20	-1761.1	0.3	46.3	3.4	12.8	219
	min MRE_{teste}	16	-1670.1	0.7	4.2	4.0	9.0	187
12	min BIC	24	-1761.1	0.3	46.3	3.4	12.8	219
	min MRE_{teste}	16	-1670.1	0.7	4.2	4.0	9.0	187

Tab. 4.14- Mínimos de BIC treino e MRE teste para B-Splines de 2ª ordem.

Os valores correspondentes ao conjunto de teste apresentados na Tab. 4.16 têm carácter informativo, não intervindo no desenvolvimento da rede neuronal. Nesta tabela, para cada SP, são apresentados dois estados do mesmo processo de aprendizagem:

- iteração para a qual se obtém o primeiro mínimo do BIC treino,
- iteração para a qual se obtém o mínimo absoluto do BIC treino⁴.

A última coluna refere-se à complexidade do modelo que conduz ao mínimo absoluto do BIC treino. Para cada modelo, a complexidade apresentada corresponde à totalidade de

⁴ Permitiu-se 15 valores superiores consecutivos após este mínimo

funções de activação, independentemente do facto de estarem ou não ligadas a ramos com pesos nulos.

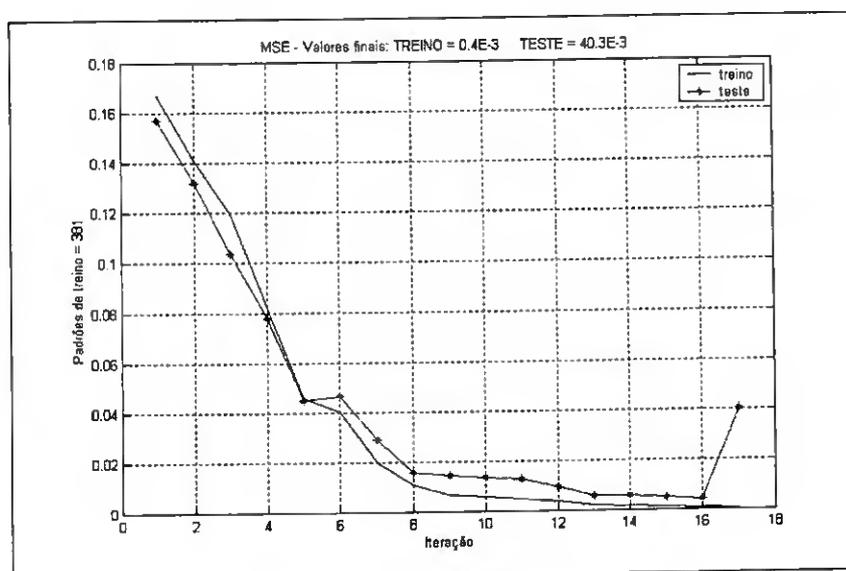


Fig. 4.12 – Aprendizagem em diferido de *B-Spline*.

A leitura das Tab. 4.12 e Tab. 4.16 permite-nos concluir que as duas estratégias apresentadas (procura do modelo com melhor capacidade de generalização e procura do modelo mais exacto para o conjunto de treino) podem ser inseridas dentro dum mesmo plano de aprendizagens. Para cada valor do parâmetro *SP*, os valores das medidas de desempenho registadas nessas tabelas apenas se referem a estados diferentes do processo de aprendizagem. Para todas as situações estudadas, esses estados aparecem pela ordem apresentada na Tab. 4.15.

1º	1º mínimo do <i>BIC</i> treino
2º	mínimo absoluto do <i>MRE</i> teste
3º	mínimo absoluto do <i>BIC</i> treino

Tab. 4.15 – Diferentes estados dum processo de aprendizagem.

SP	BIC treino	Iter.	BIC treino	MSE (10^{-3})		MRE (%)		Complex.
				Treino	Teste	Treino	Teste	
2	1º mínimo	10	-1968.6	2.3	2.5	23.1	22.1	239
	mínimo absol.	48	-2651.6	0.035	2.61	1.9	473.3	
3	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	75
	mínimo absol.	15	-1980.4	1.8	1.9	19.7	19.28	
4	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	104
	mínimo absol.	44	-2614.7	0.034	3.06	2.0	509.2	
5	1º mínimo	10	-1968.6	2.8	2.5	23.1	22.1	234
	mínimo absol.	45	-2543.1	0.046	348	2.7	60.2	
6	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	113
	mínimo absol.	54	-2589.8	0.036	3410	2.3	528.2	
7	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	241
	mínimo absol.	28	-2681.2	0.032	1.5E+9	1.9	4.0E+5	
8	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	251
	mínimo absol.	41	-2724.8	0.029	1.4E+9	1.7	3.8E+5	
10	1º mínimo	10	-1968.6	2.3	2.5	23.1	22.1	59
	mínimo absol.	10	-1968.6	2.3	2.5	23.1	22.1	
12	1º mínimo	9	-1962.7	2.3	2.5	23.1	22.1	61
	mínimo absol.	9	-1962.7	2.3	2.5	23.1	22.1	

Tab. 4.16- Aprendizagem com vista à exactidão do modelo para o conjunto de treino.

Nesta fase do problema, pode-se dizer que o critério usualmente utilizado, segundo o qual o processo de aprendizagem termina no momento em que o *BIC* sobe pela primeira vez, trás como consequência o facto de poder não se encontrar o mínimo absoluto de *BIC* treino (ver Tab. 4.16). De forma análoga, a paragem do processo de aprendizagem nessas circunstâncias não conduz ao modelo com melhor capacidade de generalização. Observando-se a Tab. 4.12, conclui-se que o mínimo absoluto do *MRE* teste ocorre sempre depois de se ter atingido o 1º mínimo do *BIC* treino. O mínimo absoluto do *BIC* treino ocorre, na maior parte dos casos, muito mais tardiamente, (ver Tab. 4.16); este estado de aprendizagem corresponde a modelos para os quais a rede neuronal fica demasiadamente ajustada ao conjunto de treino. Nessas circunstâncias, a capacidade de generalização é perdida; este facto é apoiado pelos elevados valores obtidos para o *MRE* teste no estado de mínimo absoluto do *BIC* treino (ver Tab. 4.16).

A observação conjunta de ambas as tabelas Tab. 4.12 e Tab. 4.16 permite concluir que os valores relativos *MRE* andam muito à volta dos 20% para ambos os conjuntos de aprendizagem, com apenas dois tipos de excepções:

1. Tab. 4.12: $SP=5$, no estado em que é mínimo o *MRE* teste
2. Tab. 4.16: estado em que se atinge o mínimo absoluto do *BIC* treino.

A situação a que se refere o ponto 1 apresenta-se isolada, impedindo que se dê especial relevância ao facto de se utilizar um $SP = 5$; apesar disso, os valores de *MRE* para essa situação apresentam-se consideravelmente melhores que os restantes dentro da mesma tabela. Mesmo assim, os valores de *MRE* aqui obtidos são piores do que aqueles conseguidos para as *MLPs* e *RBFs*.

No ponto 2 reforça-se a ideia já apreendida para outros tipos de redes neuronais, segundo a qual uma aprendizagem demasiadamente extensa faz com que a rede neuronal decore o conjunto de treino, em vez de ganhar capacidade de generalização. Este aspecto, sempre presente em estudos que envolvem redes neuronais, apoia a importância de se trabalhar com um conjunto de teste que possibilite inferir quanto à capacidade de generalização do modelo obtido, em cada momento da sua construção.

Por último, e antes de se passar ao problema da adaptação em linhas das redes neuronais do tipo *B-Spline*, salienta-se que o modelo mais adequado é o que se apresenta na Tab. 4.13. Dado que apenas se verificou uma ligeira melhoria após se executar as 100 inicializações dos conjuntos de aprendizagem, pode-se concluir que o conjunto de teste extraído aleatoriamente com distribuição de probabilidades uniforme do universo base é representativo desse universo base.

4.5.4 Actualização de pesos padrão a padrão de *B-Splines*

Como já foi referido anteriormente, nesta fase de adaptação assume-se a estrutura do modelo optimizado em diferido, incidindo as modificações apenas sobre os pesos da camada de saída, de acordo com a regra *NLMS*, conforme se apresentou na secção 2.2.4.1.

Cada sequência de adaptação consiste em 50 iterações, para valores de tx relevantes que permitem inferir sobre o comportamento do modelo.

A adaptação em linha teve lugar a partir do modelo correspondente à linha assinalada na Tab. 4.14, que resultou da iteração 16 da aprendizagem em diferido que se apresenta na Fig. 4.12.

A constituição desse modelo apresenta-se na Tab. 4.17, tendo-se adoptado as seguintes convenções:

Nsm - totalidade de sub-modelos que constituem a rede neuronal; para cada sub-modelo:

Var. - variáveis que constituem o sub-modelo,

Nint. - totalidade de nós interiores considerados para cada dimensão,

Ord. - ordem das funções de base, para cada dimensão.

Nsm	Var.	Nint.	Ord.	Var.	Nint.	Ord.	Var.	Nint.	Ord.
5	1	1	2	2	0	2	3	0	2
	Var.	Nint.	Ord.	Var.	Nint.	Ord.			
	1 2	5 4	2 2	3 1 2	3 3 4	2 2 2			

Tab. 4.17- Modelo obtido com aprendizagem em diferido de *B-Splines*.

Quando se submetem os conjuntos de adaptação (Tab. 4.2) a esse modelo obtêm-se as medidas de desempenho que se apresentam na Tab. 4.18.

Iter.	<i>MSE</i> (10^{-3})		<i>MRE</i> (%)	
	Treino	Teste	Treino	Teste
0	2.6	1.5	3.8	13.3

Tab. 4.18- *MSE* e *MRE* iniciais para adaptação em linha de *B-Splines*.

A partir deste estado inicial, tomando as definições apresentadas na secção 4.3.1 a propósito da Tab. 4.6, foram feitos lotes de actualizações dos pesos padrão a padrão, para vários valores para a taxa de aprendizagem. As situações mais relevantes registam-se na Tab. 4.19.

Tendo em conta a evolução do *MSE* verifica-se que a progressiva melhoria verificada para o conjunto de treino é acompanhada duma deterioração para o conjunto de teste. Assim, será seleccionada a situação correspondente a $tx = 4\%$, que corresponde a uma sensível

melhoria para o conjunto de treino, sem que o desempenho para o conjunto de teste se veja demasiadamente deteriorado. A evolução da aprendizagem para essa situação apresenta-se na Fig. 4.13.

<i>tx</i> (%)	<i>MSE</i>				<i>MRE</i>			
	ADAPTAÇÃO		TESTE		ADAPTAÇÃO		TESTE	
	Ganho	MIN. (E-3)	Ganho	MIN. (E-3)	Ganho	MIN.(%)	Ganho	MIN.(%)
1	1.03	1.53	0.87	2.77	0.99	3.8	0.93	13.3
2	1.06	1.49	0.77	2.77	0.98	3.8	0.88	13.3
3	1.08	1.46	0.69	2.77	0.98	3.8	0.83	13.3
4	1.10	1.44	0.63	2.77	0.98	3.8	0.80	13.3
5	1.11	1.42	0.59	2.77	0.97	3.8	0.78	13.3
6	1.12	1.41	0.55	2.77	0.96	3.8	0.76	13.3
10	1.14	1.38	0.46	2.77	0.88	3.8	0.70	13.3
20	1.14	1.39	0.36	2.77	0.81	3.8	0.62	13.3
30	1.13	1.40	0.29	2.77	0.82	3.8	0.57	13.3
40	1.14	1.39	0.24	2.77	0.86	3.8	0.53	13.3
60	1.21	1.31	0.17	2.77	0.87	3.8	0.46	13.3

Tab. 4.19- Adaptação em linha para as *B-Splines*.

4.6 ANÁLISE COMPARATIVA DE RESULTADOS

Nesta secção vão ser comparados os resultados relevantes referentes às 3 redes neuronais em estudo [6]. A sumarização de resultados que vai ser apresentada possibilita concluir sobre os 3 tipos de redes neuronais, no que diz respeito à capacidade de produção de modelos exactos com boa capacidade de generalização e de adaptação. As leituras efectuadas de seguida vão usar como suporte as medidas de desempenho apresentadas ao longo do capítulo e os valores das complexidades dos modelos correspondentes.

Os processos de aprendizagem em diferido que conduziram aos modelos com melhor desempenho para cada tipo de rede neuronal, correspondem às representações gráficas das Fig. 4.3, Fig. 4.5 e Fig. 4.12, respectivamente. A partir desses processos de aprendizagem, podem-se resumir os valores que se apresentam na Tab. 4.20.

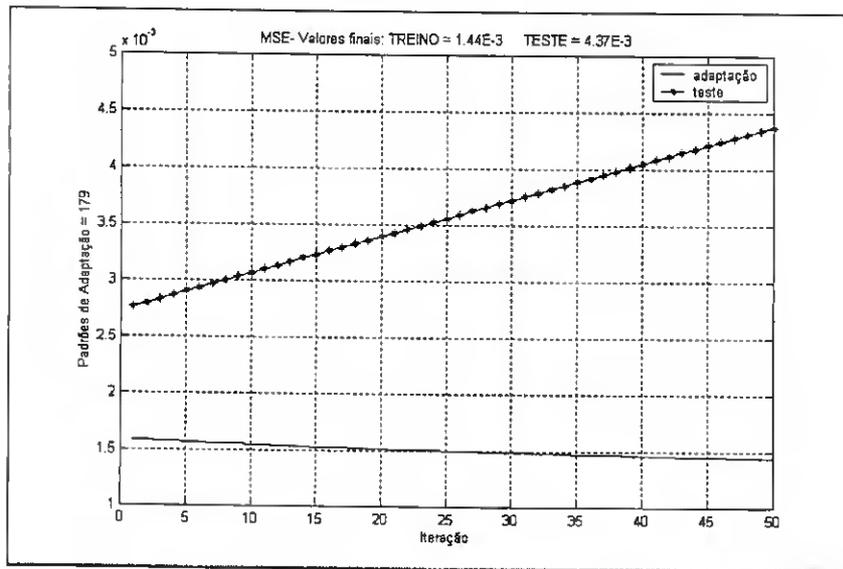


Fig. 4.13 – Adaptação de *B-Spline* para $\alpha=4\%$.

RN	$MSE(x10^{-3})$		$MRE(\%)$		Nós	Comp.
	Treino	Teste	Treino	Teste		
<i>MLPs</i>	0.15	0.61	2.5	3.9	15	92
<i>RBFs</i>	0.20	1.40	2.7	7.1	142	569
<i>B-Spline</i>	0.70	4.20	4.0	9.0	56	187

Tab. 4.20- Aprendizagem em diferido para os 3 tipos de redes neuronais.

Tomando os modelos que conduziram à Tab. 4.20, mas aplicados aos conjuntos de adaptação (coluna da direita da Tab. 4.2), obtêm-se os valores iniciais⁵ para adaptação em linha, que se apresentam na Tab. 4.21.

RN	$MSE(10^{-3})$		$MRE(\%)$	
	Treino	Teste	Treino	Teste
<i>MLPs</i>	0.2	0.4	1.9	3.5
<i>RBFs</i>	0.9	0.6	3.2	4.9
<i>B-Splines</i>	2.6	1.5	3.8	13.3

Tab. 4.21- Resultados iniciais para a adaptação em linha.

A observação da Tab. 4.20 permite verificar que as exactidões alcançadas para o conjunto de treino pelos três tipos de redes neuronais andam dentro da mesma ordem de grandeza. A tabela de valores iniciais para adaptação em linha (Tab. 4.21) corrobora esta observação.

⁵ Estes valores não são mais do que uma compilação das Tab. 4.5, Tab. 4.9 e Tab. 4.18.

No que diz respeito à complexidade, tomando por base a Tab. 4.20, verifica-se que as redes neuronais do tipo *RBF* são as mais complexas; estas redes juntamente com as redes neuronais do tipo *B-Spline* são as que apresentam pior capacidade de generalização. No extremo oposto estão as redes neuronais do tipo *MLP* que, apresentando os menores valores das medidas de desempenho para o conjunto de teste (maior capacidade de generalização), são as que têm menor complexidade.

No que se refere à capacidade de adaptação das redes neuronais após aprendizagem em diferido, apresentam-se na Tab. 4.22 os valores dos ganhos de adaptação para o *MSE*, conseguidos para os modelos referentes a cada tipo de rede neuronal, para o qual a distorção foi menor.

<i>RN</i>	Ganho de <i>MSE</i>		
	Treino	Teste	Treino \cup Teste
<i>MLPs</i>	1.18	0.33	0.38
<i>RBFs</i>	1.80	0.21	0.33
<i>B-Splines</i>	1.10	0.63	0.69

Tab. 4.22- Ganhos de adaptação do *MSE*.

A informação apresentada na Tab. 4.22 permite-nos verificar que as redes neuronais do tipo *B-Spline* foram aquelas que obtiveram um melhor balanceamento entre o ganho no conjunto de treino e deterioração no conjunto de teste (4ª coluna da Tab. 4.22). Outros resultados podem ser encontrados em [6].

4.7 CONCLUSÕES

Este capítulo versou sobre a escolha do tipo de rede neuronal mais apropriado para a construção de modelos neuronais, tendo em conta a exactidão, boa capacidade de generalização e de adaptação.

Incluindo o objectivo do trabalho em curso o controlo de processos variantes no tempo, facilmente se conclui que o espaço de aprendizagem das redes neuronais é muito extenso; assim, optou-se por se cingir o estudo, nesta fase, a um processo tipo invariante no tempo.

Com base na análise desse processo, se concluirá qual o tipo de rede neuronal a utilizar no seguimento do trabalho.

Cada um dos três tipos de redes neuronais: *MLP*, *RBF* e *B-Spline*, foi tratado separadamente; após aprendizagem em diferido para conclusão sobre exactidão e generalização, tiveram lugar os correspondentes processos de adaptação.

Tomando por base os valores obtidos para o *MSE* treino na fase de aprendizagem em diferido, conclui-se que todos os tipos de redes neuronais utilizados apresentam exactidão aceitável para os mapeamentos em causa. No que respeita a capacidade de generalização, as *MLPs* são as que conduzem aos melhores resultados, provavelmente devido ao algoritmo adoptado e ao facto de utilizarem o modelo de menor complexidade. Os resultados a que se chegaram com as *RBFs* foram penalizados pelo facto de se ter utilizado um valor comum para todos os raios. Posteriores ensaios levados a cabo com diferenciação dos raios apontam para melhoria da capacidade de generalização para este tipo de rede neuronal.

Considerando a capacidade de adaptação, as *B-Splines* são o único tipo que conduz a resultados aceitáveis, dada a sua estrutura em oposição ao perfil distribuído utilizado pelas funções de activação das *MLPs*. No que respeita às *RBFs*, a escolha de raio comum que se mantém durante a adaptação em linha condiciona o desempenho deste tipo de rede neuronal.

Face ao exposto, as redes neuronais utilizadas no seguimento do trabalho são do tipo *B-Spline*.

Nos capítulos que se seguem, será estabelecido um espaço de trabalho adequado à modelização pretendida; serão testadas diferentes estratégias no sentido de se construir modelos neuronais utilizando *B-Spline* que acomodem processos variantes no tempo.

5. CONJUNTOS DE APRENDIZAGEM PARA MODELAÇÃO DE PROCESSOS VARIANTES NO TEMPO

5.1 INTRODUÇÃO

No capítulo anterior, após estudo comparativo dos tipos de redes neuronais: *MLP*, *RBF* e *B-Spline*, verificou-se que as redes neuronais do tipo *B-Spline* são aquelas que conduzem a um melhor desempenho em termos de adaptação, quando considerada a totalidade do espaço de aprendizagem; tais conclusões basearam-se no controlo dum processo invariante no tempo.

Antes de se estender a aprendizagem de redes neuronais do tipo *B-Spline* a processos variantes no tempo, determina-se no presente capítulo os conjuntos para a referida aprendizagem.

Na secção 5.2 é feita uma discriminação de funções de transferência de processos típicos encontrados na indústria, tendo como base um exemplo de um sistema de controlo de nível. Na secção 5.3 são determinados parâmetros *PID* para os processos obtidos na secção anterior que, juntamente com as medidas de identificação dos respectivos processos, constituirão os conjuntos de aprendizagem para a modelação de processos variantes no tempo.

O capítulo termina com a secção 5.4, onde são apresentadas algumas conclusões acerca dos conjuntos que irão ser utilizados nos procedimentos de aprendizagem referentes ao capítulo seguinte.

5.2 CONJUNTO BASE DE SISTEMAS DE EXPERIMENTAÇÃO

Pretende-se aplicar a sintonia neuro-genética de controladores *PID* a processos industriais comuns. Dada a grande diversidade deste tipo de processos, a possibilidade desta sintonia obter igual sucesso, independentemente do processo, revela-se bastante reduzida. Assim sendo, vai-se restringir o número de processos a alguns exemplos típicos, definindo-se, um universo que constituirá um conjunto base de experimentação, para o qual a sintonia neuro-genética será preferencialmente testada.

Nesta secção vai ser feito um estudo sobre a topologia duma malha típica de controlo de nível, caracterizada pelas ordens dos seus blocos constituintes, constantes de tempo e tempos de atraso. Desta forma obtêm-se as funções de transferência que constituem o conjunto base de experimentação.

5.2.1 Sistema de controlo de nível

Considere-se o sistema de controlo de nível [136] que se apresenta na Fig. 5.1.

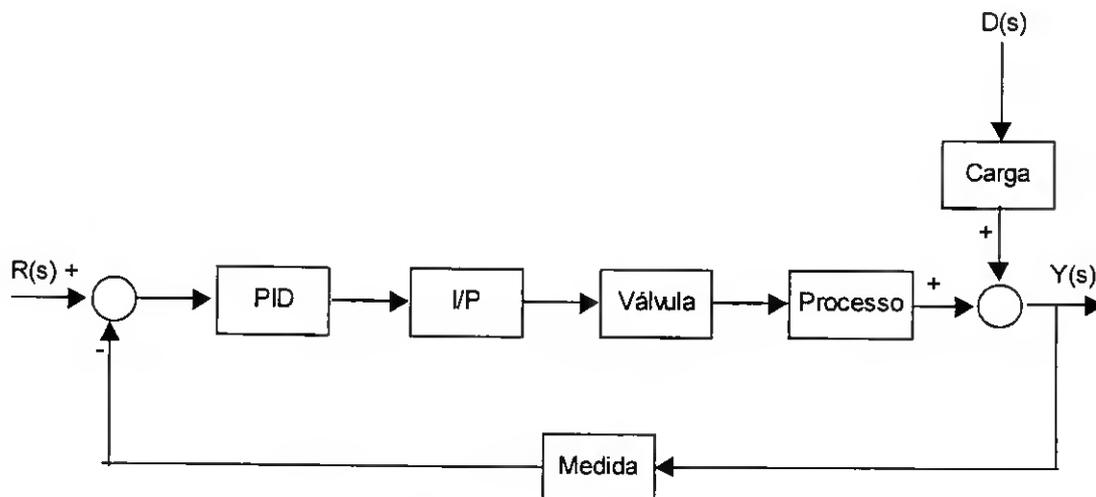


Fig. 5.1 – Sistema de controlo de nível.

Pretende-se que o controlador *PID* regule a abertura da válvula que dará acesso ao recipiente para o qual se pretende controlar o nível de fluído contido. O bloco *I/P* é responsável pela actuação da válvula fazendo a transformação corrente → pressão; em

termos da dinâmica de sistema, este bloco introduz um ganho k . A saída $Y(s)$ do sistema está sujeita a perturbações aditivas representadas por $D(s)$, sendo a sua saída medida e comparada com a referência $R(s)$.

A válvula e o equipamento de medida podem ser modelados pelas funções de transferência $V(s)$ e $M(s)$, com as constantes de tempo em segundos, dadas respectivamente por:

$$V(s) = \frac{k_v}{T_v s + 1}, \quad T_v \in [1 \ 10] \quad M(s) = \frac{k_m}{T_m s + 1}, \quad T_m \in [1 \ 60] \quad (5.1)$$

O processo é modelado por uma função de transferência genérica $G(s)$ com tempo de atraso, de primeira ordem para processos rápidos e de segunda ordem para processos lentos. Dum modo geral [136], consideram-se rápidos os processos cujas constantes de tempo são inferiores a 2 minutos e lentos os processos cujas constantes de tempo são superiores a 1 minuto, até horas. Os valores considerados para os tempos de atraso não deverão exceder 10% da constante de tempo para sistemas de 1ª ordem e 20% da constante de tempo mais elevada para sistemas de 2ª ordem.

5.2.2 Formas gerais de sistemas a controlar

Os processos que constituem o conjunto base de experimentação dividem-se em duas categorias fundamentais: processos lentos e processos rápidos.

Quando se trabalha com processos lentos, pode-se desprezar a dinâmica da válvula e do equipamento de medida, face à dinâmica do processo, obtendo-se então exemplos de 1ª e 2ª ordem dados respectivamente por (5.2) e (5.3).

$$G(s) = \frac{e^{-Ls}}{T_1 s + 1}; \quad T_1 \in [60 \ 120]s; \quad L \in \left[0 \ \frac{T_1}{5}\right] \quad (5.2)$$

$$G(s) = \frac{e^{-Ls}}{(T_1 s + 1)(T_2 s + 1)}; \quad T_1, T_2 \in [60 \ 120]s; \quad L \leq \frac{\max(T_1, T_2)}{10} \quad (5.3)$$

Situação diferente pode ser constatada quando se trabalha com processos rápidos, em que as dinâmicas da válvula e do equipamento de medida têm que ser tidas em conta na

definição do sistema onde vai actuar o controlador *PID*. Assim, os processos rápidos (constantes de tempo não excedem os 2mn) podem ser considerados de 1ª ou 2ª ordem.

Na presente aplicação considera-se apenas processo lentos de 1ª e 2ª ordem com tempo de atraso dados genericamente por:

$$H(s) = k \frac{e^{-Ls}}{(T_1s+1)(T_2s+1)} \quad (5.4)$$

A partir da expressão (3.5) obtêm-se as medidas de identificação $F(\sigma)$ do processo, dadas por (5.5).

$$F(\sigma) = \frac{e^{-\frac{L}{T_r}\sigma}}{\left(1 + \frac{T_1}{T_r}\sigma\right)\left(1 + \frac{T_2}{T_r}\sigma\right)} \quad (5.5)$$

Seja por hipótese $\sigma \in \{0.1, 0.5, 1, 2, 5, 10, 15, 20\}$.

Para os sistemas em estudo verificou-se que $T_2 = 0 \vee \frac{T_2}{T_1} \in [0.1 \ 1]$ e $\frac{L}{\max(T_1, T_2)} \in [0 \ 1]$.

Para estas medidas de identificação, apenas interessa a relação entre as constantes de tempo e tempo de atraso e não os seus valores absolutos. Fixando $T_1=1s$, definindo-se variáveis auxiliares x_1 e x_2 tais que $\frac{T_2}{T_1} = x_1$ e $\frac{L}{\max(T_1, T_2)} = x_2$, verifica-se que:

$$\begin{aligned} T_1 &= 1 \\ T_2 &= x_1 & x_1 &= 0 \vee x_1 \in [0.1 \ 1] \text{ e } x_2 \in [0 \ 1] \\ L &= x_2 \max(T_1, T_2) = x_2 \end{aligned} \quad (5.6)$$

Com estas definições, a expressão das medidas de identificação (5.5) será dada em termos de variáveis auxiliares x_1 e x_2 por (5.7).

$$F(\sigma) = \frac{e^{-\frac{x_2}{1+x_1+x_2}\sigma}}{\left(1 + \frac{1}{1+x_1+x_2}\sigma\right)\left(1 + \frac{x_1}{1+x_1+x_2}\sigma\right)} \quad (5.7)$$

A obtenção das várias funções de transferência é conseguida fazendo as variáveis auxiliares x_1 e x_2 percorrerem, com determinado passo, os intervalos onde estão definidas. Considerando um passo de 0.1 para o intervalo $[0 \ 1]$, consegue-se 121 concretizações possíveis a partir das variáveis x_1 e x_2 . Na Fig. 5.2, representam-se as evoluções de $F(\sigma)$ contra o padrão (x_1, x_2) para alguns valores de σ .

5.2.3 Determinação dos valores de σ

As medidas de identificação (5.7) estão definidas em termos dos dois parâmetros x_1 e x_2 ; assim, para cada função de transferência definem-se duas medidas de identificação que resultam da sua concretização para dois diferentes valores de σ . Dado haver uma forte correlação entre as duas medidas de identificação, pretende-se seleccionar dois valores de σ para os quais os correspondentes vectores $F(\sigma)$ apresentem o máximo grau de independência linear.

Seja a matriz N de 121 linhas¹: $N = [F(\sigma_1) \dots F(\sigma_8)]$, $\sigma_i \in \{0.1, 0.5, 1, 2, 5, 10, 15, 20\}$.

Normalizando cada linha de tal forma que a média seja nula e o desvio padrão seja unitário, obtém-se: $N_n = [F_n(\sigma_1) \dots F_n(\sigma_8)]$. Tomando as colunas duas a duas, obtém-se

matrizes de 121×2 , do tipo: $M_{ij} = [F_n(\sigma_i) \ F_n(\sigma_j)]$, $i=2 : 8$, $j=1 : i-1$, num total de

$\frac{8^2 - 8}{2} = 28$ matrizes. Para cada matriz M_{ij} , existem dois valores singulares s_i a partir dos

quais se determina: $Q(\%) = 100 \frac{\min(s_i^2)}{\sum_{i=1}^2 s_i^2}$. Admitindo que dois vectores estão tão mais

próximo da ortogonalidade quanto maior fôr Q^\dagger , apresenta-se, na Tab. 5.1, os valores deste parâmetro para as 28 matrizes, com vista à escolha do valor mais elevado [137].

¹ Correspondente ao mesmo número de processos.

[†] Semelhante análise poderia ser feita recorrendo à determinação de

$$\cos \angle(F(\sigma_i), F(\sigma_j)) = \frac{F(\sigma_i) \cdot F(\sigma_j)}{\|F(\sigma_i)\| \|F(\sigma_j)\|}$$

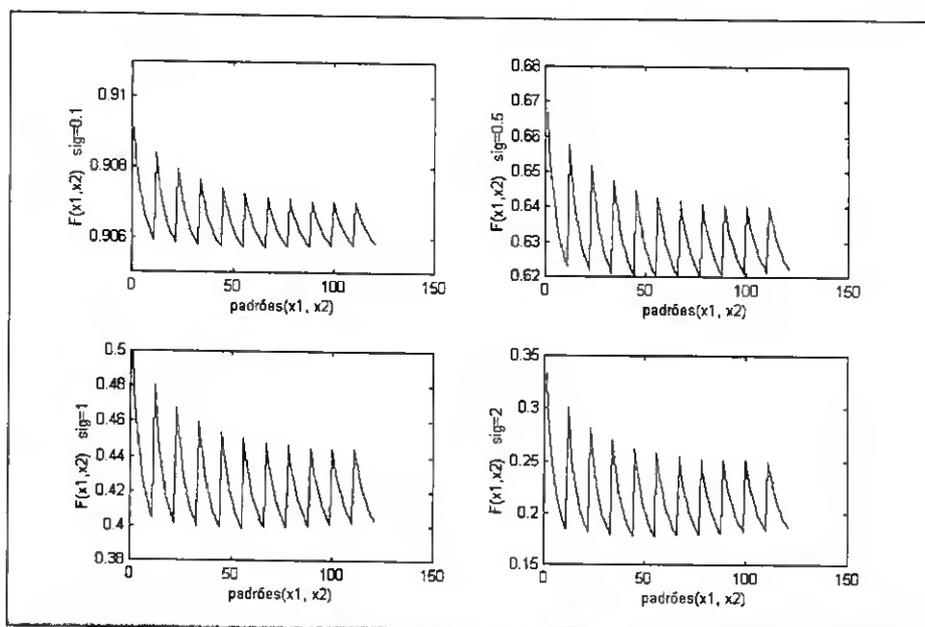


Fig. 5.2 – Evolução das medidas de identificação para $\sigma \in \{0.1, 0.5, 1, 2\}$.

σ	0.1	0.5	1	2	5	10	15	20
0.1		0.001	0.021	0.060	0.010	0.010	0.028	0.039
0.5			0.042	0.196	0.030	0.008	0.033	0.049
1				1.307	0.180	0.028	0.009	0.006
2					0.296	0.367	0.404	0.421
5						0.075	0.135	0.167
10							0.010	0.020
15								0.002
20								

Tab. 5.1 – Cociente dos valores singulares para as medidas de identificação não rectificadas.

A leitura da Tab. 5.1 permite concluir que o valor mais elevado é $Q=1.307\%$, verificado para $\sigma_i = 1, \sigma_j = 2$.

5.2.3.1 Linearização das medidas de identificação

Considere-se a seguinte hipótese:

O andamento linearizado por troços das medidas de identificação, (Fig. 5.2) conduz a valores mais elevados de Q .

Observando a expressão (5.7), ou a correspondente Fig. 5.2, verifica-se que:

- os máximos ocorrem para $x_2=0$ e x_1 varia de 0 a 1;
- os mínimos ocorrem para $x_2=1$ e x_1 varia de 0 a 1;
- entre um máximo e o mínimo consecutivo, x_1 mantém-se constante e x_2 varia de 0 a 1.

A rectificação dos gráficos foi feita em duas etapas que visam a alteração das discretizações de x_1 e x_2 de modo que:

1. O andamento monótono decrescente dos máximos seja linear, impondo a discretização de x_1 . Sabendo-se que os máximos ocorrem para $x_2=0$, a partir da expressão (5.7) obtém-se, para esse caso:

$$F(\sigma) = \frac{1}{\left(1 + \frac{1}{1+x_1}\sigma\right)\left(1 + \frac{x_1}{1+x_1}\sigma\right)} \quad (5.8)$$

Considerando σ como um parâmetro na expressão (5.8), o problema em estudo resume-se à determinação dos pontos x_1^\dagger para os quais $F(\sigma)$ vem equiespaçado entre $F_{\max}(\sigma)$ e $F_{\min}(\sigma)$. Tendo em conta as definições apresentadas em (5.6), verifica-se que os dois primeiros valores de x_1 serão forçosamente 0 e 0.1; assim, os limites a considerar são, mais precisamente, $F_{\max}(\sigma) = F(\sigma)|_{x_1=0.1}$ e $F_{\min}(\sigma) = F(\sigma)|_{x_1=1}$. Os 10 pontos $F_i(\sigma)$ vão estar equiespaçados entre esses dois limites, sendo os correspondentes valores de x_{1i} obtidos por inversão de (5.8) em ordem a x_1 . Numericamente esta inversão foi feita recorrendo-se a $G(x_1)$ definida a partir de (5.8):

$$G(x_1) = F(\sigma) - \frac{1}{\left(1 + \frac{1}{1+x_1}\sigma\right)\left(1 + \frac{x_1}{1+x_1}\sigma\right)} \quad (5.9)$$

[†] medidas de identificação diferentes (diferentes σ), conduzem a diferentes sequências de x_1 .

Para cada $F_i(\sigma)$, x_{1i} será a solução de $G(x_1) = 0$.

2. Cada segmento não linear entre um máximo e o mínimo consecutivo será linearizado, impondo a discretização de x_2 . Tal como no ponto anterior, a determinação da sequência de pontos x_2 vai ser feita tendo em vista a rectificação de cada arco nas representações gráficas da Fig. 5.2 para um valor de σ fixo, que funciona como parâmetro. Desta forma, a rectificação de cada sub-gráfico origina 11 sequências diferentes² de valores de x_2 que irão constituir uma matriz 11×11 .

Considere-se um arco compreendido entre um máximo e o mínimo seguinte. Ao longo do andamento monótono decrescente, verifica-se que x_1 é constante³ e que x_2 varia entre 0 e 1.

Considerando na expressão (5.7) σ um parâmetro, para cada valor de x_1 , pretende-se determinar uma sequência de pontos x_2 para os quais os correspondentes valores de $F(\sigma)$ venham equiespaçados entre $F_{\max}(\sigma)$ e $F_{\min}(\sigma)$. Para a situação em estudo, tem-se $F_{\max}(\sigma) = F(\sigma)|_{x_2=0, x_1=x_{1i}}$ e $F_{\min}(\sigma) = F(\sigma)|_{x_2=1, x_1=x_{1i}}$, ficando os 11 pontos $F_i(\sigma)$ equiespaçados entre esses dois limites.

Analogamente ao que foi feito na secção anterior, os pontos x_2 serão determinados com recurso à função auxiliar $D(x_2)$ definida por:

$$D(x_2) = F(\sigma) - \frac{e^{\frac{x_2}{1+x_1+x_2}\sigma}}{\left(1 + \frac{1}{1+x_1+x_2}\sigma\right)\left(1 + \frac{x_1}{1+x_1+x_2}\sigma\right)} \quad (5.10)$$

Fixando x_{1i} , para cada ponto $F_i(\sigma)$ conhecido, o correspondente valor de x_{2i} será a raiz da equação $D(x_2) = 0$ para um dado σ .

Fazendo o parâmetro $\sigma = 0.1$, na Fig. 5.3 apresenta-se:

² correspondentes aos 11 arcos e igual número de valores de x_1 .

³ previamente determinado na secção anterior.

- Na metade superior, os pontos x_1 equiespaçados entre 0 e 1, sobrepostos com os pontos x_2 determinados de acordo com a estratégia apresentada.
- Na metade inferior, a média de cada uma das 11 colunas da matriz de pontos x_2 , sobrepostos à evolução equiespaçada da mesma variável.

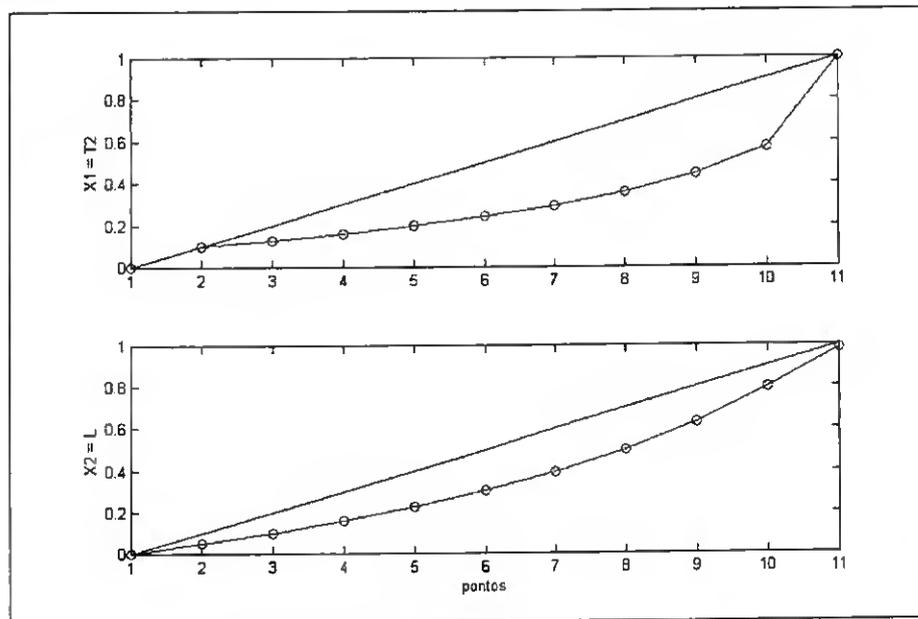


Fig. 5.3 – Discretizações de x_1 e x_2 com $\sigma = 0.1$.

Com esta estratégia há a salientar o seguinte:

- A evolução dos máximos, embora monótona decrescente para todos os casos, depende dos valores de σ ; assim, para cada medida de identificação vai ser definido um vector diferente de valores de x_1 .
- Para cada σ , cada troço não linear entre um máximo e o mínimo seguinte corresponde a x_1 constante, diferindo de troço para troço. Assim, para cada valor de x_1 existe um vector de pontos x_2 que linearizará apenas o troço não linear correspondente. Para cada σ , a reunião de todos⁴ os vectores de pontos x_2 conduz a uma matriz de pontos x_2 que possibilita o andamento rectificado em dente de serra para a medida de identificação σ correspondente.

⁴ 11 vectores pois consideram-se 11 discretizações para x_1 .

Observando estes pontos, conclui-se que, fixando um determinado conjunto de funções de transferência, não é possível obter andamentos rectificadas para todas as medidas de identificação.

Com o intuito de se determinar qual a rectificação que conduz ao valor mais elevado de Q , procedeu-se às rectificações dos andamentos das medidas de identificação para todos os valores de σ em estudo. Tomando os vectores das medidas de identificação $F(\sigma)$ assim obtidos, procedeu-se de forma análoga à que foi descrita na secção 5.2.3, obtendo-se tabelas análogas à Tab. 5.1. A observação de tais tabelas permite concluir que o valor mais elevado de Q surge sempre para $\sigma_i = 1$ e $\sigma_j = 2$. Na Tab. 5.2 registam-se os valores mais elevados de Q para os diferentes valores de σ , quando é feita a rectificação das medidas de identificação.

σ	$Q_{\max}(\sigma_i = 1, \sigma_j = 2) \%$
0.1	1.681
0.5	1.674
1	1.673
2	1.675
5	1.626
10	1.451
15	1.358
20	1.328

Tab. 5.2 – Máximos valores de Q para rectificações com diferentes valores de σ .

A observação desta tabela permite-nos verificar que o valor mais elevado de $Q_{\max} = 1.681\%$ ocorre quando a linearização é feita para $\sigma = 0.1$. Lembrando que de acordo com a Tab. 5.1 o valor mais elevado é $Q = 1.307$, pode-se concluir que é verdadeira a hipótese segundo a qual o andamento linearizado por troços conduz a valores mais elevados de Q .

Na Fig. 5.4 observam-se as evoluções de algumas medidas de identificação rectificadas para $\sigma = 0.1$, sendo os valores de x_1 e x_2 (Fig. 5.3) não equiespaçados.

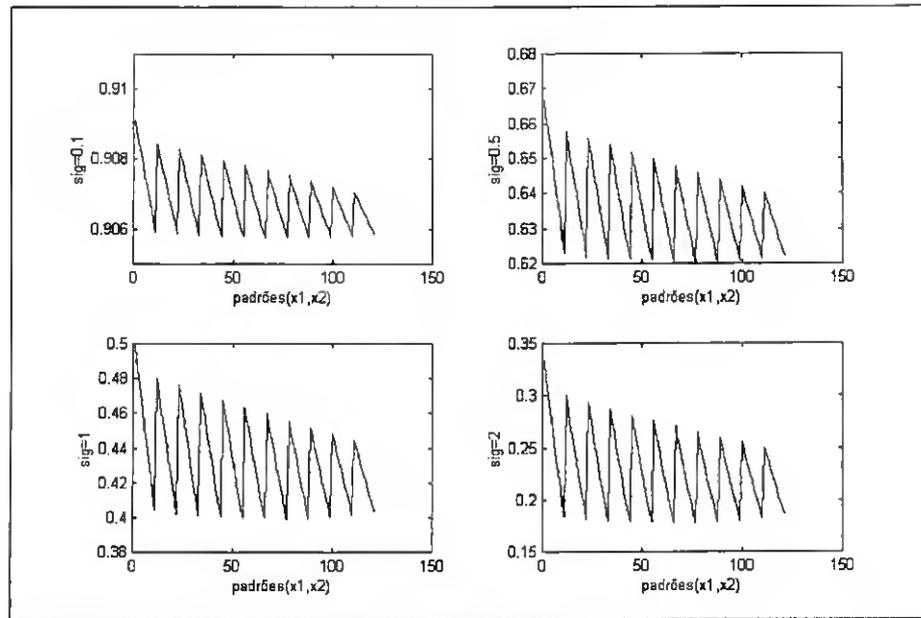


Fig. 5.4 – Evolução das medidas de identificação rectificadas para $\sigma = 0.1$ ($\sigma \in \{0.1, 0.5, 1, 2\}$).

5.2.4 Distribuição no espaço das medidas de identificação

A escolha dos valores de σ para os quais se avaliam as medidas de identificação, bem como a escolha das discretizações das funções de transferência, é ser feita com vista a uma cobertura mais ou menos homogênea do espaço de trabalho, dentro dos limites considerados de interesse, face aos sistemas que se decidiu contemplar [137]. Nesta secção mostra-se como varia a homogeneidade da cobertura do espaço de trabalho, quando se passa da situação em que as medidas de identificação têm um andamento predominantemente não linear para um andamento rectificado. Por último, vão ser vistas as alterações provocadas, quando os dados são transformados de acordo com uma transformação linear proposta [137].

Na Fig. 5.5 sobrepõem-se os pontos $(F(1), F(2))$ para as 121 funções de transferência que originam as medidas de identificação com andamento não linear [137], com os pontos análogos $(F(1), F(2))$ resultantes das funções de transferência cujos andamentos das medidas de identificação já vêm rectificadas.

