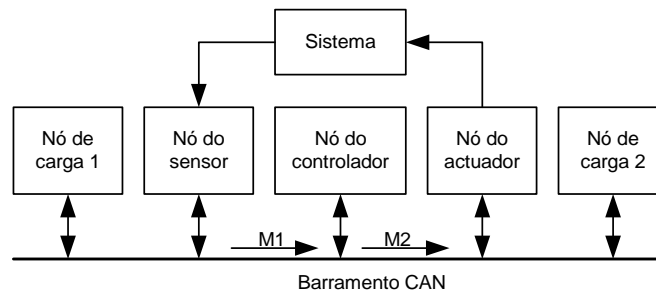




Ana Luisa
Lopes Antunes

Algoritmos de Controlo Distribuído em Sistemas Baseados em Microprocessadores





**Ana Luisa
Lopes Antunes**

**Algoritmos de Controlo Distribuído em Sistemas
Baseados em Microprocessadores**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica, realizada sob a orientação científica do Doutor Alexandre Manuel Moutela Nunes da Mota, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente

Doutor José Fernando Ferreira Mendes
Professor Catedrático da Universidade de Aveiro.

Doutor José António Tenreiro Machado
Professor Coordenador com Agregação do Instituto Superior de Engenharia do Porto.

Doutor José Alberto Gouveia Fonseca
Professor Associado da Universidade de Aveiro.

Doutor Alexandre Manuel Moutela Nunes da Mota
Professor Associado da Universidade de Aveiro (Orientador).

Doutor Mário João Barata Calha
Professor Auxiliar da Faculdade de Ciências da Universidade de Lisboa.

Doutora Teresa Maria de Gouveia Torres Feio Mendonça
Professora Auxiliar da Faculdade de Ciências da Universidade do Porto.

Doutor Paulo Bacelar Reis Pedreiras
Professor Auxiliar da Universidade de Aveiro.

dedicatória

Aos meus Pais pelo grãozinho de loucura.

agradecimentos

A realização deste trabalho só foi possível com a colaboração e com o apoio de um conjunto alargado de pessoas às quais serei sempre grata.

Agradeço ao meu orientador, Professor Alexandre Mota, pela criação das condições para o desenvolvimento deste trabalho, por todo o apoio científico e técnico, pela grande disponibilidade, pelos ensinamentos e pelo encorajamento, que tornaram possível este (longo) percurso.

Agradeço ao Laboratório de Sistemas Electrónicos, no âmbito do qual desenvolvi este trabalho, pelo acolhimento; ao Luís Almeida e ao Paulo Pedreiras em especial, pelas sugestões e pelas conversas que resultaram na realização de trabalho conjunto, mas também aos Professores José Alberto Fonseca, Pedro Fonseca e Ernesto Martins, que de modos diversos contribuíram para este trabalho, assim como aos colegas Valter Silva, Paulo Bartolomeu e Ricardo Marau pelo apoio e pela ajuda.

Agradeço ao Fernando e ao Zé António pela disponibilidade e pelo trabalho conjunto, e ao Fernando também pelas produtivas discussões.

Agradeço à ESTSetúbal pela dispensa de serviço docente no período de Março a Julho de 2008, através do Programa de Formação Avançada de Docentes, que permitiu a conclusão desta tese, e a todos os colegas que me apoiaram e incentivaram, em particular aos do DEE.

Agradeço aos meus amigos por partilharem comigo ao longo desta caminhada os melhores e os piores momentos, em especial ao Fernando, ao Victor, à Susana, ao Nandinho, à Gisela, ao Alf, ao Mário, ao Teixeira e à Ana Carla.

Agradeço a toda a minha família pelo interesse com que seguiu este trabalho e pelo encorajamento.

Agradeço aos meus Pais e à minha irmã pelo apoio incondicional.

palavras-chave

Sistemas distribuídos de tempo-real, controlo distribuído, sistemas embutidos

resumo

O trabalho de investigação apresentado nesta tese trata dos problemas associados à distribuição de controladores digitais em sistemas baseados em microprocessadores. A utilização deste tipo de sistemas tornou-se comum ao longo da última década em domínios como a automação industrial, o controlo de processos, a robótica, a domótica, assim como na indústria automóvel e aeronáutica. A distribuição de uma malha de controlo faz-se recorrendo a nós fisicamente distintos para a implementação do sensor, do controlador e do actuador, ligados entre si através de uma rede de comunicação. Na maior parte dos sistemas embutidos distribuídos actuais a rede de comunicação e os processadores locais de cada nó são utilizados por vários tipos de aplicações tanto de tempo-real como de outro tipo. Os sistemas de controlo distribuído implementados deste modo apresentam características próprias que diferem das que caracterizam os sistemas de controlo centralizado. Uma dessas características é a existência de um atraso variável entre o momento da amostragem e o momento da actuação. Este tipo de atrasos provoca a degradação do desempenho de controlo, podendo mesmo em alguns casos conduzir à perda de estabilidade da malha de controlo. Neste trabalho propõem-se técnicas de controlo lineares e não-lineares para melhorar o desempenho de controlo neste tipo de sistemas embutidos. As técnicas propostas são validadas através da simulação de vários sistemas utilizando um simulador para sistemas de tempo-real baseado em MATLAB/Simulink.

keywords

Real-time distributed systems, networked control systems, embedded systems

abstract

This thesis proposes solutions to the problems that arise with the distribution of control systems in systems based in microprocessors. The use of this kind of systems grew over the past decade in areas like industrial automation, process control, robotics, building automation and the automobile and aeronautical industries. The distribution of a control loop consists in the placement of the sensor, the controller and the actuator in different nodes, connected through a communication network. In modern embedded distributed systems both the communication network and the local processors are used by several real-time and non real-time applications. This kind of implementations of the distributed controllers presents characteristics that are different from the ones of centralized controllers. They usually present variable delay between the sampling and the actuation instants. These delays degrade the performance of the controller and can even destabilize the control loops.

In this thesis linear and non-linear control techniques are proposed to improve the control performance of such embedded control systems. The proposed techniques are validated by the simulation of different plants using a real-time systems simulator based on MATLAB/Simulink.

Conteúdo

1	Introdução	1
1.1	Perspectiva histórica	2
1.2	Motivação	3
1.3	Pressupostos	4
1.4	Principais contributos	4
1.5	Estrutura da tese	5
1.6	Publicações realizadas no âmbito desta tese	6
1.7	Conclusão	8
2	Sistemas de controlo distribuído em tempo-real	9
2.1	Distribuição de controladores	10
2.1.1	Incerteza quanto aos instantes de amostragem e de actuação	11
2.2	Sistemas de tempo-real	13
2.2.1	Conceitos básicos	13
2.2.2	Técnicas de escalonamento	14
2.3	Sistemas de tempo-real distribuídos	19
2.3.1	Comunicação em tempo-real	19
2.3.2	Controlo de acesso ao meio	22
2.3.3	Escalação das mensagens	24
2.3.4	Barramentos de campo	25
2.4	Técnicas de controlo utilizadas em sistemas de controlo distribuído	37
2.4.1	Problemas associados à operação de sistemas de controlo distribuído	38
2.4.2	Métodos para o projecto do controlador	39
2.4.3	Métodos integrados para o controlo e o escalonamento do sistema	40
2.5	Abordagens propostas nesta tese	42
2.6	Conclusão	42
3	Modelação e controlo adaptativo distribuído de sistemas lineares	43
3.1	Implicações da utilização de controladores digitais	44
3.1.1	Amostragem do sistema	45
3.1.2	Passagem do plano s para o plano z	47
3.1.3	Utilização de modelos discretos do sistema	49
3.2	Amostragem de um sistema contínuo	50
3.2.1	Representação do tempo morto	50

3.2.2	Amostragem de sistemas sem tempo morto	52
3.2.3	Amostragem de sistemas com tempo morto	53
3.2.4	Seleccção do período de amostragem	55
3.3	Modelos discretos do sistema	57
3.3.1	Classes de modelos mais utilizadas	57
3.3.2	Outras classes de modelos	59
3.3.3	Modelos discretos sem tempo morto fraccionário	61
3.3.4	Modelos discretos com tempo morto fraccionário	62
3.4	Identificação dos parâmetros do modelo do sistema	63
3.4.1	Método dos mínimos quadrados	64
3.4.2	Implementação recursiva do método dos mínimos quadrados	66
3.4.3	Identificação de parâmetros variáveis no tempo	68
3.4.4	Identificação de sistemas em malha fechada	73
3.4.5	Implementação do processo de identificação	74
3.5	Controlo adaptativo “clássico”	76
3.5.1	Tipos de controladores adaptativos	77
3.5.2	Exemplos de técnicas de controlo adaptativas	80
3.6	Controladores adaptativos distribuídos	91
3.6.1	Modelação com tempo morto fraccionário	92
3.6.2	Adaptação dinâmica do período de amostragem	93
3.6.3	Implementação do controlador adaptativo com período de amostragem variável	95
3.7	Controlo adaptativo distribuído por posicionamento de pólos	98
3.8	Conclusão	103
4	Técnicas de controlo lineares para sistemas distribuídos	105
4.1	Condições de teste	105
4.1.1	Arquitectura base do sistema de teste	105
4.1.2	Padrões de carga do barramento	107
4.1.3	Sistemas a controlar	111
4.1.4	Controladores	111
4.1.5	Critério de avaliação do desempenho	113
4.1.6	O simulador TrueTime	113
4.2	Controladores adaptativos por posicionamento de pólos	117
4.2.1	Sistema S1	117
4.2.2	Sistema S2	123
4.2.3	Sistema S3	128
4.2.4	Sistema S4	134
4.3	Controladores adaptativos distribuídos por posicionamento de pólos	139
4.3.1	Sistema S1	140
4.3.2	Sistema S2	146
4.3.3	Sistema S3	152
4.3.4	Sistema S4	158
4.4	Técnica de adaptação dinâmica da taxa de amostragem	165

4.4.1	Características da implementação com FTT-CAN	166
4.4.2	Abordagem estática	168
4.4.3	Abordagem dinâmica	175
4.4.4	Abordagem híbrida	182
4.4.5	Optimização local de malhas de controlo (resultados preliminares)	191
4.5	Conclusão	195
5	Técnicas de controlo não-lineares para sistemas distribuídos	197
5.1	O princípio do compensador de atraso	198
5.1.1	Compensador difuso	199
5.1.2	Compensador neuronal	202
5.2	Implementações do compensador de atraso	204
5.2.1	Condições de teste	204
5.2.2	Implementação difusa	210
5.2.3	Implementação neuronal	215
5.3	Comparação de desempenho entre as técnicas lineares e não-lineares . .	219
5.4	Conclusão	220
6	Conclusões e trabalho futuro	221
6.1	Conclusões	221
6.2	Trabalho futuro	223
	Anexo A - Dedução dos parâmetros do regulador	237
	Anexo B - O simulador TrueTime	253
	Anexo C - Lista de Abreviaturas	261

Lista de Figuras

1.1	Diagrama de blocos de um sistema de controlo distribuído.	2
2.1	Diagrama de blocos de um sistema de controlo distribuído.	9
2.2	Diagrama de blocos detalhado de um sistema de controlo distribuído. .	10
2.3	Representação temporal do tempo entre a amostragem e a actuação. . .	12
2.4	Parâmetros típicos para a caracterização de uma tarefa.	13
2.5	Classificação dos algoritmos de escalonamento de tempo-real.	15
2.6	Camadas do modelo de referência OSI.	20
2.7	Camadas do modelo EPA.	21
2.8	Topologia em barramento.	21
2.9	Divisão do tempo nominal de <i>bit</i> em segmentos.	26
2.10	Exemplo do processo de arbitragem CSMA/NBA.	28
2.11	Formato da trama de dados	29
2.12	Campos de arbitragem e controlo, formato 2.0A – <i>standard</i>	29
2.13	Campos de arbitragem e controlo, formato 2.0A – estendido.	30
2.14	Diagrama das transições de estatuto num nó CAN.	31
2.15	Formato de uma trama WorldFIP.	32
2.16	Formato do campo CAD das tramas ID_DAT e RP_DAT.	33
2.17	Formato das mensagens PROFIBUS.	35
2.18	Estrutura de um <i>Elementary Cycle</i> em FTT-CAN.	36
3.1	Diagrama de blocos de um controlador realimentado genérico.	44
3.2	Diagrama de blocos de um controlador digital genérico.	45
3.3	Detalhe do sistema de controlo digital.	46
3.4	Mapeamento entre os planos s e z utilizando a transformação $z = e^{sh}$. .	47
3.5	Sistema com perturbação.	50
3.6	Representação gráfica do erro de estimativa.	63
3.7	Sistema em malha fechada.	73
3.8	Diagrama de blocos de um sistema em malha fechada com introdução de <i>dither</i>	74
3.9	Diagrama de blocos de um controlador adaptativo genérico.	76
3.10	Diagrama de blocos de um controlador por escalonamento de ganho. . .	77
3.11	Diagrama de blocos de um controlador adaptativo por modelo de re- ferência.	78
3.12	Diagrama de blocos de um regulador auto-ajustável.	79

3.13	Diagrama de blocos de um regulador óptimo.	79
3.14	Diagrama de blocos de um controlador adaptativo inverso básico.	80
3.15	Diagrama de blocos de um sistema controlado por vários reguladores auto-ajustáveis dependendo de um critério de selecção.	81
3.16	Diagrama de blocos do controlador GPC.	82
3.17	Diagrama de blocos detalhado da função de controlo de um controlador PP.	86
3.18	Implementação estática.	96
3.19	Implementação dinâmica.	97
3.20	Implementação híbrida.	98
3.21	Comutação entre valores de h , abordagem híbrida.	99
4.1	Diagrama de blocos do sistema de teste sem implementação do protocolo FTT-CAN (A1).	106
4.2	Diagrama de blocos do sistema de teste com implementação do protocolo FTT-CAN (A2).	106
4.3	Sequência temporal da troca de mensagens da malha de controlo.	108
4.4	Atraso da amostragem para a actuação para o padrão de carga P1.	109
4.5	Atraso da amostragem para a actuação para o padrão de carga P2.	110
4.6	Atraso da amostragem para a actuação para o padrão de carga P3.	111
4.7	Biblioteca TrueTime para MATLAB/Simulink.	114
4.8	Diagrama da implementação em TrueTime da arquitectura A1.	116
4.9	Diagrama da implementação em TrueTime da arquitectura A2.	116
4.10	Sinais para o teste de referência (sem TMF), sistema S1.	118
4.11	Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S1.	118
4.12	Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S1.	119
4.13	Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S1.	119
4.14	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S1.	121
4.15	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S1.	121
4.16	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S1.	122
4.17	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S1.	122
4.18	Sinais para o teste de referência (sem TMF), sistema S2.	123
4.19	Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S2.	124
4.20	Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S2.	124
4.21	Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S2.	125
4.22	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S2.	126
4.23	Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P1 (sem TMF), sistema S2.	126
4.24	Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P2 (sem TMF), sistema S2.	127

4.25	Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P3 (sem TMF), sistema S2.	127
4.26	Sinais para o teste de referência (sem TMF), sistema S3.	129
4.27	Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S3. .	129
4.28	Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S3. .	130
4.29	Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S3. .	130
4.30	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S3.	131
4.31	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S3.	132
4.32	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S3.	132
4.33	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S3.	133
4.34	Sinais para o teste de referência (sem TMF), sistema S4.	134
4.35	Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S4. .	135
4.36	Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S4. .	135
4.37	Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S4. .	136
4.38	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S4.	137
4.39	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S4.	137
4.40	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S4.	138
4.41	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S4.	138
4.42	Sinais para o teste de referência (com TMF), sistema S1.	140
4.43	Sinais para o teste com o padrão de carga P1 (com TMF), sistema S1. .	141
4.44	Sinais para o teste com o padrão de carga P2 (com TMF), sistema S1. .	141
4.45	Sinais para o teste com o padrão de carga P3 (com TMF), sistema S1. .	142
4.46	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S1.	144
4.47	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S1.	144
4.48	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S1.	145
4.49	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S1.	145
4.50	Sinais para o teste de referência (com TMF), sistema S2.	147
4.51	Sinais para o teste com o padrão de carga P1 (com TMF), sistema S2. .	147
4.52	Sinais para o teste com o padrão de carga P2 (com TMF), sistema S2. .	148
4.53	Sinais para o teste com o padrão de carga P3 (com TMF), sistema S2. .	148
4.54	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S2.	150

4.55	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S2.	150
4.56	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S2.	151
4.57	Evolução dos parâmetros do modelo discreto do sistema, para o teste com o padrão de carga P3 (com TMF), sistema S2.	151
4.58	Sinais para o teste de referência (com TMF), sistema S3.	153
4.59	Sinais para o teste com o padrão de carga P1 (com TMF), sistema S3.	153
4.60	Sinais para o teste com o padrão de carga P2 (com TMF), sistema S3.	154
4.61	Sinais para o teste com o padrão de carga P3 (com TMF), sistema S3.	154
4.62	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S3.	156
4.63	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S3.	156
4.64	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S3.	157
4.65	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S3.	157
4.66	Sinais para o teste de referência (com TMF), sistema S4.	159
4.67	Sinais para o teste com o padrão de carga P1 (com TMF), sistema S4.	159
4.68	Sinais para o teste com o padrão de carga P2 (com TMF), sistema S4.	160
4.69	Sinais para o teste com o padrão de carga P3 (com TMF), sistema S4.	160
4.70	Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S4.	162
4.71	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S4.	162
4.72	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S4.	163
4.73	Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S4.	163
4.74	Sinais para o ensaio E1, abordagem estática, sistema S1.	169
4.75	Sinais para o ensaio E2, abordagem estática, sistema S1.	170
4.76	Sinais para o ensaio com $h = 0.12$ s utilizando os parâmetros identificados para $h = 0.28$ s, abordagem estática, sistema S1.	172
4.77	Sinais para o ensaio E1, abordagem estática, sistema S2.	173
4.78	Sinais para o ensaio E2, abordagem estática, sistema S2.	174
4.79	Sinais para o ensaio E1, abordagem dinâmica, sistema S1.	176
4.80	Sinais para o ensaio E2, abordagem dinâmica, sistema S1.	177
4.81	Evolução dos parâmetros identificados durante o ensaio E1, abordagem dinâmica, sistema S1.	177
4.82	Evolução dos parâmetros identificados durante o ensaio E2, abordagem dinâmica, sistema S1.	178
4.83	Sinais para o ensaio E1, abordagem dinâmica, sistema S2.	179
4.84	Sinais para o ensaio E2, abordagem dinâmica, sistema S2.	180

4.85	Evolução dos parâmetros identificados durante o ensaio E1, abordagem dinâmica, sistema S2.	180
4.86	Evolução dos parâmetros identificados durante o ensaio E2, abordagem dinâmica, sistema S2.	181
4.87	Sinais para o ensaio E1, abordagem híbrida, sistema S1.	183
4.88	Sinais para o ensaio E2, abordagem híbrida, sistema S1.	183
4.89	Evolução dos parâmetros identificados durante o ensaio E1, abordagem híbrida, sistema S1.	184
4.90	Evolução dos parâmetros identificados durante o ensaio E2, abordagem híbrida, sistema S1.	184
4.91	Sinais para o ensaio E1, abordagem híbrida, sistema S2.	185
4.92	Sinais para o ensaio E2, abordagem híbrida, sistema S2.	186
4.93	Evolução dos parâmetros identificados durante o ensaio E1, abordagem híbrida, sistema S2.	186
4.94	Evolução dos parâmetros identificados durante o ensaio E2, abordagem híbrida, sistema S2.	187
4.95	Sinais para o teste em que $h < 0.28$ s a maior parte do tempo, sistema S1.	188
4.96	Sinais para o ensaio C1, abordagem híbrida, sistema S1.	189
4.97	Evolução dos parâmetros do modelo discreto do sistema durante o ensaio C1, abordagem híbrida, sistema S1.	189
4.98	Sinais para o ensaio C2, abordagem híbrida, sistema S1.	190
4.99	Evolução dos parâmetros do modelo discreto do sistema para o teste C2, abordagem híbrida, sistema S1.	191
4.100	Atraso entre a amostragem e a actuação, padrão utilizado nos ensaios preliminares.	192
4.101	Sinais para o ensaio L1, sistema S1.	193
4.102	Sinais para o ensaio L1, sistema S2.	193
4.103	Sinais para o ensaio L2, sistema S1.	194
4.104	Sinais para o ensaio L2, sistema S2.	194
5.1	Estrutura utilizada para o controlo aditivo.	198
5.2	Diagrama de blocos do princípio do compensador de atraso.	199
5.3	Diagrama de blocos da implementação do compensador de atraso utilizando lógica difusa.	200
5.4	Função de pertinência da entrada tsa.	201
5.5	Função de pertinência da entrada difupp.	201
5.6	Função de pertinência da saída ucomp.	201
5.7	Superfície resultante do modelo difuso implementado para o compensador difuso.	202
5.8	Diagrama de blocos do método utilizado para calcular a saída do modelo neuronal.	203
5.9	Diagrama de blocos da implementação do compensador de atraso utilizando redes neuronais.	204
5.10	Diagrama de blocos do sistema utilizado para o teste do compensador de atraso.	205

5.11	Diagrama da implementação em TrueTime do compensador de atraso. . .	206
5.12	Sinais para o teste do compensador de atraso (sem o compensador), teste de referência.	207
5.13	Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P1.	208
5.14	Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P2.	208
5.15	Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P3.	209
5.16	Sinais para o teste com o compensador difuso, ensaio de referência. . .	210
5.17	Contribuição do controlador PP e do compensador difuso durante o ensaio de referência.	210
5.18	Sinais para o teste com o compensador difuso, para o padrão de carga P1.	211
5.19	Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P1.	211
5.20	Sinais para o teste com o compensador difuso, para o padrão de carga P2.	212
5.21	Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P2.	212
5.22	Sinais para o teste com o compensador difuso, para o padrão de carga P3.	213
5.23	Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P3.	213
5.24	Sinais para o teste com o compensador neuronal, ensaio de referência. .	215
5.25	Contribuição do controlador PP e do compensador neuronal durante o ensaio de referência.	216
5.26	Sinais para o teste com o compensador neuronal, para o padrão de carga P1.	216
5.27	Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P1.	217
5.28	Sinais para o teste com o compensador neuronal, para o padrão de carga P2.	217
5.29	Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P2.	218
5.30	Sinais para o teste com o compensador neuronal, para o padrão de carga P3.	218
5.31	Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P3.	219
1	Biblioteca de blocos TrueTime.	253

Lista de Tabelas

3.1	Polinómios $Bd(z)$	48
3.2	Modelos de 1ª e de 2ª ordem de sistemas sem tempo morto fraccionário.	75
3.3	Modelos de 1ª e de 2ª ordem de sistemas com tempo morto fraccionário.	75
4.1	Características das mensagens da malha de controlo.	107
4.2	Características do conjunto de mensagens utilizadas para gerar os padrões de carga P1 e P2.	109
4.3	Características do conjunto de mensagens utilizadas para gerar o padrão de carga P3.	110
4.4	Funções de transferência dos sistemas utilizados nos ensaios.	112
4.5	Características do sistema em malha fechada, sistema S1.	112
4.6	Características do sistema em malha fechada, sistemas S2, S3 e S4.	112
4.7	Valor de ISE e de ITSE para os testes (sem TMF), sistema S1.	120
4.8	Parâmetros identificados no final dos testes (sem TMF), sistema S1.	120
4.9	Localização do pólo identificado (sem TMF), sistema S1.	122
4.10	Valor de ISE e de ITSE para os testes (sem TMF), sistema S2.	125
4.11	Parâmetros identificados no final dos testes (sem TMF), sistema S2.	128
4.12	Localização dos pólos identificados (sem TMF), sistema S2.	128
4.13	Valor de ISE e de ITSE para os testes (sem TMF), sistema S3.	131
4.14	Parâmetros identificados no final dos testes (sem TMF), sistema S3.	133
4.15	Localização dos pólos identificados (sem TMF), sistema S3.	133
4.16	Valor de ISE e de ITSE para os testes (sem TMF), sistema S4.	136
4.17	Parâmetros identificados no final dos testes (sem TMF), sistema S4.	139
4.18	Localização dos pólos identificados (sem TMF), sistema S4.	139
4.19	Valor de ISE e de ITSE para os testes (com TMF), sistema S1.	142
4.20	Melhoria para o sistema S1.	143
4.21	Parâmetros identificados no final dos testes (com TMF), sistema S1.	146
4.22	Localização dos pólos identificados (com TMF), sistema S1.	146
4.23	Valor de ISE e de ITSE para os testes (com TMF), sistema S2.	149
4.24	Melhoria para o sistema S2.	149
4.25	Parâmetros identificados no final dos testes (com TMF), sistema S2.	152
4.26	Localização dos pólos identificados (com TMF), sistema S2.	152
4.27	Valor de ISE e de ITSE para os testes (com TMF), sistema S3.	155
4.28	Melhoria para o sistema S3.	155
4.29	Parâmetros identificados no final dos testes (com TMF), sistema S3.	158

4.30	Localização dos pólos identificados (com TMF), sistema S3.	158
4.31	Valor de ISE e de ITSE para os testes (com TMF), sistema S4.	161
4.32	Melhoria para o sistema S4.	161
4.33	Parâmetros identificados no final dos testes (com TMF), sistema S4. . .	164
4.34	Localização dos pólos identificados (com TMF), sistema S4.	164
4.35	Características do conjunto de mensagens síncronas de base (período, <i>offset</i> e <i>deadline</i> em EC).	167
4.36	Mensagens admitidas em <i>runtime</i> (período, <i>offset</i> e <i>deadline</i> em EC). .	167
4.37	Valor dos parâmetros do modelo discreto, abordagem estática, sistema S1.	169
4.38	Valor de ISE e de ITSE para os ensaios da abordagem estática, sistema S1.	170
4.39	Valor de ISE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.	171
4.40	Valor de ITSE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.	171
4.41	Valor dos parâmetros do modelo discreto, abordagem estática, sistema S2.	173
4.42	Valor de ISE e de ITSE para os ensaios da abordagem estática, sistema S2.	174
4.43	Valor de ISE para os ensaios com h diferente do valor de projecto do controlador, sistema S2.	175
4.44	Valor de ITSE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.	175
4.45	Valor de ISE e de ITSE para os ensaios da abordagem dinâmica, sistema S1.	178
4.46	Valor de ISE e de ITSE para os ensaios da abordagem dinâmica, sistema S2.	181
4.47	Valor de ISE e de ITSE para os ensaios da abordagem híbrida, sistema S1.	185
4.48	Valor de ISE e de ITSE para os ensaios da abordagem híbrida, sistema S2.	187
4.49	Valor de ISE e de ITSE para os ensaios preliminares, sistema S1.	192
4.50	Valor de ISE e de ITSE para os ensaios preliminares, sistema S2.	192
5.1	Características da mensagem da malha de controlo.	205
5.2	Valor dos parâmetros da função de controlo para o controlador PP. . .	206
5.3	Valor de ISE e de ITSE para os testes do compensador de atraso (sem compensador).	209
5.4	Valor de ISE e de ITSE para os testes com o compensador difuso. . . .	214
5.5	Melhoria obtida com o compensador difuso.	214
5.6	Valor de ISE e ITSE para os testes com o compensador neuronal. . . .	215
5.7	Melhoria obtida para o compensador neuronal.	215
1	Comandos utilizados para criar e inicializar objectos TrueTime.	255
2	Comandos utilizados para atribuir e determinar os atributos das tarefas.	256
3	Primitivas de tempo-real.	257

Capítulo 1

Introdução

Este trabalho trata do estudo de sistemas de controlo distribuído, em particular dos que são implementados com recurso a microprocessadores e microcontroladores. Num sistema de controlo distribuído a comunicação entre sensor(es), actuador(es) e controlador(es) é feita recorrendo a uma infraestrutura de comunicação comum.

Este tipo de sistemas é utilizado actualmente numa grande variedade de domínios. Exemplos desses domínios são a automação industrial, a domótica, a robótica, a indústria aeronáutica e a indústria automóvel. Um exemplo paradigmático da aplicação de sistemas de controlo distribuído é o automóvel. Um modelo actual pode conter mais de 50 unidades de controlo, designadas neste domínio por ECUs (*Electronic Control Units*) com vários níveis de complexidade. Muitas dessas unidades implementam malhas de realimentação com o objectivo de controlar o motor, a tracção, o processo de travagem, a estabilidade do automóvel, a velocidade e a temperatura [1].

Devido ao facto dos sistemas de controlo distribuído serem de utilização comum em dispositivos que são consumidos em larga escala, o seu desenvolvimento está sujeito a vários tipos de pressões, relacionadas não só com o seu desempenho mas também com o seu custo.

A área de estudo dos sistemas de controlo distribuído é uma área multidisciplinar, em que se cruzam os domínios do controlo, dos sistemas de tempo-real e das redes de comunicação.

A pergunta de investigação que serviu de base a este trabalho é a seguinte:

Será possível utilizar técnicas baseadas em algoritmos de controlo convencionais em sistemas de controlo distribuído?

Na procura de resposta para esta pergunta foram seguidas duas abordagens distintas: por um lado procuraram-se soluções de controlo que permitissem o projecto de raiz de um controlador distribuído, tendo em conta as características específicas do ambiente em que este vai operar; por outro procuraram-se soluções que permitissem compensar os efeitos da distribuição do sistema de controlo, quando se utilizam controladores convencionais como base desses sistemas.

A primeira perspectiva originou o desenvolvimento de controladores distribuídos por posicionamento de pólos e a segunda o princípio do compensador de atraso.

Neste capítulo é apresentado um breve resumo histórico sobre a evolução desta área, a motivação para o desenvolvimento deste trabalho e são descritas as principais contribuições. Por último apresenta-se a estrutura da tese e as publicações relacionadas com o trabalho desenvolvido.

1.1 Perspectiva histórica

Um controlador distribuído é composto por vários nós de processamento, como o nó do sensor, o nó do controlador e o nó do actuador ligados entre si, como se pode ver na figura 1.1, através de uma rede de comunicação.

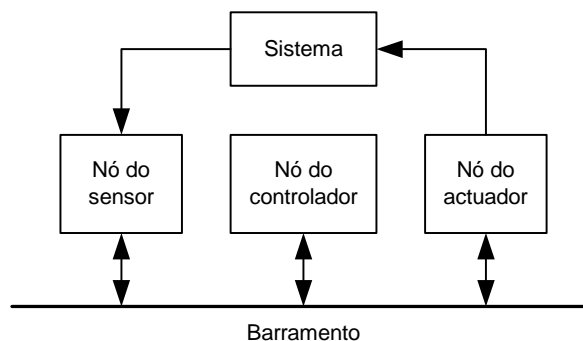


Figura 1.1: Diagrama de blocos de um sistema de controlo distribuído.

A rede de comunicação, tradicionalmente um barramento de campo, é utilizada para trocar informação entre os nós, ficando a malha de controlo fechada através dela.

Para implementar um controlador distribuído são necessários nós de processamento, uma rede de comunicação e o *hardware* necessário para fazer a interface entre os nós de processamento e a rede de comunicação.

O controlo distribuído surge na sequência de uma série de evoluções tecnológicas, que permitiram o aparecimento e a massificação do *hardware* adequado à sua implementação, como os circuitos integrados, particularmente os microprocessadores e os microcontroladores, assim como os protocolos e tecnologias associados aos barramentos de campo. O surgimento do controlo distribuído está, por isso, estreitamente relacionado com a evolução dos computadores e do controlo com recurso a computadores (controlo digital).

O controlo digital começou a ser aplicado nos anos 50 do séc. XX em aplicações militares, como o controlo de mísseis e a detecção de aeronaves. Nessa altura os computadores eram implementados com base em válvulas, sendo dispendiosos e pouco práticos, por ocuparem muito espaço e serem difíceis de operar.

As aplicações ao controlo de processos surgiram nos anos 60, ao mesmo tempo que os computadores passaram a ser implementados com transístores, e, acompanhando o surgimento dos microprocessadores e dos microcontroladores na década de 70 e o aumento do nível de integração, tiveram um grande desenvolvimento ao longo dos anos

80. O surgimento dos microprocessadores e o aumento do nível de integração permitiram a diminuição do tamanho das implementações e do seu custo. A democratização do controlo digital ocorreu nos anos 90, quando o custo dos microprocessadores diminuiu drasticamente, permitindo a sua aplicação em áreas de consumo massivo como a indústria automóvel e as telecomunicações.

Os controladores auto-ajustáveis, que também são alvo de estudo neste trabalho, foram desenvolvidos na década de 70, tendo as primeiras aplicações comerciais surgido na década de 80.

A primeira implementação de um controlador distribuído surgiu em 1975, na área de controlo de processos, tendo a sua aplicação nessa área crescido ao longo dos anos 80, até se popularizar nos anos 90 com a expansão a outras áreas como a indústria automóvel e aeronáutica, a gestão de edifícios, a automação industrial, a robótica, a gestão ferroviária e as telecomunicações. Este desenvolvimento coincidiu também com o aparecimento e desenvolvimento de uma série de barramentos de campo, ao longo dos anos 80, e com a consolidação no mercado de alguns deles, como por exemplo o barramento CAN [2], através da disponibilização de *hardware* para interface com microprocessadores a preços muito competitivos.

As primeiras aplicações de controlo adaptativo a sistemas distribuídos surgiram nos anos 90.

Com a utilização em larga escala de controladores distribuídos surge também uma enorme pressão económica no sentido de limitar o custo dos produtos finais. Essa pressão levou, mais recentemente, a uma tendência para a integração de várias aplicações, sobre a mesma rede de comunicação, e à necessidade de lidar com os problemas associados a esse tipo de sistemas.

No final dos anos 80 e início dos anos 90 surgiram os primeiros artigos sobre os problemas específicos das implementações de controladores distribuídos. Ao longo da década de 90 iniciou-se o estudo desta área e das suas especificidades tendo sido publicados os primeiros artigos propondo soluções. Os primeiros trabalhos de vulto, como teses de doutoramento, tratando dos problemas temporais associados à implementação de controladores surgiram no início da década actual (por exemplo [3], [4] e [5]).

1.2 Motivação

Na última década tem-se assistido a uma tendência generalizada para a integração do controlo, da computação e da comunicação tanto em sistemas industriais como comerciais. Devido à evolução tecnológica e à diminuição dos custos de implementação, passou a ser possível a utilização massiva de sistemas de controlo distribuído embutidos em produtos de consumo em larga escala. Por razões económicas a pressão no sentido da diminuição do custo destes sistemas é enorme, tendo originado nos últimos anos uma tendência para uma cada vez maior distribuição dos sistemas, tendência essa que se vai manter.

A distribuição do controlo apresenta vantagens: diminuição dos custos com cablagens, melhor utilização dos recursos, composição dos sistemas por módulos, facilidade de reconfiguração, manutenção e diagnóstico. No entanto, ela apresenta também de-

safios relacionados com as características próprias deste tipo de implementação, como por exemplo o aparecimento de atrasos introduzidos na malha pelo processo de comunicação, o que degrada o desempenho do sistema.

No início da utilização dos sistemas de controlo distribuído as restrições impostas às implementações não eram tão severas como são hoje, permitindo assim a utilização dos modelos e técnicas desenvolvidos para os sistemas centralizados, sem que o desempenho fosse muito afectado. Nos últimos anos a pressão para uma melhor utilização dos recursos disponíveis, devido a razões económicas, tem levado à necessidade de desenvolver técnicas de controlo adequadas às novas realidades de implementação, que incluam de modo explícito as características específicas das aplicações distribuídas, de modo a permitir a obtenção do melhor desempenho possível.

1.3 Pressupostos

O trabalho desenvolvido ao longo desta tese assenta em alguns pressupostos que se passam a enunciar:

1. Apesar de, como já foi referido, a área de estudo tratada nesta tese ser multidisciplinar, o trabalho desenvolvido centra-se nas técnicas de controlo. Os objectivos são os de propor técnicas de controlo que permitam melhorar o desempenho de uma malha de controlo distribuído a nível local, mas que sejam também flexíveis de modo a poderem adaptar-se a restrições impostas pelo sistema de tempo-real, do qual fazem parte, sem comprometer o desempenho global de modo significativo.
2. Como ficará claro na exposição efectuada no capítulo 2, num sistema de controlo distribuído pode existir incerteza quanto ao instante de amostragem e quanto ao instante de actuação. Neste trabalho considera-se que não há incerteza quanto ao instante de amostragem. Admite-se a incerteza no momento de actuação e a consequente existência de um tempo variável entre a amostragem e a actuação.
3. Considera-se ainda que o atraso introduzido na malha de controlo é menor do que o período de amostragem da malha, não se considerando também as situações de perda de amostras.
4. Nas situações em que se permite que o período de amostragem da malha de controlo possa ser alterado, ao longo da operação da mesma, considera-se que se garante o tempo mínimo necessário à estabilização da malha de controlo antes de uma nova alteração desse valor.

1.4 Principais contributos

Esta tese apresenta diversos contributos no domínio do controlo distribuído sendo estes pormenorizadamente descritos ao longo dos capítulos 3 e 5. Os principais contributos são:

1. A utilização sistemática da modelação do tempo morto fraccionário nos modelos discretos utilizados para caracterizar o sistema, sempre que a implementação implique a existência de tempos variáveis entre o instante de amostragem e o instante de actuação. A utilização deste tipo de modelação permite a implementação de controladores que têm em conta o atraso introduzido sem ser necessário efectuar medições desse atraso. Este método de modelação é apresentado no capítulo 3.
2. A técnica de comutação de controladores auto-ajustáveis, que propõe a comutação entre vários controladores auto-ajustáveis durante a operação da malha, de acordo com critérios que podem ser internos à malha, como por exemplo a melhoria do desempenho local, ou externos à malha, como a optimização da utilização do barramento em situações de sobrecarga momentânea. Esta técnica é descrita no capítulo 3.
3. A técnica de adaptação dinâmica do período de amostragem, implementada através da técnica de comutação referida anteriormente, que consiste na alteração do período de amostragem de uma dada malha de controlo seja em benefício do seu próprio desempenho local, seja em benefício da optimização global do sistema de tempo-real em que a malha se insere. Esta técnica é apresentada no capítulo 3.
4. A técnica do compensador de atraso, que consiste na inclusão, numa malha já existente em que o controlador foi projectado sem ter em conta o atraso induzido, de um bloco de compensação com o objectivo de minimizar a degradação de controlo provocada pelo atraso existente na malha. Esta técnica necessita do conhecimento do valor do atraso entre a amostragem e a actuação presente no sistema a cada iteração, sendo apresentada no capítulo 5.

1.5 Estrutura da tese

Esta tese está organizada em seis capítulos. No capítulo 2 é feita a análise das características específicas de um sistema de controlo distribuído sendo estas características comparadas com as de um sistema de controlo centralizado tradicional. Porque a área de estudo dos sistemas de controlo distribuído é multidisciplinar descrevem-se os conceitos de base de sistemas de tempo-real e de sistemas de tempo-real distribuídos. Por fim é apresentado o resumo do trabalho já existente sobre técnicas de controlo para sistemas de controlo distribuído.

No capítulo 3 são descritas as técnicas de modelação e controlo adaptativo “clássicas” e são propostas técnicas para modelação e controlo adaptativo distribuído.

No capítulo 4 apresentam-se os resultados de simulação obtidos por via da aplicação das técnicas de controlo adaptativo distribuído propostas.

No capítulo 5 é apresentada a técnica do compensador de atraso e são propostas implementações para o compensador que utilizam técnicas de controlo não-lineares

como lógica difusa e redes neuronais. Os resultados obtidos através da simulação das técnicas propostas são também apresentados.

Finalmente no capítulo 6 são apresentadas as conclusões e as perspectivas de trabalho futuro.

Na parte final deste documento são ainda apresentados dois anexos onde são descritos alguns detalhes relativos à dedução das funções de controlo dos controladores por posicionamento de pólos (anexo A) e do simulador TrueTime (anexo B).

1.6 Publicações realizadas no âmbito desta tese

O trabalho realizado no âmbito desta tese permitiu a publicação de vários artigos em conferências, em revistas e em livro. As publicações são aqui apresentadas por áreas temáticas e por ordem cronológica.

Publicações sobre as condições de operação dos sistemas de controlo distribuído:

1. “CAN-based real time adaptive distributed control”, Ana Antunes, Fernando Morgado Dias, Alexandre Manuel Mota, 8th International CAN Conference, Las Vegas, EUA, Fevereiro de 2002.
2. “Influence of the sampling period in the performance of a real-time distributed system under jitter conditions”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, 6th WSEAS International Conference on Telecommunications and Informatics (TELE-INFO'04), Cancun, México, Maio de 2004.
3. “Influence of the sampling period in the performance of a real-time distributed system under jitter conditions”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, WSEAS Transactions on Communications, vol. 3/1, pp. 248-253, 2004.

Publicações sobre a modelação do tempo morto fraccionário:

1. “Control performance of a real-time adaptive distributed control system under jitter conditions”, Ana Antunes, Alexandre Manuel Mota, Conference Control 2004, Bath, Reino Unido, Setembro de 2004.
2. “Fractional dead-time modelling for variable sampling to actuation delay compensation in distributed real-time control systems”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, International Transactions on Computer Science and Engineering, GESTS, Vol. 24/1, pp. 9-18, 2005.
3. “Improving the control performance of a real-time distributed control system under variable sampling to actuation delay”, Ana Antunes, Fernando Dias, Alexandre Mota, 25th IASTED International Conference on Modelling, Identification and Control, Lanzarote, Espanha, Fevereiro de 2006.

Publicações sobre a técnica de adaptação dinâmica do período de amostragem:

1. “Adapting the sampling period of a real-time adaptive distributed controller to the bus load”, Ana Antunes, Paulo Pedreiras, Alexandre Mota, 10th IEEE International Conference on Emerging Technologies and Factory Automation, Catania, Itália, Setembro de 2005.
2. “Dynamic rate adaptation in distributed computer control systems”, Ana Antunes, Paulo Pedreiras, Luís Almeida, Alexandre Mota, ANIPLA 2006 – Methodologies for Emerging Technologies in Automation, Roma, Itália, Novembro de 2006.
3. “Improving operational flexibility in distributed control systems: the Dynamic Rate Adaptation technique”, Ana Antunes, Paulo Pedreiras, Luís Almeida, Alexandre Mota, Automazione e Strumentazione - Ellectronica Industriale, ano LV, n^o2, Fevereiro de 2007.
4. “Dynamic rate adaptation: a method do improve operational flexibility in distributed control systems”, poster convidado na sessão organizada pelo ARTIST2-Network of Excellence on Embedded Systems Design no âmbito do 2nd IEEE Int. Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, Munique, Alemanha, Maio de 2007.
5. “Dynamic rate and control adaptation in networked control systems”, Ana Antunes, Paulo Pedreiras, Luís Almeida, Alexandre Mota, 5th IEEE International Conference on Industrial Informatics, Viena, Áustria, Julho de 2007.
6. “Dynamic rate adaptation technique for distributed computer control systems”, Ana Antunes, Paulo Pedreiras, Luís Almeida, Alexandre Mota, seleccionado para publicação no livro Modelling and control of discrete production systems, ISA/O3NEDIA em Julho de 2007.

Publicações sobre a técnica do compensador de atraso:

1. “Delay Compensator: an approach to reduce the variable sampling to actuation delay effect in distributed real-time control systems”, Ana Antunes, Fernando Morgado Dias, José Vieira, Alexandre Mota, 11th IEEE International Conference on Emerging Technologies and Factory Automation, Praga, República Checa, Setembro de 2006.
2. “Comparing the delay compensator approach with fractional dead-time modeling in distributed real-time control systems”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, aceite para publicação na 8th Portuguese Conference on Automatic Control- CONTROLO 2008, Vila Real, Portugal, Julho de 2008.

3. “A neural model for delay correction in a distributed control system”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, aceite para publicação na 18th International Conference on Artificial Neural Networks, Praga, República Checa, Setembro de 2008.
4. “A neural network delay compensator for networked control systems”, Ana Antunes, Fernando Morgado Dias, Alexandre Mota, aceite para publicação na 13th IEEE International Conference on Emerging Technologies and Factory Automation, Hamburgo, Alemanha, Setembro de 2008.

Publicações sobre o barramento CAN:

1. “O protocolo CAN”, Ana Antunes, Alexandre Manuel Mota, Ias Jornadas Politécnicas de Engenharia, Leiria, Portugal, Novembro de 2001.

1.7 Conclusão

Neste capítulo foi apresentada uma breve perspectiva histórica sobre a área de controlo distribuído, a motivação para a realização deste trabalho, os pressupostos de base para o desenvolvimento do mesmo, os principais contributos que dele resultaram, assim como as publicações efectuadas neste âmbito.

Capítulo 2

Sistemas de controlo distribuído em tempo-real

Os sistemas de controlo distribuído são utilizados em aplicações com as quais se convive diariamente, embora muitas vezes não se tenha consciência disso pois a maior parte das aplicações deste tipo de sistemas são embutidas. Eles são utilizados por exemplo em automóveis, aviões, domótica, robótica, sistema de apoio médico e em linhas de produção de indústrias de vários tipos.

A figura 2.1 apresenta a arquitectura de um sistema de controlo distribuído. Este tipo de sistema caracteriza-se pela utilização de uma rede de comunicação, normalmente um barramento de campo, que serve para a troca de informação entre os nós do sensor, do actuador e do controlador. O nó do sensor é responsável pela medição da saída do sistema, designada por amostragem. O controlador efectua os cálculos necessários de acordo com a função de controlo a implementar. Finalmente o nó do actuador, actua sobre o sistema através da aplicação ao sistema do valor dado pelo controlador. Este procedimento designa-se por actuação. Este tipo de sistemas são também muitas vezes designados por *Networked Control Systems* (NCS).

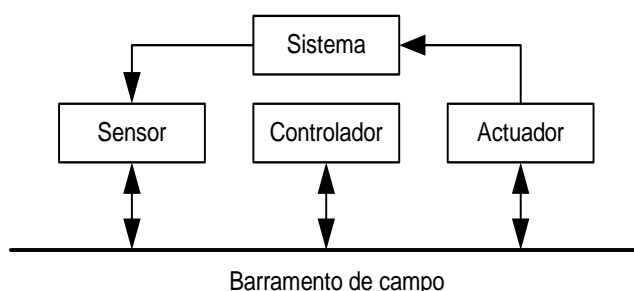


Figura 2.1: Diagrama de blocos de um sistema de controlo distribuído.

A distribuição do sistema de controlo apresenta vantagens como a redução da cablagem, modularidade, flexibilidade de operação, melhor utilização dos recursos, maior

facilidade de diagnóstico, de manutenção, de reconfiguração e custo de implementação mais reduzido [6], [7], [8], [9] e [3].

Essa distribuição apresenta também desvantagens como a introdução de atrasos, muitas vezes variáveis, na malha de controlo que podem originar incerteza nos instantes de amostragem e de actuação do sistema. O aparecimento de atraso variável em cada iteração de controlo está relacionado com as condições de implementação do sistema distribuído e será analisado em detalhe na secção seguinte deste capítulo.

A implementação bem sucedida de um sistema de controlo distribuído em tempo-real requer conhecimentos em vários domínios, conhecimentos esses que têm que ser aplicados de modo integrado. Esses domínios são: os sistemas de tempo-real, sistemas de comunicação e sistemas de controlo.

Este capítulo apresenta o resumo dos tópicos mais importantes nas áreas de sistemas de tempo-real, sistemas de tempo-real distribuídos, assim como um resumo das técnicas de controlo utilizadas em controlo distribuído.

2.1 Distribuição de controladores

As características dos sistemas de controlo distribuído, resultantes do modo como são implementados, têm implicações no desempenho do controlador.

A distribuição do sistema de controlo pressupõe a localização do sensor (S), do controlador (C) e do actuador (A) em nós distintos, ligados entre si através de uma rede de comunicação que serve de suporte para a troca de informação entre os vários nós. Nas aplicações de controlo a rede de comunicação, como será referido na secção 2.3.1, é normalmente implementada com recurso a um barramento de campo. A figura 2.2 apresenta o diagrama de blocos detalhado de um sistema de controlo distribuído, assim como os tempos de processamento e de comunicação associados ao seu funcionamento.

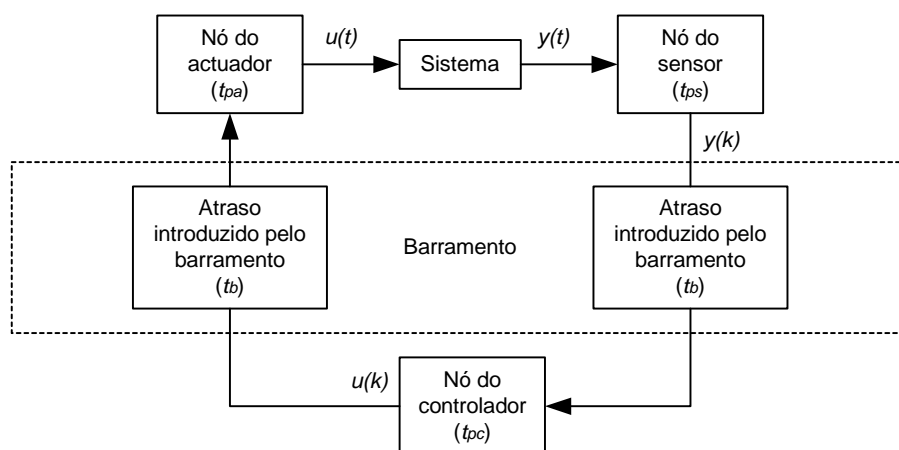


Figura 2.2: Diagrama de blocos detalhado de um sistema de controlo distribuído.

Como se pode ver existem atrasos associados tanto ao processamento das tarefas em cada nó como à utilização do barramento. Esses atrasos levam a que exista incerteza

quanto ao instante de actuação sobre o sistema. Pode também existir incerteza em relação ao instante de amostragem, dependendo do modo como é feita a activação da tarefa responsável pela mesma.

2.1.1 Incerteza quanto aos instantes de amostragem e de actuação

Na figura 2.2 é possível identificar os atrasos que medeiam entre o instante de amostragem e o instante de actuação. Esses atrasos são de dois tipos: tempo de processamento interno em cada um dos nós, como t_{ps} , t_{pc} e t_{pa} , e tempo introduzido pelo mecanismo de controlo do acesso ao barramento, como t_b . Tanto os atrasos decorrentes do processamento das tarefas como os atrasos introduzidos pelo acesso ao barramento podem ser variáveis, dependendo do modo como é feita a implementação do sistema distribuído. De seguida esses atrasos serão analisados em detalhe.

O tempo de acesso ao barramento depende do método de controlo de acesso ao meio (MAC da designação em inglês *Medium Access Control*) específico de cada barramento, do tipo de tráfego [10] e dos mecanismos de escalonamento utilizados, sendo normalmente variável¹.

O tempo de processamento associado a cada um dos nós pode ser, ou não, variável dependendo do modo como o nó é implementado. No caso de se utilizar um microprocessador para a implementação (o que acontece no caso do nó do controlador), basta que ele seja partilhado por várias aplicações para que o tempo de processamento varie em função do escalonamento interno das várias tarefas que competem pela utilização da unidade central de processamento (UCP). O tempo de execução do algoritmo de controlo pode, também ele, ser variável de acordo com o fluxo de controlo que é seguido em cada iteração. A nível do processador existem também outros factores que contribuem para que o tempo de processamento seja variável, como o serviço de interrupções ou o acesso a memórias *cache*.

A figura 2.3 apresenta os tempos envolvidos entre o instante de amostragem e o instante de actuação para cada ciclo de controlo. O período de amostragem é representado por h . As setas \downarrow e \uparrow representam, respectivamente, o instante de amostragem e o instante de actuação.

Como se pode ver na figura 2.3 o tempo entre a amostragem e a actuação pode ser expresso por

$$t_{sa} = t_{sc} + t_{ca} \quad (2.1)$$

sendo variável de ciclo para ciclo de controlo e originando por isso incerteza no instante de actuação. Essa incerteza é designada, neste contexto, por *jitter* (de actuação neste caso). Podem obter-se expressões mais pormenorizadas para t_{sa} . Basta comparar a figura 2.3 com a figura 2.2. para se obterem as expressões:

$$t_{sc} = t_{ps} + t_b \quad (2.2)$$

¹Os sistemas totalmente *time-triggered* constituem a excepção.

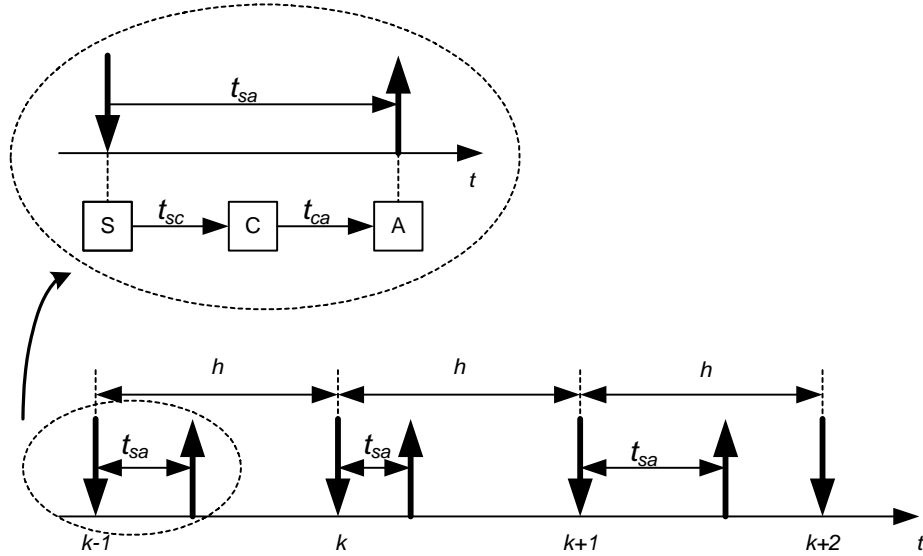


Figura 2.3: Representação temporal do tempo entre a amostragem e a actuação.

$$t_{ca} = t_{pc} + t_b + t_{pa} \quad (2.3)$$

O atraso introduzido pelo dispositivo de controlo do acesso ao barramento é variável para cada acesso, dependendo do tráfego presente no barramento e do mecanismo de escalonamento utilizado, o que faz com que t_{sa} seja variável.