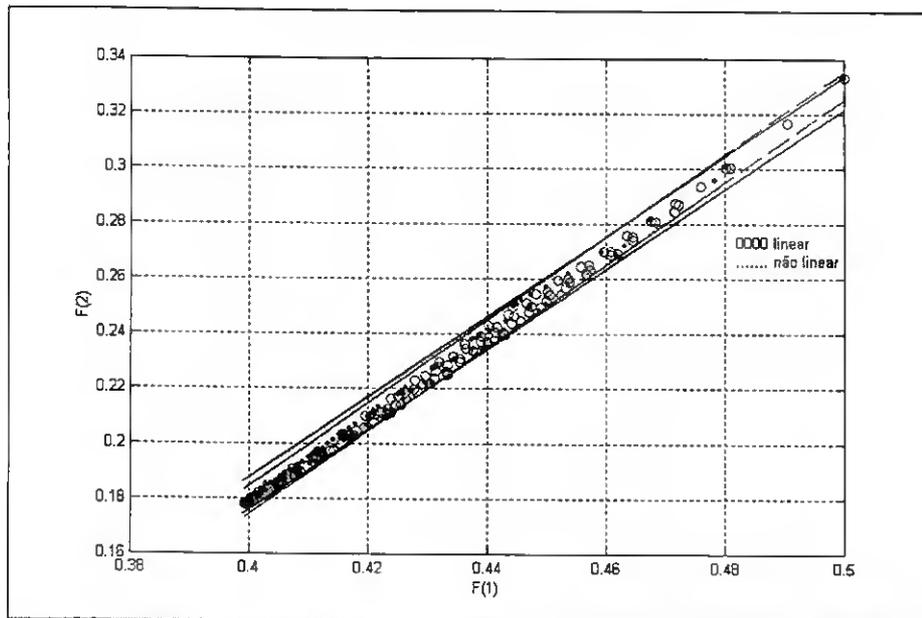


Fig. 5.5 – $(F(1), F(2))$ para as funções de transferência linearizadas e não linearizadas.

Para se ter ideia da deficiente cobertura do espaço, procede-se à divisão de cada eixo ($F(1)$ e $F(2)$) em 5 intervalos iguais originando uma grelha de 25 células. Na Tab. 5.3[‡] apresenta-se o número de pontos de cada célula, tanto para a situação rectificada como para a situação não rectificada.

Com vista a uma melhor distribuição dos pontos, foi proposta [137] uma transformação linear que conduziu às distribuições que se apresentam na Fig. 5.6. O método proposto visa a transformação do espaço de entrada de modo a cobrir o mais possível todo o intervalo de trabalho. Esta transformação é composta por 2 etapas:

- determinação de duas rectas paralelas que englobem todos os dados,
- rotação das rectas de modo a que a variação máxima dos dados segundo o eixo dos yy , seja igual à distância entre as duas rectas, ou seja, pretende-se colocar as rectas paralelas ao eixo dos xx .

[‡] as células vazias correspondem a 0 pontos.

$F(2)$	5									1	2
	4							3	9	1	2
	3					15	27	1	3		
	2			36	35	1	3				
	1	62	39	1	1						
		1	2		3		4		5		
$F(1)$											

Tab. 5.3 – Distribuição das medidas de identificação: não rectificadas | rectificadas.

Para se formalizar este método, considere-se $X (m \times n)$ a matriz de dados com m padrões e n variáveis. As variáveis $x_i, i > 1$ são rodadas de tal forma que os hiperplanos paralelos à distância mínima que englobam os dados, sejam paralelos à dimensão i . Os dados rodados \bar{x}_i podem ser determinados pela expressão (5.11)

$$\bar{x}_i = x_i - \sum_{j=1}^{i-1} w_j x_j, \quad i > 1 \tag{5.11}$$

Na 2ª etapa, recorrendo-se às expressões (5.12), os dados são escalados de forma a cobrirem a gama $[-1 \ 1]$.

$$\hat{x}_i = \bar{x}_i k_i + \sigma_i, \quad k_i = \frac{2}{x_{i_{\max}} - x_{i_{\min}}}, \quad \sigma_i = -1 - x_{i_{\min}}, \quad i \in [1 \ n] \tag{5.12}$$

Conclui-se assim que a transformação proposta fica definida por 3 parâmetros: k, σ e w . Aplicada esta transformação aos pontos que se apresentam na Fig. 5.5, obtém-se a distribuição referente à Fig. 5.6. A distribuição mais uniforme dos pontos desta figura, face à que se observa na Fig. 5.5 é confirmada numericamente pela Tab. 5.4†, obtida de forma análoga à Tab. 5.3.

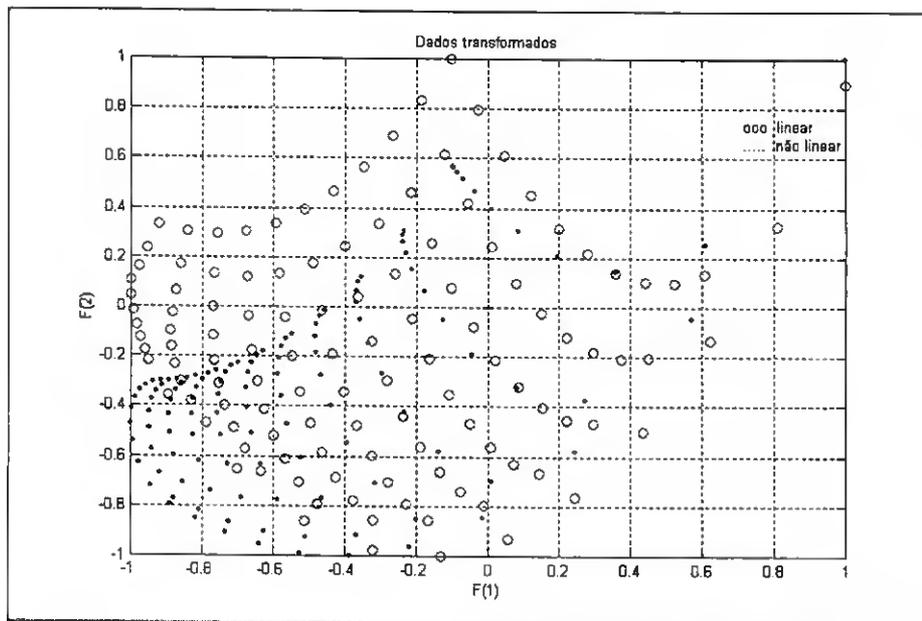


Fig. 5.6 – $(F(1), F(2))$ para dados transformados.

F(2)	5				1		5			1	1
	4		5	4	7	8	5		1	1	1
	3	2	18	15	9	3	4	2	5		2
	2	43	14	9	9	2	8	2	5		
	1	17	2	9	10	3	8		1		
		1	2	3	4	5					
F(1)											

Tab. 5.4 – Dados transformados: distribuição das medidas de identificação: não rectificadas | rectificadas.

A observação da Fig. 5.6 e da Tab. 5.4 permite igualmente verificar que a transformação apresentada na secção 5.2.4, quando aplicada às medidas de identificação rectificadas, conduz a uma melhor distribuição no plano do que aquela que se observa quando a transformação é aplicada às medidas de identificação não rectificadas.

Com as estratégias implementadas nas secções anteriores, define-se o conjunto base de experimentação formado por:

110 processos de 1ª e 2ª ordem com tempo de atraso, sem zeros e com ganho estático unitário.

Repare-se que os 110 processos aqui mencionados resultam dos 121 processos iniciais aos quais foram retirados os 11 com tempo de atraso nulo⁵.

A definição de cada um destes sistemas é feita em termos dos 3 parâmetros: T_1 , T_2 , L . A constante de tempo T_1 é mantida unitária para todas as funções de transferência; a constante de tempo T_2 é definida por um vector (1×11) e o tempo de atraso L é definido numa matriz (11×11). Os valores discretizados (não equiespaçados) destas variáveis estão definidos nos intervalos que se apresentam nas expressões (5.13).

$$L \in]0.04 \quad 0.99[\cup \{1\}, \quad T_2 \in [0.1 \quad 0.59[\cup \{0, 1\} \quad (5.13)$$

5.3 PADRÕES DE APRENDIZAGEM

Na secção anterior apresentou-se uma estratégia de selecção de funções de transferência que constituem o conjunto base de experimentação. A presente secção visa o estabelecimento de conjuntos de vectores *PID* que, concatenados com as medidas de identificação, venham a constituir os elementos dos conjuntos de aprendizagem para os mapeamentos em estudo.

Para cada elemento do conjunto base de experimentação são considerados dois tipos de vectores *PID*:

- *PID* minimizantes dos critérios de desempenho em estudo,
- *PID* representativos, obtidos a partir da base de *PID* minimizantes que discretizam o espaço de aprendizagem dentro do domínio de interesse.

⁵ Estes 11 processos de 1ª e 2ª ordem quando sujeitos ao controlo PID iriam apresentar erro nulo para ganho proporcional infinito, conduzindo a alguns critérios com o mínimo nulo.

5.3.1 Parâmetros *PID* mínimos

Nesta secção vai ser considerada a optimização de dois dos critérios estabelecidos em [137]: *ITAE* e tempo de crescimento. Dada a semelhança dos procedimentos de optimização para estes dois critérios, vai ser dado destaque apenas ao *ITAE*.

A determinação de cada mínimo é feita em duas etapas:

- Na 1ª etapa pretende-se determinar uma aproximação à solução. Desta forma, procedeu-se à minimização do critério para cada um dos 110 sistemas. Este procedimento de optimização não restringida utiliza o método *simplex* de Nelder-Mead (procura directa), recorrendo à implementação disponível em MATLAB [134] na rotina *fminsearch*.
- Na 2ª etapa, utilizando a solução a que se chegou na 1ª etapa, é feita uma discretização do espaço de soluções na sua vizinhança e procurado o mínimo para os pontos discretizados.

Minimização - 1ª Etapa

O ponto inicial para a minimização é determinado de acordo com os seguintes métodos:

- Regras de Ziegler e Nichols: O *PID* inicial será determinado por estas regras.
- Utilização do mínimo encontrado para o sistema anterior: Dada a regularidade que esteve por base na geração das funções de transferência em estudo (secções 5.2.2 e 5.2.3), será de esperar que os mínimos de duas funções de transferência consecutivas não se apresentem muito diferentes. Assim, o *PID* minimizante do processo será tomado como *PID* inicial do processo seguinte.
- Utilização do mínimo do sistema mais próximo: As duas medidas de identificação de cada função de transferência determinam um ponto no espaço bidimensional. Assim, para cada processo é considerado como *PID* inicial o *PID* minimizante do processo mais próximo de entre os que já foram minimizados.
- Utilização do valor da passagem anterior: O processo de supervisão que assistiu a esta etapa de optimização permitiu verificar que não estava concluída, mesmo após a execução com sucesso dos 110 ciclos de optimização das funções de transferência

respectivas. Nestas circunstâncias, os lotes de 110 ciclos de otimização executados a seguir ao primeiro terão a possibilidade de usar como *PID* inicial o valor encontrado para o respectivo sistema na execução do lote anterior.

Atendendo à regularidade da sequência das 110 funções de transferência, seria de esperar uma representação gráfica de cada componente *PID* ou do *ITAE*, com uma evolução igualmente regular [137]. Verificou-se, no entanto, que o padrão regular não se verificava em grande parte dos casos, havendo necessidade de se prosseguir com a 2ª etapa.

Mínimização - 2ª Etapa

Nesta etapa é feita uma pesquisa mais fina em torno do "ótimo" obtido na 1ª Etapa. Para cada sistema foi determinado um conjunto adicional de vectores *PID*, cujas componentes resultam das correspondentes "ótimas" aplicando-se uma determinada percentagem de perturbação. Para cada *PID* do conjunto adicional assim formado, é determinado o correspondente valor do critério. Para cada conjunto de valores do critério é determinado o mínimo; se este for inferior ao "ótimo" inicial, substituí-se.

Optimização do Tempo de Crescimento

Os procedimentos levados a cabo para minimizar o tempo de crescimento são em tudo análogos aqueles que foram apresentados para o *ITAE*; sendo neste caso, a função minimizar dada por (5.14).

$$TC = t_{90} - t_{10}, \text{ sendo } y(t_k) = \frac{k}{100} y_{final} \quad (5.14)$$

Nas situações para as quais se verifica oscilação em torno dos 10% ou dos 90% do valor final da resposta, no cálculo do tempo de crescimento, consideram-se os instantes de tempo para os quais tais valores foram atingidos pela primeira vez. Recordar-se que, tal como acontece para o *ITAE*, no cálculo do tempo de crescimento apenas se consideram os sistemas estáveis com tempo de estabelecimento aceitáveis.

Verifica-se que os sistemas com bons tempos de crescimento se apresentam, por vezes, excessivamente oscilatórios; como tal, foi necessário definir um "limiar de oscilação". Assim, só se consideram sistemas aceitáveis aqueles que, adicionalmente, apresentem uma

razão entre o primeiro e o segundo máximo superior à que se verifica quando o mesmo sistema é sintonizado pelas regras de Ziegler e Nichols.

5.3.2 Parâmetros *PID* representativos do espaço de aprendizagem

Para além dos *PID* óptimos descritos anteriormente, são gerados adicionalmente cerca de 30 vectores *PID* por processo. A geração destes *PID* adicionais foi feita tomando por base os 3 pontos óptimos para o *ITAE*, tempo de crescimento e sobrelevação respectivamente.

A determinação de *PID* adicionais consiste num problema de geometria no espaço tridimensional cujas dimensões são k_c , t_i e t_d .

Foram considerados dois tipos de distribuições espaciais: prismática hexagonal recta e esférica.

5.3.2.1 Distribuição prismática hexagonal recta

Verificada a não colinearidade dos 3 óptimos, O_1 , O_2 e O_3 , serão geradas duas categorias de novos pontos complanares (Fig. 5.7):

- pontos internos - 3 pontos médios dos 3 lados do triângulo (I_1 , I_2 e I_3 da Fig. 5.7).
- pontos externos - para cada vértice, são considerados os prolongamentos dos dois lados que o compõem. Para cada prolongamento, é determinado o ponto que dista do vértice o comprimento do lado que o origina: Ex: $E_4 = O_2 + \overline{O_1O_2} = O_2 + O_2 - O_1 = 2O_2 - O_1$.

Caso alguma componente de um ponto externo seja negativa, esse ponto será aproximado a meia distância do vértice que o gerou, percorrendo o prolongamento respectivo. O processo repete-se até que todas as componentes do ponto em causa sejam positivas.

Cada um dos 12 pontos até agora obtidos é projectado em dois planos paralelos ao plano dos óptimos (um para cada lado), a uma distância igual ao máximo dos comprimentos dos 3 lados do triângulo Fig. 5.8. Se aparecerem pontos com componentes negativas, o plano correspondente será sucessivamente aproximado a meia distância do plano dos óptimos, até que todos os pontos tenham todas as componentes positivas.

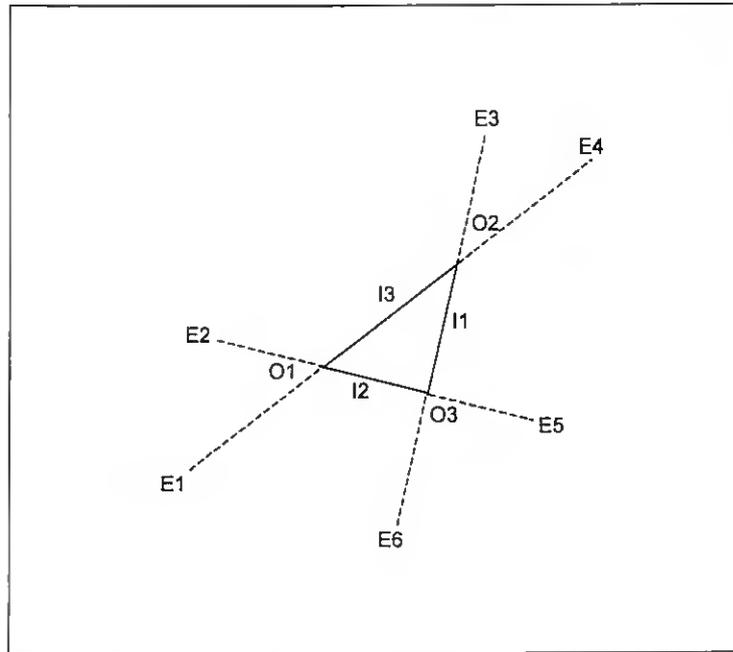


Fig. 5.7 – Plano dos ótimos, pontos internos e externos.

Terminado este processo, obtém-se uma estrutura prismática hexagonal recta Fig. 5.8, acomodando 36 pontos distribuídos por 3 planos paralelos, estando os ótimos para os 3 critérios localizados no plano intermédio. Fica garantido que todas as componentes são maiores que zero, faltando no entanto ainda garantir que todos os pontos correspondem a *PID* estáveis.

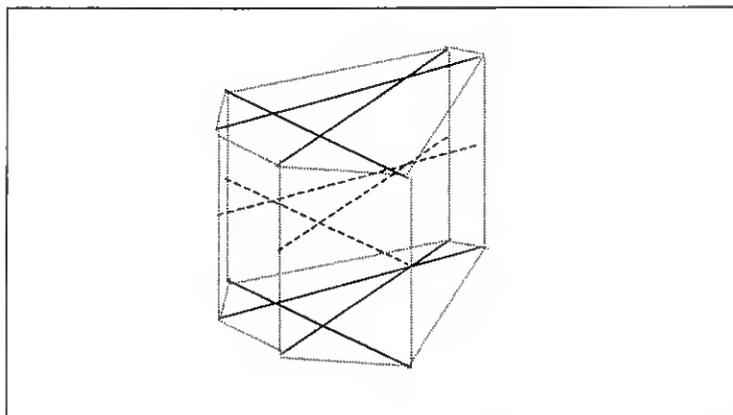


Fig. 5.8 – Estrutura prismática hexagonal recta com plano dos ótimos a meia altura.

5.3.2.2 Distribuição esférica

Para cada sistema é determinado um centro e um raio tomando-se como base os 3 valores ótimos dos critérios. Os pontos que irão constituir o conjunto de aprendizagem distribuir-se-ão regularmente pela superfície esférica e por radiais.

Determinação de centros e raios: A partir do triângulo formado pelos 3 óptimos definem-se 3 centros, correspondentes à:

1. Intersecção das mediatrizes dos 3 lados .
2. Intersecção das semi-rectas tiradas de cada vértice para o ponto médio do lado oposto.
3. Intersecção das 3 alturas (perpendiculares tiradas de cada vértice para o lado oposto).

Numa primeira fase o centro foi determinado como: $C = \frac{\sum_{i=1}^3 PID_i}{3}$, sendo PID_i o ponto minimizante do critério i ; posteriormente, foram testados os pontos 1 e 2.

Observando os óptimos dos critérios em estudo para todos os sistemas, verifica-se, dum modo geral, grande proximidade dos minimizantes do *ITAE* e da Sobreelevação, ficando o minimizante do tempo de crescimento afastado destes dois. Verificou-se que a técnica do ponto 1 define uma área que tende a afastar-se do espaço coberto pelos óptimos. Desta forma, surgiu a necessidade de se implementar e testar o posicionamento do centro de acordo com o ponto 2, tendo o raio R comprimento igual à distância do centro ao vértice mais afastado.

A discretização da esfera assim obtida é feita de acordo com os pontos seguintes:

- Seja N o número de pontos situados sobre cada circunferência pertencente à superfície esférica.
- Define-se uma circunferência paralela ao plano $t_i Ot_d$ centrada em C e de raio R (meridiano, Fig. 5.9).
- Essa circunferência é dividida em N arcos de igual amplitude ficando definida a posição de N pontos.
- Para cada ponto assim determinado, define-se uma circunferência perpendicular à inicial, ou seja, paralela ao plano $k_c Ot_i$ (paralelo, Fig. 5.9), na qual serão posicionados pontos que a dividem em N arcos de igual amplitude.

Discretização do interior da esfera:

- De cada ponto definido na superfície esférica é tirado um raio;
- Sobre esse raio são definidos equiespaçadamente $N/2$ ou $(N-1)/2$ pontos, consoante N seja par ou ímpar.

Desta forma, para um dado N , o número de pontos obtidos é o seguinte:

PONTOS	N par	N ímpar
Superfície Esférica	$\frac{(N-2)^2}{2} + N$	$\frac{N^2 + 1}{2}$
Interior	$\left(\frac{N}{2} - 1\right)\left(\frac{N^2}{2} - N + 2\right)$	$\frac{(N^2 + 1)(N - 3)}{4}$
Totais	$\left(\frac{N^2}{2} - N + 2\right)\frac{N}{2}$	$\frac{(N^2 + 1)(N - 1)}{4}$

Tab. 5.5 – Discretização duma esfera para um dado N .

Na aplicação em estudo utiliza-se $N=5$, correspondendo a uma totalidade de 26 pontos. Adicionando-se os 3 ótimos e o centro, obtêm-se 30 padrões de aprendizagem por processo. Na Fig. 5.9 exemplifica-se o resultado desta técnica para o processo do conjunto base de experimentação com $T_1=1$, $T_2=0.199$ e $L=0.304$.

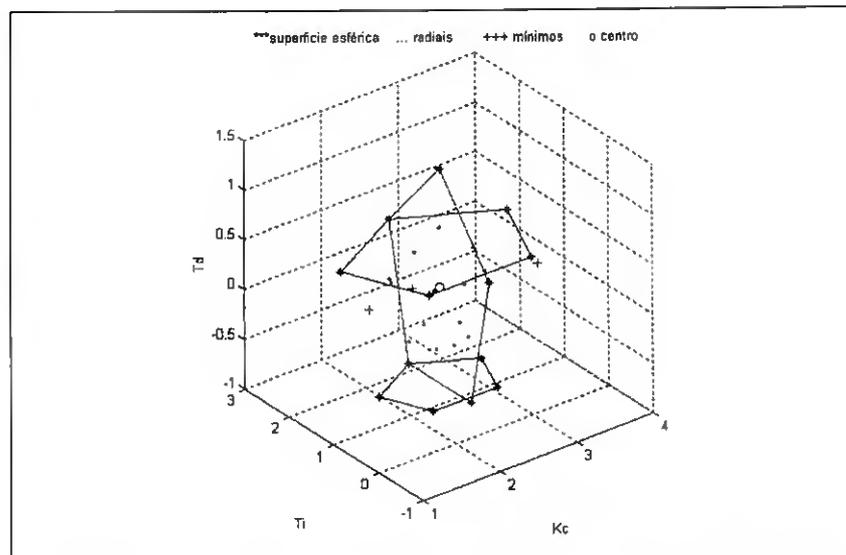


Fig. 5.9 – Discretização para um exemplo do conjunto base de experimentação fazendo-se $N=5$.

De acordo com a estratégia acabada de descrever, os vectores *PID* encontrados poderão apresentar componentes negativas (ver Fig. 5.9); esses vectores deverão ser deslocados para o 1º octante do espaço tridimensional. O método adoptado tem como objectivo a inclusão de pontos no volume inicialmente definido, nas zonas de menor densidade.

Para se ter uma noção da compactação provocada pelo deslocamento de pontos para o 1º octante, utiliza-se o cálculo do coeficiente c_v dado por (5.15).

$$C_v = \frac{dv_i}{dv_f} \quad (5.15)$$

sendo:

dv_i – distância entre os dois pontos mais próximos antes da transformação;

dv_f – distância entre os dois pontos mais próximos depois da transformação.

Para cada sistema, o procedimento utilizado para a migração de pontos utiliza o seguinte algoritmo:

-
- As linhas com todas as componentes positivas retiradas da matriz *PID* inicial irão constituir uma sub-matriz denominada sub-matriz positiva.
 - Para cada linha com alguma componente negativa são executadas as operações A e B.
 - Operação A:
 - Substitui-se por um ponto P_m que é a média dos pontos da matriz positiva.
 - Determina-se a distância d_1 desse ponto ao ponto mais próximo de entre os que compõem a matriz positiva.
 - Operação B:
 - Determina-se as distâncias entre todos os pontos da matriz positiva.
 - Para cada ponto, determina-se o vizinho mais próximo e correspondente distância.
 - Seja P_x o ponto para o qual é máxima a distância dx ao vizinho mais próximo.
 - Seja P_y o ponto mais afastado de P_x .
 - O novo ponto P_z será dado por: $P_z = P_x + \frac{dx (P_y - P_x)}{2 \|P_y - P_x\|}$ isto é, situa-se sobre o segmento $P_x P_y$, à distancia $dx/2$ de P_x .
 - determina-se a distância d_2 de P_z ao ponto mais próximo de entre os que compõem a matriz positiva.

- Se $d1 > d2$ a operação A corresponde a um menor adensamento e, como tal, inclui-se Pm na matriz positiva.
- Caso contrário, a operação B corresponde a um menor adensamento e como tal inclui-se Pz na matriz positiva.

Terminado o processo de posicionamento de pontos no primeiro octante, volta-se a determinar a distância entre os dois pontos mais próximos (d_{ij}). Utilizando a expressão (5.15) fica-se com uma medida da concentração os pontos devido à operação descrita.

O resultado do algoritmo acabado de descrever quando aplicado à distribuição da Fig. 5.9, apresenta-se na Fig. 5.10.

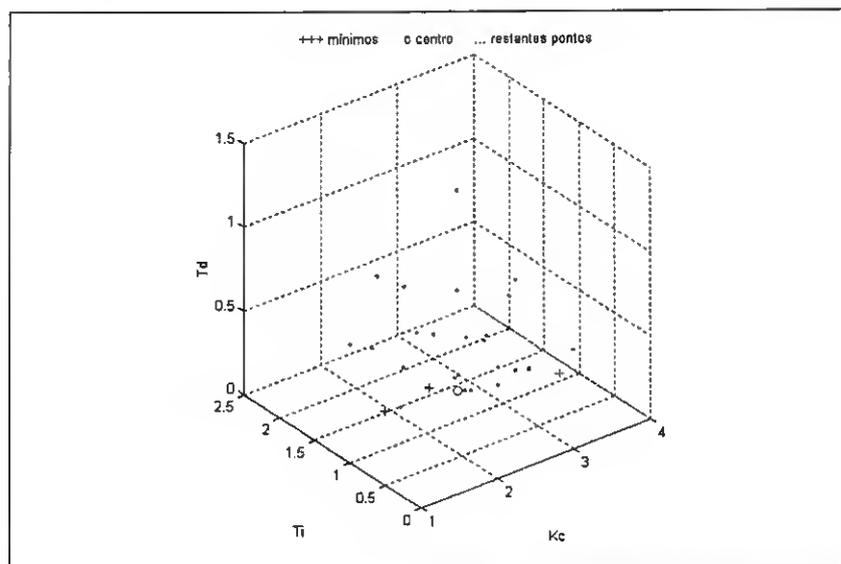


Fig. 5.10 – Distribuição de pontos após migração para o 1º octante.

Para que os pontos obtidos de acordo com esta estratégia possam ser utilizados na aprendizagem das redes neuronais, deverão corresponder a *PID* que conduzam a sistemas estáveis.

Aplicando o referido algoritmo aos 110 processos do conjunto base de experimentação com 30 *PID* por processo, verificou-se que existiam 35.8% de sistemas instáveis.

Tal como aconteceu na secção anterior, a deslocação dos *PID* instáveis deverá ser feita para zonas ainda dentro da esfera, de preferência com baixa representatividade. O algoritmo de recolocação dos pontos instáveis deverá conduzir a valores baixos de (5.15).

Foram testados 3 critérios de recolocação de pontos instáveis, de acordo com a Tab. 5.6.

1. Aproximação de cada ponto instável a meia distância do ponto estável mais próximo.
2. Aproximação de cada ponto instável a meia distância do ponto a que corresponde menor *ITAE*. Para cada sistema, um ponto que seja considerado de menor *ITAE* só será considerado para alvo de aproximação uma vez.
3. Aproximação de cada ponto instável a meia distância do ponto estável mais afastado.

Tab. 5.6 – Critérios para estabilização de pontos.

Tomando os 110 sistemas do conjunto base de experimentação, na Tab. 5.7 são indicados os valores máximo, mínimo e média do cociente (5.15), para os 3 métodos apresentados.

Tendo por base a Tab. 5.7, para cada sistema decidiu-se adoptar o 1º ou 3º critérios consoante aquele que conduza a menor C_v , obtendo-se os valores da Tab. 5.8.

	1	2	3
mínimo	1.8	1	1
máximo	7.6×10^5	3.2×10^{15}	480
médio	4.4×10^4	8×10^{13}	9.5

Tab. 5.7 – Cociente de aproximação.

	1º e 3º
mínimo	1
máximo	480
médio	8.7

Tab. 5.8 – Cociente de aproximação tomando o 1º e 3º critérios.

5.3.3 Estabelecimento dos conjuntos de aprendizagem

Com os procedimentos levados a cabo nas secções anteriores, obtiveram-se para os 110 processos que constituem o conjunto base de experimentação:

30 vectores *PID* por processo, correspondentes a sistemas estáveis em malha fechada, incluindo os vectores *PID* óptimos de 3 critérios, definidos no interior e fronteira das correspondentes superfícies esféricas, com centros e raios determinados de acordo com os procedimentos apresentados.

Na presente secção mostra-se como os padrões obtidos até aqui vão ser normalizados e transformados, para posteriormente, serem aleatoriamente seleccionados os conjuntos de treino e teste de redes neuronais.

O processo de aprendizagem depende da normalização operada sobre os padrões de entrada. Para se ter uma ideia de qual é a normalização que conduz a um processo de aprendizagem mais eficiente, fez-se uma análise prévia do mapeamento linear do espaço de entrada em cada critério que vai constituir as saídas, *ITAE* e tempo de crescimento (*TC*).

O problema é formulado em termos da resolução, em ordem ao vector w , do sistema $A \cdot w = b$.

$$A \text{ é uma matriz do tipo } (n \times 6): A = \begin{bmatrix} 1 & F(1) & F(2) & K_c^* & T_i^* & T_d^* \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & F(1) & F(2) & K_c^* & T_i^* & T_d^* \end{bmatrix}.$$

n é a totalidade de padrões que constituem o espaço de entrada das redes neuronais, considerando os 110 processos com os respectivos conjuntos de vectores *PID*.

b é sucessivamente, os correspondentes vectores *ITAE* e tempo de crescimento, originando dois tipos de sistemas distintos.

Os parâmetros *PID* que figuram na matriz A são sujeitos a 4 normalizações⁶ diferentes como se apresenta na Tab. 5.9. Atendendo a que se pretende mapear para dois critérios, vão ser resolvidos 8 sistemas do tipo $A \cdot w = b$.

Para se decidir qual a normalização mais apropriada, para cada solução w encontrada é determinada a resposta do modelo linear e comparada com os valores objectivo. O valor do erro resultante servirá para determinar as medidas de desempenho, *MSE* e *MSRE*. Estas

⁶ inclui-se a normalização apresentada em (5.16). Após cada normalização, é feita a transformação linear apresentada na secção 5.2.4 e por último resolvido o sistema.

medidas, juntamente com $\|w\|$ e $\frac{\max(abs(w))}{\min(abs(w))}$, dão uma ideia da normalização mais

conveniente. Na Tab. 5.9, apresentam-se as medidas de desempenho para os dois mapeamentos lineares, *ITAE* e tempo de crescimento.

A observação da Tab. 5.9 permite verificar que a medida de desempenho com valor mais baixo depende da normalização adoptada e que diferentes normalizações conduzem a diferentes valores das medidas de desempenho. Neste contexto, a escolha da normalização dentro das 4 possibilidades apresentadas é irrelevante, pelo que se decidiu utilizar a 1ª realização.

		<i>MSE</i>	<i>MSRE</i>	$\ w\ $	Max/min
[1 F(1) F(2) k_c $T_i/(T_T k_c)$ T_d/T_T]	<i>ITAE</i>	5.27	677.21	6.14	3.92
	<i>TC</i>	0.02	0.68	2.72	109.40
[1 F(1) F(2) k_c T_i/T_T T_d/T_T]	<i>ITAE</i>	5.48	256.01	5.17	9.18
	<i>TC</i>	0.03	0.37	0.69	4.12
[1 F(1) F(2) $1/k_c$ $T_i/(T_T k_c)$ T_d/T_T]	<i>ITAE</i>	5.25	772.30	5.26	49.26
	<i>TC</i>	0.02	0.86	1.47	5.61
[1 F(1) F(2) $1/k_c$ T_i/T_T T_d/T_T]	<i>ITAE</i>	5.29	877.69	7.39	14.04
	<i>TC</i>	0.02	0.29	0.79	2.77

Tab. 5.9 – Aproximação linear para o *ITAE* e para o tempo de crescimento.

Considerando o factor das constantes de tempo T_T dado por (3.4), a normalização dos critérios, *ITAE*, tempo de crescimento e Sobrelevação é dada por (5.16).

$$ITAE^* = \frac{ITAE}{T_T^2}, \quad TC^* = \frac{TC}{T_T}, \quad S^* = S, \quad (5.16)$$

Vão ser realizados dois tipos de mapeamentos:

- $[F(1) \ F(2)] \rightarrow \{k_c^*, \ t_i^*, \ t_d^*\}$: Mapeamento de sintonia.

- $[F(1) \ F(2) \ k_c^* \ t_i^* \ t_d^*] \rightarrow C_i^*$: Mapeamento de critério.

5.3.3.1 Conjuntos de aprendizagem para o mapeamento no critério

As medidas de identificação de cada processo são concatenadas com os vectores *PID*, constituindo o universo de entrada para o mapeamento no respectivo critério; são treinadas tantas redes neuronais quantos os critérios. Os 30 vectores *PID* para cada um dos 110 processos do conjunto base de experimentação são normalizados de acordo com a secção 5.3.3 dando origem a 3300 padrões de treino + teste, organizados numa matriz do tipo 3300×5 : $[\text{matriz_iden}|\text{matriz_pid}]$; a esta matriz é aplicada a transformação apresentada na secção 5.2.4. Extraíndo-se aleatoriamente 30% dos padrões para teste, obtém-se, numa primeira fase, 2310 padrões de treino e 990 padrões de teste.

Os conjuntos de aprendizagem, assim obtidos, foram testados no mapeamento: $MI + PID \rightarrow ITAE$, recorrendo-se ao algoritmo de aprendizagem supervisionada *ASMOD* [138].

Resultados prévios desta fase de aprendizagem supervisionada permitiram verificar a tendência para um aumento sucessivo da complexidade das redes neuronais envolvidas. Este facto mostra a dificuldade apresentada pelo algoritmo *ASMOD* na construção dum modelo único para o *ITAE*, tendo em conta todo o espaço de aprendizagem então definido. Convém lembrar que, o espaço definido nesta fase para os mapeamentos nos critérios, inclui todos os processos do conjunto base de experimentação, tendo sido definidos para cada um apenas 30 vectores *PID*.

A captura de modelos para os vários processos dependerá da distribuição dos padrões de aprendizagem no espaço dos critérios (*ITAE*, *TC*), pelo que, para cada elemento do conjunto base de experimentação, foi analisada a representação gráfica da relação $TC \equiv f(ITAE)$. Para um elemento tipo do conjunto base de experimentação apresenta-se na Fig. 5.11 a relação $TC \equiv f(ITAE)$; nesta figura pode-se observar uma aglomeração principal de pontos onde se incluem os óptimos para os dois critérios, restando uma minoria de pontos que se afasta deste domínio. Assim, para que a captura dos modelos dos critérios de desempenho seja facilitada, decidiu-se suprimir dos conjuntos de aprendizagem os pontos que se afastam da aglomeração principal, de acordo com os limites apresentados na Fig. 5.11.

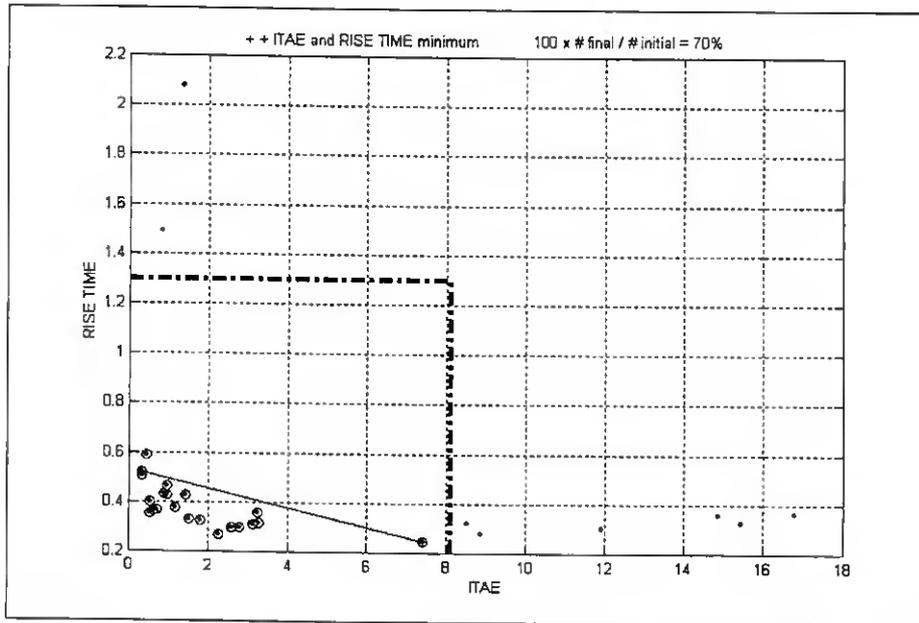


Fig. 5.11 – $TC=f(ITAE)$, para um processo com $k_p = 1$, $T_I=1$, $T_2=0.16$ e $L=0.64$.

Para se ter uma ideia da quantidade de pontos suprimidos por processo, apresenta-se nessa figura a percentagem de pontos restantes após a supressão, em relação à totalidade de pontos iniciais. Para cada elemento do conjunto base de experimentação o procedimento adoptado de supressão de pontos é, resumidamente, o seguinte:

- Considera-se o vector de padrões normalizados de acordo com (5.16), antes da transformação apresentada na secção 5.2.4., para o qual se procede à representação gráfica da relação $TC \equiv f(ITAE)$.
- Localizam-se os pontos: $P_1 = (ITAE, TC_{\min})$ e $P_2 = (ITAE_{\min}, TC)$, em geral não coincidentes e ligados, na Fig. 5.11, por um segmento de recta.
- Estabelecem-se limites em cada dimensão, $ITAE$ e TC , dados respectivamente por $L_1 = P_1(1) + \frac{\max(ITAE) - P_1(1)}{2}$ e $L_2 = P_2(2) + \frac{\max(TC) - P_2(2)}{2}$.
- Os pontos situados acima dos limites são suprimidos, determinando-se a percentagem de pontos restantes.
- Se a percentagem de pontos restantes for superior a 70% e se permanecerem ainda pontos que se afastam da aglomeração principal, então os limites são reduzidos de

forma a que os pontos restantes estejam todos na aglomeração principal, garantindo-se que P_1 e P_2 não são suprimidos.

Após a filtragem acabada de descrever, segue-se a transformação apresentada na secção 5.2.4 e por último são gerados os novos conjuntos de aprendizagem, cuja distribuição de padrões se apresenta na Tab. 5.10.

TREINO	TESTE
1840 padrões	788 padrões

Tab. 5.10 – Conjuntos de aprendizagem após filtragem.

5.4 CONCLUSÕES

Este capítulo faz a transição entre o capítulo anterior, onde se tratava o problema da selecção do tipo de rede neuronal mais adequado à modelação dos critérios de desempenho em estudo, e o capítulo seguinte onde, para o tipo de rede neuronal encontrado no capítulo anterior, se procurará modelos de critérios para processos variantes no tempo.

Neste enquadramento é necessário obter-se conjuntos de aprendizagem adequados à situação que se avizinha, tanto no que respeita ao âmbito de processos coberto pelo sistema de auto-sintonia, quanto ao espaço de parâmetros do controlador. Assim, neste capítulo foi determinado o âmbito das constantes de tempo para processos tipo, de tal forma que se tornasse viável uma identificação adequada de acordo com o método preconizado. Para além disso, foram estudadas diferentes distribuições no espaço para os parâmetros *PID* com vista a uma cobertura dentro do âmbito de interesse para uma sintonia optimizada.

Os conjuntos de aprendizagem obtidos conduzem em todas as situações a sistemas estáveis com tempos de estabelecimento aceitáveis, para além de originarem distribuições, no espaço dos critérios de desempenho, com alguma regularidade.

6. APRENDIZAGEM DE REDES NEURONAIS PARA PROCESSOS VARIANTES NO TEMPO

6.1 INTRODUÇÃO

No capítulo anterior foi apresentada uma discriminação de processos com o objectivo de se determinar um conjunto base de funções de transferência onde o sistema de auto-sintonia proposto é preferencialmente testado. Associado a cada processo foram determinados vectores *PID* que cobrem o espaço de parâmetros do controlador onde os procedimentos de optimização terão lugar.

O objectivo do presente capítulo é a obtenção de redes neuronais do tipo *B-Spline* treinadas, para mapeamentos definidos sobre o novo conjunto base de experimentação estabelecido no capítulo anterior. Para isso efectuou-se a aprendizagem de 2 tipos de redes neuronais:

1. redes neuronais de modelação de critérios de desempenho: modelam os mapeamentos nos critérios de desempenho *ITAE* (secção 6.2) e tempo de crescimento (secção 6.3),
2. redes neuronais de sintonia do controlador: modelam os mapeamentos nos parâmetros *PID* óptimos para os respectivos critérios de desempenho (secção 6.4).

No que diz respeito ao mapeamento do *ITAE*, foram comparadas diferentes estratégias de modelação. Na secção 6.2.1 estabelecem-se modelos distintos para cada um dos processos do conjunto base de experimentação. Dada a heterogeneidade de modelos que se obtêm com a abordagem levada a cabo nessa secção, é feita na secção 6.2.2 uma caracterização mais rigorosa da função $ITAE \equiv f(PID)$, nas vizinhanças do seu mínimo, para um

elemento representativo do conjunto base de experimentação. Nas secções 6.2.3 e 6.2.4 é apresentada uma estratégia alternativa de modelação do *ITAE* a partir da qual se construirão modelos de 2ª ordem nas vizinhanças do seu valor mínimo, com vista à obtenção de estruturas parcialmente fixas. As secções 6.2.5 a 6.2.9 debruçam-se sobre a modelação neuronal dos parâmetros dos modelos de 2ª ordem, seleccionando-se a estratégia de modelação mais conveniente; na secção 6.2.10 são validados os referidos modelos.

O capítulo termina com a secção 6.5 onde é feita uma análise comparativa dos resultados obtidos para o *ITAE* de acordo com a estratégia inicial e com aquela que se apresenta na secção 6.2.3. São apresentadas conclusões sobre a modelação do tempo de crescimento e sobre a modelação dos parâmetros *PID* de sintonia, optimizados para ambos os critérios de desempenho em estudo.

6.2 REDES NEURONAIS PARA MAPEAMENTO DO *ITAE*

A partir dos conjuntos a que se refere a Tab. 5.10, ocorreram simulações com vista à aprendizagem de redes neuronais do tipo *B-Spline* utilizando o *ASMOD*, para o mapeamento $MI + PID \rightarrow ITAE$. De entre os modelos obtidos, apresenta-se na Tab. 6.1 a estrutura daquele que globalmente conduz ao melhor desempenho e na Tab. 6.2 as respectivas medidas de desempenho.

Sub-Mod.	1	2	3	4	5	6	7	8	9
Var.	1	2	3	4	2 5	1 3	1 5	1 2	4 5
Nint.	3	3	1	3	2 0	1 2	0 1	0 1	2 0
Ord.	3	3	3	3	3 3	3 3	3 3	2 2	3 3

Tab. 6.1 – Modelo para o mapeamento $MI + PID \rightarrow ITAE$ utilizando o *ASMOD*.

<i>BIC</i>	Complexidade	<i>MSE</i> treino	<i>MSRE</i> treino
-2.7E3	89	5.6	0.2

Tab. 6.2 – Desempenho do modelo da Tab. 6.1.

6.2.1 Determinação de uma estrutura comum

Na secção anterior constatou-se que a obtenção de um modelo neuronal para o *ITAE*, segundo estratégias convencionais, apresenta grande peso computacional, conduzindo a resultados com valores de erro significativos para o conjunto de teste, evidenciando dificuldade de generalização.

Um modo de minorar a complexidade computacional deste problema seria determinar uma possível estrutura para o mapeamento $PID \rightarrow ITAE$, comum para os vários processos em causa, e fixar posteriormente essa estrutura no mapeamento global $[MI | PID] \rightarrow ITAE$.

Assim, nesta secção vão ser treinados 110 modelos (um para cada processo), e analisadas as respectivas constituições no que respeita a semelhanças estruturais.