Pode também existir incerteza, ou *jitter*, associada ao instante de amostragem, dependendo do modo como é feita a activação da tarefa que efectua a amostragem do sistema e da utilização dada ao processador onde essa tarefa é executada. Se a activação for feita através da recepção de uma mensagem enviada através do barramento ou se o processador for partilhado por várias tarefas vai existir incerteza.

Ao longo deste trabalho considera-se que não existe incerteza no instante de amostragem, ou seja o *jitter* de amostragem é nulo, pelo que o período de amostragem é constante. Este pressuposto deriva do facto de se considerar que a tarefa que efectua a amostragem do sistema é autónoma, sendo o início da sua execução controlado localmente no nó sensor, e que o processador não é partilhado por outras tarefas, ou no caso de ser, que a tarefa responsável pela amostragem do sistema é a mais prioritária.

Pressupõe-se ainda que $t_{sa} < h$, ou seja que o atraso introduzido entre o instante de amostragem e o instante de actuação não excede o valor do período de amostragem. Não é por isso considerada a situação em que se perdem amostras entre ciclos de controlo (*vacant sampling*). Esta situação é tratada por exemplo em [11], [12], [13], [14], [15] e [16].

A existência de incerteza no instante de amostragem e de actuação e a consequente introdução de atrasos variáveis na malha de controlo pode provocar degradação no desempenho do controlador podendo mesmo levar à instabilidade da malha [15], [4], [17], [5] e [8]. Na secção 2.4 será apresentado um resumo das técnicas existentes para

o projecto deste tipo de sistemas de controlo.

2.2 Sistemas de tempo-real

Os sistemas de tempo-real fazem parte do dia-a-dia das sociedades modernas estando presentes em dispositivos com os quais se interage diariamente de forma directa, como por exemplo os automóveis, os electrodomésticos, os sistemas de telecomunicações e de entretenimento, assim como, a um nível mais genérico, em aplicações industriais, sistemas de transporte público, indústria aeronáutica, entre outras.

A sua introdução iniciou-se em aplicações industriais e a sua popularização deu-se quando os avanços tecnológicos permitiram que o custo das implementações baixasse o suficiente para poderem ser utilizados em dispositivos de uso quotidiano e massivo. A relevância actual deste tipo de sistemas é elevada porque o campo de aplicações que a eles recorrem é muito vasto. O facto das implementações deste tipo de sistemas serem embutidas faz com que os utilizadores finais não tenham muitas vezes consciência da sua importância, ou mesmo da sua existência.

2.2.1 Conceitos básicos

Um sistema diz-se de tempo-real quando a correcção do seu comportamento depende não só do valor final resultante da sua acção mas também do instante temporal em que esse valor é produzido. Um dos exemplos típicos de um sistema de tempo-real é um controlador digital, ou seja um controlador implementado com recurso a um computador. Num controlador de tempo-real importa não só o valor de controlo que vai actuar sobre o sistema mas também o momento em que esse valor é aplicado. Isto porque se a actuação sobre o sistema for tardia ele pode deixar de operar nas condições desejadas, podendo produzir resultados indesejáveis ou mesmo catastróficos. Normalmente existe um computador associado à implementação do sistema de tempo-real. Esse computador é responsável pela execução das tarefas necessárias.

A figura 2.4 apresenta os parâmetros típicos utilizados para a caracterização de uma tarefa.

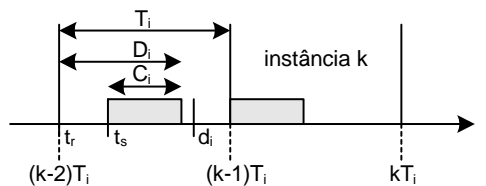


Figura 2.4: Parâmetros típicos para a caracterização de uma tarefa.

Designa-se por *deadline* (D_i) o intervalo temporal dentro do qual a tarefa deve ser terminada. Este intervalo é dado em relação ao instante em que a tarefa fica pronta para execução que se designa por *release time* (t_r). Este *deadline* é designado por

deadline relativo, sendo designado por *deadline* absoluto (d_i) o instante de tempo até ao qual a tarefa deve estar terminada. Na figura anterior *start time* (t_s) designa o instante em que a tarefa começa a ser executada e C_i o tempo de execução no pior caso. As tarefas periódicas são activadas periodicamente com período T_i . Cada uma dessas activações designa-se por instância (*job*).

Os sistemas de tempo-real podem classificar-se de acordo com o tipo de restrições temporais que impõem [18] em *hard* ou *soft*.

Um sistema de tempo-real é do tipo *hard* se a produção do resultado para além do instante temporal imposto pela *deadline* pode originar falhas catastróficas na operação do sistema ou no ambiente em que este se insere. Neste tipo de sistemas a falha produzida pode ter consequências fatais, como por exemplo no caso de uma passagem de nível automática. Se a cancela não for fechada dentro de um período de tempo bem determinado após a detecção da aproximação do comboio pode dar-se um acidente que pode resultar na perda de vidas. Os sistemas de controlo são exemplos típicos de sistemas de tempo-real *hard*.

Se por outro lado o *deadline* puder ser ocasionalmente ultrapassado, com perda do desempenho do sistema mas sem consequências desastrosas, então o sistema de tempo-real diz-se do tipo *soft*. Um exemplo típico são as aplicações multimédia como por exemplo o *streaming* de vídeo.

Este trabalho trata de sistemas de controlo, por isso de sistemas de tempo-real do tipo *hard*.

A investigação na área dos sistemas de tempo-real foi desenvolvida inicialmente para sistemas centralizados, em que o recurso partilhado era o processador e não a rede de comunicação, como acontece nas aplicações distribuídas, embora para os sistemas distribuídos se utilizem técnicas semelhantes.

2.2.2 Técnicas de escalonamento

Em muitas das aplicações embutidas actuais o processador é utilizado por vários sistemas para executar mais do que uma tarefa. As técnicas de escalonamento servem para estabelecer a ordem pela qual essas tarefas devem ser executadas de modo a que se cumpram os requisitos temporais estabelecidos para o sistema. Certas técnicas podem também ser utilizadas para prever se os *deadlines* de todas as tarefas são ou não cumpridos. O escalonamento faz-se com base num modelo da tarefa e num algoritmo de escalonamento.

O problema genérico de escalonar um conjunto de tarefas em que existam relações de precedência e restrições temporais associadas é um problema do tipo *NP-complete* [19], ou seja, não é exequível num contexto em que existam recursos de processamento limitados. De modo a reduzir a complexidade dos cálculos associados ao processo de escalonamento e a garantir que esse processo conduz a soluções praticáveis é necessário simplificar os modelos que se utilizam para caracterizar as tarefas considerando, por exemplo, que as tarefas são independentes umas das outras, o que acontece muitas vezes nas aplicações reais.

As tarefas podem ser periódicas, aperiódicas ou esporádicas. As tarefas periódicas

são activadas regularmente com um período, que neste contexto é usualmente designado pela letra T . Uma tarefa aperiódica é activada de modo irregular. As tarefas aperiódicas que possam ser caracterizadas por um tempo mínimo entre activações sucessivas são designadas esporádicas.

Tradicionalmente uma tarefa é representada por um período de valor fixo (T), por um *deadline* (D), igual ao período, e pelo tempo de execução no pior caso possível (C) [5]. No caso das tarefas esporádicas o valor T representa o tempo mínimo entre activações sucessivas da tarefa e não o período.

Existem outros modelos para representar as tarefas como por exemplo o modelo elástico [20], [21] e [22], que admite que certos parâmetros da tarefa podem variar dentro de uma dada gama de valores. Este modelo é muito útil para modelar tarefas com período variável.

Os algoritmos de escalonamento podem ser classificados de diversos modos. A figura 2.5 apresenta o esquema de classificação mais comum [23].

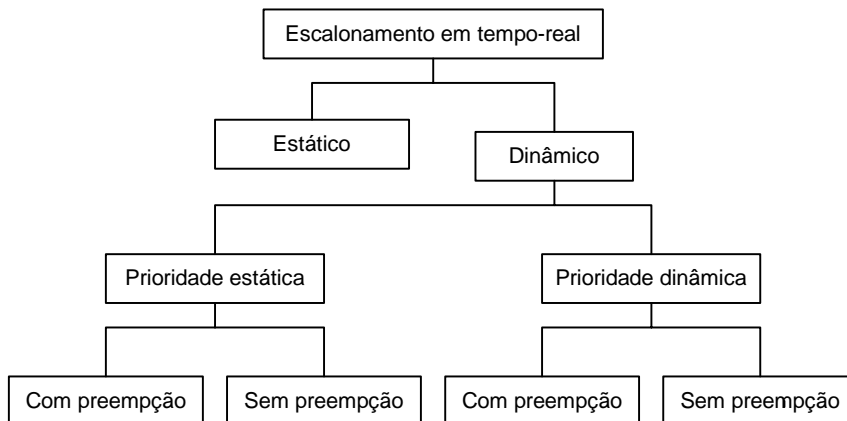


Figura 2.5: Classificação dos algoritmos de escalonamento de tempo-real.

Existem inúmeros algoritmos de escalonamento para cada classe e nesta área continua a ser efectuada muita pesquisa. Neste documento são referidos apenas os algoritmos considerados mais importantes. Para mais detalhes sobre outros tipos de algoritmos de escalonamento consultar por exemplo [23]. Mohammadi e Akl [24] apresentam também um resumo bastante exaustivo do problema do escalonamento em sistemas de tempo-real uniprocessador e multiprocessador.

O escalonamento pode ser estático ou dinâmico. O escalonamento estático efectua-se *off-line* recorrendo a algoritmos de optimização de modo a gerar uma tabela contendo a ordem pela qual um determinado grupo de tarefas deve ser executado para que todos os *deadlines* sejam cumpridos. O escalonamento dinâmico é feito durante a operação do sistema, ou seja *online*, e não requer o conhecimento das características de todo o conjunto de tarefas, uma vez que é efectuado quando uma nova tarefa entra no sistema e sempre que uma tarefa fica pronta para execução ou é concluída [4].

A abordagem estática tem a vantagem de ser simples de implementar. No entanto, apresenta uma série de desvantagens [25], entre elas o facto de tornar difícil a inclusão

de tarefas esporádicas ou aperiódicas e de não permitir a alteração do conjunto de mensagens durante a operação do sistema, exibindo assim um grau de flexibilidade muito limitado. A grande vantagem deste método é o facto de ser determinístico.

A abordagem dinâmica tem a vantagem de ser flexível e adaptável e apresenta as desvantagens de não ser completamente previsível e de implicar *overheads* devido ao facto do escalonamento ser feito durante a operação do sistema [4]. Também existem técnicas mistas que tentam tirar partido das vantagens dos dois métodos, como por exemplo as referidas por Fohler [26], embora não sejam as de uso mais comum.

As técnicas dinâmicas podem basear-se em prioridade fixa (FP) como nos algoritmos *Rate-Monotonic* (RM) [27] e *Deadline-Monotonic* (DM) [27] ou em prioridade dinâmica como no algoritmo *Earliest-Deadline-First* (EDF) e nos algoritmos baseados em servidores (por exemplo [28], [29] e [30]).

Os algoritmos de escalonamento podem admitir ou não a preempção das tarefas. Um algoritmo de escalonamento diz-se com preempção quando a tarefa que está a ser executada pode ser interrompida para a execução de outra tarefa considerada mais prioritária, sendo depois retomada a execução da tarefa anterior a partir da altura em que tinha sido interrompida. No caso da tarefa executar sem possibilidade de interrupção até terminar, o escalonamento não admite preempção [31].

As técnicas de prioridade dinâmica podem ainda classificar-se em técnicas baseadas em planeamento e técnicas baseadas no melhor esforço. As técnicas baseadas em planeamento garantem que se uma tarefa é aceite para execução tanto ela como todas as tarefas aceites anteriormente vão cumprir os seus requisitos temporais. Este tipo de algoritmos tentam melhorar a resposta e o desempenho do sistema em relação às tarefas aperiódicas e de tempo-real *soft* continuando a garantir o cumprimento dos requisitos temporais das tarefas de tempo-real do tipo *hard* [24]. Exemplo de técnicas deste tipo são os já referidos algoritmos EDF e baseados em servidores. Quanto às técnicas de melhor esforço elas tentam garantir às tarefas as melhores condições de escalonamento quando há sobrecarga do sistema. A definição dessas condições é feita pelo utilizador e pode estar relacionada por exemplo com o consumo de energia [24]. Estas técnicas são normalmente aplicadas para escalonar tarefas do tipo *soft*.

Neste texto serão abordadas algumas das técnicas dinâmicas mais importantes como sejam os algoritmos de prioridade fixa *rate-monotonic* e *deadline-monotonic* e os algoritmos de prioridade dinâmica *earliest-deadline-first* e baseados em servidores.

Na análise dos algoritmos de escalonamento considera-se que cada tarefa τ_i é caracterizada por um período T_i , um *deadline* D_i e um tempo de execução no pior caso C_i , de acordo com o modelo mais habitual, anteriormente referido.

Escalonamento com Prioridade Fixa

A técnica de prioridade fixa é a mais comum sendo suportada pela maioria dos sistemas operativos comerciais [5]. Quando se utiliza este tipo de escalonamento atribui-se a cada tarefa τ_i uma prioridade fixa P_i . Quando várias tarefas estão prontas para execução selecciona-se a que tiver prioridade maior. Nos sistemas com preempção, se ficar pronta para execução uma tarefa com prioridade maior do que a da tarefa que

está a ser executada, é efectuada a troca entre ambas de modo a que a cada momento esteja a ser executada a tarefa mais prioritária.

A atribuição de prioridades do tipo *rate-monotonic* [27] baseia-se no período das tarefas. Tarefas com período mais curto têm prioridade maior. Foi demonstrado por Liu e Layland [27] que para $D_i = T_i$ este esquema de atribuição é óptimo, no sentido em que se um dado conjunto de tarefas não for escalonável sob este esquema não será sob nenhum outro com prioridades fixas.

Em muitos casos as tarefas têm *deadlines* inferiores ao seu período, $D_i \leq T_i$. Quando isso se verifica para todas as tarefas o esquema de atribuição *deadline-monotonic* é óptimo, no sentido que foi descrito anteriormente [32]. A atribuição de prioridades faz-se neste caso de acordo com o *deadline*. Tarefas com *deadlines* mais curtos ficam com prioridade mais elevada. O esquema RM é um caso particular do esquema DM.

Dado um conjunto de n tarefas a análise da escalonabilidade faz-se recorrendo às características das tarefas, como sejam o tempo de execução no pior caso possível C_i e o período T_i . Para o esquema RM foi demonstrado por Liu e Layland [27] que a condição 2.4 é suficiente (mas não necessária) para garantir a escalonabilidade. A condição testa a utilização do processador obtida para o conjunto de tarefas. Com o aumento do número de tarefas, a utilização U (neste caso do processador) aproxima-se de 0.63 [33].

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq n(2^{\frac{1}{n}} - 1) \quad (2.4)$$

Existe um teste de escalonabilidade exacto para o esquema RM baseado no conhecimento do tempo de resposta no pior caso R_i de cada tarefa [34]. O tempo de resposta é definido como o tempo que decorre entre o momento em que a tarefa fica pronta para execução e o momento em que termina a sua execução [5]. O valor máximo de R_i obtém-se quando todas as tarefas ficam simultaneamente disponíveis para execução e determina-se através da equação recursiva seguinte

$$R_i = C_i + \sum_{j \in mp(i)} \left\lceil \frac{R_j}{T_j} \right\rceil C_j$$

em que $mp(i)$ representa o conjunto de tarefas com prioridade maior do que τ_i e $\lceil \cdot \rceil$ representa a função limite superior.

O conjunto é escalonável se e só se $R_i \leq D_i$ para todas as tarefas.

Mais recentemente foram desenvolvidas análises que permitem determinar o tempo de resposta das tarefas no melhor caso [35]. A análise combinada dos tempos de resposta no melhor e no pior caso permite determinar a incerteza associada à resposta de cada tarefa [5].

Escalonamento *Earliest-Deadline First*

O escalonamento *earliest-deadline first* selecciona para execução a tarefa que esteja mais próxima de atingir o *deadline*. Este tipo de escalonamento permite escalonar mais conjuntos de tarefas do que o escalonamento com prioridade fixa devido à sua

natureza mais dinâmica [5]. Este tipo de escalonamento tem sido utilizado em sistemas operativos experimentais [36].

A análise de escalonabilidade é determinada de modo exacto pela condição 2.5 quando $D_i = T_i$ para todas as tarefas [27]. O conjunto de tarefas é escalonável se e só se a condição 2.5 se verificar.

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (2.5)$$

Uma vez mais U representa a utilização do processador.

É imediato concluir que este tipo de escalonamento permite a utilização total do processador. Este teste tem ainda a vantagem de ser mais simples de implementar do que o teste, baseado no tempo de resposta, requerido para o escalonamento de prioridade fixa.

Burns e Wellings em [37] descrevem uma série de desvantagens do escalonamento EDF face ao escalonamento FP relacionadas com os tempos de *overhead*, a atribuição de *deadlines* e o comportamento durante sobrecargas mas Cervin [5] desvaloriza os argumentos utilizados contrapondo com a apresentação de exemplos. Mais recentemente Buttazzo [36] faz uma comparação exaustiva entre os algoritmos RM e EDF para concluir que muitas das desvantagens atribuídas ao algoritmo EDF não têm fundamentação real, salientando que a única vantagem do algoritmo RM face ao EDF é a sua simplicidade de implementação, confirmada pela sua aplicação no topo de sistemas operativos comerciais.

No caso em que $D_i \leq T_i$ a análise de escalonabilidade torna-se mais complexa. Stankovic et al. [38] apresentam o seguinte teste que é muito genérico. Dado um conjunto arbitrário de tarefas, em que cada instância de uma tarefa é descrita pelo tempo de execução C_k , um tempo de *release* r_k e um *deadline* absoluto d_k , define-se a exigência E_x do conjunto de tarefas para um dado intervalo de tempo $[t_1, t_2]$ como

$$E_x [t_1, t_2] = \sum_{r_k \geq t_1 \wedge d_k \leq t_2} C_k$$

O factor de carga das tarefas L é dado pela expressão 2.6 em que a maximização é calculada sobre todos os intervalos de tempo possíveis.

$$L = \max_{0 \leq t_1 \leq t_2} \frac{E_x [t_1, t_2]}{t_2 - t_1} \quad (2.6)$$

O conjunto de tarefas é escalonável com o algoritmo EDF se e só se $L \leq 1$.

A análise do pior tempo de resposta das tarefas em sistemas que utilizam este tipo de escalonamento é mais difícil do que no caso do escalonamento FP porque é difícil definir o instante crítico em que as tarefas sofrem o máximo de interferência. Stankovic et al. [38] apresentam também fórmulas para cálculo do tempo de resposta para este caso. Mais recentemente Alberts e Slomka [39] apresentam novos testes de escalonabilidade exacta cuja execução é mais rápida do que os anteriores.

Escalonamento baseado em servidores

As técnicas anteriores utilizam-se para fazer o escalonamento de tarefas periódicas e não se adequam ao tráfego aperiódico ou do tipo *soft*. De modo a permitir o escalonamento deste tipo de tarefas, sem contudo deixar de garantir o escalonamento das tarefas do tipo *hard*, surge o escalonamento baseado em servidores.

Este tipo de escalonamento permite a existência de uma tarefa, designada servidor, que é utilizada para escalonar o tráfego aperiódico que estiver pendente. Ao servidor é associado um limite de utilização designado quota (em inglês *budget*), que pode ser gasta para escalonar e executar as instâncias das tarefas que estejam pendentes. O servidor nunca pode ser executado para além da sua quota. Por isso a execução das tarefas aperiódicas pode ter que ser interrompida se a quota do servidor se esgotar. A execução é retomada da próxima vez que o servidor for executado.

Foram desenvolvidas técnicas de escalonamento baseado em servidores com prioridades fixas, como o *Priority-Exchange Server* (PES) [28], o *Deferrable Server* (DS) [28] e o *Sporadic Server* (SS) [29], e com prioridades dinâmicas como por exemplo o *Constant Bandwidth Server* (CBS) [30], [40] e [41]. A diferença entre os vários tipos de servidores relaciona-se com o modo como a quota é repostada e com a capacidade do servidor.

2.3 Sistemas de tempo-real distribuídos

A pressão associada à redução de custos bem como as vantagens já descritas da distribuição levaram a um nível cada vez maior de automação numa série de indústrias, como sejam a indústria de processos, aeronáutica, automóvel e domótica, entre outras. O recurso a soluções distribuídas implica a utilização de meios de comunicação em tempo-real fiáveis que permitam obter os elevados níveis de rendimento e fiabilidade necessários para este tipo de aplicações.

2.3.1 Comunicação em tempo-real

A distribuição do sistema de controlo implica a utilização de uma rede de comunicação através da qual são enviadas mensagens contendo a informação relevante para a malha de controlo. Como um sistema de controlo é um sistema de tempo-real, a rede de comunicação deve garantir o transporte da informação dentro de um intervalo de tempo limitado pelo *deadline* imposto à aplicação e com grande fiabilidade. Este tipo de comunicação, em que existem restrições temporais associadas à operação da rede, designa-se por comunicação em tempo-real.

De acordo com Kopetz [31] as características requeridas para um sistema de comunicação de tempo-real são as seguintes: promover a modularidade e permitir flexibilidade; proporcionar mecanismos de detecção de erros; apresentar uma latência²

²Latência designa o tempo que medeia entre o início da transmissão da mensagem na interface de comunicação com a rede do nó emissor e a entrega dessa mensagem na interface correspondente do nó receptor.

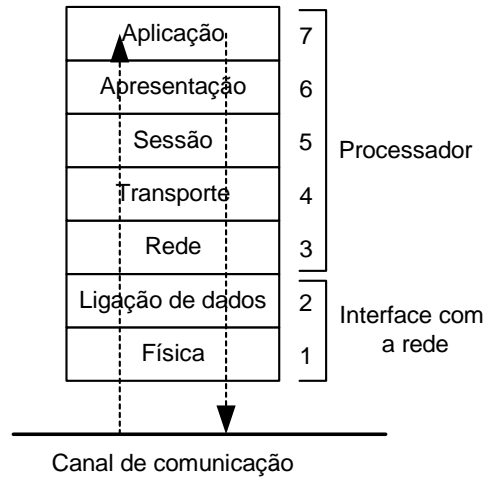


Figura 2.6: Camadas do modelo de referência OSI.

previsível, pequena e com um mínimo de *jitter*.

O modelo de referência OSI (*Open Systems Interconnection*) serve de base para descrever a arquitectura conceptual de um sistema de comunicação genérico que deve ser implementada em cada nó. Este modelo é composto por sete camadas, como se pode ver na figura 2.6. Desde que a informação é recebida da rede, pela camada física, até chegar à camada da aplicação ela passa pelas camadas: ligação de dados, rede, transporte, sessão e apresentação.

As duas camadas da base são implementadas pela unidade de interface com a rede e as cinco do topo pelo processador. Normalmente para cada camada é implementado um protocolo específico.

Em aplicações de tempo-real recursos como a capacidade de processamento e a quantidade de memória são muitas vezes limitados. Por outro lado é desejável que a latência associada à transferência de uma mensagem de um nó para outro seja a mínima possível. Devido a estes factores nas redes de comunicação de tempo-real o modelo de referência OSI não é utilizado na totalidade sendo substituído pelo modelo simplificado apresentado na figura 2.7. Este modelo designa-se *Enhanced Performance Architecture* (EPA) [42] e [43].

A camada física é responsável pela codificação, descodificação e temporização dos *bits*, pela sincronização e pelas características físicas do barramento, como a topologia da rede, o canal físico utilizado para a comunicação, a taxa de transmissão dos sinais, o tamanho máximo do barramento, o número máximo de nós que podem ser ligadas à rede e a imunidade às interferências electromagnéticas [44].

A camada da ligação de dados ocupa-se da transferência de unidades de informação entre vários pontos da rede, sendo responsável pelo encapsulamento/dencapsulamento da informação sob a forma de tramas, pelo controlo do acesso ao meio e pela detecção, sinalização e limitação de erros.

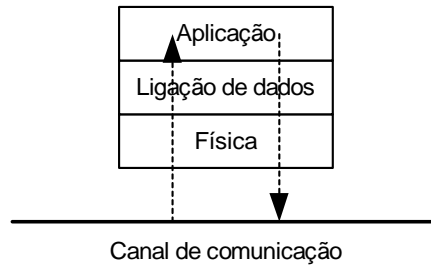


Figura 2.7: Camadas do modelo EPA.

A camada da aplicação selecciona os serviços apropriados para cada uma das aplicações.

Normalmente as comunicações de tempo-real são implementadas recorrendo a redes de acesso múltiplo [45].

Existem diferentes tipos de topologias possíveis para as infra-estruturas de comunicação de tempo-real em sistemas distribuídos, como por exemplo barramento, estrela, anel, malha ou árvore [46] e [47]. Destas as mais utilizadas em sistemas de tempo-real *hard* são as topologias em barramento e em anel [31]. A figura 2.8 apresenta a topologia em barramento, a qual é utilizada neste trabalho.

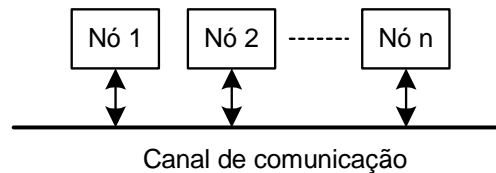


Figura 2.8: Topologia em barramento.

Neste tipo de topologia todos os nós estão ligados sobre um canal de comunicação comum podendo por isso partilhar a comunicação. O sinal percorre todo o canal de comunicação. Para evitar a reflexão do sinal em cada extremo do canal utilizam-se terminadores. Num barramento os nós só podem comunicar um de cada vez sendo necessário definir um critério de acesso ao meio que determine em cada momento qual dos nós é que pode transmitir.

Comunicação *Time-Triggered* e comunicação *Event-Triggered*

O modo como é iniciada a comunicação permite distinguir duas abordagens distintas utilizadas pelas aplicações, são elas a abordagem *Time-Triggered* (TT) e a abordagem *Event-Triggered* (ET).

Na abordagem TT a comunicação ocorre em instantes temporais pré-definidos. A taxa de ocorrência desses instantes é determinada pela dinâmica do sistema sob controlo. Uma das vantagens desta abordagem é o facto de permitir o estabelecimento de

um desfasamento relativo entre as mensagens que são enviadas para o sistema de comunicação [45]. Esta característica permite que os sistemas que utilizam a abordagem TT sejam considerados modulares em relação ao seu comportamento temporal [31], uma das características mais importantes deste tipo de sistemas. Esta abordagem é muito utilizada para o tratamento das mensagens periódicas.

Na abordagem ET a comunicação é iniciada pela ocorrência de um dado acontecimento, como por exemplo a alteração do valor de uma entrada. Esta abordagem é mais adequada ao tratamento de mensagens aperiódicas.

Ambas as abordagens têm vantagens e desvantagens [45] e embora tenha existido um grande debate acerca de qual delas seria a melhor ([48] e [49]) não se chegaram a conclusões definitivas.

Na maioria das aplicações existe a necessidade de permitir tanto tráfego TT como tráfego ET e por isso de modo explícito ou implícito a maioria dos protocolos permite que as duas abordagens coexistam.

Modelos de cooperação

Num sistema distribuído os nós têm que cooperar entre si para realizarem as operações necessárias. Existem diversos modelos de cooperação como sejam o modelo cliente-servidor, o modelo produtor-consumidor [50] e o modelo Produtor-Distribuidor-Consumidor (PDC) [51].

No modelo produtor-consumidor a ênfase é colocada no significado da informação trocada e não na origem ou destino dessa informação. Assim os nós produtores produzem informação e enviam-na através de mensagens e o nó ou nós consumidores recebem as mensagens se a informação nelas contida for relevante para eles. Normalmente a troca de informação é iniciada pelo nó produtor e o nó consumidor limita-se a reagir. A comunicação pode dar-se entre um produtor e um consumidor ou simultaneamente entre um produtor e vários consumidores.

Nos casos em que existem diversos nós produtores é necessário determinar uma ordem para eles acederem ao meio de comunicação. Nesta situação pode utilizar-se um modelo PDC em que os nós produtores se comportam como *slaves* em relação a um nó de arbitragem (*master*). A troca de informação entre os nós consumidores e os nós produtores é assim mediada por um nó distribuidor.

No modelo cliente-servidor os nós que produzem informação relevante para outros comportam-se como servidores. Os nós que necessitam da informação produzida (clientes) enviam um pedido para o servidor que em resposta lhes devolve a informação requerida.

2.3.2 Controlo de acesso ao meio

Pela sua influência directa no atraso introduzido pelo sistema de comunicação na malha de controlo serão referidas algumas das estratégias de controlo de acesso ao meio (MAC) mais utilizadas pelos barramentos de campo.

A estratégia de controlo de acesso ao meio de um protocolo de comunicação específica qual o nó que tem acesso ao canal de comunicação em cada instante [31].

As estratégias utilizadas ao nível do MAC podem dividir-se em dois tipos: controladas e não controladas [52]. Nas estratégias controladas existe um sinal de controlo explícito ou implícito que determina quando é que determinado nó deve transmitir. Nas estratégias não controladas esse sinal externo não existe e a arbitragem pelo acesso ao meio baseia-se no estado do barramento e em informação local [44]. Cada um destes grupos pode ainda subdividir-se em dois: centralizado ou distribuído. As estratégias seguintes são as mais comuns nos protocolos de comunicação utilizados por sistemas de controlo distribuído: CSMA/CD, CSMA/NBA, passagem de testemunho (*token-passing*), controlo centralizado num *master* e TDMA. Uma análise de outros tipos de estratégias pode ser encontrada em [31], [43] e [44].

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

Este MAC é do tipo não-controlado e distribuído. De acordo com esta estratégia um nó que deseje transmitir deve monitorizar o estado do barramento e começar a transmissão quando detectar que o mesmo não está a ser utilizado. No caso de se verificarem colisões, o que acontece quando dois nós começam a transmitir simultaneamente, os nós envolvidos devem parar a transmissão e voltar a tentar depois de um intervalo de tempo aleatório. Esta estratégia permite a ocorrência de colisões em cadeia antes que o acesso seja definitivamente concedido a um nó. Esta situação verifica-se particularmente nos momentos em que há uma grande carga de tráfego e leva a que não seja muitas vezes possível determinar um limite máximo para a latência que pode afectar as mensagens [44]. Este tipo de estratégia é utilizado pelo protocolo Ethernet (IEEE 802.3). Devido à dificuldade em determinar a latência máxima das mensagens este protocolo não foi inicialmente considerado apropriado para comunicação em tempo-real, no entanto nos últimos anos têm surgido aplicações de controlo recorrendo a ele. Pedreiras [45] apresenta um resumo exaustivo das razões que levaram ao interesse pela utilização deste protocolo de comunicação com fins genéricos para aplicações de tempo-real e das alterações que podem ser efectuadas ao nível do MAC de modo a permitir a sua aplicação em comunicações de tempo-real.

Carrier Sense Multiple Access with Non-destructive Bitwise Arbitration (CSMA/NBA)

Este MAC é também do tipo não controlado e distribuído. De acordo com esta estratégia um nó que deseje transmitir deve esperar até que o barramento esteja livre. Quando o barramento está livre os nós podem começar a transmitir as suas mensagens monitorizando simultaneamente o barramento. A arbitragem no barramento é feita *bit* a *bit* utilizando o valor do identificador de cada mensagem de modo não destrutivo. À medida que os nós detectam que perderam o processo de arbitragem vão deixando de transmitir. No final do processo apenas um dos nós tem o acesso ao barramento para transmitir o resto da mensagem (ver secção 2.3.4). Esta de estratégia é utilizada no protocolo CAN (*Controller Area Network*) [2]. Este tipo de MAC é determinístico e permite o cálculo das latências máximas que as mensagens podem sofrer, dado um conjunto estático de mensagens [53] e [54].

Passagem de testemunho

Este MAC é do tipo controlado e pode ser implementado de forma centralizada ou distribuída. O direito de transmitir uma mensagem é concedido através da posse de uma mensagem de controlo designada testemunho. Na versão distribuída os nós estão organizados num anel virtual e o testemunho circula ao longo desse anel. O nó que tem o testemunho transmite as suas mensagens até se esgotar o tempo de que dispõe para manter o testemunho ou, no caso disso não chegar a acontecer, até não ter mais mensagens para enviar. Nessa altura o nó regenera o testemunho e envia-o para o nó seguinte no anel. Se um nó não tem mensagens para enviar limita-se a passar o testemunho para o nó seguinte. Esta estratégia é utilizada pelo protocolo PROFIBUS [55]. O tempo máximo de latência é determinado pelo tempo de rotação do testemunho através da rede [3].

Controlo centralizado num *master*

Neste MAC o acesso dos nós ao barramento é controlado por um nó especial designado *master* que serve de árbitro. Este MAC é do tipo controlado e centralizado. A comunicação pode efectuar-se seguindo o modelo *master-slave*. No modelo *master-slave* o nó *master* concede explicitamente, por meio de uma mensagem de controlo, o direito de transmitir a cada um dos outros nós designados por *slaves*. Esta estratégia é simples de implementar e o seu comportamento é previsível [44]. Alguns protocolos permitem que, no caso do *master* falhar, outro nó assuma o papel de *master* de modo a garantir a fiabilidade do barramento. Esta estratégia é utilizada pelo protocolo WorldFIP [55].

Time Division Multiple Access (TDMA)

Este MAC é do tipo controlado e distribuído. Nesta estratégia a concessão do acesso ao barramento é controlada pela passagem do tempo. A estratégia TDMA requer a implementação de uma base de tempo global acessível em todos os nós. O tempo global é dividido em janelas. Durante cada janela temporal apenas um nó pode aceder ao barramento. Esta técnica é estática e previsível. Ela é utilizada no protocolo *Time-Triggered Protocol (TTP)* [31].

2.3.3 Escalonamento das mensagens

De modo a garantir que todas as mensagens são entregues dentro dos seus *deadlines* utilizam-se técnicas de escalonamento para gerir o envio das mensagens através do barramento. Existem diferenças entre o escalonamento de tarefas em sistemas de computação e o escalonamento de mensagens numa rede de comunicação (ver por exemplo [56] para uma descrição dessas diferenças). A mais óbvia deve-se ao facto de não ser possível parar o envio de uma mensagem a meio da sua transmissão para depois se poder retomar o seu envio a partir desse ponto, o que inviabiliza o recurso a políticas de escalonamento com preempção.

Apesar dessas diferenças as técnicas de escalonamento utilizadas nos sistemas de comunicação de tempo-real são semelhantes às que se utilizam para as tarefas num

sistema de computação [44], havendo no entanto lugar a ajustes nos testes de escalabilidade de acordo com as características dos meios de acesso à rede utilizados pelos barramentos e protocolos.

2.3.4 Barramentos de campo

O interesse suscitado em torno das implementações distribuídas na área de controlo, cujas vantagens já foram referidas, levou ao longo das últimas décadas ao aparecimento de um enorme número de protocolos de comunicação. Os protocolos destinados a este tipo de aplicações designam-se barramentos de campo. Eles são utilizados no local das implementações para ligarem os vários dispositivos que constituem o sistema de controlo como os sensores, lógica programável, como os *Programmable Logic Controllers* (PLC), e os actuadores e utilizam a topologia de barramento.

Os barramentos de campo possuem características específicas que os distinguem dos protocolos utilizados em redes locais. Essas características são [45]: tratamento eficiente de mensagens curtas, suporte para tráfego periódico e aperiódico, tempo de resposta limitado, fiabilidade e custos reduzidos de instalação, de operação e de manutenção.

Actualmente utilizam-se inúmeros barramentos de campo em aplicações de controlo. Na impossibilidade de referir todos eles, referem-se apenas alguns dos mais relevantes como sejam os protocolos CAN (por exemplo automação industrial e de edifícios, indústria automóvel e aeronáutica), WorldFIP (por exemplo automação industrial) e PROFIBUS (por exemplo automação industrial e de edifícios). Destes o protocolo CAN será analisado com maior detalhe uma vez que é utilizado nas implementações apresentadas nos capítulos 4 e 5. Pela mesma razão será também analisado o protocolo FTT-CAN [44] e [45].

Algumas publicações apresentam resumos relativos ao desenvolvimento e aplicações dos barramentos de campo. Uns são de âmbito mais genérico como [52], [43], [42] ou [57] e outros de âmbito mais comparativo como [58] e [59].

Controller Area Network

O protocolo *Controller Area Network* (CAN) [2] foi desenvolvido durante a década de 80 do século passado pela *Robert Bosch GmbH* para aplicações de tempo-real tendo em vista a indústria automóvel. O objectivo era o de minimizar a quantidade e o custo das cablagens utilizadas nos veículos. Este protocolo foi rapidamente adoptado pela indústria automóvel e durante a década de 90 a sua utilização alargou-se a outras áreas como, por exemplo, o controlo industrial. O interesse e aceitação conseguidos pelo protocolo CAN levaram ao desenvolvimento do *standard* internacional ISO 11898-2 para aplicações de velocidade elevada (1 Mbps) [60] e ISO 11519-2 para aplicações de velocidade reduzida (125 kbps) [61].

O protocolo CAN é um protocolo série que segue o paradigma produtor-consumidor [43] e apresenta um elevado desempenho e robustez. A sua implementação consegue-se com custos reduzidos e é permitida a utilização de vários suportes físicos. Normalmente é implementado recorrendo a um par de cabos entrelaçados.

A camada física define dois níveis lógicos complementares designados “recessivo” e “dominante” para os sinais no barramento. A transmissão simultânea de um *bit* “recessivo” e de um *bit* “dominante” deverá resultar num *bit* “dominante” (implementação do tipo *wired AND*). A codificação dos *bits* é feita utilizando o método NRZ (*Non-Return-to-Zero*), em que o nível eléctrico do *bit* é mantido constante durante todo o tempo de transmissão do mesmo.

É definida a taxa nominal de transmissão no barramento como sendo o número de *bits* por segundo transmitidos na ausência de re-sincronização por um transmissor ideal. O tempo nominal de *bit* corresponde ao inverso da taxa nominal de transmissão e é programado pelo utilizador. Este tempo está dividido em quatro segmentos temporais diferentes: sincronização, propagação, memorização de fase 1 (*phase buffer 1*) e memorização de fase 2 (*phase buffer 2*). A divisão do tempo nominal de *bit* nos diversos segmentos é apresentada na figura 2.9.

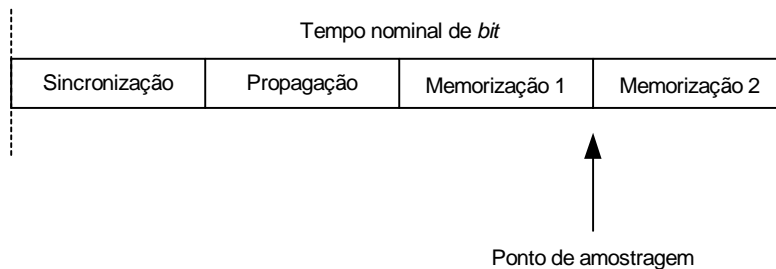


Figura 2.9: Divisão do tempo nominal de *bit* em segmentos.

O segmento de sincronização é utilizado para sincronizar os vários nós do barramento. Durante este segmento é esperada a ocorrência de transições. O segmento de propagação é utilizado para compensar os atrasos físicos dentro do barramento. Os segmentos de memorização de fase são utilizados para compensar os erros de fase das transições. Estes dois últimos segmentos podem ser aumentados ou diminuídos para re-sincronização. O ponto de amostragem é o ponto em que o nível eléctrico do barramento é lido e interpretado como o valor do *bit* correspondente. Este ponto localiza-se no fim do primeiro segmento de memorização de fase. O comprimento de cada um dos segmentos temporais é programável. A programação da temporização do *bit* é definida, dentro dos limites impostos pelo protocolo CAN, pelos utilizadores e depende do período do oscilador utilizado no circuito e do meio de transmissão escolhido para implementar o barramento, dando origem a diferentes taxas de transmissão na rede. O protocolo suporta taxas de transmissão desde cerca de 1 kbps até um máximo de cerca de 1 Mbps. O comprimento máximo depende da taxa de transmissão pretendida, por exemplo 1300 m a 50 kbps ou 40 m a 1 Mbps.

A sincronização pode ser de dois tipos: sincronização total e re-sincronização. A sincronização total força a transição que lhe deu origem a ficar dentro do segmento de sincronização. A re-sincronização corresponde ao processo através do qual o primeiro segmento de memorização de fase pode ser aumentado ou o segundo segmento de

memorização diminuído para compensar os desvios de fase detectados.

Embora o número máximo de nós ligados à rede CAN não seja limitado no protocolo, na prática a limitação é imposta pelo meio de transmissão escolhido e pelas suas características eléctricas.

O protocolo permite a qualquer nó iniciar a transmissão de uma mensagem quando detecta que o barramento está livre. Esta característica permite designar este protocolo como *multimaster*.

Cada mensagem que transita no barramento possui um identificador que define o significado da informação contida na mensagem e não o seu destino. É esta orientação produtor–consumidor que permite uma enorme flexibilidade na configuração dos sistemas baseados em CAN.

A filtragem de mensagens em cada nó receptor é feita tendo por base os identificadores de cada mensagem. Como este identificador está associado ao significado da informação contida na mensagem todos os nós sabem à partida as mensagens de que precisam para operarem correctamente. Nas implementações em circuito integrado do protocolo CAN, os controladores CAN, são classificados em duas classes de acordo com as suas capacidades de filtragem de mensagens: *fullCAN* e *basicCAN* [47]. Nos controladores *fullCAN* a filtragem de mensagens é efectuada pelo controlador. A aplicação só é notificada se a mensagem recebida for de interesse para ela. No caso dos controladores *basicCAN* a filtragem é efectuada pela aplicação (em *software*).

Na maioria dos casos a consistência da informação é mantida em toda a rede sendo cada mensagem simultaneamente aceite por todos os nós ou, alternativamente, não sendo aceite por nenhum nó. No entanto podem existir cenários específicos de inconsistência associados à existência de um erro no último *bit* da trama CAN [62]. A probabilidade de ocorrência destes cenários depende da taxa de erro de *bit* na rede CAN, podendo ou não constituir um problema nas aplicações críticas em termos de segurança, dependendo de cada implementação específica da rede CAN [63]. Caso seja necessário, há técnicas que permitem garantir a consistência da entrega de mensagens [62] e [64].

As prioridades no acesso à rede são estáticas e definidas pelo valor do identificador da mensagem. O MAC utilizado é do tipo CSMA/NBA e a arbitragem é feita com base no valor do identificador de cada mensagem. A figura 2.10 apresenta um exemplo do processo de arbitragem em que se consideram três nós a enviar simultaneamente mensagens para a rede. Deve considerar-se o nível eléctrico mais baixo como sendo o nível “dominante”.

Quando o barramento está livre os nós podem começar a transmitir as suas mensagens monitorizando simultaneamente o barramento. Quando um nó envia um *bit* e detecta no barramento um *bit* de nível igual pode continuar a sua transmissão, se o nível do *bit* enviado for “recessivo” e o nível monitorizado na rede for “dominante” o nó em causa perde a arbitragem e automaticamente deixa de enviar mais *bits* para o barramento. Este método permite que não seja perdido nem tempo nem informação durante o processo de arbitragem.

Todos os nós sinalizam através da rede uma mensagem correctamente recebida, através do envio de um *bit* “dominante” no segmento de trama com esse fim.

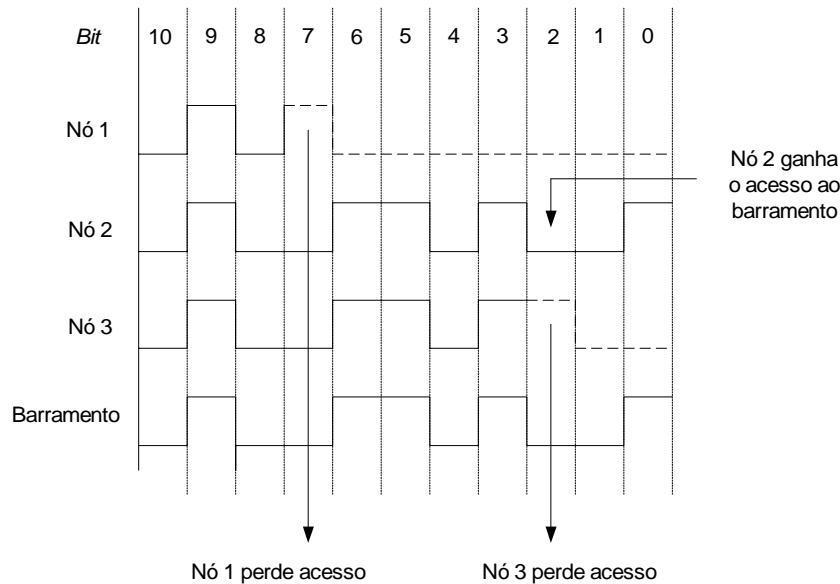


Figura 2.10: Exemplo do processo de arbitragem CSMA/NBA.

Existem quatro tipos diferentes de mensagens numa rede CAN: trama de dados, trama remota, trama de erro e trama de sobrecarga (*overload*). A trama de dados serve para o envio de dados, a trama remota é utilizada para fazer o pedido de dados com o mesmo identificador da trama enviada, a trama de erro informa a detecção da ocorrência de um erro e a trama de sobrecarga constitui um pedido de atraso da próxima transmissão de tramas de dados ou tramas remotas.

A trama de dados é constituída por sete segmentos: início de trama, campo de arbitragem, campo de controlo, campo de dados, campo CRC (*Cyclic Redundancy Check*), campo de confirmação e fim de trama. A figura 2.11 apresenta a formatação de uma trama de dados.

O segmento início de trama é constituído por um *bit* “dominante”. O campo de arbitragem contém o identificador de mensagem e o *bit* RTR (*Remote Transfer Request*). Nas tramas de dados o *bit* RTR é transmitido como “dominante”. O identificador poderá ter 11 *bits*, formato *standard*, correspondente à especificação 2.0A, ou 29 *bits*, formato estendido, correspondente à especificação 2.0B. Neste último formato o campo de arbitragem inclui ainda o *bit* IDE (*IDentifier Extension*) e o *bit* SRR (*Substitute Remote Request*), transmitidos como *bits* “recessivos”. O campo de controlo é composto por seis *bits*, quatro dos quais são utilizados para codificar o tamanho da mensagem, que pode variar entre zero e oito *bytes*, designados por DLC (*Data Length Code*), e dois *bits* reservados cujo significado difere de acordo com o tipo de formato. No formato *standard* os dois *bits* restantes são designados por IDE e r0. O *bit* IDE é enviado “dominante” significando o fim do identificador. O *bit* r0 está reservado. No formato estendido os dois *bits* r0 e r1 estão reservados. Os formatos *standard* e estendido, referentes aos campos de arbitragem e de controlo, são apresentados, respectivamente, nas figuras 2.12 e 2.13.

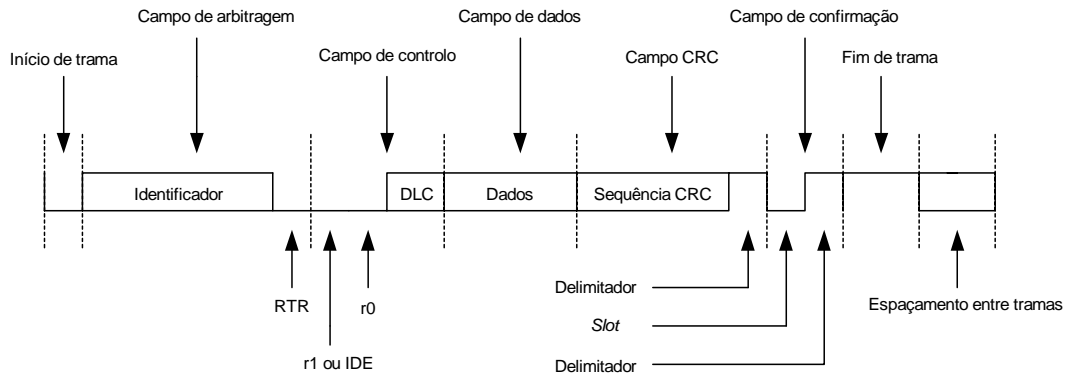
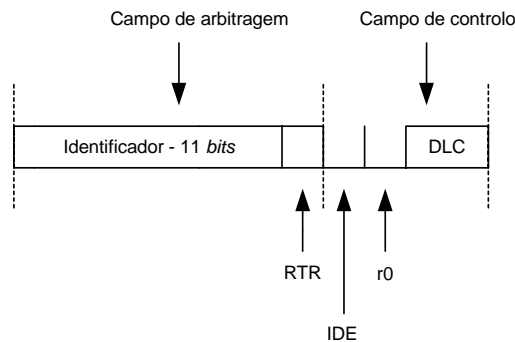


Figura 2.11: Formato da trama de dados

O campo CRC é constituído por uma sequência de quinze *bits* calculada de acordo com as regras definidas pelo protocolo e por um *bit* delimitador “recessivo”. O campo de confirmação tem dois *bits*. Um designado por *slot* de confirmação e outro por delimitador de confirmação (*bit* “recessivo”). Um nó transmissor envia sempre os dois *bits* do campo de confirmação como *bits* “recessivos”. Os nós receptores quando recebem uma mensagem correctamente, sinalizam o nó transmissor enviando um *bit* “dominante” durante o *slot* de confirmação. O fim de trama é composto por sete *bits* “recessivos”.

Figura 2.12: Campos de arbitragem e controlo, formato 2.0A – *standard*.

As tramas remotas são constituídas por seis segmentos: início de trama, campo de arbitragem, campo de controlo, campo CRC, campo de confirmação e fim de trama. Nas tramas remotas o *bit* RTR do campo de arbitragem é “recessivo”, por isso quando é enviada simultaneamente uma trama remota a pedir determinada informação e a trama de dados equivalente, a trama de dados ganha o processo de arbitragem, ou seja o acesso ao barramento. Os *bits* DLC do campo de controlo podem, nas tramas remotas, apresentar qualquer valor dentro da gama possível.

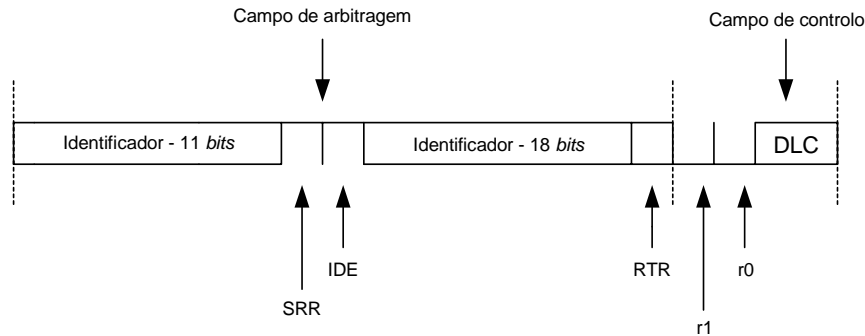


Figura 2.13: Campos de arbitragem e controlo, formato 2.0A – estendido.

Alguns dos segmentos de trama são codificados utilizando o método de *bit stuffing*, que consiste na inclusão forçada de um *bit* de polaridade contrária depois da ocorrência de cinco *bits* consecutivos da mesma polaridade. O objectivo deste método de codificação é permitir que os nós receptores mantenham o sincronismo com o nó emissor. Os segmentos de trama codificados recorrendo ao método de *bit stuffing* são: início de trama, campo de arbitragem, campo de controlo, campo de dados e sequência CRC.

As tramas de erro são constituídas por dois segmentos: *flag* e delimitador de erro. A *flag* de erro pode ser activa ou passiva dependendo do estatuto do nó transmissor. No caso de ser activa são enviados seis *bits* “dominantes”, se for passiva são enviados seis *bits* “recessivos”. O formato destas tramas, constituídas por seis *bits* sucessivos da mesma polaridade viola a regra de *bit stuffing* fazendo com que todos os nós da rede se apercebam do erro e rejeitem a mensagem que estava a ser transmitida.

As tramas de sobrecarga são constituídas por dois segmentos: *flag* de sobrecarga e delimitador de sobrecarga. A *flag* de sobrecarga é composta por seis *bits* “dominantes” e o delimitador de sobrecarga por oito *bits* “recessivos”. No máximo podem ser geradas duas tramas de sobrecarga para atrasar o envio da próxima trama de dados ou trama remota.

As tramas de dados e as tramas remotas são separadas das tramas precedentes por um formatador designado por espaçamento entre tramas. As tramas de erro e de sobrecarga não são precedidas deste formatador. Este campo é constituído por dois segmentos: *intermission* e *bus idle*. A *intermission* é constituída por três *bits* “recessivos”. Durante o envio deste segmento só pode ser iniciada a transmissão de uma trama de sobrecarga e não de uma trama de dados ou remota. O segmento *bus idle* é de comprimento arbitrário e sinaliza que o barramento está livre. Qualquer estação pode iniciar a transmissão de dados assim que detecte a condição *bus idle*. A detecção de um *bit* “dominante” durante *bus idle* é interpretada como o início de trama. Existe ainda um formatador designado suspensão de transmissão, constituído por oito *bits* “recessivos” que é enviado por todos os nós passivos depois de terem sido transmissores.

O protocolo CAN é extremamente robusto. Esta robustez tem por base as capacidades de detecção e limitação de erros definidas pelo protocolo. Existem cinco tipos

de erros que podem ocorrer durante a transmissão de mensagens: erro de *bit*, erro de *stuff*, erro de CRC, erro de forma e erro de confirmação. A detecção do erro de *bit* é feita por cada nó através de monitorização directa do barramento. Se o nó envia um *bit* de determinada polaridade e detecta na rede um *bit* de polaridade diferente detecta o erro. Como excepção a esta regra temos o envio de um *bit* “recessivo” e a monitorização de um *bit* “dominate” durante o campo de arbitragem ou o *slot* de confirmação, de acordo com as características referidas do método de arbitragem e do método de confirmação. O erro de *stuff* é detectado quando se verifica a ocorrência na rede do sexto *bit* consecutivo da mesma polaridade. O erro de CRC é detectado quando a sequência CRC calculada pelo nó receptor é diferente daquela enviada pelo nó emissor. Esta diferença indica a possibilidade de corrupção dos dados enviados. O erro de forma é detectado nos campos de trama constituídos por níveis fixos quando é detectada a ocorrência de polaridades ilegais para esses *bits*. Finalmente o erro de confirmação é detectado pelo nó emissor quando não monitoriza na rede nenhum *bit* “dominante” durante o *slot* de confirmação, indicando que nenhum dos nós receptores recebeu correctamente a mensagem.