Os padrões de aprendizagem utilizados nesta fase resultam do lote de padrões que se utilizaram na secção anterior, discriminados por sistema e após a supressão dos elementos a que se refere a secção 6.2.

Estes dados, normalizados de acordo com (5.16), foram utilizados sucessivamente em 3 configurações distintas:

1. Sem qualquer transformação linear
2. Transformados conforme o método apresentado secção 5.2.4;
3. Transformados linearmente de tal forma que os parâmetros *PID* pertençam ao intervalo $[-1 \ 1]$, de acordo com a secção 4.2.1.

Os modelos neuronais são avaliados de acordo com a percentagem de *MRE* definida pela expressão (4.8).

A evolução da aprendizagem dos 110 modelos para cada estado de transformação faz-se em várias etapas. A primeira etapa consiste na aprendizagem de cada um dos 110 processos, a partir dos respectivos modelos iniciais gerados aleatoriamente. Concluída a primeira etapa, é registado o modelo que conduziu ao menor valor da % *MRE*, sendo este utilizado como modelo inicial na etapa seguinte. O processo repete-se até que a média da % *MRE* avaliada para os 110 processos não decresça. Concluído este procedimento,

obtem-se 3 tabelas de modelos para os 110 processos, correspondentes às 3 configurações respeitantes às transformações apresentadas, totalizando 330 modelos.

Para o espaço de entrada nas 3 configurações referidas, são apresentados na Tab. 6.3 os valores mínimos, máximos e médios da % *MRE* para a iteração a partir da qual a média deste indicador já não decresce.

% <i>MRE</i>	Min.	Máx.	Média
1	0.2	49.4	13.8
2	0.8	58.8	18.4
3	3.9E-12	48.8	11.0

Tab. 6.3 – % de *MRE* para o espaço de entrada em cada uma das 3 configurações.

Tomando como exemplo um processo a meio do conjunto base de experimentação¹ caracterizado por $T_1=1$, $T_2=0.242$ e $L=0.306$, apresenta-se na Tab. 6.4 os modelos das redes neuronais obtidos para as 3 configurações referidas.

	Iter.	Nsm	Var.	Nint.	Ord.	Var.	Nint.	Ord.	Var.	Nint.	Ord.
1	4	3	1	2	3	2	1	3	3	0	3
2	2	3	1	0	3	2	1	3	3	0	3
3	1	3	1	0	3	2	0	2	3	12	3

Tab. 6.4 – Rede neuronal para o processo em estudo quando o espaço de entrada assume as 3 configurações.

Analisando os dados obtidos para todos os modelos e para as 3 configurações referidas, podem-se tirar as seguintes conclusões:

- Os modelos neuronais obtidos são maioritariamente constituídos por 3 sub-modelos univariáveis, observando-se no entanto, para alguns casos, modelos constituídos por 2 sub-modelos. Nalguns casos obtiveram-se 4 sub-modelos² dos quais um sub-modelo é

¹ Atendendo à ordenação que esteve na origem deste conjunto, o processo considerado corresponde à posição 55.

² Mais predominantemente para a situação em que os dados de entrada não sofrem qualquer alteração.

bivariável. Pontualmente temos situações em que o modelo obtido é composto apenas por um sub-modelo.

- Os sub-modelos encontrados são maioritariamente de 3^a ordem, observando-se contudo alguns de 2^{a†} ordem.

O número de nós interiores, para cada variável de cada sub-modelo, varia maioritariamente entre 0 e 3, verificando-se adicionalmente que:

- Pontualmente, sub-modelos associados à variável 2 apresentam 4, 5 ou 6 nós interiores.
- O número de nós interiores para a variável 3 eleva-se até 16 para muitas situações em que os padrões de entrada são transformados de acordo com o ponto 3.

Com base nestas observações, pode-se concluir que as aprendizagens levadas a cabo não conduzem a um modelo com estrutura comum para todos os processos. Considerando cada configuração isoladamente, também não existe um modelo que sirva para todos os sistemas. Este facto poderá ser explicado pelo reduzido número de padrões utilizados para treino que, para a presente situação, não excede os 21 padrões. Desta forma, surgiu a necessidade de se gerar maior número de padrões para um dado elemento do conjunto base de experimentação, para se verificar esta hipótese.

6.2.2 Modelo neuronal para um único elemento do conjunto base de experimentação

Nesta altura está provado que não existe um modelo neuronal cuja estrutura permita aproximar simultaneamente os 110 mapeamentos do tipo $PID \rightarrow ITAE$, com bons resultados. Um dos factores apontados para este facto é o reduzido número de padrões PID (entre 20 e 30) utilizados para cada processo do conjunto base de experimentação. O aumento desse número para tratamento simultâneo dos 110 processos não é viável, pois conduz a simulações com um peso computacional inabarcável.

Sabendo-se que sintonias adequadas do controlador PID correspondem a baixos valores do $ITAE$, então os pontos de interesse estão maioritariamente nas vizinhanças do seu mínimo. Assim, nesta secção vai ser escolhido um processo representativo do conjunto base de

[†] Os sub-modelos de 2^a ordem aparecem maioritariamente para a configuração 3.

experimentação, para o qual se definirá um maior número de vectores PID nas vizinhanças do mínimo do $ITAE$. Seguidamente, procede-se ao estudo e caracterização da função $ITAE \equiv f(PID)$ no sub-domínio escolhido, bem como ao treino de redes neuronais para o mapeamento $PID \rightarrow ITAE$.

Caracterização da função $ITAE \equiv f(PID)$: Utilizando-se o algoritmo apresentado na Tab. 6.5, foram gerados novos vectores PID em torno do $ITAE$ mínimo, alternativos aos que se obtiveram ao longo das secções 5.3.

1. Para cada processo, toma-se o $PID = [k_c^o \quad t_i^o \quad t_d^o]$, optimizado para o $ITAE$.
2. Considerando constante k_c^o , determinam-se limites de discretização para cada um dos outros dois parâmetros. Começando por t_i , esses limites são respectivamente dados por: $t_i^{\max} = \left(1 + \frac{p}{100}\right)t_i^o$, $t_i^{\min} = \left(1 - \frac{p}{100}\right)t_i^o$, sendo p a percentagem de perturbação em torno do óptimo; no presente estudo foi feito $p=50\%$.
3. Determinação de um vector de n pontos t_i uniformemente espaçados entre os limites definidos no ponto anterior: v_Ti . Na corrente aplicação fez-se $n=20$.
4. Repetição dos pontos 2 e 3 para o parâmetro t_d , obtendo-se um vector de 20 pontos: v_Td .
5. Tomando K_c^o constante para todos os padrões, constrói-se uma matriz de padrões PID mat_kc , cruzando-se os pontos de v_Ti e v_Td ; esta matriz terá n^2 linhas = 400 linhas.
6. Construção de matrizes mat_ti e mat_td por repetição dos pontos 2 a 5, considerando sucessivamente t_i^o constante e t_d^o constante; as 3 matrizes totalizam 1200 padrões³.
7. Para cada padrão PID estável e com tempo de estabelecimento aceitável, é determinado o respectivo valor de $ITAE$.
8. Os novos padrões são normalizados de acordo com (5.16).

Tab. 6.5 – Heurística para obtenção de padrões nas vizinhanças do mínimo do $ITAE$.

Dada a regularidade com que foi obtido o conjunto base de experimentação, vai ser escolhido o processo (caracterizado por $T_1=1$, $T_2=0.242$ e $L=0.306$) para a caracterização da função $ITAE \equiv f(PID)$. Para que esta função pudesse ser visualizada no espaço

³ este número poderá não ser atingido, pois alguns dos PID gerados são instáveis ou apresentam tempos de estabelecimento bastante elevados, logo, não serão considerados.

tridimensional, houve a necessidade de assumir sucessivamente constantes e iguais ao óptimo cada um dos 3 parâmetros *PID* (Fig. 6.1 e Fig. 6.2).

Aprendizagem do mapeamento no critério: As superfícies que se apresentam na Fig. 6.1 e Fig. 6.2, permitem-nos antever a viabilidade da aprendizagem dum modelo para o mapeamento em estudo: $PID \rightarrow ITAE$.

Continuando-se a utilizar o processo da secção anterior ($T_1=1$, $T_2=0.242$ e $L=0.306$), após algumas aprendizagens iniciais em que se verificavam valores elevados para os pesos, apresenta-se na Fig. 6.3 a evolução gráfica do *BIC*, complexidade, *MSE* e *MSRE*, para uma situação em que o factor de regularização $\lambda = 10^{-3}$.

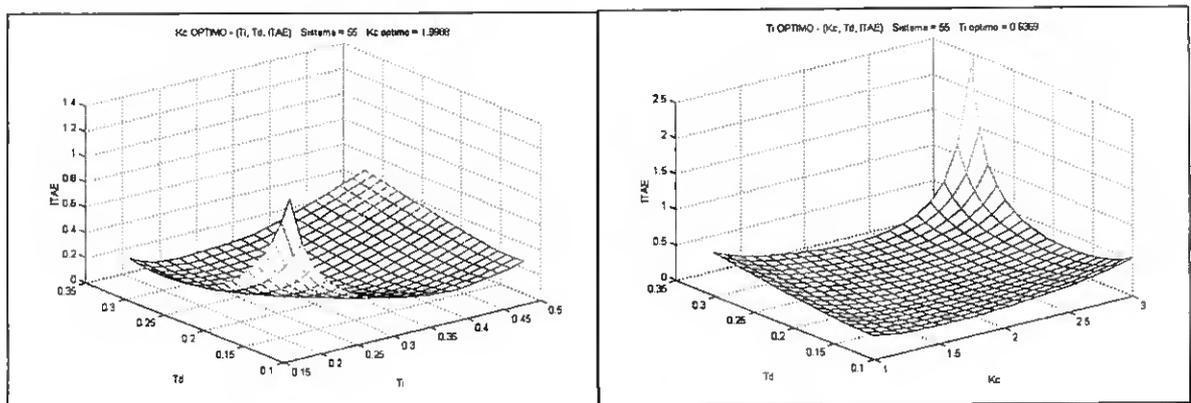


Fig. 6.1 – Esquerda k_c^o constante $ITAE \equiv f(t_i, t_d)$. Direita t_i^o constante $ITAE \equiv f(k_c, t_d)$.

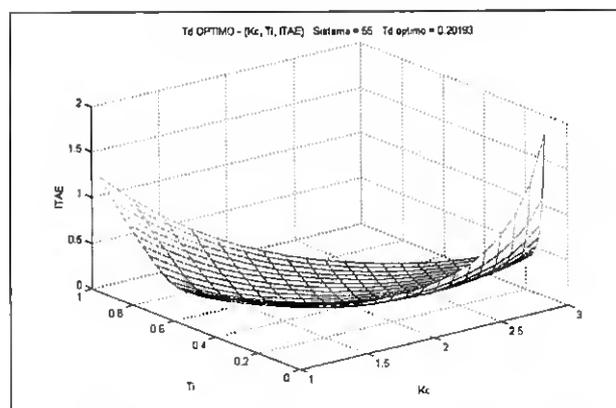


Fig. 6.2 – t_d^o constante $ITAE \equiv f(k_c, t_i)$.

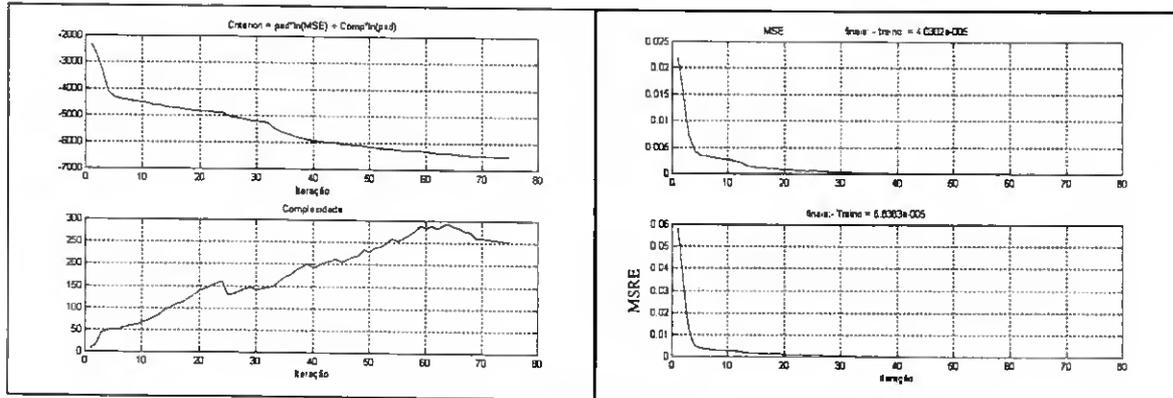


Fig. 6.3 – Modelo *PID* → *ITAE* com $\lambda = 10^{-3}$.

O modelo correspondente à iteração 74 (valor mínimo do *BIC*), apresenta-se na Tab. 6.6.

Nsm	Var	Nint.	Ord.																
4	3	0	3	12	9	12	3	3	13	5	4	3	3	3	2	4	6	3	3

Tab. 6.6 – Modelo óptimo.

Analisando a Fig. 6.3, verifica-se que com a geração de um maior conjunto de dados se consegue obter uma rede neuronal de elevada exactidão, conduzindo a valores dos parâmetros de desempenho: $MSE = 4 \times 10^{-5}$ e $MRE = 5.8 \times 10^{-5}$.

6.2.3 Modelo com estrutura parcialmente fixa

No início da secção 6.2 foi verificado que, para a dimensionalidade deste problema (5 variáveis de entrada), a aprendizagem das redes neuronais do tipo *B-Spline* apresentava uma complexidade computacional considerável. Os resultados obtidos não apresentavam grande exactidão. Foi demonstrado na secção 6.2.1, que o algoritmo *ASMOD* não consegue descobrir uma estrutura semelhante para o mapeamento *PID* → *ITAE* para todos os processos do conjunto base. Na secção 6.2.2 também foi demonstrado que, aumentando o número de dados, para um único processo, as redes neuronais conseguiram obter resultados excelentes. A solução de utilizar um maior volume de dados não é, no entanto viável, dado traduzir-se em cargas computacionais incontroláveis.

Nesta secção vai ser descrita uma técnica que, de algum modo, combina as estratégias desenvolvidas nas secções 6.2.1 e 6.2.2.

Sabendo-se que na vizinhança de um mínimo, uma função multivariada pode ser aproximada por uma função quadrática, vai-se admitir que, para cada sistema, tomando um vector PID representado por $\mathbf{x}^T = [x_1 \ x_2 \ x_3]$, o correspondente $ITAE$ pode ser aproximado por (6.1):

$$\mathbf{xGx}^T + \mathbf{h}^T \mathbf{x} + C \quad (6.1)$$

Um sub-modelo multivariável com *Splines* de 3ª ordem para cada entrada, tem a saída dada pela expressão (6.2).

$$f(\mathbf{x}) = (a_1 x_1^2 + b_1 x_1 + c_1)(a_2 x_2^2 + b_2 x_2 + c_2) \cdots (a_n x_n^2 + b_n x_n + c_n) \quad (6.2)$$

Repare-se que a expressão (6.2) gera mais termos do que a expansão em série de Taylor de 2ª ordem.

Pode-se ter uma implementação mínima de $f(\mathbf{x})$ com a seguinte composição:

- n sub-modelos bivariáveis que implementam cada termo do tipo $G_{i,i} x_i^2$
- $\binom{n}{2}$ sub-modelos bivariáveis, que combinam todas as n variáveis de entrada aos pares e implementam os termos do tipo $2G_{i,j} x_i x_j$.
- n sub-modelos univariáveis que implementam os termos do tipo $h_i x_i$.
- Uma *Spline* de 1ª ordem, com $w = c$.

Assim, a saída da rede neuronal pode-se representar pela expressão (6.3).

$$y = \sum_{i=1}^n \sum_{j=1}^n A_{i,j} x_i x_j + \sum_{i=1}^n b_i x_i + d \quad (6.3)$$

Para se determinar o mínimo da função $f(\mathbf{x})$ vai-se assumir que a variável vectorial \mathbf{x} é composta por $\#ind$ componentes independentes e $\#dep$ componentes dependentes. A determinação do óptimo \mathbf{x}_{dep} para um valor particular de \mathbf{x}_{ind} , pode ser conseguida pela resolução da equação (6.4).

$$\left. \frac{\partial f}{\partial \mathbf{x}_{dep}^T} \right|_{\mathbf{x}_{ind}} = \left. h_{\mathbf{x}_{dep}} \right|_{\mathbf{x}_{ind}} = 0 \quad (6.4)$$

Pretende-se então, determinar a topologia da rede neuronal do tipo *B-Spline* mais adequada.

Considere-se um modelo do tipo *B-Spline* cuja saída é dada pela expressão (6.3). Assumindo que cada conjunto de coeficientes $\{A_{i,j}, b_i, d\}$ é multiplicado por sub-modelos do tipo $S_{i,j}(\mathbf{x}_{ind})$, então a expressão (6.3) pode assumir a forma (6.5).

$$y = \sum_{i=1}^{\#dep} \sum_{j=1}^{\#dep} S_{i,j}(\mathbf{x}_{ind}) x_i x_j + \sum_{i=1}^{\#dep} S_{i,0}(\mathbf{x}_{ind}) x_i + S_{0,0}(\mathbf{x}_{ind}) \quad (6.5)$$

Assim, a solução de (6.4) é dada por (6.6).

$$\hat{\mathbf{x}}_{dep} \Big|_{ind} = -[S_{i,j}(\mathbf{x}_{ind})]^{-1} [S_{i,0}] / 2 \quad (6.6)$$

Considere-se uma situação em que apenas existe uma variável independente e uma variável dependente; assim, a implementação de (6.5) é dada por (6.7).

$$S_{i,0}(\mathbf{x}_{ind}) = \sum_{i_{ind}=1}^{N_{ind}} w_{i_{ind}} N_{i_{ind}}^k(\mathbf{x}_{ind}) \quad (6.7)$$

A expressão (6.7) representa uma soma ponderada de N_{ind} funções de base de ordem k . Incorporando uma *B-Spline* bivariável ao modelo (6.7), a saída y vem dada por (6.8)

$$y = \sum_{i_{dep}=1}^{N_{dep}} \sum_{i_{ind}=1}^{N_{ind}} w_{i_{dep}, i_{ind}} N_{i_{ind}}^k(\mathbf{x}_{ind}) N_{i_{dep}}^{k'}(\mathbf{x}_{dep}) \quad (6.8)$$

Designando por $w_{1,i_{ind}} = \lambda_0 w_{i_{ind}}$ e $w_{2,i_{ind}} = \lambda_1 w_{i_{ind}}$, o desenvolvimento de (6.8) conduz à seguinte expressão:

$$y = \lambda_0 S_{i,0}(\mathbf{x}_{ind}) N_1^2(\mathbf{x}_{dep}) + \lambda_1 S_{i,0}(\mathbf{x}_{ind}) N_2^2(\mathbf{x}_{dep}) \quad (6.9)$$

Deste modo, o processo de treino pode ser decomposto em duas etapas:

1. Determinação dos parâmetros do modelo de 2ª ordem para todos os processos do conjunto base.
2. Determinação de redes neuronais para aproximação dos parâmetros do modelo de 2ª ordem.

A 1ª etapa vai ser detalhada na secção 6.2.4 e a 2ª etapa na secção 6.2.5.

6.2.4 Modelo de 2ª ordem para aproximação nas vizinhanças do mínimo

Considerando a expressão (6.1) pode-se definir o *ITAE* à custa da matriz G (3×3), do vector h (3×1) e do escalar c , tal como se apresenta em (6.10).

$$x^T G x + x^T h + c = ITAE \quad (6.10)$$

Considerando os elementos da matriz G tais que $G_{21} + G_{12} = g_{12}$, $G_{23} + G_{32} = g_{23}$ e $G_{13} + G_{31} = g_{13}$, a expansão da expressão (6.10), resulta na seguinte forma:

$$G_{11}x_1^2 + G_{22}x_2^2 + G_{33}x_3^2 + g_{12}x_1x_2 + g_{23}x_3x_2 + g_{13}x_3x_1 + x_1h_1 + x_2h_2 + x_3h_3 + c = ITAE \quad (6.11).$$

Definindo-se:

$$x_v = [x_1^2 \quad x_2^2 \quad x_3^2 \quad x_1x_2 \quad x_3x_2 \quad x_3x_1 \quad x_1 \quad x_2 \quad x_3 \quad 1] \text{ e}$$

$g_v = [G_{11} \quad G_{22} \quad G_{33} \quad g_{12} \quad g_{23} \quad g_{13} \quad h_1 \quad h_2 \quad h_3 \quad c]^T$, a expressão (6.11) pode ser escrita como produto interno dos vectores x_v e g_v .

$$x_v \cdot g_v = ITAE \quad (6.12).$$

Considerando n vectores *PID* para um dado processo, o vector x_v passa a matriz X_m de n linhas, e o escalar *ITAE* passa a vector $ITAE_v$ de dimensão n , verificando-se:

$$X_m \cdot g_v = ITAE_v \quad (6.13)$$

Designando o i -ésimo padrão PID por $x(i) = [x_1(i) \ x_2(i) \ x_3(i)]$, a expressão (6.13) para os n padrões, assume o seguinte aspecto na forma expandida:

$$\begin{bmatrix}
 x_1^2(1) & x_2^2(1) & x_3^2(1) & x_1x_2(1) & x_3x_2(1) & x_3x_1(1) & x_1(1) & x_2(1) & x_3(1) & 1 \\
 \vdots & \vdots \\
 x_1^2(n) & x_2^2(n) & x_3^2(n) & x_1x_2(n) & x_3x_2(n) & x_3x_1(n) & x_1(n) & x_2(n) & x_3(n) & 1
 \end{bmatrix}
 \begin{bmatrix}
 G_{11} \\
 G_{22} \\
 G_{33} \\
 g_{12} \\
 g_{23} \\
 g_{13} \\
 h_1 \\
 h_2 \\
 h_3 \\
 c
 \end{bmatrix}
 =
 \begin{bmatrix}
 ITAE(1) \\
 \vdots \\
 ITAE(n)
 \end{bmatrix}
 \quad (6.14)$$

A expressão (6.14) representa um sistema de n equações lineares a 10 incógnitas, em geral inconsistente. Resta então a hipótese de se determinar uma solução que minimize o quadrado do erro. Havendo 10 variáveis independentes, podem-se fixar algumas de tal forma que o erro, ou a sua derivada, sejam nulos nalguns pontos; desta forma, assumiu-se a seguinte restrição:

Para cada processo, o modelo de 2ª ordem será tal que o seu mínimo⁴, deverá coincidir com o mínimo da base de padrões⁵ usada para a constituição do modelo:

$$\begin{aligned}
 PID_{\min}^{\text{mod.}} &\equiv PID_{\min}^{\text{base}} \\
 ITAE_{\min}^{\text{mod.}} &\equiv ITAE_{\min}^{\text{base}}
 \end{aligned}
 \quad (6.15).$$

Tendo o vector PID 3 componentes e sendo $ITAE$ um escalar, as expressões (6.15) implicam 4 restrições no modelo apresentado na expressão (6.14), pelo que vão ser fixados 4 dos 10 parâmetros, de acordo com as expressões seguintes:

- Imposição de $ITAE_{\min}$, a partir de (6.11):

⁴ determinado analiticamente: $PID_{\min} = -G^{-1} * h/2$.

⁵ base de padrões obtida de acordo com a heurística da Tab. 6.5. Por vezes o valor mínimo assim encontrado não é exactamente igual ao que se obteve numericamente, mas andará muito próximo.

$$G_{11}x_1^2 + G_{22}x_2^2 + G_{33}x_3^2 + g_{12}x_1x_2 + g_{23}x_3x_2 + g_{13}x_3x_1 + \dots \quad (6.16)$$

$$\dots x_1h_1 + x_2h_2 + x_3h_3 + c = ITAE_{\min} \left| \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \right. = PID_{\min}$$

- No mínimo, dever-se-á obter a anulação das derivadas parciais em ordem a cada componente do *PID*:

$$\frac{\partial ITAE}{\partial x_1} = 0, \quad \frac{\partial ITAE}{\partial x_2} = 0, \quad \frac{\partial ITAE}{\partial x_3} = 0 \left| \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \right. = PID_{\min} \quad (6.17)$$

A escolha dos 4 parâmetros para cumprimento das restrições foi feita tendo em conta que os termos quadráticos e os termos cruzados do modelo se prendem com a curvatura deste nas vizinhanças do mínimo. Desta forma, os parâmetros correspondentes deverão ser determinados por um processo de minimização para que o erro seja o menor possível, tendo-se estabelecido que:

- Os elementos da matriz G^\dagger ($g_{11} \ g_{22} \ g_{33} \ g_{12} \ g_{23} \ g_{13}$), serão determinados por resolução do sistema (6.18), que se obtém de (6.14):

$$\begin{bmatrix} x_1^2(1) & x_2^2(1) & x_3^2(1) & x_1x_2(1) & x_3x_2(1) & x_3x_1(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^2(n) & x_2^2(n) & x_3^2(n) & x_1x_2(n) & x_3x_2(n) & x_3x_1(n) \end{bmatrix} \cdot \begin{bmatrix} g_{11} \\ g_{22} \\ g_{33} \\ g_{12} \\ g_{23} \\ g_{13} \end{bmatrix} = \begin{bmatrix} ITAE(1) \\ \vdots \\ ITAE(n) \end{bmatrix} \quad (6.18)$$

Sendo o número de equações superior ao número de incógnitas, o sistema é em geral inconsistente. Lembrando que X_m^+ representa a matriz pseudo-inversa de X_m , a solução que minimiza o quadrado do erro será dada por:

$$G = X_m^+ \cdot ITAE_v \quad (6.19).$$

† Para a matriz G real simétrica verifica-se: $G_{21}=G_{12}=g_{12}/2$; $G_{23}=G_{32}=g_{23}/2$ e $G_{13}=G_{31}=g_{13}/2$;

- Os elementos $[h_1 \ h_2 \ h_3 \ c]$ são determinados por imposição das restrições (6.16) e (6.17), conduzindo às expressões (6.20) nas quais $[x_1 \ x_2 \ x_3] = PID_{\min}$ e $ITAE = ITAE_{\min}$.

$$\begin{aligned}
 h_1 &= -2G_{11}x_1 - g_{12}x_2 - g_{13}x_3 \\
 h_2 &= -g_{12}x_1 - 2G_{22}x_2 - g_{23}x_3 \\
 h_3 &= -g_{13}x_1 - g_{23}x_2 - 2G_{33}x_3 \\
 c &= ITAE - (G_{11}x_1^2 + G_{22}x_2^2 + G_{33}x_3^2 + g_{12}x_1x_2 + g_{23}x_3x_2 + g_{13}x_3x_1 + h_1x_1 + h_2x_2 + h_3x_3)
 \end{aligned}
 \tag{6.20}.$$

O algoritmo implementado para a avaliação dos 10 parâmetros do modelo de 2ª ordem para cada processo do conjunto base de experimentação começa por considerar a totalidade de padrões do tipo $PID \rightarrow ITAE$ estipulados de acordo com a secção 6.2.2. Segue-se um processo cíclico em que o raio da vizinhança centrada no PID_{\min} é reduzido sucessivamente por aplicação do factor 0.95 e, conseqüentemente, menor será o número de padrões utilizados na construção do modelo. Para cada ciclo, é primeiramente determinada a matriz G de acordo com (6.19) e seguidamente, os restantes parâmetros do modelo são determinados de acordo com (6.20).

Convém notar que o PID_{\min} é o padrão que conduz ao menor valor de $ITAE$ dentro da discretização feita para cada sistema.

O processo repete-se até que se verifiquem as condições de paragem segundo as quais o modelo encontrado será tal que:

- O número de padrões utilizados na sua construção é $\geq 20^\ddagger$,
- A % de MRE calculada para os referidos padrões é ≤ 10.3 ,
- A matriz G correspondente ao modelo é definida positiva.

Ensaio prévios para determinação dos modelos permitiram concluir que os 1200 padrões PID iniciais por sistema poderiam ser insuficientes, pelo que houve necessidade de se aumentar esse número. Gerou-se 1200 novos padrões por sistema, utilizando-se o

[‡] $20 = 2 \times n^\circ$ de parâmetros do modelo.

algoritmo da Tab. 6.5, sendo de 20% a perturbação utilizada em torno de cada parâmetro óptimo, em vez dos iniciais 50%. Verificou-se, para cada processo, que entre o novo conjunto de padrões assim obtido e o conjunto inicial não havia padrões repetidos⁶.

A estratégia apresentada faz com que a cada processo fique associado um modelo definido por um vector de 10 parâmetros P_i . Posteriormente, serão treinadas 10 redes neuronais para modelarem mapeamentos do tipo: $MI \rightarrow P_i$.

A avaliação do desempenho dos modelos depende da adequação dos mesmos e da exactidão com que são obtidos. Esta avaliação é cumprida em duas etapas:

1ª Etapa: Tratada na presente secção com base nos resultados já obtidos. Tem como objectivo a verificação da adequação de cada modelo de 2ª ordem face ao conjunto de vectores PID que o originou. Com este procedimento pretende-se medir a representatividade dos vectores PID na construção do modelo, bem como a capacidade deste para avaliar os correspondentes valores de $ITAE$.

2ª Etapa: Tratada na secção 6.2.10, após a aprendizagem de redes neuronais para o mapeamento $MI \rightarrow P_i$. Os modelos aproximados obtidos⁷, deverão ser validados com os modelos exactos. Simultaneamente deverá ser testada a exactidão com que esses modelos são capazes de avaliar a função $ITAE \equiv f(PID)$, em especial para pontos na vizinhança do mínimo.

Os modelos obtidos permitiram verificar que, tal como o algoritmo o exigia:

- a matriz G é sempre positiva definida,
- o mínimo do modelo determinado analiticamente afasta-se do menor valor da base de padrões de cada processo, de entre 0 a 10^{-12} .

As medidas de desempenho para os 110 processos foram registadas numa tabela com as seguintes colunas:

⁶ Posteriormente, e para alguns processos, houve necessidade de se aumentar os 2400 padrões.

⁷ Redes neuronais treinadas.

- N° Pad.: Totalidade de padrões na vizinhança do mínimo utilizados para definir o modelo.
- % Pad.: Percentagem do valor da coluna anterior em relação à totalidade de padrões utilizados na 1ª iteração.
- *MSE* total: *MSE* avaliado para a totalidade de padrões pertencentes à vizinhança, utilizada na definição do modelo.
- % *MRE*: Valor percentual de *MRE* avaliado para a totalidade de padrões intervenientes na definição do modelo.
- %Rf./Ri.: Valor percentual do raio utilizado para definir a vizinhança que originou o modelo, em relação ao raio inicial⁸.
- %Rf./PID: Percentagem do raio utilizado para definir a vizinhança que originou o modelo, em relação à norma do PID_{\min} que serviu de centro.

Na Tab. 6.7, apresentam-se os valores mínimos, médios e máximos, para as medidas de desempenho referidas, tendo em conta os 110 modelos dos processos que constituem o conjunto base de experimentação.

	N° Pad.	% Pad.	<i>MSE</i>	% <i>MRE</i>	%Rf./Ri.	%Rf./PID
máximo	1724	72,3	1,1	10,3	59,9	29,5
mínimo	40	1,1	4,7E-07	5,5	0,5	0,3
Médio	958,2	40,0	0,1	9,8	32,2	15,8

Tab. 6.7 – Valores de desempenho para os 110 modelos do conjunto base de experimentação.

A evolução de cada parâmetro em função das medidas de identificação segue um andamento que se exemplifica para dois parâmetros típicos, respectivamente parâmetros 1 e 6 (Fig. 6.4).

⁸ Raio inicial para cada sistema = distância do PID_{\min} ao PID mais afastado.

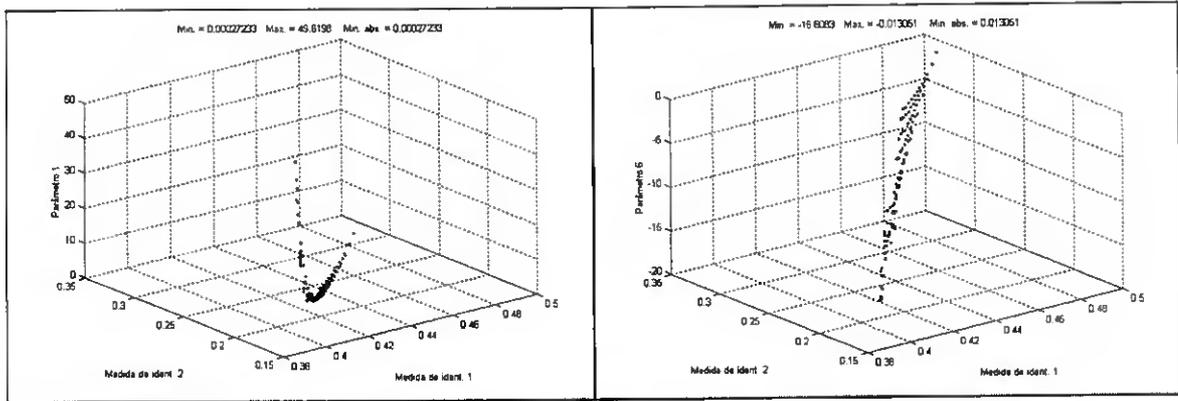


Fig. 6.4 – Parâmetros 1 e 6 em função das medidas de identificação.

Nesta altura obtiveram-se 110 modelos de 2ª ordem para os respectivos processos que integram o conjunto base de experimentação. Tal como ficou provado no decurso das últimas secções, tais modelos são exactos no ponto $(PID_{\min}, ITAE_{\min})$ e constituem uma boa aproximação para os pontos contidos na sua vizinhança. A composição de cada vizinhança bem como a representatividade no conjunto diferem consoante o processo.

Seguidamente pretende-se construir 10 modelos neuronais que aproximem o mapeamento $MI \rightarrow P_i$, possibilitando a avaliação do modelo de 2ª ordem para o respectivo processo. Cada modelo assim obtido possibilitará a avaliação do *ITAE* do respectivo sistema, para os valores de *PID* contidos na vizinhança de $(PID_{\min}, ITAE_{\min})$, com particular exactidão para o centro.

A observação das evoluções dos parâmetros em função das medidas de identificação (das quais se apresenta como exemplo a Fig. 6.4), permite-nos supor que a constituição de tais modelos neuronais poderá ser viável. No entanto, verifica-se que o 5º parâmetro é nulo para alguns sistemas; para além disso, a quase totalidade dos restantes parâmetros apresenta uma variação em termos de módulo que está compreendida entre $10E-3$ e $10E+2$. A gama de variação das saídas das redes neuronais será muito elevada, o que poderá ter como consequência:

- dificuldade na avaliação da qualidade da aprendizagem,
- dificuldade em extrair informação sobre medidas de desempenho relativas tais como *MRE* e *MSRE*.

6.2.5 Redes neuronais para avaliação dos parâmetros dos modelos de 2ª ordem

O modelo de 2ª ordem de cada processo fica definido num vector de 10 componentes: $[P_1 \dots P_{10}]$. Sabendo-se que o conjunto base de experimentação é composto por $n=110$ processos, os respectivos modelos de 2ª ordem que o caracterizam organizam-se numa matriz P (110×10) do tipo (6.21):

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} & P_{1,5} & P_{1,6} & P_{1,7} & P_{1,8} & P_{1,9} & P_{1,10} \\ \vdots & \vdots \\ P_{n,1} & P_{n,2} & P_{n,3} & P_{n,4} & P_{n,5} & P_{n,6} & P_{n,7} & P_{n,8} & P_{n,9} & P_{n,10} \end{bmatrix} \quad (6.21).$$

Esta secção debruça-se sobre a aprendizagem de redes neuronais do tipo *B-Spline*, capazes de aproximar cada parâmetro do modelo, dado um par de medidas de identificação do processo respectivo. Tal será conseguido através do mapeamento: $MI \rightarrow P_i, \quad i=1\dots 10$.

O mapeamento das medidas de identificação nos 10 parâmetros que vão constituir os modelos de 2ª ordem recorre à aprendizagem de 10 redes neuronais do tipo *B-Spline*, em duas etapas:

1. Ensaio de diferentes estratégias de aprendizagem de redes neuronais do tipo *B-Spline*.
2. Utilização da estratégia mais conveniente de acordo com o estudo efectuado no ponto anterior, para a aprendizagem das 10 redes neuronais para os respectivos mapeamentos.

O ponto 1 será tratado nesta sub-secção e o ponto 2 constituirá matéria da secção 6.2.9.

Os algoritmos utilizados para a aprendizagem de redes neuronais do tipo *B-Spline* foram o *ASMOD* e a programação genética.

A multiplicidade de situações que a programação genética multi-objectivo origina em termos de objectivos, prioridades e metas, faz com que nesta fase de comparação de estratégias de aprendizagem seja suficiente considerar apenas o mapeamento no 1º parâmetro, correspondente à 1ª coluna da matriz P .

Conjuntos de aprendizagem: O conjunto de entrada é composto pelas medidas de identificação organizadas numa matriz (110×2) ; esta matriz é transformada linearmente de acordo com o método apresentado na secção 5.2.4. Cada conjunto de saída é composto pela coluna da matriz P , definida em (6.21) e correspondente ao parâmetro do modelo para o qual se pretende fazer o mapeamento. Os conjuntos de treino e teste são extraídos aleatoriamente originando 77 padrões de treino e 33 padrões de teste.

Os modelos obtidos são tais que alguns elementos da matriz P são nulos o que levaria a surgirem problemas quando se fossem calcular medidas de desempenho relativas; desta forma, nesta fase de escolha de estratégia de aprendizagem, optou-se por substituir os valores nulos por quantidades na ordem de 10^{-12} .

6.2.6 Aprendizagem de *B-Splines* utilizando *ASMOD*

A utilização do *ASMOD* para aprendizagem deste tipo de redes neuronais foi estruturada em 3 fases:

1. Utilização apenas do conjunto de treino: Os padrões de teste foram ignorados tanto na execução do algoritmo como na monitorização das medidas de desempenho ao longo da aprendizagem. Verificou-se que se consegue obter modelos tão precisos quanto se queira, desde que se descuide a capacidade de generalização dos mesmos bem como o tempo levado a obtê-los. Então, os resultados previamente obtidos permitem concluir que a melhoria das medidas de desempenho apenas depende da extensão da aprendizagem.
2. Utilização do conjunto de treino e monitorização dos indicadores de desempenho de treino e teste: Trata-se do processo de aprendizagem corrente, em que a evolução dos indicadores de desempenho é observada tanto para o conjunto de treino como para o conjunto de teste, sendo este último condicionante na paragem do processo de aprendizagem.
3. Aprendizagem utilizando-se a totalidade dos padrões: Para se obter uma maior base de treino são utilizados nesta fase ambos os conjuntos de treino e teste para treino. O conjunto de teste deixa de existir; assim, esta estratégia deverá conduzir a modelos mais precisos, embora a capacidade de generalização seja desconhecida.

Critério de paragem: O processo de aprendizagem termina na iteração em que o valor do *BIC* sobe; adicionalmente, quando o conjunto de teste é utilizado, a subida de 4 vezes consecutivas da medida de desempenho[†] para este conjunto provoca igualmente a paragem do processo.

Os resultados obtidos com base nos pontos 2 e 3 encontram-se sistematizados respectivamente na 1ª e 2ª metades da Tab. 6.8. As medidas de desempenho apresentadas são as respeitantes à iteração imediatamente anterior à subida de %*MRE* para o conjunto de teste.

Parâmetros →		1	2	3	4	5	6	7	8	9	10
<i>MSE</i>	Treino	8.75	0.40	0.79	1.19	1.36	1.13	4.53	3.14	7.77	14.57
	Teste	4.16	0.30	1.10	249.94	0.35	0.89	5.39	2.28	3.37	3.24
<i>MSRE</i>	Treino	0.36	0.03	0.01	0.02	3.7E-3	0.49	0.69	0.07	0.03	0.06
	Teste	6.70	0.75	0.01	9E4	0.04	2.94	16.12	5.86	0.77	0.28
% <i>MRE</i>	Treino	47.31	12.46	6.01	9.84	4.62	56.66	75.01	17.69	12.66	17.57
	Teste	112.47	39.08	7.20	7.3E3	11.61	104.11	175.07	87.75	29.69	27.18
<i>BIC</i>		-34.4	-224.2	-360.1	-157.7	-346.4	-19.5	7.4	-155.8	-228.1	-156.2
Complexidade		10	9	8	30	16	8	8	8	6	7
Treino	<i>MSE</i>	7.9E-4	0.11	0.82	0.03	0.39	0.42	5.3E-4	0.71	1.47	3.19
	<i>MSRE</i>	1.5E-6	3.9E-3	1E-2	1.7E-4	3.7E-3	2E-2	4.2E-5	3.2E-3	4.4E-3	1.3E-2
Teste	% <i>MRE</i>	0.04	4.08	6.40	0.55	4.55	8.7	0.35	3.72	4.8	7.79
<i>BIC</i>		-959.3	-432.7	-522.6	-530.3	-447.8	-160.1	-591.0	-334.9	-339.2	-225.1
Complexidade		110	38	8	91	36	57	110	63	55	53

Tab. 6.8 – Medidas de desempenho para o mapeamento $MI \rightarrow P_i$ usando-se *ASMOD*.

[†] A medida de desempenho será *MSE* ou *MRE* consoante se esteja a otimizar uma medida absoluta ou uma medida relativa.

6.2.7 Aprendizagem de *B-Splines* utilizando programação genética

A programação genética foi utilizada na aprendizagem de redes neuronais do tipo *B-Spline*, nas duas variantes: uni-objectivo e multi-objectivo.

Aprendizagem de *B-Splines* utilizando programação genética uni-objectivo: Para cada mapeamento, foram geradas aleatoriamente cerca de 10 populações iniciais de 65 indivíduos, ocorrendo a partir de cada uma delas, um processo de aprendizagem com a duração de 70 gerações. A função de custo utilizada foi o *BIC* tendo em conta o *MSE* treino, de acordo com a expressão (2.53).

Concluído o lote das aprendizagens para cada parâmetro, foi escolhida a rede neuronal que conduzia ao valor mínimo de um dado critério, tomando em consideração todas as gerações de todas as inicializações. Os critérios tidos em conta para a referida selecção foram o *BIC* e o *MSE* teste; para ambas as situações, os modelos obtidos conduzem a valores muito elevados para as medidas de desempenho relativas. Este facto resulta da substituição de alguns valores objectivo nulos por quantidades na ordem de 10^{-12} , como foi referido anteriormente, sendo esta constatação suportada pela Fig. 6.5.

Na Fig. 6.5 sobrepõem-se as saídas propagadas e as saídas objectivo para o conjunto de teste do modelo para o 1º parâmetro que conduz ao menor *MSE* teste. Embora a metade superior da Fig. 6.5 mostre que, globalmente, a saída da rede neuronal siga de muito perto os objectivos, antevendo-se baixos valores para o erro, na metade inferior, tendo sido feito um aumento de escala nas vizinhanças de 0, pode-se observar as discrepâncias em torno desse valor, justificando-se assim os elevados valores relativos para o conjunto de teste.

Os processos de aprendizagem realizados nesta fase permitiram verificar que as medidas de desempenho referentes aos conjuntos de treino são menores quando os modelos escolhidos apontam para valores baixos do critério *BIC*. No entanto, nessas circunstâncias, as medidas de desempenho no conjunto de teste são muito superiores àquelas que se verificam quando se procuram modelos com baixos valores de *MSE* no conjunto de teste. Desta forma, dos resultados obtidos durante a procura de valores baixos de *BIC*, apenas se registam para comparação os da coluna B da Tab. 6.15.