A sinalização de erro é feita por qualquer nó através do envio de uma trama de erro. Quando uma trama de erro é detectada pelo nó transmissor, a transmissão é abortada e a retransmissão é automática logo que seja possível. O tempo de recuperação é de no máximo 29 vezes o tempo nominal de *bit*. Em relação à ocorrência de erros um nó pode ter um de três estatutos possíveis: activo, passivo e desligado. Um nó activo envia *flags* de erro activas, um nó passivo envia *flags* de erro passivas e quando termina uma transmissão suspende a transmissão seguinte enviando oito *bits* “recessivos”. Um nó desligado não tem influência activa no funcionamento da rede. O estatuto de cada nó é controlado através de contadores de erro, um de transmissão e outro de recepção, designados TEC (*Transmission Error Count*) e REC (*Receive Error Count*), respectivamente. Existem regras bem definidas de incremento e decremento dos contadores de acordo com a detecção ou não de erros para cada situação, recepção ou transmissão. A passagem entre estatutos em cada nó processa-se de acordo com o diagrama da figura 2.14, tendo em conta o valor dos contadores de erro referidos.

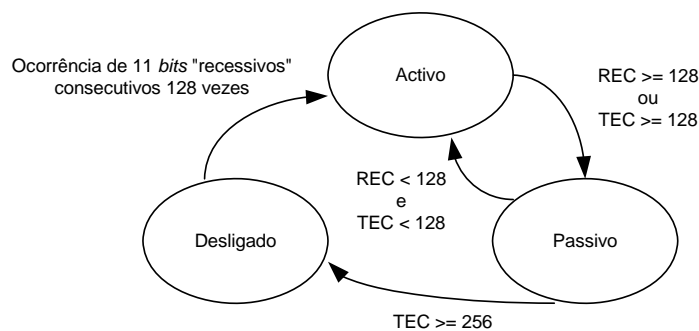


Figura 2.14: Diagrama das transições de estatuto num nó CAN.

WorldFIP

O protocolo WorldFIP é baseado no paradigma de comunicação produtor-distribuidor-consumidor. Este protocolo está definido sob a forma dos *standards* europeu EN50170 [55] e internacional IEC61158 [65].

De acordo com o paradigma PDC as variáveis de interesse para o sistema são disponibilizadas pelos nós que as produzem, uma de cada vez, para serem consumidas pelos nós consumidores. A distribuição das mensagens produzidas é controlada por um distribuidor que solicita a informação desejada. A função de distribuição é efectuada de modo centralizado por um nó designado árbitro (*bus arbitrator*) que faz o escalonamento do acesso dos produtores ao barramento.

A transmissão faz-se normalmente recorrendo a um par entrelaçado blindado. Existem três perfis definidos para a taxa de transmissão: o perfil *standard* a 1 Mbps (perfil S2) e dois perfis especiais a 31.25 kbps (perfil S1) e a 2.5 Mbps (perfil S3). Existe ainda um perfil adicional a 5 Mbps para utilização sobre fibra óptica. Na versão *standard* a 1 Mbps o barramento pode ter até 500 m. O tamanho pode ser aumentado até ao máximo de 2000 m interligando até 4 redes através de repetidores. O número máximo de nós por rede é de 32.

A codificação dos bits é feita de acordo com o código de Manchester que codifica um “1” como uma transição do nível lógico alto para o baixo e um “0” como uma transição do nível lógico baixo para o alto. Este tipo de codificação tem a vantagem de permitir a transmissão do relógio junto com a informação.

As tramas são compostas por três campos como se pode ver na figura 2.15: um campo de início de trama FSS (*Frame Start Sequence*), um campo de controlo e dados CAD (*Control And Data*) e um campo de fim de trama FES (*Frame End Sequence*).

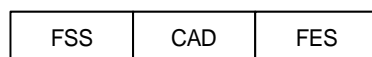


Figura 2.15: Formato de uma trama WorldFIP.

O campo de início de trama é composto por um preâmbulo e por um delimitador ambos de valor fixo. Este campo serve para fazer a sincronização dos nós. O campo de controlo e dados serve para o envio da informação útil para a aplicação. Ele é composto por uma zona para envio de informação de controlo, outra para dados e por uma zona para controlo de erros designada FCS (*Frame Check Sequence*). A quantidade de dados por trama pode ir até 128 *bytes*. Finalmente o campo de fim de trama é composto por um valor fixo e serve para identificar o fim da trama.

O protocolo fornece dois tipos de transmissões: troca de variáveis identificadas e troca de mensagens. Estas trocas de informação podem ocorrer de modo cíclico ou a pedido do utilizador. A informação sobre as variáveis que devem ser processadas de modo cíclico é utilizada para construir uma tabela. É com base nessa tabela que o árbitro faz o escalonamento do acesso ao barramento. Esta tabela estática é normalmente construída *off-line*.

Também é possível a existência de variáveis aperiódicas. Neste caso o nó que deseja enviar a variável informa o árbitro dessa necessidade quando lhe é concedido o acesso ao barramento. O árbitro faz depois o escalonamento do tráfego aperiódico utilizando a largura de banda que fica disponível depois de efectuado o escalonamento do tráfego periódico [45].

O serviço de transmissão de mensagens existe para comportar o tráfego que não é crítico em termos temporais. Este tipo de transmissão pode ser efectuado com ou sem confirmação e só é escalonado depois das transmissões de variáveis periódicas e aperiódicas.

Existem dois modos de endereçamento: um esquema global em que cada variável do sistema tem associado um identificador de 16 *bits* e um esquema local para as mensagens que utiliza 24 *bits* para identificar diferentes segmentos da rede e os respectivos dispositivos.

A interface entre a camada de ligação de dados e a camada física ao nível de cada nó é efectuada através de um conjunto de *buffers* de consumo e de produção. Esses *buffers* contêm o valor mais recente das variáveis e são actualizados pelo utilizador (*buffer* de produção) ou pela rede (*buffer* de consumo). A actualização dos *buffers* de consumo é efectuada por um serviço da rede designado *buffer transfer*. Este tipo de transferência é implementado sob a forma de uma transacção atómica, ou seja que não pode ser efectuada parcialmente, constituída pelo envio de uma trama de identificação (ID_DAT) seguida do envio da trama de resposta (RP_DAT) respectiva. A trama ID_DAT é enviada pelo árbitro e identifica a variável a ser produzida. A trama RP_DAT é enviada em resposta pelo nó produtor da variável. A trama de resposta é recebida pelo nó ou nós consumidores e é efectuada a actualização dos *buffers* de consumo respectivos. O formato do campo CAD para as tramas de ID_DAT e RP_DAT é apresentado na figura 2.16.

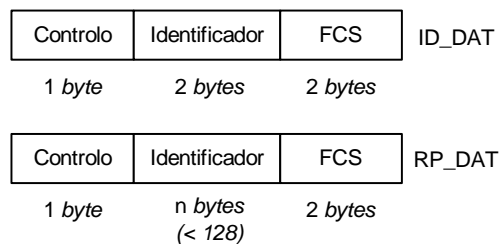


Figura 2.16: Formato do campo CAD das tramas ID_DAT e RP_DAT.

Este protocolo não inclui nenhum método de sinalização de erros.

PROFIBUS

O protocolo PROFIBUS foi desenvolvido para comunicação distribuída em ambiente industrial entre dispositivos de campo como sensores, válvulas, PLCs, etc. Este protocolo está definido sob a forma dos *standards* europeu EN50170 [55] e internacional IEC61158 [65].

O acesso ao barramento é efectuado através de um mecanismo híbrido que inclui a passagem de testemunho entre os nós *master* e a técnica *master-slave* centralizada entre cada *master* e os seus *slaves*.

Quando um nó *master* fica na posse do testemunho pode iniciar a troca de informação com outros nós (*master* ou *slave*). Cada nó possui um endereço único através do qual é endereçado. A gama de endereços vai de 0 a 126, pelo que uma rede pode no máximo ter 127 nós.

A passagem de testemunho entre os nós *master* faz-se de acordo com uma topologia de anel virtual em que a ordem é determinada pelo endereço dos nós. O tempo requerido para a rotação do testemunho entre todos os nós *master* designa-se por tempo de rotação do testemunho. O tempo máximo permitido para essa rotação é especificado.

A versão básica deste protocolo é suportada por um canal de transmissão implementado com recurso a um par entrelaçado blindado. A velocidade de transmissão varia entre 9.6 kbps e 12 Mbps sendo o comprimento máximo da rede correspondente de 1000 m e de 100 m, respectivamente [66]. Cada segmento de rede pode comportar um máximo de 32 nós e podem ligar-se vários segmentos entre si recorrendo a repetidores. Existem também especificações para implementações em fibra óptica. A codificação dos *bits* é feita utilizando o código NRZ.

A camada de ligação de dados, designada neste protocolo por *Fieldbus Data Link* (FDL), é responsável pelo controlo do acesso ao barramento, pela segurança dos dados e pelo processamento das mensagens. Existem diversos tipos de mensagens: mensagens de formato fixo com e sem campo de dados, mensagens com campo de dados de formato variável, mensagens de confirmação e a mensagem do testemunho. O formato das tramas correspondentes é apresentado na figura 2.17.

Existem mecanismos de detecção de erros no envio das mensagens. Se forem detetados erros as mensagens são automaticamente repetidas pelo menos uma vez.

Os serviços base da FDL comportam o envio/pedido de dados com confirmação e o envio de dados sem confirmação.

Para além das transferências de dados ponto a ponto também são permitidas transmissões para vários pontos do tipo *broadcast* ou *multicast*. No caso dos *broadcasts* a comunicação é efectuada do nó *master* para todos os outros nós. No caso dos *multicasts* a comunicação é efectuada do nó *master* para um grupo de estações (por exemplo todas as estações num dado segmento).

A comunicação entre um nó *master* e os *slaves* respectivos faz-se através do *polling* sucessivo de cada *slave*. Esse *polling* é feito de acordo com uma lista definida antecipadamente. Cada troca *master-slave* inicia-se com o envio de uma trama de pedido de dados, enviada pelo *master*, seguida da mensagem de resposta enviada pelo *slave*. Cada ciclo de acesso ao barramento comporta uma parte destinada ao tráfego periódico e outra destinada ao tráfego aperiódico.

FTT-CAN (*Flexible Time-Triggered on CAN*)

O protocolo FTT-CAN [44] e [45] combina flexibilidade de operação com comunicação do tipo *time-triggered*, permitindo a alteração das características de tráfego durante a operação do sistema sem com isso deixar de garantir os requisitos temporais impostos

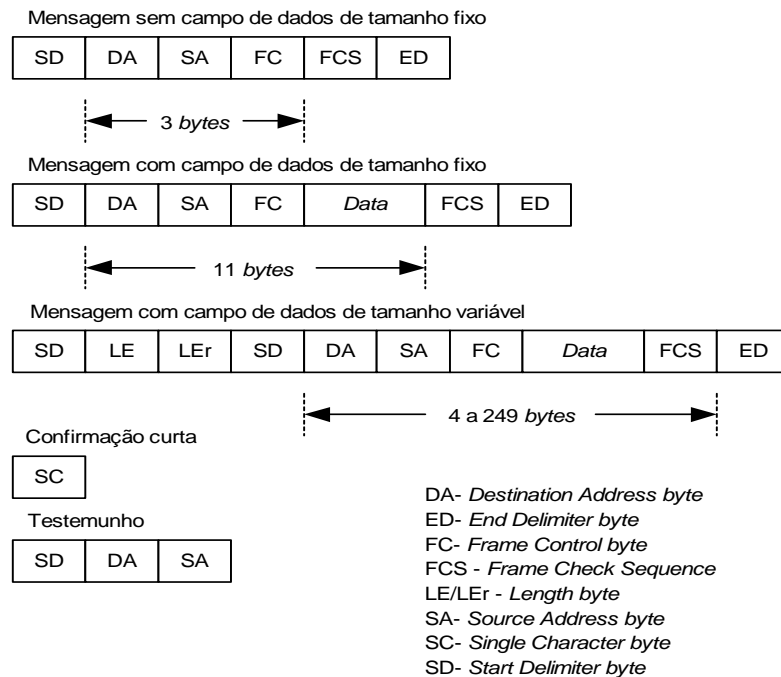


Figura 2.17: Formato das mensagens PROFIBUS.

pele tráfego de tempo-real. Este protocolo destina-se a aplicações de tempo-real implementadas com microcontroladores com baixa capacidade de processamento, típicas em sistemas embutidos distribuídos. Como o nome do protocolo indica ele é baseado em CAN.

No protocolo FTT-CAN o tempo de barramento é dividido em janelas de duração fixa designados *Elementary Cycles* (ECs). Todos os nós são sincronizados no início de cada EC através da recepção de uma mensagem especial designada *Trigger Message* (TM). Esta mensagem é enviada por um nó especial designado *master* e tem duração fixa.

Cada EC está dividido em duas janelas consecutivas designadas assíncrona e síncrona, que correspondem a duas fases distintas de acesso ao barramento. A janela assíncrona é utilizada para transmitir o tráfego *event-triggered*, normalmente aperiódico, que pode ser requisitado em qualquer altura. A janela síncrona é utilizada para transmitir o tráfego *time-triggered*. O tráfego síncrono é escalonado de modo centralizado no nó *master*. O tamanho da janela síncrona ($lsw(n)$) em cada EC é determinado em função do tráfego escalonado para esse EC e por isso varia de EC para EC. A informação sobre o tráfego escalonado para cada EC é enviada na respectiva TM. A duração da janela assíncrona ($law(n)$) corresponde ao tempo que medeia entre o final da transmissão da TM e o início da janela síncrona. A figura 2.18 apresenta a estrutura de um EC. As mensagens assíncronas são representadas por AM e as síncronas por SM.

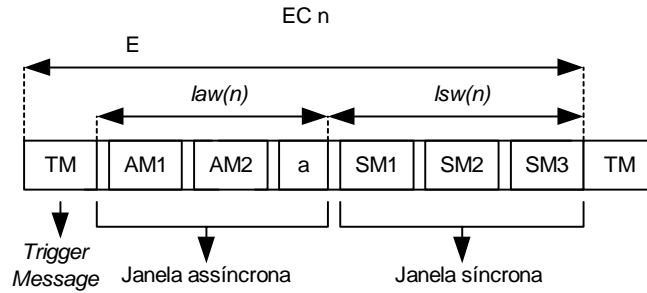


Figura 2.18: Estrutura de um *Elementary Cycle* em FTT-CAN.

O protocolo permite a definição de uma duração máxima para a janela síncrona (LSW) garantindo desse modo a existência de uma largura de banda mínima para a transmissão de tráfego assíncrono. Pode existir no final da janela assíncrona um certo tempo *idle* durante o qual o barramento não é utilizado (representado pela letra *a* na figura 2.18). Esse tempo é inserido pelo protocolo de modo a garantir que o tráfego assíncrono não põe em risco os requisitos temporais do tráfego síncrono.

As características temporais do tráfego síncrono são expressas em função da duração de um EC (*E* na figura). O tráfego síncrono é caracterizado pelo tamanho dos dados (DLC), o tempo máximo de transmissão (*C*), o período (*P*), o *deadline* (*D*) e a prioridade (*Pr*), quando necessário. Esta informação é guardada sob a forma de uma tabela designada *Synchronous Requirements Table* (SRT) no nó *master*. O escalonador constrói *online* o escalonamento para cada EC com base na tabela SRT. Esse escalonamento é enviado no campo de dados da TM para toda a rede (*broadcast*). Deste modo é garantida alguma flexibilidade, podendo ser alterado o conjunto ou as características do tráfego, durante a operação do sistema. É também permitida a utilização de qualquer política de escalonamento, RM, DM, EDF, etc.

A progressão do tempo é sinalizada pelo envio da TM. Os nós ao receberem a TM devem descodificar o seu conteúdo de modo a determinarem se são produtores de alguma das mensagens síncronas. Este protocolo utiliza por isso o paradigma produtor-consumidor. Cada nó possui uma tabela local com indicação das mensagens a produzir/consumir. Quando se inicia a janela síncrona todos os nós produtores escalonados iniciam a transmissão das suas mensagens sendo a arbitragem feita através do MAC do protocolo CAN (do tipo CSMA/NBA). O *overhead* introduzido pelo mecanismo centralizado de escalonamento é reduzido [49]. Foram desenvolvidos testes de escalonabilidade específicos para as implementações utilizando este protocolo [45]. Para a política de escalonamento RM o teste é dado pela condição

$$U = \sum_{i=1}^{N_s} \frac{C_i}{T_i} \leq N_s (2^{\frac{1}{N_s}} - 1) \left(\frac{LSW - X}{E} \right) \quad (2.7)$$

em que, para além dos símbolos já conhecidos, *X* representa o tempo *idle* máximo inserido na janela síncrona, que é dado pelo tempo de transmissão da maior mensagem síncrona.

Para a política de escalonamento EDF o teste é dado pela condição

$$U = \sum_{i=1}^{N_s} \frac{C_i}{P_i} \leq \frac{LSW - X}{E} \quad (2.8)$$

Para que seja sempre garantida a escalonabilidade do tráfego síncrono existe um mecanismo de controlo de admissão que analisa todas as alterações efectuadas à tabela SRT de modo a garantir que elas são praticáveis. O controlador de admissão utiliza os testes de escalonabilidade descritos para fazer essa análise e só são admitidas as alterações que garantam o cumprimento dos requisitos temporais necessários. Este mecanismo é implementado no nó *master*.

O tráfego assíncrono é transmitido no barramento durante a janela assíncrona. Os nós que desejem enviar este tipo de tráfego iniciam a sua transmissão no início da janela assíncrona e abortam a transmissão no final da mesma. Apenas as mensagens que não tenham sido transmitidas são de facto abortadas. Neste caso a fila de espera não é modificada, sendo efectuada nova transmissão no ciclo seguinte. A arbitragem do barramento é feita utilizando o MAC do protocolo CAN. O escalonamento do tráfego assíncrono utiliza o paradigma do melhor-esforço de modo dinâmico [49]. Os métodos de análise da escalonabilidade nas condições de operação da janela assíncrona são apresentados em [45].

A TM pode também ser utilizada para fazer o *dispatching* de tarefas pendentes nos nós [67], esta funcionalidade é utilizada numa das implementações descritas no capítulo 4.

O protocolo FTT-CAN permite ainda a utilização de um mecanismo para gestão da qualidade de serviço (QoS) em conjunto com o controlador de admissões [45]. Para tal a SRT deve conter um campo em que seja possível exprimir a gama de níveis de QoS aceitáveis. O gestor de QoS é implementados no nó *master*. O utilizador pode assim definir as suas próprias políticas de QoS. Em [68] são apresentados exemplos de políticas de gestão baseadas em prioridades e no modelo elástico. Este mecanismo permite a utilização eficiente da largura de banda disponível no barramento assim como o tratamento de situações de possível sobrecarga do barramento.

2.4 Técnicas de controlo utilizadas em sistemas de controlo distribuído

Como referido no início deste capítulo a distribuição de um controlador por diferentes nós ligados através de uma rede de comunicação introduz uma série de desafios no seu projecto. Esta área de investigação é complexa devido ao seu contexto pluridisciplinar englobando conhecimentos nos domínios das redes, dos sistemas de computação e do controlo.

A utilização crescente de sistemas de controlo distribuído em diversos domínios de aplicação, devida às vantagens operacionais e de custos que oferecem, levou a que a investigação nesta área tivesse aumentado ao longo da última década. Esse interesse

levou nos últimos anos ao aparecimento de números especiais de revistas conceituadas do IEEE sobre estes temas [69], [70] e [71].

O problema sob análise é o problema de controlo em sistemas em que existe um atraso variável entre o momento da amostragem e o momento da actuação. No caso em estudo neste documento o atraso é provocado pela distribuição do controlador, no entanto convém referir que uma parte da investigação existente sob este tipo de problema se refere a sistemas em que o atraso é devido ao sistema de computação e em que o controlador é implementado de modo centralizado. Esses trabalhos não podem ser ignorados pois abordam um problema semelhante. No entanto nem todas as soluções são transponíveis para os sistemas de controlo distribuído.

2.4.1 Problemas associados à operação de sistemas de controlo distribuído

Um dos primeiros trabalhos a tratar os problemas temporais em sistemas de controlo distribuído foi o de Halevi e Ray [16] e [72] onde é analisado o efeito dos atrasos variáveis e da perda de amostras devidas à rede de comunicação no desempenho do controlador. São também sugeridos métodos para modelar esses atrasos e estratégias para reduzir ou eliminar as variações temporais. Wittenmark et al. em [73] descrevem para além dos problemas associados aos atrasos, os problemas associados à existência de *jitter* propondo modelos para sistemas com atrasos constantes e com atrasos variáveis. A modelação de atrasos variáveis pode ser difícil de fazer uma vez que o sistema que se obtém é variável no tempo e pode haver dificuldades na determinação das propriedades estatísticas dos elementos variáveis no tempo. No entanto ambos os trabalhos são unânimes em considerar que os atrasos variáveis diminuem o desempenho de controlo e podem causar a instabilização do sistema. Em [74] Thorngren apresenta um resumo do trabalho efectuado até à altura nesta área, chamando a atenção para o fosso existente entre os pressupostos feitos pela teoria de controlo “clássica” e as condições reais de operação de um sistema de controlo distribuído. É referido também que a sensibilidade dos sistemas de controlo realimentado em relação a atrasos variáveis ou a atrasos constantes diferentes dos atrasos nominais pode variar substancialmente e que depende da largura de banda do sistema controlado e da relação entre o atraso e o período de amostragem.

Em [75] Sanfridson apresenta um estudo exaustivo dos problemas temporais associados aos sistemas de controlo distribuído onde para além de referências ao atraso de controlo e ao *jitter* surgem também referências à importância do período de controlo e dos erros transitórios no desempenho do sistema.

Lian [3] apresenta uma caracterização dos atrasos provocados por três tipos de protocolos (Ethernet, ControlNet e DeviceNet) e estuda os compromissos entre a largura de banda da rede e as taxas de amostragem dos controladores a que é necessário chegar.

Tipsuwan e Chow em [17] também apresentam exemplos de degradação do controlo e instabilização do sistema em consequência dos atrasos introduzidos na malha de controlo, assim como um resumo das técnicas de controlo desenvolvidas para estes sistemas.

Mais recentemente Jianyong et al. [8] apresentam um resumo dos factores que mais afectam o desempenho dos sistemas de controlo distribuído, como sejam o atraso introduzido pela rede, o período de amostragem, o *jitter*, a perda de pacotes e o escalonamento.

Ao longo do tempo tem sido reconhecida a dificuldade na modelação deste tipo de sistemas [73], [74] e [8] porque o facto de existirem atrasos variáveis na malha de controlo pode tornar o sistema variante no tempo e por isso as ferramentas de análise convencionais deixam de ser aplicáveis.

O modo como os atrasos introduzidos na malha de controlo afectam o desempenho do sistema também não é fácil de determinar *a priori* como o demonstram alguns resultados relatados em [76] e [77]. Do ponto de vista do controlo um sistema com atrasos variáveis deixa de ser invariante no tempo [15] e por isso a sua análise torna-se difícil.

Para facilitar a análise de sistemas de controlo sujeitos a atrasos surgiram duas ferramentas baseadas em MATLAB designadas TrueTime [78] e [79] e Jitterbug [78] e [80]. A ferramenta Jitterbug facilita o cálculo de um critério de desempenho quadrático para sistemas de controlo lineares sob condições temporais variáveis (como atraso e *jitter*). Esta ferramenta pressupõe o conhecimento do período de amostragem e da distribuição do atraso. A ferramenta TrueTime facilita a simulação de sistemas de tempo-real. Ela permite simular o comportamento temporal de um *kernel* de tempo-real multitarefa durante a execução de tarefas de controlo. Também permite a simulação de modelos simples de redes de comunicação. Por isso esta ferramenta é adequada para a simulação de sistemas de controlo distribuído. Esta ferramenta foi utilizada nos testes apresentados nos capítulos 4 e 5.

Como refere Liu [81] as estratégias utilizadas no projecto e análise de sistemas de controlo distribuído são de três tipos: métodos de escalonamento que garantem qualidade de serviço (QoS) na rede, métodos de controlo que garantem a qualidade de controlo do sistema (QoC) e métodos integrados de controlo e escalonamento que consideram tanto a qualidade de serviço como a qualidade de controlo. Neste resumo serão apenas considerados os dois últimos tipos de métodos, visto que a ênfase é colocada no controlo.

2.4.2 Métodos para o projecto do controlador

Halevi e Ray [16] propõem no trabalho já referido uma metodologia baseada nos modelos discretos habituais mas com o espaço de estados aumentado através da utilização de um novo estado com informação do atraso. Este método é utilizado para o controlo de um sistema linear através de uma rede de comunicação mas considera-se que essa rede introduz apenas atrasos periódicos.

Outro método surgido sensivelmente na mesma altura propõe que os atrasos aleatórios sejam eliminados através de técnicas de sincronização e *buffering* [82]. Em [83] e [84] Luck e Ray propõem um modelo de *queuing* que utiliza um observador para implementar uma técnica de controlo preditivo baseada em valores passados da saída. O observador estima o estado actual do sistema com base nos valores passados da saída

e com base nessa informação o preditor prevê o estado seguinte do sistema e o controlador determina o valor de controlo a partir do estado previsto. Esta técnica requer a utilização de um modelo dinâmico muito preciso do sistema a controlar. Como é sabido em geral isso é difícil e por isso este tipo de método foi abandonado [11].

Thorngren [85] e Shin e Cui [86] consideram o atraso como uma perturbação externa introduzida no sistema e projectam os controladores com base nesse pressuposto.

Nilsson [15] propõe a utilização de um controlador óptimo do tipo LQG para controlar um sistema distribuído sujeito a atrasos aleatórios. O atraso aleatório é modelado num dos casos considerando de modo independente o atraso entre o sensor e o controlador e o atraso entre o controlador e o actuador, e no outro caso determinando a distribuição de probabilidades do atraso através de uma cadeia de Markov. Esta técnica tem a desvantagem de necessitar da medição do atraso.

Goktas [87] propõe a utilização de técnicas de controlo robusto no projecto de sistemas de controlo distribuído. Estas técnicas têm a desvantagem de necessitarem do conhecimento da distribuição probabilística dos atrasos introduzidos na malha.

As técnicas de controlo difuso também têm sido aplicadas neste contexto. Almutairi et al. em [88] utilizam um controlador PI e um bloco não-linear baseado em lógica difusa. A modulação actua como um factor multiplicativo aplicado à saída do controlador PI. No entanto a modulação é baseada apenas no sinal de erro e no sinal de controlo do controlador PI e não é claro se a compensação é feita para obviar o efeito do atraso na malha de controlo ou devido à necessidade de melhorar o desempenho do controlador PI. Em Lee [89] é apresentado um controlador difuso para compensar o atraso introduzido na malha de controlo num sistema implementado com o barramento Profibus-DP.

Liu et al. [81] apresentam um esquema de controlo baseado num preditor de controlo, que é utilizado para gerar um conjunto de valores de controlo futuros, e de um compensador para o atraso da rede que é utilizado para compensar os atrasos introduzidos. Esta técnica obriga a que as mensagens enviadas contenham informação do instante temporal em que foram geradas (*time-stamping*).

Hu et al. [6] propõem um modelo para os sistemas de controlo distribuído em que o atraso em vez de ser modelado no domínio discreto é modelado no domínio contínuo. A partir do modelo proposto é depois sintetizado um controlador realimentado.

2.4.3 Métodos integrados para o controlo e o escalonamento do sistema

Os métodos de projecto integrado do controlador e do escalonador são muitas vezes designados por *co-design*.

Hong [90] apresenta um algoritmo para determinar o período de amostragem de um conjunto de controladores distribuídos garantindo em simultâneo que os requisitos de desempenho de cada malha são satisfeitos.

Seto e al. [91] propõem uma metodologia semelhante. Neste caso os períodos das tarefas podem variar dentro de uma gama de valores que garanta o desempenho mínimo desejado para os controladores. É apresentado um algoritmo que optimiza os períodos

a utilizar para as tarefas tendo por objectivo obter o melhor desempenho de controlo possível. Este trabalho é depois desenvolvido em [92] onde o método é utilizado *online*, podendo ser aplicado para a adaptação do sistema a novas condições de funcionamento e para a recuperação de falhas. Estes dois trabalhos referem-se a implementações centralizadas do controlador.

Também Martí [4] e Cervin [5] adoptam uma estratégia de projecto integrado para o controlador e para o escalonador. Estes trabalhos referem-se ambos a sistemas de controlo sujeitos a atrasos e *jitter* em que a implementação do controlador é centralizada, ou seja, os atrasos são devidos ao sistema de computação.

Em [4] o modelo proposto para as tarefas a escalonar é flexibilizado sendo permitida a especificação de uma gama de valores para o período de amostragem e para o atraso. A análise de escalonabilidade é efectuada *off-line* de modo a determinar os intervalos previstos para o momento de amostragem e o atraso previsto em cada iteração entre o momento de amostragem e o de actuação. Para cada par de valores (tempo de amostragem, atraso) é projectado um controlador. A cada par é posteriormente associado um parâmetro relativo ao desempenho do controlador correspondente. O escalonador pode assim escolher durante a operação do sistema qual o controlador mais adequado tendo também em conta o desempenho de controlo. Esta técnica foi designada escalonamento QoC (*Quality Of Control*). Os controladores apresentados nos exemplos são do tipo PID e por posicionamento de pólos “clássico”. Na mesma linha de investigação Buttazzo et al. [93] propõem um método para reduzir o número de controladores a projectar *off-line* garantindo no entanto um dado nível de desempenho de controlo quando se altera o período da tarefa.

Cervin [5] propõe um mecanismo de realimentação que permite que o escalonador possa alterar, com base na medição do tempo de execução das tarefas, o período de amostragem das tarefas de modo a permitir maior flexibilidade geral visto que o escalonamento deixa de ser efectuada com base na situação de pior-caso. Também é permitido que as tarefas de controlo informem previamente o escalonador das alterações do modo de funcionamento a efectuar para que o escalonador possa recalcular os períodos de amostragem do conjunto de tarefas. É ainda proposto um novo modelo para as tarefas de controlo de tempo-real designado *control server* que determina que seja utilizado um único parâmetro ajustável, o factor de utilização da UCP para determinar o desempenho de controlo e a escalonabilidade da tarefa. Esta estratégia recorre à técnica de escalonamento baseada em servidores *constant-bandwidth server* para garantir que cada tarefa não utiliza mais do que uma dada quota da UCP. O mecanismo de realimentação referido pode ser utilizado com este modelo de tarefa de modo a permitir alterações na quota utilizada por cada controlador, de acordo com as suas exigências em termos de controlo.

Henriksson e Cervin [94] definem expressões para a determinação dos tempos de amostragem de vários controladores que partilham o mesmo processador de modo a otimizar o desempenho de controlo em face da limitação de recursos, neste caso o processador. Consideram no entanto que o tempo entre a amostragem e a actuação deve ser nulo ou constante, o que é uma limitação.

Castane et al. [95] propõem uma estratégia de gestão de recursos para tarefas

de controlo que maximiza o desempenho de controlo dentro dos limites dos recursos existentes através do reajuste dos períodos das tarefas em *run-time*. É utilizado um escalonador com realimentação do mesmo tipo do apresentado em [94] para determinar o valor óptimo dos períodos das tarefas. Este método considera a resposta dos sistemas sob controlo durante um horizonte de tempo finito de modo a determinar quando é que os períodos da tarefa podem ser alterados, tendo por base a ideia de que numa fase transitória os períodos de amostragem têm que ser menores do que quando o sistema está num estado estacionário, para se obter uma boa qualidade de controlo.

2.5 Abordagens propostas nesta tese

Neste trabalho propõem-se dois tipos de abordagens: uma para o projecto de raiz de controladores distribuídos e outra para a compensação do efeito do atraso introduzido na malha em sistemas já existentes em que esse factor não foi tido em conta.

A primeira abordagem baseia-se na modelação do atraso entre a amostragem e a actuação como tempo morto fraccionário nos modelos utilizados para a identificação do sistema e na sua aplicação ao projecto de controladores adaptativos distribuídos do tipo auto-ajustável, utilizando a técnica de posicionamento de pólos para a função de controlo. Esta abordagem tem a vantagem de não necessitar do conhecimento do atraso que afecta a malha em cada iteração. Ela permite obter um bom desempenho de controlo por comparação com as técnicas “clássicas” especialmente nos casos em que quando se utilizam estas últimas o atraso provoca oscilações ou instabilidade na malha de controlo. Esta abordagem será tratada a nível teórico no capítulo 3 e a nível prático no capítulo 4.

A segunda abordagem consiste na aplicação a um sistema de controlo já existente e que não tem em conta o efeito do atraso entre a amostragem e a actuação de um bloco de compensação com a função exclusiva de compensar o efeito do atraso sobre o sistema de controlo. O bloco de compensação actua com base no valor do atraso e pode ter, para além dessa, qualquer outra entrada conveniente. São propostas técnicas não-lineares para a implementação do compensador como a lógica difusa e as redes neuronais. Esta abordagem requer o conhecimento do atraso que afecta a malha de controlo a cada iteração pois o princípio de compensação é baseado nesse conhecimento. A técnica de compensação do atraso será apresentada no capítulo 5.

2.6 Conclusão

Neste capítulo apresentaram-se as características dos sistemas de controlo distribuído, os conceitos associados aos sistemas de tempo-real como as técnicas de escalonamento, os conceitos associados aos sistemas de tempo-real distribuídos, como a necessidade de comunicações em tempo-real, os tipos de controlo de acesso ao meio e os barramentos de campo. Apresentou-se ainda um resumo das técnicas de controlo utilizadas até aqui em sistemas de controlo distribuído. Por fim descreveram-se as abordagens propostas neste trabalho.

Capítulo 3

Modelação e controlo adaptativo distribuído de sistemas lineares

O controlo digital surge com o aparecimento dos primeiros sistemas de computação e utiliza um computador para a implementação do controlador.

Anteriormente os controladores eram implementados recorrendo a componentes analógicos (resistências, condensadores, amplificadores operacionais, etc) que para além de serem dispendiosos, tinham um ciclo de vida curto. A popularização do controlo digital dá-se à medida que o avanço tecnológico permite a diminuição do custo dos computadores.

Os controladores digitais apresentam uma série de vantagens face aos controladores analógicos das quais as mais significativas são a maior flexibilidade e fiabilidade e o custo mais reduzido. Os controladores digitais permitem a implementação de técnicas de controlo mais poderosas do que as utilizadas em controlo analógico. Essas técnicas surgem para aproveitar as potencialidades do novo método de implementação. É no âmbito do controlo digital que surge o controlo adaptativo.

A definição de controlador adaptativo não foi durante muito tempo consensual. Pode no entanto dizer-se genericamente que um controlador adaptativo é um controlador que possui parâmetros ajustáveis e um mecanismo que permite o ajuste desses parâmetros [96]. Existem vários tipos de controladores adaptativos alguns dos quais se baseiam na identificação, em tempo-real, das características do sistema a controlar. Por isso as técnicas de identificação e modelação de sistemas são também importantes no contexto deste trabalho.

Neste capítulo descrevem-se: as principais características dos controladores digitais; a amostragem de sistemas contínuos; a modelação e identificação de sistemas; as técnicas mais comuns de controlo adaptativo centralizado; os desafios associados às implementações de controladores distribuídos e propõem-se técnicas para modelação e controlo de sistemas distribuídos, nomeadamente o controlo adaptativo distribuído por posicionamento de pólos.

3.1 Implicações da utilização de controladores digitais

A figura 3.1 mostra o diagrama de blocos de um controlador contínuo genérico em que $r(t)$ representa o sinal de referência ou comando, $u(t)$ o sinal de controlo e $y(t)$ a saída do sistema. Neste tipo de controladores o controlo é efectuado tendo por base um sinal contínuo.

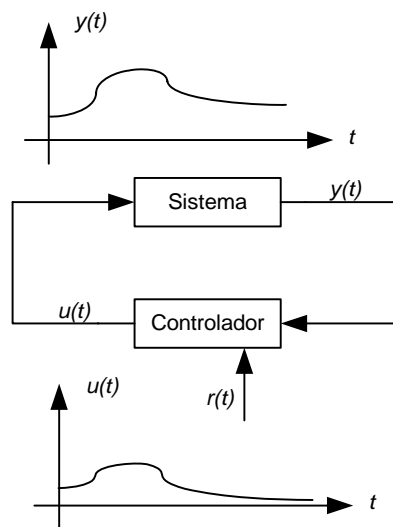


Figura 3.1: Diagrama de blocos de um controlador realimentado genérico.

As técnicas para análise e controlo deste tipo de sistemas baseiam-se em modelos que expressam o comportamento dinâmico (contínuo) do sistema que se pretende controlar. Se o sistema for linear e invariante no tempo ele poderá ser representado, depois da aplicação da transformada de Laplace, em termos da variável complexa s .

Os controladores digitais apresentam características diferentes, como se pode ver no diagrama de blocos da figura 3.2.

Num controlador digital o sinal de saída $y(t)$ é amostrado dando origem ao sinal discreto $y(k)$. O controlador é implementado sob a forma de um algoritmo que é executado por um computador. A amostragem do sistema é efectuada periodicamente sendo o período de amostragem h imposto pelo relógio do computador. O valor amostrado é digitalizado através de um conversor analógico-digital (A/D). O controlador gera o sinal de controlo $u(k)$ que é mantido constante entre cada ciclo de controlo através do circuito designado por *Zero-Order Hold* (ZOH). O sinal resultante é posteriormente convertido num sinal analógico, $u(t)$, através de um conversor digital-analógico (D/A), que é aplicado ao sistema. Para a implementação de um controlador digital podem adoptar-se duas soluções: a solução centralizada, em que o sensor, o actuador e o controlador partilham o mesmo computador, como é o caso do sistema da figura 3.2, ou a solução distribuída, em que o sensor, o actuador e o controlador estão fisicamente

localizados em locais distintos. A segunda abordagem será analisada detalhadamente na secção 3.6.

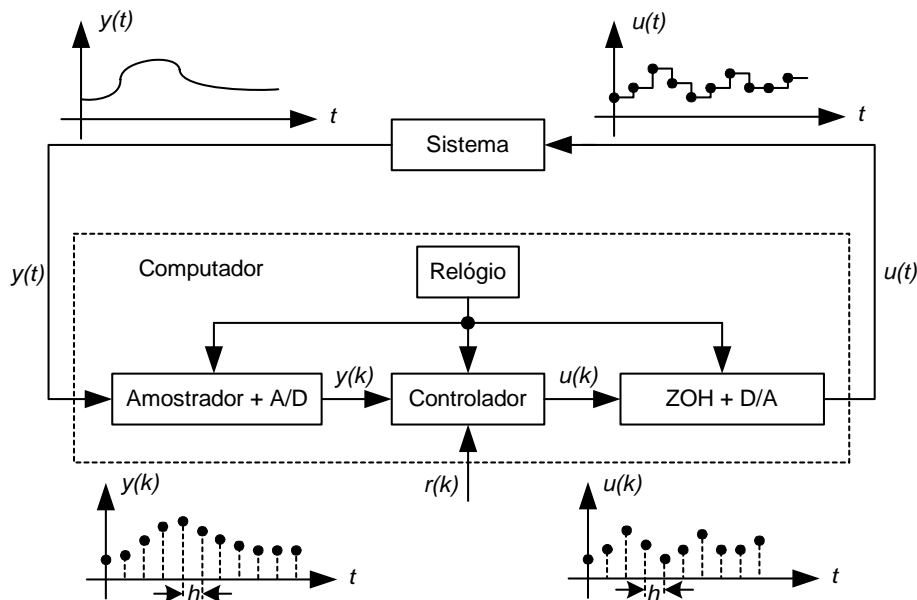


Figura 3.2: Diagrama de blocos de um controlador digital genérico.

A passagem do domínio contínuo para o domínio discreto tem implicações evidentes do ponto de vista prático. Torna-se necessária a selecção de um valor para o período de amostragem e passam, muitas vezes, a utilizar-se modelos discretos do sistema, visto que o controlo passa a ser feito no domínio discreto. Assim muitas das técnicas para análise e controlo passam a ser baseadas em modelos discretos sendo utilizada a transformada de Z e a variável complexa z para descrever o sistema no domínio discreto.

3.1.1 Amostragem do sistema

O sistema é normalmente representado através de um modelo dinâmico que descreve o seu comportamento no domínio contínuo, proporcionando uma relação entre os sinais $u(t)$ e $y(t)$. Em muitos casos essa relação é dada pela função de transferência contínua $G(s) = \frac{Y(s)}{U(s)}$, em que $Y(s)$ e $U(s)$ representam as transformadas de Laplace de $y(t)$ e de $u(t)$, respectivamente.

A função de transferência discreta do sistema pode ser obtida directamente a partir da função de transferência contínua, tendo em consideração a acção dos dispositivos de conversão utilizados para converter os sinais do domínio analógico para o domínio digital e vice-versa. A figura 3.3 apresenta, com maior pormenor, o sistema contínuo e os conversores utilizados. O bloco do conversor A/D foi substituído por um interruptor e o bloco do conversor D/A por um ZOH e um interruptor.

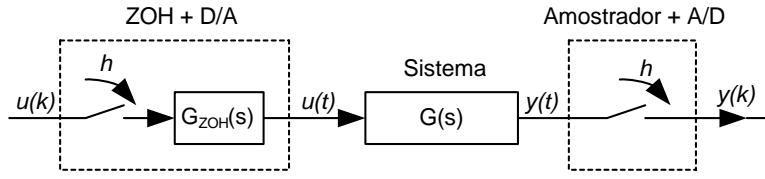


Figura 3.3: Detalhe do sistema de controlo digital.

A função de transferência contínua do circuito de ZOH é dada pela equação 3.1 [97].

$$G_{ZOH}(s) = \frac{1 - e^{-sh}}{s} \quad (3.1)$$

Se o sistema contínuo tiver como função de transferência $G(s)$, a função de transferência discreta $G(z)$ será dada pela expressão 3.2, em que $Y(z)$ e $U(z)$ representam as transformadas de Z das sequências $y(k)$ e $u(k)$, respectivamente. $Z\{\cdot\}$ representa a transformada de Z e $L^{-1}[\cdot]$ a transformada inversa de Laplace.

$$G(z) = \frac{Y(z)}{U(z)} = Z \left\{ L^{-1} [G_{ZOH}(s)G(s)] \right\} = \frac{z-1}{z} Z \left\{ L^{-1} \left[\frac{G(s)}{s} \right] \right\} \quad (3.2)$$

$Z \left\{ L^{-1} \left[\frac{G(s)}{s} \right] \right\}$ pode ser calculado através da expansão em fracções parciais da função $\frac{G(s)}{s}$ e pela posterior aplicação da transformada de Z, utilizando uma tabela de conversão que entre em conta com o dispositivo de ZOH (ver por exemplo [97] ou [98]) ou através do cálculo da transformada inversa de Laplace e posterior amostragem do sistema contínuo obtido. Tradicionalmente na literatura “de controlo” as funções de transferência discretas são representadas utilizando o operador q , definido pela relação 3.3, em vez da variável complexa z . Formalmente as expressões $G(z)$ e $G(q)$ são semelhantes. Assim na representação das funções de transferência discretas será utilizado o operador q cuja definição se faz através da expressão seguinte

$$qx(k) = x(k+1) \quad (3.3)$$

Em geral as funções discretas obtidas através do procedimento anterior podem ser escritas genericamente na forma 3.4, em que q^{-1} representa o atraso de uma amostra introduzido pelo ZOH entre a entrada e a saída do sistema.

$$G(q) = q^{-1} \frac{B(q)}{A(q)} \quad (3.4)$$

em que

$$A(q) = q^{n_a} + a_1 q^{n_a-1} + \dots + a_{n_a}$$

$$B(q) = b_1 q^{n_b-1} + b_2 q^{n_b-2} + \dots + b_{n_b}$$

Chamam-se pólos do sistema às raízes do denominador de $G(q)$, ou seja as raízes do polinómio característico $A(q)$. Os zeros obtêm-se a partir da equação $B(q) = 0$, ou seja são as raízes do polinómio $B(q)$. Os pólos reflectem o modo como as variáveis internas se interligam e determinam o seu comportamento autónomo. Os zeros reflectem o modo como o sistema reage a estímulos externos.

A ordem do sistema corresponde ao número de pólos do sistema ou seja à dimensão da representação de espaço de estados. No caso do sistema cuja função de transferência discreta é dada pela equação 3.4 a ordem será igual a n_a .

3.1.2 Passagem do plano s para o plano z

O mapeamento entre o plano s e o plano z pode ser feito através de várias transformações. A mais utilizada neste âmbito é a transformação $z = e^{sh}$ porque mantém igual a resposta do sistema a um degrau em ambos os domínios. Esta transformação origina uma correspondência directa entre os pólos do modelo contínuo e os pólos do modelo discreto [98]. Sistemas com pólos em s cujo módulo seja negativo correspondem a pólos no plano z cuja localização se situa no interior do círculo unitário, sendo assim preservada a estabilidade do sistema. Pólos no semi-plano positivo em s correspondem a pólos no exterior do círculo unitário em z . Finalmente os pólos sobre o eixo imaginário em s são mapeados sobre o círculo unitário em z . A figura 3.4 representa esse mapeamento. A transformação não é bijectiva, pelo que cada faixa a cinzento no plano s é mapeada sobre o círculo unitário no plano z , ou seja pontos distintos no plano s serão mapeados sobre o mesmo ponto no plano z .

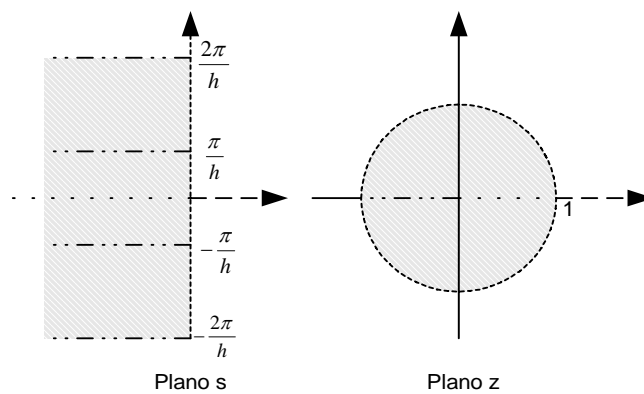


Figura 3.4: Mapeamento entre os planos s e z utilizando a transformação $z = e^{sh}$.

Não existe uma relação directa entre os zeros do modelo contínuo e os zeros do modelo discreto [98], logo não é possível determinar de uma forma simples e directa a correspondência entre uns e outros.

Se a função de transferência contínua do sistema for uma função racional, então a função de transferência discreta terá os zeros correspondentes às raízes do polinómio

do numerador e mais d zeros no infinito, em que d é o chamado excesso de pólos, que é igual à diferença entre o número de pólos e o número de zeros do sistema contínuo.

Quando s tende para infinito a função de transferência contínua pode ser aproximada por [99]

$$G(s) \approx s^{-d}$$

o que, tendo em conta o dispositivo de ZOH, irá corresponder à função de transferência discreta

$$G(z) \approx \left(\frac{h^d}{d!}\right) \left(\frac{B_d(z)}{(z-1)^d}\right)$$

A função discreta vai possuir zeros em

$$z_i \approx e^{s_i h}$$

sendo s_i os zeros do sistema contínuo. Os $d - 1$ zeros correspondentes às raízes não expressas vão surgir em localizações definidas pelas raízes do polinómio $B_d(z)$, à medida que o período de amostragem h se aproxima de zero [99] e [98]

$$B_d(z) = b_1 z^{d-1} + b_2 z^{d-2} + \dots + b_d$$

cujos coeficientes são apresentados na tabela 3.1.

d	$B_d(z)$
1	1
2	$z + 1$
3	$z^2 + 4z + 1$
4	$z^3 + 11z^2 + 11z + 1$
5	$z^4 + 26z^3 + 66z^2 + 26z + 1$
6	$z^5 + 57z^4 + 302z^3 + 302z^2 + 57z + 1$

Tabela 3.1: Polinómios $B_d(z)$.

Nestas circunstâncias os zeros introduzidos pelo processo de amostragem estarão localizados no exterior ou em cima do círculo unitário desde que $d \geq 2$ [98].

Um sistema discreto que tenha zeros fora do círculo unitário é designado por sistema de fase não-mínima e os zeros correspondentes dizem-se de fase não-mínima.

Se considerarmos, por exemplo, um sistema contínuo com a função de transferência:

$$G(s) = \frac{2}{(s+1)(s+2)}$$

Podemos determinar a função de transferência discreta do sistema utilizando uma tabela de conversão [98], obtendo-se a função seguinte

$$G(z) = \frac{(2(1 - e^{-h}) - (1 - e^{-2h}))z + (1 - e^{-2h})e^{-h} - 2(1 - e^{-h})e^{-2h}}{z^2 - (e^{-h} + e^{-2h})z + e^{-3h}}$$

O zero da função de transferência discreta é dado por

$$z = -\frac{(1 - e^{-2h})e^{-h} - 2(1 - e^{-h})e^{-2h}}{2(1 - e^{-h}) - (1 - e^{-2h})}$$

Para $h = 0.001$, $z = -0.999$ e quando h tende para 0, z tende para -1 .

Estes zeros podem levantar problemas pois certas técnicas de controlo adaptativo baseiam-se no cancelamento, por parte do controlador, de alguns zeros do sistema original, no sentido de obter um dado comportamento para o sistema em malha fechada. Esse cancelamento é efectuado através da introdução de pólos, sendo que para os zeros de fase não-mínima terão que se utilizar pólos que são instáveis ou marginalmente estáveis. Como o processo de cancelamento nunca é exacto, os pólos referidos podem tornar o sistema em malha fechada instável [99].

3.1.3 Utilização de modelos discretos do sistema

Com a amostragem do sistema contínuo, o controlo do sistema passa a ser efectuado no domínio discreto. Assim torna-se, em geral, necessária a existência de modelos discretos que descrevam o comportamento do sistema.

No caso dos sistemas de controlo digitais, o controlador descreve o sistema apenas nos instantes de amostragem [98], podendo o sistema discreto ser modelado através da equação 3.5.

$$y(k+1) = f[y(k), y(k-1), \dots, u(k), u(k-1), \dots] \quad (3.5)$$

em que $y(k)$ e $u(k)$ representam a saída e a entrada do sistema, respectivamente, nos instantes t_k , com $t_k = kh$, sendo h o período de amostragem utilizado no sistema e $k = 1, 2, \dots$. Este tipo de modelos, que fornecem informação sobre o sistema apenas nos instantes de amostragem, designam-se por modelos estroboscópicos. O modelo apresentado é também um modelo paramétrico, para o qual é necessário especificar uma ordem. A obtenção de um bom modelo depende da escolha certa da ordem do mesmo, que deve ser semelhante à do sistema contínuo.

A figura 3.2 e a equação 3.5 podem fazer supor que será sempre possível determinar um modelo do sistema tendo por base apenas os sinais $y(k)$ e $u(k)$. No entanto na maioria das situações essa análise é simplista porque o sistema é afectado por sinais para lá do controlo do utilizador [100]. Esses sinais são designados perturbações e podem corresponder, por exemplo, a ruído de medição ou a entradas que não são controladas pelo utilizador. Os efeitos das perturbações podem ser tidos em conta considerando um termo $v(k)$ adicionado ao sistema como se mostra na figura 3.5.

Neste caso a saída passa a ser dada pela expressão 3.6:

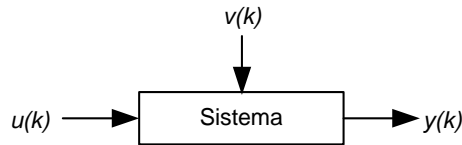


Figura 3.5: Sistema com perturbação.

$$y(k) = \eta(k) + v(k) \quad (3.6)$$

em que $\eta(k)$ representa a parte da saída que não é afectada por ruído e $v(k)$ a parte afectada pelas perturbações.

A escolha do modelo a utilizar é baseada no conhecimento prévio do sistema e das perturbações que o afectam. Em muitos casos o conhecimento relativo ao sistema é reduzido, sabendo-se apenas que o sistema é linear dentro de uma dada gama de funcionamento. Nestes casos torna-se natural a utilização de representações gerais de sistemas lineares discretos que se designam por modelos de caixa negra. Os parâmetros obtidos para estes sistemas não têm interpretação física directa limitando-se a expressar a relação entre a entrada e a saída do sistema. Alguns desses modelos são apresentados na secção 3.3.

3.2 Amostragem de um sistema contínuo

Antes de se abordar a amostragem de um sistema contínuo apresenta-se o modo como é representado o tempo morto num sistema discreto.

3.2.1 Representação do tempo morto

O tempo morto corresponde ao tempo que uma variação na entrada do sistema demora a manifestar-se na saída. O sistema da equação 3.4 apresenta um tempo morto igual a uma amostra. Na prática muitos sistemas apresentam tempo morto, devido a factores internos e a factores externos, por exemplo, a atrasos computacionais decorrentes do processamento do algoritmo de controlo.

Consideremos um sistema com tempo morto t_d . Se t_d é múltiplo inteiro do período de amostragem, ou seja

$$t_d = Kh$$

sendo K um número inteiro positivo, ele pode ser representado na função de transferência discreta sob a forma de um atraso como mostra a expressão seguinte:

$$G(q^{-1}) = q^{-K} \frac{B(q^{-1})}{A(q^{-1})} \quad (3.7)$$

em que K engloba também o atraso introduzido pelo ZOH. Como se pode ver a existência de tempo morto discreto origina no modelo discreto o aparecimento de K zeros na origem.

Na prática o tempo morto nunca é um múltiplo exacto do período de amostragem, sendo possível representá-lo em termos de um múltiplo inteiro de h e de um tempo morto fraccionário [99] através da expressão seguinte

$$\tau_d = Kh - mh$$

em que

$$0 \leq m < 1$$

A existência de Tempo Morto Fraccionário (TMF) origina o aparecimento de um zero extra na função de transferência discreta, como se pode ver no exemplo seguinte [97] e [99].