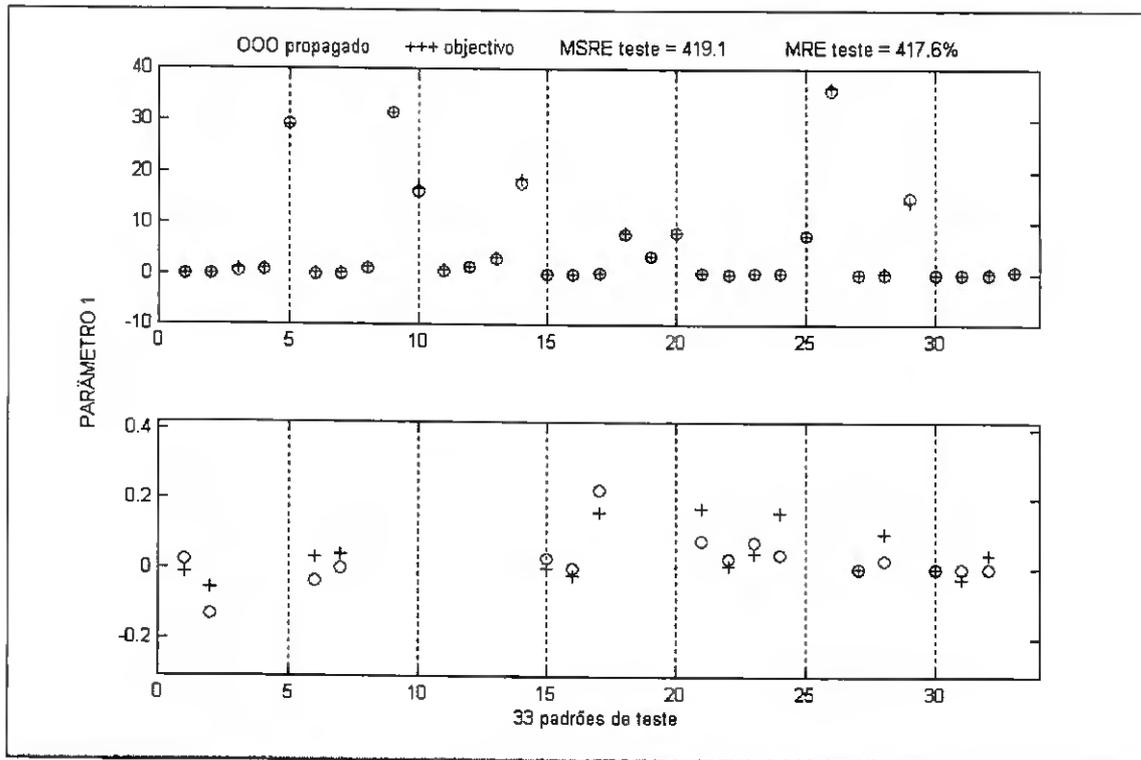


Fig. 6.5 – Saídas de teste para o parâmetro 1.

Na Tab. 6.9 registam-se as medidas de desempenho para os 10 parâmetros, quando os modelos conduzem aos menores valores de *MSE* no conjunto de teste. Dessa tabela será considerada a 1ª coluna, que contem os valores que irão ser comparados com as restantes estratégias de aprendizagem.

Parâmetros →	1	2	3	4	5	6	7	8	9	10	
<i>MSE</i>	Treino	1.4E-3	0.11	1.29	0.04	0.12	0.07	0.15	2.22	8.38	1.16
	Teste	5.9E-2	0.17	1.92	0.10	0.41	0.12	0.10	1.00	3.30	3.00
<i>MSRE</i>	Treino	18.01	0.07	0.01	65.23	2.5E21	35.70	1.9E21	8.7E22	7.3E23	2.2E23
	Teste	419.10	0.20	0.01	353.57	0.25	2.84	6.0E21	4.2E22	1.2E23	5.9E22
% <i>MRE</i>	Treino	106.67	14.25	8.30	222.57	8.2E11	130.11	8.2E11	6.7E12	1.8E13	1.0E13
	Teste	417.58	27.17	8.78	415.86	24.69	97.93	1.4E12	3.6E12	6.1E12	5.6E12
<i>BIC</i>		-293.3	-88.31	32.5	-118.08	-76.70	-106.92	-58.89	83.0	185.4	80.6
Complexidade		49	19	3	28	20	23	20	5	5	16

Tab. 6.9 – Mapeamento $MI \rightarrow P_i$ usando-se programação genética uni-objectivo para o modelo com menor *MSE* teste.

Tal como foi referido anteriormente, os elevados valores observados para as medidas de desempenho relativas devem-se ao facto, dos objectivos em relação aos quais estão a ser calculadas, estarem muito próximo de 0.

Aprendizagem de *B-Splines* utilizando programação genética multi-objectivo: Nesta fase de escolha da estratégia mais apropriada, apenas se considerou o mapeamento no 1º parâmetro (P_1), dado que o tratamento da totalidade dos parâmetros, recorrendo-se à programação genética multi-objectivo, implicaria elevado peso computacional, o que não se justificaria nesta fase inicial.

Com base nos valores obtidos previamente com a programação genética uni-objectivo, apresenta-se na Tab. 6.10 os objectivos, prioridades e metas a atingir, utilizados nesta fase de programação genética multi-objectivo.

Objectivos	<i>MSE</i> treino	<i>MSRE</i> treino	Complexidade
Prioridades	0	3	1
Metas	0.2	0.5	40

Tab. 6.10 – Valores para aprendizagem de redes neuronais do tipo *B-Spline* recorrendo à programação genética multi-objectivo.

Salienta-se que nesta fase de aprendizagem não foram considerados como objectivos medidas de desempenho referentes ao conjunto de teste. Pode-se assegurar, assim, que a sucessão das gerações e escolha das soluções durante o processo evolutivo se efectua desconhecendo por completo o conjunto de teste. A monitorização dos valores dos critérios para o conjunto de teste permite concluir sobre a capacidade de generalização do modelo obtido.

Terminado o processo de aprendizagem, obtém-se um conjunto de soluções não dominadas, das quais será seleccionado um modelo, utilizando-se o critério das soluções preferidas [93].

Foram geradas aleatoriamente aproximadamente 55 populações iniciais com 65 indivíduos, ocorrendo a partir de cada uma delas o correspondente processo de aprendizagem, com a duração de 50 gerações.

Para cada inicialização, considerando as $n=50$ gerações, verifica-se que:

$$\text{N.D.} \left(\bigcup_{i=1}^n S_i \right) \subset \bigcup_{i=1}^n S_i \quad (6.22)$$

sendo:

S_i – conjunto de soluções não dominadas da geração i ,

$\text{N.D.}(X)$ – conjunto das soluções não dominadas no conjunto de soluções X .

Para $m=55$ inicializações, obteve-se:

$$\# \bigcup_{j=1}^m T_j = 6803, \quad \# \text{N.D.} \left(\bigcup_{j=1}^m T_j \right) = 456, \quad \# \text{PRF.} \left(\bigcup_{j=1}^m T_j \right) = 1 \quad (6.23)$$

sendo:

T_j – conjunto de soluções não dominadas da inicialização j ,

$\text{PRF.}(X)$ – conjunto das soluções preferidas no conjunto de soluções X .

Escolha de soluções preferidas: Com o intuito de se obter soluções preferidas com maior capacidade de generalização do que aquelas a que se refere (6.23), tomou-se a base das 6803 soluções de (6.23), sobre a qual foram redefinidos diferentes vectores de objectivos, prioridades e metas a atingir, tendo em conta os seguintes princípios:

- Considerar todas as medidas de desempenho como objectivos.
- Dar mais prioridade aos indicadores relativos do que aos absolutos.
- Para a mesma medida de desempenho, dar a mesma prioridade ao valor do treino e teste.
- Atribuir valores das metas com base nas aprendizagens efectuadas nas etapas anteriores para o parâmetro em estudo.

Dado que, os pesos lineares da camada de saída da rede neuronal são obtidos à custa do cálculo da pseudo-inversa da matriz de saída das *B-Splines* dada por (2.38), é importante saber até que ponto esta matriz tem linhas próximas da dependência linear. Para se avaliar o grau de dependência linear da referida matriz recorre-se ao cálculo do seu número de condição, definido como sendo o cociente entre o maior e o menor valor singular da matriz. Valores elevados do número de condição significam proximidade de dependência

linear. Desta forma o número de condição da matriz de saída das *B-Splines* ($Cond(A)$) foi considerado também como um objectivo durante a aprendizagem.

Com base nos princípios apontados, estabeleceram-se os valores que se apresentam na Tab. 6.11.

Objectivos	<i>MSE</i> treino	<i>MSRE</i> treino	Comp.	$\ w\ $	$Cond(A)$	<i>MSE</i> teste	<i>MSRE</i> teste
Prioridades	0	3	1	1	1	0	3
Valores	0.1	1	50	100	100	0.1	1

Tab. 6.11 – Parâmetros para obtenção de soluções não dominadas e soluções preferidas tomando por base as 6803 soluções da expressão (6.23).

A utilização destes valores conduziu à obtenção de 3520 soluções não dominadas e de 6 soluções preferidas.

Na Tab. 6.12 apresentam-se os valores mínimos, médios e máximos das medidas de desempenho avaliadas, tanto para o conjunto de treino como para o conjunto de teste, tomando em consideração as 3520 soluções não dominadas.

	<i>MSE</i>		<i>MSRE</i>		% <i>MRE</i>		<i>BIC</i>	Compl.
	treino	teste	treino	Teste	treino	teste		
Min.	5.5E-4	5.8E-2	4.6E-4	5.1E-1	1.0	53.8	-299.4	1
Max.	120.7	1.2E+16	5.7E+7	3.6E+18	2.0E+5	3.8E+10	436.2	76
Média	8.9	3.6E+12	1.0E+5	1.8E+15	1.1E+3	2.4E+7	146.9	27

Tab. 6.12 – Medidas de desempenho para as 3520 soluções não dominadas.

Na Tab. 6.13 apresentam-se as medidas de desempenho para as 6 soluções preferidas.

Para que seja possível efectuar um estudo comparativo com as restantes abordagens, deverá ser seleccionada uma solução única; assim, adopta-se a solução com menor *BIC* (solução 5), cujo modelo se apresenta na Tab. 6.14.

6.2.8 Escolha duma estratégia de aprendizagem

Para que seja possível fazer uma leitura comparativa da informação recolhida e tirar conclusões referentes à estratégia a adoptar, apresenta-se a tabela resumo (Tab. 6.15). Esta tabela reúne a informação recolhida respeitante à aprendizagem de redes neuronais para o mapeamento $MI \rightarrow P_1$, quando se usam as estratégias seguintes:

- A. *ASMOD* (1ª coluna da Tab. 6.8).
- B. programação genética uni-objectivo para o menor *BIC*.
- C. programação genética uni-objectivo para o menor *MSE* do conjunto de teste (1ª coluna da Tab. 6.9).
- D. programação genética multi-objectivo com menor *BIC* para o conjunto das soluções preferidas (5ª solução da Tab. 6.13).

Solução →		1	2	3	4	5	6
<i>MSE</i>	Treino	0.35	0.28	0.22	0.28	0.22	0.93
	Teste	1.94	2.56	2.58	2.57	2.57	1.51
<i>MSRE</i>	Treino	0.62	0.61	0.32	0.30	0.52	0.80
	Teste	0.51	0.76	0.75	0.71	0.74	0.72
% <i>MRE</i>	Treino	39.34	34.09	24.33	25.28	32.79	48.88
	Teste	57.38	65.36	66.02	63.99	64.75	66.59
<i>BIC</i>		92.18	67.31	77.74	83.40	63.11	128.77
Complexidade		40	38	45	42	41	31

Tab. 6.13 – Soluções preferidas.

Nsm	Var.	Nint.	Ord.
1	1 2	19 2	1 1

Tab. 6.14 – Modelo para a solução preferida 5.

Estratégia →		A	B	C	D
<i>MSE</i>	Treino	8.75	2E-5	1.4E-3	0.22
	Teste	4.16	2.2E4	5.9E-2	2.57
<i>MSRE</i>	Treino	0.36	0.15	18.01	0.52
	Teste	6.70	1.8E11	419.10	0.74
% <i>MRE</i>	Treino	47.31	8.82	106.67	32.79
	Teste	112.47	1.04E7	417.58	64.75
<i>BIC</i>		-34.4	-507.0	-293.3	63.11
Complexidade		10	76	49	41

Tab. 6.15 – Medidas de desempenho para mapeamento $MI \rightarrow P_1$

Medidas absolutas e medidas relativas: Tendo em consideração os resultados obtidos ao longo desta secção, com particular destaque para os que se apresentam na tabela resumo (Tab. 6.15), verificam-se elevados valores para as medidas de desempenho relativas, para ambos os conjuntos de aprendizagem. Como referido anteriormente, tal situação justificase pelo facto de alguns dos valores nulos dos parâmetros para os quais se está a fazer o mapeamento terem sido substituídos por valores na ordem de 10^{-12} ; esta observação é suportada pela Fig. 6.5.

Desta forma, para que as medidas relativas pudessem constituir bons indicadores de desempenho, os valores das saídas desejadas deveriam ser deslocados das proximidades de zero.

Escolha duma solução final: Considerando essencialmente as medidas de desempenho absolutas e tomando como base a Tab. 6.15, a solução adoptada seria a que se obtém com uma das estratégias C ou D. Embora C conduza a modelos mais precisos, segundo D, os valores de *MSE* para treino e teste são menos discrepantes, antevendo-se melhor capacidade de generalização. Para além disso, no que se refere às medidas de desempenho relativas, estas são consideravelmente piores para a estratégia C.

Deve ser notado que o uso da programação genética multi-objectivo para aprendizagem de redes neuronais não foi esgotado. Como referido anteriormente, esta estratégia possibilita grande número de configurações a nível de vectores de objectivos, prioridades e metas a

atingir, tanto na fase de aprendizagem, como para a determinação final das soluções não dominadas e soluções preferidas. Desta forma, uma melhor exploração desse campo, poderia conduzir a soluções mais robustas do que aquelas que agora se apresentam. Por esta razão, a estratégia adoptada nas secções seguintes para modelação dos mapeamentos $MI \rightarrow P_i$, será a programação genética **multi-objectivo**.

6.2.9 Aprendizagem de redes neuronais do tipo *B-Spline* para modelar os 10 parâmetros

O comportamento das diferentes estratégias de aprendizagem face ao mapeamento $MI \rightarrow P_i$, estudado na secção 6.2.5, é generalizado aos restantes 9 mapeamentos necessários à construção do modelo de 2ª ordem apresentado na expressão (6.13). Nesta secção apresentam-se os procedimentos que conduziram à aprendizagem das redes neuronais para os 10 mapeamentos, utilizando-se a programação genética multi-objectivo.

Dado que as redes neuronais obtidas servirão para, conjuntamente com as medidas de identificação e os parâmetros *PID*, avaliarem os critérios de desempenho, opta-se nesta fase por não se efectuar qualquer alteração nos padrões de saída, como a que se apresentou na secção 6.2.7. Este facto tem que ser tido em conta na observação das medidas relativas de desempenho das aprendizagens; concretamente, dado existirem valores nulos para o 5º parâmetro, estas não poderão ser avaliadas para esse parâmetro.

Estabelecimento de vectores de objectivos prioridades e metas: Tal como aconteceu no estudo prévio realizado na secção 6.2.7, os vectores de objectivos, prioridades e de metas a atingir foram definidos com base nos valores resultantes da execução prévia dum lote de aprendizagens usando programação genética uni-objectivo. Nesta primeira fase da aprendizagem, optimizou-se o critério *BIC* tendo em conta o *MSE* no conjunto de treino, utilizando-se 20 inicializações por parâmetro com lotes de aprendizagens de 50 gerações por inicialização. Foram analisados os valores das medidas de desempenho tanto para os modelos que conduzem ao menor *BIC*, como para os que conduzem ao menor *MSE* no conjunto de teste. Verificou-se, no entanto, que na primeira situação, a discrepância entre os valores de treino e teste é muito elevada, mesmo para a medida de desempenho absoluta. Desta forma, optou-se por se apresentar na Tab. 6.16, as medidas de desempenho referentes aos modelos que conduzem ao menor *MSE* no conjunto de teste.

Parâmetros →		1	2	3	4	5
<i>MSE</i>	Treino	6.1E-02	1.9E-01	8.0E-02	3.9E-01	3.5E-01
	Teste	2.1E-01	2.3E-01	7.0E+00	2.1E-01	1.9E-01
<i>MSRE</i>	Treino	9.9E+01	2.6E+00	9.8E-04	2.8E+01	
	Teste	2.5E+02	2.4E-01	2.3E-02	3.2E-01	
%MRE	Treino	283.7	46.9	2.4	122.2	
	Teste	368.6	30.0	9.0	34.3	
<i>BIC</i>		-29.1	-68.9	-94.4	61.9	-28.7
Complexidade		43	14	23	31	12
Parâmetros →		6	7	8	9	10
<i>MSE</i>	Treino	2.8E-01	5.9E-03	1.7E-02	1.0E-02	3.6E-04
	Teste	2.8E-01	7.9E-03	1.4E-02	1.4E-02	7.2E-03
<i>MSRE</i>	Treino	1.9E+00	1.8E+02	1.4E+00	2.5E+00	1.0E-03
	Teste	6.5E-01	4.0E+01	1.8E+00	6.2E+00	9.3E-01
%MRE	Treino	30.5	297.1	59.9	68.6	1.8
	Teste	33.6	204.1	63.7	111.5	36.6
<i>BIC</i>		-20.3	-308.5	-227.4	-259.9	-384.0
Complexidade		18	20	20	21	52

Tab. 6.16 – Mapeamento $MI \rightarrow P_i$ usando-se algoritmos genéticos uni-objectivo para o modelo com menor *MSE* teste.

Os elevados valores das medidas relativas devem-se ao facto de algumas saídas objectivo estarem muito próximas de zero; esta situação é ilustrada na Fig. 6.6, onde se sobrepõem as saídas propagadas às saídas objectivo, para o conjunto de teste do 1º parâmetro (pior caso da Tab. 6.16).

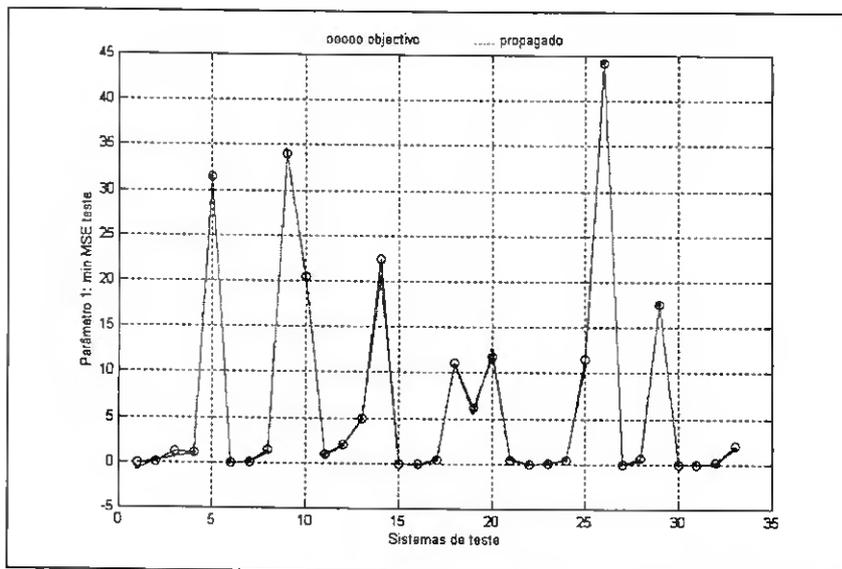


Fig. 6.6 – Padrões de Teste: objetivo e propagado pelo modelo para o qual MSE teste é mínimo.

Com base nos valores da Tab. 6.16 estabeleceram-se os vectores de objectivos, prioridades e valores das metas, que se registam na Tab. 6.17, e que servirão para a fase seguinte da aprendizagem utilizando programação genética multi-objectivo.

	Objectivos	MSE treino	Comp.
	Prioridades	3	2
Valores das metas para os 10 parâmetros	1	6E-3	45
	2	1E-2	20
	3	1E-2	22
	4	3E-2	40
	5	3E-2	25
	6	2E-2	20
	7	5E-4	25
	8	1E-3	30
	9	1E-3	25
	10	3E-5	40

Tab. 6.17 – Valores para os 10 parâmetros da aprendizagem de redes neuronais do tipo *B-Spline* recorrendo à programação genética multi-objectivo.

Obtenção de soluções não dominadas e soluções preferidas para os vários lotes de inicializações: Na aprendizagem recorrendo à programação genética multi-objectivo, para cada um dos 10 parâmetros, foram utilizadas 40 gerações aleatórias de populações iniciais

com 65 indivíduos, a partir das quais ocorreram aprendizagens com a duração de 70 gerações. Esta base inicial de trabalho foi para alguns parâmetros expandida, tanto no número de inicializações como no número de gerações do processo evolutivo.

Designando por m o número de inicializações e T_j o número de soluções não dominadas da inicialização j , apresenta-se na Tab. 6.18 a totalidade de soluções não dominadas e de soluções preferidas obtidas para cada parâmetro.

parâmetro	1	2	3	4	5	6	7	8	9	10
$\#\bigcup_{j=1}^m T_j$	1800	2074	2163	1713	1551	1549	1785	1858	980	858
$\#N.D.\left(\bigcup_{j=1}^m T_j\right)$	53	58	61	49	37	46	53	46	53	44
$\#PRF.\left(\bigcup_{j=1}^m T_j\right)$	2	34	28	8	2	3	16	14	1	1

Tab. 6.18 – Totais de soluções não dominadas e soluções preferidas para os 10 parâmetros.

Escolha duma solução por parâmetro: A aplicação do algoritmo de selecção de soluções preferidas conduz, para a maioria dos parâmetros, à selecção de mais do que uma solução (Tab. 6.18). Surge então a necessidade de se eleger uma solução por parâmetro que deverá, tanto quanto possível, ser exacta e ter boa capacidade de generalização. Desta forma, com base nos valores dos objectivos, prioridades e valores das metas que se definiram para a fase de aprendizagem (Tab. 6.17), estabeleceram-se novos valores para esta fase final de selecção (Tab. 6.19). Deve ser notado que na fase de selecção se inclui objectivos ligados ao conjunto de teste. Dado que a fase de aprendizagem está concluída, a inclusão de objectivos ligados ao conjunto de teste não desvirtua o objectivo deste conjunto; no entanto, este procedimento permitirá obter soluções com melhor capacidade de generalização.

A partir da base de soluções a que se refere a 2ª linha da Tab. 6.18, considerando os valores da Tab. 6.19, obtêm-se diferentes soluções não dominadas e preferidas, cujas totalidades se registam na Tab. 6.20.

Objectivos	<i>MSE</i> treino	Comp.	$\ w\ $	Cond(A)	<i>MSE</i> teste	
Prioridades	3	3	2	2	3	
Valores das metas para os 10 parâmetros	1	6E-3	45	100	100	6E-3
	2	1E-2	20	100	100	1E-2
	3	1E-2	22	100	100	1E-2
	4	3E-2	40	100	100	3E-2
	5	3E-2	25	100	100	3E-2
	6	2E-2	20	100	100	2E-2
	7	5E-4	25	100	100	5E-4
	8	1E-3	30	100	100	1E-3
	9	1E-3	25	100	100	1E-3
	10	3E-5	40	100	100	3E-5

Tab. 6.19 – Valores para a escolha da solução preferida com melhor capacidade de generalização.

parâmetro	1	2	3	4	5	6	7	8	9	10
$\# N.D. \left(\bigcup_{j=1}^m T_j \right)$	536	635	825	600	553	515	490	511	315	262
$\# PRF. \left(\bigcup_{j=1}^m T_j \right)$	13	102	88	41	33	46	45	49	51	28

Tab. 6.20 – Totais de soluções não dominadas e preferidas com as definições da Tab. 6.19.

Para cada parâmetro a solução adoptada foi escolhida de entre as soluções preferidas, sendo aquela que conduz ao **menor valor de *MSE* teste**.

Na parte superior da Tab. 6.21 apresentam-se as medidas de desempenho da solução preferida para cada parâmetro e na parte inferior apresentam-se os respectivos modelos.

Parâmetro →	1	2	3	4	5	6	7	8	9	10	
<i>MSE</i>	Treino	1.4E-01	7.6E-02	1.2E-01	1.7E-01	3.2E-01	1.7E-01	4.4E-03	1.1E-02	2.7E-03	1.9E-03
	Teste	1.4E-01	1.4E-01	6.6E+00	3.3E-01	2.9E-01	2.4E-01	5.4E-03	6.9E-03	1.6E-02	6.6E-03
<i>MSRE</i>	Treino	2.0E+02	1.9E-01	1.4E-03	2.1E+01		8.9E-02	5.3E+01	3.5E+00	1.4E+00	9.0E-04
	Teste	4.5E+02	2.6E-01	2.5E-02	1.2E+01		7.4E+00	2.6E+00	9.4E-01	4.1E+01	6.6E-02
% <i>MRE</i>	Treino	348.3	18.0	2.9	116.0		13.2	127.0	71.5	38.6	2.0
	Teste	547.7	28.0	9.4	101.6		83.9	68.1	57.3	189.2	11.1
<i>BIC</i>		-31.9	-55.5	-64.0	5.5	-0.9	-10.8	-304.9	-231.0	-226.1	-318.1
Complexidade		28	33	23	33	20	29	26	27	53	38
MODELOS	Nsm	1	1	2	1	2	2	1	1	2	2
	Var.	1 2	1 2	1 2	2 1	1	2 1	1 2	2 1	1 2	1
	Nint.	4 1	8 0	1 3	0 8	2	0 4	11 0	1 6	4 4	7
	Ord.	3 3	3 3	3 2	3 3	3	3 3	2 2	2 3	2 3	3
	Var.			1		2 1	1			1	1 2
	Nint.			3		0 4	5			17	5 2
	Ord.			2		3 1	3			2	3 2

Tab. 6.21 – Soluções preferidas para os 10 parâmetros.

6.2.10 Validação dos modelos para o conjunto base de experimentação

Nas secção anterior estudou-se a modelação por redes neuronais de cada parâmetro constituinte dos modelos de 2ª ordem para o *ITAE*, definido segundo a expressão (6.10) e tomando-se como base de trabalho o conjunto base de experimentação. A exactidão e capacidade de generalização das redes obtidas estão expressas na Tab. 6.21. Anteriormente (secção 6.2) tinha sido estudada uma estratégia alternativa segundo a qual o *ITAE* era obtido sem fixar a topologia.

Nesta secção irá ser tomado cada um dos processos do conjunto base de experimentação em separado e analisado o desempenho para ambas as estratégias:

- Topologia parcialmente fixa,
- Topologia livre.

Para cada processo, foram registadas medidas de erro para os seguintes critérios:

- I Modelo⁹: Para cada processo do conjunto base de experimentação é determinado um modelo em função dos parâmetros G , h e C , conforme a expressão (6.10) da secção 6.2.4. Conforme foi visto na secção 6.2.5, os parâmetros do referido modelo são determinados por redes neuronais. Com este critério de teste pretende-se, para cada processo do sistema base de experimentação, avaliar o erro entre o modelo a que se chegou na secção 6.2.4 e o modelo obtido pelas redes neuronais a que se refere a secção 6.2.5.
- II PID mínimo: Pretende-se avaliar a exactidão dos modelos no mínimo. Com este critério, irá ser medido o erro entre o PID minimizante dentro do conjunto de padrões utilizados para construir o modelo de 2ª ordem, e o PID minimizante calculado por um processo de minimização, aplicado tanto ao modelo neuronal obtido pela estratégia a), como pela estratégia b).
- III ITAE mínimo: Pretende-se avaliar a exactidão da variável dependente no ponto de mínimo. São confrontados os valores de $ITAE$ correspondentes aos PID a que se referem o ponto II.
- IV ITAE das vizinhanças: São tomados os valores de $ITAE$ correspondentes a todos os PID que intervieram na construção da estratégia a). A partir desses vectores PID serão calculados os correspondentes valores de $ITAE$ de acordo com ambas as estratégias. Os valores de $ITAE$ assim obtidos são confrontados com os correspondentes valores verdadeiros.

As medidas de erro provenientes de todos os critérios, à excepção do ponto III, são vectores; assim, designando por $m = [m_1 \dots m_n]^\dagger$ o vector objectivo, e por $m^* = [m_1^* \dots m_n^*]$ o correspondente vector estimado ou calculado, definem-se as medidas de erro usuais: MSE , $MSRE$, % de MRE , e adicionalmente, a % de NE dada por (6.24).

$$\%NE = 100 \frac{\|m^* - m\|}{\|m\|} \quad (6.24)$$

⁹ Este critério está definido apenas para a estratégia a)

[†] n será 10 para o ponto I, 3 para o ponto II; para o ponto IV, depende do processo em causa.

Para cada um dos processos do conjunto base de experimentação foram calculadas as medidas de erro referidas, para todos os critérios de I a IV no que respeita à estratégia a) e de II a IV no que se refere à estratégia b). Foram construídas as correspondentes tabelas, a partir das quais se geraram as tabelas resumo (Tab. 6.22 e Tab. 6.23) para as estratégias a) e b), respectivamente, com os valores máximos, mínimos e médios dos vários indicadores de desempenho estendidos a todos os processos do conjunto base de experimentação.

	MODELOS P_i (ponto I)				PID min (ponto II)			
	<i>MSE</i>	<i>MSRE</i>	<i>%MRE</i>	<i>%NE</i>	<i>MSE</i>	<i>MSRE</i>	<i>%MRE</i>	<i>%NE</i>
Máximo	16.0	530.0	963.6	59.3	2015.7	740.0	2708.3	2674.8
Mínimo	1.8E-06	5.2E-04	1.2	1.0E-01	0.0	5.1E-04	2.2	2.4
Médio	0.3	18.6	65.1	6.7	39.9	13.0	117.6	116.5
	ITAE vizinhança (ponto IV)				ITAE min (ponto III)			
	<i>MSE</i>	<i>MSRE</i>	<i>%MRE</i>	<i>%NE</i>	<i>MSE</i>	<i>MSRE</i>	<i>%MRE</i>	
Máximo	4.0E+03	1.7E+07	4.0E+05	3.4E+05	490.0	690.0	2622.9	
Mínimo	1.2E-05	1.6E-02	9.7	14.0	5.7E-06	2.5E-05	0.5	
Médio	65.8	2.3E+05	6888.9	5786.0	8.1	12.5	119.2	

Tab. 6.22 – Resumo das medidas de desempenho para a topologia parcialmente fixa.

Salienta-se que de acordo com a estratégia a) as medidas de erro referentes aos pontos II, III e IV utilizam 70% dos elementos do conjunto base de experimentação, correspondentes aos sistemas para os quais se encontraram o mínimo do *PID* dos respectivos modelos.

A leitura do desempenho obtido para cada processo, reflectido em parte nas tabelas resumo apresentadas (Tab. 6.22 e Tab. 6.23), permite concluir que os modelos obtidos têm um comportamento muito distinto face aos diferentes processos do conjunto base de experimentação. Tal observação é baseada no facto de em todos os critérios de ambas as estratégias de modelação se verificar grande discrepância entre os valores mínimo e máximo.

	<i>PID</i> min (ponto II)				<i>ITAE</i> min (ponto III)		
	<i>MSE</i>	<i>MSRE</i>	% <i>MRE</i>	% <i>NE</i>	<i>MSE</i>	<i>MSRE</i>	% <i>MRE</i>
Máximo	2.1	897.7	1811.2	415.7	41.5	6.4E+4	2.5E+4
Mínimo	3.1E-04	1.5E-03	3.1	2.5	3.4E-5	6.8E-03	8.2
Médio	0.6	19.7	175.0	147.9	5.9	1164.4	1674.8

	<i>ITAE</i> vizinhança (ponto IV)			
	<i>MSE</i>	<i>MSRE</i>	% <i>MRE</i>	% <i>NE</i>
Máximo	43.3	0.8	91.8	92.8
Mínimo	8.3E-06	0.1	28.8	34.6
Médio	2.8	0.5	63.4	66.4

Tab. 6.23 – Resumo das medidas de desempenho para a topologia livre.

6.3 REDES NEURONAIS PARA MAPEAMENTO DO TEMPO DE CRESCIMENTO

O sistema de auto-sintonia neuro-genética proposto na Fig. 3.2 acomoda a possibilidade de serem atendidos em simultâneo diferentes critérios de desempenho. Desta forma, para além do *ITAE* mencionado na secção anterior considera-se, na presente secção, a modelação do tempo de crescimento (5.14) da resposta do sistema, possibilitando-se assim o teste do sistema de auto-sintonia face a 2 critérios de desempenho normalmente complementares entre si. Nesta secção procede-se à construção do bloco “RN critério 2” da Fig. 3.2 referente ao tempo de crescimento, efectuando-se para tal a aprendizagem de redes neuronais com vista ao mapeamento de critério: $[F(1) F(2) k_c^* t_i^* t_d^*] \rightarrow TC^*$.

A normalização dos parâmetros *PID* resulta do estudo apresentado na secção 5.3.3; o tempo de crescimento é normalizado de acordo com a expressão (5.16). Após a filtragem mencionada na secção 5.3.3.1, seguida da transformação referida na secção 5.2.4, foram extraídos os padrões de aprendizagem obtendo-se 1840 padrões de treino e 788 padrões de teste (Tab. 5.10).

Os processos de aprendizagem levados a cabo utilizaram o algoritmo *ASMOD* [138], sendo a medida de desempenho da aprendizagem o *BIC* (2.53) referente ao conjunto de treino, definido alternativamente para o *MSE* ou para o *MSRE*. Em ambas as situações ocorreram

aprendizagens com e sem factor de regularização. O critério de paragem utilizado determina interrupção do processo na iteração em que o *BIC* cresce pela primeira vez, desde que até então não tenham ainda ocorrido 4 subidas consecutivas¹⁰ da medida de erro para o conjunto de teste¹¹. Terminado um processo de aprendizagem, a rede neuronal resultante é aquela que corresponde à penúltima iteração caso não tenha ocorrido *early stopping*, ou a respeitante à iteração anterior à primeira subida da medida de erro no conjunto de teste, no caso contrário.

Na Tab. 6.24 resumem-se as medidas de desempenho dos modelos obtidos, registando-se os respectivos factores de regularização λ , bem como os critérios que estiveram na base da optimização. Verificou-se que $\lambda = 5 \times 10^{-3}$ e $\lambda = 10^{-2}$ com o *BIC* definido para o *MSRE* conduziram a *early stopping*.

Critério		<i>BIC(MSE)</i>				<i>BIC(MSRE)</i>			
		λ	0	10^{-3}	5×10^{-3}	10^{-2}	0	10^{-3}	5×10^{-3}
<i>MSE</i>	Treino	4.6E-3	7.4E-3	7.9E-3	8.0E-3	1.2E-2	1.3E-2	1.2E-2	1.4E-2
	Teste	5.2E-3	5.2E-3	5.4E-3	5.5E-3	0.8E-2	0.8E-2	0.7E-2	0.8E-2
<i>MSRE</i>	Treino	3.3E-2	4.0E-2	4.4E-2	4.5E-2	1.8E-2	1.8E-2	1.7E-2	2.0E-2
	Teste	9.5E-2	4.8E-2	5.1E-2	5.1E-2	2.0E-2	2.0E-2	2.2E-2	2.1E-2
% <i>MRE</i>	Treino	11.3	11.8	12.3	12.2	8.1	8.4	8.0	8.8
	Teste	15.2	12.5	12.9	12.7	8.3	8.2	8.2	8.7
<i>BIC</i>		-8.9E+3	-8.5E+3	-8.4E+3	-8.4E+3	-6.9E+3	-6.8E+3	-6.7E+3	-6.5E+3
Complexidade		135	66	69	66	66	80	103	84

Tab. 6.24 – Aprendizagem de redes neuronais para mapeamento do tempo de crescimento.

¹⁰ *Early stopping*

¹¹ A medida de erro considerada é o MSE ou o MSRE dependendo da que foi utilizada na avaliação do *BIC*.

A escolha da rede neuronal que servirá de modelo para o tempo de crescimento, integrado no bloco “RN critério 2” da Fig. 3.2, faz-se com base nas leituras proporcionadas pela Tab. 6.24. A observação da referida tabela permite salientar os seguintes pontos principais:

- Utilização de factor de regularização – Os valores de qualquer critério de desempenho de aprendizagem não são sensivelmente alterados pela utilização de factor de regularização. Os vários valores que se obtêm pelo facto de se utilizarem factores de regularização diferentes estão todos dentro da mesma ordem de grandeza. Como evoluções típicas pode-se apontar o *MSE* avaliado tanto para o conjunto de treino como para o conjunto de teste referente ao *BIC(MSE)*; nestas situações a utilização de factores de regularização crescentes conduzem a ligeira pioria dos valores de desempenho. Por outro lado, a observação do *MSRE* e da *%MRE* para o *BIC(MSRE)* permite concluir que tal tendência deixa de ser evidente.
- Medida de erro utilizada no cálculo do *BIC* – Como referido anteriormente, as medidas de erro utilizadas no cálculo do *BIC* foram o *MSE* e o *MSRE*. Desta forma, foram levadas a cabo dois lotes principais de aprendizagens, segundo as quais se procuravam redes neuronais que conduziam, a menores valores de erro absoluto e a menores valores de erro relativo, respectivamente. Os resultados apresentados na Tab. 6.24 estão em consonância com os propósitos enunciados. De facto, os valores referentes ao *BIC(MSE)* para o *MSE* são globalmente inferiores aos correspondentes para o *BIC(MSRE)*. Observando as medidas de desempenho relativas, *MSRE* e *%MRE*, verifica-se que os menores valores ocorrem quando a aprendizagem utiliza *BIC(MSRE)*.
- Conjuntos de treino e teste – Tal como esperado, o valor de qualquer critério de desempenho calculado para o conjunto de treino não se afasta muito do correspondente valor para o conjunto de teste, sendo, dum modo geral inferior a este último. Verifica-se no entanto que por vezes o conjunto de teste apresenta valores inferiores, não tendo especial relevância tal facto. Pode-se concluir então que o conjunto de treino é representativo do universo de aprendizagem tendo as redes obtidas boas capacidades de generalização.

Com base nas observações acabadas de referir sobre os pontos principais do processo de aprendizagem, pode-se seleccionar a rede neuronal obtida na optimização do *BIC(MSRE)*

com factor de regularização $\lambda = 5 \times 10^{-3}$ como a mais adequada para a modelação do tempo de crescimento. De facto, esta rede é a que apresenta simultaneamente os menores valores para a % *MRE* tanto para o conjunto de treino como para o conjunto de teste; a estrutura da referida rede neuronal apresenta-se na Tab. 6.25.

Sub-Mod.	1	2	3	4	5	6	7	8	9
Var.	1	3	4	5	1 5	2 4	2 3	1 4	4 5
Nint.	0	1	0	3	0 0	0 0	3 0	0 0	4 3
Ord.	3	3	3	3	3 3	3 3	3 3	3 3	3 3

Tab. 6.25 – Modelo para o mapeamento *MI + PID* → *TC* utilizando o *ASM*OD.

Na Fig. 6.7 apresentam-se as evoluções das medidas de desempenho da aprendizagem referentes ao modelo seleccionado.

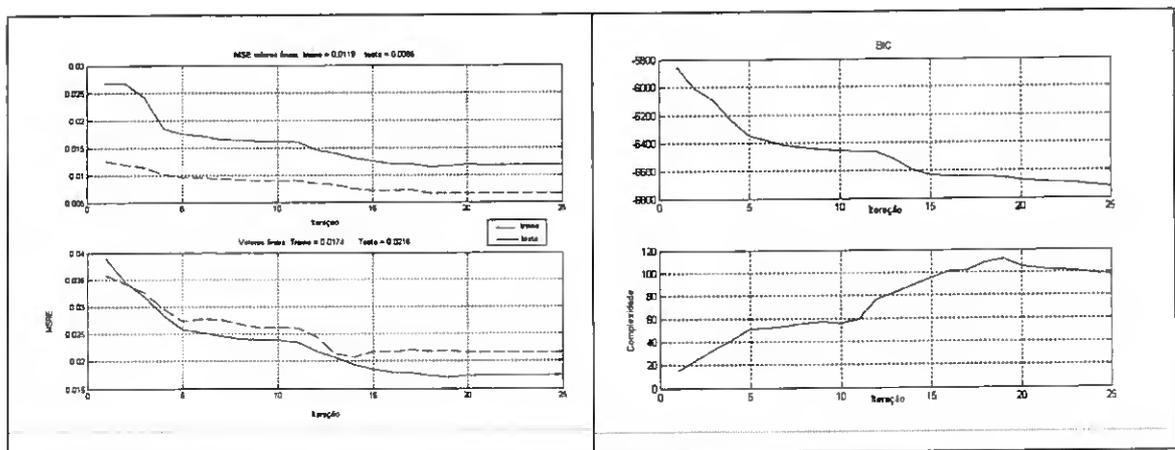


Fig. 6.7 – *BIC*, complexidade, *MSE* e *MSRE* para o mapeamento *MI + PID* → *TC* utilizando o *ASM*OD.

6.4 REDES NEURONAIS DE SINTONIA

Esta secção tem como objectivo a aprendizagem de 6 redes neuronais de sintonia, que são responsáveis pelo mapeamento das medidas de identificação em cada componente do vector *PID* minimizante do *ITAE* e do tempo de crescimento:

- $MI \rightarrow k_c^{ITAE}$, $MI \rightarrow t_i^{ITAE}$ e $MI \rightarrow t_d^{ITAE}$,

- $MI \rightarrow k_c^{TC}$, $MI \rightarrow t_i^{TC}$ e $MI \rightarrow t_d^{TC}$

6.4.1 Conjuntos de aprendizagem para os mapeamentos de sintonia

O mapeamento das medidas de identificação nas componentes optimizadas do vector *PID* é utilizado para a sintonia em linha do controlador. Cada par de medidas de identificação de cada processo é mapeado nas 3 componentes do *PID* óptimo para cada critério em estudo, *ITAE* e tempo de crescimento; desta forma, a base de 110 processos origina igual número de padrões de aprendizagem. Os valores óptimos de ambos os critérios determinados na secção 5.3.1 são utilizados na presente secção como valores objectivo dos conjuntos de aprendizagem dos mapeamentos de sintonia.

De acordo com o esquema de auto-sintonia proposto na Fig. 3.2, os valores óptimos a que se refere a presente secção deverão coincidir com os mínimos encontrados pelo bloco “Algoritmo Genético” da figura mencionada. A optimização levada a cabo pelo referido bloco poderá ser uni ou multi-objectivo; na presente situação considera-se uni-objectivo, sendo a função objectivo uma soma ponderada de critérios. Não estando, à partida, definida a função objectivo não será possível com exactidão determinar em diferido os vectores *PID* óptimos. Vão, assim, ser preparados conjuntos de aprendizagem para mapeamentos das medidas de identificação nos parâmetros *PID* óptimos para o *ITAE* e para o tempo de crescimento num total de 6 mapeamentos distintos.

A matriz das medidas de identificação é transformada de acordo com o método apresentado na secção 5.2.4 para constituir os padrões de entrada. São extraídos aleatoriamente 30% dos padrões para teste, obtendo-se a distribuição de padrões da Tab. 6.26.

TREINO	TESTE
77 padrões	33 padrões

Tab. 6.26 – Conjuntos de aprendizagem para mapeamentos de sintonia.

Foi utilizado o algoritmo *ASMOD* [138] para aprendizagem de redes neuronais para todos os mapeamentos.

6.4.2 Redes neuronais de sintonia óptima para o ITAE

Nas Fig. 6.8, Fig. 6.9 e Fig. 6.10 apresentam-se as evoluções das medidas de desempenho da aprendizagem dos mapeamento, $MI \rightarrow k_c^{ITAE}$, $MI \rightarrow t_i^{ITAE}$ e $MI \rightarrow t_d^{ITAE}$, respectivamente.

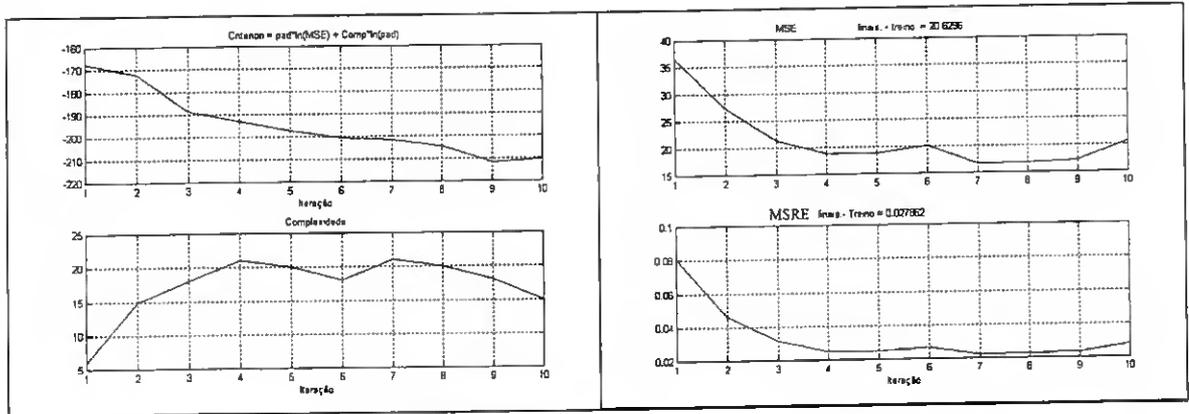


Fig. 6.8 – *BIC*, *Complexidade*, *MSE* e *MSRE* para o mapeamento $MI \rightarrow k_c^{ITAE}$.

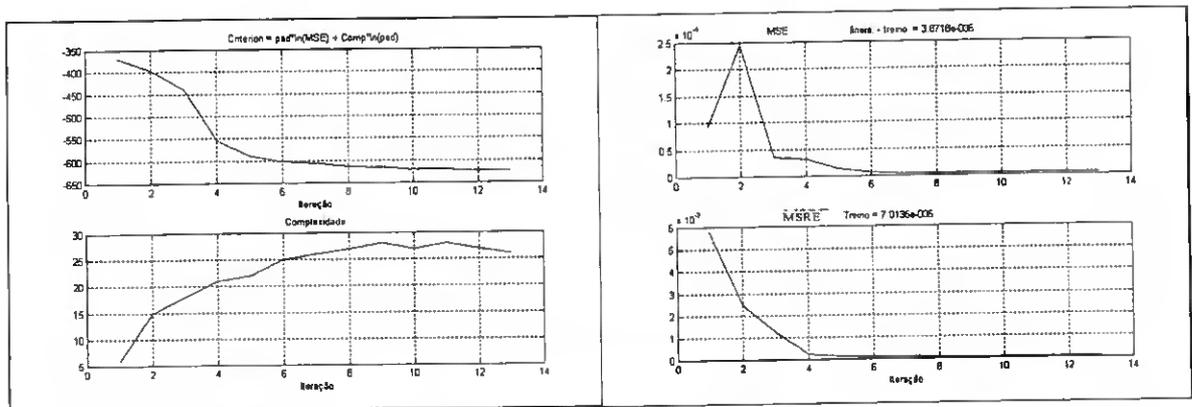


Fig. 6.9 – *BIC*, *Complexidade*, *MSE* e *MSRE* para o mapeamento $MI \rightarrow t_i^{ITAE}$.

Na Tab. 6.30 aqui apresentam-se as descrições dos modelos obtidos para as três redes neuronais de sintonia, referentes à iteração ocorrida antes da primeira subida do *BIC*.

Para cada modelo apresentado é mencionado o número da iteração respectiva, a totalidade de sub-modelos constituintes, a sua descrição em termos das variáveis utilizadas, o número de nós interiores e ordens das variáveis utilizadas.

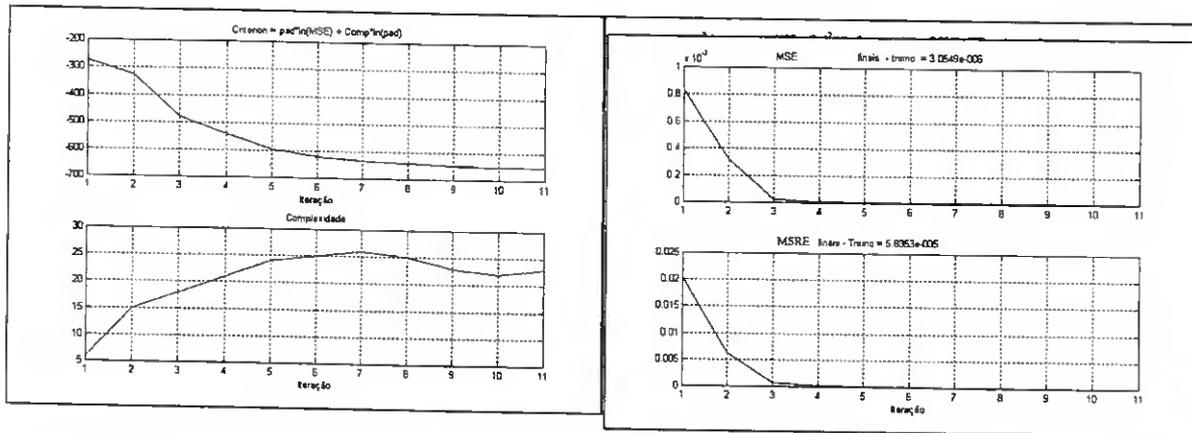


Fig. 6.10 – *BIC*, *Complexidade*, *MSE* e *MSRE* para o mapeamento $MI \rightarrow t_d^{ITAE}$.

Os processos de aprendizagem descritos com vista à obtenção das redes neuronais de sintonia óptima para o *ITAE* conduzem a modelos com grande exactidão. O problema foi posto no sentido de se verificar a existência e exactidão de tais modelos, independentemente das respectivas capacidades de generalização; assim, o conjunto de teste foi omitido durante todo o procedimento.

Os modelos obtidos para cada mapeamento apresentam estruturas distintas, não se podendo definir nenhum padrão comum com a informação de que se dispõe.

6.4.3 Redes neuronais de sintonia óptima para o tempo de crescimento

Nesta secção apresentam-se os procedimentos utilizados para se obter os mapeamentos de sintonia $MI \rightarrow k_c^{TC}$, $MI \rightarrow t_i^{TC}$ e $MI \rightarrow t_d^{TC}$.

Contrariamente ao que se verificou nos mapeamentos de sintonia para o *ITAE*, na presente situação vão ser testadas as capacidades de generalização das redes neuronais obtidas.

Tal como aconteceu na secção 6.3, vai-se testar a influência do factor de regularização no desempenho da aprendizagem. De forma análoga à referida secção utiliza-se o *BIC* (2.53) avaliado alternativamente para o *MSE* ou para o *MSRE*, como medida a otimizar no decurso da aprendizagem. O processo de aprendizagem termina na iteração em que o *BIC* sobe pela primeira vez caso não tenha ocorrido *early stopping*.

Os desempenhos dos 3 lotes de aprendizagem dos mapeamentos $MI \rightarrow k_c^{TC}$, $MI \rightarrow t_i^{TC}$ e $MI \rightarrow t_d^{TC}$ estão registados respectivamente nas Tab. 6.27, Tab. 6.28 e Tab. 6.29.

Critério		<i>BIC(MSE)</i>				<i>BIC(MSRE)</i>			
		0	10 ⁻³	5x10 ⁻³	10 ⁻²	0	10 ⁻³	5x10 ⁻³	10 ⁻²
<i>MSE</i>	Treino	6.4E-4	11.2	11.3	11.7	6.8E-4	36.9	34.2	43.5
	Teste	34.5	14.2	13.5	13.0	34.6	21.6	16.9	24.8
<i>MSRE</i>	Treino	6.9E-5	0.2	0.2	0.2	6.6E-5	4.8E-2	3.6E-2	5.8E-2
	Teste	0.2	0.2	0.2	0.2	0.2	5.1E-2	3.4E-2	6.4E-2
% <i>MRE</i>	Treino	0.5	34.9	37.1	39.5	0.5	13.6	9.2	15.4
	Teste	11.8	31.7	33.4	35.6	11.8	15.1	10.5	16.9
<i>BIC</i>		-379.4	216.1	217.2	219.8	-562.9	-203.1	-213.2	-192.8
Complexidade		43	7	7	7	41	7	10	6

Tab. 6.27 – Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow k_c^{TC}$.

A observação da Tab. 6.27 permite verificar que as redes neuronais obtidas para o mapeamento $MI \rightarrow k_c^{TC}$ conduzem a valores da % *MRE* para o conjunto de treino superiores a 10% na maior parte dos casos; exceptuando-se duas situações:

- Não utilização de factor de regularização,
- Factor de regularização $\lambda = 5 \times 10^{-3}$ quando a aprendizagem decorre com optimização do *BIC* avaliado para o *MSRE*.

Os elevados valores das medidas de desempenho de aprendizagem verificados na maior parte dos casos pode-se atribuir às baixas complexidades das redes neuronais obtidas, (em todos os casos inferior a 10).

A diferença essencial entre as duas situações apontadas em a) e b) relaciona-se com os valores dos pesos; tal como seria de esperar, a ausência de factor de regularização leva a situações em que os pesos chegam a 4 ordens de grandeza superiores. Assim, embora as

duas situações do ponto a) correspondam, a redes neuronais bastante mais exactas, (0.5% contra 11% de %MRE treino), a situação do ponto b) conduz a uma rede neuronal com melhor capacidade de generalização (10.5% contra 11.8% de %MRE teste).

Com base nestas observações a rede neuronal para modelação do mapeamento $MI \rightarrow k_c^{TC}$ é a correspondente ao ponto b) estando sua estrutura está registada na 4ª linha da Tab. 6.30.

Critério		$BIC(MSE)$				$BIC(MSRE)$			
		0	10^{-3}	5×10^{-3}	10^{-2}	0	10^{-3}	5×10^{-3}	10^{-2}
MSE	Treino	5.1E-5	5.1E-5	5.1E-5	5.2E-5	5.2E-5			
	Teste	6.2E-5	6.2E-5	6.5E-5	6.9E-5	6.7E-5			
$MSRE$	Treino	4.0E-3	3.4E-3	2.7E-3	4.4E-3	1.4E-3			
	Teste	5.9E-3	6.1E-3	7.2E-3	9.0E-3	1.3E-2			
% MRE	Treino	4.1	4.0	3.7	4.3	3.1			
	Teste	5.5	5.6	6.0	6.5	6.6			
BIC		-735.5	-735.4	-734.8	-732.9	-469.8			
Complexidade		6	6	6	6	8			

Tab. 6.28 – Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow t_i^{TC}$.

Como primeira leitura da Tab. 6.28 pode-se apontar o facto das medidas de desempenho serem insensíveis à utilização do factor de regularização. Nas aprendizagens em que se optimizava o $BIC(MSRE)$, para a precisão utilizada, não se verificou diferenças com as alterações do factor de regularização. Na situação em que se optimizou o $BIC(MSE)$ as variações que se podem observar pelo facto de diferentes factores de regularização serem utilizados, não são significativas.

As duas redes neuronais obtidas com melhor capacidade de generalização são as correspondentes à aprendizagem com optimização de $BIC(MSE)$ nas seguintes situações:

- a) Sem factor de regularização,
- b) Com factor de regularização $\lambda = 10^{-3}$.

Embora a situação a) corresponda a um modelo com melhor capacidade de generalização em termos das medidas relativas, a situação b) para além de apresentar melhor capacidade de generalização sob o ponto de vista do *MSE* conduz a modelos mais precisos. Verifica-se contudo que as diferenças apontadas são ligeiras em todas as situações.

Desta forma, a rede neuronal escolhida para modelação do mapeamento $MI \rightarrow t_i^{TC}$ é a correspondente ao ponto b), cuja estrutura se apresenta na 5ª linha da Tab. 6.30.

Critério		<i>BIC(MSE)</i>				<i>BIC(MSRE)</i>			
		λ	0	10^{-3}	5×10^{-3}	10^{-2}	0	10^{-3}	5×10^{-3}
<i>MSE</i>	Treino	1.7E-5	1.8E-5	2.4E-5	2.1E-5	1.7E-5	1.7E-5	1.7E-5	1.7E-5
	Teste	4.0E-5	3.5E-5	5.6E-5	5.1E-5	4.0E-5	3.9E-5	3.7E-5	3.7E-5
<i>MSRE</i>	Treino	1.4E-3	1.5E-3	2.2E-3	2.0E-3	1.4E-3	1.4E-3	1.4E-3	1.4E-3
	Teste	1.6E-2	6.6E-3	2.3E-2	1.6E-3	1.4E-2	1.2E-2	9.2E-3	8.0E-3
% <i>MRE</i>	Treino	3.0	3.2	3.8	3.7	3.0	3.0	3.0	3.0
	Teste	6.8	5.5	8.4	7.5	6.8	6.6	6.3	6.2
<i>BIC</i>		-784.8	-780.8	-759.4	-758.2	-445.4	-441.0	-444.9	-444.5
Complexidade		14	14	14	16	14	15	14	14

Tab. 6.29 – Aprendizagem de redes neuronais para o mapeamento $MI \rightarrow t_d^{TC}$.

A Tab. 6.29 permite verificar que a melhor capacidade de generalização, tanto no que se refere ao erro absoluto como o que respeita à % *MRE* é conseguida com a aprendizagem que optimiza o *BIC(MSE)*, quando se utiliza um factor de regularização $\lambda = 10^{-3}$. No que se refere à exactidão do modelo correspondente, verifica-se que os valores de desempenho

respeitantes a essa situação estão dentro dos menores apresentados na tabela. Desta forma, a exactidão do modelo pode ser considerada boa sendo então escolhido para modelar o mapeamento $MI \rightarrow t_d^{TC}$; a estrutura correspondente está registada na 6ª linha da Tab. 6.30.

Na Tab. 6.30 apresentam-se as estruturas das redes neuronais escolhidas para os 6 mapeamentos nos parâmetros *PID* óptimos para o *ITAE* e tempo de crescimento.

Map.	Iter.	Nsm	Var.	Nint.	Ord.	Var.	Nint.	Ord.	Var.	Nint.	Ord.
$MI \rightarrow k_c^{ITAE}$	9	1	1 2	3 0	3 3						
$MI \rightarrow t_i^{ITAE}$	12	3	1	1	2	2	3	3	1 2	3 0	3 3
$MI \rightarrow t_d^{ITAE}$	10	2	2	1	3	1 2	3 0	3 3			
$MI \rightarrow k_c^{TC}$	7	2	1	0	2	1 2	1 0	3 2			
$MI \rightarrow t_i^{TC}$	3	2	1	1	3	2	0	2			
$MI \rightarrow t_d^{TC}$	8	2	1	2	3	1 2	0 0	3 3			

Tab. 6.30 – Modelos para os 6 mapeamentos de sintonia óptima para os critérios de desempenho.

6.5 CONCLUSÕES

Este capítulo, juntamente com os dois capítulos anteriores, encerra o estudo sobre a modelação neuronal de processos quando sujeitos ao controlador *PID*. Este problema foi dividido em duas etapas principais:

- Escolha do tipo de rede neuronal (capítulo 4).
- Determinação do conjunto base de experimentação, selecção da estratégia mais adequada à aprendizagem e obtenção dos modelos respectivos (capítulos 5 e 6).

Enquanto o estudo levado a cabo no capítulo 4 teve em conta um processo invariante no tempo, o presente capítulo trata a situação em que o processo é variante no tempo. Desta forma houve a necessidade de se definir um conjunto base de experimentação para o qual o

controlador é testado, bem como estruturas de vectores *PID* adequados às sintonias pretendidas; estes aspectos foram tratados no capítulo 5.

A aprendizagem de redes neuronais efectuada no presente capítulo teve como objectivo a determinação de modelos para dois tipos de blocos do sistema auto-adaptativo (ver secção 3.3):

- Bloco de modelação de critério,
- Bloco de sintonia.

Para a modelação dos critérios de desempenho foram construídos 2 blocos referentes ao *ITAE* e ao tempo de crescimento.

No que se refere à modelação do *ITAE* foram delineadas duas estratégias:

- Modelo de estrutura livre,
- Modelo de estrutura parcialmente fixa.

Nas secções 6.2 a 6.2.9 a aprendizagem das redes neuronais centrou-se na obtenção de modelos cujo desempenho se pretende tão bom quanto possível, tendo em conta a totalidade de processos do conjunto base de experimentação. Por último, na secção 6.2.10, a modelação do *ITAE* é encerrada com a análise comparativa de cada modelo referente a cada processo, confrontando-se os resultados que se obtêm utilizando a estrutura livre com aqueles provenientes da metodologia que utiliza estrutura parcialmente fixa. Desta análise conclui-se que a estrutura livre é preferível relativamente à estrutura parcialmente fixa no que respeita à modelação do *ITAE*; assim sendo, a modelação do tempo de crescimento é feita utilizando apenas a metodologia de estrutura livre.

Pode-se concluir ainda na secção 6.2.10 que, para ambas as estratégias estudadas (topologia parcialmente fixa e topologia livre) se conseguem razoáveis modelos para alguns elementos do conjunto base de experimentação. Contudo, existem elementos desse conjunto para os quais não foi possível obter bons resultados com qualquer dos métodos.

No que diz respeito à pesquisa de redes neuronais de estrutura parcialmente fixa para o *ITAE*, foram testados comparativamente 4 algoritmos/variantes de aprendizagem: *ASMOD*, programação genética uni-objectivo com duas variantes, para procura de modelo com

menor *BIC* e para procura de modelo com menor *MSE* no conjunto de teste e por último, programação genética multi-objectivo com procura do modelo com menor *BIC* dentro das soluções preferidas. Os resultados que se obtiveram permitiram concluir que o algoritmo que utiliza a programação genética multi-objectivo conduz a melhores modelos.

No que respeita à modelação do tempo de crescimento com a estratégia da estrutura livre, foi testada a aprendizagem com o *BIC* avaliado para uma medida de erro relativo alternativamente à utilização do *BIC* avaliado para uma medida de erro absoluto. Com vista à procura de modelos com boa capacidade de generalização foram utilizados diferentes valores para o factor de regularização. Os testes obtidos para a modelação do tempo de crescimento revelaram a capacidade de obtenção de modelos com boa exactidão e capacidade de generalização.

A modelação do bloco de sintonia consistiu na aprendizagem de redes neuronais para mapeamento das medidas de identificação dos processos do conjunto base de experimentação, nos valores óptimos dos parâmetros *PID* para os dois critérios de desempenho: *ITAE* e tempo de crescimento. Estes processos de aprendizagem revelaram um peso computacional consideravelmente menor quando comparado com o que se verificou para a modelação dos critérios. O algoritmo *ASMOD* foi utilizado para a aprendizagem das redes neuronais de sintonia, tendo conduzido a modelos com boas medidas de desempenho.

7. SISTEMA DE SINTONIA AUTOMÁTICA

7.1 INTRODUÇÃO

No capítulo 3 apresentou-se uma arquitectura adaptativa para sintonia automática do controlador *PID*. Com base no diagrama funcional da Fig. 3.2 desse capítulo, descreveram-se os principais blocos envolvidos na identificação do processo, modelação dos critérios de desempenho, adaptação dos modelos, optimização dos parâmetros do controlador e sintonia do mesmo. Ainda no capítulo 3 é estudado o bloco referente à optimização genética; os capítulos 4, 5 e 6 debruçam-se sobre os blocos de modelação tanto dos critérios de desempenho quanto dos parâmetros *PID* de sintonia optimizados para o *ITAE* e para o tempo de crescimento.

O presente capítulo incide sobre uma implementação do referido diagrama funcional no ambiente de simulação dinâmica, MATLAB [134] [139], com vista ao teste da metodologia proposta. Assim, na secção 7.2 apresenta-se o simulador desenvolvido bem como os principais eventos que levam ao seu funcionamento adaptativo.

O funcionamento deste simulador é ilustrado nas secções 7.3 e 7.4 onde se apresentam exemplos de sintonia adaptativa optimizada para os critérios de desempenho em estudo. Na secção 7.3 os referidos critérios são estudados isoladamente, enquanto que na secção 7.4 os mesmos critérios são considerados em simultâneo, definindo-se a importância de cada um recorrendo-se a um factor de ponderação.

O capítulo termina com a secção 7.5, onde se apresentam as conclusões, referindo-se as consequências da optimização preferencial de cada critério de desempenho, no perfil da resposta do sistema à onda quadrada.

7.2 SIMULADOR NEURO-GENÉTICO

A simulação dinâmica da arquitectura da Fig. 3.2 foi implementada em MATLAB [134], recorrendo-se ao ambiente de visualização gráfica e simulação disponível no SIMULINK [139]. Desta forma, o recurso a esta ferramenta possibilitou o desenvolvimento do modelo "Auto-sintonia neuro-genética do controlador *PID*", que se apresenta na Fig. 7.1.

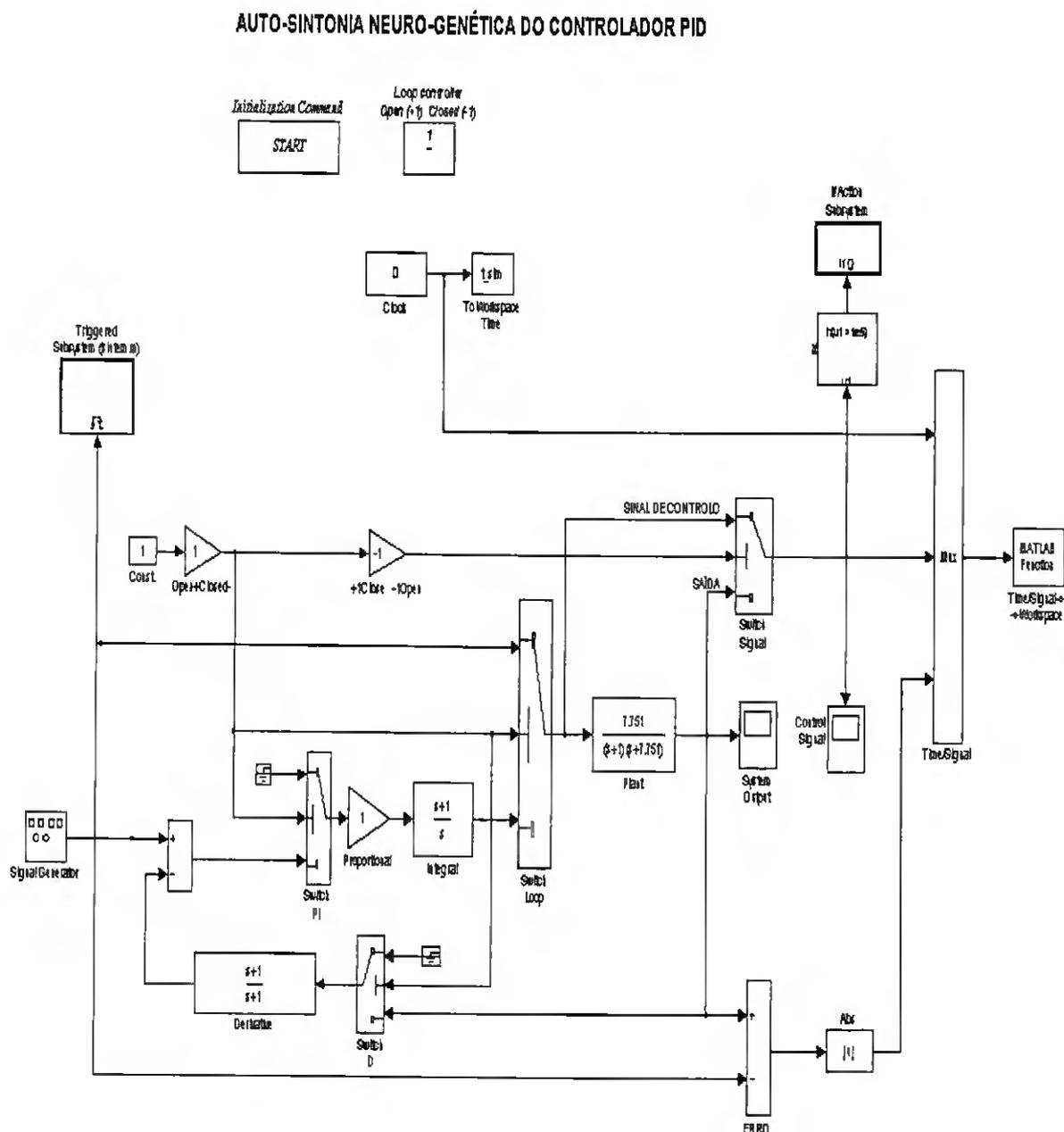


Fig. 7.1 – Modelo SIMULINK para auto-sintonia neuro-genética do controlador *PID*.

Tendo em conta o método proposto, este simulador está preparado para acomodar processos variantes no tempo. Assim, eventuais alterações no processo permitem que se proceda à sintonia do controlador, sem que haja a necessidade de se abrir a malha.

O processo de auto-sintonia a que se refere o sub-sistema A da Fig. 3.2, implementado no modelo da Fig. 7.1, implica o cumprimento das tarefas registadas na Tab. 7.1.

• IDENTIFICAÇÃO
<ul style="list-style-type: none"> • ADAPTAÇÃO <ul style="list-style-type: none"> • Modelos dos critérios • Optimização de parâmetros • Modelos de sintonia
• SINTONIA

Tab. 7.1 – Tarefas para auto-adaptação do controlador neuro-genético.

IDENTIFICAÇÃO: Esta etapa é cumprida com igual sucesso tanto em malha aberta (no início do funcionamento do sistema), como em malha fechada, ou seja, durante o normal funcionamento do mesmo. O cálculo das medidas de identificação nestas duas situações de funcionamento, malha aberta e malha fechada, recorre respectivamente às expressões (3.8) e (3.10). O cálculo correcto das medidas de identificação requer que o regime transitório resultante da transição na onda quadrada seja completamente extinto; tal situação limita a frequência máxima para a onda quadrada de entrada. Em termos práticos, a identificação do processo utiliza o sinal de controlo ou de saída, desde o instante imediatamente anterior a uma transição da onda quadrada, até ao instante imediatamente anterior à transição seguinte. Na fase de teste do simulador, os processos utilizados fazem parte do conjunto base de experimentação; desta forma, paralelamente à utilização das expressões (3.8) e (3.10) para avaliação numérica das medidas de identificação em linha, recorre-se à determinação analítica das mesmas, utilizando-se a expressão (3.5). A diferença entre as medidas de identificação avaliadas numericamente em linha e as correspondentes analíticas deve ser tão pequena quanto possível. O simulador desenvolvido testa essa diferença e só permite a ocorrência das etapas seguintes (adaptação e sintonia), caso esta seja inferior a determinado limite; na presente implementação fixou-se 2% de acordo com a expressão (7.1).

$$\left| \frac{F_n(\sigma) - F_a(\sigma)}{F_a(\sigma)} \right| < 2\% \quad (7.1)$$

$F_n(\sigma)$ e $F_a(\sigma)$ representam respectivamente os valores das medidas de identificação calculadas numérica e analiticamente. Imposição análoga, (7.1), é feita no que se refere à avaliação numérica de K_p e de T_T , conforme a secção 3.2.1.

ADAPTAÇÃO: Esta etapa constitui uma preparação para a etapa seguinte de sintonia e só tem lugar caso a etapa anterior de identificação tenha sido bem sucedida. Tal como é sugerido pela Tab. 7.1, a etapa de adaptação compreende a execução sequencial de 3 sub-etapas, respectivamente, a adaptação dos modelos dos critérios, a optimização dos parâmetros do controlador e a adaptação dos modelos de sintonia tendo em conta os parâmetros optimizados.

Modelos dos critérios: A adaptação dos modelos dos critérios consiste na actualização dos pesos das redes neuronais que modelam os respectivos critérios, com base no padrão obtido em linha. Este padrão é composto pelas medidas de identificação determinadas na etapa anterior juntamente com o vector PID actual. Como ficará explícito, as redes neuronais dos critérios, adaptadas, são usadas na sub-etapa seguinte de optimização.

Optimização de parâmetros: A sub-etapa de optimização utiliza o algoritmo genético de população real para determinação dos parâmetros do controlador optimizados. A função objectivo f consiste numa soma ponderada dos critérios de desempenho em estudo de acordo com a expressão (7.2).

$$f(PID, MI) = \sum_{i=1}^n w_i g_i(PID, MI) \quad (7.2)$$

No presente capítulo consideram-se $n = 2$ critérios de desempenho: $g_1 \equiv ITAE$ e $g_2 \equiv TC$. O vector de ponderação w é arbitrado em cada simulação permitindo dar mais ou menos importância a cada critério de desempenho em estudo.

Os valores dos critérios de desempenho, para cada indivíduo, são calculados recorrendo-se à propagação do padrão actual pela rede neuronal do critério respectivo, adaptada de acordo com a sub-etapa anterior. Para que a presente aplicação seja viável em tempo real, o tempo de execução do algoritmo genético é fundamental. Desta forma, a constituição da

população inicial é feita de tal forma que contenha algum indivíduo que constitua uma razoável aproximação da solução óptima. Assim, é colocado na população inicial o vector *PID* resultante da propagação das medidas de identificação pelas redes neuronais de sintonia. Sendo esta uma boa estimativa da solução óptima, o número de gerações necessárias para se obter o verdadeiro óptimo é menor, viabilizando-se o método numa situação de tempo real.

Os parâmetros utilizados na execução do algoritmo genético estão registados na Tab. 3.2. Para a aplicação em curso, ensaios preliminares permitiram verificar que, durante o procedimento evolutivo, as variáveis independentes deixam de sofrer alteração antes de completadas as 60 gerações a que se refere a Tab. 3.2; assim, foi possível diminuir o número de gerações para 40, com a vantagem evidente que daí resulta para o processo adaptativo.

Modelos de sintonia: As medidas de identificação propagadas pelas 3 redes neuronais de sintonia conduzem a um vector *PID* diferente do óptimo que se obteve na sub-etapa anterior. O erro daí resultante, juntamente com as medidas de identificação serve para adaptar as 3 redes neuronais de sintonia.

SINTONIA: As medidas de identificação são propagadas pelas redes neuronais de sintonia adaptadas de acordo com a sub-etapa anterior, originando o novo vector *PID* que é aplicado ao controlador.

A descrição acabada de apresentar permite deduzir o seguinte algoritmo:

1. Determinação numérica das medidas de identificação e dos critérios de desempenho; captura do *PID* actual;
2. Propagação das medidas de identificação e *PID* actual pelas redes de modelação dos critérios, avaliação dos respectivos critérios e correspondentes erros relativamente aos valores determinados em 1;
3. Adaptação das redes de modelação dos critérios com base nos respectivos erros do ponto 2;

4. Propagação das medidas de identificação pelas redes de sintonia e obtenção duma estimativa do PID_o^* ;
 5. Constituição da população inicial para o algoritmo genético com base no PID do ponto anterior;
 6. Execução da optimização com função objectivo dada por (7.2) e nas circunstâncias aí descritas, determinação do PID óptimo (PID_o);
 7. Avaliação do erro $PID_o - PID_o^*$;
 8. Utilização do erro do ponto 7, para adaptação das redes de sintonia;
 9. Propagação das medidas de identificação pelas redes de sintonia adaptadas para obtenção do novo vector PID .
-
10. Sintonia do controlador com o novo vector PID .

O simulador desenvolvido permite, para além da validação das medidas de identificação determinadas em linha por comparação com as correspondentes determinadas analiticamente, verificar se o sinal utilizado atingiu o regime estacionário; se tal não tiver acontecido a identificação não será feita. As situações de instabilidade provocadas, por exemplo, por variação brusca de algum parâmetro do processo, são detectadas pelo simulador fazendo com que o sistema seja posto em malha aberta, até que ocorra nova transição na onda quadrada de entrada.

7.3 SINTONIA OPTIMIZADA PARA OS CRITÉRIOS DE DESEMPENHO ISOLADOS

Nesta secção exemplifica-se o funcionamento adaptativo, do simulador proposto, quando se pretende obter auto-sintonia do sistema de controlo optimizada para cada um dos critérios de desempenho, considerados separadamente. Para tal, apresentam-se resultados mais pormenorizados para um sub-conjunto do conjunto base de experimentação, para o qual se testa o comportamento adaptativo do sistema de auto-sintonia neuro-genética da Fig. 7.1.

7.3.1 Verificação da estabilidade do sistema

Pretende-se determinar de que forma os modelos neuronais obtidos para os critérios de desempenho, juntamente com as correspondentes redes neuronais de sintonia, conduzem a sistemas estáveis quando integrados no modelo de simulação a que se refere a Fig. 7.1; este teste teve em conta apenas o tempo de crescimento.

Nas secções 6.3 e 6.4.3 seleccionaram-se respectivamente, a rede neuronal de modelação do tempo de crescimento e 3 redes neuronais de sintonia óptima para o referido critério. Nesta sequência, para se cumprir o objectivo enunciado, procedeu-se ao funcionamento do sistema de auto-sintonia durante 450s, para cada um dos processos do conjunto base de experimentação, considerando-se à entrada uma onda quadrada de amplitude unitária e período $T=160s$. Nesta fase inicial de validação estática dos modelos considerou-se ausência de adaptação das redes neuronais envolvidas.

Na Fig. 7.2 regista-se, como exemplo tipo, a evolução da resposta à onda quadrada bem como a evolução do respectivo tempo de crescimento, para o processo¹ do conjunto base de experimentação com função de transferência dada por (7.3).

$$G(s) = \frac{e^{-0.306s}}{(1+s)(1+0.242s)} \quad (7.3)$$

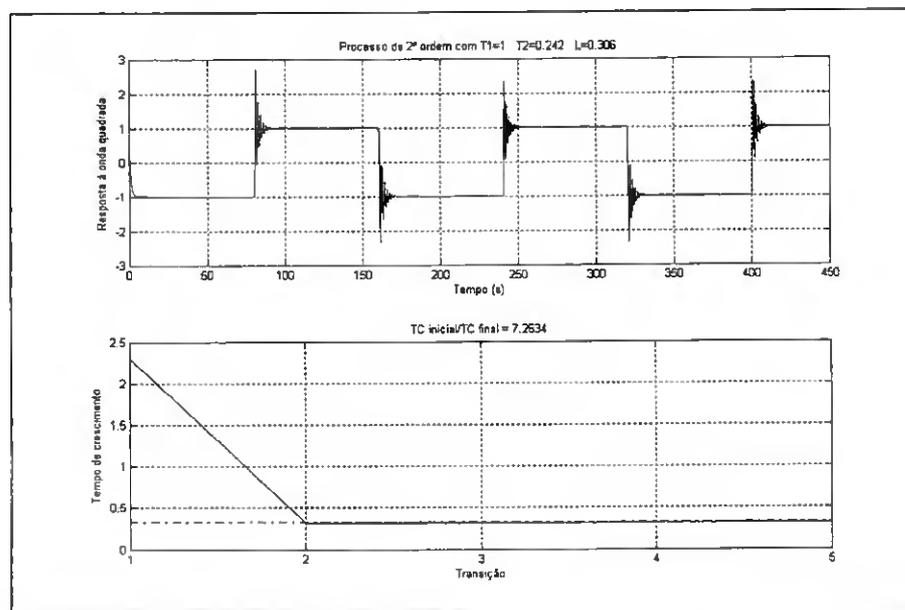


Fig. 7.2 – Sintonia sem adaptação para o processo com $T_1=1$, $T_2=0.242$ e $L=0.306$.

¹ Processo na posição 55.

Na metade inferior da Fig. 7.2 representa-se por um traço interrompido o mínimo do tempo de crescimento determinado de acordo com a secção 5.3.1. para o processo em causa. A observação desta figura mostra que o mínimo assinalado é rapidamente atingido e confirma a tendência para respostas com elevadas sobrelevações, quando se otimiza o tempo de crescimento.

Seja p o número de processos dum sub-conjunto retirado do conjunto base de experimentação. Para cada processo, designando por m^* o mínimo do tempo de crescimento determinado na secção 5.3.1 e por m o mínimo dos valores de tempo de crescimento ocorridos durante um dado horizonte de sintonia, define-se d_{min} como sendo a aproximação percentual ao mínimo de acordo com a expressão (7.4).

$$d_{min} (\%) = \frac{100}{p} \sum_{j=1}^p \frac{m_j - m_j^*}{m_j^*} \quad (7.4)$$

Para o exemplo apresentado na Fig. 7.2, $d_{min} (\%) = -3.4\%$.

Nas condições apontadas, pretende-se concluir até que ponto os modelos obtidos conduzem a sistemas estáveis. Assim, registam-se na Tab. 7.2 as percentagens de sistemas estáveis e instáveis observados dentro do conjunto base de experimentação quando se utiliza o sistema de auto-sintonia da Fig. 7.1, bem como o valor de d_{min} avaliado para o sub-conjunto de sistemas estáveis.

	Total	%	$d_{min} (\%)$
Sistemas estáveis	101	92	0.8
Sistemas instáveis	9	8	

Tab. 7.2 – Verificação da estabilidade utilizando-se sintonia otimizada para o tempo de crescimento.

A percentagem de sistemas estáveis bem como o valor médio de aproximação aos mínimos d_{min} é justificada pelo desempenho dos modelos utilizados, provenientes do estudo efectuado nas secções 6.3 e 6.4.3.

De acordo com o estudo levado a cabo na secção 5.3.1, valores óptimos para o tempo de crescimento conduzem em geral a elevadas sobrelevações e conseqüentemente a sistemas com baixas margens de estabilidade. Assim, qualquer pequeno desvio na exactidão dos modelos poderá ser o suficiente para se atingir uma situação de instabilidade, justificando-se desta forma o surgimento de 8% de sistemas instáveis, conforme a Tab. 7.2.

A interpretação dos desvios entre os valores de m^* e m deve ter em conta o facto dos primeiros (m^*) terem resultado de uma optimização com restrições. De acordo com a secção 5.3.1, no que se refere à optimização do tempo de crescimento, houve a necessidade de se rejeitarem os processos para os quais a razão entre o primeiro e segundo máximo da resposta ao degrau era inferior à que se verifica quando o controlador é sintonizado de acordo com as regras de Ziegler e Nichols. Nessa altura, verificou-se que não se impondo essa restrição surgiam sistemas demasiadamente oscilatórios, com elevados tempos de estabelecimento e muito próximos da instabilidade.

Na presente situação de sintonia para validação, não se verificando tal imposição, observou-se que os modelos utilizados conduzem, para alguns processos, a sistemas com margens de estabilidade menores, mais oscilatórios, e como tal com tempos de crescimento inferiores aos que foram considerados como mínimos durante a fase de modelação. Este facto traduz-se pelo aparecimento de parcelas negativas na avaliação da expressão (7.4), como a correspondente ao exemplo que se apresenta na Fig. 7.2 ($d_{\min}(\%) = -3.4\%$). Desta forma, uma avaliação da exactidão dos mínimos obtidos durante a sintonia, é feita recorrendo-se à determinação duma variante da expressão (7.4), designada por desvio absoluto médio $d_{\min}^a(\%)$, definida de acordo com a expressão (7.5).

$$d_{\min}^a(\%) = \frac{100}{P} \sum_{j=1}^p \left| \frac{m_j - m_j^*}{m_j^*} \right| \quad (7.5)$$

Considerando-se apenas os sistemas estáveis mencionados na Tab. 7.2, apresenta-se na Tab. 7.3 a avaliação das medidas de desvio dadas por (7.4) e (7.5), para cada um dos sub-conjuntos de processos que conduziram respectivamente a tempos de crescimento mínimos superiores e inferiores aos correspondentes obtidos na secção 5.3.1.

A observação da Tab. 7.3 permite verificar que os mínimos do tempo de crescimento determinados na secção 5.3.1, que serviram como valores objectivo para a aprendizagem

das redes neuronais, são transpostos por valores inferiores para mais de metade dos processos (55%) durante a fase de auto-sintonia. A verificação da ocorrência de tais processos é apoiada pela diferença existente entre $d_{\min}(\%)$ e $d_{\min}^a(\%)$ no que respeita à totalidade de sistemas estáveis. Os processos para os quais os mínimos encontrados durante a sintonia são superiores aos mínimos determinados para a modelação (44.6%), são os que conduzem às piores aproximações dos mínimos (4.3%).

	Total	%	$d_{\min}(\%)$	$d_{\min}^a(\%)$
Sistemas estáveis	101	100	0.8	3.1
com $m > m^*$	45	44.6	4.3	4.3
com $m < m^*$	56	55.4	-2.1	2.1

Tab. 7.3 – Aproximação dos mínimos para os sistemas estáveis.

Conclui-se que, para o conjunto base de experimentação em estudo, os sistemas mais oscilatórios são aqueles para os quais se consegue uma melhor aproximação ao mínimo do tempo de crescimento, embora seja de esperar que estes processos apresentem margens de estabilidade menores.

7.3.2 Sintonia otimizada para cada critério de desempenho

Na presente secção, a função objectivo do sistema de auto-sintonia da Fig. 7.1 é definida para cada um dos critérios de desempenho, *ITAE* e tempo de crescimento, tomado isoladamente. A simulação inicia-se em malha aberta, com uma onda quadrada na referência de amplitude unitária, com período e duração necessários à completa caracterização do fenómeno em estudo.

Conforme referido anteriormente, a identificação do processo exige que o regime transitório se veja completamente extinto. Os testes levados a cabo permitiram verificar que o tempo de estabelecimento da resposta à transição da onda quadrada difere consoante o critério de desempenho para o qual a sintonia tenha sido otimizada. Assim, de acordo com o que se regista na Tab. 7.4, houve a necessidade de se utilizar períodos da onda quadrada diferentes para procedimentos de sintonia otimizada para cada um dos critérios

de desempenho, *ITAE* e tempo de crescimento; conseqüentemente, as durações das simulações também diferem conforme o critério em estudo.

Durante o funcionamento do sistema de controlo, para cada processo em uso, terão lugar os procedimentos de identificação, adaptação e sintonia conforme exposto na secção 7.2.

De acordo com a metodologia proposta, as taxas de adaptação dos modelos de sintonia e dos modelos de critério são diferentes. Os modelos de sintonia devem estar preparados para avaliar os parâmetros do controlador optimizado, assim, a sua adaptação deve ser lenta quando comparada com a adaptação dos modelos dos critérios. Estes modelos devem ter a capacidade de se adaptar tão rapidamente quanto possível a variações que ocorram no processo; desta forma, assume-se que a taxa de adaptação dos modelos de critério é 10 vezes superior à taxa de adaptação dos modelos de sintonia. Levando em consideração o estudo realizado na secção 4.5.4, decidiu-se utilizar taxas de adaptação de modelação de critério e de sintonia definidas respectivamente nos conjuntos *M* e *S* representados por (7.6).

$$M = \{4, 2, 1\} \quad S = \{0.4, 0.2, 0.1\} \quad (7.6)$$

	Duração (s)	T (s)
<i>ITAE</i>	300	60
<i>TC</i>	450	160

Tab. 7.4 – Duração das simulações e períodos da onda quadrada.

Para cada um dos critérios de desempenho, a capacidade de adaptação do sistema durante uma determinada janela de simulação é quantificada pelo ganho de adaptação *G*, tal que: $G = CRIT_{final} / CRIT_{inicial}$. $CRIT_{inicial}$ e $CRIT_{final}$ representam o valor do critério no início e no fim da janela de adaptação; para que haja adaptação efectiva deverá verificar-se $G > 1$.

O teste da metodologia nesta fase de sintonia optimizada para os critérios de desempenho isolados é dividido em 3 etapas:

1ª ETAPA: Exploração do espaço das taxas de adaptação nos conjuntos *M* e *S*: Para cada um dos critérios de desempenho, *ITAE* e tempo de crescimento considerados isoladamente,

foram testados todos os processos do conjunto base de experimentação para todas as possibilidades de taxas de adaptação dentro dos conjuntos M e S . Desta forma, para cada critério e para cada processo existem 9 configurações diferentes para as taxas de adaptação; assim, a cobertura da totalidade do conjunto base de experimentação para ambos os critérios implica a realização de 1980 simulações. Lembrando que os processos têm função de transferência genérica dada por (7.7) e tendo em conta o elevado número de simulações implicado nesta fase de teste, foi seleccionado um sub-conjunto representativo do conjunto base de experimentação cujas constantes de tempo e atrasos se registam na Tab. 7.5[†].

$$G(s) = \frac{e^{-Ls}}{(1+T_1s)(1+T_2s)} \quad (7.7)$$

T_1	T_2	L
1	0	0.198
1	0.1	0.282
1	0.129	0.627
1	0.162	0.502
1	0.242	0.399

Tab. 7.5 – Processos tipo para validação da adaptação, considerando-se os critérios de desempenho isolados.

Na Tab. 7.6 apresentam-se os ganhos de adaptação para o referido sub-conjunto de processos, para as 9 combinações de taxas de adaptação extraídas dos conjuntos (7.6) e para ambos os critérios de desempenho: $ITAE$ e tempo de crescimento.

Os resultados apresentados da Tab. 7.6 evidenciam diferenças significativas de ganhos de adaptação no que respeita aos 2 critérios de desempenho estudados: $ITAE$ e tempo de crescimento. Para o 1º critério, $ITAE$, os ganhos de adaptação não ultrapassam as unidades verificando-se alguns casos, embora minoritários, onde não ocorreu ganho efectivo. No que respeita ao tempo de crescimento, verifica-se adaptação efectiva em todas as situações indo a ordem de grandeza do ganho de adaptação até às dezenas.

[†] Atendendo à ordenação que esteve na origem da construção do conjunto base de experimentação, os processos registados nesta tabela correspondem às posições 4, 15, 28, 37 e 56.

T_2	L	$S \downarrow$ $M \rightarrow$	ITAE			TC		
			4	2	1	4	2	1
0	0.198	0.4	0.57	0.79	2.75	19.80	19.80	19.80
		0.2	0.55	1.59	10.43	20.16	19.98	19.98
		0.1	0.84	2.05	9.45	19.98	20.16	20.16
0.1	0.282	0.4	1.07	1.95	3.47	9.35	9.47	9.47
		0.2	1.40	2.66	5.15	9.51	9.47	9.43
		0.1	1.60	3.19	6.30	9.51	9.51	9.47
0.129	0.627	0.4	1.70	1.91	2.17	5.27	5.23	5.30
		0.2	1.81	2.04	2.20	5.30	5.27	5.25
		0.1	1.89	2.16	2.22	5.27	5.28	5.28
0.162	0.502	0.4	2.18	2.05	2.50	5.87	5.81	5.80
		0.2	1.82	2.18	2.72	5.84	5.81	5.81
		0.1	1.93	2.26	2.76	5.86	5.83	5.84
0.242	0.399	0.4	1.95	2.26	3.11	6.17	6.19	6.17
		0.2	2.32	2.70	3.26	6.22	6.19	6.19
		0.1	2.54	2.86	3.12	6.21	6.21	6.24

Tab. 7.6 - Ganhos de adaptação para os processos da Tab. 7.5.