Considere-se o sistema

$$G(s) = \frac{ae^{-s(Kh-mh)}}{s+a}$$

com $0 \leq m < 1$. A função de transferência discreta é dada por

$$G(z) = \frac{z-1}{z} z^{-K} Z \left\{ L^{-1} \left[\frac{ae^{smh}}{s(s+a)} \right] \right\}$$

Utilizando o método de expansão em fracções parciais obtém-se

$$G(z) = \frac{z-1}{z} z^{-K} \left(Z \left\{ L^{-1} \left[\frac{1}{s} \right] \right\} - Z \left\{ L^{-1} \left[\frac{e^{amh}}{s+a} \right] \right\} \right)$$

$$G(z) = \frac{z-1}{z} z^{-K} \left(\frac{z}{z-1} - \frac{ze^{-amh}}{z-e^{-ah}} \right)$$

$$G(z) = z^{-K} \left(\frac{z(1-e^{-amh}) - e^{-ah} + e^{-amh}}{z-e^{-ah}} \right) \quad (3.8)$$

A função $G(z)$ dada por 3.8 tem um zero em

$$z_m = \frac{e^{-ah} - e^{-amh}}{1 - e^{-amh}}$$

Para $m = 1$, o zero está posicionado em $z_m = 0$ e $G(z)$ é dada por

$$G(z)_{m=1} = z^{-(K-1)} \left(\frac{1 - e^{-ah}}{z - e^{-ah}} \right)$$

Quando m tende para 0, z_m tende para $-\infty$ e $G(z)$ é dada por

$$G(z)_{m=0} = z^{-K} \left(\frac{1 - e^{-ah}}{z - e^{-ah}} \right)$$

Verifica-se que à medida que m varia entre 1 e 0, a localização do zero da função de transferência discreta vai-se deslocando da origem até $-\infty$, pelo que para determinados valores de m ele irá residir fora do círculo unitário, dando origem a um sistema de fase não-mínima.

O tempo morto fraccionário pode ser constante ou variável. Em geral em aplicações de controlo em que o controlador é implementado de modo centralizado e sem partilha de recursos ele será constante; nas implementações distribuídas ele será variável. Nas abordagens tradicionais o tempo morto fraccionário não é tido em conta, utilizando-se para expressar o tempo morto apenas a aproximação inteira.

A consideração do tempo morto fraccionário tem consequências na implementação do controlador, originando implementações mais complexas que requerem, como se verá na secção 3.7, uma maior capacidade computacional.

3.2.2 Amostragem de sistemas sem tempo morto

A relação entre os sinais $y(t)$ e $u(t)$ pode ser expressa sob a forma de equações de espaço de estados. Neste caso também é possível proceder à amostragem do sistema contínuo, tendo em conta o efeito dos conversores utilizados, de forma a obter a relação entre as sequências $u(k)$ e $y(k)$

Dado um sistema contínuo representado pelas equações de espaço de estados 3.9 e 3.10:

$$\frac{dx}{dt} = Ax(t) + Bu(t) \quad (3.9)$$

$$y(t) = Cx(t) \quad (3.10)$$

Se se puder considerar que a entrada $u(t)$ e a saída $y(t)$ são amostradas no mesmo instante, sendo h o período de amostragem, então as equações seguintes descrevem o sistema amostrado invariante no tempo [98]:

$$x(kh + h) = \Phi x(kh) + \Gamma u(kh) \quad (3.11)$$

$$y(kh) = Cx(kh) \quad (3.12)$$

em que

$$\Phi = e^{Ah} \quad (3.13)$$

$$\Gamma = \int_0^h e^{As} ds B \quad (3.14)$$

O modelo de espaço de estados do sistema discreto será dado por

$$\begin{bmatrix} x(kh + h) \\ u(kh) \end{bmatrix} = \begin{bmatrix} \Phi \\ 0 \end{bmatrix} \begin{bmatrix} x(kh) \\ 0 \end{bmatrix} + \Gamma \begin{bmatrix} u(kh) \\ I \end{bmatrix} \quad (3.15)$$

e a função de transferência discreta por

$$G(q) = C(qI - \Phi)^{-1}\Gamma \quad (3.16)$$

O modelo obtido para o sistema discreto representa a relação entre a entrada e a saída do sistema apenas nos instantes de amostragem.

Considere-se por exemplo o sistema de 1^a ordem

$$\frac{dx}{dt} = 2x(t) + 3u(t)$$

Aplicando as equações 3.13 e 3.14 obtém-se

$$\Phi = e^{2h}$$

$$\Gamma = \int_0^h e^{2s} ds 3 = \frac{3}{2}(e^{2h} - 1)$$

O sistema amostrado é dado por

$$x(kh + h) = e^{2h}x(kh) + \frac{3}{2}(e^{2h} - 1)u(kh)$$

Como se pode ver o estado interno do sistema amostrado depende apenas de $u(kh)$.

3.2.3 Amostragem de sistemas com tempo morto

No caso do sistema possuir tempo morto a sua representação através de equações de estado é dada por 3.17 e 3.18.

$$\frac{dx}{dt} = Ax(t) + Bu(t - \tau) \quad (3.17)$$

$$y(t) = Cx(t) \quad (3.18)$$

em que τ representa o tempo morto.

Partindo do pressuposto que $\tau < h$, ou seja que τ é um tempo morto fraccionário, e considerando o efeito do dispositivo de ZOH, o sistema amostrado é dado pelas equações seguintes [98]:

$$x(kh + h) = \Phi x(kh) + \Gamma_0 u(kh) + \Gamma_1 u(kh - h) \quad (3.19)$$

$$y(kh) = Cx(kh) \quad (3.20)$$

em que

$$\Phi = e^{Ah} \tag{3.21}$$

$$\Gamma_0 = \int_0^{h-\tau} e^{As} ds B \tag{3.22}$$

$$\Gamma_1 = e^{A(h-\tau)} \int_0^{\tau} e^{As} ds B \tag{3.23}$$

O modelo de espaço de estados do sistema discreto será dado por

$$\begin{bmatrix} x(kh+h) \\ u(kh) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(kh) \\ x(kh-h) \end{bmatrix} + \Gamma_0 \begin{bmatrix} u(kh) \\ I \end{bmatrix} \tag{3.24}$$

e a função de transferência discreta é dada por

$$G(q) = C(qI - \Phi)^{-1}(\Gamma_0 + \Gamma_1 q^{-1}) \tag{3.25}$$

Apresenta-se, de seguida, uma reformulação do exemplo anterior que considera a existência de um tempo morto fraccionário de valor τ . O sistema de 1ª ordem é o seguinte

$$\frac{dx}{dt} = 2x(t) + 3u(t - \tau)$$

Aplicando as equações 3.21, 3.22 e 3.23 obtém-se

$$\Phi = e^{2h}$$

$$\Gamma_0 = \int_0^{h-\tau} e^{2s} ds 3 = \frac{3}{2}(e^{2(h-\tau)} - 1)$$

$$\Gamma_1 = e^{2(h-\tau)} \int_0^{\tau} e^{2s} ds 3 = \frac{3}{2}(e^{2h} - e^{2(h-\tau)})$$

O sistema amostrado é dado por

$$x(kh+h) = e^{2h}x(kh) + \frac{3}{2}(e^{2(h-\tau)} - 1)u(kh) + \frac{3}{2}(e^{2h} - e^{2(h-\tau)})u(kh-h)$$

Comparando este resultado com o obtido para o caso em que o tempo morto fraccionário não era considerado pode concluir-se que o estado do sistema amostrado passa a depender não só de $u(kh)$ mas também de $u(kh-h)$. A determinação do estado do sistema torna-se mais complexa. Em termos da função de transferência discreta do sistema isso vai originar a existência de um zero adicional.

3.2.4 Selecção do período de amostragem

O período de amostragem (h) é um parâmetro fundamental no funcionamento de um controlador digital porque o seu valor influencia o desempenho do mesmo.

Existem vários factores que condicionam a escolha do valor do período de amostragem. Numa aplicação de controlo esses factores estão relacionados com [101]: a aplicação (largura de banda, ruído, ambiente), a técnica de controlo (precisão) e com as capacidades do computador (velocidade e número de *bits*). Apesar disso não existe nenhuma regra genérica que permita determinar o valor exacto do período de amostragem para uma dada aplicação.

Como ponto de partida podem analisar-se os factores que limitam a escolha do valor máximo e do valor mínimo do período de amostragem.

O valor máximo do período de amostragem é determinado pela capacidade de se poder reconstruir o sinal contínuo a partir do sinal amostrado. Esse limite é dado pelo teorema da amostragem de Shannon que diz que para que um sinal contínuo com uma frequência máxima igual a f possa ser completamente reconstruído a partir da versão amostrada (em instantes igualmente espaçados no tempo), ele terá que ser amostrado com um período igual ou inferior a $\frac{1}{2f}$. A frequência $2f$ designa-se por frequência de Nyquist. Na prática e de modo a garantir que o sistema a amostrar não apresenta componentes de frequência superior ao esperado (devidas por exemplo à introdução de ruído) é habitual proceder-se à filtragem do sinal.

O valor mínimo do período de amostragem é determinado, em última instância, pelo tempo de conversão dos dispositivos de conversão A/D e D/A utilizados para amostrar o sinal. No entanto de acordo com Ogata [102] este acaba por não ser o factor de limitação mais importante uma vez que a selecção de h depende também da capacidade de processamento do computador utilizado para implementar o algoritmo de controlo, que origina normalmente tempos superiores ao tempo de conversão.

Por outro lado pode existir a tentação de utilizar um período de amostragem tão curto quanto possível tentando aproximar o funcionamento de um controlador contínuo. Porém, convém não esquecer que, tal como foi referido na secção anterior, a utilização de valores muito pequenos para h leva ao aparecimento de zeros de fase não-mínima que podem criar problemas de estabilidade.

No caso de sistemas de controlo a operar em malha fechada a escolha do período de amostragem deve basear-se no conhecimento da sua influência sobre o desempenho final do controlador. Assim admite-se que a máxima frequência de interesse esteja relacionada com a largura de banda do sistema em malha fechada. A selecção pode assim basear-se na largura de banda (ou no tempo de subida) do sistema em malha fechada [98].

Existem várias regras “de bolso” que sugerem um intervalo dentro do qual deverá ser seleccionado o período de amostragem, intervalo esse consideravelmente mais limitado do que aquele dado pelos limites referidos anteriormente.

Em Houpis e Lamont [101] propõe-se que o valor da frequência de amostragem seja 4 ou 5 vezes superior à frequência de Nyquist ou alternativamente que o período de amostragem seja igual a $\frac{1}{2}$ a $\frac{1}{3}$ do tempo de subida.

Ljung [100] sugere que a frequência de amostragem seja cerca de 10 vezes superior

à largura de banda do sistema.

Por seu turno Holmberg [103] afirma que é prática comum que a selecção do período de amostragem se relacione com a largura de banda do sistema em malha fechada de acordo com a expressão seguinte

$$\omega_B h \approx 0.5 \text{ a } 1$$

em que ω_B representa a frequência para a qual o ganho do sistema em malha fechada baixa para 0.7.

De acordo com Aström e Wittenmark [98], o período de amostragem deve ser 10 a 30 vezes superior à largura de banda do sistema ou corresponder a 4 a 10 amostras por tempo de subida de acordo com a equação

$$N_r = \frac{T_r}{h} \simeq 4 \text{ a } 10 \quad (3.26)$$

No caso de sistemas de 1ª ordem o tempo de subida é igual à constante de tempo. Para sistemas de 2ª ordem o tempo de subida está relacionado com o coeficiente de amortecimento ζ e com a frequência natural ω_n . Para sistemas com amortecimento aproximadamente igual a 0.7 obtém-se a relação seguinte [98]:

$$\omega_n h \approx 0.2 \text{ a } 0.6$$

Algumas destas regras aparecem sumariadas em [104].

Em conclusão, não existe uma regra geral e unânime que sirva para seleccionar qual o valor do período de amostragem. Ao longo deste trabalho utiliza-se o critério de Aström e Wittenmark [98], já descrito. Este critério foi seleccionado devido à credibilidade e experiência dos seus autores, sendo considerado um critério conservador.

Tal como já foi referido, a escolha do período de amostragem tem implicações directas no desempenho do controlador. Pode afirmar-se genericamente que para sistemas de controlo com h constante e sem tempo morto fraccionário, quanto menor o período de amostragem melhor o desempenho do controlador. No entanto num sistema com tempo morto fraccionário não é possível fazer uma análise assim directa, uma vez que passa a ser mais importante a relação entre o valor do tempo morto fraccionário e o valor do período do que o valor absoluto do tempo morto fraccionário.

Em controlo digital “tradicional” o período de amostragem é um parâmetro de projecto que, depois de seleccionado, se mantém constante ao longo do ciclo de vida do controlador. Na última década contudo apareceram aplicações em que vários controladores digitais partilham os mesmos recursos físicos, como por exemplo o processador ou a infraestrutura de comunicação, levando à necessidade de optimização do desempenho global desses sistemas que pode passar, em alguns casos, pela alteração do valor de h durante a operação do sistema de controlo. Assim tem-se assistido ao surgimento de uma série de técnicas (ver por exemplo [4] e [5]) que pressupõem que os controladores podem ter um período de amostragem variável, técnicas essas que tentam optimizar o valor instantâneo de h em função de parâmetros que não eram considerados até então como, por exemplo, a utilização total do processador ou da infraestrutura de comunicação. Na secção 3.6 estas questões serão novamente abordadas.

3.3 Modelos discretos do sistema

Como foi referido no início deste capítulo, a passagem do domínio contínuo para o domínio discreto resultante do processo de amostragem faz com que surja a necessidade de se utilizarem modelos discretos do sistema durante a fase de projecto do controlador.

Um modelo linear discreto pode ser descrito pela equação 3.27 de acordo com Ljung [100].

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k) \quad (3.27)$$

em que q^{-1} representa o operador atraso ou seja, $q^{-1}u(k) = u(k-1)$, $G(q^{-1})$ representa a função de transferência discreta do sistema, $H(q^{-1})$ representa a influência das perturbações no sistema e $e(k)$ representa ruído branco. Comparando este modelo com a equação 3.6 podemos ver que o termo livre de ruído $\eta(k)$ e o termo afectado por ruído $v(k)$, são dados, respectivamente, por $G(q^{-1})u(k)$ e $H(q^{-1})e(k)$.

3.3.1 Classes de modelos mais utilizadas

A estrutura mais genérica de um modelo discreto, a partir da qual é possível obter várias classes de modelos, é apresentada na equação 3.28. As diferentes classes obtêm-se através da selecção dos polinómios A , B , C , D e F .

$$A(q^{-1})y(k) = q^{-t_d} \frac{B(q^{-1})}{F(q^{-1})} u(k) + \frac{C(q^{-1})}{D(q^{-1})} e(k) \quad (3.28)$$

Na equação 3.28 t_d representa o tempo morto discreto, ou seja o número de amostras entre a aplicação do sinal de entrada $u(k)$ e o aparecimento do respectivo efeito na saída. Os polinómios são do tipo:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n_a} \\ B(q^{-1}) &= b_1 q^{-1} + \dots + b_m q^{-n_b} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_i q^{-n_c} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \dots + d_j q^{-n_d} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \dots + f_r q^{-n_f} \end{aligned} \quad (3.29)$$

onde n_a , n_b , n_c , n_d e n_f representam a ordem dos polinómios A , B , C , D e F , respectivamente.

De seguida apresentam-se algumas das classes de modelos mais utilizadas.

Estrutura *Finite Impulse Response* (FIR)

Esta é a estrutura mais simples que se pode obter a partir do modelo genérico. Dos polinómios do modelo genérico só se utiliza o polinómio B . Os outros polinómios são dados por

$$A(q^{-1}) = C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$$

originando

$$G(q^{-1}) = q^{-td}B(q^{-1})$$

$$H(q^{-1}) = 1$$

O modelo FIR é dado por

$$y(k) = q^{-td}B(q^{-1})u(k) + e(k) \quad (3.30)$$

Um modelo desta classe apresenta algumas limitações. Como se pode ver pela equação 3.30, ele não inclui nenhuma informação sobre a saída e , sendo de ordem finita, não pode descrever de forma exacta um sistema com pólos. A aproximação será tanto melhor quanto maior for o número de parâmetros utilizados, sendo por isso necessário utilizar um número elevado o que pode originar problemas de implementação.

Estrutura *AutoRegressive with eXogenous signal* (ARX)

Este modelo utiliza os polinómios A e B do modelo genérico. Os outros polinómios são dados por

$$C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$$

originando

$$G(q^{-1}) = \frac{q^{-td}B(q^{-1})}{A(q^{-1})}$$

$$H(q^{-1}) = \frac{1}{A(q^{-1})}$$

O modelo ARX é dado por

$$y(k) = \frac{q^{-td}B(q^{-1})}{A(q^{-1})}u(k) + \frac{1}{A(q^{-1})}e(k) \quad (3.31)$$

Este modelo é o mais simples de estimar porque corresponde à estimativa de uma regressão linear [105]. A única desvantagem prende-se com o facto da modelação da perturbação incluir os pólos do sistema o que pode levar a uma estimativa incorrecta da dinâmica do sistema, visto que o polinómio $A(q^{-1})$ serve para descrever tanto a dinâmica do sistema como a das perturbações. No entanto se a relação sinal-ruído for boa esta desvantagem torna-se menos importante [105].

Na prática Ljung e Glad [105] recomendam que se utilize a estrutura ARX sempre que possível porque tal como foi referido é a que torna o processo de estimativa dos parâmetros mais simples e permite a obtenção de bons resultados se a relação sinal-ruído for boa.

Estrutura *AutoRegressive Moving Average with eXogenous signal* (ARMAX)

Este modelo utiliza os polinómios A , B e C do modelo genérico. Os outros polinómios são dados por

$$D(q^{-1}) = F(q^{-1}) = 1$$

que originam

$$G(q^{-1}) = \frac{q^{-t_d} B(q^{-1})}{A(q^{-1})}$$

$$H(q^{-1}) = \frac{C(q^{-1})}{A(q^{-1})}$$

O modelo ARMAX é dado por

$$y(k) = \frac{q^{-t_d} B(q^{-1})}{A(q^{-1})} u(k) + \frac{C(q^{-1})}{A(q^{-1})} e(k) \quad (3.32)$$

O modelo ARMAX permite maior flexibilidade na descrição da perturbação em relação ao modelo ARX. Tanto o modelo ARX como o modelo ARMAX pressupõem que a perturbação sofre a mesma dinâmica que o sinal de entrada e por isso utilizam-se em situações nas quais as perturbações dominantes “entram” no sistema perto da entrada do sistema, por exemplo junto com o sinal de entrada.

3.3.2 Outras classes de modelos

Para além das classes apresentadas anteriormente, que são as mais utilizadas, existem ainda outras que são apresentadas de seguida.

Estrutura *Output Error* (OE)

Este modelo utiliza os polinómios B e F do modelo genérico. Os outros polinómios são dados por

$$A(q^{-1}) = C(q^{-1}) = D(q^{-1}) = 1$$

dando origem a

$$G(q^{-1}) = \frac{q^{-t_d} B(q^{-1})}{F(q^{-1})}$$

$$H(q^{-1}) = 1$$

O modelo OE é dado por

$$y(k) = \frac{q^{-t_d} B(q^{-1})}{F(q^{-1})} u(k) + e(k) \quad (3.33)$$

Este modelo permite a descrição da dinâmica do sistema de modo independente, não sendo utilizados parâmetros na modelação da perturbação. No entanto a determinação dos parâmetros pode ser mais difícil do que no caso do modelo ARMAX [105].

Estrutura *Box-Jenkins* (BJ)

Este modelo utiliza os polinómios B , C , D e F do modelo genérico. O polinómio A é dado por

$$A(q^{-1}) = 1$$

dando origem a

$$G(q^{-1}) = \frac{q^{-t_d} B(q^{-1})}{F(q^{-1})}$$

$$H(q^{-1}) = \frac{C(q^{-1})}{D(q^{-1})}$$

O modelo BJ é dado por

$$y(k) = \frac{q^{-t_d} B(q^{-1})}{F(q^{-1})} u(k) + \frac{C(q^{-1})}{D(q^{-1})} e(k) \quad (3.34)$$

Este modelo é o mais complexo. Permite um maior grau de liberdade na modelação do termo de perturbação, que é independente do termo associado à dinâmica do sistema, e utiliza-se para sistemas em que a perturbação “entra” no sistema mais perto da saída do mesmo, por exemplo sob a forma de ruído de medida.

Estrutura *AutoRegressive Integrating Moving Average with eXogenous signal* (ARIMAX)

Este modelo é semelhante ao modelo ARMAX. Utiliza os polinómios A , B e C do modelo genérico e inclui uma integração extra Δ no termo associado às perturbações, o que o distingue do modelo ARMAX. Os outros polinómios são dados por

$$D(q^{-1}) = F(q^{-1}) = 1$$

que originam

$$G(q^{-1}) = \frac{q^{-t_d} B(q^{-1})}{A(q^{-1})}$$

$$H(q^{-1}) = \frac{C(q^{-1})}{A(q^{-1})}$$

e

$$\Delta = 1 - q^{-1}$$

O modelo ARIMAX é dado por

$$y(k) = \frac{q^{-t_d} B(q^{-1})}{A(q^{-1})} u(k) + \frac{C(q^{-1})}{A(q^{-1})} \frac{e(k)}{\Delta} \quad (3.35)$$

Este modelo utiliza-se em sistemas afectados por perturbações lentas [100].

3.3.3 Modelos discretos sem tempo morto fraccionário

Nesta secção utiliza-se o modelo discreto ARX para modelar um sistema de 1ª e outro de 2ª ordem sem ter em consideração o tempo morto fraccionário.

A amostragem de um sistema contínuo sem tempo morto obtém-se através das equações de espaço de estados que são recuperadas da secção 3.2.2.

$$\begin{aligned} x(kh + h) &= \Phi x(kh) + \Gamma u(kh) \\ y(kh) &= Cx(kh) \end{aligned}$$

em que $\Phi = e^{Ah}$ e $\Gamma = \int_0^h e^{As} ds B$.

Considerando que $C = 1$ a função de transferência discreta do sistema é dada por

$$G(q) = (qI - \Phi)^{-1} \Gamma$$

Se o sistema tiver uma única entrada e uma única saída obtém-se uma expressão do tipo

$$G(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}$$

A classe de modelos ARX é dada por (ver secção 3.3.1)

$$y(k) = \frac{q^{-t_d} B(q^{-1})}{A(q^{-1})} u(k) + \frac{1}{A(q^{-1})} e(k)$$

Se o sistema for de 1ª ordem e utilizando a estrutura ARX obtém-se o modelo discreto dado pelas equações 3.36 e 3.37

$$G(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} \quad (3.36)$$

$$H(q^{-1}) = \frac{1}{1 + a_1 q^{-1}} \quad (3.37)$$

ou seja

$$y(k) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} u(k) + \frac{1}{1 + a_1 q^{-1}} e(k) \quad (3.38)$$

Se o sistema for de 2ª ordem e utilizando a mesma estrutura obtém-se o modelo discreto dado pelas equações 3.39 e 3.40

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}} \quad (3.39)$$

$$H(q^{-1}) = \frac{1}{1 + a_1q^{-1} + a_2q^{-2}} \quad (3.40)$$

ou seja

$$y(k) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}}u(k) + \frac{1}{1 + a_1q^{-1} + a_2q^{-2}}e(k)$$

3.3.4 Modelos discretos com tempo morto fraccionário

A amostragem do sistema contínuo tendo em conta o tempo morto fraccionário é de particular interesse em sistemas distribuídos nos quais o acesso ao barramento (entre outros factores) introduz atrasos variáveis na malha de controlo. A amostragem de um sistema tendo em conta o tempo morto fraccionário origina modelos discretos distintos dos anteriores, como se pode verificar de seguida.

A amostragem tendo em conta o tempo morto fraccionário obtém-se através das equações de espaço de estados seguintes, recuperadas de secção 3.2.3. O tempo morto fraccionário é representado por τ e considera-se que $\tau < h$.

$$\begin{aligned} x(kh + h) &= \Phi x(kh) + \Gamma_0 u(kh) + \Gamma_1 u(kh - h) \\ y(kh) &= Cx(kh) \end{aligned}$$

em que $\Phi = e^{Ah}$, $\Gamma_0 = \int_0^{h-\tau} e^{As} ds B$ e $\Gamma_1 = e^{A(h-\tau)} \int_0^{\tau} e^{As} ds B$.

Considerando que $C = 1$ a função de transferência discreta do sistema é dada pela expressão

$$G(q) = (qI - \Phi)^{-1}(\Gamma_0 + \Gamma_1 q^{-1})$$

Se o sistema tiver uma única entrada e um única saída obtém-se uma expressão do tipo

$$G(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}$$

Se o sistema for de 1ª ordem e utilizando a estrutura ARX obtém-se o modelo discreto dado pelas equações 3.41 e 3.37. A função $H(q^{-1})$ não sofre alteração em relação à situação em que não se considera o tempo morto fraccionário porque a equação característica do sistema se mantém igual.

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1}} \quad (3.41)$$

O modelo ARX do sistema de 1ª ordem com tempo morto fraccionário é dado por

$$y(k) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1}}u(k) + \frac{1}{1 + a_1q^{-1}}e(k)$$

Se o sistema for de 2ª ordem e utilizando a mesma estrutura obtém-se o modelo discreto dado pelas equações 3.42 e 3.40.

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2} + b_3q^{-3}}{1 + a_1q^{-1} + a_2q^{-2}} \quad (3.42)$$

ou seja

$$y(k) = \frac{b_1q^{-1} + b_2q^{-2} + b_3q^{-3}}{1 + a_1q^{-1} + a_2q^{-2}}u(k) + \frac{1}{1 + a_1q^{-1} + a_2q^{-2}}e(k)$$

Os modelos que consideram o tempo morto fraccionário apresentam um zero extra na função de transferência discreta do sistema o que leva a implementações mais complexas do que aquelas em que o tempo morto fraccionário não é considerado. Estes modelos apresentam mais um parâmetro do que no caso anterior.

3.4 Identificação dos parâmetros do modelo do sistema

No contexto do controlo de sistemas o objectivo da identificação é o de fornecer um modelo matemático do sistema a controlar tendo por base dados obtidos experimentalmente relativos ao comportamento do sistema. Esse modelo deve representar as características dinâmicas mais importantes do mesmo.

Depois de seleccionado o modelo torna-se necessária a definição do critério que permite a determinação dos parâmetros.

Normalmente o objectivo da identificação é o de minimizar o chamado erro de estimativa, ou seja o erro entre o valor de saída do sistema real $y(k)$ e o valor de saída proposto pelo modelo $\hat{y}(k)$. A figura 3.6 apresenta uma representação gráfica do erro de estimativa.

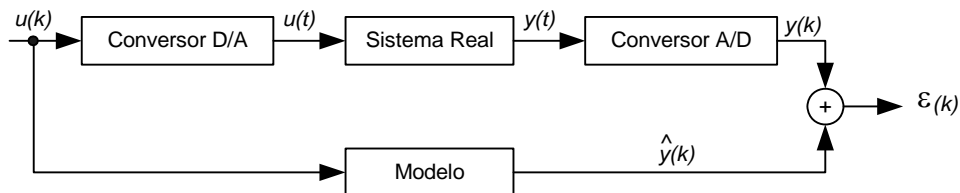


Figura 3.6: Representação gráfica do erro de estimativa.

A função de custo utilizada mais usualmente é baseada no quadrado do erro de estimativa e é dada pela equação 3.43.

$$J = J(y, \hat{y}) = \int (y(k) - \hat{y}(k))^2 dk = \int \varepsilon^2(k) dk \quad (3.43)$$

Existem vários métodos para minimizar uma função de custo deste tipo. O método dos mínimos quadrados (LS da designação em inglês *Least Squares*) é um dos mais simples e eficazes. A versão recursiva do método dos mínimos quadrados (RLS da designação em inglês *Recursive Least Squares*) é a mais adequada à utilização em sistemas de controlo adaptativo porque nestes sistemas a estimativa tem que ser efectuada em tempo-real. Refiram-se ainda outros dois métodos. O primeiro é o método *Approximate Maximum Likelihood* (AML) [100] que se utiliza quando o ruído que afecta o sistema não é ruído branco e que é um método baseado em inferência estatística. Este método é de aplicação mais complexa e demora mais tempo a convergir do que o RLS. O segundo é o método *Least Mean Square* (LMS) [106], baseado na análise do gradiente. Este método é conhecido pela sua simplicidade e rapidez de computação.

3.4.1 Método dos mínimos quadrados

O método mais utilizado, pela sua simplicidade e eficiência, para minimizar a função de custo é o método dos mínimos quadrados. De seguida faz-se uma breve apresentação desse método.

O sistema deve poder ser representado por uma função linear em θ , do tipo

$$\hat{y}(k) = \theta^T \varphi(k) \quad (3.44)$$

designada por equação de regressão, onde $\hat{y}(k)$ representa o valor estimado da saída do sistema, θ o vector que contém os parâmetros (a determinar) do sistema, da forma

$$\theta = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n] \quad (3.45)$$

e $\varphi(k)$ o vector dos regressores, representa um conjunto de variáveis conhecidas que geralmente são versões atrasadas dos sinais de entrada e de saída do sistema. O vector dos regressores é dado por

$$\varphi(k) = [\varphi_1(k) \quad \varphi_2(k) \quad \dots \quad \varphi_N(k)] \quad (3.46)$$

Nas equações 3.45 e 3.46 n é o número de parâmetros do modelo e N o número de observações recolhidas do sistema.

O método dos mínimos quadrados tem por objectivo determinar uma estimativa dos parâmetros θ de modo a minimizar a função de custo J dada pela expressão 3.47, com base no conhecimento da ordem do sistema n e de um conjunto de dados relativos ao sistema (observações) para $k = 1, \dots, N$. Os dados são obtidos a partir do sistema real e devem ter todos a mesma precisão.

$$J(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N \varepsilon^2(i) \quad (3.47)$$

O vector $\hat{\boldsymbol{\theta}}$ representa os parâmetros estimados e ε o erro de predição, que é dado por 3.48 (ver figura 3.6):

$$\varepsilon(k) = y(k) - \hat{y}(k) = y(k) - \hat{\boldsymbol{\theta}}^T \boldsymbol{\varphi}(k) \quad (3.48)$$

Substituindo 3.48 em 3.47 e utilizando notação matricial obtém-se

$$J(\hat{\boldsymbol{\theta}}) = \frac{1}{2} \mathbf{E}^T \mathbf{E} = \frac{1}{2} \|\mathbf{E}\|^2 \quad (3.49)$$

em que

$$\mathbf{E} = [\varepsilon_1 \quad \varepsilon_2 \quad \dots \quad \varepsilon_N] \quad (3.50)$$

Considerando os seguintes vectores

$$\mathbf{Y} = [y_1 \quad y_2 \quad \dots \quad y_N]^T$$

$$\hat{\boldsymbol{\theta}} = [\hat{\theta}_1 \quad \hat{\theta}_2 \quad \dots \quad \hat{\theta}_n]^T$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}^T(1) \\ \boldsymbol{\varphi}^T(2) \\ \dots \\ \boldsymbol{\varphi}^T(N) \end{bmatrix}$$

pode escrever-se

$$\mathbf{E} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \boldsymbol{\Phi} \hat{\boldsymbol{\theta}} \quad (3.51)$$

A função de custo pode ser escrita como:

$$2J(\hat{\boldsymbol{\theta}}) = (\mathbf{Y} - \boldsymbol{\Phi} \hat{\boldsymbol{\theta}})^T (\mathbf{Y} - \boldsymbol{\Phi} \hat{\boldsymbol{\theta}}) = \mathbf{Y}^T \mathbf{Y} - \mathbf{Y}^T \boldsymbol{\Phi} \hat{\boldsymbol{\theta}} - \boldsymbol{\Phi}^T \hat{\boldsymbol{\theta}}^T \mathbf{Y} + \boldsymbol{\Phi}^T \hat{\boldsymbol{\theta}}^T \boldsymbol{\Phi} \hat{\boldsymbol{\theta}}$$

Como a matriz $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ é sempre positiva (ou nula) a função J possui um mínimo. Esse mínimo obtém-se derivando a função J em ordem a $\boldsymbol{\theta}$ e igualando o resultado a zero, como se mostra de seguida.

$$\frac{\delta J}{\delta \hat{\boldsymbol{\theta}}} = 0 \iff -\mathbf{Y}^T \boldsymbol{\Phi} - \boldsymbol{\Phi}^T \mathbf{Y} + 2\boldsymbol{\Phi}^T \boldsymbol{\Phi} \hat{\boldsymbol{\theta}} = 0 \iff \boldsymbol{\Phi}^T \boldsymbol{\Phi} \hat{\boldsymbol{\theta}} = \boldsymbol{\Phi}^T \mathbf{Y}$$

Se a matriz $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ possuir determinante não nulo, e por isso for invertível, o vector dos parâmetros estimados será dado por

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{Y} \quad (3.52)$$

Designando $\mathbf{P} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1}$, obtém-se:

$$\hat{\boldsymbol{\theta}} = \mathbf{P} \left(\sum_{i=1}^N \boldsymbol{\varphi}(i)y(i) \right) \quad (3.53)$$

A dimensão da matriz \mathbf{P} é determinada pelo número de parâmetros do modelo n . Os valores de \mathbf{P} são proporcionais à variância da estimativa e, por esse facto, \mathbf{P} é designada por matriz de covariância.

Pode fazer-se uma interpretação estatística das estimativas efectuadas através do método dos mínimos quadrados. Supondo que os dados são gerados através do processo dado pela expressão seguinte, em que a saída é afectada por ruído

$$y = \boldsymbol{\Phi}\boldsymbol{\theta}_r + \boldsymbol{\vartheta}$$

onde $\boldsymbol{\theta}_r$ representa o vector contendo o valor correcto dos parâmetros, $\boldsymbol{\vartheta}$ é um vector de ruído branco com variância igual a σ^2 , sendo $\boldsymbol{\vartheta}$ independente de $\boldsymbol{\Phi}$, então a estimativa obtida através da equação 3.52 apresenta as seguintes características:

1- $E \left\{ \hat{\boldsymbol{\theta}} \right\} = \boldsymbol{\theta}_r$, em que $E \{ \cdot \}$ representa a esperança matemática. Esta propriedade do método dos mínimos quadrados significa que o desempenho do estimador melhora quando o número de observações tende para infinito, levando à convergência do estimador sem apresentar desvios. Isto só é possível porque se considera que o ruído tem média nula. Noutros casos a estimativa pode apresentar desvios. Esta característica permite também afirmar que este método só identifica sem desvios sistemas do tipo ARX, em que o polinómio $C(q^{-1}) = 1$ [98].

2- $cov \left\{ \hat{\boldsymbol{\theta}} \right\} = \sigma^2(\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} = \sigma^2 \mathbf{P}$, em que $cov \left\{ \hat{\boldsymbol{\theta}} \right\}$ representa a covariância. A variância dos parâmetros estimados é proporcional à variância do ruído [98].

3- Pode obter-se uma estimativa da variância do ruído através da expressão seguinte [98]:

$$s^2 = \frac{2J(\boldsymbol{\theta})}{N - n}$$

3.4.2 Implementação recursiva do método dos mínimos quadrados

A expressão 3.53 corresponde à implementação não recursiva do método dos mínimos quadrados. Neste tipo de implementação os dados provenientes do sistema são recolhidos antecipadamente procedendo-se depois à identificação do sistema com base no conteúdo dos mesmos.

Por vezes torna-se necessário efectuar a identificação do sistema em tempo-real, com os dados a serem recolhidos a cada iteração. É o que se verifica, por exemplo, para certos tipos de controladores adaptativos, como se verá na secção 3.5.

Quando é necessário fazer a estimativa em tempo-real, e no sentido de economizar tempo de processamento, torna-se necessária a utilização de um algoritmo recursivo

que possibilite a estimativa dos parâmetros do sistema para k observações, tendo por base a estimativa efectuada para $k - 1$ observações.

A equação 3.52 permite determinar a estimativa dos parâmetros do sistema. Para $k - 1$ pares de valores obtém-se

$$\hat{\boldsymbol{\theta}}(k-1) = (\boldsymbol{\Phi}(k-1)^T \boldsymbol{\Phi}(k-1))^{-1} \boldsymbol{\Phi}(k-1)^T \mathbf{Y}(k-1)$$

Na próxima observação as matrizes $\boldsymbol{\Phi}$ e \mathbf{Y} têm que ser actualizadas do seguinte modo

$$\boldsymbol{\Phi}(k) = \begin{bmatrix} \boldsymbol{\Phi}(k-1) \\ \boldsymbol{\varphi}^T(k) \end{bmatrix}$$

$$\mathbf{Y}(k) = \begin{bmatrix} \mathbf{Y}(k-1) \\ y(k) \end{bmatrix}$$

A estimativa para k observações será dada por

$$\hat{\boldsymbol{\theta}}(k) = (\boldsymbol{\Phi}^T(k) \boldsymbol{\Phi}(k))^{-1} \boldsymbol{\Phi}^T(k) \mathbf{Y}(k)$$

ou seja

$$\hat{\boldsymbol{\theta}}(k) = (\boldsymbol{\Phi}^T(k-1) \boldsymbol{\Phi}(k-1) + \boldsymbol{\varphi}^T(k) \boldsymbol{\varphi}(k))^{-1} (\boldsymbol{\Phi}^T(k-1) \mathbf{Y}(k-1) + \boldsymbol{\varphi}(k) y(k))$$

A versão recursiva da técnica dos mínimos quadrados ([98] e [99]) obtém-se a partir da equação anterior e é dada pelas equações seguintes

$$\left\{ \begin{array}{l} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k) \varepsilon(k) \\ \varepsilon(k) = y(k) - \boldsymbol{\varphi}^T(k) \hat{\boldsymbol{\theta}}(k-1) \\ \mathbf{K}(k) = \frac{\mathbf{P}(k-1) \boldsymbol{\varphi}(k)}{1 + \boldsymbol{\varphi}^T(k) \mathbf{P}(k-1) \boldsymbol{\varphi}(k)} \\ \mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k) \boldsymbol{\varphi}^T(k)) \mathbf{P}(k-1) \end{array} \right. \quad (3.54)$$

A versão recursiva do método dos mínimos quadrados, dada pelas equações 3.54, permite obter uma estimativa do valor dos parâmetros do sistema tendo por base o valor anterior, que é corrigido através de um termo proporcional ao erro $\varepsilon(k)$. Este sinal pode ser interpretado como o erro de estimativa do sinal $y(k)$. O vector $\mathbf{K}(k)$ pode ser interpretado como um conjunto de pesos que indica de que modo deve ser corrigida a estimativa efectuada na iteração anterior.

Para que seja possível aplicar a versão recursiva do método dos mínimos quadrados é necessário obter uma condição inicial para a matriz $\mathbf{P}(k)$. Em termos práticos faz-se $\mathbf{P}(0) = \mathbf{P}_0$, em que \mathbf{P}_0 é uma matriz diagonal do tipo $\alpha \mathbf{I}$. Um valor elevado de α (entre 100 e 1000) implica pouca confiança no estimador e origina variações rápidas na estimativa, enquanto que um valor pequeno de α (entre 1 e 10) indica alguma confiança no estimador e faz com que ele varie lentamente [99].

3.4.3 Identificação de parâmetros variáveis no tempo

A função de custo 3.47 atribui o mesmo peso a todas as observações do sistema. Se o sistema for invariante no tempo faz todo o sentido que assim seja. No entanto se esse não for o caso, a equação de estimativa 3.54 deixa de ser adequada pois o ganho $\mathbf{K}(k)$ diminui à medida que k cresce, o que significa que o ajuste possível dos parâmetros passa a ser cada vez menor [99]. Isto verifica-se porque, tal como já foi referido, este método proporciona um aumento constante da precisão dos valores estimados com o aumento do número de observações.

Em sistemas variantes no tempo a qualidade do estimador não está directamente relacionada com o número de observações pelo que é necessário um ganho elevado de modo a que o valor estimado se aproxime mais rapidamente do valor real dos parâmetros. Nestes casos torna-se necessário incrementar de modo sistemático e adequado o vector do ganho $\mathbf{K}(k)$ [99]. Existem várias formas para o conseguir. De seguida descrevem-se os seguintes métodos: método dos mínimos quadrados pesados com factor de esquecimento, método de selecção do factor de esquecimento em função do erro de estimativa, método dos mínimos quadrados com reinicialização da matriz de covariância e método dos mínimos quadrados com esquecimento direccionado.

Mínimos quadrados pesados com factor de esquecimento

Para que os dados relativos ao sistema sejam pesados de modo não uniforme durante o processo de identificação dos parâmetros pode adoptar-se a função de custo seguinte

$$J_\lambda(\hat{\boldsymbol{\theta}}) = \frac{1}{2} \sum_{i=1}^N \lambda^{N-i} (y(i) - \boldsymbol{\varphi}^T \hat{\boldsymbol{\theta}})^2 \quad (3.55)$$

onde λ se designa por factor de esquecimento e indica a rapidez com que a informação mais antiga é esquecida. O factor de esquecimento pode variar dentro do intervalo $]0, 1[$. Para $\lambda = 1$ obtêm-se a formulação “clássica” do método dos mínimos quadrados descrita anteriormente.

A função de custo pode ser reescrita sob a seguinte forma

$$J_\lambda(\hat{\boldsymbol{\theta}}) = \frac{1}{2} \mathbf{E}^T \boldsymbol{\Lambda} \mathbf{E} \quad (3.56)$$

onde \mathbf{E} é o vector de erros dado por 3.50 e $\boldsymbol{\Lambda}$ é uma matriz diagonal que contém os pesos.

$$\begin{aligned} \mathbf{\Lambda} &= \mathbf{\Lambda}(k) = \begin{bmatrix} \lambda^{k-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda^{k-2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} w(1) & 0 & 0 & 0 & 0 & 0 \\ 0 & w(2) & 0 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & w(k-2) & 0 & 0 \\ 0 & 0 & 0 & 0 & w(k-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & w(k) \end{bmatrix} \end{aligned}$$

em que

$$w(k-1) = \lambda w(k)$$

com $w(k) = 1$ e $k = 1, \dots, N$ sendo o índice das observações obtidas a partir do sistema.

Substituindo \mathbf{E} por 3.51 na função de custo 3.56 obtém-se

$$2J(\hat{\boldsymbol{\theta}}) = (\mathbf{Y} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\theta}})^T \mathbf{\Lambda} (\mathbf{Y} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\theta}}) = \mathbf{Y}^T \mathbf{\Lambda} \mathbf{Y} - \mathbf{Y}^T \hat{\boldsymbol{\Phi}} \mathbf{\Lambda} \hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\theta}}^T \mathbf{\Lambda} \mathbf{Y} + \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\theta}}^T \mathbf{\Lambda} \hat{\boldsymbol{\Phi}} \hat{\boldsymbol{\theta}}$$

$$\frac{\delta J}{\delta \hat{\boldsymbol{\theta}}} = 0 \iff -\mathbf{Y}^T \hat{\boldsymbol{\Phi}} \mathbf{\Lambda} - \hat{\boldsymbol{\Phi}}^T \mathbf{\Lambda} \mathbf{Y} + 2\hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}} \mathbf{\Lambda} \hat{\boldsymbol{\theta}} = 0 \iff \hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}} \mathbf{\Lambda} \hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\Phi}}^T \mathbf{\Lambda} \mathbf{Y}$$

$$\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\Phi}}^T \mathbf{\Lambda} \hat{\boldsymbol{\Phi}})^{-1} \hat{\boldsymbol{\Phi}}^T \mathbf{\Lambda} \mathbf{Y} \quad (3.57)$$

A expressão da estimativa de $\boldsymbol{\theta}$ dada pela equação 3.57 é designada por mínimos quadrados pesados [99]. Tal como para o método original dos mínimos quadrados, também para este é possível obter uma versão recursiva.

Considerando $\mathbf{R}(k)$ e $\mathbf{X}(k)$ dados por

$$\mathbf{R}(k) = \hat{\boldsymbol{\Phi}}^T(k) \mathbf{\Lambda}(k) \hat{\boldsymbol{\Phi}}(k) = \sum_{i=1}^k w(i) \boldsymbol{\varphi}(i) \boldsymbol{\varphi}^T(i)$$

$$\mathbf{X}(k) = \hat{\boldsymbol{\Phi}}^T(k) \mathbf{\Lambda}(k) \mathbf{Y}(k) = \sum_{i=1}^k w(i) \boldsymbol{\varphi}(i) y(i)$$

em que $w(i)$ representa os pesos que constituem a diagonal da matriz $\mathbf{\Lambda}$, pode escrever-se a equação para a estimativa de $\boldsymbol{\theta}$ do seguinte modo

$$\hat{\boldsymbol{\theta}}(k) = (\mathbf{R}(k))^{-1} \mathbf{X}(k) \quad (3.58)$$

No sentido de obter uma expressão recursiva para $\hat{\boldsymbol{\theta}}$ começa-se por analisar $\mathbf{R}(k)$ e $\mathbf{X}(k)$.

$$\mathbf{R}(k) = \sum_{i=1}^{k-1} w(i)\boldsymbol{\varphi}(i)\boldsymbol{\varphi}^T(i) + w(k)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

como $w(k) = 1$ pode escrever-se

$$\mathbf{R}(k) = \mathbf{R}(k-1)w(k-1) + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

e como $w(k-1) = \lambda w(k)$ pode escrever-se

$$\mathbf{R}(k) = \mathbf{R}(k-1)\lambda w(k) + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

$$\mathbf{R}(k) = \mathbf{R}(k-1)\lambda + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) \quad (3.59)$$

Do mesmo modo para $\mathbf{X}(k)$ pode escrever-se

$$\mathbf{X}(k) = \mathbf{X}(k-1)\lambda + \boldsymbol{\varphi}(k)y(k)$$

e substituindo em 3.58 obtém-se

$$\hat{\boldsymbol{\theta}}(k) = (\mathbf{R}(k))^{-1}(\mathbf{X}(k-1)\lambda + \boldsymbol{\varphi}(k)y(k))$$

Com base nesta expressão e através de manipulações sucessivas (ver [99], por exemplo) obtém-se o conjunto de expressões que permitem a implementação recursiva do método dos mínimos quadrados com factor de esquecimento que são apresentadas de seguida:

$$\left\{ \begin{array}{l} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\varepsilon(k) \\ \varepsilon(k) = y(k) - \boldsymbol{\varphi}^T(k)\hat{\boldsymbol{\theta}}(k-1) \\ \mathbf{K}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\varphi}(k)}{\lambda\mathbf{I} + \boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)} \\ \mathbf{P}(k) = \frac{1}{\lambda}(\mathbf{I} - \mathbf{K}(k)\boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)) \end{array} \right. \quad (3.60)$$

A função de custo 3.56 atribui peso 1 ao par de valores recolhidos mais recentemente, atribuindo ao par de valores obtido n amostras atrás um peso de λ^n , por isso este método também é designado por esquecimento exponencial.

O factor de esquecimento λ é normalmente uma constante de valor ligeiramente inferior a 1 (tipicamente entre 0.95 e 0.999). Pode obter-se o número de resíduos [99] N_m utilizados em cada iteração para a actualização do valor dos parâmetros estimados através da expressão

$$N_m = \frac{1}{1 - \lambda}$$

Para $\lambda = 0.95$ (valor utilizado para variações rápidas do valor dos parâmetros) e para $\lambda = 0.999$ (valor utilizado para variações lentas do valor dos parâmetros) o tempo médio de utilização de cada amostra será de 20 e de 1000 amostras [99], respectivamente. A escolha do valor λ corresponde assim a um compromisso entre a capacidade do estimador seguir variações rápidas e a qualidade das estimativas a longo prazo.

Uma desvantagem da utilização deste método é que ele provoca o aumento da variância das estimativas, em resultado do aumento da matriz $\mathbf{P}(k)$ quando não há variações dos parâmetros. Ou seja se existirem longos períodos de tempo durante os quais o sistema não sofre qualquer excitação ($\varphi(k)$ muito pequeno ou nulo), o estimador vai “esquecendo” progressivamente os valores “autênticos” do vector dos parâmetros e os erros de estimativa vão aumentando. Esta situação é denominada *estimator windup* e pode provocar oscilação em malhas de controlo que operem em tempo-real com recurso a este tipo de estimadores.

Existem vários métodos para lidar com este problema, todos eles baseados na necessidade de manter a matriz $\mathbf{P}(k)$ dentro de limites bem definidos. Alguns desses métodos vão ser descritos de seguida.

Seleccção do factor de esquecimento em função do erro de estimativa

Este método propõe a seleccção do factor de esquecimento em função do valor absoluto do erro de estimativa. Quando o erro de estimativa é pequeno, como acontece quando o sistema não sofre excitação, λ deve ter um valor próximo de 1, nesta situação, como referido anteriormente, o estimador opera com base num número elevado de amostras. Se o valor absoluto do erro de estimativa for grande o “ganho” do estimador deve ser aumentado, diminuindo o valor de λ . Para a aplicação bem sucedida deste método é necessária a escolha correcta do valor limiar do erro de estimativa que provoca a alteração de λ . Por outro lado a escolha do valor limiar tem ainda que ter em conta a amplitude do ruído que afecta o sistema. Se o valor limiar for demasiado pequeno a matriz de covariância vai divergir [99]. No caso dele ser demasiado grande a convergência para o valor real dos parâmetros torna-se mais demorada. Esta solução não é, no entanto, a mais indicada para a identificação de sistemas variantes no tempo, uma vez que a diagonal da matriz de covariância tem nestes sistemas um valor baixo o que dificulta o acompanhamento da evolução dos parâmetros reais por parte do estimador.

Mínimos quadrados com reinicialização da matriz de covariância

Este método baseia-se no controlo dos valores da matriz de covariância. É utilizado um factor de esquecimento sempre igual a 1 e força-se um dado valor para a matriz de covariância sempre que o erro de estimativa exceder um determinado limiar. Deste modo o ganho do estimador é aumentado quando o erro de estimativa excede um determinado valor.

A reinicialização da matriz de covariância permite prevenir o problema de *windup*. As considerações relativas à determinação do valor limiar do erro de estimativa são no entanto idênticas às referidas anteriormente.

Mínimos quadrados com esquecimento direccionado

No sentido de ultrapassar as limitações dos métodos descritos anteriormente, Hagglund [107] propõe o método de esquecimento direccionado baseado no princípio de que o factor de esquecimento deve ser aplicado de modo a manter constante uma determinada quantidade de informação sempre que os parâmetros permaneçam constantes. A técnica do esquecimento direccionado é aqui aplicada ao método dos mínimos quadrados de acordo com a formulação apresentada em [99].

Como $\mathbf{P}(k) = (\mathbf{R}(k))^{-1}$ a equação 3.59 pode ser escrita sob a forma

$$\mathbf{P}(k) = (\mathbf{R}(k-1)\lambda + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k))^{-1}$$

ou ainda

$$(\mathbf{P}(k))^{-1} = (\mathbf{P}(k-1))^{-1}\lambda + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

que é equivalente a

$$(\mathbf{P}(k))^{-1} = (\mathbf{P}(k-1))^{-1} - (1 + \lambda)(\mathbf{P}(k-1))^{-1} + \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

Nesta expressão pode ver-se que a actualização da inversa da matriz de covariância é efectuada com base na nova informação disponível (termo $\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$) e num termo proporcional ao seu valor anterior. Assim qualquer que seja a nova informação disponível existe sempre esquecimento da informação anterior. Para evitar este comportamento utiliza-se a expressão seguinte para o cálculo de $(\mathbf{P}(k))^{-1}$,

$$(\mathbf{P}(k))^{-1} = (\mathbf{P}(k-1))^{-1} + \alpha(k)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)$$

Se $\alpha(k)$ for relacionado com o factor de esquecimento pode dizer-se que o mesmo é aplicado apenas em relação à nova informação que é utilizada pelo estimador e não em relação à informação existente até aí (daí o nome de esquecimento direccionado).

Uma vez mais como $\mathbf{P}(k) = (\mathbf{R}(k))^{-1}$

$$\mathbf{P}(k) = (\mathbf{R}(k-1)^{-1} + \alpha(k)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k))^{-1}$$

e a partir daqui obtêm-se as equações de base para o cálculo recursivo dos parâmetros.

O valor de $\alpha(k)$ pode ser calculado, de acordo com [97], de forma a que

$$\alpha(k) = \begin{cases} \lambda_{df} - \frac{1-\lambda_{df}}{\boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k)} & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) > 0 \\ 1 & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) = 0 \end{cases}$$

Assim, o conjunto de equações para a implementação deste método é o seguinte

$$\left\{ \begin{array}{l} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\varepsilon(k) \\ \varepsilon(k) = y(k) - \boldsymbol{\varphi}^T(k)\hat{\boldsymbol{\theta}}(k-1) \\ \mathbf{K}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\varphi}(k)}{\mathbf{I} + \boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)\alpha(k)} \\ \mathbf{P}(k) = (\mathbf{I} - \alpha(k)\mathbf{K}(k)\boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)) \\ \alpha(k) = \begin{cases} \lambda_{df} - \frac{1-\lambda_{df}}{\boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k)} & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) > 0 \\ 1 & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) = 0 \end{cases} \end{array} \right. \quad (3.61)$$

Este método foi o escolhido para fazer a identificação em tempo-real dos parâmetros do sistema necessária para a implementação dos controladores adaptativos distribuídos descritos na secção 3.6. Embora ele seja um pouco mais complexo, e por isso requiera uma maior capacidade de processamento, permite obviar o problema do *estimator windup*.

3.4.4 Identificação de sistemas em malha fechada

Certos tipos de controladores adaptativos, como é o caso dos reguladores auto-ajustáveis (ver a secção 3.5.1), necessitam que a identificação do sistema seja efectuada em tempo-real a partir de dados recolhidos do sistema em malha fechada.

A estimativa directa dos parâmetros de um sistema utilizando dados obtidos em malha fechada, recorrendo às técnicas descritas anteriormente, coloca alguns problemas. Estes problemas estão relacionados com o mecanismo de realimentação da malha que faz com que os dados utilizados contêm menos informação sobre o sistema em malha aberta. Isto porque o processo de realimentação pode introduzir ambiguidade na relação entre o sinal de entrada e o sinal de saída [97].