2ª ETAPA: Análise do procedimento adaptativo para processos invariantes no tempo: Pretende-se verificar, para além do ganho de adaptação (1ª etapa), a evolução da resposta do sistema na janela de adaptação face ao critério que se está optimizando. Nesta perspectiva, foram considerados 3 processos² de entre os que foram tabelados anteriormente, cujos ganhos de adaptação se realçam na Tab. 7.6. Para cada processo foi escolhido um par de taxas de adaptação (M, S), conforme se regista na Tab. 7.7; o cumprimento desta etapa foi feito em duas fases. Dado que se está a tratar cada critério de

² Correspondentes às posições 4, 28 e 56 no conjunto base de experimentação.

desempenho isoladamente, na primeira fase o procedimento adaptativo otimiza o *ITAE*, enquanto que na segunda fase procura-se otimizar o tempo de crescimento.

T_1	T_2	L	$M(\%)$	$S(\%)$
1	0	0.198	1	0.2
1	0.129	0.627	1	0.4
1	0.242	0.399	1	0.2

Tab. 7.7 – Processos seleccionados e respectivas taxas de adaptação.

As Fig. 7.3 a Tab. 7.8 apresenta na metade superior a resposta à onda quadrada enquanto que na metade inferior se pode observar as evoluções do tempo de crescimento e do *ITAE* ao longo das transições da onda quadrada. Para cada critério de desempenho pode-se igualmente observar a traço interrompido o valor do respectivo mínimo.

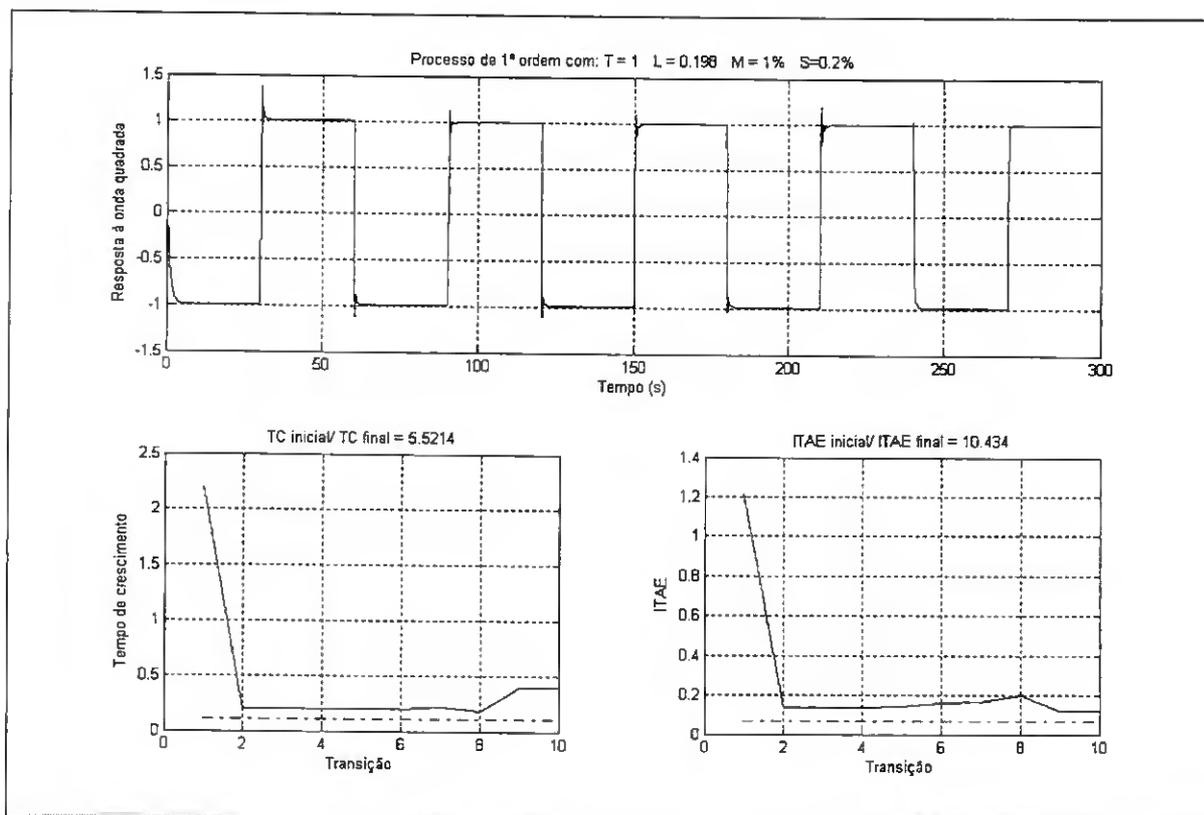


Fig. 7.3 – Adaptação para o *ITAE* para o processo com $T_1=1$, $T_2=0$ e $L=0.198$.

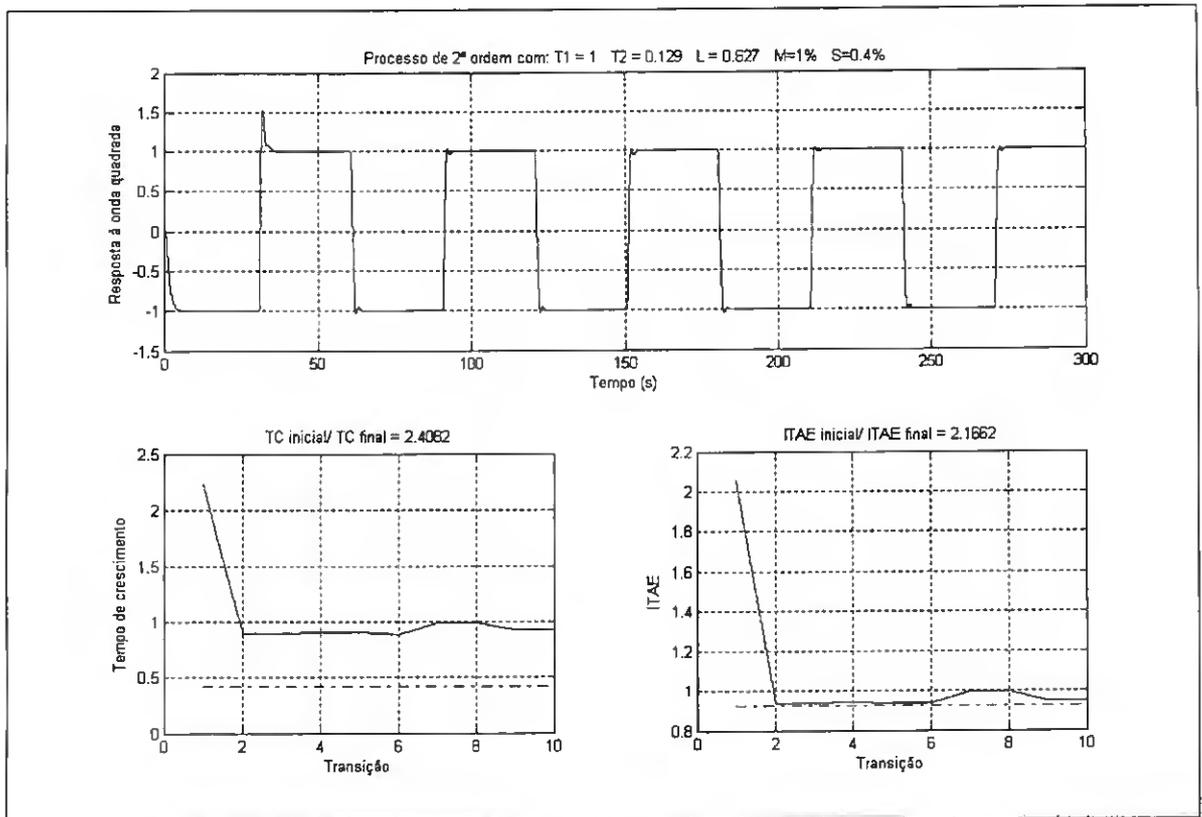


Fig. 7.4 – Adaptação para o *ITAE* para o processo com $T_1=1$, $T_2=0.129$ e $L=0.627$.

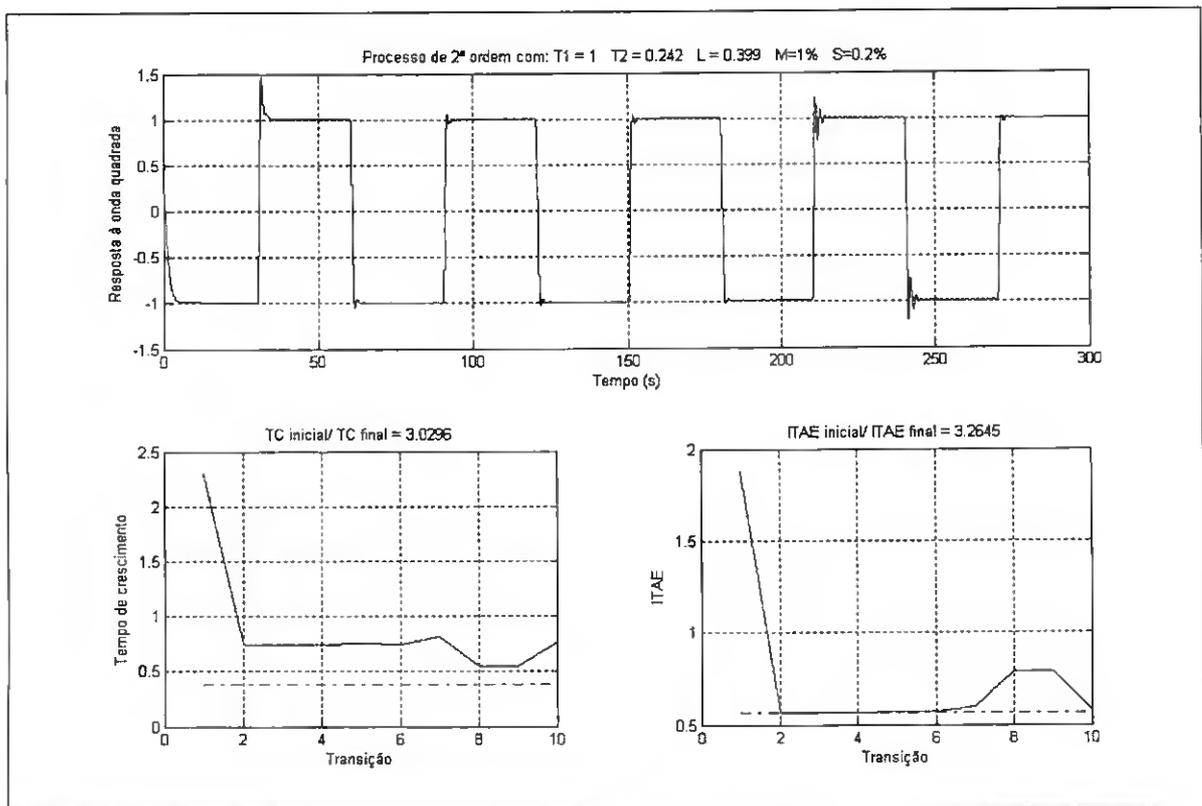


Fig. 7.5 – Adaptação para o *ITAE* para o processo com $T_1=1$, $T_2=0.242$ e $L=0.399$.

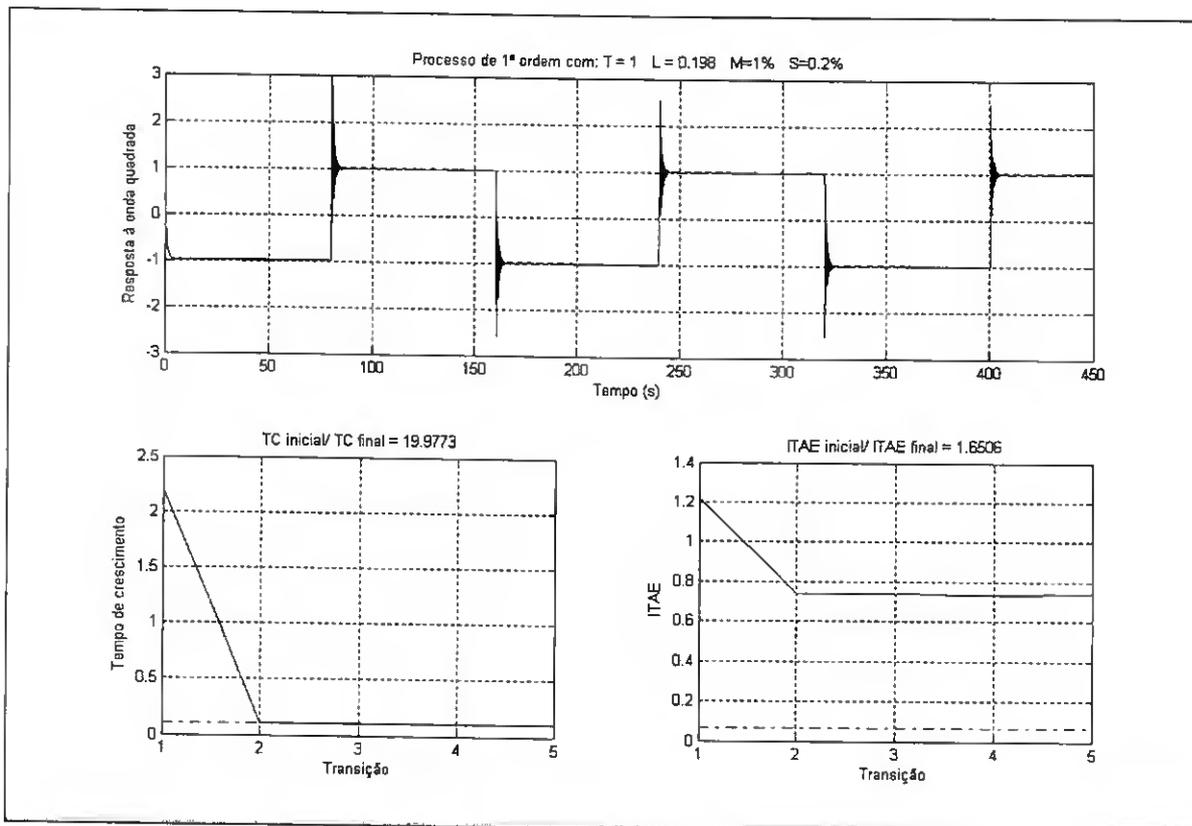


Fig. 7.6 – Adaptação para o tempo de crescimento para o processo com $T_1=1$, $T_2=0$ e $L=0.198$.

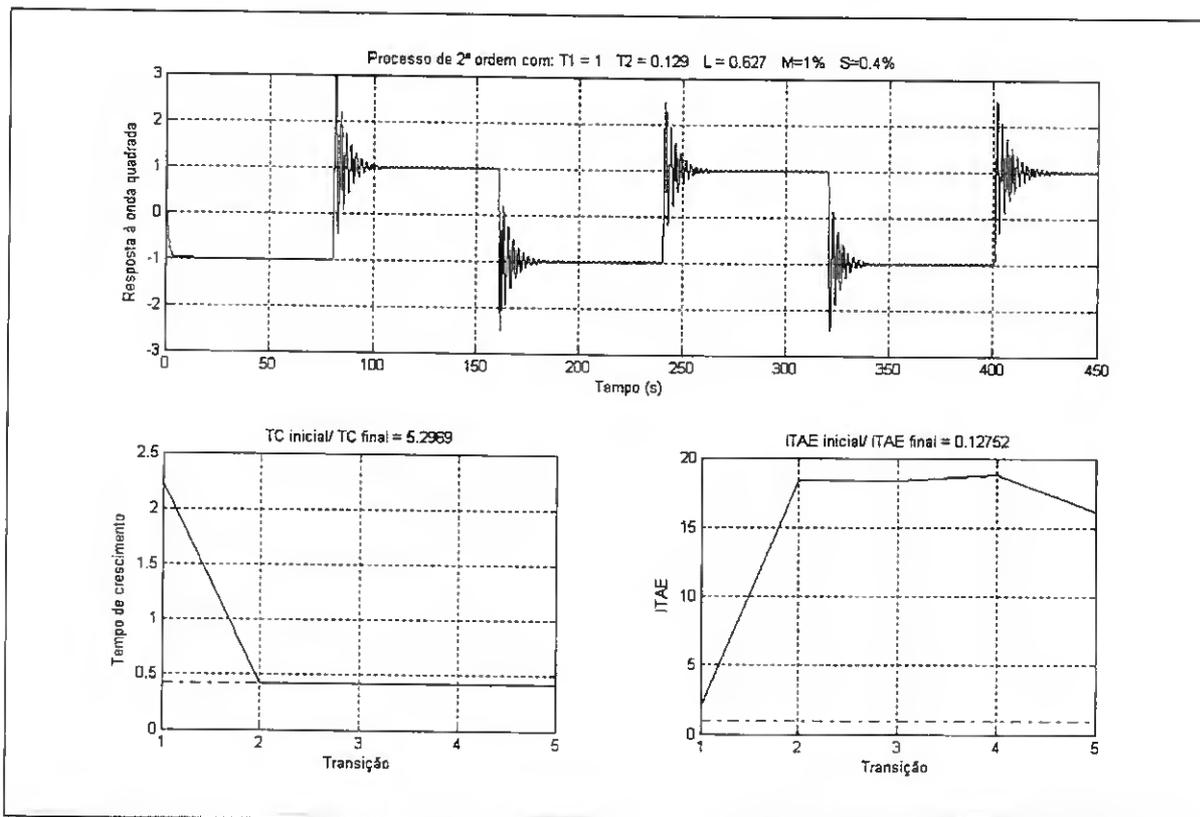


Fig. 7.7 – Adaptação para o tempo de crescimento para o processo com $T_1=1$, $T_2=0.129$ e $L=0.627$.

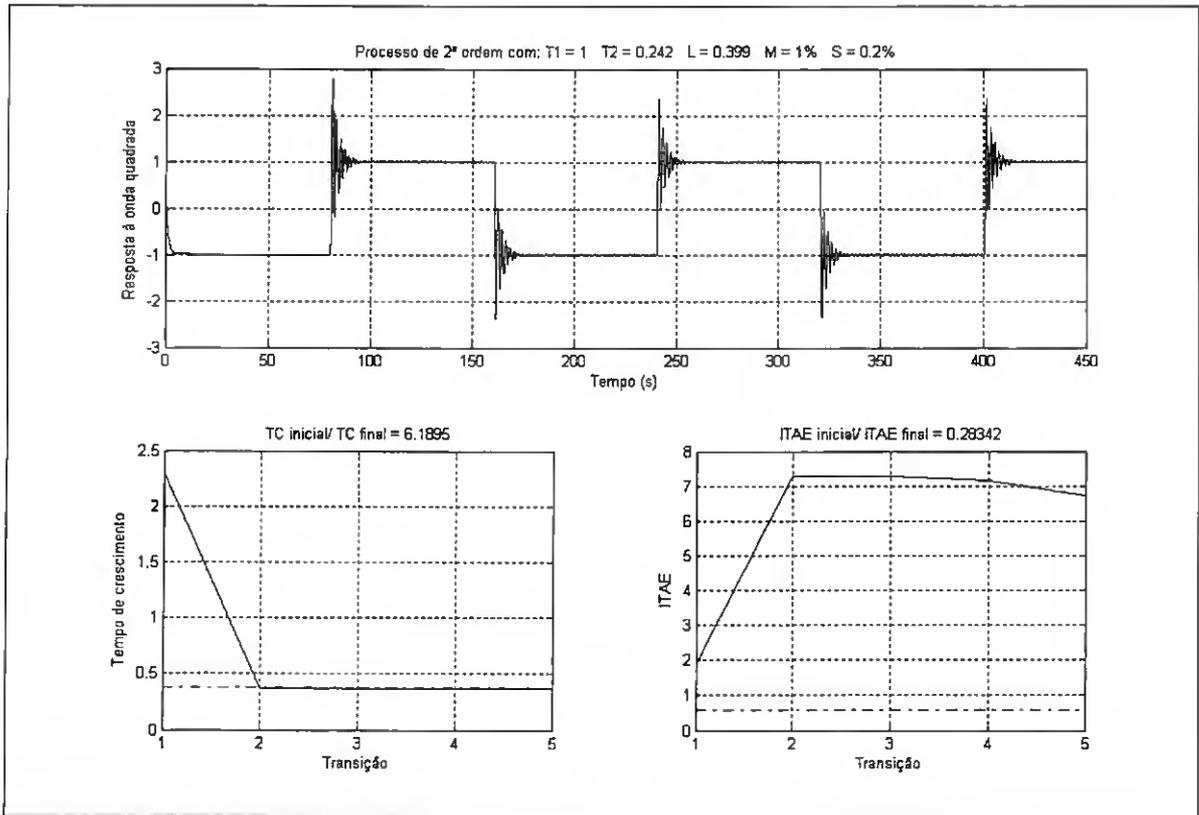


Fig. 7.8 – Adaptação para o tempo de crescimento para o processo com $T_1=1$, $T_2=0.242$ e $L=0.399$.

Para todos os exemplos apresentados nas Fig. 7.3 a Fig. 7.8 verifica-se uma boa aproximação do mínimo para o critério que está sendo otimizado. Deve ser salientado, no entanto, que essa aproximação é mais marcada para a otimização do tempo de crescimento.

A otimização do *ITAE* implica uma diminuição do tempo de crescimento em relação ao seu valor em malha aberta, contudo, o ganho para esse critério é sempre inferior àquele que se observa quando se procede à sua própria otimização. Por outro lado, a otimização do tempo de crescimento implica, dum modo geral, penalização do *ITAE*. De acordo com os 3 exemplos de otimização do tempo de crescimento (Fig. 7.6 a Fig. 7.8), apenas um exemplo mostra ganho efectivo para o *ITAE*; mesmo assim, o valor que se obtém para esse critério afasta-se significativamente do mínimo correspondente.

Dum modo geral a aproximação aos mínimos mantém-se ao longo das transições da onda quadrada, marcando desta forma o carácter adaptativo do sistema em estudo.

3ª ETAPA: Análise do procedimento adaptativo para processos variantes no tempo:

Pretende-se verificar o comportamento da técnica proposta quando o processo varia a sua dinâmica; nesta secção mantém-se invariante o critério de desempenho a otimizar. Desta forma, contrariamente ao que se verificou nas duas etapas de teste anteriores, durante a janela de adaptação o processo é alterado. À semelhança do que foi feito até agora, o presente teste ocorre em duas fases para cada um dos critérios de desempenho, *ITAE* e tempo de crescimento.

Na fase de teste do *ITAE*, a simulação iniciou-se com um processo³ com função de transferência dada por (7.7), caracterizada por $T_1=1$, $T_2=0.129$ e $L=0.491$. Seguidamente o tempo de atraso foi alterado nos instantes t_1 e t_2 de acordo com a esquematização (7.8).

$$L = 0.491 \xrightarrow{t_1=195s} L = 0.627 \xrightarrow{t_2=375s} L = 0.803 \quad (7.8)$$

Na metade superior da Fig. 7.9 pode-se observar a resposta à onda quadrada para uma simulação com duração de 500s durante a qual as taxas de adaptação se mantiveram a $M=2\%$ e $S=0.4\%$. Na metade inferior da mesma figura pode-se observar a evolução do *ITAE* sobreposta com os valores mínimos deste critério a traço interrompido, para os 3 processos em estudo.

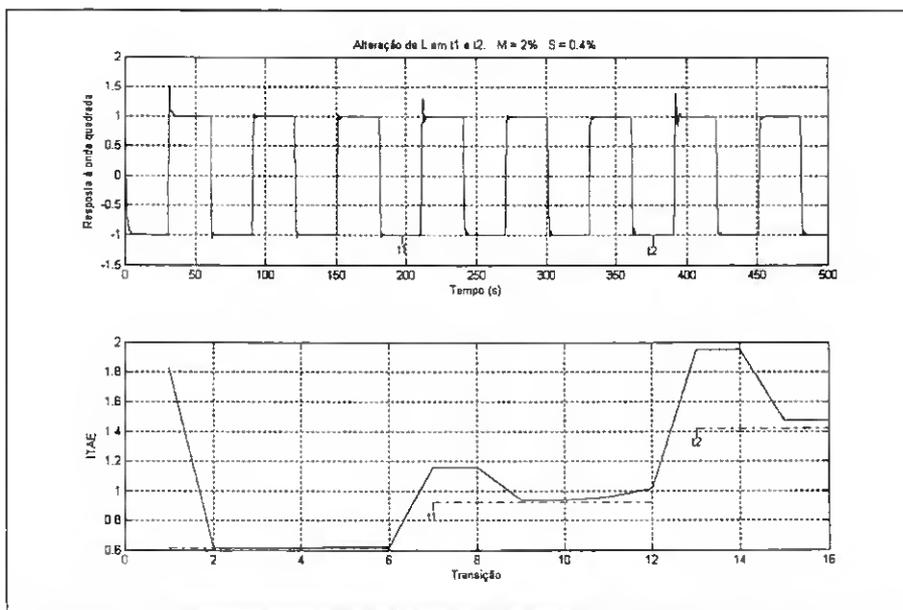


Fig. 7.9 – Adaptação otimizada para o *ITAE* com o processo variante no tempo.

³ N° 27 no conjunto base de experimentação.

A observação da Fig. 7.9 permite concluir que o controlador se adapta à nova dinâmica do processo, de facto, embora a variação do tempo de atraso implique um aumento do *ITAE*, verifica-se contudo de o seu valor volta a aproximar-se do mínimo respeitante ao novo processo. De salientar também que os novos controladores obtidos após cada uma das alterações só são aplicados na transição que ocorre respectivamente para $t=240s$ e $t=420s$, ou seja, na 1ª transição da perturbação da entrada, completado meio período após a alteração do processo.

O teste da auto-sintonia otimizada para o tempo de crescimento de processos variantes no tempo teve em conta a totalidade de elementos do conjunto base de experimentação, tomados sequencialmente. A transição entre um processo e o seguinte faz-se automaticamente, percorrendo-se assim os 110 processos que constituem o referido conjunto. Desta forma, tal como nos testes anteriores, considera-se uma onda quadrada de amplitude unitária e período $T=160s$, na referência. Adicionalmente define-se uma constante, intervalo de alteração T_a , como sendo o tempo ao fim do qual se passa para o processo seguinte; considerou-se $T_a = 3T$ para que ocorram 6 transições na onda quadrada durante a vigência de cada processo. A duração total da simulação deverá ser maior ou igual a $110T_a$, para que todos os processos sejam utilizados.

De acordo com a metodologia de auto-sintonia para processos variantes no tempo, os parâmetros do controlador determinados para um dado processo manter-se-ão para o processo seguinte. Idealmente a malha de controlo nunca será aberta; as tarefas de auto-adaptação e sintonia do controlador (Tab. 7.1), são suprimidas no meio período em que ocorre a transição do processo.

O procedimento de auto-sintonia descrito anteriormente foi aplicado a todos os processos do conjunto base de experimentação, tendo-se observado como mais frequentes as seguintes situações:

- a) Detectada a 1ª ocorrência de transição na onda quadrada após a alteração do processo, não são executadas as tarefas de auto-adaptação e sintonia pois o valor final do sinal de controlo ainda não é conhecido para o novo processo. Verifica-se instabilidade do sistema e abertura da malha de controlo antes da próxima transição da onda quadrada. Seguidamente ocorrem as etapas normais de identificação em

malha aberta, sintonia e fecho da malha, identificação em malha fechada e sintonia, como habitualmente.

- b) No início a situação é semelhante à anterior, diferindo desta pelo facto de não ocorrer instabilidade após a 1ª transição da onda quadrada. No entanto, verificou-se após concluída a 2ª transição que o sistema era instável, ou o regime estacionário ainda não tinha sido atingido, implicando a abertura da malha de controlo. Seguem-se (tal como na situação anterior) as etapas normais de identificação em malha aberta, sintonia e fecho da malha, identificação em malha fechada e nova sintonia, como numa situação comum.

Ambas as situações descritas anteriormente reflectem a recuperação da sintonia do controlador após ter ocorrido a abertura da malha de controlo. A situação a) é justificada pelas (já referidas) reduzidas margens de estabilidade quando a sintonia é optimizada para o tempo de crescimento. Assim, estando o controlador sintonizado para que este critério de desempenho seja óptimo para um dado processo, qualquer pequena alteração neste poderá ser o suficiente para que se passe de um tempo de crescimento óptimo, para uma situação de instabilidade.

A justificação da situação descrita no ponto b) poderá ser considerada como um meio termo em relação à que se apresenta para o ponto a); ou seja, os parâmetros óptimos do controlador para um dado processo, quando utilizados para outro processo, conduzem a uma resposta muito oscilatória cujo tempo de estabelecimento é muito elevado. Nestas circunstâncias, a observação da resposta do sistema durante a simulação do procedimento de auto-sintonia permitiu verificar que o meio período de 80s da onda quadrada deixa de ser suficiente para que o regime transitório seja completamente extinto, que é condição necessária para a correcta determinação das medidas de identificação. Assim, a malha é aberta para que ocorra uma correcta identificação do processo, retomando-se de seguida o normal procedimento de adaptação e sintonia.

Face à justificação que se apresenta para as situações que se enquadram no ponto b), conclui-se que é necessário que se aumente o período da onda quadrada, para que se consiga uma correcta identificação, quando o sistema se torna mais oscilatório e com tempo de estabelecimento mais elevado. De facto, considerando-se posteriormente a onda quadrada com $T=320s$ (dobro do período inicial), verificou-se que é possível obter

alterações na dinâmica do processo atingindo-se o regime estacionário, sem a necessidade de se abrir a malha de controlo, tal como se exemplifica nas Fig. 7.10 a Fig. 7.12.

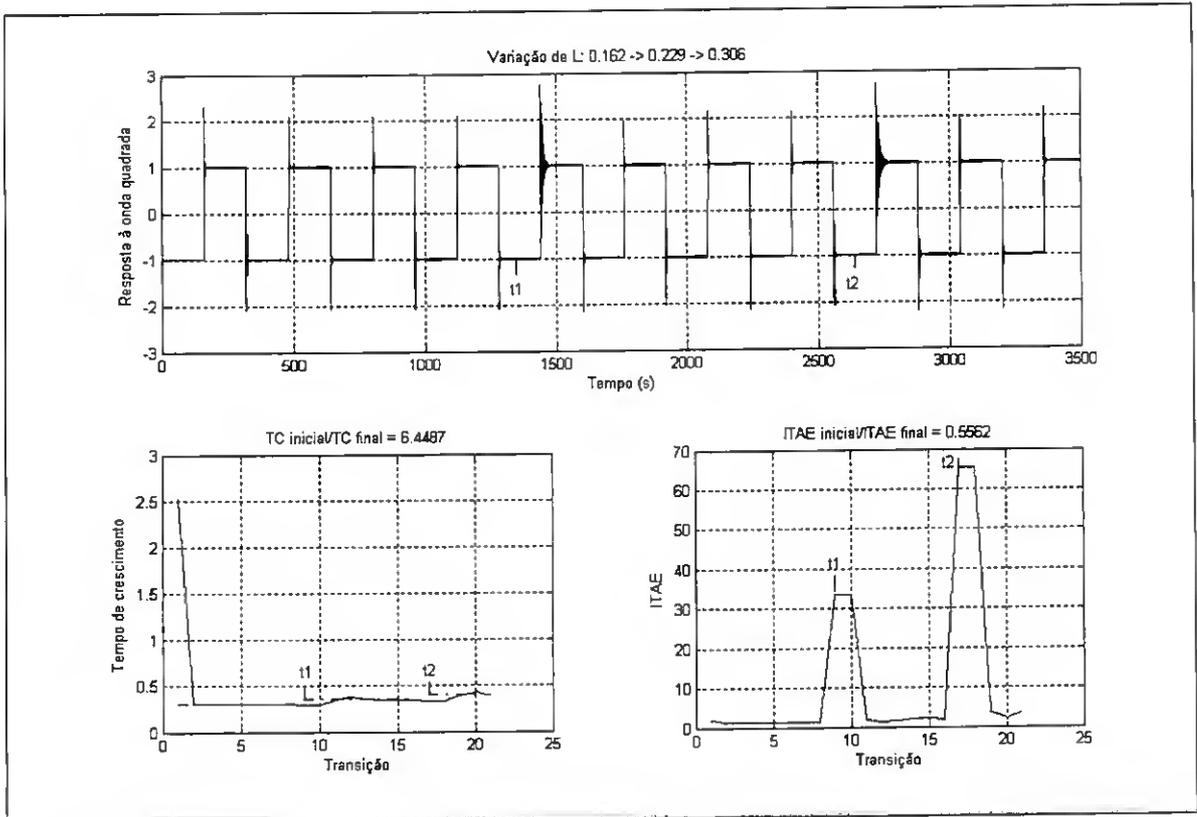


Fig. 7.10 – Adaptação otimizada para o tempo de crescimento considerando o processo variante no tempo (L: 0.162 -> 0.229 -> 0.306).

No exemplo que se apresenta na Fig. 7.10 partiu-se ($t=0$) do sistema em malha aberta com um processo⁴ com função de transferência dada por (7.9).

$$G(s) = \frac{e^{-0.162s}}{(1+s)(1+0.443s)} \tag{7.9}$$

A duração da simulação foi de 3500s com alteração da dinâmica do processo nos instantes t_1 e t_2 , conforme se ilustra em (7.10).

$$L = 0.162 \xrightarrow{t_1=1360s} L = 0.229 \xrightarrow{t_2=2640s} L = 0.306 \tag{7.10}$$

Na metade superior da Fig. 7.10 pode-se observar a evolução da resposta à onda quadrada face à alteração da dinâmica do processo; na metade inferior registam-se as respectivas evoluções do tempo de crescimento e do *ITAE*, assinalando-se a traço interrompido os 3

⁴ N° 83 no conjunto base de experimentação.

mínimos do critério que se está otimizando. Para uma melhor leitura da figura, registre-se que para os 3 processos em estudo os mínimos do tempo de crescimento são respectivamente 0.31s, 0.36s e 0.41s.

Face a este exemplo, pode-se concluir que se consegue uma boa aproximação do mínimo do tempo de crescimento referente ao processo cuja função de transferência se apresenta em (7.9), logo após o fecho da malha. Os 2 processos que se seguem nesta simulação apresentam evoluções do tempo de crescimento que constituem igualmente boas aproximações aos respectivos mínimos. No que respeita ao *ITAE*, verificam-se irregularidades nas transições da dinâmica do processo; tal facto justifica-se atendendo a que a optimização está a ser feita apenas para o tempo de crescimento.

O 2º exemplo corresponde a uma simulação com a duração de 4500s, cuja resposta à onda quadrada se apresenta na Fig. 7.11. Este exemplo inicia-se com o sistema em malha aberta utilizando-se um processo⁵ com função de transferência dada por (7.11).

$$G(s) = \frac{e^{-0.192s}}{(1+s)(1+s)} \quad (7.11)$$

Conforme se ilustra na Fig. 7.11, ocorreram 4 alterações na dinâmica do processo de acordo com as transições que se esquematizam em (7.12).

$$L = 0.192 \xrightarrow{t_1=1040s} L = 0.269 \xrightarrow{t_2=2000s} L = 0.355 \xrightarrow{t_3=2960s} L = 0.451 \xrightarrow{t_4=3920s} L = 0.560 \quad (7.12)$$

Na Fig. 7.12 apresentam-se as evoluções do tempo de crescimento e do *ITAE* correspondentes à simulação cuja resposta à onda quadrada se mostra na Fig. 7.11.

As observações feitas para este exemplo em tudo corroboram com aquelas que se registam para o exemplo anterior, diferindo pelo facto de no presente caso estarem em estudo 5 processos em vez de 3 considerados anteriormente. Tal como dantes, os mínimos do critério que está sendo otimizado são rapidamente atingidos. Dado tratarem-se de valores muito próximos (variam entre 0.42s e 0.7s), fazem com que o andamento global para o tempo de crescimento não apresente variações significativas. O mesmo já não acontece

⁵ N° 103 conjunto base de experimentação.

com o *ITAE*, para o qual, a alteração do processos leva a aumentos significativos para este critério.

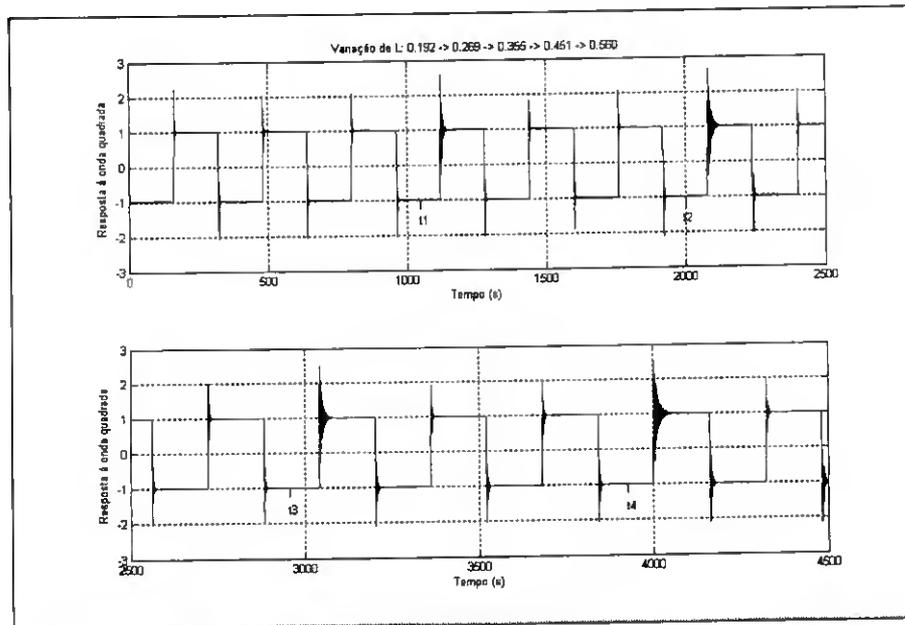


Fig. 7.11 – Adaptação otimizada para o tempo de crescimento considerando o processo variante no tempo (L: 0.192 -> 0.269 -> 0.355 -> 0.451 -> 0.560).

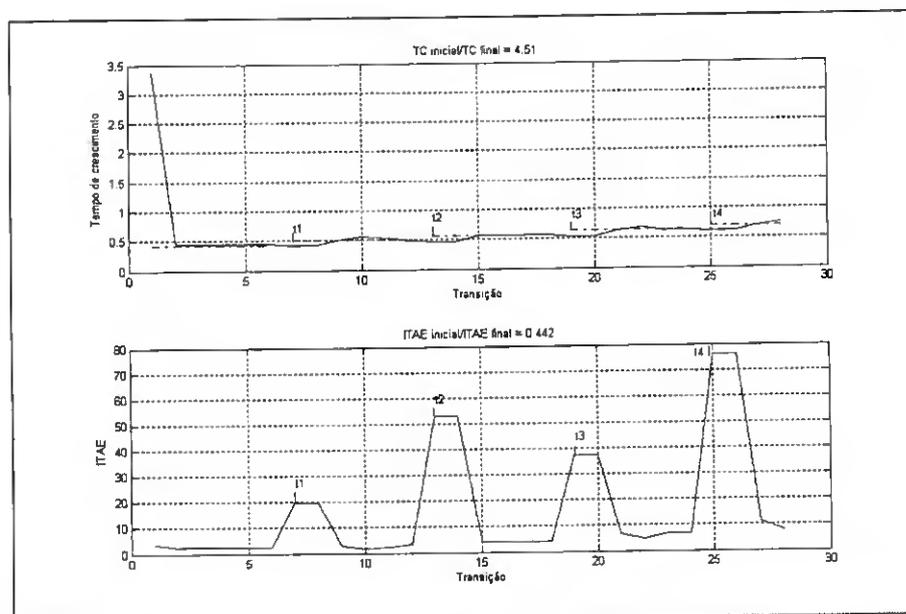


Fig. 7.12 – *ITAE* e tempo de crescimento para a adaptação referente à Fig. 7.11.

7.4 SINTONIA OPTIMIZADA COM PONDERAÇÃO DE CRITÉRIOS

A fase final de teste do sistema de auto-sintonia neuro-genética considera em simultâneo os 2 critérios de desempenho em estudo: *ITAE* e tempo de crescimento. Nestas condições, embora a optimização permaneça uni-objectivo, a função objectivo (7.2) assume a forma apresentada em (3.13).

Para que a ponderação simultânea de ambos os critérios de desempenho seja regulada por apenas um parâmetro, é estabelecida a dependência (7.13) para as componentes do vector w da expressão (3.13).

$$w_1 = k, \quad w_2 = 1 - k, \quad k \in [0 \ 1] \quad (7.13)$$

Assim, o peso de cada critério durante um dado procedimento de auto-sintonia é regulado pelo factor de ponderação de critérios, k , assumindo a função objectivo a forma que se apresenta na expressão (7.14).

$$f(PID) = kg_1(PID) + (1 - k)g_2(PID) \quad (7.14)$$

com: $PID = [k_c \ t_i \ t_d]$, $g_1 \equiv ITAE$ e $g_2 \equiv TC$.

Tendo-se verificado que para os processos que compõem o conjunto base de experimentação as gamas de variação dos critérios de desempenho em estudo são aproximadamente da mesma ordem de grandeza, decidiu-se não se proceder a nenhuma normalização adicional das parcelas que compõem a expressão (7.14).

Nestas circunstâncias, para cada um dos processos do conjunto base de experimentação, foram executados procedimentos de auto-sintonia para valores de k , definidos no conjunto dado por (7.15).

$$K = \{0, \ 0.25, \ 0.5, \ 0.75, \ 1\} \quad (7.15)$$

Analogamente ao que aconteceu nas secções anteriores, as taxas de adaptação são definidos nos conjuntos (7.6). Considerando os 110 processos que compõem o conjunto base de experimentação e atendendo aos cardinais dos conjuntos M , S , K , conclui-se que o procedimento acabado de descrever originou 4950 simulações, das quais, alguns exemplos

se apresentam na Tab. 7.8. Os ganhos de adaptação que se apresentam nesta tabela foram obtidos para 3 processos do conjunto base de experimentação, considerando-se os valores de k definidos em (7.15) e as taxas de adaptação $M=1$ e $S=0.1$.

Processo	Ponderação (k)	Ganho	
		<i>ITAE</i>	<i>TC</i>
$G(s) = \frac{e^{-0.198s}}{1+s}$	0	1.65	19.98
	0.25	1.61	19.97
	0.5	1.56	20.16
	0.75	11.97	11.36
	1	10.14	11.61
$G(s) = \frac{e^{-0.269s}}{1+s}$	0	1.20	14.36
	0.25	1.19	14.45
	0.5	1.19	14.36
	0.75	6.54	5.80
	1	6.77	5.76
$G(s) = \frac{e^{-0.587s}}{1+s}$	0	0.23	6.78
	0.25	0.24	6.76
	0.5	0.26	6.68
	0.75	3.07	3.29
	1	3.05	3.29

Tab. 7.8 – Ganhos de adaptação obtidos nos procedimentos de auto-sintonia com ponderação de critérios.

A observação de cada uma das 3 partes da Tab. 7.8, correspondentes a cada processo, permite verificar que o aumento do factor de ponderação de critério implica para o ganho de adaptação, em termos globais, um aumento no caso do *ITAE* e uma diminuição no que se refere ao tempo de crescimento. Esta constatação justifica-se atendendo à forma como se define a função objectivo (7.14) em termos de *k*. De acordo com esta expressão, um aumento progressivo de *k* faz com que seja dado mais peso ao *ITAE* e menos peso ao tempo de crescimento, tendo como consequência a obtenção dos valores observados.

Nas figuras Fig. 7.13 a Fig. 7.15 apresentam-se as respostas à onda quadrada e as respectivas evoluções dos critérios de desempenho, para o processo $G(s) = \frac{e^{-0.587s}}{1+s}$, quando se utiliza $k=0, 0.5$ e 1 ; os ganhos de adaptação respectivos estão assinalados na Tab. 7.8.