Certos tipos de realimentação podem introduzir dependência entre o ruído e o sinal de entrada do sistema [108], como se pode ver na figura 3.7. Nestas condições não é possível determinar, de modo exacto, os parâmetros do sistema.

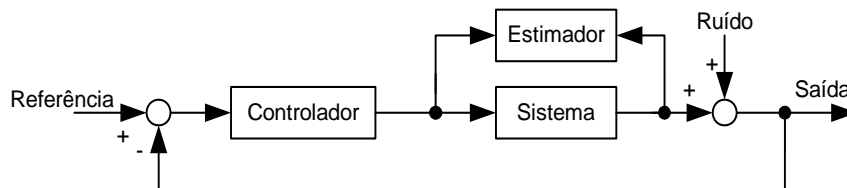


Figura 3.7: Sistema em malha fechada.

O problema descrito acontece essencialmente para sistemas em que a realimentação é linear e de ordem pequena [96]. Para recuperar a identificabilidade do estimador basta que o mecanismo de realimentação seja mais complexo, por exemplo que ele seja não-linear ou variante no tempo. Nos casos em que não é possível alterar o mecanismo de realimentação pode ainda assim recuperar-se a identificabilidade do estimador através da introdução de uma fonte de ruído independente, por vezes designada por *dither* [97]. Esta deve ser capaz de fornecer ao sistema aquilo que se designa por excitação persistente, que permite activar todos os modos de funcionamento do sistema e deste modo estimar os parâmetros de forma correcta. A figura 3.8 apresenta o diagrama de blocos de um estimador a operar em malha fechada com introdução de *dither*.

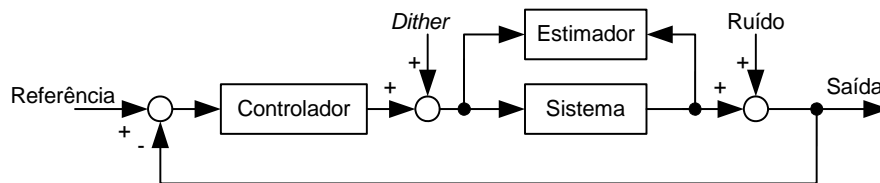


Figura 3.8: Diagrama de blocos de um sistema em malha fechada com introdução de *dither*.

Existem outras abordagens para a estimativa de parâmetros em malha fechada, como os métodos indirectos ou os métodos entrada-saída (ver por exemplo [108]). No entanto a identificação directa, embora tenha as suas limitações, continua a ser considerada como o método de referência porque funciona independentemente da complexidade do controlador e não requer conhecimento prévio do tipo de realimentação da malha. Além disso recorre a algoritmos bem conhecidos e permite obter consistência e precisão óptima se a estrutura escolhida para o modelo for a correcta (tanto para o sistema como para o ruído). Os outros métodos para além de serem mais complexos não permitem obter tanta precisão como o método de identificação directa.

3.4.5 Implementação do processo de identificação

Nos ensaios apresentados no capítulo 4 é efectuada a identificação de sistemas afectados por tempo morto fraccionário com o objectivo de determinar os parâmetros desses sistemas para depois sintetizar o respectivo controlador. A estrutura do modelo discreto escolhida para modelar o sistema com e sem tempo morto fraccionário é a estrutura ARX por ser adequada para utilização com a técnica de controlo por posicionamento de pólos. Esta técnica de controlo será descrita adiante.

Para a classe de modelos ARX a equação de predição é dada por

$$\hat{y}(k+1) = q^{-t_d} B(q^{-1})u(k) + (1 - A(q^{-1}))y(k) \quad (3.62)$$

a partir da qual se obtém a equação de regressão

$$\hat{y}(k+1) = \boldsymbol{\theta}^T \boldsymbol{\varphi}(k)$$

sendo os vectores φ e θ dados por:

$$\varphi(k) = [y(k), \dots, y(k - n_a), u(k - t_d - 1), \dots, u(k - t_d - n_b)]$$

$$\theta = [-a_1, \dots, -a_{n_a}, b_1, \dots, b_{n_b}]$$

onde t_d é o tempo morto discreto e n_a e n_b a ordem dos polinómios A e B , respectivamente.

Os modelos discretos usualmente utilizados para sistemas de 1^a e de 2^a ordem, que não têm em conta o tempo morto fraccionário são dados pelas expressões 3.63 e 3.64 respectivamente (ver secção 3.3.3):

$$G(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} \quad (3.63)$$

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \quad (3.64)$$

Os modelos propostos para sistemas de 1^a e 2^a ordem que têm em conta o tempo morto fraccionário são dados pelas expressões 3.65 e 3.66 respectivamente (ver secção 3.3.4):

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1}} \quad (3.65)$$

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}}{1 + a_1 q^{-1} + a_2 q^{-2}} \quad (3.66)$$

Para proceder à identificação é necessário determinar os parâmetros do sistema necessários e os regressores. Comparando cada um dos modelos anteriores com a equação de predição 3.62 obtêm-se as tabelas seguintes onde se apresentam os parâmetros e os regressores para cada um dos modelos. Para os modelos que não consideram o tempo morto fraccionário obtêm-se a tabela 3.2 e para os modelos que consideram o tempo morto fraccionário obtêm-se a tabela 3.3.

Sistema	Parâmetros	Regressores
1 ^a ordem	$\theta = [-a_1, b_1]$	$\varphi_k = [y_k, u_{k-1}]$
2 ^a ordem	$\theta = [-a_1, -a_2, b_1, b_2]$	$\varphi_k = [y_k, y_{k-1}, u_{k-1}, u_{k-2}]$

Tabela 3.2: Modelos de 1^a e de 2^a ordem de sistemas sem tempo morto fraccionário.

Sistema	Parâmetros	Regressores
1 ^a ordem	$\theta = [-a_1, b_1, b_2]$	$\varphi_k = [y_k, u_{k-1}, u_{k-2}]$
2 ^a ordem	$\theta = [-a_1, -a_2, b_1, b_2, b_3]$	$\varphi_k = [y_k, y_{k-1}, u_{k-1}, u_{k-2}, u_{k-3}]$

Tabela 3.3: Modelos de 1^a e de 2^a ordem de sistemas com tempo morto fraccionário.

A estimativa dos parâmetros dos modelos é efectuada utilizando a versão recursiva do algoritmo dos mínimos quadrados com factor de esquecimento direccionado dada pelas equações 3.67.

$$\left\{ \begin{array}{l} \hat{\boldsymbol{\theta}}(k) = \hat{\boldsymbol{\theta}}(k-1) + \mathbf{K}(k)\varepsilon(k) \\ \varepsilon(k) = y(k) - \boldsymbol{\varphi}^T(k)\hat{\boldsymbol{\theta}}(k-1) \\ \mathbf{K}(k) = \frac{\mathbf{P}(k-1)\boldsymbol{\varphi}(k)}{\mathbf{I} + \boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)\boldsymbol{\varphi}(k)\alpha(k)} \\ \mathbf{P}(k) = (\mathbf{I} - \alpha(k)\mathbf{K}(k)\boldsymbol{\varphi}^T(k)\mathbf{P}(k-1)) \\ \alpha(k) = \begin{cases} \lambda_{df} - \frac{1-\lambda_{df}}{\boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k)} & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) > 0 \\ 1 & \text{se } \boldsymbol{\varphi}(k)\mathbf{P}(k-1)\boldsymbol{\varphi}^T(k) = 0 \end{cases} \end{array} \right. \quad (3.67)$$

3.5 Controlo adaptativo “clássico”

Um controlador adaptativo é um controlador que possui parâmetros ajustáveis e um mecanismo que permite fazer o ajuste desses parâmetros [96]. Por este facto são úteis em aplicações em que a dinâmica do sistema varia ao longo do tempo surgindo assim a necessidade do ajuste dos parâmetros. A figura 3.9 apresenta o diagrama de blocos de um controlador adaptativo genérico.

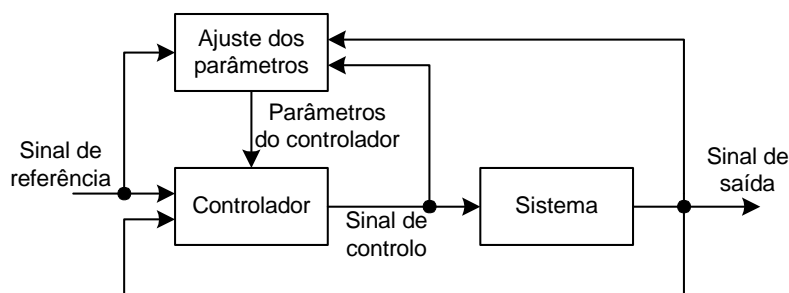


Figura 3.9: Diagrama de blocos de um controlador adaptativo genérico.

Como se pode ver na figura 3.9 o controlador adaptativo é composto por duas malhas: uma malha realimentada “comum”, constituída pelo sistema e pelo controlador, e uma malha para o ajuste dos parâmetros do controlador. A existência desta segunda malha torna o controlador não-linear. O ajuste dos parâmetros do controlador pode ser efectuada de diferentes modos dando origem a diferentes tipos de adaptação.

3.5.1 Tipos de controladores adaptativos

Os esquemas mais utilizados para fazer o ajuste dos parâmetros do controlador são, de acordo com a abordagem de Åström [96], os seguintes: escalonamento de ganho, controlo adaptativo por modelo de referência, controlo auto-ajustável e controlo óptimo. Refira-

-se ainda a técnica de controlo adaptativo inverso [106].

A utilização da nomenclatura “adaptativo” ou “auto-ajustável” não é consensual por razões históricas. O termo auto-ajustável começou por ser utilizado para designar apenas a regulação inicial do controlador de um sistema que não sendo variável com o tempo era desconhecido. O termo adaptativo refere-se ao processo de ajuste contínuo dos parâmetros do controlador. Actualmente existem autores que consideram que os dois termos podem ser utilizados indiscriminadamente com o sentido dado ao termo “adaptativo”. Neste documento utiliza-se o termo “adaptativo” com o sentido unânime e só se utiliza o termo “auto-ajustável” para designar um dos tipos de adaptação possível, de acordo com a sua designação em inglês (*self-tuning*).

Controlo por escalonamento de ganho

O controlo por escalonamento de ganho (*gain scheduling* na designação em inglês) é utilizado quando é possível relacionar variáveis mensuráveis do sistema com a dinâmica do processo. Neste caso as variáveis podem ser utilizadas para proceder à alteração dos parâmetros do controlador. A figura 3.10 apresenta o diagrama de blocos de um controlador adaptativo por escalonamento de ganho.

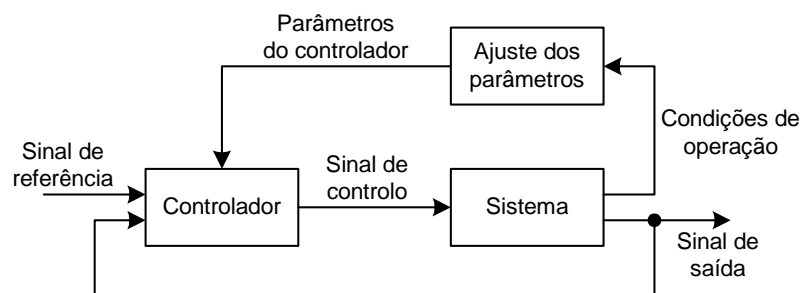


Figura 3.10: Diagrama de blocos de um controlador por escalonamento de ganho.

Este tipo de adaptação pode ser visto como um mapeamento dos parâmetros do sistema sobre os parâmetros do controlador e pode ser implementado sob a forma de uma simples tabela ou função. Pelo facto do mapeamento entre a saída do sistema e os parâmetros do controlador ser fixo, certos autores não consideram este tipo de esquema como sendo adaptativo, embora ele se enquadre na definição dada no início desta secção.

Este tipo de controlador é muito utilizado na indústria de processos, em sistemas para controlo de voo e em robótica [96] e [97].

Controlo adaptativo por modelo de referência

A técnica de controlo adaptativo por modelo de referência pressupõe a existência de um modelo que descreve o modo como o sistema deve reagir aos sinais de referência. Neste caso o ajuste dos parâmetros é feito de modo a que a diferença entre a saída do sistema e a saída do modelo de referência seja tão pequena quanto possível. A figura 3.11 apresenta o diagrama de blocos deste tipo de controlador.

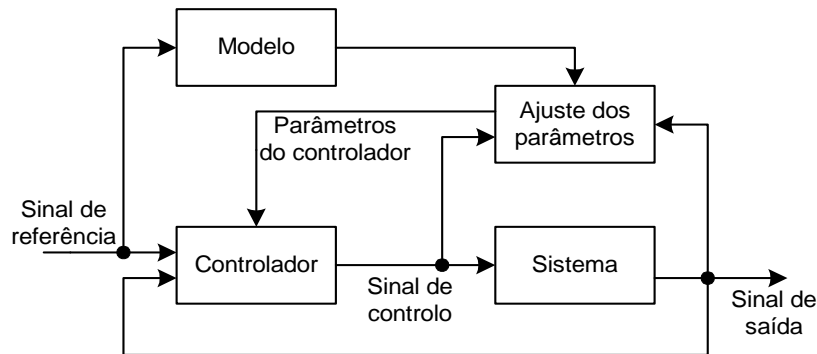


Figura 3.11: Diagrama de blocos de um controlador adaptativo por modelo de referência.

Esta técnica foi inicialmente aplicada para controlo de voo [96].

Controlo auto-ajustável

Num controlador auto-ajustável (STR da designação em inglês *Self-Tuning Regulator*) os parâmetros do controlador são calculados a partir do valor estimado dos parâmetros do sistema, de acordo com as especificações impostas para a aplicação, como se pode ver na figura 3.12. O bloco de síntese do controlador é responsável pela determinação em tempo-real da solução do problema de controlo (dado um conjunto de especificações) para um sistema cujos parâmetros são conhecidos. Os parâmetros do sistema são obtidos pelo estimador que também actua em tempo-real e em malha fechada, como se pode ver no diagrama de blocos.

Esta técnica é muito flexível permitindo a selecção de diferentes métodos para a síntese do controlador e para o estimador dos parâmetros do sistema. Algumas áreas de aplicação desta técnica são a indústria de processos e a indústria automóvel [97] e [96].

Controlo óptimo

As técnicas de controlo apresentadas anteriormente estimam os parâmetros do sistema e de controlo e depois consideram-nos como estando correctos. Não são consideradas incertezas nesses parâmetros. A alternativa será a formulação do problema de controlo de um modo mais genérico em que o processo e os seus parâmetros, assim como o

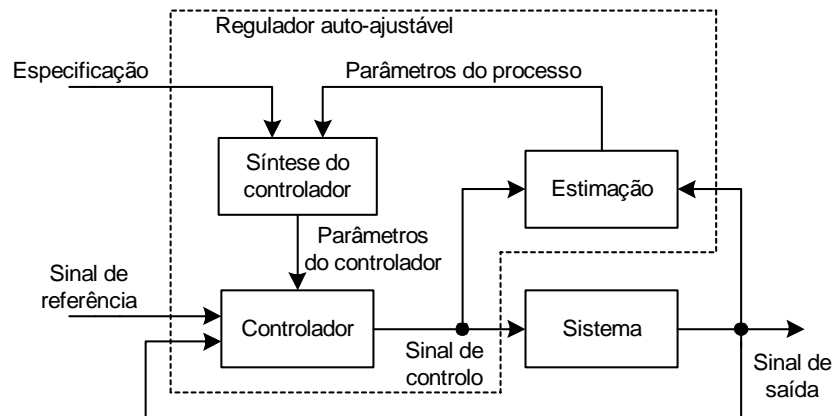


Figura 3.12: Diagrama de blocos de um regulador auto-ajustável.

ambiente em que ele opera, são tratados através de modelos estatísticos. O critério de controlo é formulado de modo a minimizar uma dada função de custo associada à operação do controlador. A figura 3.13 apresenta o diagrama de blocos de um regulador óptimo.

Este esquema é constituído por duas malhas: a malha de optimização que gera os parâmetros para o controlador de acordo com as observações do sistema e com o critério de optimização, e a malha de controlo que implementa o controlador. O método de optimização não separa os parâmetros dos estados do sistema o que torna a resolução do problema de controlo mais complexa. Este tipo de controladores promove simultaneamente um bom desempenho de controlo e erros de estimativa pequenos.

Esta técnica começou por ser aplicada em aeronáutica mas hoje em dia aplica-se também no controlo de processos e em robótica [109].

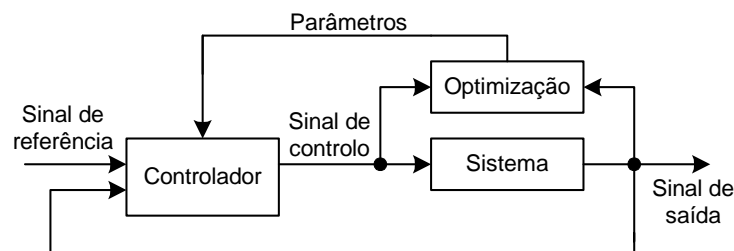


Figura 3.13: Diagrama de blocos de um regulador óptimo.

Controlo adaptativo inverso

Esta técnica de controlo propõe uma abordagem diferente das anteriores. A ideia é actuar sobre o sistema através de um controlador que tenha uma função de transferência inversa da do sistema de modo a minimizar o erro de comando. A figura 3.14 apresenta

o conceito do controlo adaptativo inverso através de um diagrama de blocos.

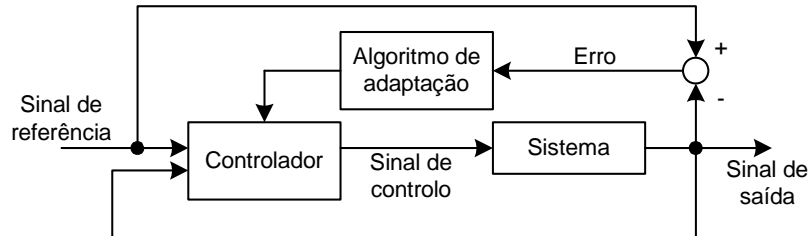


Figura 3.14: Diagrama de blocos de um controlador adaptativo inverso básico.

O objectivo é fazer com que o sistema siga o sinal de referência. Os parâmetros do controlador são ajustados através de um algoritmo de adaptação de modo a obter uma função inversa da do sistema. Esse algoritmo utiliza o valor do erro de comando para obter os parâmetros do controlador de modo a minimizar o erro quadrático de comando.

Exemplos de aplicação desta técnica são o controlo de processos e aplicações biomédicas [106].

Esta técnica pode apresentar alguns problemas. Se o sistema for de fase não-mínima, possuindo zeros fora do círculo unitário, o modelo inverso possuirá pólos também fora do círculo unitário sendo portanto instável, o que levanta problemas para a realização do controlador.

Comutação de controladores auto-ajustáveis

Normalmente as especificações relativas ao controlador auto-ajustável são fixas e os parâmetros são ajustados de modo a cumprir essas especificações de acordo com as variações que o sistema sofre ao longo do tempo. No entanto existem situações em que durante o ciclo de operação do sistema pode ser vantajosa a utilização de diferentes especificações para o controlador, seleccionadas de acordo com critérios que podem ser internos ou externos ao controlador. Nesta situação temos um sistema com características variáveis no tempo e um conjunto de especificações, também elas variáveis.

Para este tipo de casos propõe-se a utilização de um conjunto de controladores auto-ajustáveis, um para cada conjunto de especificações, procedendo-se à comutação entre eles de acordo com um critério de selecção predefinido. O critério pode estar, ou não, dependente da operação da malha de controlo. O diagrama de blocos desta estratégia é apresentado a figura 3.15, na qual o sinal de referência foi omitido para facilitar a legibilidade.

3.5.2 Exemplos de técnicas de controlo adaptativas

Nesta secção são apresentadas as seguintes técnicas utilizadas em controlo adaptativo: controlo preditivo generalizado (*Generalized Predictive Control - GPC*), *Multis-*

teq *Multivariable Adaptive Regulator* (MUSMAR), *Linear Quadratic Gaussian* (LQG) e controlo por Posicionamento de Pólos (PP).

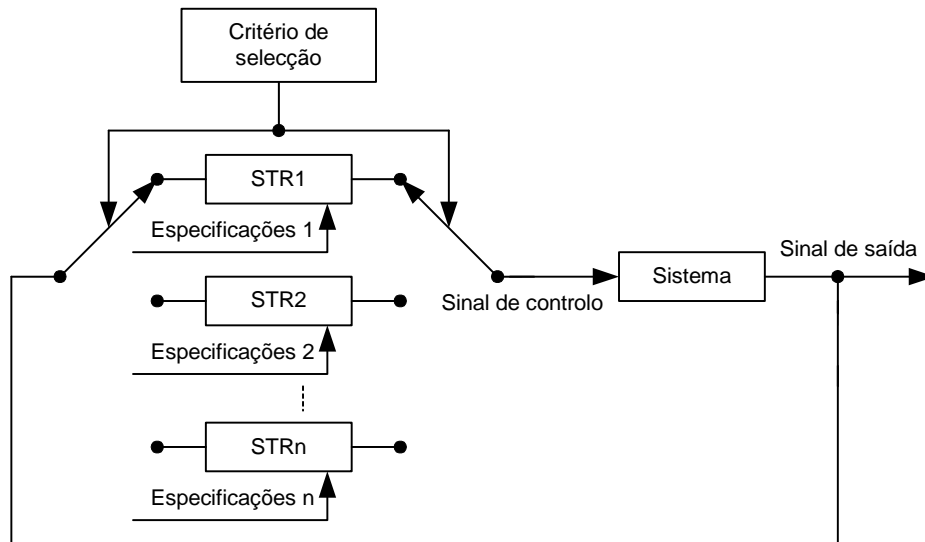


Figura 3.15: Diagrama de blocos de um sistema controlado por vários reguladores auto-ajustáveis dependendo de um critério de selecção.

Existem outras técnicas de controlo adaptativo para além das que foram referidas. Van Doren [110], Aström [96] e Wellstead [97] apresentam exemplos de aplicações industriais de várias técnicas de controlo adaptativo, assim como controladores adaptativos industriais como o EXACT da FOXBORO, o *BrainWave* da *Universal Dynamics Technologies* e o *CyboCon* da *CyboSoft*.

Controlo preditivo generalizado

A técnica de controlo preditivo generalizado ([111], [112], [113] e [114]) enquadra-se no grupo de métodos de controlo preditivo, que se utilizam em controladores auto-ajustáveis em que o problema de controlo é de regulação, ou seja, de rejeição das perturbações. Esta técnica é considerada uma das mais bem sucedidas para o controlo robusto de sistemas dinâmicos [115]. Inicialmente era utilizada apenas em controlo de processos devido à grande capacidade de processamento que era requerida mas com a diminuição do custo dos processadores e o aumento da sua capacidade de processamento (a partir do final da década de 80) a sua aplicação pôde estender-se a outras áreas.

Os elementos base da técnica GPC, tal como para qualquer controlador preditivo, são: um modelo de predição, uma função de custo e uma função de controlo. O modelo de predição serve para estimar o valor futuro da saída do sistema, tendo por base as entradas e saídas passadas e o valor futuro do sinal de controlo, visto que o cálculo é feito para um determinado horizonte temporal. O sinal de controlo a aplicar é determinado através da optimização de um índice de desempenho apropriado, de modo a que a saída do sistema siga o sinal de comando da melhor maneira possível.

O índice de desempenho é normalmente uma função quadrática do erro de comando previsto e dos incrementos de controlo. A optimização é efectuada para cada ciclo de controlo e da sequência de valores determinada para o sinal de controlo só se utiliza o primeiro, sendo o processo repetido no ciclo de controlo seguinte. Os parâmetros de projecto são: N_1 e N_2 o horizonte temporal associado ao cálculo da saída do sistema, N_u o horizonte temporal associado ao sinal de controlo e λ a penalização associada ao sinal de controlo na função de custo. Todos estes parâmetros podem ser alterados. O diagrama de blocos de um sistema GPC é apresentado na figura 3.16.

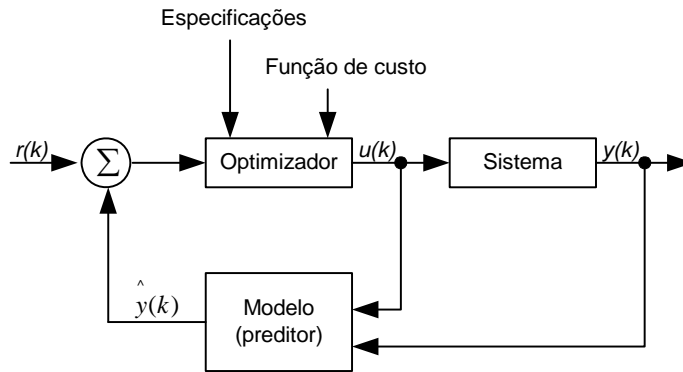


Figura 3.16: Diagrama de blocos do controlador GPC.

Se o sistema for descrito por um modelo do tipo ARIMAX

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k-1) + C(q^{-1})\frac{e(k)}{\Delta}$$

em que, $y(k)$ e $u(k)$ representam o sinal de saída e de entrada do sistema, $e(k)$ é ruído branco e d o tempo morto do sistema. Os polinómios $A(q^{-1})$, $B(q^{-1})$ e $C(q^{-1})$ são dados pelas equações 3.29 e $\Delta = 1 - q^{-1}$, como referido na secção 3.3.2.

A função de custo utilizada nesta técnica de controlo é do tipo

$$J(N_1, N_2, N_u, \lambda) = E \left\{ \sum_{j=N_1}^{N_2} \left[\hat{y}(k+j) - r(k+j) \right]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(k+j-1)]^2 \right\} \quad (3.68)$$

em que $N_y = N_2 - N_1$ é o horizonte de predição para a saída, N_u é o horizonte de controlo e λ é a sequência de pesos para o sinal de controlo. Como se pode ver pela equação anterior são penalizados os erros de comando sobre o horizonte de predição e a energia de controlo ao longo do horizonte de controlo. Considera-se que os incrementos de controlo para lá do horizonte de controlo são nulos. O objectivo é determinar a sequência de controlo óptima $\bar{u}(k)$ de tal modo que a função de custo seja minimizada. Os parâmetros de projecto N_y , N_u e λ são seleccionados de acordo com critérios relacionados com o desempenho e a estabilidade do sistema. Como foi referido anteriormente só é aplicado ao sistema o primeiro valor da sequência $\bar{u}(k)$. Os restantes são ignorados

e o processo de optimização é repetido a cada ciclo de controlo. Esta técnica requer por isso uma velocidade e capacidade de processamento elevadas.

***MU*ltiStep *MU*ltivariable *AD*aptive *RE*gulator**

Esta técnica de controlo foi proposta no início dos anos 80. Tal como o GPC é uma técnica de controlo preditivo. Ela proporciona melhores resultados do que o GPC no caso em que há desvio entre a estrutura do modelo escolhida e o sistema real [116] porque se baseia em estimativas obtidas por vários preditores.

A técnica MUSMAR([117] e [118]) é baseada num conjunto de vários modelos preditivos que cobrem o comportamento futuro do sistema dentro do horizonte de predição. O seu funcionamento pode “resumir-se” a um conjunto de estimadores RLS que partilham um regressor comum e operam em paralelo, e um mecanismo de realimentação “simples” que é actualizado a cada iteração. Tem a vantagem de ser insensível a atrasos do sistema.

Considerando que o sistema é descrito por um modelo do tipo ARMAX

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k-1) + C(q^{-1})e(k)$$

em que, $y(k)$ e $u(k)$ representam o sinal de saída e de controlo do sistema, respectivamente, $e(k)$ é ruído branco e d o tempo morto do sistema. Os polinómios $A(q^{-1})$, $B(q^{-1})$ e $C(q^{-1})$ são dados pelas equações 3.29.

A função de custo é dada por

$$J(T) = E \left[\frac{1}{T} \sum_{i=0}^{T-1} y^2(k+i+1) + \rho u^2(k+i) | y^k, u^{k-1} \right]$$

em que T é o horizonte de previsão, $E[. | y^k, u^{k-1}]$ é a média condicionada pelas observações de y^k e u^{k-1} até ao instante k e $\rho \geq 0$ é a penalização introduzida. Os sinais $y(k)$ e $u(k)$ representam, a saída e a entrada do sistema, respectivamente.

A minimização da função de custo anterior é efectuada através de um conjunto de preditores dados por

$$\hat{y}(k+1) = \theta_i u(k) + \Psi'_i(k)$$

$$\hat{u}(k+i-1) = \mu_{i-1} u(k) + \Phi'_{i-1} s(k)$$

em que $\hat{y}(k+1)$ e $\hat{u}(k+i-1)$ são os valores estimados através de RLS, de $y(k+1)$ e de $u(k+i-1)$, respectivamente e $s(k)$ é um vector designado por pseudo-estado [116] que é dado por

$$s(k) = [y(k) \dots y(k-n_a+1) u(k-1) \dots u(k-n_b)]'$$

A estrutura do controlador é definida pelas entradas de $s(k)$. Também podem ser incluídas amostras de outras variáveis, como perturbações mensuráveis, por exemplo. Normalmente também são incluídas amostras do sinal de referência na variável $s(k)$.

Os coeficientes θ_i , μ_{i-1} e os vectores Ψ_i e Φ_{i-1} representam os parâmetros que devem ser estimados em tempo-real. Para cada valor de i , com $i = 1, \dots, T$, este conjunto de parâmetros descreve o comportamento dinâmico do sistema entre os instantes $k + 1$ e $k + T$, assumindo que o mecanismo de realimentação na malha permanece constante. Os valores de n_a e n_b são escolhidos de acordo com a ordem do sistema.

A função de controlo que se obtém depois do processo de minimização utilizando os preditores descritos é a seguinte

$$u(k) = F' s(k) + \eta(k)$$

em que F é dado por

$$F = - \frac{\sum_{i=1}^T \theta_i \Psi_i + \rho \sum_{i=1}^{T-1} \mu_i \Phi_i}{\sum_{i=1}^T \theta_i^2 + \rho(1 + \sum_{i=1}^{T-1} \mu_i^2)}$$

e $\eta(t)$ é um sinal de *dither*, introduzido para cumprir a condição de excitação persistente necessária para a correcta identificação dos parâmetros do sistema [116].

Linear Quadratic Gaussian

Os controladores LQG são controladores óptimos que minimizam uma função de custo quadrática. O problema foi inicialmente formulado durante a década de 70 [119] mas por requerer uma grande capacidade de processamento e por inicialmente não haver uma solução para o problema de regulação, não teve aplicação imediata. Durante a década de 80 foram propostas novas soluções para o problema de controlo [120], [121] e [122] e com a massificação dos computadores esta técnica tornou-se mais comum.

Considerando que o sistema é descrito pelo modelo ARMAX

$$A(q)y(k) = B(q)u(k) + C(q)e(k)$$

em que, $y(k)$ e $u(k)$ representam o sinal de saída e de controlo do sistema, respectivamente e $e(k)$ é ruído branco. Os polinómios $A(q)$, $C(q)$ têm os coeficientes de maior ordem unitários e têm grau igual a n .

A função de custo que estabelece o problema de controlo do tipo LQG é a seguinte

$$J(y, u) = E [(y(k) - r(k))^2 + \rho u^2(k)]$$

Considera-se que todos os zeros de $C(q)$ estão dentro do círculo unitário, que não há factores comuns aos três polinómios e que poderá existir um factor comum aos polinómios $A(q)$ e $B(q)$ com todos os zeros dentro do círculo unitário. Sendo $P(q)$ o polinómio com grau n com o coeficiente de maior ordem unitário e com todos os zeros dentro do círculo unitário que satisfaz a equação

$$rP(q)P(q^{-1}) = \rho A(q)A(q^{-1}) + B(q)B(q^{-1})$$

então a função de controlo que minimiza a função de custo é dada por [98]:

$$u(k) = -\frac{S(q)}{R(q)}y(k)$$

em que $R(q)$ e $S(q)$ são as soluções da equação de Diophantine seguinte

$$A(q)R(q) + B(q)S(q) = P(q)C(z)$$

Existem várias soluções possíveis para $R(q)$ e $S(q)$. A estrutura do controlador é determinada por estes polinómios. Por exemplo, se não existir nenhuma amostra de atraso no controlador o grau de $R(q)$ deve ser igual ao de $S(q)$ [98].

Controlo por Posicionamento de Pólos

O controlo por posicionamento de pólos é do tipo auto-ajustável. Pressupõe que o modelo do sistema obtido pelo estimador é correcto e procede à síntese do controlador tendo por base o comportamento desejado em malha fechada. A especificação do sistema em malha fechada é feita através do posicionamento dos pólos nas posições escolhidas.

Esta técnica tem a vantagem de possibilitar o controlo de sistemas com tempo morto desconhecido e/ou variável e a possibilidade de se obterem respostas em malha fechada distintas face a variações de carga e da entrada de comando [123]. Esta técnica de controlo é a utilizada na maior parte deste trabalho e por isso vai ser aqui analisada em maior detalhe.

O comportamento em malha fechada do sistema é determinado pelo modelo em malha aberta e pela função de controlo escolhida. Considere-se que o modelo do sistema em malha aberta é do tipo ARX, dado pela equação 3.69:

$$A(q)y(k) = B(q)u(k) + v(k) \quad (3.69)$$

em que $y(k)$ e $u(k)$ são a saída e a entrada do sistema, respectivamente e $v(k)$ uma perturbação. Os polinómios $A(q)$ e $B(q)$ são dados pelas equações seguintes, onde se assume que $n_b < n_a$ para que o sistema seja causal e que os polinómios não têm factores comuns.

$$\begin{aligned} A(q) &= q^{n_a} + a_1q^{n_a-1} + \dots + a_{n_a} \\ B(q) &= b_1q^{n_b-1} + \dots + b_{n_b} \end{aligned}$$

Para que o sistema em malha fechada seja causal devem verificar-se as condições

$$n_{am} \geq n_{bm}$$

e

$$n_{am} - n_{bm} \geq n_a - n_b$$

O excesso de pólos do sistema em malha fechada deve ser igual ou superior ao excesso de pólos do sistema em malha aberta, o que implica que o atraso no modelo do sistema em malha fechada seja igual ou superior ao atraso no modelo em malha aberta [98].

Utiliza-se uma função de controlo linear do tipo

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k) \tag{3.70}$$

em que $u_c(k)$ é a entrada de comando e $R(q)$, $T(q)$ e $S(q)$ são dados por

$$\begin{aligned} R(q) &= q^{n_r} + r_1q^{n_r-1} + \dots + r_{n_r} \\ T(q) &= t_0q^{n_t} + t_1q^{n_t-1} + \dots + t_{n_t} \\ S(q) &= s_0q^{n_s} + s_1q^{n_s-1} + \dots + s_{n_s} \end{aligned}$$

A partir da equação 3.70 pode obter-se a expressão do sinal $u(k)$.

$$u(k) = \frac{T(q)}{R(q)}u_c(k) - \frac{S(q)}{R(q)}y(k) \tag{3.71}$$

Esta função de controlo apresenta dois graus de liberdade: um associado à função de transferência $\frac{T(q)}{R(q)}$ e outro associado à função de transferência $\frac{S(q)}{R(q)}$. Para que a função de controlo seja causal é necessário que $n_r \geq n_t$ e $n_r \geq n_s$. De acordo com [98] pode escolher-se

$$n_r = n_s = n_t$$

O diagrama de blocos detalhado do sistema em malha fechada é apresentado na figura 3.17.

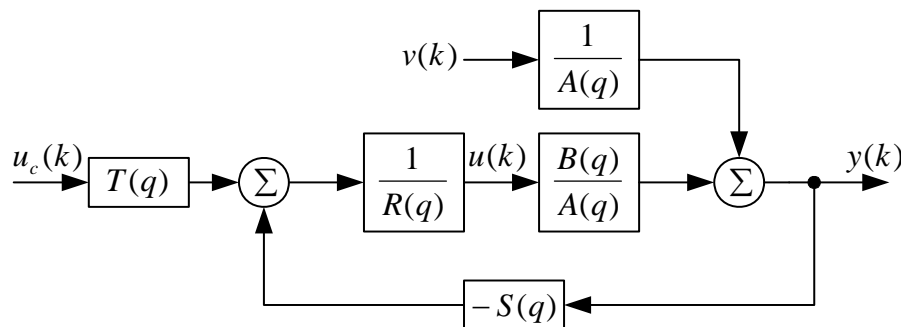


Figura 3.17: Diagrama de blocos detalhado da função de controlo de um controlador PP.

A determinação da equação do sistema em malha fechada obtêm-se através da eliminação de $u(k)$ entre as equações 3.69 e 3.71, obtendo-se

$$y(k) = \frac{B(q)T(q)}{A(q)R(q) + B(q)S(q)}u_c(k) + \frac{R(q)}{A(q)R(q) + B(q)S(q)}v(k) \quad (3.72)$$

Podemos reescrever a equação anterior da seguinte forma

$$y(k) = H_u(q)u_c(k) + H_v(q)v(k)$$

em que $H_u(q)$ representa a função de transferência em relação ao sinal de comando e $H_v(q)$ representa a função de transferência referente à perturbação com

$$\begin{aligned} H_u(q) &= \frac{B(q)T(q)}{A(q)R(q) + B(q)S(q)} \\ H_v(q) &= \frac{R(q)}{A(q)R(q) + B(q)S(q)} \end{aligned}$$

O polinómio característico de $H_u(q)$ é dado por

$$A_c(q) = A(q)R(q) + B(q)S(q) \quad (3.73)$$

O polinómio A_c é escolhido de modo a que o comportamento do sistema em malha fechada seja o desejado, daí o nome da técnica de posicionamento de pólos, visto que as raízes de A_c correspondem aos pólos do sistema em malha fechada. A partir desta equação podem determinar-se os polinómios S e R . A equação 3.73 designa-se por equação de Diophantine. De acordo com [98] é sempre possível chegar a uma solução se os polinómios A e B não tiverem factores comuns.

Para determinar o polinómio T é necessário fazer considerações sobre a resposta do sistema ao sinal de comando u_c . Essa resposta pode ser descrita pela equação 3.74, em que y_m representa o sinal de saída do modelo que se pretende.

$$A_m(q)y_m(k) = B_m(q)u_c(k) \quad (3.74)$$

Os polinómios $A_m(q)$ e $B_m(q)$ assumem a forma:

$$\begin{aligned} A_m(q) &= q^{n_{am}} + a_{m1}q^{n_{am}-1} + \dots + a_{n_{am}} \\ B_m(q) &= b_{m1}q^{n_{bm}-1} + \dots + b_{n_{bm}} \end{aligned}$$

A escolha dos coeficientes dos polinómios $A_m(q)$ e $B_m(q)$ deve ser feita de modo a que o sistema em malha fechada seja estável e que possua ganho estático unitário. Os coeficientes de B_m devem ser tais que o controlador não tente cancelar zeros fora do círculo unitário. Para que o sistema em malha fechada seja causal é necessário que $n_{bm} < n_{am}$ e que o excesso de pólos do sistema em malha fechada seja igual ou superior ao excesso de pólos do sistema em malha aberta que lhe dá origem, como já foi mencionado.

Por comparação entre 3.74 e 3.72 obtém-se

$$\frac{B(q)T(q)}{A(q)R(q) + B(q)S(q)} = \frac{B(q)T(q)}{A_c(q)} = \frac{B_m(q)}{A_m(q)} \quad (3.75)$$

Introduzindo um polinómio observador designado A_o obtém-se

$$\begin{aligned} T(q) &= \frac{B_m(q)A_o(q)}{B(q)} \\ A_c(q) &= A_m(q)A_o(q) \end{aligned}$$

A introdução deste polinómio não afecta a resposta do sistema em relação ao sinal de comando, porque ele é cancelado na função de transferência $H_u(q)$. Esse cancelamento não acontece porém na função de transferência $H_v(q)$ pelo que o polinómio observador vai influenciar a resposta do sistema às perturbações. Na ausência de informação sobre a dinâmica das perturbações a escolha dos coeficientes de A_o deve ser encarada como um parâmetro de projecto [98]. A sua dinâmica deve no entanto ser mais rápida do que a dinâmica da malha fechada imposta por A_m para uma melhor rejeição das variações de carga [98].

Para um sistema de 1ª ordem genérico, sem modelação do tempo morto fraccionário, cuja função de transferência discreta é dada por:

$$G(q) = \frac{b}{q + a}$$

ou seja

$$\begin{aligned} A(q) &= q + a \\ B(q) &= b \end{aligned}$$

Os polinómios que definem o comportamento em malha fechada e o polinómio observador são dados por

$$A_m(q) = q + a_m$$

$$A_o(q) = q + a_o$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= q - 1 \\ S(q) &= s_0q + s_1 \end{aligned}$$

Os valores dos coeficientes de $R(q)$ e $S(q)$ obtêm-se através da resolução da equação de Diophantine (ver anexo A).

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

O polinómio $T(q)$ obtém-se a partir da equação

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

e de considerações adicionais, como o facto de ser desejável que o ganho estático em malha fechada seja unitário (ver anexo A).

A função de controlo é dada por

$$u(k) = u(k-1) + t_0 u_c(k) + t_1 u_c(k-1) - s_0 y(k) - s_1 y(k-1) \quad (3.79)$$

onde

$$\left\{ \begin{array}{l} t_0 = \frac{b_m}{b} \\ b_m = 1 + a_m \\ t_1 = \frac{b_m a_o}{b} \\ s_0 = \frac{1 - a + a_m + a_o}{b} \\ s_1 = \frac{a_m a_o - a}{b} \end{array} \right.$$

O parâmetro b_m determina-se de modo a que o ganho estacionário em malha fechada seja unitário. Na função de controlo os parâmetros a e de b são substituídos pelos valores \hat{a} e \hat{b} , estimados em tempo-real.

Para um sistema de 2^a ordem genérico, sem modelação do tempo morto fraccionário, cuja função de transferência discreta é dada por

$$G(q) = \frac{b_1 q + b_2}{q^2 + a_1 q + a_2}$$

ou seja

$$\begin{aligned} A(q) &= q^2 + a_1 q + a_2 \\ B(q) &= b_1 q + b_2 \end{aligned}$$

Os polinómios que definem o comportamento em malha fechada e o polinómio observador são dados por

$$A_m(q) = q^2 + a_{m1} q + a_{m2}$$

$$A_o(q) = q^2 + a_{o1} q + a_{o2}$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómio $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= (q - 1)(q + r_1) = q^2 + (r_1 - 1)q - r_1 \\ S(q) &= s_0q^2 + s_1q + s_2 \end{aligned}$$

Os valores dos coeficientes de $R(q)$ e $S(q)$ obtêm-se através da resolução da equação de Diophantine (ver anexo A).

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

O polinómio $T(q)$ obtêm-se a partir da equação

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

e de considerações adicionais, como o facto de ser desejável que o ganho estático em malha fechada seja unitário (ver anexo A).

A função de controlo é dada por

$$\begin{aligned} u(k) &= b_m(u_c(k) + a_{o1}u_c(k - 1) + a_{o2}u_c(k - 2)) - s_0y(k) \\ &\quad - s_1y(k - 1) - s_2y(k - 2) - (r_1 - 1)u(k - 1) + r_1u(k - 2) \end{aligned} \quad (3.82)$$

onde

$$\left\{ \begin{aligned} r_1 &= P_0 - b_1s_0 \\ s_0 &= -\frac{P_1 + b_1s_1}{P_2} \\ s_1 &= -\frac{b_1(P_3P_2 - P_1a_2b_1) + b_2P_5}{b_2(P_4b_1 - P_2b_2) - a_2b_1^2} \\ s_2 &= \frac{P_5 + (P_4b_1 - P_2b_2)s_1}{P_2b_1} \\ b_m &= \frac{1 + a_{m1} + a_{m2}}{b_1 + b_2} \end{aligned} \right.$$

com

$$\left\{ \begin{array}{l} P_0 = a_{m1} + a_{o1} + 1 - a_1 \\ P_1 = a_2 + a_1 P_0 - a_{m1} - a_{m2} - a_{o2} - a_{o1}(1 + a_{m1}) - 1 \\ P_2 = b_1(1 - a_1) + b_2 \\ P_3 = a_2(P_0 - 1) - a_1 P_0 - a_{m1} a_{o2} - a_{m2} a_{o1} \\ P_4 = b_1(a_1 - a_2) \\ P_5 = -P_3 P_2 + P_4 P_1 \\ P_6 = -a_2 P_0 - a_{m2} a_{o2} \end{array} \right.$$

Na função de controlo os parâmetros a_1 , a_2 , b_1 e b_2 são substituídos pelos valores estimados em tempo-real.

3.6 Controladores adaptativos distribuídos

A distribuição do sistema de controlo tem consequências no desempenho do sistema. Como foi referido no capítulo 2 a distribuição do controlador induz o aparecimento de atraso variável entre a amostragem e a actuação. As técnicas de modelação e controlo “clássicas” não têm em conta este tipo de atraso.

As técnicas de modelação que ignoram os atrasos introduzidos na malha de controlo conduzem à identificação de valores para os parâmetros do sistema discreto que diferem dos valores reais. A localização dos pólos identificados deste modo varia dependendo do valor efectivo do atraso que afecta o sistema.

A consequência natural da modelação incorrecta dos parâmetros do sistema discreto é a degradação do desempenho de controlo. Essa degradação é, em parte, mascarada pelo facto da malha de controlo ser uma malha realimentada e por isso capaz de compensar alguns dos efeitos nefastos resultantes da modelação incorrecta.

Na análise do efeito do atraso sobre o desempenho de controlo surge como mais relevante a relação entre o valor do atraso e o período de amostragem, do que o valor absoluto do atraso. Essa relação passa a ser designada por R_{ha} e é dada por

$$R_{ha} = \frac{t_{sa}}{h} \quad (3.83)$$

em que t_{sa} é o tempo entre o instante de amostragem e o instante de actuação e h o período de amostragem.

Só a partir de um dado valor de R_{ha} é que o efeito negativo do atraso sobre o desempenho de controlo começa a ser sentido. Cervin [5] sugere que para valores de R_{ha} até 0.2 não é necessário tomar medidas adicionais, podendo ignorar-se o atraso introduzido na malha sem grande perda em termos do desempenho de controlo.

Esta realidade também explica porque é que nas implementações iniciais de sistemas distribuídos a perda de desempenho não era muito sentida, uma vez que, como já foi referido, as implementações eram dedicadas e por isso o atraso introduzido não era muito elevado.

No entanto com a evolução da complexidade das implementações e a pressão para que um maior número de aplicações coexistam no mesmo barramento o atraso introduzido na malha e o *jitter* aumentam e os seus efeitos nefastos sobre o desempenho de controlo fazem-se sentir, podendo levar, em situações extremas, à perda de estabilidade do sistema [124], [125], [5] e [4], o que é inaceitável.

Torna-se assim necessária uma estratégia que permita conviver com os atrasos introduzidos pelo processo de distribuição dos controladores sem contudo descuidar o desempenho de controlo.

Nesta secção são apresentadas as soluções propostas neste trabalho, que se podem dividir em dois tipos: soluções locais e soluções globais.

As soluções locais destinam-se a melhorar o desempenho de cada malha de controlo distribuído sem ter em conta o resto do sistema distribuído.

As soluções globais visam a optimização da operação de sistemas distribuídos constituídos por várias malhas de controlo e/ou outros tipos de aplicações que partilham a mesma rede de comunicação. Durante a operação deste tipo de sistemas torna-se, por vezes, necessário chegar a compromissos entre o nível de desempenho de cada malha de controlo individualmente e o nível de utilização da rede de comunicação.

Em ambos os casos a abordagem centra-se sobre o problema de controlo.

3.6.1 Modelação com tempo morto fraccionário

Ao nível local, para cada malha de controlo distribuída sujeita a atrasos variáveis entre a amostragem e a actuação, propõe-se que seja considerado sistematicamente esse atraso como um tempo morto fraccionário e que ele seja modelado de acordo com os modelos de identificação propostos em [126] e [127] para que os parâmetros identificados se aproximem mais dos valores reais e com isso se consiga um melhor desempenho em malha fechada.

Assim, devem utilizar-se os modelos seguintes para sistemas discretos de 1^a e de 2^a ordem sujeitos a atrasos variáveis entre a amostragem e a actuação (ver secção 3.3.4):

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1}}$$

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2} + b_3q^{-3}}{1 + a_1q^{-1} + a_2q^{-2}}$$

A identificação será efectuada recorrendo ao modelo ARX, como descrito na secção 3.4.5, sendo utilizados para um sistema de 1^a ordem os parâmetros

$$\theta = [-a_1, b_1, b_2]$$

e os regressores

$$\varphi(k) = [y(k), u(k-1), u(k-2)]$$

Para sistemas de 2ª ordem temos os parâmetros

$$\theta = [-a_1, -a_2, b_1, b_2, b_3]$$

e os regressores

$$\varphi(k) = [y(k), y(k-1), u(k-1), u(k-2), u(k-3)]$$

Este método permite, em última análise, identificar um sistema discreto afectado por um atraso que corresponde ao valor médio do tempo morto fraccionário que afecta a malha de controlo.

Os resultados obtidos através da aplicação desta técnica a sistemas de controlo distribuído de 1ª e de 2ª ordem foram apresentados nos artigos [124], [128] e [129].

3.6.2 Adaptação dinâmica do período de amostragem

Actualmente e por razões essencialmente económicas existe a tendência para que haja uma cada vez maior integração de sistemas. Nesse sentido o mesmo barramento tende a ser partilhado por diversas aplicações de tempo-real do tipo *hard* e *soft* e por aplicações que não são de tempo-real, como por exemplo aplicações para configuração. O projecto de sistemas deste tipo para o cenário de pior caso origina implementações em que os recursos existentes, nomeadamente a largura de banda do barramento, não são bem aproveitados. Por outro lado se forem tidos em conta cenários menos pessimistas podem ocorrer situações de sobrecarga temporária no barramento. Torna-se por isso interessante a possibilidade de alterar o período de amostragem das malhas de controlo no sentido de libertar largura de banda que pode ser utilizada durante a situação de sobrecarga. No entanto esta alteração deve permitir manter um desempenho de controlo “aceitável” de acordo com o sistema em causa.

Por outro lado em malhas de controlo afectadas por atrasos entre a amostragem e a actuação elevados a alteração do período de amostragem pode servir para melhorar o desempenho de controlo da malha.

De modo a suportar a alteração do período de amostragem das malhas de controlo em sistemas distribuídos sujeitos a atrasos entre a amostragem e a actuação propõem-se dois tipos de comutação entre controladores adaptativos, ambos baseados na alteração dinâmica em tempo-real do período de amostragem. O primeiro tipo de comutação tem por objectivo a optimização local do desempenho da malha de controlo sem ter em conta o resto das aplicações que possam partilhar o mesmo barramento. O segundo tipo de comutação tem por objectivo a optimização global do sistema distribuído como um todo, e não de cada malha de controlo em particular, com vista por exemplo à resolução de situações transitórias de sobrecarga no barramento. Estas técnicas serão descritas de seguida.

Aplicação à optimização local de malhas de controlo

Face à constatação de que o desempenho de controlo é afectado não tanto em função do valor absoluto do atraso presente na malha de controlo mas sim em função da relação entre o valor absoluto desse atraso e o período de amostragem, propõe-se que, no intuito de melhorar o desempenho de controlo de uma dada malha, se proceda à monitorização do valor de R_{ha} e sempre que se exceda um determinado valor limiar seja efectuada a alteração do período de amostragem, aumentando-o, de modo a que o efeito do atraso sobre o desempenho da malha seja reduzido. Esta solução parece, à primeira vista, ir contra o senso comum, uma vez que se considera normalmente que o desempenho melhora com a diminuição do período de amostragem. No entanto convém não esquecer que a introdução de atrasos na malha de controlo leva a que o sistema se comporte do modo diferente do que acontece quando não há atrasos na malha.

Esta proposta tem por base os resultados preliminares apresentados no final do capítulo 4 (secção 4.4.5) que sugerem o interesse deste tipo de optimização.

Resumindo, neste caso o que se propõe é a alteração do tempo de amostragem da malha de controlo em função do atraso entre a amostragem e a actuação que afecta a malha a cada momento. A selecção do período de amostragem é efectuada com base em critérios internos à própria malha de controlo. Neste caso o valor da relação entre o atraso entre a amostragem e a actuação e o período de amostragem. A implementação do controlador adaptativo pode ser efectuada de diferentes modos, como será discutido na secção seguinte.

Em relação à gama de valores de h a utilizar, a sua determinação deve seguir os critérios usuais descritos na secção 3.2.4. Em particular o critério utilizado neste trabalho é o proposto por Aström e Wittenmark [98], que é sabido ser um critério bastante conservador.

Aplicação para a optimização global de sistemas distribuídos

Esta proposta pretende impedir a ocorrência de situações temporárias de sobrecarga na rede de comunicação utilizada pelos vários sistemas distribuídos de controlo ou por outro tipo de aplicações. Isso é conseguido à custa do desempenho das malhas de controlo locais, mas permite uma optimização global do funcionamento do sistema.

Esta solução só é aceitável porque a diminuição do desempenho das malhas de controlo não é muito grande e porque a sobrecarga que afecta o tráfego no barramento é, como foi referido, transitória. Ela apresenta assim um compromisso entre a operação com sucesso do sistema global durante um período curto de tempo e o desempenho local de cada uma das malhas de controlo.

A proposta consiste no aumento do período de amostragem das malhas de controlo com o objectivo de libertar largura de banda de modo a evitar a ocorrência de sobrecarga no barramento devida à necessidade de enviar um determinado conjunto de mensagens durante um período de tempo limitado. A situação inicial, em termos do valor de h , deve ser reposta depois do período transitório ter passado de modo a que o desempenho das malhas de controlo seja o melhor possível. O critério de selecção para

o período de amostragem é, neste caso, externo à malha de controlo e imposto pelo mecanismo de gestão da qualidade de serviço do barramento.

As considerações efectuadas anteriormente em relação à gama de valores permitidos para o período de amostragem são também válidas para esta proposta.

Os resultados obtidos através da aplicação desta técnica de optimização deram origem à publicação dos artigos [130], [131], [132] e [133] e à aceitação para publicação em livro do seguinte [134].