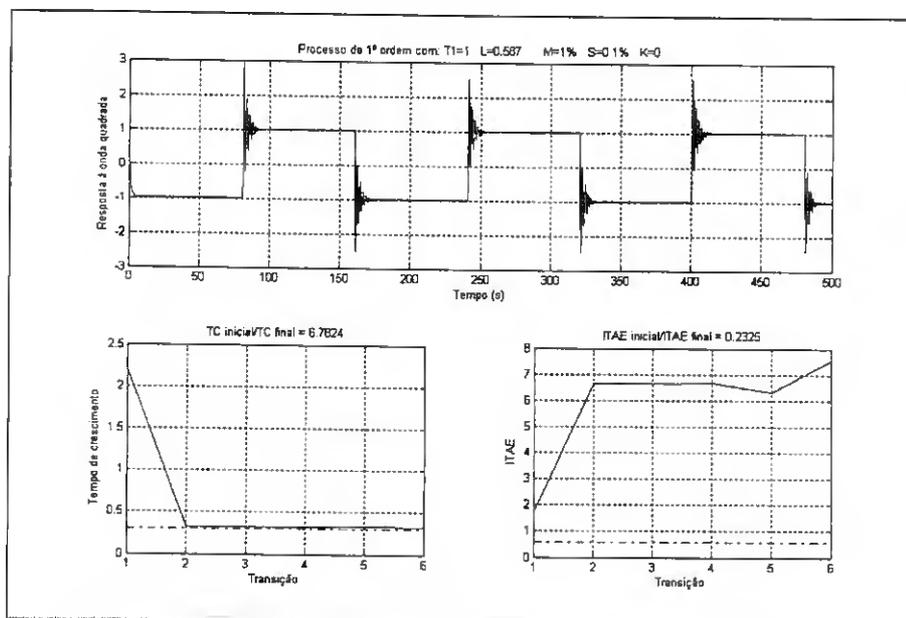


Fig. 7.13 – Adaptação para ponderação de critérios com $k=0$.

A partir do exemplo que se apresenta nas Fig. 7.13 a Fig. 7.15 pode-se confirmar a tendência do *ITAE* para aproximação do mínimo quando se utiliza $k=1$; contrariamente, o tempo de crescimento afasta-se do respectivo mínimo nessa situação ($k=1$). Inversamente ($k=0$), verifica-se que o tempo de crescimento aproxima-se o respectivo mínimo enquanto que o *ITAE* tende a aumentar. No que respeita ao perfil da resposta do sistema, como seria de esperar, pode-se observar que as situações em que o *ITAE* é menor a resposta à onda quadrada apresenta menor sobrelevação, contrariamente, as respostas que optimizam

preferencialmente o tempo de crescimento tornam-se bastante oscilatórias, apresentando tempos de estabelecimento maiores.

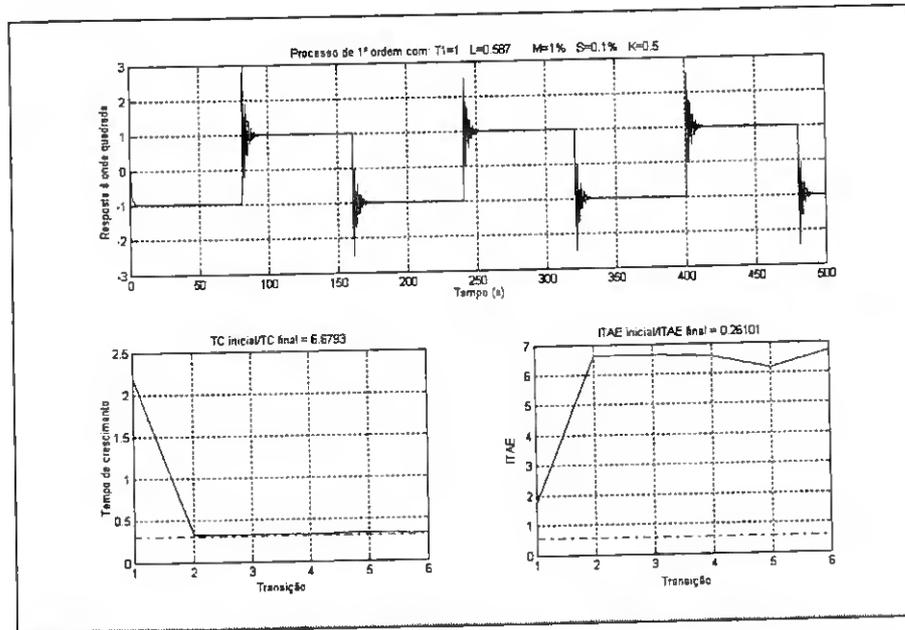


Fig. 7.14 – Adaptação para ponderação de critérios com $k=0.5$.

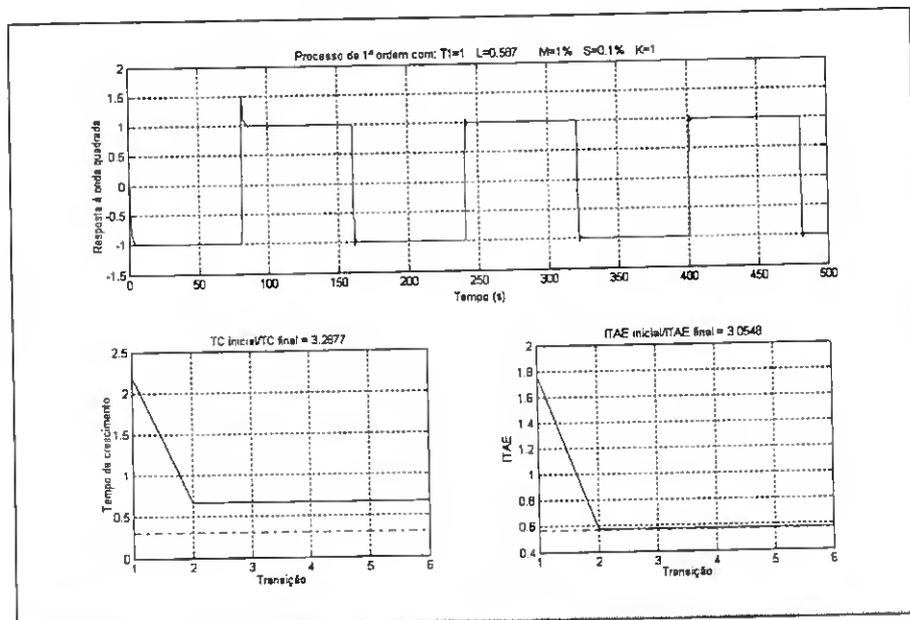


Fig. 7.15 – Adaptação para ponderação de critérios com $k=1$.

7.5 CONCLUSÕES

Neste capítulo apresentou-se um simulador auto-adaptativo neuro-genético desenvolvido em MATLAB, recorrendo ao ambiente e simulação dinâmica e visualização gráfica facultado pelo SIMULINK.

Os exemplos estudados permitem verificar que existe um andamento globalmente adaptativo, para os sistemas em estudo e em ambas as configurações: auto-sintonia para cada critério de desempenho tomado isoladamente e auto-sintonia para uma função objectivo definida por uma soma ponderada dos critérios de desempenho.

Verificou-se contudo que o desempenho do sistema de auto-sintonia proposto é fortemente dependente da exactidão dos modelos, tanto dos critérios, quanto de sintonia. As taxas de adaptação são igualmente importantes para a evolução dos modelos quando utilizados em linha. Assim, houve ocasião de se observar que os modelos criados para o tempo de crescimento conduzem a sintonias mais exactas do que os modelos utilizados para o *ITAE*. O processo adaptativo foi testado tanto para processos invariantes no tempo como para situações em que o processo experimenta alteração na sua dinâmica; verificou-se que a recuperação da sintonia é fortemente influenciada pela exactidão dos modelos dos respectivos critérios.

A identificação dos processos é fundamental para todo o procedimento de auto-sintonia, exigindo-se assim, erros de identificação baixos para que o procedimento de auto-sintonia tenha sucesso. É igualmente indispensável que o regime transitório esteja extinto para que as medidas de identificação sejam calculadas com sucesso, este facto limita inferiormente o período utilizado para a onda quadrada, obrigando a utilizar valores diferentes para critérios de desempenho diferentes.

A técnica proposta permitiu observar, tal como seria de esperar, que quando a resposta à onda quadrada está otimizada para o tempo de crescimento, esta apresenta-se bastante oscilatória e com tempos de estabelecimento consideráveis. Contrariamente, quando a sintonia privilegia o *ITAE*, observam-se respostas com sobrelevações consideravelmente menores; tais factos justificam-se pelas próprias naturezas dos critérios de desempenho em estudo.

8. COMPARAÇÃO DO DESEMPENHO DE DIFERENTES MÉTODOS DE SINTONIA

8.1 INTRODUÇÃO

O capítulo 7 centrou-se sobre a metodologia de auto-sintonia proposta; a sua implementação recorreu ao MATLAB [134] [139], desenvolvendo-se nesse ambiente os testes respectivos. O sistema desenvolvido nessa plataforma utilizou os processos do conjunto base de experimentação, tendo-se apresentado resultados relativos a alguns elementos desse conjunto. Foi verificado nesse capítulo como decorre a auto-sintonia tanto na presença de variação no tempo da dinâmica do processo, quanto face a diferentes ponderações dos critérios de desempenho na avaliação da função objectivo que se pretende otimizar.

No presente capítulo apresenta-se comparativamente o desempenho de diferentes métodos de sintonia do controlador *PID*, procedendo-se à análise da resposta ao degrau unitário do sistema controlado. Os critérios de desempenho utilizados na análise comparativa são: o *ITAE*, o *ITAY*, o tempo de crescimento, o tempo de estabelecimento e a sobrelevação. Os métodos de sintonia do controlador *PID* utilizados para comparação com a metodologia proposta são de: Ziegler e Nichols, Cohen-Coon e Åström-Hägglund.

Tal como acontece com a metodologia de auto-sintonia proposta, nos métodos de sintonia que se apresentam para comparação é fundamental a modelação dos processos que se pretendem controlar. Desta forma na secção 8.2 são feitas considerações sobre a estimação de modelos, apresentando-se exemplos para alguns elementos do conjunto base de experimentação. Na secção 8.3 são determinados parâmetros do controlador *PID* tendo em conta os métodos de sintonia referidos. Para cada conjunto de parâmetros *PID* é

determinada a resposta ao degrau e avaliados, comparativamente, os critérios de desempenho mencionados. O capítulo termina com a secção 8.4 onde se apresentam as conclusões referentes à análise comparativa dos diferentes métodos de sintonia.

8.2 ESTIMAÇÃO DE MODELOS

O algoritmo implementado para comparação de métodos recorreu à *toolbox* desenvolvida na Universidade de Sunderland [140] para a sintonia de controladores *PID*. De acordo com a abordagem levada a cabo nessa *toolbox*, a identificação do processo é feita a partir da sua resposta em malha aberta a uma entrada em degrau unitário. Esta resposta, juntamente com a base de tempo correspondente e a perturbação (em degrau) que a originou, permite identificar modelos de 1ª ou de 2ª ordem com tempo de atraso e modelos de 2ª ordem com tempo de atraso e 1 zero.

Optando-se pela determinação de modelos de 1ª ordem com tempo de atraso, a partir da resposta ao degrau unitário nas condições referidas, identificam-se os 3 parâmetros (L , T e L) da função de transferência cuja expressão geral se apresenta em (1.33).

Designado-se por $y_{proc}(t)$ a resposta do processo ao degrau unitário e $y_{mod}(t)$ a resposta análoga do respectivo modelo; a avaliação da adequação da modelação recorre à determinação de $d(t) = y_{mod}(t) - y_{proc}(t)$. A partir de $d(t)$, para cada processo, determina-se a percentagem de *MRE* de acordo com (8.1), para as amostras $y_{proc}(t_k)$ não nulas e considerando constante a duração do degrau unitário.

$$MRE(\%) = \frac{100}{n} \sum_{k=1}^n \left| \frac{y_{mod}(t_k) - y_{proc}(t_k)}{y_{proc}(t_k)} \right| \quad (8.1)$$

Na Tab. 8.1 apresentam-se, para 3 exemplos¹ pertencentes ao conjunto base de experimentação, os respectivos parâmetros do modelo equivalente de 1ª ordem e os valores da percentagem de *MRE* para respostas ao degrau com duração constante e igual a 30s.

¹ Processos nas posições 5, 41 e 109.

	K	T	L	$MRE(\%)$
$G(s) = \frac{e^{-0.27s}}{1+s}$	1	0.92	0.34	0.75
$G(s) = \frac{e^{-0.05s}}{(1+s)(1+0.20s)}$	1	1.01	0.23	0.82
$G(s) = \frac{e^{-0.83s}}{(1+s)(1+s)}$	1	1.65	1.14	2.87

Tab. 8.1 – Exemplos de modelos de 1ª ordem.

Relativamente aos valores da Tab. 8.1 para o processo de 1ª ordem, verifica-se que o erro entre a resposta gerada pelo próprio processo e pelo respectivo modelo não é nulo, este facto deve-se às diferenças entre os parâmetros do processo e os parâmetros identificados pela *toolbox*, justificadas pelas opções tomadas no desenvolvimento desta.

Nas metades superiores das Fig. 8.1 a Fig. 8.3 apresentam-se as respostas ao degrau para os exemplos da Tab. 8.1, sobrepostas com as respostas análogas geradas a partir dos modelos respectivos; para cada processo, na metade inferior regista-se a diferença $d(t)$.

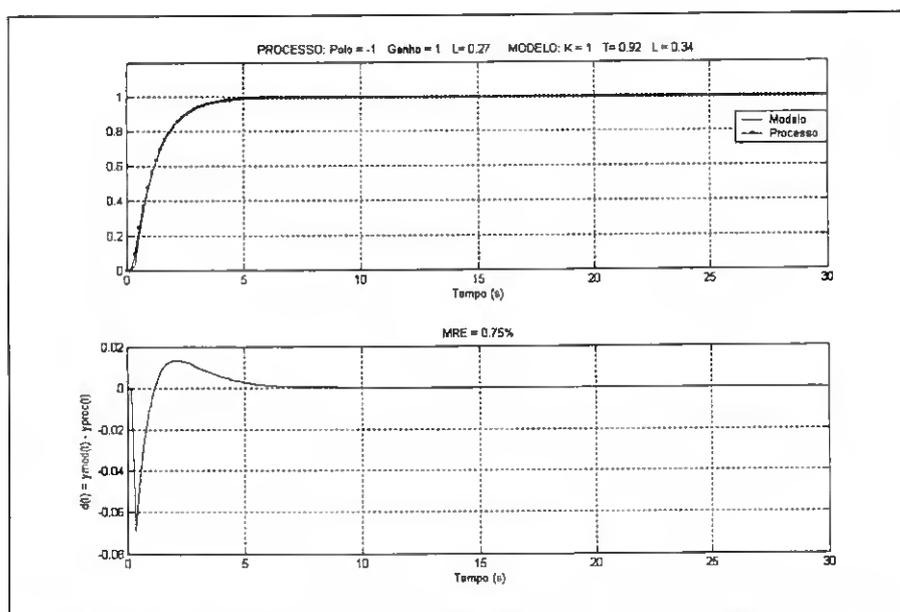


Fig. 8.1 – Comparação das respostas ao degrau para o processo com $T=1$ e $L=0.27$.

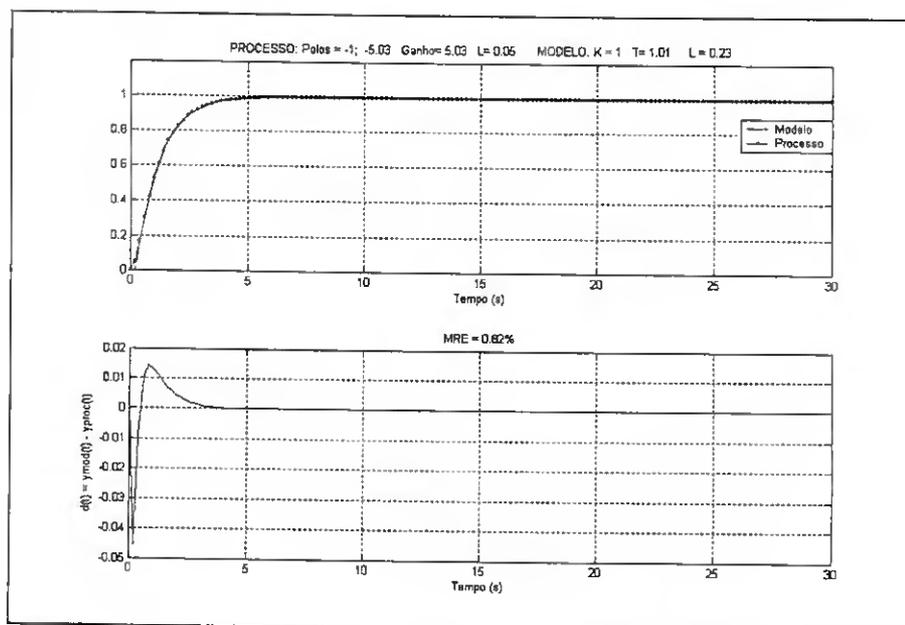


Fig. 8.2 – Comparação das respostas ao degrau para o processo com $T_1=1$, $T_2=0.20$ e $L=0.05$.

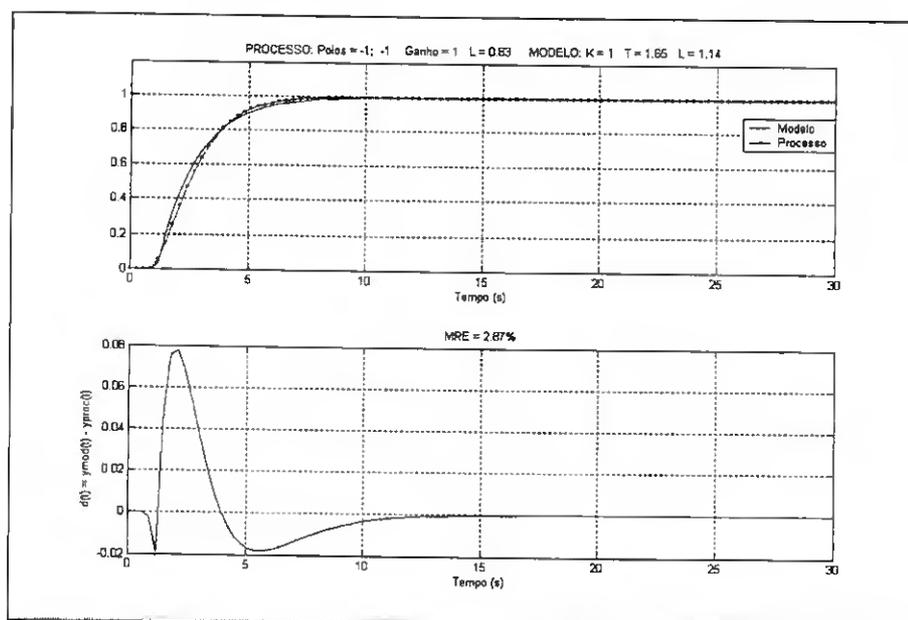


Fig. 8.3 – Comparação das respostas ao degrau para o processo com $T_1=1$, $T_2=1$ e $L=0.83$.

As percentagens de *MRE* mostram que o processo de 1ª ordem é o que apresenta resposta ao degrau mais próxima da correspondente do respectivo modelo; tal facto é de esperar dado que se trata de uma modelação de 1ª ordem. No que diz respeito aos processos de 2ª ordem, o erro é menor para aquele cuja 2ª constante de tempo é menor. Os exemplos apresentados permitem antever que a modelação de 1ª ordem é também adequada a processos de 2ª ordem. Esta conclusão é apoiada pela Tab. 8.2 onde se registam, para cada

conjunto de processos de 1ª ordem e de 2ª ordem respectivamente, os valores mínimos, médios e máximos da percentagem de *MRE*, para respostas ao degrau com duração de 30s.

Processos	% <i>MRE</i>		
	Mínimo	Médio	Máximo
1ª ordem	0.05	1.32	4.70
2ª ordem	0.18	2.04	8.47

Tab. 8.2 – Verificação da exactidão dos modelos de 1ª ordem.

A metodologia de auto-sintonia proposta na presente dissertação, no que respeita a estimação de modelos, utiliza a avaliação de medidas de identificação do processo de acordo com o que se apresenta na secção 3.2.1. Conforme visto nessa secção, essa metodologia recorre igualmente à resposta ao degrau, podendo as medidas de identificação ser calculadas tanto em malha aberta como em malha fechada. A implementação proposta valida a determinação numérica das medidas de identificação, rejeitando-as sempre que alguma apresente desvio percentual em relação ao valor analítico superior a 2%. De acordo com os testes realizados no capítulo 7, o valor de erro de 2% raramente é atingido; quando tal acontece, um aumento do período da onda quadrada possibilita, em geral, a obtenção de valores de desvio inferiores.

8.3 COMPARAÇÃO DE CRITÉRIOS DE DESEMPENHO PARA DIFERENTES MÉTODOS DE SINTONIA

Os critérios de desempenho considerados nesta fase de comparação são:

- *ITAE*
- Rejeição de perturbações à saída (*ITAY*)
- Tempo de crescimento (*TC*)
- Tempo de estabelecimento (*TS*)

- Sobreelevação (S)

O $ITAE$, o $ITAY$ e o tempo de crescimento definem-se de acordo com as expressões (3.11), (3.12) e (5.14), respectivamente.

O tempo de estabelecimento é o lapso de tempo necessário para que o regime transitório seja extinto. Em geral, para uma resposta $y(t)$, considera-se tempo de estabelecimento a 5% do valor final o menor valor t_s tal que: $\forall t > t_s \Rightarrow |y(t) - y_{final}| \leq 0.05$.

O valor da sobreelevação em percentagem para um sistema sub-amortecido define-se de acordo com a expressão (8.2).

$$S(\%) = 100 \frac{\max(y) - y_{final}}{y_{final}} \quad (8.2)$$

De acordo com a implementação realizada na *toolbox* da Universidade de Sunderland [140], após a estimação de modelos a que se refere a secção anterior, é feita a confrontação das respostas ao degrau dos processos da Tab. 8.1, quando sujeitos ao controlo PID sintonizado de acordo com os métodos de:

- Ziegler-Nichols (ZN) (secção 1.4.1.1)
- Cohen-Coon (CC) (secção 1.4.5)
- Åström-Hägglund (AH) (secção 1.4.2.1)

Os critérios de desempenho, referentes às respostas que se obtêm pela aplicação dos métodos de sintonia referidos, são comparados com os correspondentes critérios quando se aplica a metodologia proposta, considerada tanto na vertente de optimização do $ITAE$ (P_{ITAE}), como na vertente de optimização do tempo de crescimento (P_{TC}).

Na Tab. 8.3 apresentam-se, para os 3 processos referidos na Tab. 8.1, os valores dos parâmetros PID obtidos pela metodologia proposta e pelos métodos de comparação.

	<i>PID</i>	<i>ZN</i>	<i>CC</i>	<i>AH</i>	<i>P_{ITAE}</i>	<i>P_{TC}</i>
$G(s) = \frac{e^{-0.27s}}{1+s}$	k_c	3.19	3.83	2.50	2.43	4.82
	t_i	0.69	0.75	0.69	0.86	0.51
	t_d	0.17	0.12	0.17	0.11	0.09
$G(s) = \frac{e^{-0.05s}}{(1+s)(1+0.20s)}$	k_c	5.27	6.17	4.13	9.88	10.01
	t_i	0.46	0.53	0.46	0.61	0.15
	t_d	0.11	0.08	0.11	0.13	0.07
$G(s) = \frac{e^{-0.83s}}{(1+s)(1+s)}$	k_c	1.73	2.19	1.36	0.98	2.54
	t_i	2.28	2.29	2.28	1.37	2.13
	t_d	0.57	0.37	0.57	0.88	0.43

Tab. 8.3 – Parâmetros *PID* para os vários métodos de sintonia.

Para cada processo utilizado como exemplo apresenta-se, nas Fig. 8.4 a Fig. 8.6, a sobreposição das respostas ao degrau do sistema em malha fechada, tendo em conta os parâmetros dos controladores que se registam na Tab. 8.3.. A legenda para tais figuras consta da Tab. 8.4.

—	Ziegler-Nichols
- - - - -	Cohen-Coon
—	Proposto na vertente do tempo de crescimento
—	Åström-Hägglund
—	Proposto na vertente do <i>ITAE</i>

Tab. 8.4 – Legenda para Fig. 8.4 a Fig. 8.6.

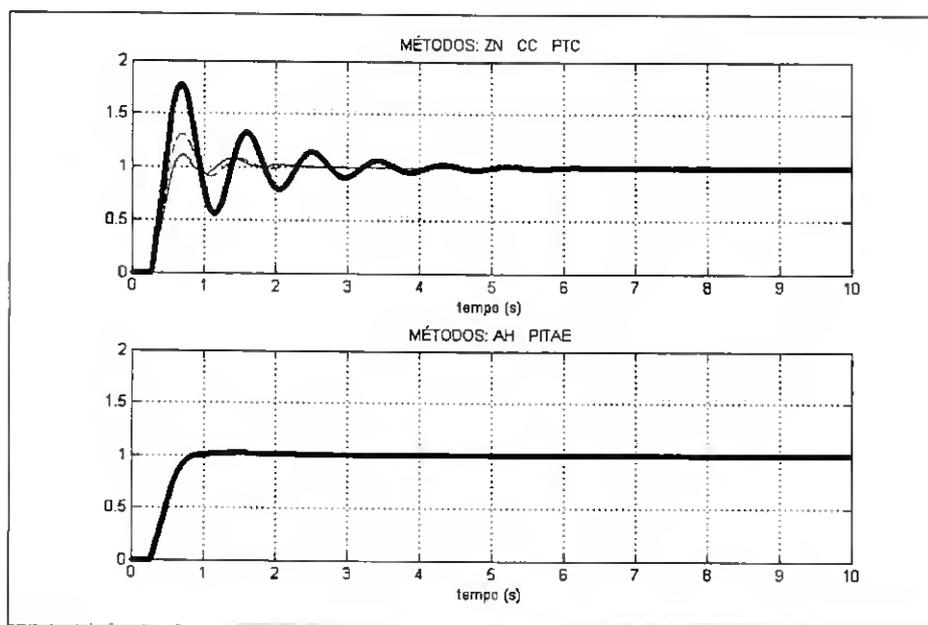


Fig. 8.4 – Respostas ao degrau do sistema em malha fechada para o processo com $T=1$ e $L=0.27$, considerando os vários métodos de sintonia.

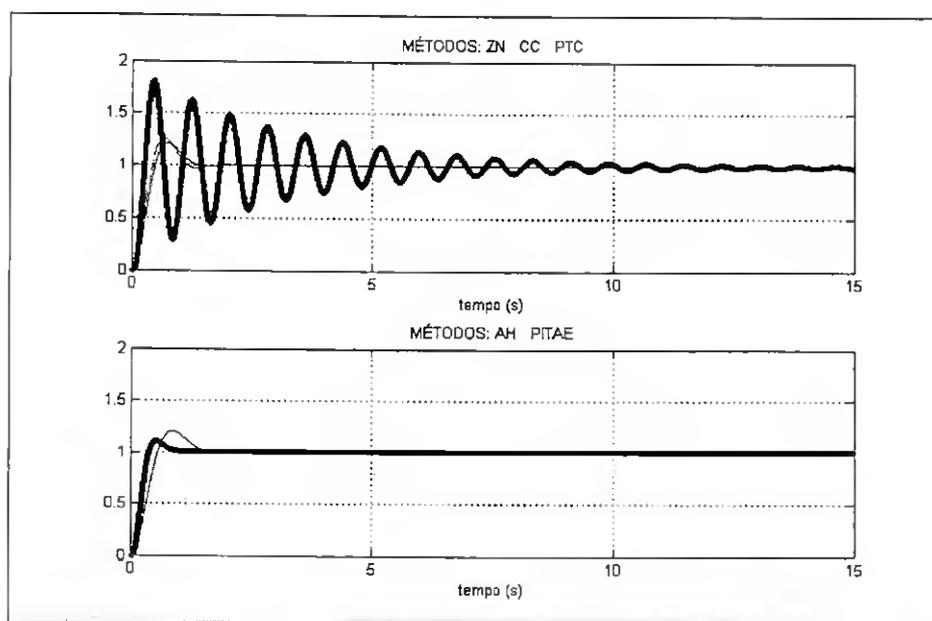


Fig. 8.5 – Respostas ao degrau do sistema em malha fechada para o processo com $T_1=1$, $T_2=0.20$ e $L=0.05$, considerando os vários métodos de sintonia.

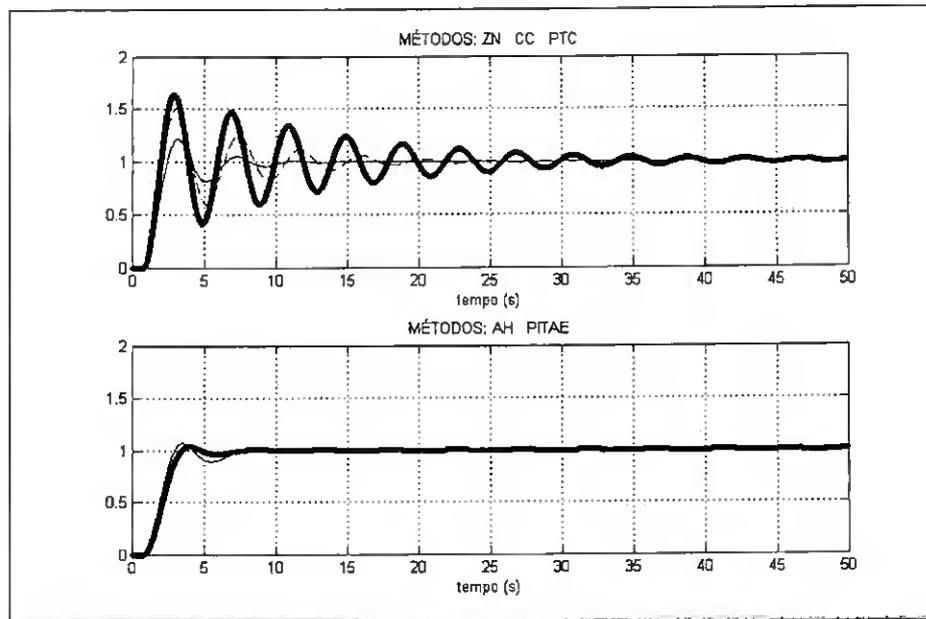


Fig. 8.6 – Respostas ao degrau do sistema em malha fechada para o processo com $T_1=1$, $T_2=1$ e $L=0.83$, considerando os vários métodos de sintonia.

A partir das respostas ao degrau que se apresentam nas Fig. 8.4 a Fig. 8.6 determinam-se os critérios de desempenho que se registam na Tab. 8.5. Para uma melhor leitura e interpretação desta tabela, para cada processo e critério de desempenho, realça-se o valor óptimo de entre os que se obtêm para os métodos de sintonia em análise.

A leitura comparativa, da evolução dos critérios de desempenho que se regista na Tab. 8.5, vai ser assistida pelos gráficos das Fig. 8.8 a Fig. 8.10 cuja legenda se apresenta na Fig. 8.7. Nessas figuras agrupam-se em gráficos de barras cada critério de desempenho para os diferentes processos, confrontando-se assim os diferentes métodos de sintonia em análise.

	Critério	<i>ZN</i>	<i>CC</i>	<i>AH</i>	<i>P_{ITAE}</i>	<i>P_{TC}</i>
$G(s) = \frac{e^{-0.27s}}{1+s}$	<i>ITAE</i>	0.20	0.22	0.23	0.19	1.09
	<i>ITAY</i>	0.22	0.24	0.17	0.15	1.21
	<i>TC(s)</i>	0.24	0.20	0.35	0.38	0.15
	<i>TS(s)</i>	1.55	1.64	0.72	0.75	3.52
	<i>S (%)</i>	11.00	30.71	4.93	2.25	77.28
$G(s) = \frac{e^{-0.05s}}{(1+s)(1+0.20s)}$	<i>ITAE</i>	0.14	0.14	0.19	0.06	5.67
	<i>ITAY</i>	0.12	0.13	0.16	0.05	6.42
	<i>TC(s)</i>	0.30	0.26	0.35	0.22	0.14
	<i>TS(s)</i>	1.16	1.04	1.36	0.73	8.77
	<i>S (%)</i>	20.64	26.29	20.31	10.66	80.52
$G(s) = \frac{e^{-0.83s}}{(1+s)(1+s)}$	<i>ITAE</i>	5.39	19.79	4.33	3.11	73.26
	<i>ITAY</i>	5.90	21.29	3.71	2.61	86.31
	<i>TC(s)</i>	1.09	0.89	1.41	1.67	0.79
	<i>TS(s)</i>	7.39	16.36	6.82	3.18	31.22
	<i>S (%)</i>	21.56	50.37	6.71	3.47	63.93

Tab. 8.5 – Critérios de desempenho para os vários métodos de sintonia.

A observação conjunta, da Tab. 8.5 e respectivas representações gráficas (Fig. 8.8 a Fig. 8.10), permite avaliar as evoluções dos critérios de desempenho face aos diferentes métodos de sintonia e tendo em conta os processos utilizados para exemplificação.

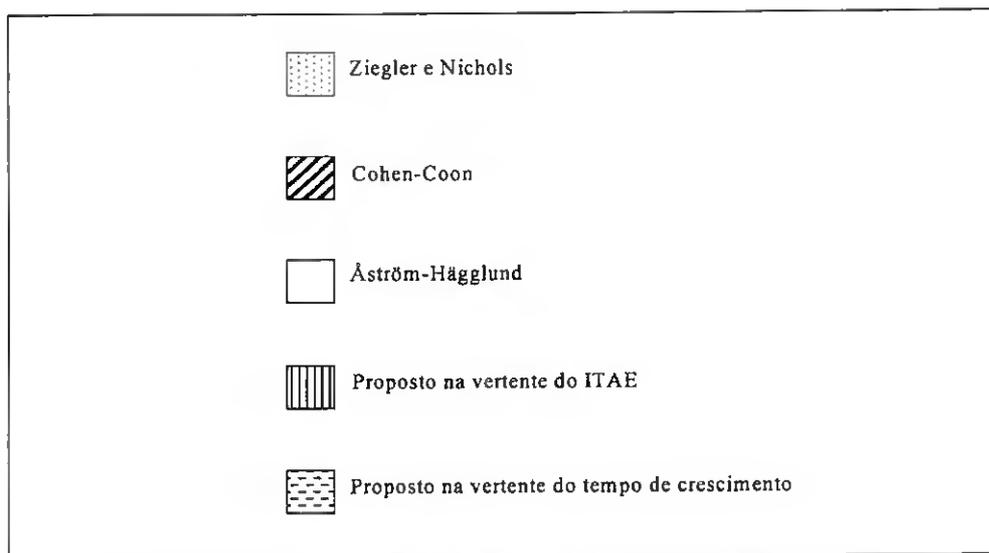


Fig. 8.7 – Legenda para os gráficos de Fig. 8.8 a Fig. 8.10.

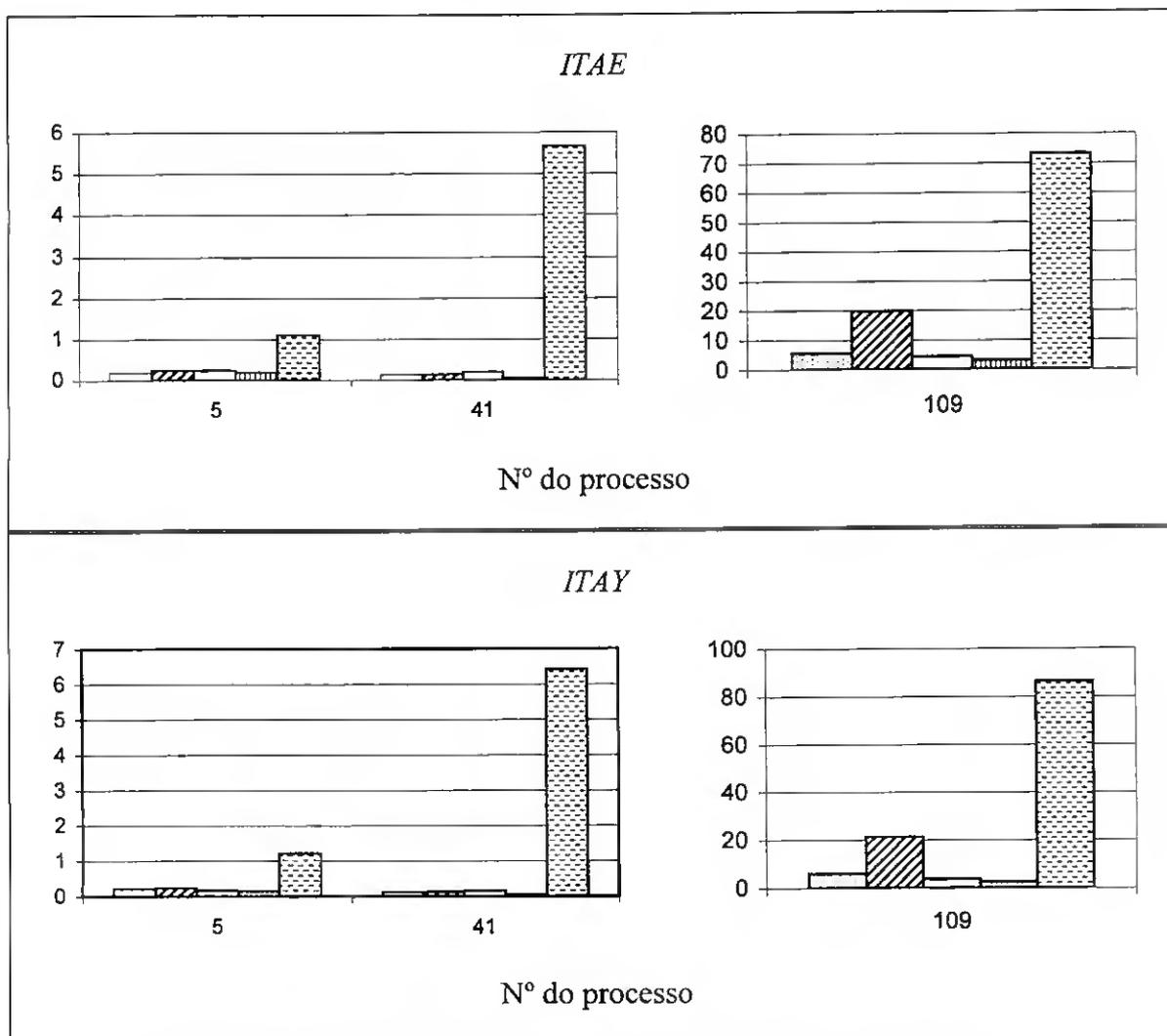


Fig. 8.8 – Evolução do *ITAE* e do *ITAY* para os exemplos em estudo.

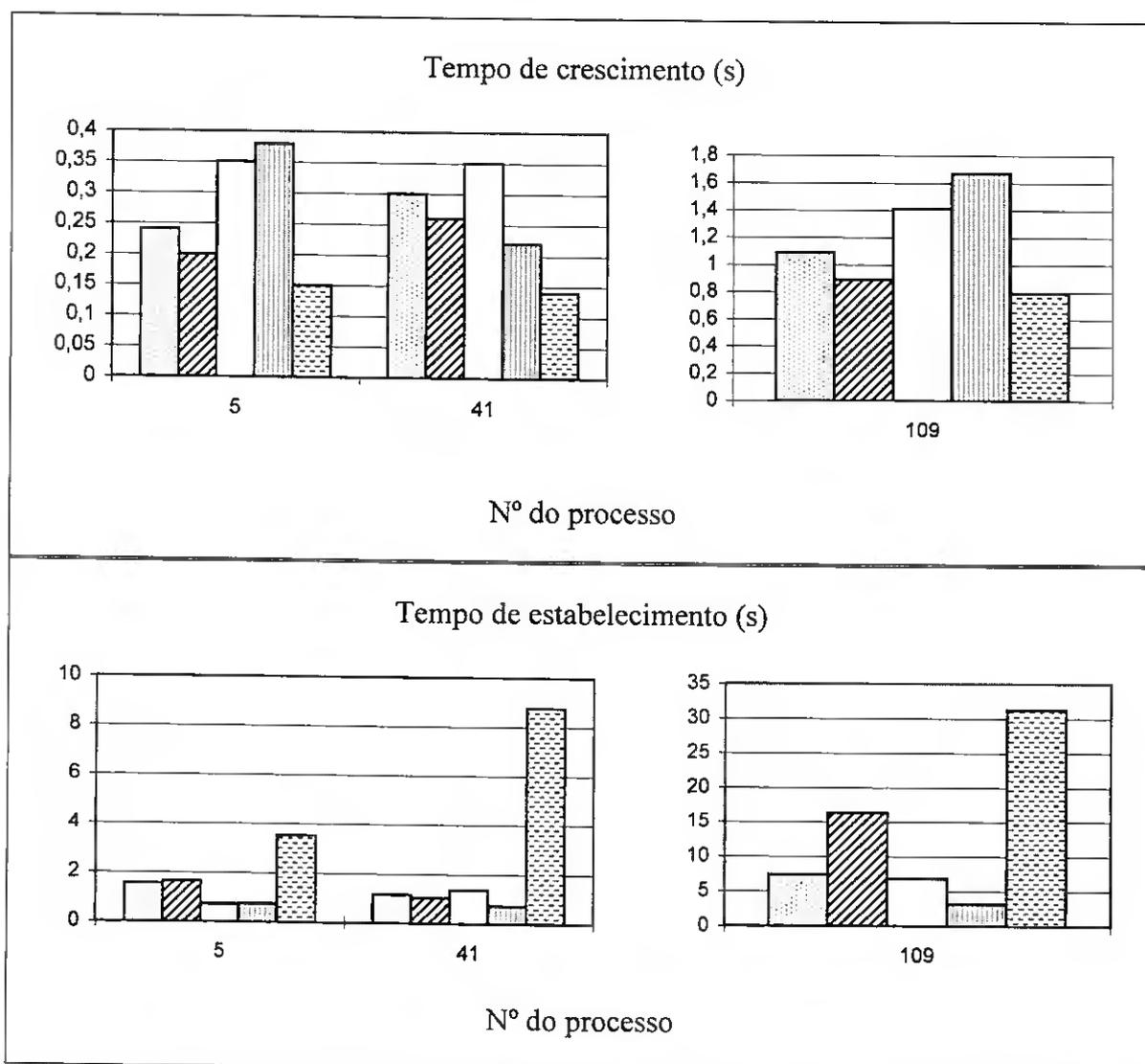


Fig. 8.9 – Evolução do tempo de crescimento e do tempo de estabelecimento para os exemplos em estudo.

O *ITAE* e o *ITAY* apresentam evoluções muito semelhantes (Fig. 8.8); de facto, para qualquer dos processos em estudo, os valores destes 2 critérios de desempenho são preponderantes quando a sintonia é feita pelo método proposto na vertente de optimização do tempo de crescimento. Considerando os restantes 4 métodos de sintonia, ambos os critérios *ITAE* e *ITAY*, mantêm-se dentro da mesma ordem de grandeza; exceptua-se o processo 109 para o qual a sintonia determinada pelo método de Cohen-Coon conduz a valores de *ITAE* e *ITAY* significativamente superiores aos que se obtêm com os outros 3 métodos de sintonia. Como seria de esperar, embora para alguns processos a diferença seja mais significativa, o menor valor de *ITAE* é verificado quando se utiliza o método de sintonia proposto na vertente de optimização deste critério; esta tendência é também verificada para o *ITAY*. De forma análoga se justificam os menores valores do tempo de

crescimento, que surgem sempre para a situação em que a sintonia utiliza o método que otimiza este critério.

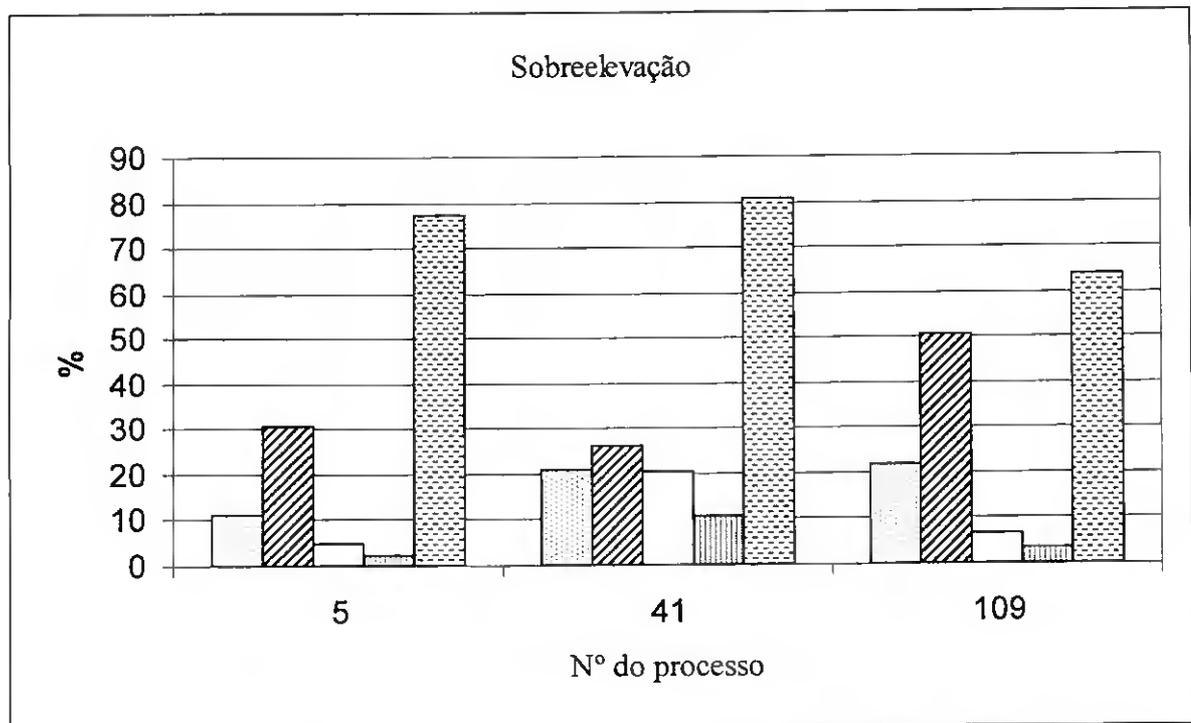


Fig. 8.10 – Evolução da percentagem de sobrelevação para os exemplos em estudo.