3.6.3 Implementação do controlador adaptativo com período de amostragem variável

A implementação do controlador adaptativo capaz de suportar a adaptação dinâmica do período de amostragem pode ser efectuada de diversos modos, de acordo com os meios disponíveis para cada aplicação em particular. Propõem-se três tipos de implementações designadas por: estática, dinâmica e híbrida. Para todas elas utilizam-se reguladores auto-ajustáveis. A especificação de projecto que varia é o período de amostragem.

Abordagem estática

A abordagem estática propõe o projecto prévio de vários reguladores auto-ajustáveis com modelação do tempo morto fraccionário, um para cada valor permitido do período de amostragem. Ou seja propõe o projecto prévio de vários reguladores auto-ajustáveis para diferentes valores de h para os quais são determinados os respectivos parâmetros. Esses parâmetros são depois utilizados para a criação de um banco de reguladores com parâmetros fixos. Durante a operação do sistema de controlo é feita a comutação entre os vários reguladores face ao critério de selecção utilizado para escolher qual o período de amostragem a cada momento.

Os parâmetros dos reguladores são calculados antecipadamente com base nos parâmetros identificados para o sistema através da realização de ensaios específicos para identificação. No nó do controlador são guardados os parâmetros de cada controlador e durante a operação do sistema é efectuada a adaptação dinâmica do período através da comutação entre cada um dos reguladores.

A figura 3.18 apresenta esta proposta de implementação. O sinal de referência não é apresentado nesta figura para melhorar a legibilidade.

A implementação estática destina-se a aplicações com pouca capacidade de processamento e alguma disponibilidade de memória e proporciona uma comutação rápida entre controladores. Por outro lado este esquema só permite que seja utilizado um conjunto discreto de valores para h e os reguladores não são capazes de seguir alterações que ocorram no modelo discreto do sistema durante a sua operação porque os parâmetros são fixos. Para que esta implementação proporcione os melhores resultados é fundamental que se conheçam as condições operacionais do sistema, como por exemplo o padrão de atrasos que o afectam, para que a identificação seja feita em condições tão perto das reais quanto possível. Como se sabe nem sempre é possível *a priori* ter

um conhecimento muito preciso do padrão do atraso que afecta o sistema por isso esta solução nem sempre é viável.

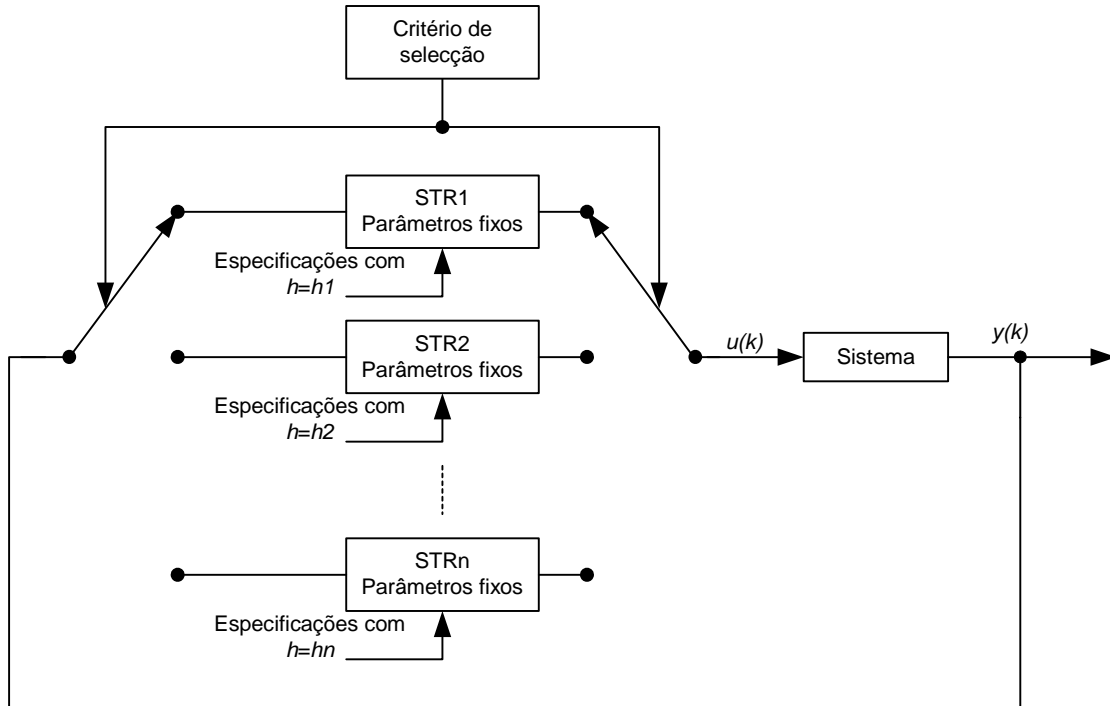


Figura 3.18: Implementação estática.

Abordagem dinâmica

A abordagem dinâmica propõe a utilização de um único regulador auto-ajustável com modelação do tempo morto fraccionário cujas especificações são comutadas. Como já foi referido a única especificação que varia é o período de amostragem. Tanto a estimativa dos parâmetros do sistema como o cálculo dos parâmetros do regulador são efectuados em tempo-real para cada ciclo de controlo utilizando um único estimador. A figura 3.19 apresenta a abordagem dinâmica.

Esta abordagem exige maior capacidade de processamento do nó do controlador e é mais flexível porque permite a utilização de qualquer valor para o período de amostragem (dentro da gama de valores possível) e a adaptação a variações no modelo discreto do sistema. Como desvantagem temos o tempo de comutação, que é maior uma vez que o regulador demora algum tempo a estimar o valor correcto dos parâmetros de cada vez que há comutação entre especificações.

Abordagem híbrida

Por último a abordagem híbrida pretende acelerar o tempo de comutação associado à abordagem dinâmica. Para isso propõe-se que, tal como no caso anterior, se utilize

um único regulador auto-ajustável com modelação do tempo morto fraccionário e com especificações comutadas mas com vários estimadores, um para cada valor de h . Neste caso temos um banco de estimadores que são comutados em paralelo com as especificações, fazendo com que a convergência dos parâmetros do regulador seja mais rápida. A figura 3.20 apresenta a abordagem híbrida.

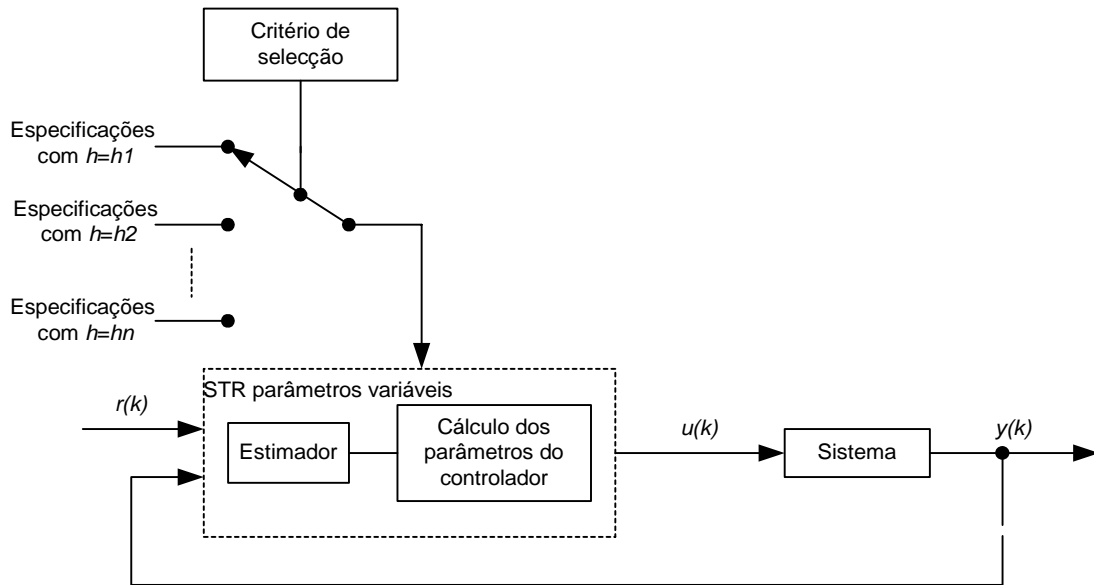


Figura 3.19: Implementação dinâmica.

Esta abordagem exige a mesma capacidade de processamento da anterior e maior capacidade de memória, para guardar a informação associada a cada estimador. Neste caso, tal como no caso da abordagem estática, também só se pode utilizar um conjunto discreto de valores para h (determinados *a priori*) e seleccionados dentro do intervalo de valores permitidos. O tempo de comutação é menor do que o da abordagem dinâmica.

Para cada estimador são guardados os valores do vector de parâmetros Φ e da matriz de covariância \mathbf{P} quando é feita a comutação para um novo valor de h . Quando o critério de selecção escolher de novo o mesmo valor de h o estimador retomará a sua função com base nos valores guardados.

No momento da comutação e durante algumas iterações existe uma discrepância entre o valor dos parâmetros e da matriz de covariância para o novo valor de h e o valor do vector de regressores φ que corresponde ao valor de h anterior. Esta situação é apresentada na figura 3.21. Na iteração $n+3$ o período de amostragem é comutado de h_1 para h_2 . O vector de parâmetros θ e a matriz \mathbf{P} correspondentes a h_1 são guardados, para serem reutilizados da próxima vez que $h = h_1$ e são repostos o vector de parâmetros θ e a matriz de covariância \mathbf{P} correspondentes a $h = h_2$ para que o estimador continue os cálculos a partir deles. Para esta iteração o vector de regressores φ contém valores de $y(k)$ e de $u(k)$ amostrados para $h = h_1$. Até que o vector de regressores seja actualizado na totalidade decorrem algumas iterações em que esta incongruência

persiste. O número de iterações necessárias para a actualização do vector de regressores depende da ordem do sistema a controlar, para um sistema de 1ª ordem será de duas iterações e para um sistema de 2ª ordem de três iterações.

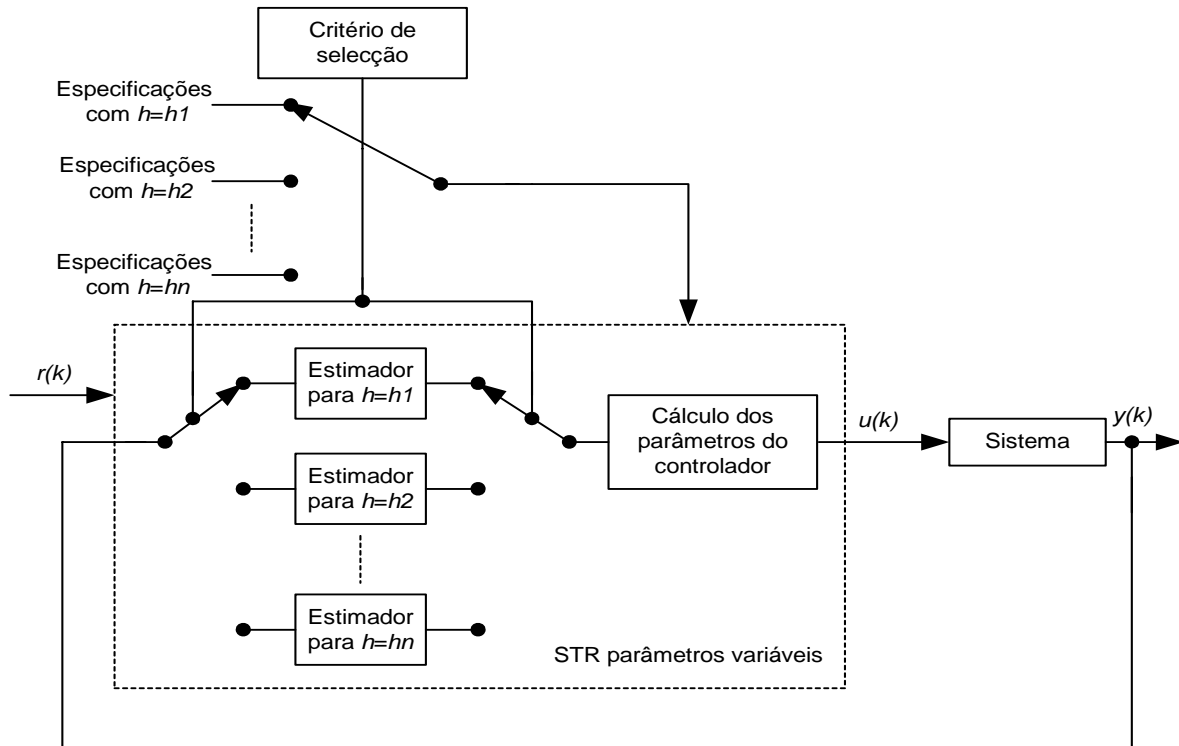


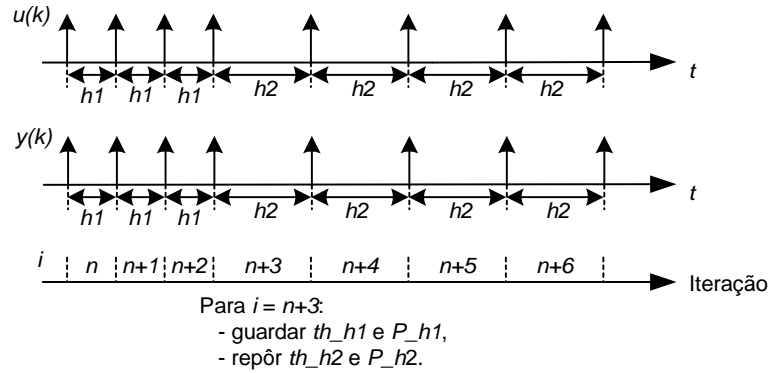
Figura 3.20: Implementação híbrida.

3.7 Controlo adaptativo distribuído por posicionamento de pólos

O projecto dos controladores distribuídos por posicionamento de pólos segue o procedimento “tradicional” descrito na secção 3.5.2. A única diferença é que se baseia em modelos do sistema que modelam o tempo morto fraccionário e por isso originam implementações mais complexas do que as que se obtém tradicionalmente. Estas implementações requerem maior capacidade de processamento e algoritmos que demoram mais tempo a serem executados.

Para um sistema de 1ª ordem genérico, com modelação do tempo morto fraccionário, a função de transferência discreta é dada por

$$G(q) = \frac{b_1q + b_2}{q(q + a_1)}$$

Figura 3.21: Comutação entre valores de h , abordagem híbrida.

Ao denominador da função foi acrescentado um pólo na origem para garantir a causalidade da função [135].

Os polinómios $A(q)$ e $B(q)$ são dados por:

$$\begin{aligned} A(q) &= q(q + a_1) \\ B(q) &= b_1q + b_2 \end{aligned}$$

O polinómio que define o comportamento em malha fechada e o polinómio observador são dados por

$$A_m(q) = q(q + a_m)$$

$$A_o(q) = q(q + a_o)$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo:

$$\begin{aligned} R(q) &= (q - 1)(q + r) \\ S(q) &= q(s_0q + s_1) \end{aligned}$$

Os valores dos coeficientes de $R(q)$ e $S(q)$ obtêm-se através da resolução da seguinte equação de Diophantine (ver anexo A):

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

O polinómio $T(q)$ obtêm-se a partir da equação

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

e de considerações adicionais, como o facto de ser desejável que o ganho estático em malha fechada seja unitário (ver anexo A). Deste modo o sinal de controlo é dado pela expressão:

$$u(k) = b_m u_c(k) + b_m a_0 u_c(k-1) - s_0 y(k) + s_1 y(k-1) - (r-1)u(k-1) + ru(k-2) \quad (3.86)$$

onde

$$\begin{cases} b_m = 1 + a_m \\ r = \frac{b_2 s_1}{a_1} \\ s_0 = \frac{P_1}{P_2} \\ s_1 = \frac{a_1(a_m + a_o) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2} \end{cases}$$

com

$$\begin{cases} P_0 = a_2 b_2 - b_2 + a_1 b_1 \\ P_1 = b_2(a_m a_o + a_1) - P_0(1 - a_1 + a_m + a_o) \\ P_2 = b_2^2 - b_1 P_0 \end{cases}$$

O parâmetro b_m determina-se de modo a que o ganho estacionário em malha fechada seja unitário. Os parâmetros a_1 , b_1 e b_2 são substituídos pelos valores estimados em tempo-real pelo controlador adaptativo.

Para um sistema de 2ª ordem genérico, com modelação do tempo morto fraccionário, cuja função de transferência discreta é dada por

$$G(q) = \frac{b_1 q^2 + b_2 q + b_3}{q(q^2 + a_1 q + a_2)}$$

em que, tal como no caso anterior é adicionado um pólo na origem de modo a manter a causalidade da função [135].

Os polinómios $A(q)$ e $B(q)$ são dados por:

$$\begin{aligned} A(q) &= q(q^2 + a_1 q + a_2) \\ B(q) &= b_1 q^2 + b_2 q + b_3 \end{aligned}$$

O polinómio que define o comportamento em malha fechada e o polinómio observador são dados por:

$$A_m(q) = q(q^2 + a_{m1} q + a_{m2})$$

$$A_o(q) = q(q^2 + a_{o1}q + a_{o2})$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómio $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= (q-1)(r_0q^2 + r_1q + r_2) \\ S(q) &= q(s_0q^2 + s_1q + s_2) \end{aligned}$$

Os valores dos coeficientes de $R(q)$ e $S(q)$ obtêm-se através da resolução da seguinte equação de Diophantine (ver anexo A):

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

O polinómio $T(q)$ obtêm-se a partir da equação

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

e de considerações adicionais, como o facto de ser desejável que o ganho estático em malha fechada seja unitário (ver anexo A).

A função de controlo é dada pela expressão:

$$\begin{aligned} u(k) &= (1-r_1)u(k-1) + (r_2-r_1)u(k-2) + r_3u(k-3) \\ &\quad -s_0y(k) - s_1y(k-1) - s_2y(k-2) \\ &\quad + b_m(u_c(k) + a_{o1}u_c(k-1) + a_{o2}u_c(k-2)) \end{aligned} \quad (3.89)$$

onde

$$\left\{ \begin{array}{l} r_0 = 1 \\ r_1 = X_1 - b_1s_0 \\ r_2 = \frac{b_3}{a_2}s_2 \\ s_0 = \frac{X_2 - X_1X_6 - \frac{b_3}{a_2}s_2 - b_1s_1}{X_7} \\ s_1 = \frac{X_{11}X_9 - X_{13}X_{10}}{X_{10}X_{12} - X_{11}X_8} \\ s_2 = -\frac{X_8s_1 + X_9}{X_{10}} \\ b_m = \frac{1+a_{m1}+a_{m2}}{b_1+b_2+b_3} \end{array} \right.$$

com

$$\left\{ \begin{array}{l} X_1 = a_{o1} + a_{m1} + 1 - a_1 \\ X_2 = a_{o2} + a_{o1}a_{m1} + a_{m2} + a_1 - a_2 \\ X_3 = a_{m1}a_{o2} + a_{o1}a_{m2} + a_2 \\ X_4 = a_{m2}a_{o2} \\ X_5 = a_2 - a_1 \\ X_6 = a_1 - 1 \\ X_7 = b_2 - X_6b_1 \\ X_8 = \frac{X_5b_1^2 - b_1b_3}{X_7} + b_2 \\ X_9 = \frac{b_3X_2 - b_3X_1X_6 - X_5b_1X_2 + X_5b_1X_1X_6}{X_7} - X_3 + X_5X_1 \\ X_{10} = b_1 + \frac{b_3}{a_2} \left(X_6 + \frac{X_5b_1 - b_3}{X_7} \right) \\ X_{11} = \frac{b_3}{a_2} X_5 + b_2 - \frac{b_1}{X_7} b_3 \\ X_{12} = b_3 - \frac{a_2b_1^2}{X_7} \\ X_{13} = \frac{a_1b_1X_2 - a_2b_1X_1X_6}{X_7} - X_4 - a_2X_1 \end{array} \right.$$

Os parâmetros a_1 , a_2 , b_1 , b_2 e b_3 são substituídos pelos valores estimados em tempo-real durante a operação do sistema adaptativo.

Como foi referido no início desta secção a implementação dos controladores tendo em conta o tempo morto fraccionário dá origem a algoritmos mais extensos que, por isso, demoram mais tempo a executar, como se conclui através da comparação das funções de controlo dos sistemas respectivos. As funções de controlo para um sistema de 1ª ordem sem e com o tempo morto fraccionário são dadas, respectivamente, pelas expressões 3.79 e 3.86, e para um sistema de 2ª ordem pelas expressões 3.82 e 3.89. O algoritmo de controlo requer ainda a identificação prévia dos parâmetros da função discreta do sistema, como foi referido na secção 3.4.5. O tempo de execução do procedimento de identificação também é maior no caso dos sistemas com tempo morto fraccionário visto que há mais parâmetros a identificar. Para o sistema de 1ª ordem passa a ser necessário identificar 3 parâmetros em vez de 2 e para o sistema de 2ª ordem 5 parâmetros em vez de 4.

Medindo o tempo de execução dos algoritmos de controlo com e sem tempo morto fraccionário nas condições de teste referidas no capítulo 4, verificou-se um acréscimo de

50%, para o sistema de 1ª ordem, e de 20%, para o sistema de 2ª ordem, nos tempos de execução dos algoritmos de controlo que têm em conta o tempo morto fraccionário em relação aos que não o têm em conta.

3.8 Conclusão

Neste capítulo apresentaram-se os principais problemas associados à distribuição de controladores digitais, descreveu-se o processo de amostragem de sistemas com e sem tempo morto e apresentaram-se os modelos discretos mais utilizados na modelação de sistemas. Descreveu-se ainda o processo de identificação de sistemas através do método dos mínimos quadrados e algumas das técnicas utilizadas na identificação de sistemas variáveis no tempo. De seguida apresentaram-se os esquemas mais comuns utilizados em controlo adaptativo “clássico” assim com exemplos de controladores adaptativos e propôs-se o esquema de comutação de controladores auto-ajustáveis. Os controladores adaptativos por posicionamento de pólos foram descritos com maior detalhe por serem utilizados nas implementações efectuadas neste trabalho. Por fim apresentaram-se algumas contribuições para o projecto de controladores distribuídos, nomeadamente a modelação sistemática do atraso introduzido na malha de controlo como tempo morto fraccionário e a adaptação dinâmica da taxa de amostragem de modo a permitir a optimização do desempenho global do sistema distribuído ou a optimização local de cada malha de controlo. Apresentaram-se ainda em detalhe os controladores adaptativos distribuídos por posicionamento de pólos.

Capítulo 4

Técnicas de controlo lineares para sistemas distribuídos

Neste capítulo são aplicadas, a um conjunto de sistemas de 1ª e de 2ª ordem, as técnicas de controlo propostas no capítulo 3. O desempenho em malha fechada é comparado com o que se obtém utilizando as técnicas de controlo “clássicas”.

Na primeira parte do capítulo são apresentadas as condições de realização dos testes, nomeadamente a arquitectura de base do sistema de teste, as características dos sistemas a controlar e dos controladores utilizados, bem como os critérios de avaliação do desempenho. Nas secções 4.2, 4.3 e 4.4 são apresentados os resultados.

4.1 Condições de teste

Nesta secção apresentam-se as condições de teste, descrevem-se a arquitectura base do sistema de teste, os sistemas a controlar, as características dos controladores utilizados e os critérios de avaliação do desempenho.

4.1.1 Arquitectura base do sistema de teste

O sistema de teste é constituído por uma malha de controlo totalmente distribuída, em que o sensor, o controlador e o actuador se encontram em nós fisicamente distintos ligados entre si pelo barramento CAN. A velocidade de transmissão no barramento CAN é igual a 125 kbps.

Considera-se que em cada nó os processadores executam apenas as tarefas relacionadas com a malha de controlo.

Utilizaram-se duas arquitecturas para o sistema de teste. Na primeira versão, mais simples de implementar, o acesso por parte dos nós ao barramento, para o envio de mensagens, é *event-triggered*. O diagrama de blocos deste sistema, designado A1 é apresentado na figura 4.1. Este sistema apresenta, para além dos nós directamente relacionados com a implementação da malha de controlo, dois nós de carga cuja finalidade é enviarem mensagens para aumentar o tráfego no barramento CAN.

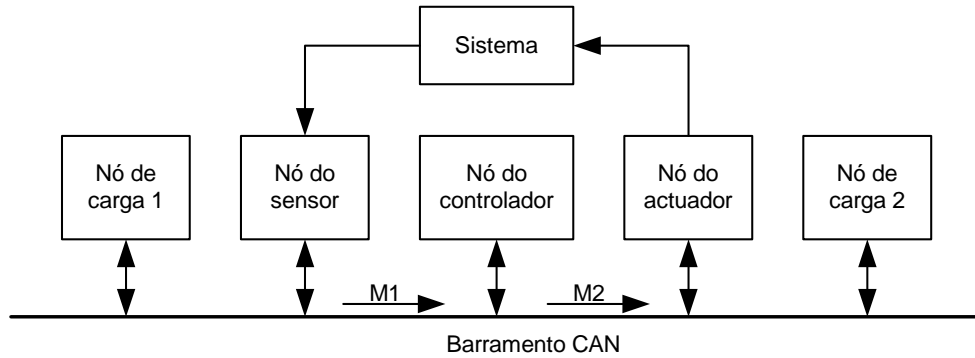


Figura 4.1: Diagrama de blocos do sistema de teste sem implementação do protocolo FTT-CAN (A1).

Na segunda versão o acesso dos nós ao barramento é feito através da utilização do protocolo FTT-CAN [44]. O diagrama de blocos desta versão, designada A2, é apresentado na figura 4.2. Neste caso o sistema apresenta, para além dos nós da malha de controlo, um nó que implementa as funções de *master*, definidas pelo protocolo FTT-CAN (ver secção 2.3.4).

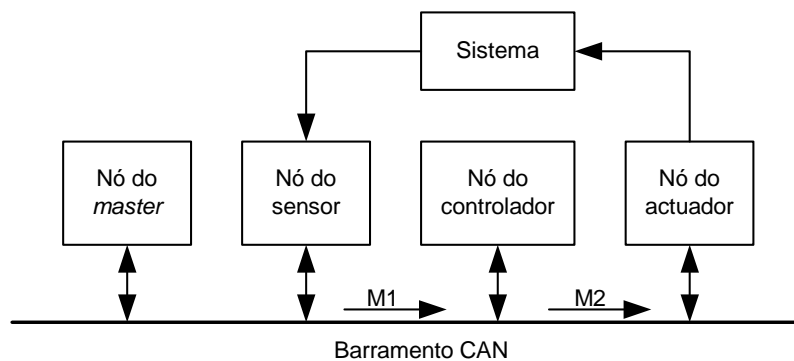


Figura 4.2: Diagrama de blocos do sistema de teste com implementação do protocolo FTT-CAN (A2).

Neste caso o tráfego no barramento é definido a nível do *master* através de uma tabela que contém a informação relativa a todas as mensagens que utilizam o sistema. Essas mensagens fazem todas parte do tráfego síncrono. O tráfego assíncrono não é utilizado nesta implementação. O *master* é responsável pelo escalonamento das mensagens de acordo com o protocolo FTT-CAN. O acesso dos nós ao barramento é do tipo *time-triggered*. Nesta arquitectura a carga no barramento é definida através do conjunto de mensagens a escalonar pelo *master* e por isso não são necessários nós de carga. A informação relativa ao conjunto de mensagens utilizado é apresentada neste capítulo na secção 4.4.1.

A malha de controlo é fechada através do barramento com a transmissão de duas mensagens: uma mensagem (M1) do sensor para o controlador, contendo o valor da amostra do sistema e uma mensagem do controlador para o actuador (M2), contendo o sinal de controlo a aplicar na entrada do sistema. As características relevantes das mensagens da malha de controlo, do ponto de vista do protocolo CAN (ver secção 2.3.4), são apresentadas na tabela 4.1, nomeadamente o identificador, o número de *bytes* de dados (DLC) e o período.

Mensagem	Identificador	DLC	Período
M1	130	1	$\simeq h$
M2	135	1	$\simeq h$

Tabela 4.1: Características das mensagens da malha de controlo.

Na tabela 4.1 h representa o período de amostragem da malha de controlo.

Na implementação que não utiliza o protocolo FTT-CAN as mensagens são *event-triggered*. A mensagem M1 é enviada na sequência da activação da tarefa que faz a amostragem do sistema no nó do sensor. A mensagem M2 é enviada pelo nó do controlador depois da recepção da mensagem M1 e do cálculo do valor do sinal de controlo. M1 e M2 são mensagens periódicas cujo envio está sujeito a *jitter*, por isso o seu período apresenta variações de iteração para iteração. Esse *jitter* é devido ao atraso variável introduzido pelos factores já discutidos em capítulos anteriores.

Na implementação que utiliza o protocolo FTT-CAN as mensagens são *time-triggered*, estando o seu envio para o barramento dependente de uma ordem emitida pelo nó do *master*. De acordo com o protocolo FTT-CAN o *master* é responsável pelo envio da TM no início de cada EC, a qual contém indicações relativas às mensagens que devem ser produzidas pelos respectivos nós durante esse EC, resultantes do processo de escalonamento.

A ordem para a amostragem do sistema é dada pelo *master* com período h definido pelo utilizador, de acordo com o mecanismo para *dispatching* das tarefas nos nós descrito em [67], não estando sujeita a incerteza. Se a amostragem do sistema for efectuada no EC número n , no melhor dos casos, a mensagem M1 é produzida no EC número $n + 1$ e a mensagem M2 é produzida no EC número $n + 2$, ou seja existe sempre pelo menos um atraso igual a dois ECs entre a amostragem e a actuação. A figura 4.3 apresenta o diagrama temporal da ocorrência destes eventos. O envio das mensagens M1 e M2 está sujeito a incerteza. Essa incerteza relaciona-se com o mecanismo de escalonamento utilizado pelo *master* e depende das características das mensagens a escalar, pelo que o atraso entre a amostragem e a actuação pode ser diferente do valor mínimo referido.

4.1.2 Padrões de carga do barramento

O atraso mínimo entre a amostragem e a actuação, devido apenas ao acesso à rede e ao tempo de processamento nos nós, é obtido quando não há mais tráfego no barramento e é igual a 8 ms utilizando a versão do simulador descrita na secção 4.1.6. Este valor do

atraso tem em conta os tempos de execução atribuídos a cada tarefa que é executada desde o momento da amostragem até ao momento de actuação assim como o tempo mínimo de acesso ao barramento. Os tempos de execução foram escolhidos de modo a reflectir implementações efectuadas com processadores de baixo custo e com “pouca” capacidade de processamento. O tempo de acesso ao barramento corresponde, neste caso, ao tempo mínimo uma vez que, para além das mensagens M1 e M2 que estão desfasadas uma da outra porque o envio de M2 pelo nó controlador depende da recepção prévia de M1, não existe tráfego no barramento, como foi referido anteriormente.

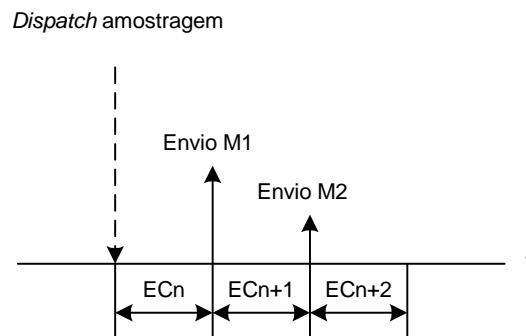


Figura 4.3: Sequência temporal da troca de mensagens da malha de controlo.

Os ensaios realizados sem carga servem de referência em termos do desempenho em malha fechada obtido.

Para simular a carga no barramento utilizaram-se diferentes tipos de padrões de carga de acordo com a arquitectura do sistema de teste.

Nos testes efectuados com a arquitectura A1 utilizam-se três padrões de carga distintos, todos eles tendo por base a *benchmark* do grupo *PSA Peugeot Citroën* (PSA) [136]. Esta *benchmark* é constituída por 12 mensagens cujas características são típicas das aplicações com redes CAN em automóveis.

Cada um dos padrões de carga utilizados produz atrasos médios entre a amostragem e a actuação distintos, fazendo com que a relação R_{ha} entre o valor médio do atraso e o período de amostragem varie dentro do intervalo $]0,1[$.

O padrão P1 é constituído pelas mensagens cujas características se apresentam na tabela 4.2.

O padrão de carga P1 resulta num valor médio do atraso entre a amostragem e a actuação relativamente pequeno por comparação com o valor de h . O valor da relação entre o atraso médio e o período de amostragem para o padrão de carga P1 é de $R_{ha} = 0.21$.

As mensagens de carga são todas mais prioritárias que as mensagens da malha de controlo. O histograma do atraso entre a amostragem e a actuação resultante da utilização deste padrão de carga é apresentado na figura 4.4.

Mensagem	Identificador	DLC	Período (ms)
1	70	8	25
2	75	3	35
3	80	3	50
4	85	2	37.5
5	90	5	50
6	95	5	100
7	100	4	37.5
8	105	5	125
9	110	4	50
10	115	7	250
11	120	5	125
12	125	1	250

Tabela 4.2: Características do conjunto de mensagens utilizadas para gerar os padrões de carga P1 e P2.

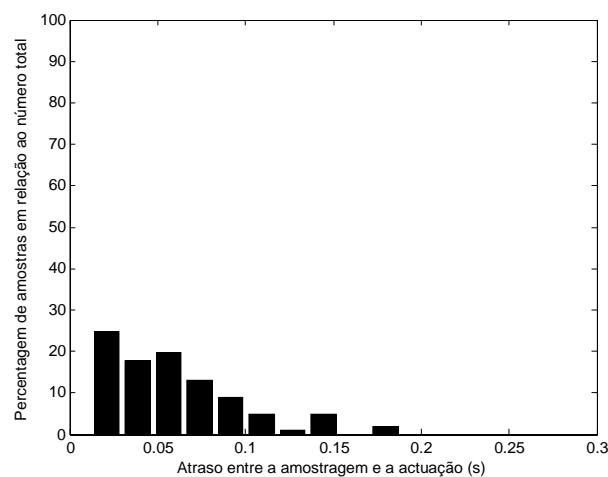


Figura 4.4: Atraso da amostragem para a actuação para o padrão de carga P1.

O padrão P2 foi obtido a partir do anterior através da soma de um valor constante. O histograma do atraso entre a amostragem e a actuação correspondente a este padrão é apresentado na figura 4.5.

O histograma correspondente ao padrão de carga P2 é uma versão deslocada do histograma correspondente ao padrão de carga P1, sendo o valor médio do atraso entre a amostragem e a actuação superior ao do padrão de carga P1. O valor da relação entre o atraso médio e o período de amostragem para o padrão de carga P2 é de $R_{ha} = 0.54$.

Por último o padrão P3 é obtido a partir de um conjunto de mensagens baseado na

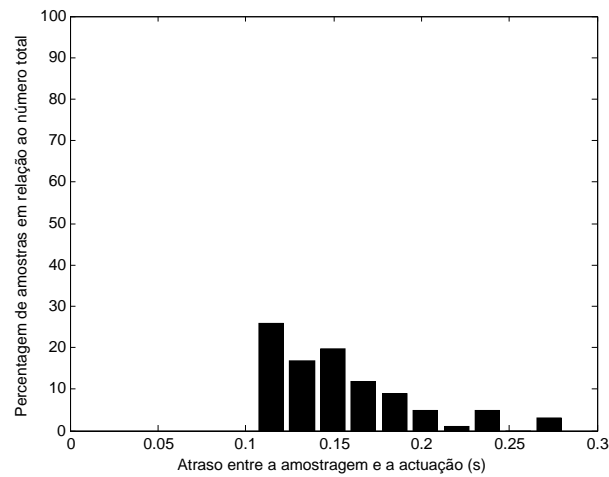


Figura 4.5: Atraso da amostragem para a actuação para o padrão de carga P2.

benchmark da PSA, ao qual também foi aplicada a soma de um valor constante. As características das mensagens utilizadas na obtenção deste padrão são apresentadas na tabela 4.3.

Mensagem	Identificador	DLC	Período (ms)
1	70	8	28
2	75	3	39.2
3	80	3	56
4	85	2	42
5	90	5	56
6	95	5	112
7	100	4	42
8	105	5	140
9	110	4	56
10	115	7	280
11	120	5	140
12	125	1	280

Tabela 4.3: Características do conjunto de mensagens utilizadas para gerar o padrão de carga P3.

O histograma do atraso da amostragem para a actuação resultante da aplicação do padrão de carga P3 é apresentado na figura 4.6.

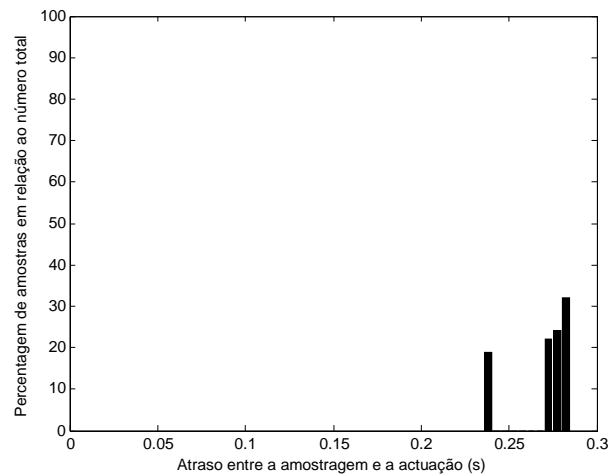


Figura 4.6: Atraso da amostragem para a actuação para o padrão de carga P3.

Para o padrão de carga P3 o valor médio do atraso é ainda maior, aproximando-se do valor de h . Neste caso o valor da relação entre o atraso médio e o período de amostragem é de $R_{ha} = 0.94$.

4.1.3 Sistemas a controlar

Nos ensaios realizados utilizaram-se quatro sistemas com funções de transferência consideradas representativas dos tipos mais comuns de sistemas.

As funções de transferência dos sistemas utilizados nos ensaios são apresentadas na tabela 4.4.

O sistema S1 é um sistema de 1ª ordem, o sistema S2 é um sistema de 2ª ordem com um pólo na origem, o sistema S3 é um sistema de 2ª ordem com dois pólos reais e o sistema S4 é um sistema de 2ª ordem com um par de pólos complexos conjugados.

4.1.4 Controladores

Os controladores utilizados são controladores adaptativos projectados utilizando a técnica de posicionamento de pólos, já referida anteriormente. Esta técnica permite especificar o comportamento do sistema em malha fechada através da imposição do posicionamento dos seus pólos. É usual utilizar-se também um polinómio observador com uma dinâmica mais rápida que a do sistema, que condiciona a resposta do sistema às perturbações.

O posicionamento escolhido para o pólo em malha fechada (em s) e as características do polinómio observador para o sistema S1 são apresentados na tabela 4.5.

O período de amostragem foi escolhido de acordo com o critério de Åström e Wittenmark [98] tendo em conta o comportamento do sistema em malha fechada. Na tabela

Sistema	Função de transferência
S1	$G(s) = \frac{0.5}{s+0.5}$
S2	$G(s) = \frac{0.5}{s(s+0.5)}$
S3	$G(s) = \frac{0.25}{(s+1)(s+0.25)}$
S4	$G(s) = \frac{1}{s^2+s+1}$

Tabela 4.4: Funções de transferência dos sistemas utilizados nos ensaios.

Sistema	Pólo em malha fechada	Observador	Período
S1	$a_{mf} = -2$	$a_{obs} = -4$	0.11 s - 0.28 s

Tabela 4.5: Características do sistema em malha fechada, sistema S1.

4.5 é também apresentada a gama de valores possível para o período de amostragem da malha de controlo de acordo com esse critério.

No caso dos sistemas de 2ª ordem apresenta-se, na tabela 4.6, o valor da frequência natural ω_n e do amortecimento ζ correspondentes à equação característica do sistema em malha fechada e do observador. A mesma tabela apresenta também a gama de valores possível para o período de amostragem da malha de controlo de acordo com o critério de Aström e Wittenmark [98].

Sistema	Polinómio característico	Polinómio observador	Período
S2	$\omega_n = 3 \text{ rad/s} ; \zeta = 1$	$\omega_n = 6 \text{ rad/s} ; \zeta = 1$	0.11 s - 0.28 s
S3	$\omega_n = 3 \text{ rad/s} ; \zeta = 1$	$\omega_n = 6 \text{ rad/s} ; \zeta = 1$	0.11 s - 0.28 s
S4	$\omega_n = 3 \text{ rad/s} ; \zeta = 1$	$\omega_n = 6 \text{ rad/s} ; \zeta = 1$	0.11 s - 0.28 s

Tabela 4.6: Características do sistema em malha fechada, sistemas S2, S3 e S4.

Foram projectados controladores por posicionamento de pólos considerando o tempo morto fraccionário existente na implementação distribuída e controladores por posicionamento de pólos sem ter em conta o tempo morto fraccionário (ver respectivamente 3.7 e 3.5.2).

Nos casos em que a identificação do sistema a controlar é efectuada em tempo-real é utilizado o método dos mínimos quadrados com esquecimento direccionado. Os pormenores de implementação do processo de identificação foram descritos na secção 3.4.5. O factor de esquecimento é igual a 0.99.

4.1.5 Critério de avaliação do desempenho

A avaliação do desempenho dos controladores é efectuada utilizando os índices quadráticos de desempenho *Integral of the Square Error* (ISE) e *Integral of the Time-weighted Square Error* (ITSE), dois dos critérios baseados no erro de controlo mais utilizados para a avaliação da resposta de um sistema em malha fechada.

O índice ISE é calculado a partir da equação 4.1

$$ISE = \int_0^{\infty} e^2(t)dt = \int_0^{\infty} (r(t) - y(t))^2 dt \quad (4.1)$$

em que $r(t)$ é o sinal de referência e $y(t)$ o sinal de saída do sistema.

O índice ITSE é calculado a partir da equação 4.2

$$ITSE = \int_0^{\infty} te^2(t)dt = \int_0^{\infty} t(r(t) - y(t))^2 dt \quad (4.2)$$

O cálculo dos índices ISE e ITSE foi implementado recorrendo à integração discreta através do método trapezoidal.

Estes critérios penalizam erros de controlo grandes e são adequados para o tratamento analítico dos resultados. O índice ISE pesa todos os erros de igual modo enquanto que o índice ITSE penaliza mais os erros em estado estacionário.

A utilização destes critérios não pode no entanto ser efectuada “cegamente” sem que se analise também o comportamento do sinal de saída para cada caso.

4.1.6 O simulador TrueTime

Os ensaios foram efectuados em MATLAB/Simulink, utilizando o simulador TrueTime [78] que possibilita a simulação de sistemas de tempo-real distribuídos. Esta ferramenta permite simular o comportamento temporal dos *kernels* de cada um dos nós e da rede de comunicação.

Neste simulador a rede de comunicação e os processadores são representados por blocos Simulink. A rede é implementada através do bloco TrueTime Network e cada processador é implementado por um bloco TrueTime Kernel.

A figura 4.7 apresenta a biblioteca TrueTime.

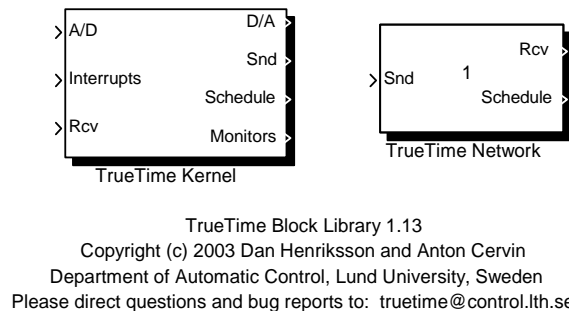


Figura 4.7: Biblioteca TrueTime para MATLAB/Simulink.

Ambos os blocos são *event-driven* e a sua execução é determinada por eventos internos, resultantes, por exemplo, da conclusão de uma tarefa ou do *overflow* de um temporizador, e externos, resultantes da ocorrência de uma interrupção externa ou da chegada de uma mensagem à rede. Os blocos incluem saídas designadas *Schedule* e *Monitors* que apresentam representações gráficas da atribuição dos recursos partilhados, como a UCP ou a rede, ao longo da simulação.

TrueTime Kernel

O bloco Kernel simula o *kernel* de um processador com conversores A/D e D/A, interface com a rede e entradas para interrupções externas.

A execução das tarefas e das rotinas de serviço das interrupções é efectuada através de funções de código definidas pelo utilizador. Essas funções podem ser escritas em C++ ou em Matlab. Os algoritmos de controlo podem também ser implementados graficamente utilizando os blocos existentes do Simulink.

A principal função do Kernel TrueTime é a de executar tarefas. Essas tarefas podem ser periódicas ou aperiódicas.

O simulador permite associar a cada tarefa um tempo de execução correspondente ao tempo de processamento do código respectivo. Esses tempos são tidos em conta quando é efectuada a simulação. Cada tarefa é caracterizada pela função que contém o código respectivo, por um conjunto de atributos estáticos como o *deadline* relativo, a prioridade, o tempo de execução no pior caso e o período (para as tarefas periódicas) e por um conjunto de atributos dinâmicos como o *deadline* absoluto, o tempo de *release* e por uma quota temporal de execução. Os atributos dinâmicos são actualizados pelo *kernel* à medida que a simulação avança. Tanto os atributos estáticos como os atributos dinâmicos podem ser alterados pelo utilizador durante a simulação.

As interrupções podem, como foi referido, ser internas ou externas. As interrupções externas estão associadas aos canais de interrupção externos e são evocadas quando o sinal associado ao canal muda de valor. As interrupções internas estão associadas a

temporizadores. Podem definir-se temporizadores periódicos ou de ocorrência única. Neste caso a interrupção é evocada quando o tempo associado ao temporizador termina. As interrupções são servidas através da execução de uma função de código definida pelo utilizador. O serviço da interrupção é definido por um nome, uma prioridade e uma função de código. Dentro do *kernel* a execução das funções de serviço à interrupção tem prioridade sobre a execução das tarefas.

A execução do simulador ocorre a três níveis: nível das interrupções, nível do *kernel* e nível da tarefa. O nível das interrupções é o mais prioritário. Neste nível os serviços são escalonados de acordo com a sua prioridade. Ao nível do *kernel* simulam-se as trocas de contexto. Finalmente ao nível da tarefa o escalonamento pode ser efectuado utilizando esquemas dinâmicos. A prioridade de cada tarefa é dada por uma função de prioridade definida pelo utilizador que depende dos atributos da tarefa. Existem funções de prioridade já definidas para implementar as políticas de escalonamento RM, DM, FP e EDF. A execução de cada tarefa ou serviço à interrupção pode ser efectuada com ou sem preempção.

O código associado a cada tarefa ou rotina de serviço à interrupção pode ser dividido em segmentos. O código pode interagir com outras tarefas ou com o ambiente no início de cada segmento. O tempo de execução de cada segmento pode ser constante, aleatório ou dependente dos dados processados. Associado ao código da tarefa existe uma estrutura de dados que corresponde à sua memória local.

A cada bloco Kernel está associado um ficheiro de inicialização onde são descritas as características do *kernel* como por exemplo: o número de conversores A/D e D/A utilizados, a política de escalonamento utilizada e a lista de tarefas e rotinas de serviço à interrupção a ela associados. No anexo B são apresentados exemplos de um ficheiro de inicialização e de uma função de código.

A sincronização entre tarefas é efectuada através de monitores e de eventos. Os monitores utilizam-se para garantir o acesso exclusivo a dados partilhados. Os eventos podem ser associados aos monitores ou podem ser autónomos. Neste último caso utilizam-se para exprimir condições de sincronização que não envolvam a partilha de dados entre tarefas.

No ficheiro de inicialização e no código das tarefas utilizam-se funções e comandos do simulador TrueTime com finalidades diversas como por exemplo, inicializar o *kernel*, criar uma tarefa, declarar um evento, criar um temporizador, etc. A lista de funções e comandos é apresentada no anexo B.

TrueTime Network

Este bloco é *event-driven* sendo executado quando as mensagens entram ou saem da rede. Cada mensagem contém informação sobre o nó emissor e o nó receptor, uma área de dados, o comprimento da mensagem e atributos facultativos como o *deadline* e a prioridade.

Dentro deste bloco é possível especificar a taxa de transmissão, o tipo de MAC, entre outros parâmetros. O MAC pode ser do tipo CSMA/CD, CSMA/NBA, *round-robin*, FDMA (*Frequency Division Multiple Access*) e TDMA.

Quando a simulação do envio da mensagem termina ela é disponibilizada num *buffer* no nó receptor, que é notificado através de uma interrupção.

A implementação das arquitecturas base A1 e A2 em TrueTime deu origem aos esquemas apresentados nas figuras 4.8 e 4.9.

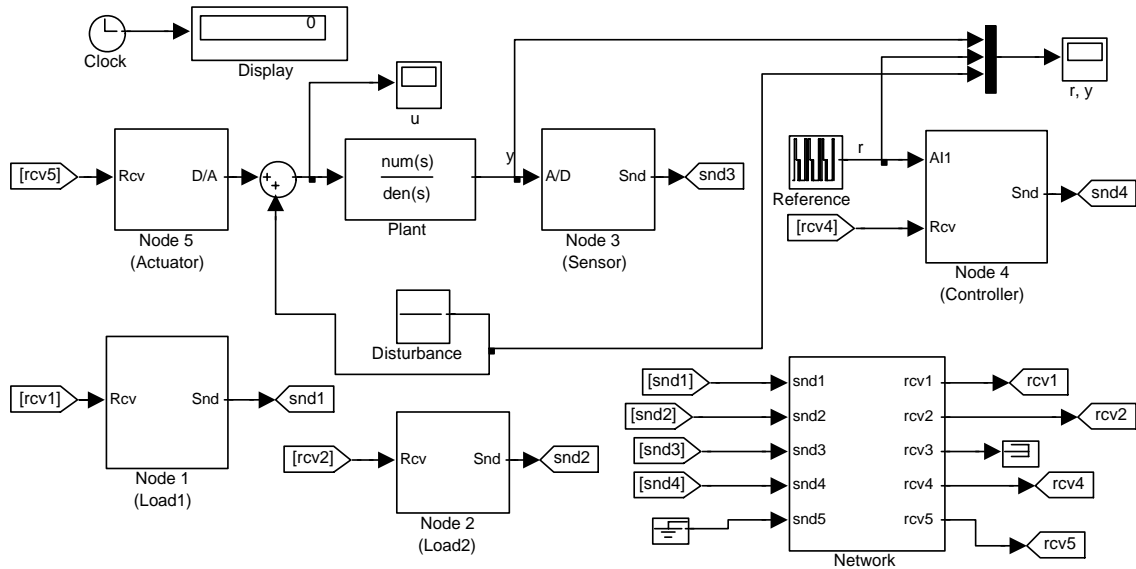


Figura 4.8: Diagrama da implementação em TrueTime da arquitectura A1.

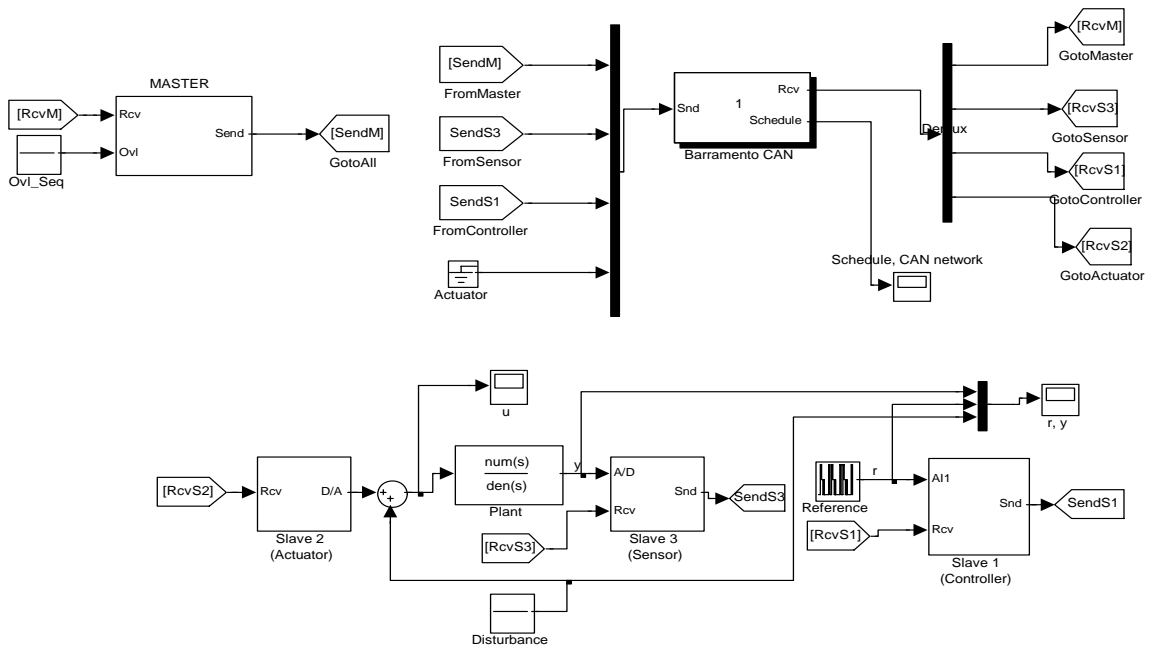


Figura 4.9: Diagrama da implementação em TrueTime da arquitectura A2.

Nos ensaios apresentados neste e no próximo capítulo utilizou-se a versão 1.13 do simulador TrueTime, com um *patch* fornecido pelos autores que permite a utilização de *broadcasts*. Esta versão é equivalente à versão 1.2 do simulador TrueTime disponibilizada em Outubro de 2004. A versão mais actual do simulador é a versão 1.6, disponibilizada em Junho de 2008. A versão do MATLAB utilizada nos ensaios foi a versão 6.5.1 (Release 13 Service Pack 1).

4.2 Controladores adaptativos por posicionamento de pólos

Esta secção apresenta os resultados obtidos utilizando controladores por posicionamento de pólos em que a identificação do sistema é feita tendo por base um modelo discreto do sistema que não tem em conta o tempo morto fraccionário introduzido na malha de controlo pela implementação distribuída.

O período de amostragem utilizado nos testes apresentados de seguida é igual a 0.28 s, correspondente ao valor mais elevado do critério de Aström e Wittenmark. A utilização deste valor permite evidenciar de forma mais clara o efeito do atraso variável entre a amostragem e a actuação no sistema de controlo [125].