A observação dos exemplos em estudo sugere que a otimização do tempo de crescimento tem como consequência tornar os restantes critérios de desempenho máximos; nestas circunstâncias, as respostas ao degrau apresentam-se bastante oscilatórias tal como indicam as Fig. 8.4 a Fig. 8.6. A evolução do tempo de crescimento registada na metade superior da Fig. 8.9 permite verificar grande irregularidade face ao método de sintonia proposto na vertente da optimização do *ITAE*. Assim, para os processos 5 e 109 esta técnica de sintonia é a que corresponde aos piores valores de tempo de crescimento, enquanto que para o processo 41 é a sintonia optimizada para o *ITAE* que conduz ao melhor tempo de crescimento, logo após a sintonia optimizada para este critério. Desta forma, à parte do método proposto na vertente do *ITAE* e de acordo com a Fig. 8.9, o tempo de crescimento apresenta valores crescentes quando a sintonia é feita pela seguinte ordem: método proposto na vertente do tempo de crescimento, Cohen-Coon, Ziegler-Nichols e Åström-Hägglund.

No que respeita o tempo de estabelecimento pode-se verificar que, em geral, este critério é mínimo quando a sintonia é feita de acordo com o método proposto na vertente de

otimização do *ITAE*. A exceção acontece com o processo n° 5 para o qual o método de Åström-Hägglund conduz a uma resposta ao degrau com tempo de estabelecimento de 0.72s contra os 0.75s obtidos quando se utiliza a otimização do *ITAE*. De facto, este método de sintonia foi o que conduziu aos menores valores de todos os critérios de desempenho em análise excepto o tempo de crescimento. Regista-se ainda a similaridade das respostas ao degrau quando a sintonia é feita por estes 2 métodos: Åström-Hägglund e método proposto na vertente de otimização do *ITAE*, tal facto é testemunhado pelas metades inferiores das Fig. 8.4 a Fig. 8.6 com especial destaque no que se refere ao processo n° 5 (Fig. 8.4).

A evolução da percentagem de sobrelevação (Fig. 8.10) permite constatar que este critério de desempenho apresenta valores crescentes para os métodos de sintonia de acordo com a seguinte ordem: método proposto na vertente de otimização do *ITAE*, Åström-Hägglund, Ziegler-Nichols, Cohen-Coon e método proposto na vertente de otimização do tempo de crescimento; esta ordem pode ser igualmente deduzida observando as respostas ao degrau unitário (Fig. 8.4 a Fig. 8.6).

8.4 CONCLUSÕES

Neste capítulo apresentou-se o desempenho da metodologia proposta, comparativamente com outros métodos de sintonia do controlador *PID* tomados como referência. Para se levar a cabo tal comparação escolheram-se 3 processos do conjunto base de experimentação.

O procedimento de modelação utilizado pelos métodos de referência utiliza a resposta ao degrau do sistema em malha aberta, contrastando com a implementação proposta, segundo a qual a identificação do processo poderá ocorrer tanto em malha aberta como em malha fechada. Face a esta constatação conclui-se que, caso ocorra variação na dinâmica do processo, de acordo com os métodos de referência, terá que ser aberta a malha para que volte a ocorrer a identificação do processo, tal situação não se verifica com a metodologia proposta, dado ser possível identificação em malha fechada.

De acordo com a metodologia proposta é possível definir arbitrariamente os critérios de desempenho para os quais se pretende obter uma sintonia otimizada; na implementação apresentada esses critérios limitaram-se ao *ITAE* e o tempo de crescimento, contudo, facilmente outros critérios poderão ser incorporados. A comparação das respostas ao degrau para os exemplos utilizados permite deduzir que, como seria de esperar, os valores otimizados para os referidos critérios de desempenho são inferiores aos obtidos pelos métodos de sintonia alternativos.

A avaliação global do desempenho das respostas ao degrau obtidas face aos diferentes métodos depende da importância dada aos critérios de desempenho em causa. Esse, no entanto, é um problema do projectista que, de acordo com a finalidade em causa deverá especificar o peso relativo a atribuir a cada critério de desempenho.

9. CONCLUSÕES

9.1 CONCLUSÕES GERAIS

O controlador *PID* apresenta características tais como a sua simples estrutura que justificam a sua grande aplicabilidade. Adicionando a este facto a robustez das respostas que estes simples controladores geralmente proporcionam, compreende-se que, independentemente do facto de novas e mais sofisticadas técnicas terem surgido nos últimos anos, a quantidade de publicações relacionadas com o controlador *PID* seja elevada. O problema da sintonia do controlador *PID*, sendo fulcral no estudo deste controlador, justifica a intensa investigação que se tem verificado nessa área, tanto num plano mais ligado à teoria e fundamentos, quanto a nível de aplicações viradas para a indústria.

O constante aumento das potencialidades dos meios computacionais permite a elaboração de estudos de sistemas em geral e, em particular, dos sistemas submetidos a controlo *PID*, que conduzam a respostas com cada vez melhor desempenho. Este facto leva a que se torne cada vez mais viável o aparecimento de arquitecturas para simulação de sistemas de controlo com módulos de sintonia e possibilitou, nesta dissertação, a apresentação duma arquitectura de sintonia automática.

A grande difusão do controlador *PID* justifica a importância de que se revestem os métodos para a sua sintonia automática. Na elaboração desta dissertação foi feita uma revisão da literatura, da qual se apresenta um resumo, que permitiu concluir que as técnicas de *softcomputing* se têm revelado promissoras nesta área. Isso conduziu a que a arquitectura proposta tenha incluído sistemas de modelação neuronal e de optimização genética.

A maior parte dos métodos de sintonia automática existentes pressupõe um modelo para o processo a controlar e com base neste, são determinados os parâmetros do controlador. Nesta dissertação propõe-se uma técnica que emula o processo de sintonia manual realizado por um operador humano, modelando o comportamento do sistema global por redes neuronais e otimizando os critérios de sintonia por algoritmos genéticos.

A capacidade de modelação (mapeamento ou aproximação de funções) das redes neuronais utilizadas mostrou-se fortemente dependente do espaço para o qual é feita a aprendizagem. Concluiu-se, assim, que a escolha dos processos e dos parâmetros de funcionamento do controlador respectivo era de fundamental importância para o estabelecimento dos blocos de modelação adaptativa. Com base neste facto foi feito um levantamento de funções de transferência dos processos mais frequentemente encontrados em aplicações industriais. A gama de funções de transferência a que se chegou denominou-se conjunto base de experimentação, pois comporta os exemplos para os quais o sistema que se propôs foi preferencialmente testado.

Os parâmetros de funcionamento foram estabelecidos por imposição de critérios de desempenho. Assim, foram feitas discretizações do espaço de vectores *PID* em torno de pontos de funcionamento de interesse (óptimos), tendo em conta a avaliação de critérios de desempenho. Para que se obtivesse boa representatividade nas vizinhanças desses pontos foram estudadas duas distribuições espaciais: prismática hexagonal recta e esférica. A observação destas distribuições permitiu concluir que a distribuição esférica conduz a relações entre coordenadas mais adequadas para o processo de aprendizagem das redes neuronais, tendo sido então esta a adoptada para o estabelecimento dos padrões *PID*.

Nessa altura foi possível verificar que o espaço alvo onde se desenrolou a aprendizagem das redes neuronais era bastante extenso. Concluiu-se então que, se por um lado o aumento de exemplos dentro dum determinado espaço de aprendizagem contribui para que a modelação seja mais robusta, por outro lado esse aumento condiciona fortemente o funcionamento dos algoritmos de treino das redes neuronais face ao *hardware* de que se dispõe.

Assim se concluiu ainda que a escolha da quantidade de padrões de aprendizagem tinha de ser criteriosa, para que se conseguisse uma modelação tão fiável quanto possível, com um

número de padrões comportável para os algoritmos de treino utilizados e correspondente *hardware*.

A revisão da literatura atrás referida, concretamente a respeitante a técnicas de *softcomputing*, permitiu concluir que actualmente os tipos de redes neuronais utilizados são diversos, reclamando-se, frequentemente, cada um como sendo apropriado para determinada aplicação ou grupo de aplicações. Nesta dissertação foram estudados três tipos de redes neuronais, escolhidos de entre os mais vulgarmente utilizados na grande maioria das aplicações da área: *MLP*, *RBF* e *B-Spline*.

Antes da inclusão dum destes tipos de redes neuronais no sistema de auto-modelação, ocorreu um processo de pré-selecção para se avaliar qual o tipo de rede mais apropriado para a aplicação em estudo. Durante este procedimento preliminar restringiu-se o espaço de aprendizagem para que houvesse maior controlo sobre a exactidão e funcionamento dos algoritmos respectivos. As conclusões tiradas nesta fase partiram do pressuposto de que o desempenho de determinado tipo de rede neuronal para um sub-espaço de aprendizagem pode ser extrapolado à totalidade do referido espaço.

Para o fim em vista decidiu-se testar comparativamente os três tipos de redes neuronais mencionados atendendo aos seguintes aspectos:

- exactidão,
- capacidade de generalização,
- capacidade de adaptação,
- complexidade do modelo obtido.

Sendo o objectivo último do projecto no qual se insere esta dissertação o funcionamento do sistema de auto-sintonia do controlador *PID* em linha, foi dada primazia à rede neuronal que apresentasse melhor capacidade de adaptação sem que a aprendizagem em diferido seja "esquecida".

Concluiu-se que as redes neuronais do tipo *B-Spline* são as que melhor cumprem os objectivos do parágrafo anterior.

Um dos problemas típicos das redes neuronais do tipo *B-Spline* é conhecido como maldição dimensional. Este fenómeno ocorre quando o treino utiliza padrões com muitas entradas, como se verifica no presente caso, onde se consideram como entradas as medidas de identificação do processo juntamente com os parâmetros do controlador; nessas circunstâncias o algoritmo poderá produzir modelos com grande complexidade. Para se evitar esta situação propôs-se uma estratégia segundo a qual se obtêm modelos de topologia parcialmente fixa. A aprendizagem de redes neuronais de acordo com este método é feita em duas fases. Na primeira fase, para cada processo é estabelecido um modelo determinístico de 2ª ordem do critério de desempenho em função do *PID*. Seguidamente é feito um mapeamento das medidas de identificação, nos parâmetros dos modelos de segunda ordem. Esta estratégia permite decompor o espaço de entrada, transformando um problema de mapeamento com 5 entradas (modelos de estrutura livre), em 10 mapeamentos diferentes cada um com 2 entradas.

O estudo desenvolvido na âmbito da aplicação dos algoritmos genéticos permite concluir que o processo de optimização utilizando estes algoritmos juntamente com as redes neuronais para avaliação da função objectivo conduz a melhores resultados, quando comparados com os que se obtêm com a utilização de métodos gradientes com recurso à simulação dinâmica do processo. Assim, os algoritmos genéticos foram utilizados no bloco de optimização da arquitectura proposta. Verificou-se ainda, no que respeita à optimização, que a avaliação da função objectivo por propagação de padrões por redes neuronais consome menos tempo do que a avaliação da mesma função por simulação dinâmica do processo; este aspecto é fulcral nas aplicações em tempo real. Os procedimentos de optimização levados a cabo durante o funcionamento do sistema de auto-sintonia utilizaram como função objectivo cada um dos critérios de desempenho em estudo (*ITAE* e tempo de crescimento), bem como uma soma ponderada de ambos.

Deve ser notado que a utilização de um procedimento de optimização em linha poderá conduzir, durante o seu funcionamento, a iterações instáveis, situação esta que é incompatível com a grande maior parte das aplicações reais, o que reforça o benefício que constitui a utilização de modelos.

Finalmente, na prática, um processo real não pode ser sujeito a um elevado número de perturbações necessárias à conclusão de um procedimento de optimização, que, só se torna viável se utilizar modelos que aproximem o seu funcionamento.

Resumindo, para a arquitectura proposta foram construídos:

- 2 bloco de modelação dos respectivos critérios de desempenho em estudo.
- 1 bloco de sintonia, sendo cada um constituído por 3 redes neuronais treinadas para os parâmetros *PID* optimizados para o respectivo critério de desempenho.
- 1 bloco de optimização genética responsável pela determinação dos parâmetros *PID* óptimos, necessários à adaptação dos blocos de sintonia a que se refere o ponto anterior.

A integração dos blocos, referidos nos pontos anteriores, na arquitectura proposta recorreu ao ambiente de simulação dinâmica e visualização gráfica SIMULINK. O sistema assim desenvolvido permitiu a obtenção de resultados segundo os quais se pode afirmar que:

- A auto-sintonia neuro-genética é conseguida tanto para a optimização de cada critério de desempenho isolado como para a optimização de uma soma ponderada de ambos os critérios.
- O carácter adaptativo do sistema proposto possibilita a variação da dinâmica do processo ao longo do procedimento de auto-sintonia.

9.2 SUGESTÕES DE TRABALHO FUTURO

As sugestões de trabalho futuro que se apresentam nesta secção destinam-se a projectos que integram a investigação continuada, no âmbito da qual se insere a presente dissertação. Assim, os projectos sugeridos podem-se dividir em dois grupos relacionados com as duas tarefas principais que geralmente integram um sistema de sintonia automática e que estão presentes na arquitectura proposta: modelação e optimização.

Modelação: Foi feita a modelação dos critérios de desempenho *ITAE* e tempo de crescimento, construindo-se assim 2 blocos de critério da arquitectura de auto-sintonia proposta. Contudo, esta arquitectura está preparada para acomodar um número arbitrário de critérios de desempenho que, de forma análoga ao que acontece com os 2 critérios agora

utilizados, poderão ser ponderados no cálculo da função uni-objectivo do procedimento de optimização. Como critério adicional pode-se começar por considerar a percentagem de sobre-elevação. de acordo com a geometria adoptada para a selecção dos vectores *PID*, os mínimos deste critério encontram-se incluídos nos volumes definidos pelos vectores *PID* utilizados até agora.

Os procedimentos de adaptação são fortemente dependentes das respectivas taxas. Estudos realizados até este momento apontam para que as taxas de adaptação dos modelos dos critérios sejam superiores às que se utilizam nos modelos de sintonia do controlador. Desta forma pretende-se na sequência desta dissertação, aprofundar o estudo sobre quais as taxas de aprendizagem mais apropriadas para ambas as situações, modelação de critério e sintonia.

A adaptação em linha de ambos os tipos de redes neuronais (de critério e de sintonia) visa a obtenção dos respectivos valores mínimos. Contudo, a determinação dos modelos em diferido introduz erros relativamente aos valores exactos determinados por simulação dinâmica; consequentemente, os mínimos para os quais se faz a aproximação durante a adaptação em linha, em princípio, não coincidem com os verdadeiros mínimos. Desta forma, existe a necessidade de que o processo adaptativo em linha seja supervisionado. Concretamente, para cada um dos sub-processos adaptativos deve ser verificada a evolução de respectivamente:

- Sub-processo adaptativo para modelação de critérios: Valores dos critérios de desempenho avaliados por propagação pelas respectivas redes e correspondentes valores reais avaliados numericamente a partir da resposta obtida.
- Sub-processo adaptativo de sintonia: Parâmetros k_c , t_i e t_d avaliados pelas redes de sintonia e correspondentes valores resultantes do procedimento de optimização.

Para cada um dos pontos anteriores, as diferenças entre as medidas acima referidas definem os erros responsáveis pela adaptação de cada sub-processo. Desta forma, deverá ser estudada a convergência das medidas necessárias ao cálculo dos erros para os 2 processos de adaptação que integram esta arquitectura. Considerando o desenrolar do processo adaptativo para um sistema invariante no tempo, pretende-se que os modelos de critério se adaptem até ao ponto em que o seu mínimo coincida com o mínimo real. Nestas

circunstâncias a adaptação dos respectivos modelos deixa de se verificar para o ponto de funcionamento em causa. Se o mínimo atingido pelo modelo coincidir com o valor real do critério, mas não com o seu mínimo real, o sistema não tem possibilidade de sair desta situação a menos que se altere o ponto de funcionamento. O estudo de estratégias de controlo que permitam resolver este problema são de grande interesse para sistemas auto-adaptativos e podem-se considerar como sugestões de trabalho para projectos seguintes.

A obtenção de modelos fiáveis é fundamental no processo de sintonia automática do controlador *PID*. Mantendo-se a hipótese de modelação neuronal podem ser explorados outros paradigmas de treino mais eficiente, nomeadamente no que respeita ao particionamento do espaço de aprendizagem das redes neuronais.

O número de nós intermédios das redes neuronais do tipo *MLP* foi estabelecido empiricamente; os métodos e técnicas que exploram o dimensionamento deste tipo de redes poderão ser desenvolvidos no sentido de uma determinação mais eficiente do número óptimo de nós intermédios.

No que diz respeito às redes neuronais do tipo *RBF* utilizaram-se métodos empíricos para determinação do número de nós e respectivos posicionamentos dos centros e raios. Alternativamente, pode-se propor a determinação destes parâmetros através de algoritmos baseados no cálculo de gradientes descendentes das respectivas funções de erro. Uma proposta semelhante pode ser feita relativamente ao posicionamento dos nós para o algoritmo de treino das redes neuronais do tipo *B-Spline*.

A estrutura dos modelos neuronais é fixa e determinada pelo processo de aprendizagem em diferido. Seria interessante verificar o desempenho de algoritmos construtivos em linha que, para além de variarem os parâmetros da rede neuronal, modifiquem também a sua estrutura.

O aumento da velocidade de treino é fundamental em geral, para qualquer aplicação que utilize redes neuronais e em particular para situações que visem a sua implementação em tempo real. Sendo o processo de aprendizagem de redes neuronais aquele que requer maior desempenho computacional, será importante avaliar diferentes arquitecturas computacionais para que os requisitos temporais possam ser cumpridos. Uma forma de se

proceder a treinos mais eficientes será, por exemplo, a exploração de outros tipos de *hardware* para aplicação específica.

Optimização: Os algoritmos genéticos simples revelaram bom desempenho na aplicação em estudo. Contudo, havendo a necessidade de se trabalhar com populações de maior dimensão, pode-se pensar na inclusão de outros operadores genéticos tais como migração entre sub-populações, ou melhoramento dos existentes, com vista a um aumento da eficiência do procedimento de optimização. Numa aplicação em tempo real, o número de gerações deverá ser tão pequeno quanto possível; para isso deverá ser explorado o conhecimento de que, caso não ocorra variação da dinâmica do processo, a solução em funcionamento após certo número de transições da onda quadrada de entrada, poderá estar muito próxima do valor óptimo.

Alternativamente à optimização uni-objectivo com ponderação de critérios utilizada na presente aplicação, a arquitectura de auto-sintonia proposta possibilita a acomodação de optimização multi-objectivo gerando-se dessa forma vectores de soluções em vez de solução única. Assim, poder-se-ia desenvolver heurísticas que permitissem a escolha duma solução de entre um conjunto de soluções não dominadas obtidas pelo algoritmo genético.

Voltando à questão da construção de modelos, relembra-se que o treino de redes neuronais pode ser visto como um problema de optimização. Desta forma, como sugestão de trabalho futuro, pode-se pensar na utilização para esta aplicação de algoritmos genéticos para resolver o problema de optimização que consiste no treino em diferido de redes neuronais, bem como a sua adaptação em linha.

Tendo esta arquitectura sido testada em simulação, poder-se-á modificar os algoritmos implementados, para que a arquitectura proposta possa ser utilizada para controlo em tempo real.

Por último, a estabilidade de todo o sistema proposto deveria ser estudada.

Referências

- [1] A. E. Ruano, J. M. Lima, R. Mamat, P. J. Fleming, "Comparison of Alternative Approaches to Neural Network PID Autotuning", *Journal of Systems Engineering*, Vol. 6, n°3, Springer-Verlag, 1996, pp.166-176.
- [2] A. Ruano, J. Lima, "New Methods for PID autotuning", *3rd Portuguese Conference on Automatic Control*, Coimbra, Portugal, September, 1998, Vol. 1, pp. 187-192.
- [3] J. Lima, A. Ruano, "Neuro-genetic PID autotuning: time invariant case", in *IMACS Journal of Mathematics and Computers in Simulation*, Vol. 51, Issue No. 3-4, Elsevier, 2000, pp. 287-300.
- [4] A. E. Ruano, J. M. Lima, A. B. Azevedo, N. M. Duarte, P. J. Fleming, "Automatic Tuning of Controllers Using a Neural-Genetic System", *14th IFAC World Congress*, Beijing, P. R. China, July 5-9, 1999, in CD-ROM.
- [5] J. Lima, A. Ruano, "On-line Adaptation of Neural Networks", *Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'99)*, Warsaw, Poland, September 13-15, 1999, pp. 159-164.
- [6] J. M. Lima, A. E. Ruano, "Comparison of off-line and on-line performance of alternative Neural Network models", *8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference (IPMU 2000)*, Madrid Spain, July 3-7, 2000, pp. 1219-1226.
- [7] J. M. Lima, A. B. Azevedo, N. Duarte, C. M. Fonseca, A. E. Ruano, P. J. Fleming, "Neuro-genetic PID autotuning", *European Control Conference 2001 (ECC' 2001)*, Porto, Portugal, September 4-7, 2001, in CD-ROM.
- [8] K. Åström, T. Hägglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, NC, USA, second edition, 1995.
- [9] L. Muhidin, G. Zoran, "A reference guide to PID controllers in the nineties", in *IFAC Workshop on Digital Control: Past, present and future of PID Control*, (PID'00), Terrassa, Spain, April 2000, pp.73-82.

- [10] 1999 American Control Conference, Hyatt Regency San Diego, California, USA, June 2-4, 1999.
- [11] 14th IFAC World Congress, Beijing, China, July 5-9, 1999.
- [12] M. H. Moradi, M. A. Johnson, "New techniques for PID Controller Design", in *Proceedings of 2003 IEEE Conference on Control Applications*, Istanbul, Turkey, June 23-25, 2003, pp. 903-908.
- [13] K. J. Åström, T. Hägglund, C. C. Hang, W. K. Ho, "Automatic tuning and adaptation for PID controllers - a survey", in *Control Engineering Practice*, Vol. 1, 1993, pp. 699-714.
- [14] R. Gorez, G. Calcev, "A survey of PID auto-tuning methods", *Proceedings of 11th Conference on Control Systems and Computer Science*, Vol. 1, 1997, pp. 18-27.
- [15] K. K. Tan, Q. G. Wang, C. C. Hang, *Advances in PID Control*, Springer Verlag London, ISBN-1-852333-138-0.
- [16] P. Cominos, N. Munro, "PID controllers: recent tuning methods and design to specification", in *IEE Proceedings Control Theory Applications*, Vol. 149, No. 1, January 2002.
- [17] M. Morari, E. Zafiriou, *Robust Process Control*, Prentice Hall, 1989.
- [18] D. E. Rivera, S. Skogestad, M. Morari, "Internal Model Control 4. PID Controller Design", in *Industrial Engineering Chem. Proc. Des & Dev*, Vol. 25, 1986, pp. 252-265.
- [19] I. L. Chien, "IMC-PID Controller Design - An Extension", in *IFAC Proceeding Series*, 6, pp. 147-152, 1988.
- [20] M. J. Grimble, "H-infinity PID Controller", *Transactions on IMC*, Vol. 13, No 5, 1991, pp. 112-120.
- [21] Q. G. Wang, C. C. Hang, X. P. Yang, "Single-Loop Controller Design Via IMC Principles", *In Proceeding Asian Control Conference*, Shanghai, P. R. China, 2000.

- [22] I. Rusnak, "Generalised PID Controllers", in *The 7th IEEE Mediterranean Conference on Control & Automation*, (MED 99), Haifa, Israel, 1999.
- [23] I. Rusnak, "The Generalised PID Controllers and its Application to Control of Ultrasonic and Electric Motors", in *IFAC Workshop PID'00*, Spain, 2000, pp. 125-130.
- [24] K. K. Tan, S. N. Huang, T. H. Lee, "Development of a GPC-based PID controller for unstable systems with dead time", in *ISA Transaction*, Vol. 39, 2000, pp. 57-70.
- [25] M. R. Katebi, M. H. Moradi, "Predictive PID Controllers", in *IEE Proceedings of Control Theory Application*, Vol.148, No. 6, November 2001, pp. 478-487.
- [26] M. H. Moradi, M. R. Katebi, M. A. Johnson, "The MIMO Predictive PID Controller Design", in *Advances in PID Control Special Issue of Asian Journal of Control*, Vol. 4, No. 4, 2002, pp. 452-463.
- [27] J. G. Ziegler, N. B. Nichols, "Optimum settings for automatic controllers", *Transactions on ASME*, 1942, pp. 759-768.
- [28] A. Ruano, *Applications of Neural Networks to Control Systems*, PhD Thesis, University of Wales, UK, February, 1992.
- [29] A. Leva, C. Cox, A. Ruano, "Hands-on PID autotuning: a guide to better utilisation", *IFAC Professional Brief*, <http://www.oeaw.ac.at/ifac/>, 2002.
- [30] R. Isermann, K. H. Lachman, D. Matko, *Adaptive Control System*, Prentice-Hall, 1992.
- [31] F. L. Pagola, R. R. Pecharromán, "PID auto-tuning based on a second point of frequency response", in *IFAC 15th World Congress*, Barcelona Spain, 21st-26th July 2002.
- [32] R. R. Pecharromán, F. L. Pagola, "Control Design for PID Controllers Auto-tuning Based on Improved Identification", in *IFAC Workshop on Digital Control: Past, present and future of PID Control*, (PID'00), Terrassa, Spain, April 2000, pp.89-94.

- [33] A. H. Jones, P. B. M. Oliveira, "Genetic Auto-Tuning of PID Controllers", *Proceedings of the First IEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA'95)*, No. 414, September, Sheffield, UK, pp. 141-145.
- [34] A. H. Jones, P. B. M. Oliveira, "Auto-Tuning of PI Smith Predictor Controllers using Genetic Algorithms", *UKACC International Conference on CONTROL '96*, September, 1996, pp. 454-459.
- [35] A. H. Jones, M. L. Tatnall, "Automated On-line Frequency-Domain Identification and Genetic Estimation of Plant Transfer Function", in *Proceedings SYSID*, Copenhagen, 1994.
- [36] A. H. Jones, Y. Lin, P. B. M. Oliveira, S. B. Kenway, "Auto-tuning of Dual Mode Controllers Using Genetic Algorithms", in *2nd IEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, September, Glasgow, Scotland, No. 446, 1997, pp. 516-521.
- [37] K. J. Åström, T. Hägglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Ratios", *Automatica*, Vol. 20, 1988, pp. 645-651.
- [38] P. B. M. Oliveira, A. H. Jones, "Co-evolutionary design of PID control structures", *IFAC Workshop on Digital Control: Past, Present and Future of PID Control, (PID'00)*, Terrassa, Spain, April 2000, pp. 325-330.
- [39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley P.C., 1989.
- [40] W. D. Hamilton, "Pathogens as causes of genetic diversity in their host", In *Population Biology of Infectious Diseases*, Anderson A. and May R M (Ed.), Springer-Verlag, 1987, pp.269-296.
- [41] R. A. Krohling, J. P. Rey, "Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms", in *IEEE Transactions on Evolutionary Computation*, Vol. 5 No. 1, February 2001, pp. 78-82.

- [42] R. A. Krohling, "Design of a PID controller for disturbance rejection: A genetic optimisation approach", in *Proceedings 2nd IEE/IEEE International Conference Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, Scotland, 1997, pp. 498-503.
- [43] A. Visioli, "Tuning of PID controllers with fuzzy logic", in *IEE Proceedings - Control Theory Applications*, Vol. 148, No. 2, 2001, pp. 180-184.
- [44] C. Houck, J. Joines, M. Kay, *A genetic algorithm for function optimization: A Matlab implementation*. NCSU-IE Technical Report 95-09, 1995.
- [45] S. G. Tzafestas, N. P. Papanikolopoulos, "Incremental fuzzy expert PID control", in *IEEE Transactions on Industrial Electronic*, Vol. 37, No. 5, 1990, pp.365-371.
- [46] P. J. Macvicar-Whelan, "Fuzzy sets for man-machine interaction", in *International Journal Man-Machine Studies*, Vol. 8, 1976, pp.687-697.
- [47] S. Z. He, S. Tan, F. L. Xu, "Fuzzy self-tuning of PID controllers", *Fuzzy Sets Systems*, 1993, pp. 37-46.
- [48] Zhao, Z. Y., Tomizuka, M., Isaka, S., "Fuzzy gain scheduling of PID controllers", in *IEEE Transactions on Systems Man, Cybernetic*, Vol. 23, No. 5, 1993, pp. 1392-1398.
- [49] A. Visioli, "Fuzzy logic based set-point weight tuning of PID controllers", *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, Vol. 29, No. 6, 1999, pp. 587-592.
- [50] Liu, C., Xu, J.-X., Hang, C.-C., "Comparison between a fuzzy PID controller and a kind of nonlinear PID controller", in *Proceedings 36th IEEE International Conference on Decision and control*, S. Diego (CA), 1997, pp.2736-2741.
- [51] F. G. Shinskey, *Process control systems: Application, design, and adjustment*, McGraw-Hill, 1998.
- [52] Q. Li, S. K. Tso, A. N. Poo, "PID Tuning Using Neural Networks", TSI Press Albuquerque, NM USA, 1998.

- [53] S. Akhyar, S. Omatu, "Neuromorphic self-tuning PID controller", *Proc. of IEEE International Conference on Neural Networks*, 1992, pp. 552-557.
- [54] B. Song, A. Koivo, "Neural adaptive control of excavators", *International Conference on Intelligent Robotics and Systems*, 1995.
- [55] S. Omatu, T. Fujinaka, Y. Kishida, M. Yoshioka, "Self-Tuning Neuro-PID for SIMO systems", *European Control Conference (ECC'99)*, 1999.
- [56] Y. Kishida, S. Omatu, M. Yoshioka, "Self-Tuning Neuro-PID for Stabilizing of Double Inverted Pendulum", *Journal of Robotic and Mechatronics*, Vol.10, No. 5, 1998, pp. 439-444.
- [57] A. Ruano, J. Lima, R. Mamat, P. Fleming, "A Neural Network PID Autotuner", *Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'95)*, Otaniemi/Helsinki, Finland, August 21-23, 1995, pp. 167-170.
- [58] M. Brown, C. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, 1994.
- [59] S. Haykin, *Neural Networks a comprehensive foundation*, 2nd Edition, Prentice Hall, 1998.
- [60] K. S. Fu, "Learning Control System and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control", in *IEEE Transaction on Automatic Control*, Vol. 16, 1971, pp. 70-72.
- [61] G. N. Saridis, K. P. Valavanis, "Analytical Design of Intelligent Machines", in *Automatica*, Vol. 24, N° 2, 1988, pp. 123-133.
- [62] W. Bloom, D. W. Fawcett, *A Textbook of Histology*, Saunders Company, 1975.
- [63] H. Gray, *Gray's Anatomy*, Churchill Livingstone, Edinburgh, 1979.
- [64] F. Rosenblatt, *Principles of Neurodynamics*, Spartan Books, 1962.

- [65] B. Widrow, *An adaptive 'Adaline' neuron using chemical 'memistors'*, Stanford Electronics Labs., Stanford, USA, Technical Report 1553-2, 1960.
- [66] B. Widrow, M. Hoff, "Adaptive switching circuits", in *1960 WESCON Convention Record*, 1960, pp. 96-104.
- [67] B. Widrow, F. Smith, "Pattern recognising control systems", in *Computer and Information Sciences, Symposium Proceedings*, Spartan Books, 1963.
- [68] L. Almeida, "Backpropagation in non-feedforward networks", in Igor Aleksander (ed.) *Neural computing architectures: the design of brain-like machines*, North Oxford Academic, 1989, pp. 74-91.
- [69] J. Pineda, "Generalisation of Backpropagation to recurrent and higher-order neural networks", in *Proceedings of the 1987 IEEE Conference on Neural Information Processing Systems – Natural and Synthetic*, 1988, pp. 602-611.
- [70] Y. Le Cun, "Une procedure d'apprentissage pour reseau a seuil assymetrique", in *Proc. Cognitiva*, Vol. 85, 1985, pp. 599-604.
- [71] D. Parker, "Learning-Logic", *Technical Report. TR-47*, Center for Computational Research in Economics and Management Science, MIT, MA, 1985.
- [72] D. E. Rumelhart, J. L. McClelland (Eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge, MA, 1986.
- [73] P. J. Werbos, *Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Masters Thesis, Harvard University, 1974.
- [74] V. S. Asirvadam, *On-line learning and construction for Neural Networks*, PhD Thesis, Faculty of Engineering of Queen's University Belfast, December 2002.
- [75] E. Hartman, J. D. Keeler, "Predicting the Future: Advantages of Semilocal Units", in *Neural Computation*, Vol. 3, N° 4, 1991, pp. 566-578.
- [76] A. E. B. Ruano, P. J. Fleming, D. I. Jones, "Connectionist Approach to PID autotuning", in *IEE Proceedings Part-D*, vol.139, n°3, 1992, pp. 279-285.

- [77] M. T. Hagan, M. B. Henhaj, "Training FeedForward Networks with Marquardt Algorithm", in *IEEE Transaction Neural Network*, Vol. 5, part 6, November 1994, pp. 989-993.
- [78] M. J. D. Powell, "Radial Basis Functions for multivariable interpolation: A Review", in *Algorithms for Approximation of Functions and Data*, Eds. J. C. Mason, M. G. Cox, Oxford University Press, 1987, pp. 143-167.
- [79] M. D. Buhmann, M. J. D. Powell, "Radial Basis Function Interpolation on an Infinite Grid", in *Algorithms for Approximation II*, Eds. J. C. Mason, M. G. Cox, Chapman and Hall, London, 1990, pp. 146-169.
- [80] D. S. Broomhead, D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks", in *Complex Systems, Vol. 2*, 1988, pp. 321-355.
- [81] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press Inc., 1998.
- [82] R. O. Duda, P. E. Hart, "*Pattern Classification and Scene Analysis*", New York: Wiley Ed., 1973.
- [83] The Math Works Inc., *Matlab Neural Network Toolbox User's Guide*, Version 2, Third printing, January, 1994.
- [84] S. Chen, C. F. N. Cowan, P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks", *IEEE Transactions on Neural Networks*, Vol. 2, N° 2, March 1991, pp. 302-309.
- [85] J. L. Carmo, *Redes de Funções de Base Radiais: Identificação e Aplicação na Previsão de Séries Irregularmente Espaçadas*, Dissertação de Mestrado, Universidade de Lisboa, 1997.
- [86] M. G. Cox, "The Numerical Evaluation of B-splines", in *J. Inst. Math. Appl.*, Vol. 10, 1972, pp. 134-149.
- [87] C. DeBoor, "On Calculating with B-splines", in *J. Approximation Theory*, Vol. 6, 1972, pp.50-62.

- [88] T. Kavli, *Learning Principles in Dynamic Control*, Dr. Sci. Thesis, Institute for Informatics, University of Oslo, Norway, 1992.
- [89] T. Kavli, "ASMOD - an Algorithm for Adaptive Spline Modelling of Observation Dada", in *Advances in Intelligent Control*, Ed Harris C. J., Taylor and Francis, London, Chapter 6, 1974.
- [90] C. DeBoor, *A Practical guide to Spline*, Springer-Verlag, NY, 1978.
- [91] A. E. B. Ruano, "*Artificial Neural Network*", Texto de apoio da disciplina de Redes Neurais, Univ. do Algarve, 2002.
- [92] C. L. Cabrita, *Algoritmos construtivos para treino de redes neuronais do tipo B-Spline*, Dissertação de Mestrado, Universidade do Algarve, 2001.
- [93] C. Fonseca, *Multiobjective Genetic Algorithms with Application to Control Engineering Problems*, PhD Thesis, The University of Sheffield, UK, September, 1995.
- [94] K. De Jong, D. B. Fogel, H. Schwefel, "A history of evolutionary computation", in *Handbook of Evolutionary Computation*, Editors in Chief T. Bäck, D. B. Fogel, Z. Michalewicz, Oxford University Press, 1997, USA, pp. A2.3:1-A2.3:12.
- [95] R. M. Friedberg, "A learning machine: part I", *IBM J.* 2, 1958, pp. 2-13.
- [96] R. M. Friedberg, B. Dunham, J. H. North, "A learning machine: part II", *IBM J.* 3, 1959, pp. 282-287.
- [97] H. J. Bremermann, "*Optimisation through evolution and recombination Self-Organizing Systems*", ed M. C. Yovits *et al* (Washington, DC: Spartan), 1962.
- [98] G. E. P. Box, "Evolutionary operation: a method for increasing industrial productivity", *Appl. Stat.*, nº 6, 1957, pp. 81-101.
- [99] G. E. P. Box, N. P. Draper, *Evolutionary operation. A Method for Increasing Industrial Productivity*, New York: Wiley, 1969.

- [100] W. Spendley, G. R. Hext, F. R. Himsforth, "Sequential application of simplex designs in optimisation and evolutionary operation", *Technometrics*, 4, 1962, pp. 441-461.
- [101] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, New York: Wiley, 1966.
- [102] J. H. Holland, "Nonlinear environments permitting efficient adaptation", *Computer and Information Sciences II*, New York: Academic, 1967.
- [103] I. Rechenberg, "Cybernetic Solution Path of an Experimental Problem", Royal Aircraft Establishment Library Translation 1122, 1965.
- [104] H. P. Schwefel, R. Männer (eds), "Parallel Problem Solving from Nature", *Proceeding 1st Workshop PPSN I, Lecture Notes in Computer Science 496*, Berlin: Springer, Dortmund, 1991.
- [105] J. H. Holland, "Outline for a logical theory of adaptive systems", in *Journal of ACM*, 9, 1962, pp. 297-314.
- [106] J. H. Holland, "Adaptive plans optimal for payoff-only environments", *Proceeding 2nd Hawaii International Conference on System Sciences*, 1969, pp. 917-920.
- [107] A. D. Bethke, *Genetic Algorithms as Function Optimisers*, PhD Thesis, University of Michigan, 1981.
- [108] L. Booker, *Intelligent Behavior as an Adaptation to the Task Environment*, PhD Thesis, University of Michigan, 1982.
- [109] D. E. Goldberg, *Computer-Aided Gas Pipeline Operation using Genetic Algorithms and Rule Learning*, PhD Thesis, University of Michigan, 1983.
- [110] D. E. Goldberg, K. Ded, B. Korb, "Don't worry, be messy", *Proceeding 4th International Conference on Genetic Algorithms*, San Diego, CA, ed. R. K. Belew and L. B. Booker, San Mateo, CA: Morgan Kaufmann, 1991, pp. 24-30.
- [111] J. R. Koza, *Genetic Programming*, Cambridge, MA: MIT Press, 1992.

- [112] K. J. Hunt, "Controller synthesis with genetic algorithms: The evolutionary metaphor in the context of control system optimisation", Report, Department Mechanical Engineering, University of Glasgow, Glasgow, U. K., 1992.
- [113] K. J. Hunt, "Optimal control system synthesis with genetic algorithms", in *Männer and Manderick*, 1992, pp. 381-389.
- [114] K. J. Hunt, "Polynomial LQG and H_{∞} controller synthesis: A genetic algorithm solution", in *Proceedings of the IEEE Conference on Decision and Control*, Tucson, USA, 1992.
- [115] T. M. Murdock, W. E. Schmitendorf, S. Forrest, "Use of genetic algorithm to analyze robust stability problems", in *Proceedings of the 1991 American Control Conference*, Boston, Massachusetts. American Automatic Control Council, Vol. 1, 1991, pp. 886-889.
- [116] K. Kristinsson, G. A. Dumont, "System identification and control using genetic algorithms", in *IEEE Transactions on Systems, Man, and Cybernetic*, Vol. 22, No. 5, 1992, pp. 1033-1046.
- [117] T. C. Fogarty, "Adaptive rule-based optimisation of combustion in multiple burner installations", in G. Gottlob and W. Nejd editors, *Expert Systems in Engineering: Principles and Applications*, 1990, Springer-Verlag, pp. 241-248.
- [118] D. A. Linkens and H. O. Nyongesa, "A real-time genetic algorithm for fuzzy control", in *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, 1992, London, U.K., The Institution of Electrical Engineers. Digest 1992/106, pp. 9/1-9/4.
- [119] C. L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm", in *Belew and Booker*, 1991, pp. 451-457.
- [120] C. L. Karr, "An adaptive system for process control using genetic algorithms", in *International Symposium on Artificial Intelligence in Real-Time Control*, 1992, Delft, The Netherlands, IFAC/IFIP/IMACS, Preprints, pp.585-590.

- [121] Y. Davidor, "A genetic algorithm applied to robot trajectory generation", in *Davis 1991*, Chapter 12, 1991, pp.144-165.
- [122] R. L. Haupt, S. E. Haupt, "*Practical Genetic Algorithms*", John Wiley & Sons, 1998.
- [123] A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca, *Genetic Algorithm TOOLBOX User's Guide*, Department of A.C.S.E. Univ. of Sheffield, Sheffield, UK, 1994.
- [124] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm", in *Grefenstette*, 1987, pp.14-21.
- [125] L. Booker, "Improving search in genetic algorithms", in L. Davis editor, *Genetic Algorithms and Simulated Annealing*, Research Notes in Artificial Intelligence, Chapter 5, Pitman, London, 1987, pp.61-73.
- [126] G. Syswerda, "Uniform crossover in genetic algorithms", in *Schaffer*, 1989, pp.2-9.
- [127] J. R. Koza, *Genetic Programming, On the programming of computers by means of natural selection*, sixth printing, Massachusetts Institute of Technology, 1998.
- [128] J. R. Koza, *Genetic Programming II, Automatic Discovery of Reusable Programs*, second printing, Massachusetts Institute of Technology, 1994.
- [129] Y. Nishikawa, N. Sannomiya, T. Ohta and H. Tanaka, "A method for auto-tuning of PID control parameters", *Automatica*, Vol. 20, Nº3, 1984, pp. 321-32.
- [130] J. Lima., "*Modelo em Simulink para Sintonia Automática de Controladores PID usando Redes Neurais*", Trabalho de síntese relativo às Provas de Aptidão Pedagógica e de Capacidade Científica, Universidade do Algarve, 1995.
- [131] A. B. Azevedo, *Sintonia automática de controladores PID usando redes neuronais com adaptação em tempo real*, Dissertação de Mestrado, Universidade do Algarve, 1997.
- [132] The Math Works Inc., *Optimization Toolbox For Use with MATLAB User's Guide*, Version 2, Third printing Jan. 1999.

- [133] A. Ruano, D. I. Jones, P. J. Fleming, "A new formulation of the learning problem for a neural network controller", *Proc. IEEE Conference on Decision & Control, Brighton*, 1991, pp. 865-866.
- [134] The Math Works Inc., "*Matlab user's guide*", The Math Works Inc., South Natick, MA01760, USA, 1991.
- [135] R. Fletcher, *Practical Methods of Optimisation*, Vol I, John Wiley & Sons.1980
- [136] J. Love, *Documentação para escolha de sistemas*, The University of Sheffield, Sheffield, UK, 2000.
- [137] A. Ruano, "*First Report Controlo de fluxo - Bytronics*", Univ. Algarve, 2000.
- [138] E. Weyer, T. Kavli *The ASMODO algorithm: some new theoretical and experimental results*, SINTEF report STF31 A95024, Oslo, March 1995.
- [139] The Math Works Inc., *SIMULINK Dynamic System Simulation for MATLAB*, Version 4, November 2000.
- [140] A. Adgar, C. Cox, "MATLAB/Simulink Toolbox Facilitates Tuning of Traditional and Predictive Control Strategies", School of Computing & Technology Sunderland, November 2003.