Os testes foram efectuados para cada um dos sistemas S1 a S4, apresentados na tabela 4.4, utilizando os padrões de carga descritos em 4.1.2.

O desempenho de controlo foi avaliado utilizando os índices ISE e ITSE calculados entre os instantes $t = 10$ s e $t = 29$ s para que a análise seja efectuada após o período de convergência do estimador.

4.2.1 Sistema S1

O sistema S1 tem a seguinte função de transferência, apresentada anteriormente na tabela 4.4.

$$G(s) = \frac{0.5}{s + 0.5}$$

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.10.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.11, 4.12 e 4.13.

Nas figuras seguintes o eixo vertical não apresenta unidades visto que os resultados foram obtidos por simulação, não tendo por isso correspondência directa com nenhuma grandeza física.

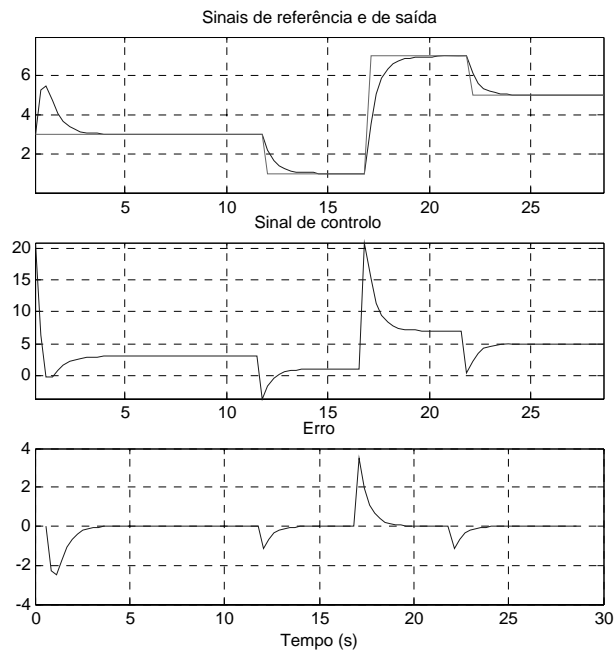


Figura 4.10: Sinais para o teste de referência (sem TMF), sistema S1.

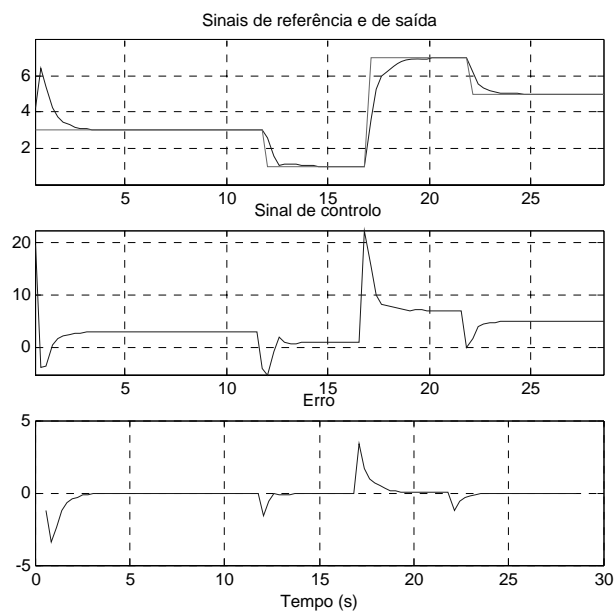


Figura 4.11: Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S1.

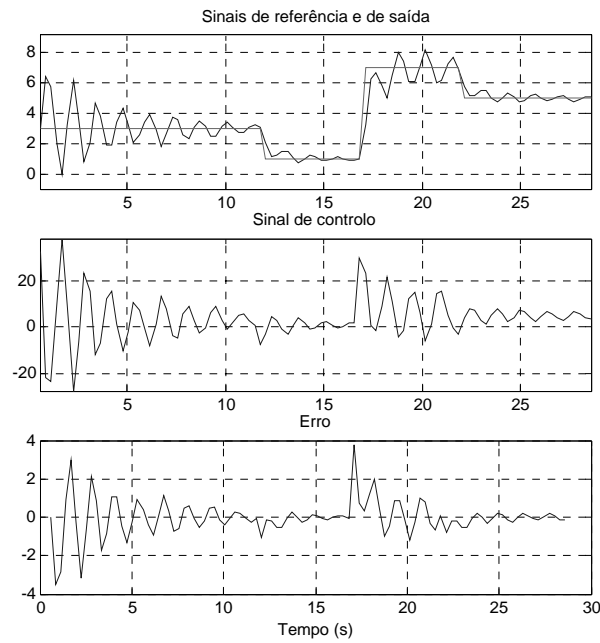


Figura 4.12: Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S1.

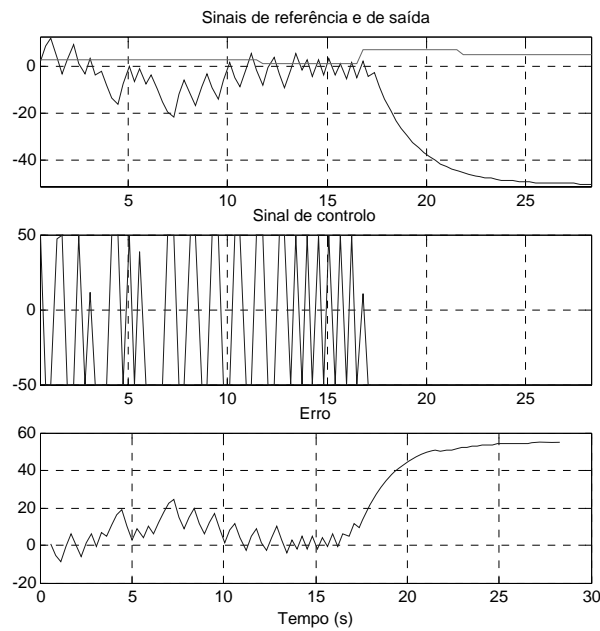


Figura 4.13: Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S1.

A análise das figuras permite verificar que os atrasos introduzidos na malha pelo padrão P1 não provocam alterações muito significativas na saída do sistema. Para o padrão de carga P2 o sinal de saída apresenta oscilações ao longo de todo o ensaio mas continua a seguir o sinal de referência. Para o padrão de carga P3 o sinal de saída apresenta fortes oscilações durante a metade inicial do teste, saturando negativamente na parte final do mesmo.

A tabela 4.7 apresenta o valor obtido para os índices de desempenho ISE e ITSE.

Teste	Referência	P1	P2	P3
ISE	18.3	18.2	21.8	26505
ITSE	310	307	366	617000

Tabela 4.7: Valor de ISE e de ITSE para os testes (sem TMF), sistema S1.

O valor de ISE confirma a análise efectuada através da observação do sinal de saída. Para o teste com o padrão de carga P1 o valor de ISE é praticamente igual ao do ensaio de referência. Para os outros padrões de carga o valor de ISE sofre alterações em relação ao valor obtido no teste de referência. No caso do teste com o padrão de carga P2 o valor obtido para ISE é muito semelhante ao do teste de referência. No teste com o padrão de carga P3 e, tal como seria de esperar, o valor de ISE é muito elevado.

Os valores obtidos com o critério ITSE diferem em ordem de grandeza dos de ISE, como seria de esperar pelo facto dos erros que acontecem mais tarde serem mais pesados, mas as variações relativas do valor de ITSE entre os vários testes são semelhantes.

O modelo utilizado para a função de transferência discreta, que não tem em conta o atraso introduzido na malha de controlo pela distribuição, é dado pela equação 4.3:

$$G(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} \quad (4.3)$$

As figuras 4.14 a 4.17 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios.

A tabela 4.8 apresenta o valor dos parâmetros (módulo) do modelo discreto do sistema identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	0.869	0.871	0.878	0.921	1.010
b_1	0.131	0.129	0.124	0.090	0.006

Tabela 4.8: Parâmetros identificados no final dos testes (sem TMF), sistema S1.

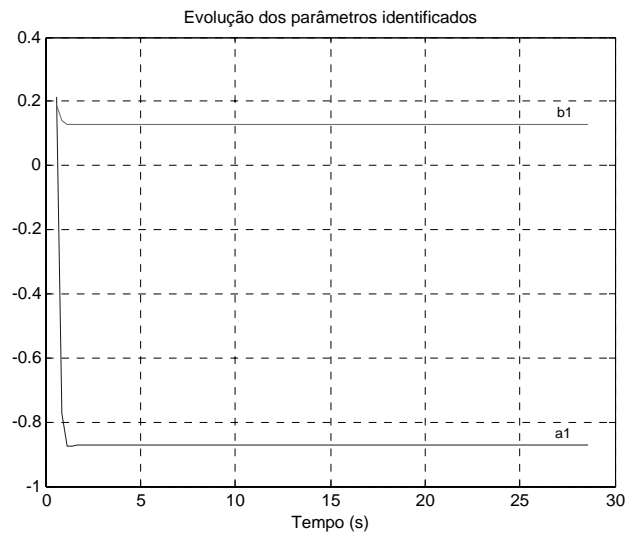


Figura 4.14: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S1.

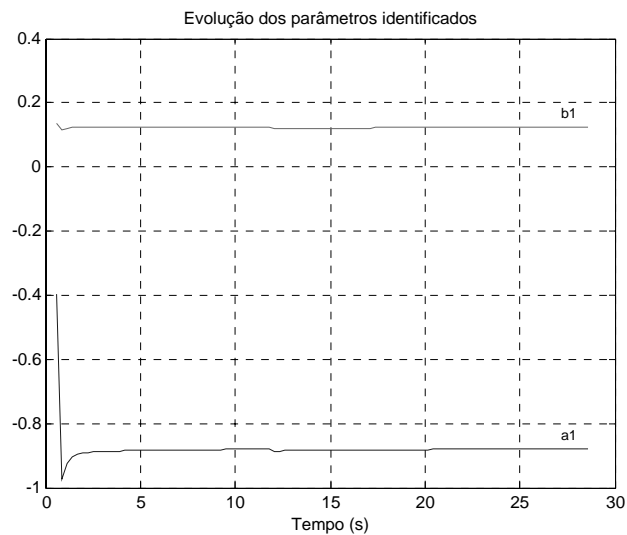


Figura 4.15: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S1.

A tabela 4.9 apresenta a localização do pólo (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes para o sistema S1.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.869	0.871	0.878	0.921	1.010

Tabela 4.9: Localização do pólo identificado (sem TMF), sistema S1.

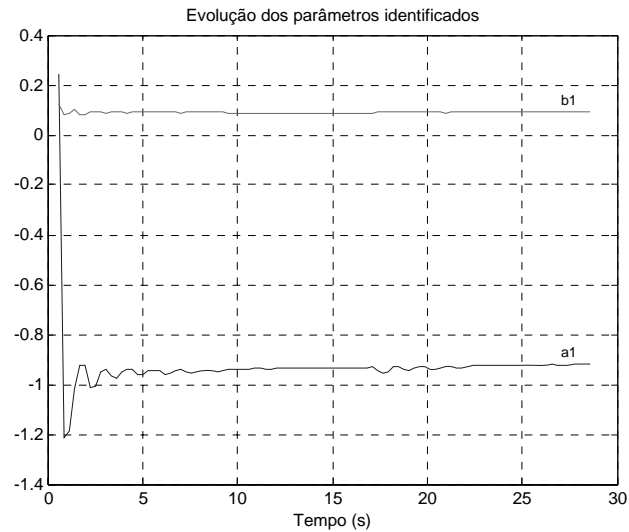


Figura 4.16: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S1.

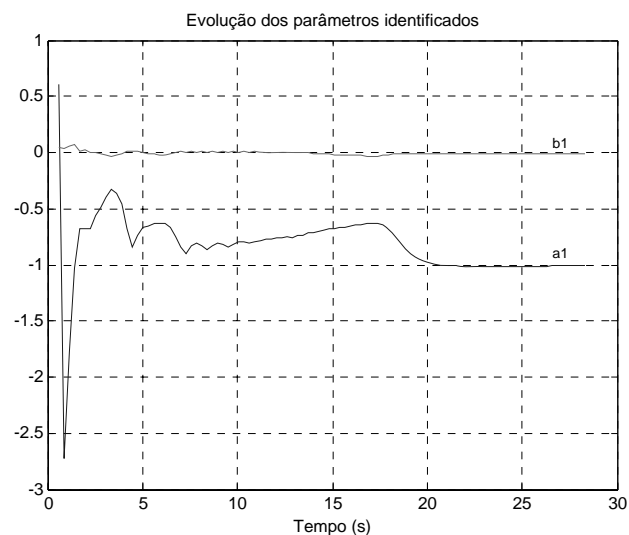


Figura 4.17: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S1.

No ensaio de referência, sem carga no barramento, o estimador é capaz de identificar quase correctamente o parâmetro a_1 a partir do qual se determina o pólo do sistema. Contudo nos ensaios em que há atraso variável entre a amostragem e a actuação a localização do pólo identificado vai-se afastando cada vez mais do valor real à medida que esse atraso (em média) aumenta. A identificação é tanto pior quanto maior é o atraso. No ensaio com o padrão de carga P2 o estimador não consegue estabilizar apresentando pequenas oscilações enquanto que no ensaio para o padrão de carga P3 a identificação converge para um valor sobre o círculo unitário, pondo em causa a estabilidade do sistema.

4.2.2 Sistema S2

O sistema S2 tem a seguinte função de transferência, apresentada anteriormente na tabela 4.4.

$$G(s) = \frac{0.5}{s(s + 0.5)}$$

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.18.

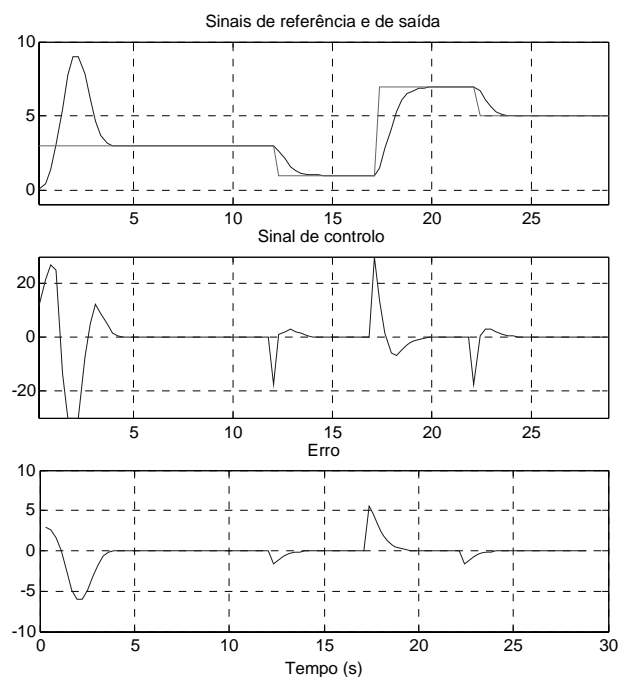


Figura 4.18: Sinais para o teste de referência (sem TMF), sistema S2.

Os resultados de controlo obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.19, 4.20 e 4.21.

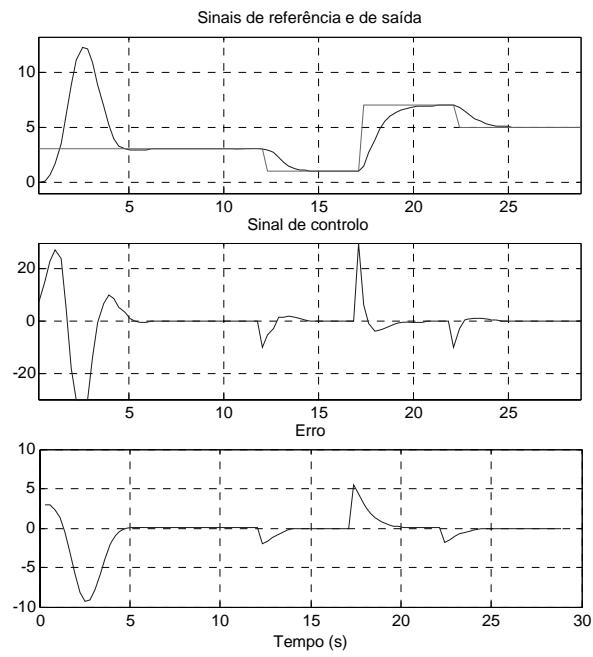


Figura 4.19: Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S2.

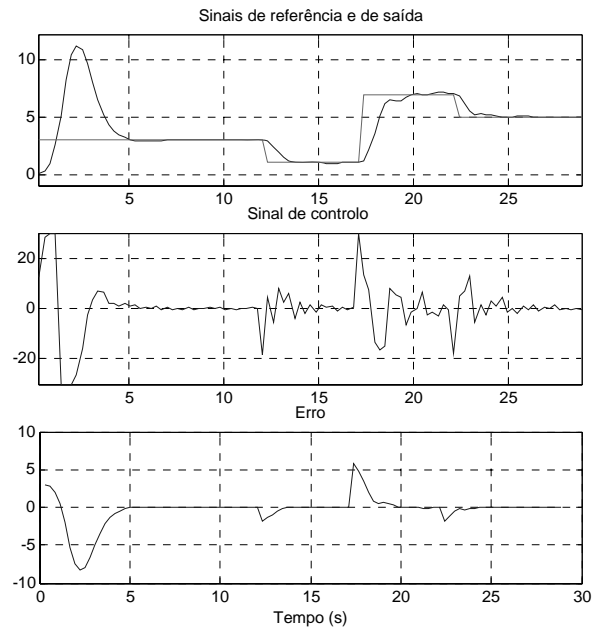


Figura 4.20: Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S2.

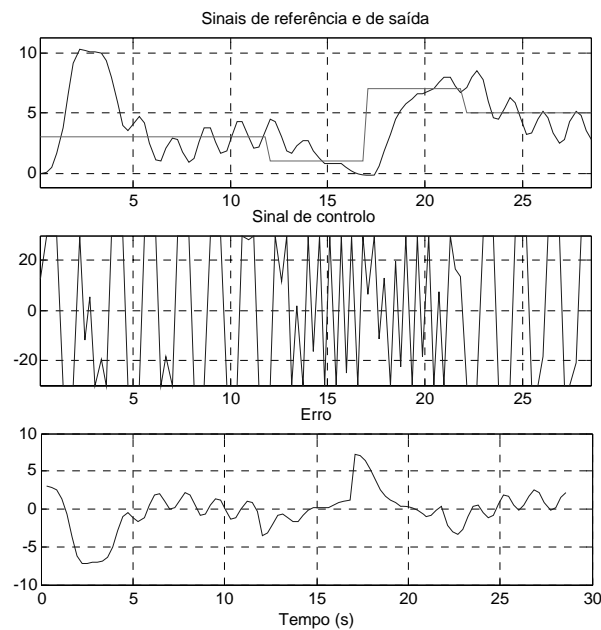


Figura 4.21: Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S2.

Para este sistema os atrasos introduzidos pelos padrões de carga P1 e P2 não provocam alterações significativas no sinal de saída, embora para o padrão P2 seja possível notar uma ligeira oscilação na metade final do teste. Os atrasos introduzidos pelo padrão de carga P3 fazem com que o sinal de saída tenha um comportamento oscilatório, perdendo-se a capacidade de controlar o sistema. O sinal de controlo apresenta ligeiras oscilações para o ensaio com o padrão de carga P2, sendo totalmente oscilatório para o ensaio com o padrão de carga P3.

A tabela 4.10 apresenta o valor dos índices ISE e ITSE para os testes efectuados.

Teste	Referência	P1	P2	P3
ISE	31.6	33.2	36.8	104.9
ITSE	549	576	639	1860

Tabela 4.10: Valor de ISE e de ITSE para os testes (sem TMF), sistema S2.

Embora os gráficos não permitam evidenciar diferenças entre o teste de referência e o teste com o padrão de carga P1, a análise dos valores de ISE e de ITSE permite concluir que há uma ligeira degradação do desempenho de controlo. Para os outros testes o valor de ISE e de ITSE confirmam as observações já efectuadas.

O modelo utilizado para a função de transferência discreta, que não tem em conta o atraso entre a amostragem e a actuação, é dado pela equação 4.4:

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1} + a_2q^{-2}} \quad (4.4)$$

As figuras 4.22 a 4.25 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados para o sistema S2.

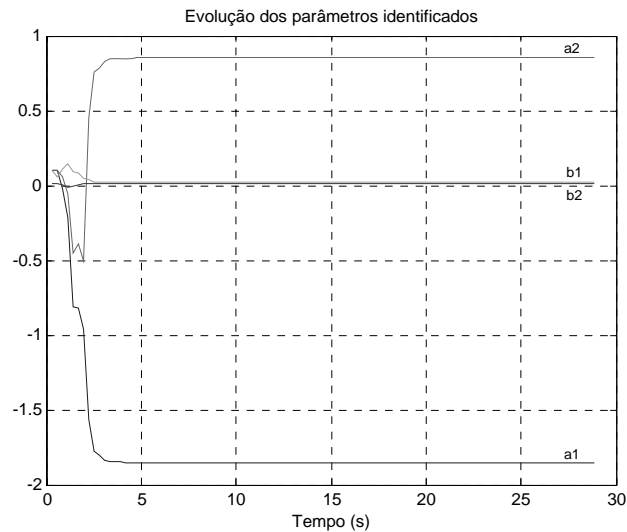


Figura 4.22: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S2.

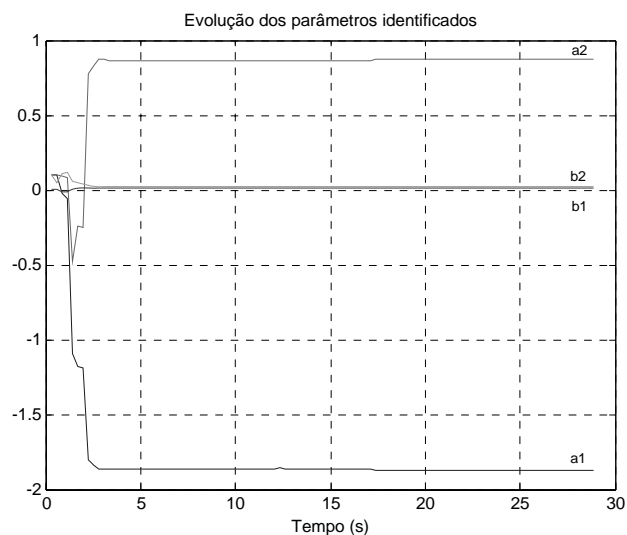


Figura 4.23: Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P1 (sem TMF), sistema S2.

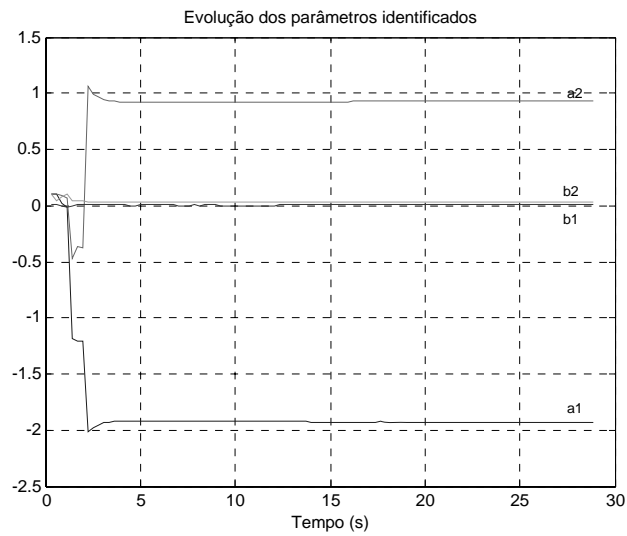


Figura 4.24: Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P2 (sem TMF), sistema S2.

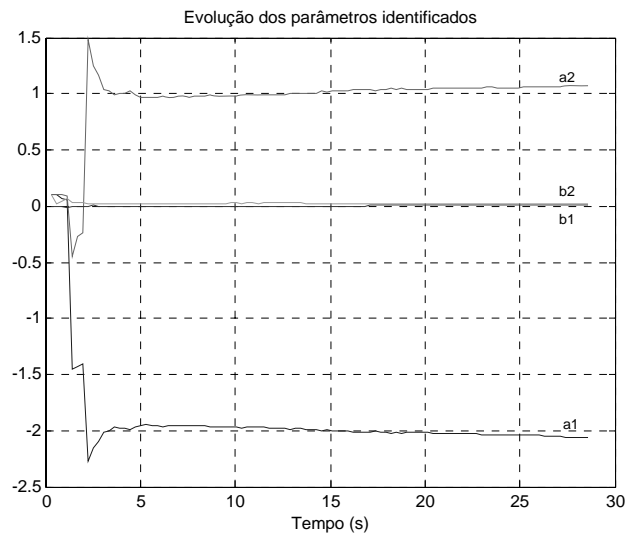


Figura 4.25: Evolução dos parâmetros do modelo discreto do sistema para o teste com padrão de carga P3 (sem TMF), sistema S2.

A tabela 4.11 apresenta o valor dos parâmetros do modelo discreto do sistema (módulo) identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.869	1.859	1.870	1.934	2.058
a_2	0.869	0.859	0.870	0.936	1.075
b_1	0.019	0.017	0.013	0.004	0.004
b_2	0.018	0.019	0.024	0.030	0.022

Tabela 4.11: Parâmetros identificados no final dos testes (sem TMF), sistema S2.

A tabela 4.12 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S2.

Pólo	Teórico	Referência	P1	P2	P3
p_1	1	1	1	0.967+j0.030	1.029+j0.127
p_2	0.869	0.859	0.870	0.967-j0.030	1.029-j0.127

Tabela 4.12: Localização dos pólos identificados (sem TMF), sistema S2.

A análise do valor identificado para os pólos da função discreta do sistema confirma os resultados obtidos para o valor de ISE e de ITSE. Com o aumento do atraso médio introduzido na malha a identificação dos pólos afasta-se do valor esperado. Para o teste com o padrão P3 o estimador apresenta oscilações permanentes nos parâmetros identificados, que têm valores fora do círculo unitário, o que justifica a oscilação obtida no sinal de saída.

4.2.3 Sistema S3

O sistema S3 tem a seguinte função de transferência, apresentada anteriormente na tabela 4.4.

$$G(s) = \frac{0.25}{(s+1)(s+0.25)}$$

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.26.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.27, 4.28 e 4.29.

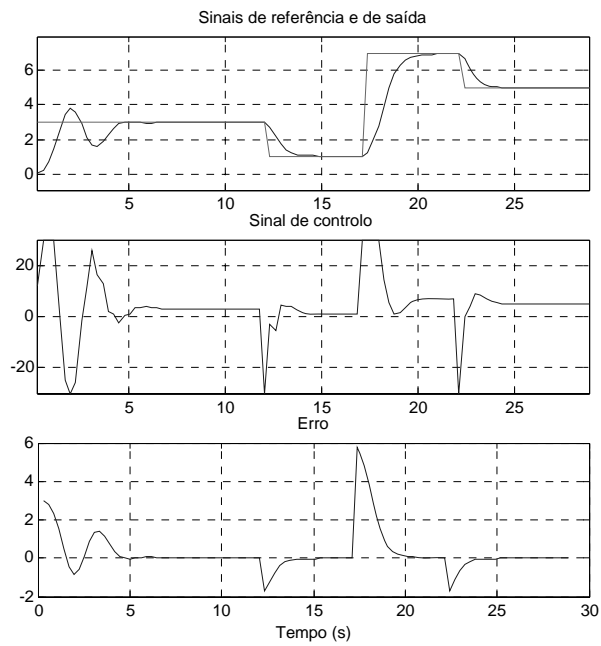


Figura 4.26: Sinais para o teste de referência (sem TMF), sistema S3.

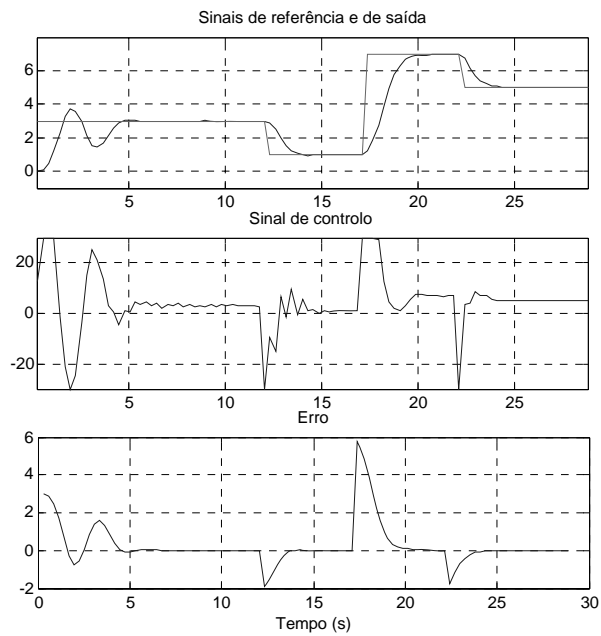


Figura 4.27: Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S3.

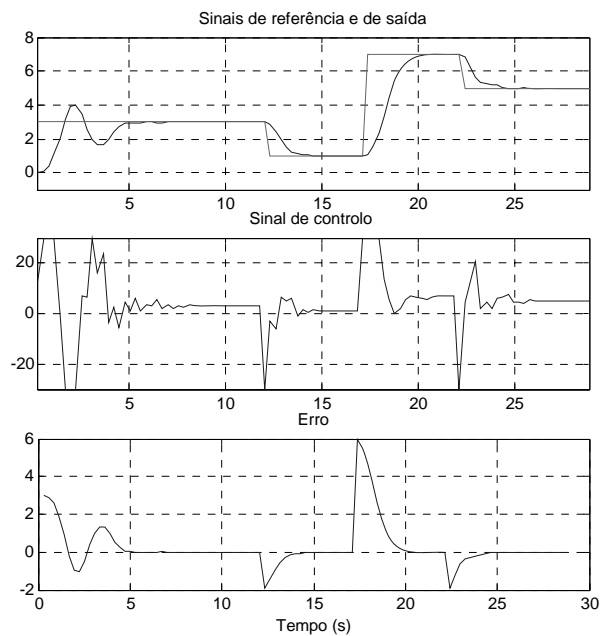


Figura 4.28: Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S3.

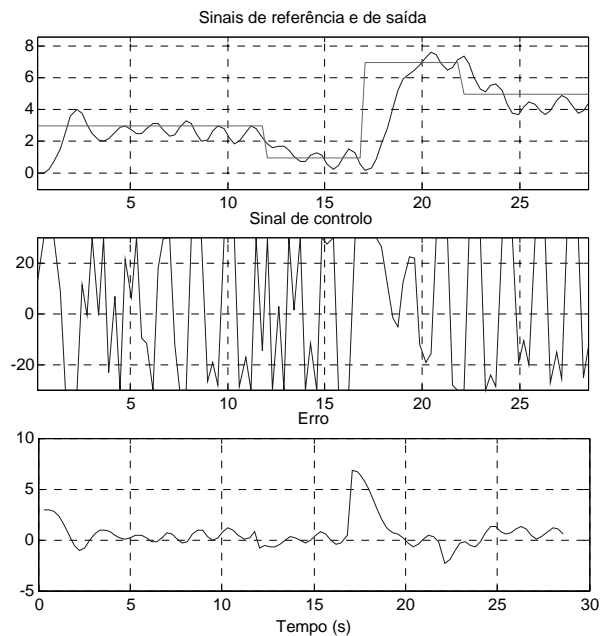


Figura 4.29: Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S3.

O sinal de controlo apresenta ligeiras oscilações nos ensaios com os padrões de carga P1 e P2, sendo totalmente oscilatório no ensaio com o padrão de carga P3. Os padrões de carga P1 e P2 não alteram significativamente o sinal de saída. O padrão P3 faz com que o sinal de saída apresente oscilações que põem em causa a capacidade de controlo do sistema.

A tabela 4.13 apresenta o valor dos índices ISE e ITSE para os testes apresentados.

Teste	Referência	P1	P2	P3
ISE	41.1	42.2	46.8	78.0
ITSE	718	735	820	1370

Tabela 4.13: Valor de ISE e de ITSE para os testes (sem TMF), sistema S3.

A análise do valor de ISE e de ITSE evidencia a diminuição da qualidade de controlo à medida que o atraso introduzido aumenta (em média). Para o ensaio com o padrão de carga P3 a degradação do desempenho de controlo, como era esperado, é grande.

O modelo utilizado para a função de transferência discreta deste sistema é igual ao do sistema S2 (ver equação 4.4). As figuras 4.30 a 4.33 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados.

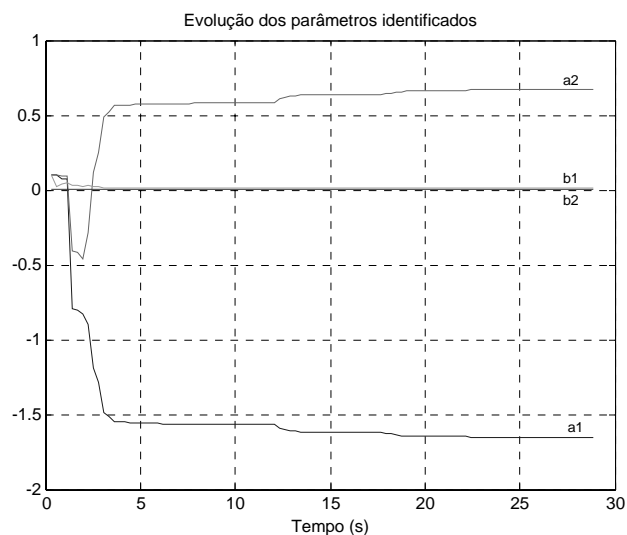


Figura 4.30: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S3.

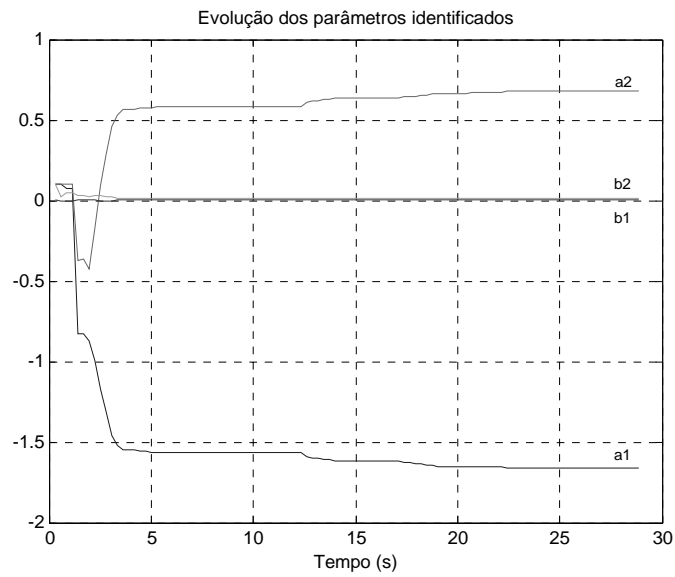


Figura 4.31: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S3.

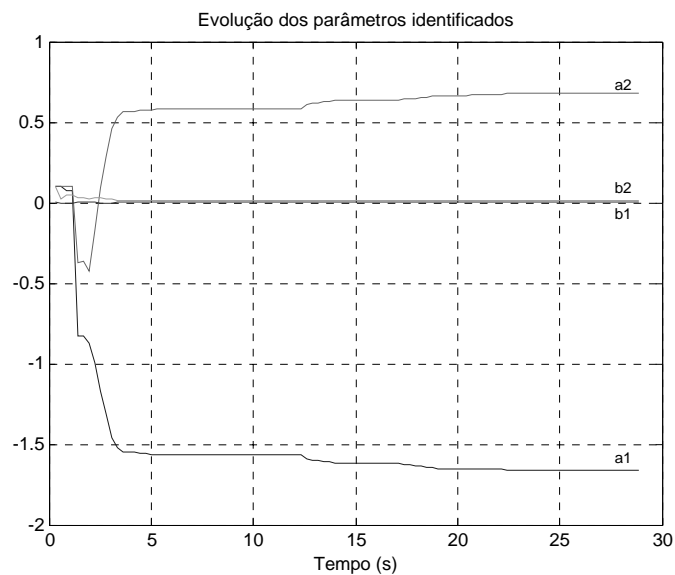


Figura 4.32: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S3.

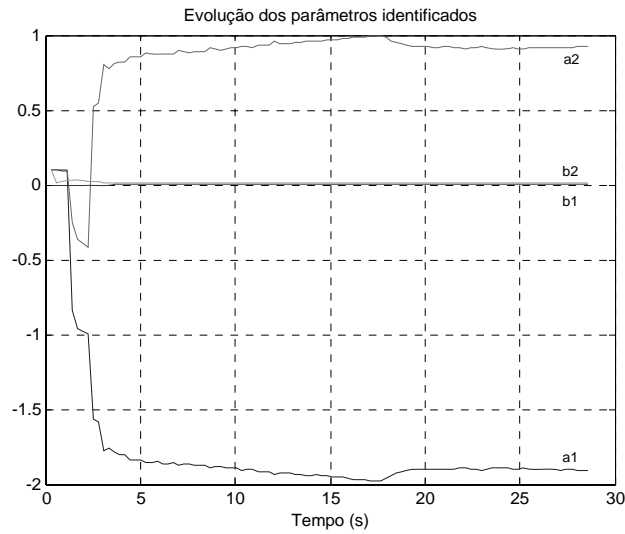


Figura 4.33: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S3.

A tabela 4.14 apresenta o valor dos parâmetros (módulo) do modelo discreto do sistema identificados no final de cada um dos testes anteriores.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.688	1.657	1.665	1.730	1.905
a_2	0.705	0.674	0.681	0.747	0.921
b_1	0.009	0.008	0.006	0.002	0.001
b_2	0.008	0.009	0.011	0.013	0.012

Tabela 4.14: Parâmetros identificados no final dos testes (sem TMF), sistema S3.

A tabela 4.15 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S3.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.940	0.940	0.942	0.900	$0.117+j0.953$
p_2	0.758	0.717	0.723	0.830	$0.117-j0.953$

Tabela 4.15: Localização dos pólos identificados (sem TMF), sistema S3.

A análise do valor identificado para os pólos da função discreta do sistema confirma os resultados obtidos para o valor de ISE e de ITSE. Com o aumento do valor médio do atraso introduzido na malha a identificação dos parâmetros afasta-se do valor esperado.

Tal como no caso do sistema anterior para o ensaio com o padrão de carga P3 os parâmetros identificados apresentam oscilações ao longo de todo o teste aproximando-se ocasionalmente do círculo unitário o que contribui para a oscilação do sinal de saída neste ensaio.

4.2.4 Sistema S4

O sistema S4 tem a seguinte função de transferência, apresentada anteriormente na tabela 4.4.

$$G(s) = \frac{1}{s^2 + s + 1}$$

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.34.

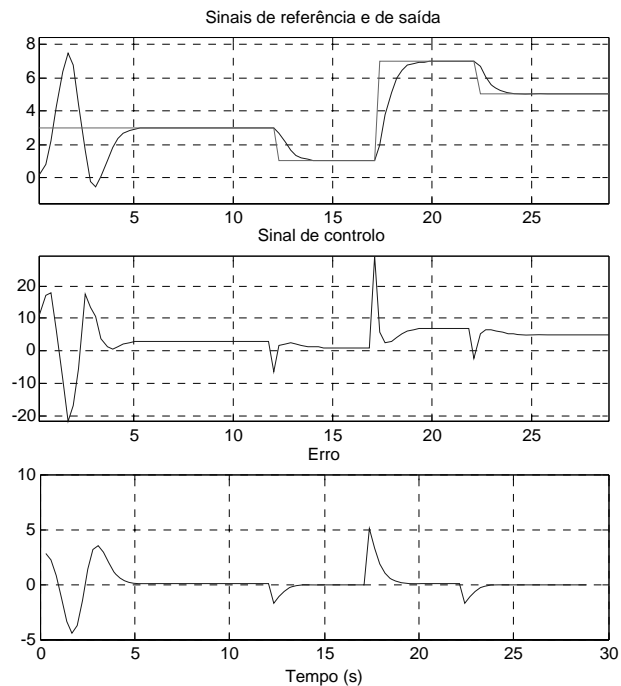


Figura 4.34: Sinais para o teste de referência (sem TMF), sistema S4.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.35, 4.36 e 4.37.

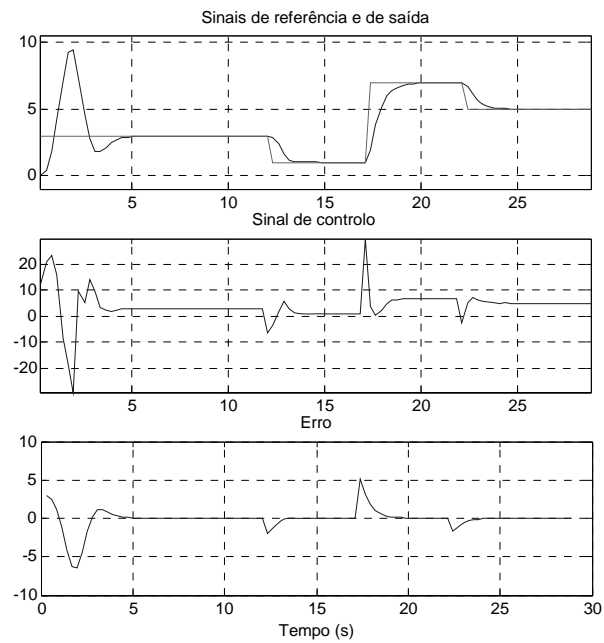


Figura 4.35: Sinais para o teste com o padrão de carga P1 (sem TMF), sistema S4.

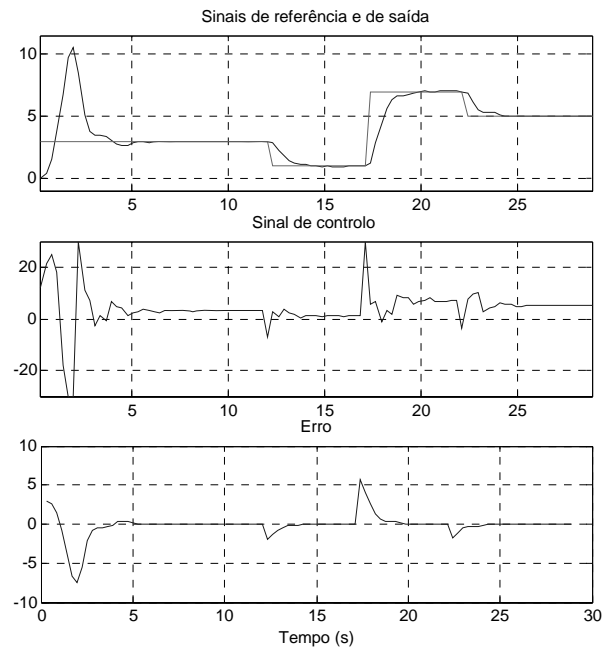


Figura 4.36: Sinais para o teste com o padrão de carga P2 (sem TMF), sistema S4.

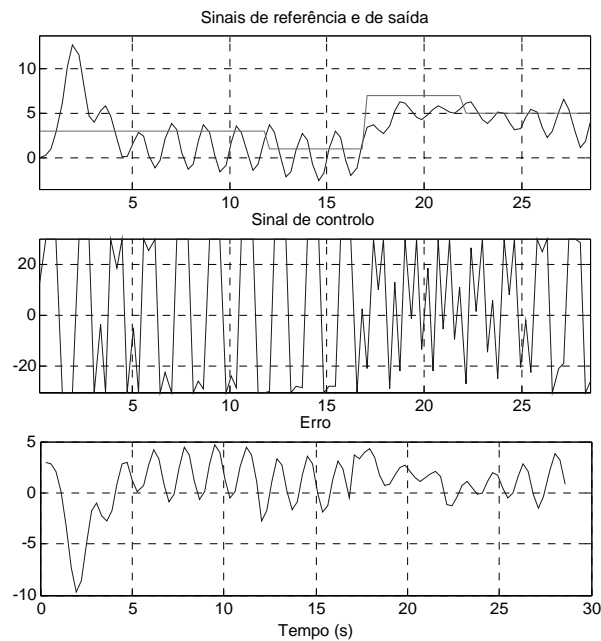


Figura 4.37: Sinais para o teste com o padrão de carga P3 (sem TMF), sistema S4.

O ensaio com o padrão de carga P1 produz resultados muito semelhantes aos do ensaio de referência. O sinal de controlo apresenta pequenas oscilações para o ensaio com o padrão de carga P2, sendo totalmente oscilatório para o ensaio com o padrão de carga P3. Os padrões de carga P1 e P2 não alteram significativamente o sinal de saída. O padrão P3 faz com que o sinal de saída apresente oscilações que põem em causa a capacidade de controlo do sistema.

A tabela 4.16 apresenta o valor dos índices ISE e ITSE obtido nos testes efectuados ao sistema S4.

Teste	Referência	P1	P2	P3
ISE	26.3	26.8	32.2	103.1
ITSE	456	462	558	1730

Tabela 4.16: Valor de ISE e de ITSE para os testes (sem TMF), sistema S4.

Tal como para os casos anteriores o desempenho de controlo diminui à medida que o atraso introduzido aumenta. Para o ensaio com o padrão de carga P3 a degradação do desempenho de controlo é grande, como era esperado.

O modelo utilizado para a função de transferência discreta deste sistema é igual ao do sistema S2 e S3 (ver equação 4.4). As figuras 4.38 a 4.41 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados.

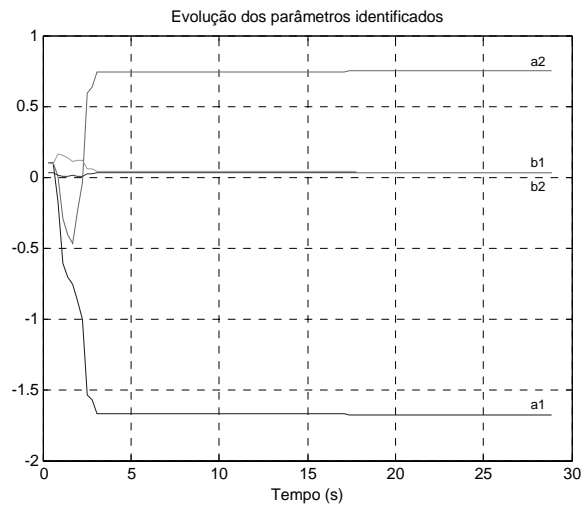


Figura 4.38: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (sem TMF), sistema S4.

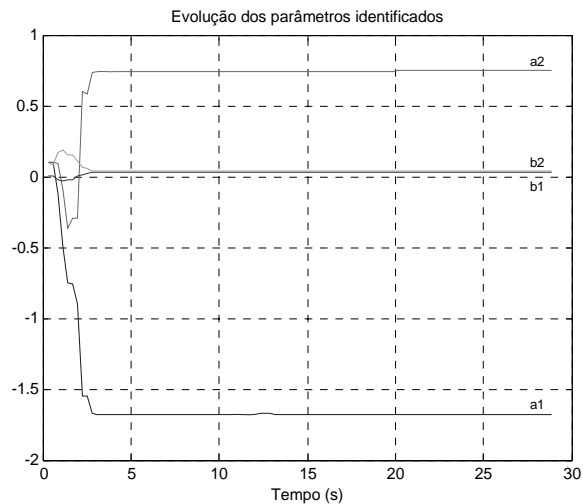


Figura 4.39: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (sem TMF), sistema S4.

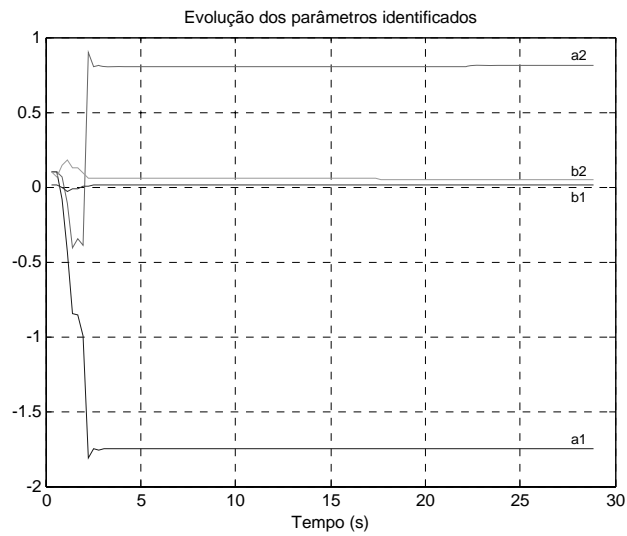


Figura 4.40: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (sem TMF), sistema S4.

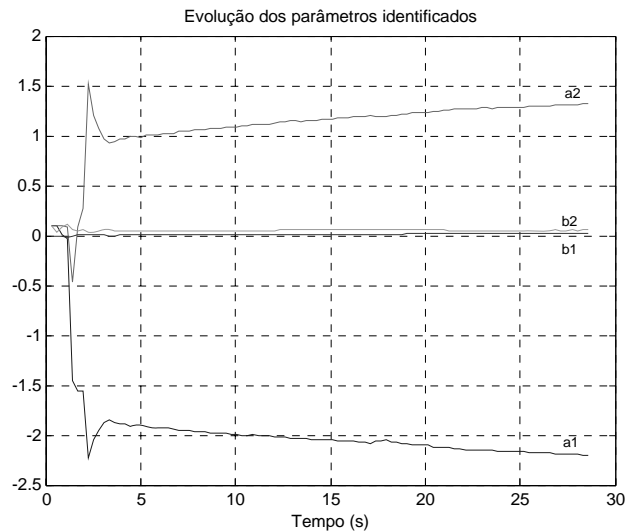


Figura 4.41: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (sem TMF), sistema S4.

A tabela 4.17 apresenta o valor dos parâmetros do modelo discreto do sistema (módulo) identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.688	1.681	1.682	1.749	2.204
a_2	0.756	0.750	0.749	0.820	1.319
b_1	0.036	0.033	0.028	0.009	0.021
b_2	0.032	0.035	0.039	0.052	0.054

Tabela 4.17: Parâmetros identificados no final dos testes (sem TMF), sistema S4.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.844+j0.209	0.841+j0.209	0.841+j0.204	0.875+j0.213	1.102+j0.323
p_2	0.844-j0.209	0.841-j0.209	0.841-j0.204	0.875-j0.213	1.102-j0.323

Tabela 4.18: Localização dos pólos identificados (sem TMF), sistema S4.

A tabela 4.18 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S4.

A análise dos resultados obtidos para o sistema S4 é semelhante à efectuada para o sistema S3.

Em resumo, nos sistemas em que o atraso introduzido na malha de controlo não é tido em conta no projecto do controlador, a qualidade da identificação dos pólos da função de transferência discreta do sistema é cada vez pior à medida que o valor médio do atraso introduzido aumenta, o que implica a degradação do desempenho de controlo. Em certos casos o sistema torna-se oscilatório e perde-se a capacidade de controlo em malha fechada.

4.3 Controladores adaptativos distribuídos por posicionamento de pólos

Para melhorar o desempenho de controlo de uma malha afectada por atrasos variáveis entre o instante da amostragem e o instante da actuação propõe-se a utilização de um modelo discreto do sistema que tenha em conta o tempo morto fraccionário introduzido na malha de controlo pela implementação distribuída. Os resultados obtidos através da utilização de controladores por posicionamento de pólos em que a identificação do sistema é feita tendo por base esta técnica são apresentados de seguida.

O período de amostragem utilizado nos testes apresentados é igual a 0.28 s, tal como na secção anterior.

Os testes foram efectuados para cada um dos sistemas S1 a S4, apresentados na tabela 4.4, utilizando cada um dos padrões de carga descritos em 4.1.2.

O desempenho de controlo foi avaliado utilizando os índices ISE e ITSE calculados entre os instantes $t = 10$ s e $t = 29$ s, como anteriormente.

4.3.1 Sistema S1

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.42.

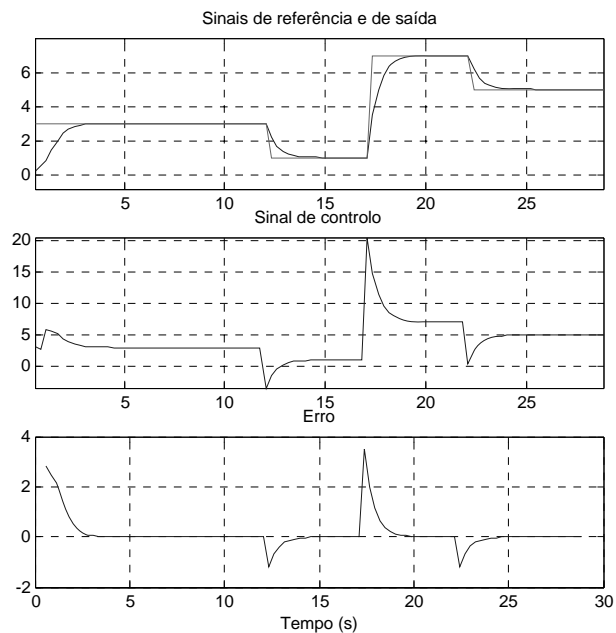


Figura 4.42: Sinais para o teste de referência (com TMF), sistema S1.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.43, 4.44 e 4.45.

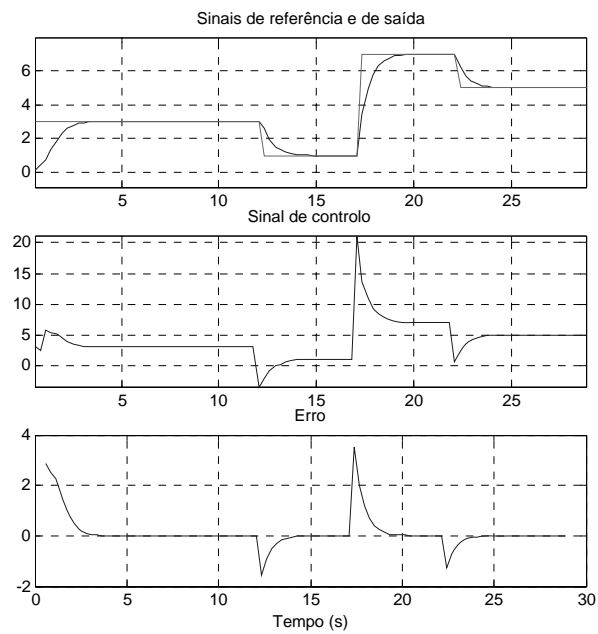


Figura 4.43: Sinais para o teste com o padrão de carga P1 (com TMF), sistema S1.

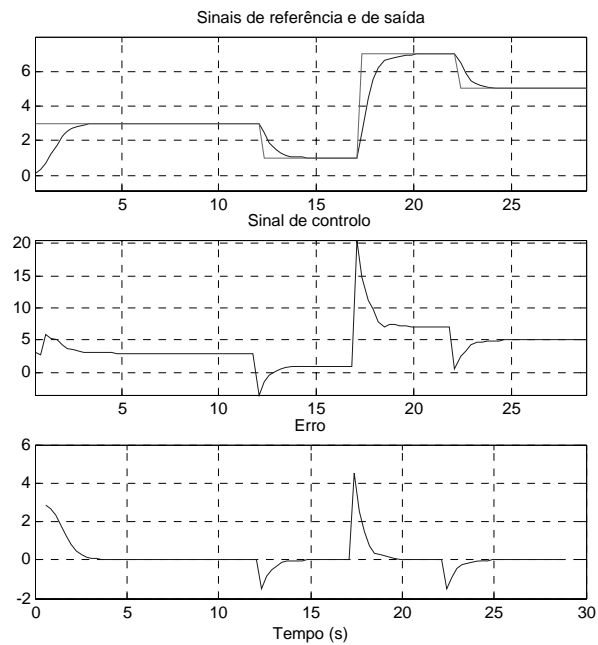


Figura 4.44: Sinais para o teste com o padrão de carga P2 (com TMF), sistema S1.

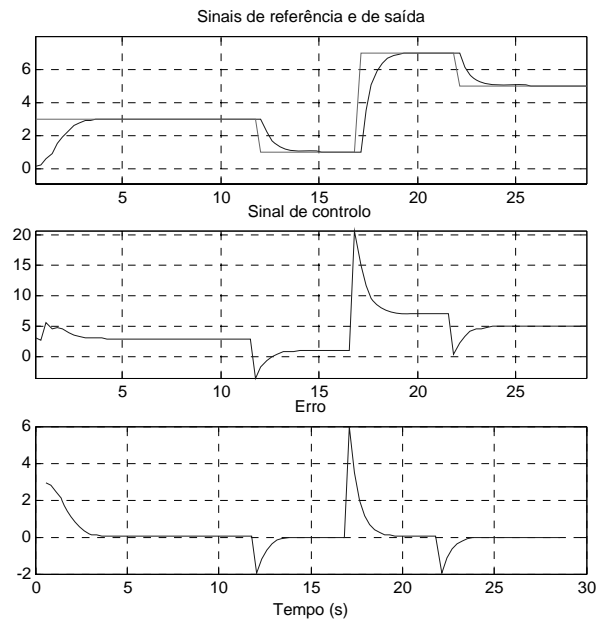


Figura 4.45: Sinais para o teste com o padrão de carga P3 (com TMF), sistema S1.

O sinal de controlo não apresenta oscilações em nenhum dos ensaios. Quanto ao sinal de saída, em todos os testes efectuados, mesmo no teste com o padrão de carga P3, o sinal de saída não apresenta oscilações nem há perda do controlo do sistema.

A tabela 4.19 apresenta o valor obtido para os índices de desempenho ISE e ITSE.

Teste	Referência	P1	P2	P3
ISE	18.6	19.3	22.5	30.8
ITSE	320	330	390	524

Tabela 4.19: Valor de ISE e de ITSE para os testes (com TMF), sistema S1.

O valor de ISE (e de ITSE) vai aumentando à medida que o valor médio do atraso introduzido aumenta.

De modo a tornar mais fácil a comparação dos resultados apresenta-se de seguida uma tabela com a relação percentual entre o valor de ISE e de ITSE para os controladores adaptativos com e sem TMF.

A tabela 4.20 apresenta a melhoria obtida com a utilização do controlador distribuído para o sistema S1.

Teste	Referência	P1	P2	P3
M_{ISE}	-2%	-6%	-3%	+99.9%
M_{ITSE}	-3%	-7%	-7%	+98%

Tabela 4.20: Melhoria para o sistema S1.

O valor da melhoria, em percentagem, foi obtido através da expressão 4.5 que relaciona o valor de cada índice de controlo nos ensaios com (cTMF) e sem (sTMF) tempo morto fraccionário.

$$M_I = \left(1 - \frac{I_{cTMF}}{I_{sTMF}}\right) * 100 \quad (4.5)$$

A melhoria obtida por comparação do valor de ISE é muito semelhante à que se obtém através da comparação do valor de ITSE, excepto no ensaio com o padrão de carga P2, em que a oscilação verificada na segunda metade do teste é mais penalizada pelo índice ITSE. Como no ensaio com TMF essa oscilação deixa de existir a melhoria obtida é maior do que para o caso de ISE, onde o erro é pesado de igual modo ao longo de todo o ensaio.

É importante referir, desde logo, que para o teste de referência o valor de ISE e de ITSE é maior no ensaio com TMF do que no ensaio sem TMF, ou seja o desempenho de controlo é pior (cerca de 2%). Isto deve-se ao facto de o modelo utilizado para modelar o atraso entre o instante de amostragem e o instante de actuação possui mais um parâmetro do que o modelo que não tem em conta o atraso. O vector de regressores utilizado para estimar os parâmetros do sistema necessita de mais informação sobre o estado anterior do sistema e, conseqüentemente, a função de controlo também, o que faz com que o efeito do controlador na saída se manifeste mais tarde.

A análise do valor de ISE e de ITSE para este sistema permite concluir que o desempenho de controlo é muito melhor apenas para o caso do teste com o padrão de carga P3. É por isso importante analisar o posicionamento dos pólos identificados durante estes ensaios.

O modelo utilizado para a função de transferência discreta, que tem em conta o tempo morto fraccionário, é dado pela equação 4.6:

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2}}{1 + a_1q^{-1}} \quad (4.6)$$

As figuras 4.46 a 4.49 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados.

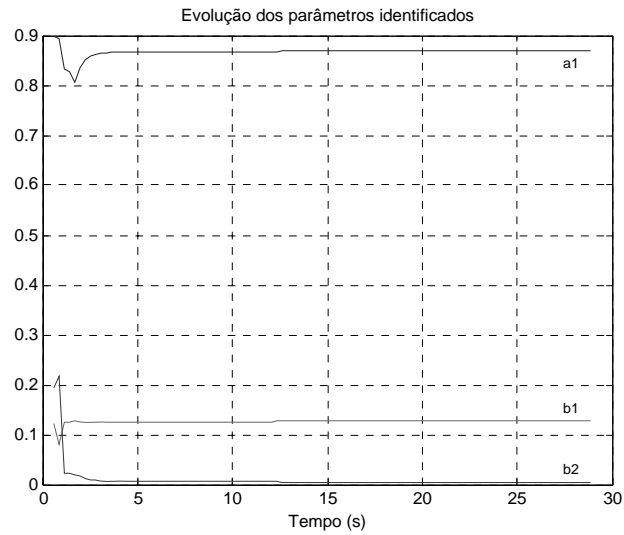


Figura 4.46: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S1.

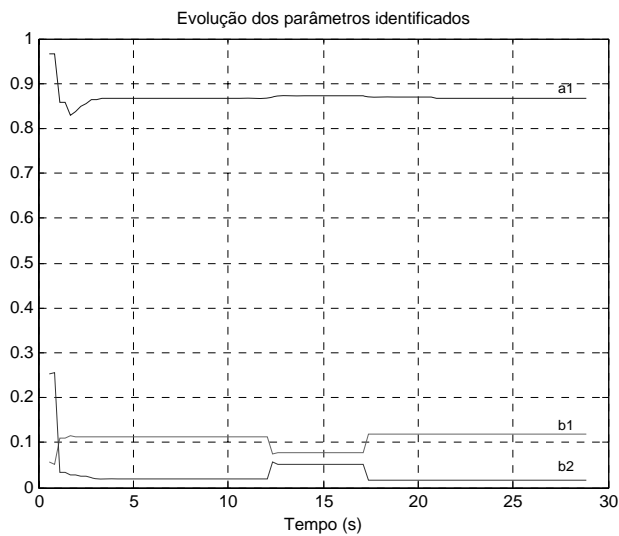


Figura 4.47: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S1.

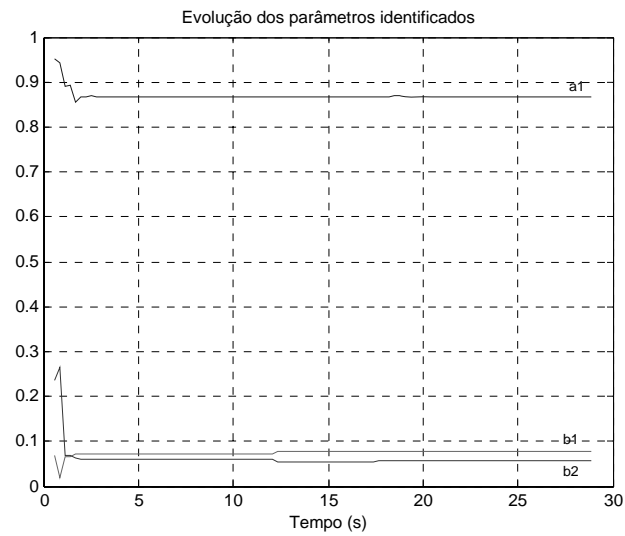


Figura 4.48: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S1.

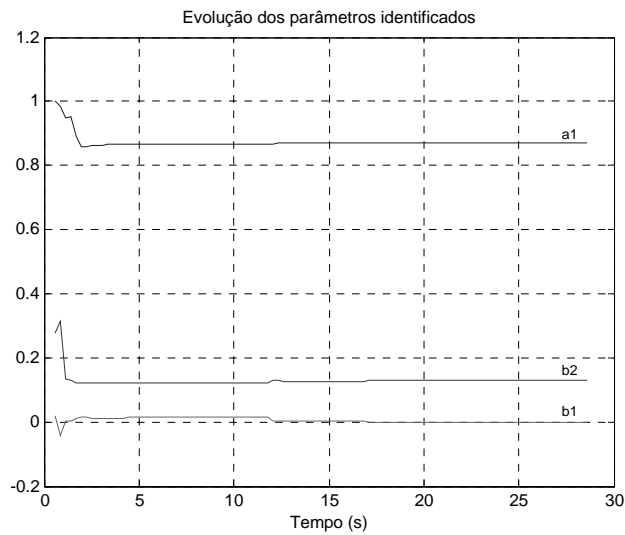


Figura 4.49: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S1.

A tabela 4.21 apresenta o valor dos parâmetros do modelo discreto do sistema (módulo) identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	0.869	0.869	0.868	0.867	0.871
b_1	0.131	0.127	0.118	0.076	0.000
b_2	-	0.004	0.015	0.057	0.129

Tabela 4.21: Parâmetros identificados no final dos testes (com TMF), sistema S1.

A tabela 4.22 apresenta a localização do pólo (em z) correspondente aos parâmetros identificados no final de cada um dos testes para o sistema S1.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.869	0.869	0.868	0.867	0.871

Tabela 4.22: Localização dos pólos identificados (com TMF), sistema S1.

Comparando esta tabela com a que se obteve utilizando o método “clássico” verifica-se que o estimador é capaz de identificar melhor o parâmetro correspondente ao pólo do sistema (a_1) em todas as situações, o que permite obter uma melhor qualidade de controlo, claramente visível nos ensaios em que o atraso introduzido piora significativamente o comportamento em malha fechada.

Pode assim concluir-se que a qualidade de controlo melhora em todos casos, embora o desempenho de controlo (medida através dos índices ISE e ITSE) só seja melhor nos casos em que a melhoria da qualidade de controlo se sobrepõe à maior “lentidão” do controlador distribuído.

4.3.2 Sistema S2

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.50.

Os resultados de controlo obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.51, 4.52 e 4.53.

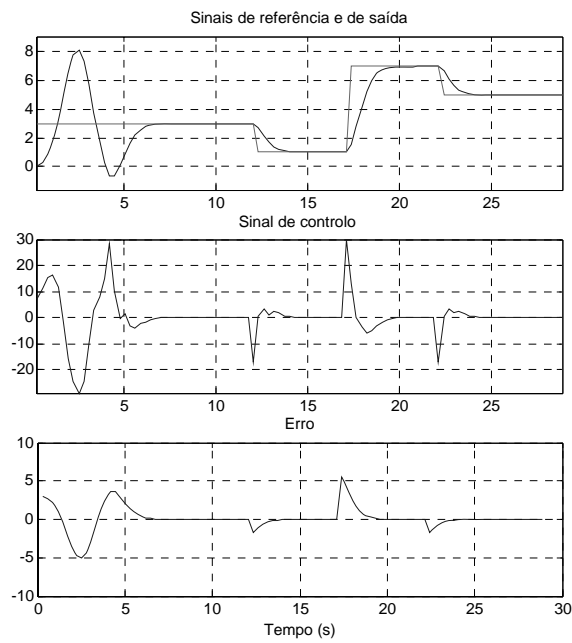


Figura 4.50: Sinais para o teste de referência (com TMF), sistema S2.

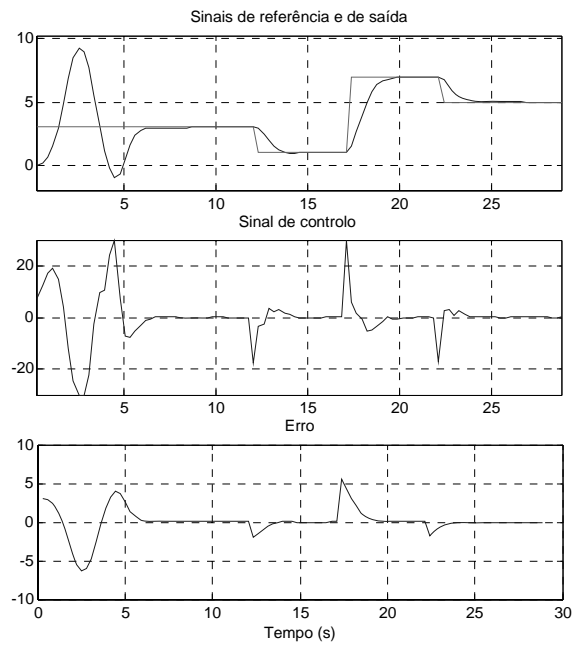


Figura 4.51: Sinais para o teste com o padrão de carga P1 (com TMF), sistema S2.

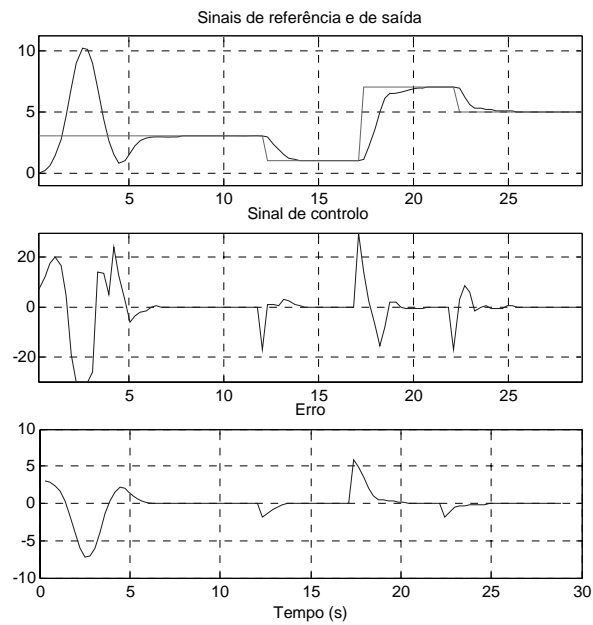


Figura 4.52: Sinais para o teste com o padrão de carga P2 (com TMF), sistema S2.

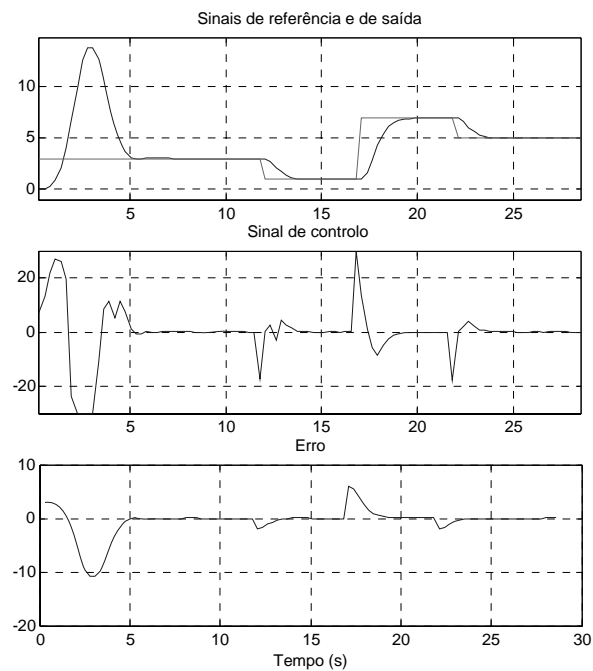


Figura 4.53: Sinais para o teste com o padrão de carga P3 (com TMF), sistema S2.

Verifica-se que o sinal de saída não apresenta oscilações em nenhum dos testes.

A tabela 4.23 apresenta o valor dos índices ISE e ITSE para os testes efectuados.

Teste	Referência	P1	P2	P3
ISE	31.9	33.9	37.0	43.3
ITSE	554	587	643	744

Tabela 4.23: Valor de ISE e de ITSE para os testes (com TMF), sistema S2.

Tal como para o sistema S1 o valor de ISE e de ITSE aumenta com o aumento do valor médio do atraso introduzido.

A tabela 4.24 apresenta a melhoria obtida através da comparação entre os valores de ISE e de ITSE para os controladores adaptativos com e sem TMF. A melhoria foi calculada de acordo com a expressão 4.5.

Teste	Referência	P1	P2	P3
M_{ISE}	-1%	-2%	-1%	+59%
M_{ITSE}	-1%	-2%	-1%	+60%

Tabela 4.24: Melhoria para o sistema S2.

Tal como no sistema S1 no teste de referência o valor de ISE e de ITSE é maior quando se faz a modelação do tempo morto fraccionário, ou seja o desempenho de controlo é pior, cerca de 1%. A razão deste comportamento é semelhante à do sistema S1, visto que o modelo utilizado para modelar o tempo morto fraccionário também possui um zero a mais, o que torna a sua resposta mais demorada.

Os benefícios da utilização do controlador distribuído tornam-se evidentes no teste com o padrão de carga P3 onde a melhoria é de 60%.

O modelo utilizado para a função de transferência discreta é dado pela equação 4.7:

$$G(q^{-1}) = \frac{b_1q^{-1} + b_2q^{-2} + b_3q^{-3}}{1 + a_1q^{-1} + a_2q^{-2}} \quad (4.7)$$

As figuras 4.54 a 4.57 apresentam a evolução dos parâmetros d modelo discreto do sistema ao longo dos ensaios realizados.

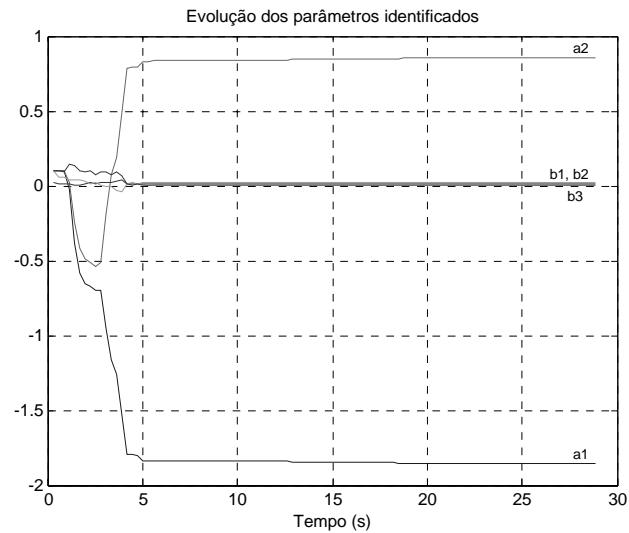


Figura 4.54: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S2.

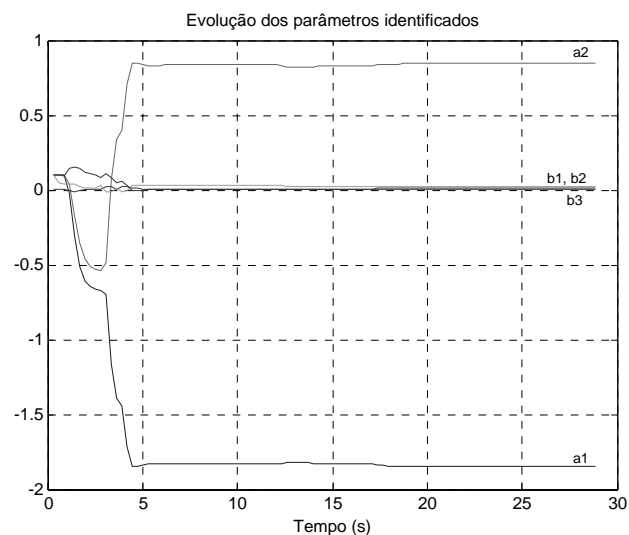


Figura 4.55: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S2.

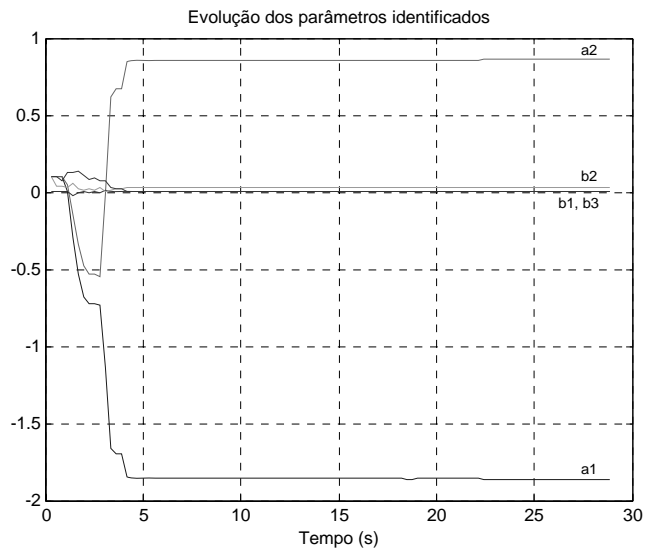


Figura 4.56: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S2.

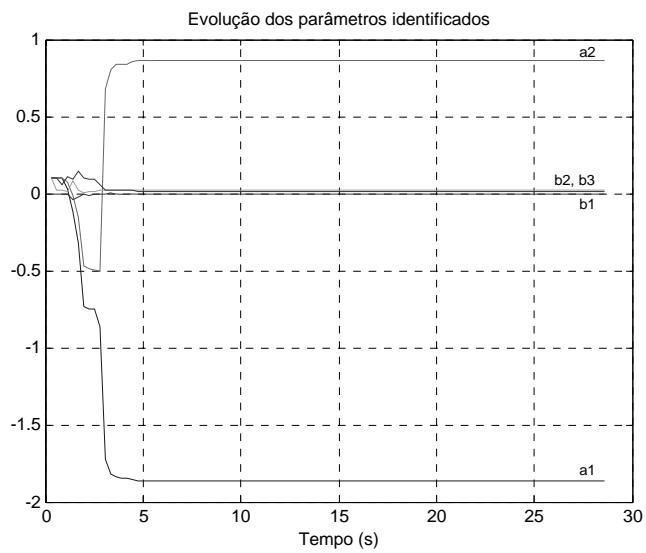


Figura 4.57: Evolução dos parâmetros do modelo discreto do sistema, para o teste com o padrão de carga P3 (com TMF), sistema S2.

A tabela 4.25 apresenta o valor dos parâmetros (módulo) da função discreta do sistema identificados no final de cada um dos testes anteriores.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.869	1.868	1.866	1.867	1.865
a_2	0.869	0.868	0.866	0.867	0.865
b_1	0.019	0.017	0.012	0.004	0.000
b_2	0.018	0.019	0.023	0.028	0.020
b_3	-	0.001	0.002	0.005	0.017

Tabela 4.25: Parâmetros identificados no final dos testes (com TMF), sistema S2.

A tabela 4.26 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S2.

Pólo	Teórico	Referência	P1	P2	P3
p_1	1	1	1	1	1
p_2	0.869	0.868	0.866	0.867	0.865

Tabela 4.26: Localização dos pólos identificados (com TMF), sistema S2.

A utilização do modelo que tem em conta o tempo morto fraccionário permite uma melhor identificação dos pólos do sistema para os testes em que o atraso introduzido é maior (testes com os padrões de carga P2 e P3).

4.3.3 Sistema S3

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.58.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.59, 4.60 e 4.61.

O sinal de saída não apresenta oscilações em nenhum dos testes.

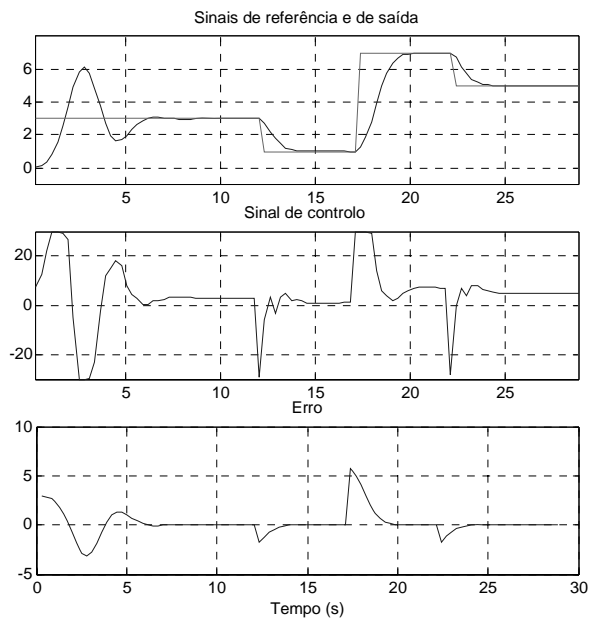


Figura 4.58: Sinais para o teste de referência (com TMF), sistema S3.

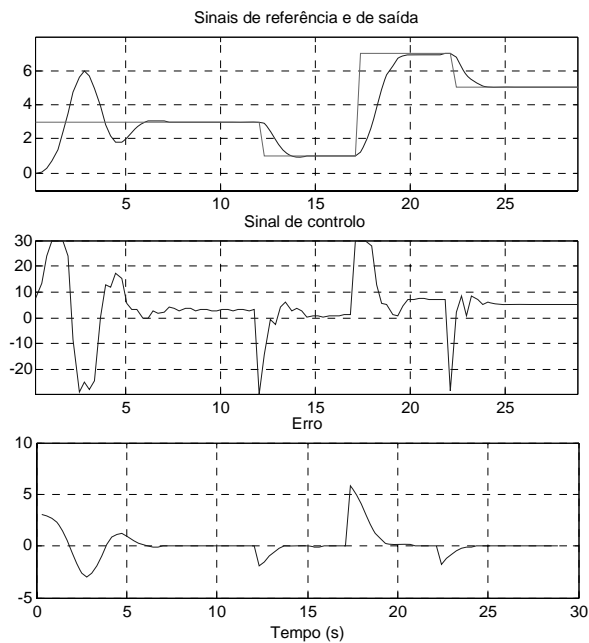


Figura 4.59: Sinais para o teste com o padrão de carga P1 (com TMF), sistema S3.

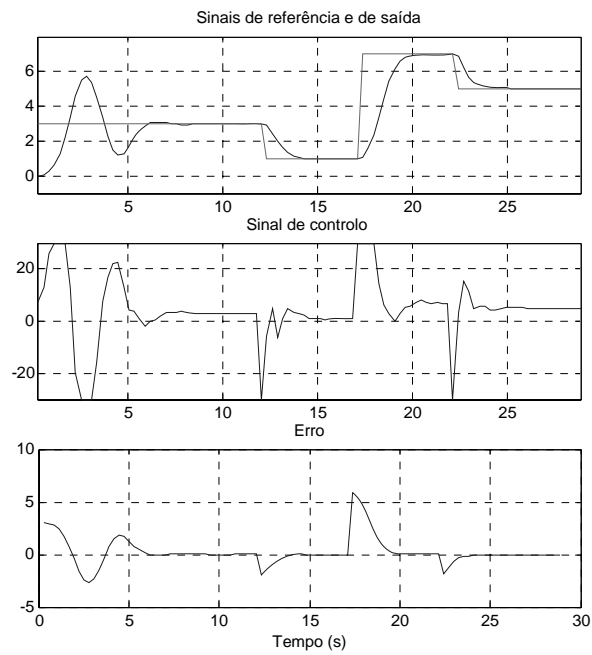


Figura 4.60: Sinais para o teste com o padrão de carga P2 (com TMF), sistema S3.

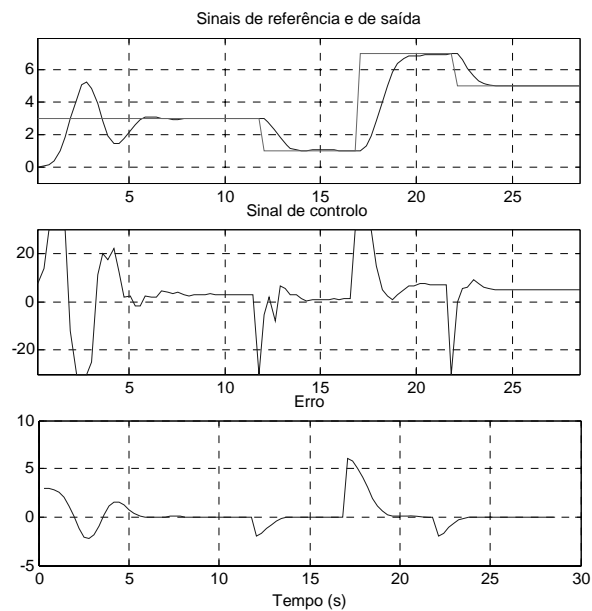


Figura 4.61: Sinais para o teste com o padrão de carga P3 (com TMF), sistema S3.

A tabela 4.27 apresenta o valor dos índices ISE e ITSE para os testes efectuados.

Teste	Referência	P1	P2	P3
ISE	41.2	42.4	46.7	52.8
ITSE	721	740	819	914

Tabela 4.27: Valor de ISE e de ITSE para os testes (com TMF), sistema S3.

O valor de ISE e de ITSE aumenta à medida que o valor médio do atraso aumenta.

A tabela 4.28 apresenta a melhoria obtida por comparação entre os valores de ISE e de ITSE para os controladores com e sem TMF. A melhoria foi calculada utilizando a expressão 4.5.

Teste	Referência	P1	P2	P3
M_{ISE}	0%	-1%	0%	+32%
M_{ITSE}	0%	-1%	0%	+33%

Tabela 4.28: Melhoria para o sistema S3.

Para este sistema o valor de ISE e de ITSE para o teste de referência e para o padrão de carga P2 é praticamente igual ao dos testes sem modelação do tempo morto fraccionário, embora, tal como para os sistemas anteriores, o modelo utilizado para modelar o tempo morto fraccionário possua um zero a mais, o que torna a sua resposta mais “lenta”.

No ensaio com os padrões de carga P1 e P2 a vantagem obtida com uma melhor modelação dos pólos do sistema não é suficiente para que se obtenha um melhor desempenho de controlo. Os benefícios da modelação do TMF só se tornam evidentes no teste com o padrão de carga P3 onde a melhoria é de 32%.

O modelo utilizado para a função de transferência discreta é dado pela equação 4.7.

As figuras 4.62 a 4.65 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados.

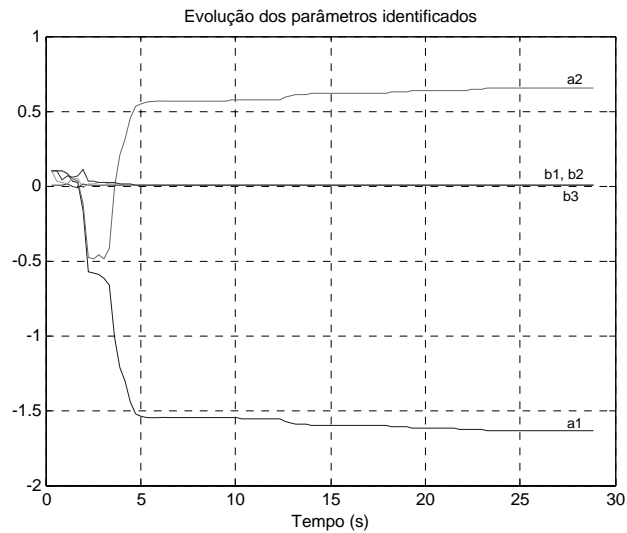


Figura 4.62: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S3.

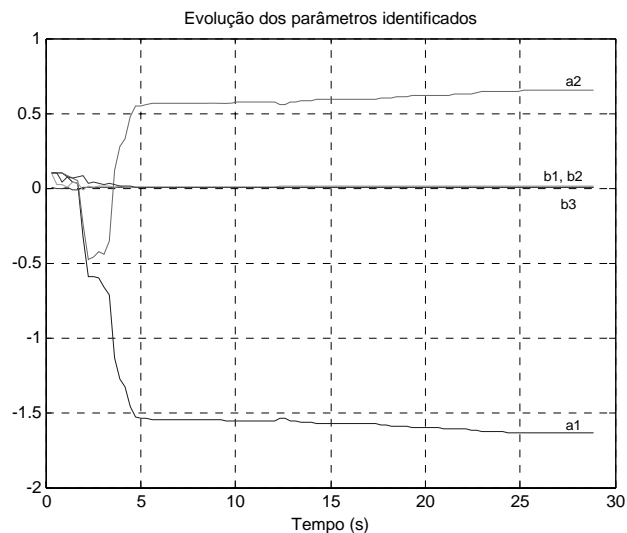


Figura 4.63: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S3.

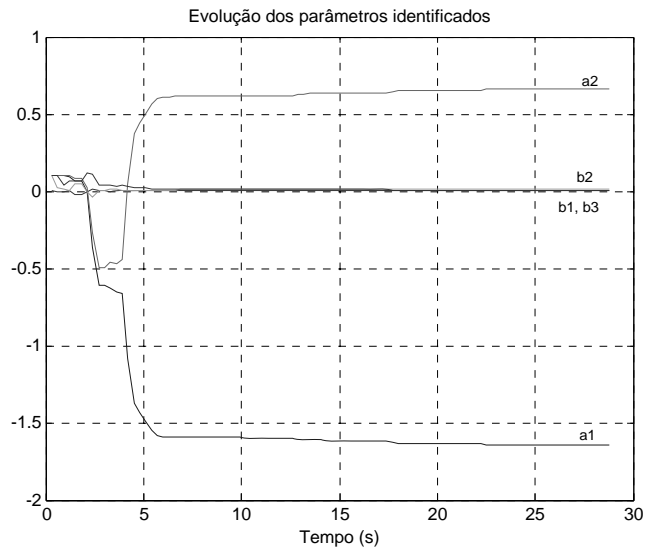


Figura 4.64: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S3.

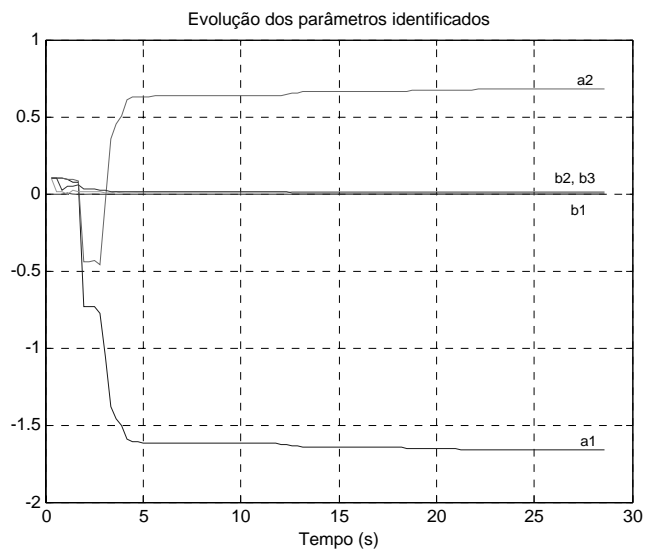


Figura 4.65: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S3.

A tabela 4.29 apresenta o valor dos parâmetros da função discreta do sistema (módulo) identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.688	1.692	1.696	1.698	1.699
a_2	0.705	0.707	0.710	0.712	0.714
b_1	0.009	0.008	0.006	0.002	0.000
b_2	0.008	0.009	0.010	0.012	0.009
b_3	-	0.001	0.002	0.003	0.008

Tabela 4.29: Parâmetros identificados no final dos testes (com TMF), sistema S3.

A tabela 4.30 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S3.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.940	0.939	0.943	0.943	0.937
p_2	0.758	0.753	0.753	0.755	0.762

Tabela 4.30: Localização dos pólos identificados (com TMF), sistema S3.

O modelo que tem em conta o tempo morto fraccionário permite uma melhor identificação dos pólos do sistema quando existem atrasos elevados na malha de controlo.

4.3.4 Sistema S4

Os resultados de controlo obtidos para o teste de referência (sem carga extra no barramento) são apresentados na figura 4.66.

Os resultados obtidos para os padrões de carga P1, P2 e P3 são apresentados nas figuras 4.67, 4.68 e 4.69.

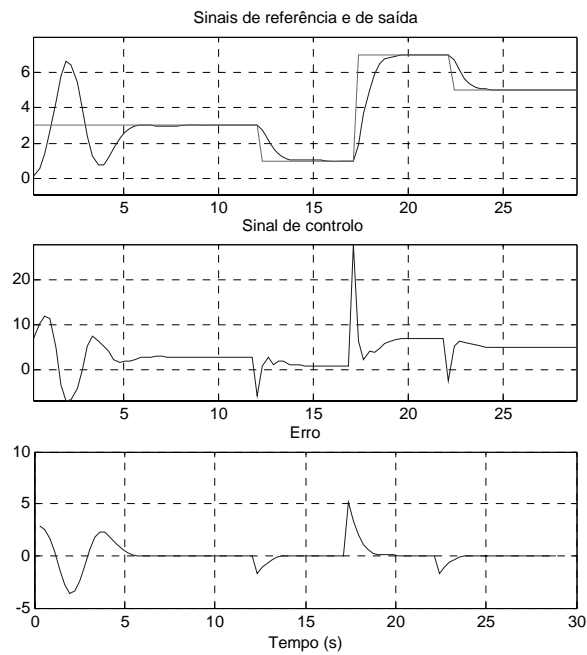


Figura 4.66: Sinais para o teste de referência (com TMF), sistema S4.

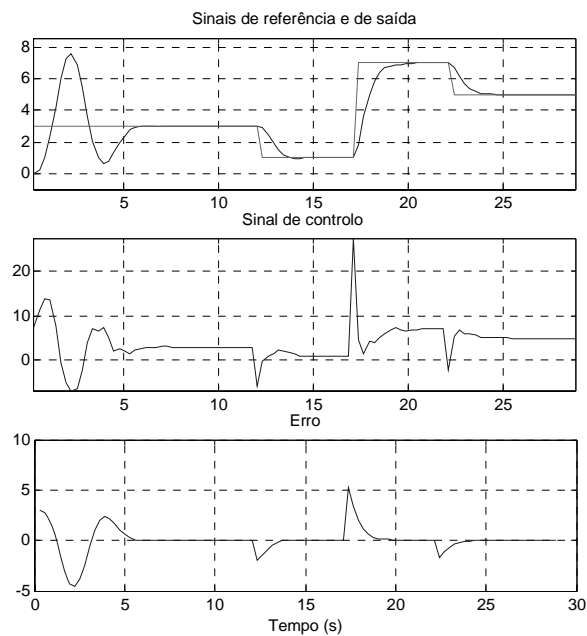


Figura 4.67: Sinais para o teste com o padrão de carga P1 (com TMF), sistema S4.

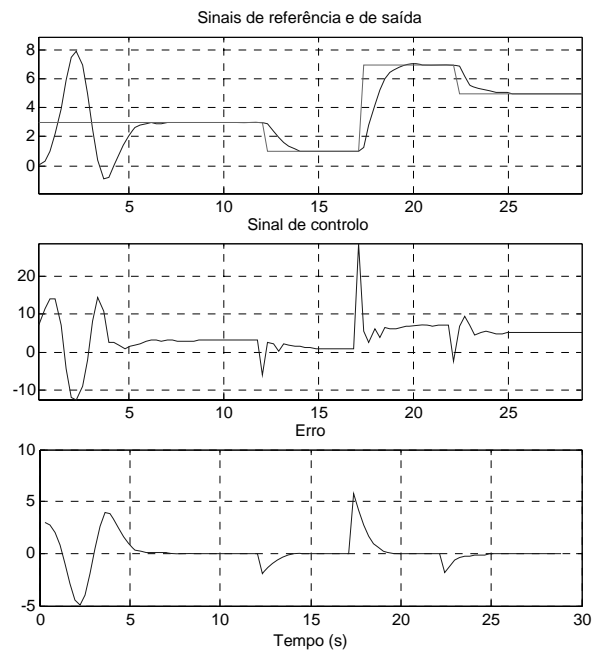


Figura 4.68: Sinais para o teste com o padrão de carga P2 (com TMF), sistema S4.

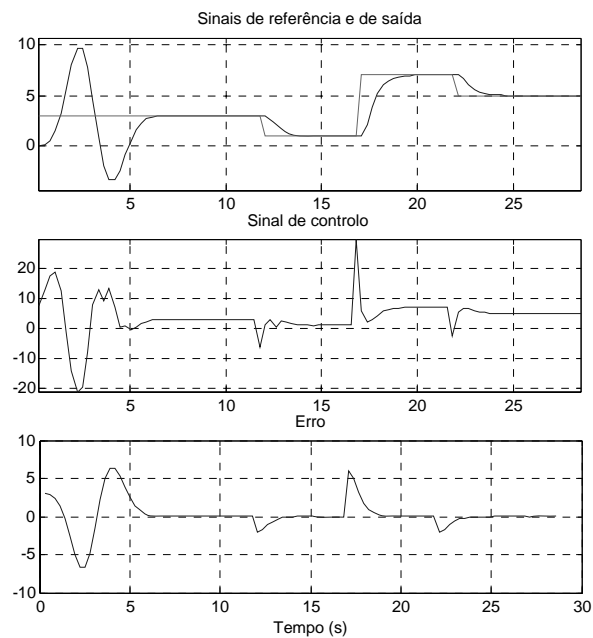


Figura 4.69: Sinais para o teste com o padrão de carga P3 (com TMF), sistema S4.

O sinal de saída não apresenta oscilações em nenhum dos testes.

A tabela 4.31 apresenta o valor dos índices ISE e ITSE obtido nos testes efectuados ao sistema S4.

Teste	Referência	P1	P2	P3
ISE	46.0	49.9	55.0	56.8
ITSE	468	485	578	650

Tabela 4.31: Valor de ISE e de ITSE para os testes (com TMF), sistema S4.

A tabela 4.32 apresenta a melhoria obtida por comparação entre os valores de ISE e de ITSE para os controladores com e sem TMF. A melhoria foi calculada de acordo com a expressão 4.5.

Teste	Referência	P1	P2	P3
M_{ISE}	-3%	-5%	-4%	+63%
M_{ITSE}	-3%	-5%	-4%	+62%

Tabela 4.32: Melhoria para o sistema S4.

Para este sistema o valor de ISE e de ITSE para os testes de referência e com os padrões de carga P1 e P2 é maior do que no caso em que não é feita a modelação do tempo morto fraccionário. Nestes casos o benefício de uma melhor identificação dos parâmetros do sistema não se sobrepõe à maior “lentidão” do controlador distribuído. Para o teste com o padrão de carga P3, e como esperado a melhoria é grande (cerca de 63%).

O modelo utilizado para a função de transferência discreta é dado pela equação 4.7.

As figuras 4.70 a 4.73 apresentam a evolução dos parâmetros do modelo discreto do sistema ao longo dos ensaios realizados.

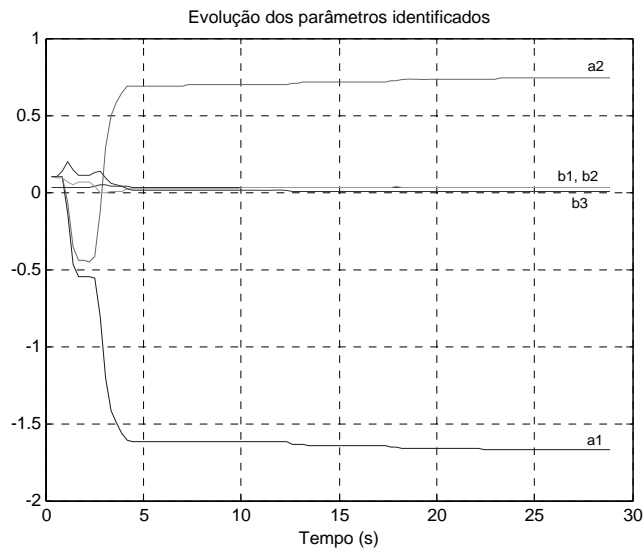


Figura 4.70: Evolução dos parâmetros do modelo discreto do sistema para o teste de referência (com TMF), sistema S4.

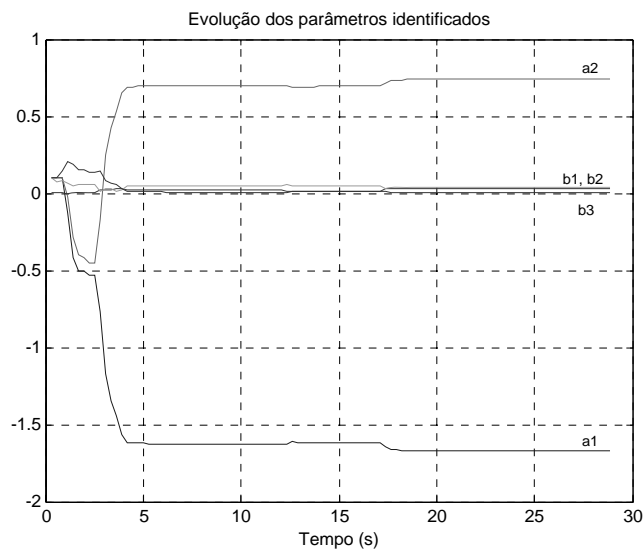


Figura 4.71: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P1 (com TMF), sistema S4.

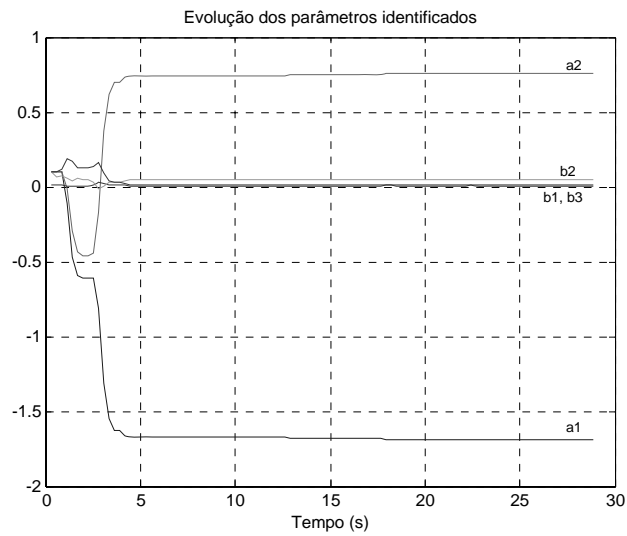


Figura 4.72: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P2 (com TMF), sistema S4.

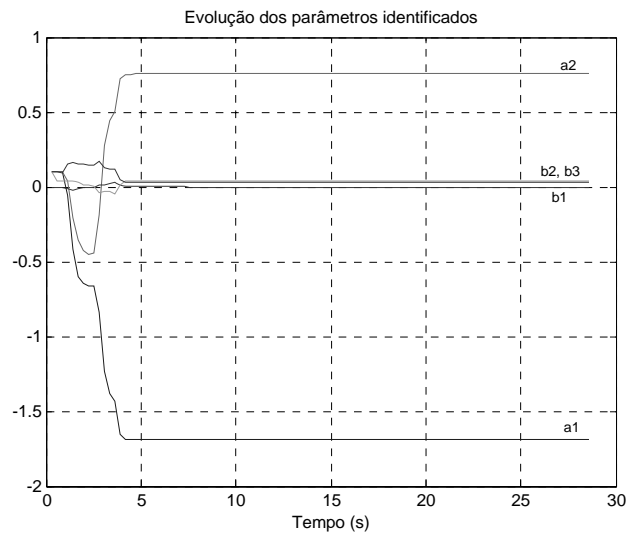


Figura 4.73: Evolução dos parâmetros do modelo discreto do sistema para o teste com o padrão de carga P3 (com TMF), sistema S4.

A tabela 4.33 apresenta o valor dos parâmetros do modelo discreto do sistema (módulo) identificados no final de cada um dos testes.

Parâmetro	Teórico	Referência	P1	P2	P3
a_1	1.688	1.669	1.673	1.688	1.691
a_2	0.756	0.738	0.743	0.757	0.758
b_1	0.036	0.033	0.027	0.009	0.000
b_2	0.032	0.035	0.041	0.050	0.037
b_3	-	0.002	0.002	0.010	0.030

Tabela 4.33: Parâmetros identificados no final dos testes (com TMF), sistema S4.

A tabela 4.34 apresenta a localização dos pólos (no plano discreto) correspondente aos parâmetros identificados no final de cada um dos testes anteriores para o sistema S4.

Pólo	Teórico	Referência	P1	P2	P3
p_1	0.844+j0.209	0.835+j0.204	0.837+j0.208	0.844+j0.211	0.846+j0.200
p_2	0.844-j0.209	0.835-j0.204	0.837-j0.208	0.844-j0.211	0.844-j0.211

Tabela 4.34: Localização dos pólos identificados (com TMF), sistema S4.

O modelo que tem em conta o tempo morto fraccionário permite uma melhor identificação dos pólos do sistema quando existem atrasos elevados na malha de controlo.

A utilização de modelos da função discreta do sistema que têm em conta o tempo morto fraccionário introduzido na malha de controlo originam controladores que, embora sejam mais lentos devido ao facto de se basearem numa função de controlo que precisa de informação mais antiga do sistema, permitem obter um melhor desempenho de controlo nos casos em que os controladores “clássicos” apresentam oscilações ou instabilidade. A utilização do modelo com tempo morto fraccionário, embora origine algoritmos para identificação e controlo mais complexos tem a vantagem de proporcionar uma boa identificação do sistema mesmo quando o atraso introduzido é elevado.

Nos ensaios aqui apresentados comparando o desempenho dos controladores distribuídos e “clássicos” verifica-se que a degradação máxima que se obtém no desempenho de controlo é de apenas 5%, correspondendo a situações em que o atraso introduzido na malha não é suficientemente grande para degradar a qualidade da estimativa do identificador clássico. Por outro lado nos casos em que o atraso introduzido piora muito o desempenho de controlo, o controlador distribuído é capaz de melhorar significativamente esse desempenho. A utilidade da modelação do atraso entre a amostragem e a actuação e a sua utilização no projecto de controladores adaptativos distribuídos por posicionamento de pólos fica assim demonstrada.

4.4 Técnica de adaptação dinâmica da taxa de amostragem

Como foi referido no capítulo 3 existe actualmente uma enorme pressão no sentido de haver uma maior integração nas implementações de sistemas distribuídos. Neste tipo de implementações o mesmo barramento de comunicação serve de suporte a vários subsistemas de tempo-real, ou sem ser de tempo-real, que podem exibir diferentes requisitos de comunicação durante a operação do sistema devido a alterações do modo de funcionamento ou a estímulos externos.

O projecto tradicional deste tipo de sistemas era feito considerando períodos constantes e *deadlines* fixos para a situação de pior caso seguindo uma abordagem muito conservadora que origina um grande desperdício de recursos e implementações dispendiosas. A alternativa a este método de projecto consiste em utilizar especificações mais realistas, considerando valores médios e permitindo a ocorrência de sobrecargas esporádicas do tráfego no barramento. Neste caso tornam-se necessários mecanismos que permitam gerir a ocorrência dessas situações esporádicas e temporárias.

Num sistema de controlo distribuído o desempenho de controlo está relacionado com a largura de banda consumida, visto que essa largura de banda depende directamente do período de amostragem considerado para a malha de controlo. Por esse motivo existe também a tendência para utilizar valores para o período de amostragem tão pequenos quanto possível (dentro da gama dada pelo critério utilizado).

Por outro lado a técnica de controlo adaptativo distribuído por posicionamento de pólos permite obter um desempenho de controlo “aceitável” no caso em que se utiliza um valor elevado para o período de amostragem (dentro da gama dada pelo critério utilizado) mesmo que a malha seja afectada por atrasos elevados, como mostram os resultados apresentados na secção anterior.

Para responder aos desafios resultantes das condições de implementação actuais é necessário admitir compromissos entre o desempenho de controlo e a largura de banda consumida.

Nesta dissertação propõe-se a técnica de adaptação dinâmica da taxa de amostragem através da qual o período de amostragem das malhas de controlo que partilham o mesmo barramento pode ser alterada, de modo a libertar largura de banda, para o envio de mensagens em situações de sobrecarga temporária, e capacidade de processamento. Essa alteração é efectuada através de um mecanismo de controlo da qualidade de serviço (QoS) do barramento que gere o desempenho global do sistema. Quando as condições de tráfego se normalizam a taxa de amostragem é de novo reposta de modo a proporcionar um melhor desempenho de controlo. A flexibilidade de operação proporcionada ao sistema pela aplicação desta técnica tem como contraponto a diminuição do desempenho de controlo durante o período em que a sua taxa de amostragem é alterada. Como referido anteriormente isso é inevitável porque o desempenho de controlo depende da taxa de amostragem.

De seguida descreve-se a implementação da técnica de adaptação dinâmica da taxa de amostragem utilizando o simulador TrueTime. Tal como descrito no capítulo 3 são proposta três abordagens distintas para a aplicação desta técnica: abordagem estática, abordagem dinâmica e abordagem híbrida.

4.4.1 Características da implementação com FTT-CAN

A implementação da técnica de adaptação dinâmica da taxa de amostragem implica a existência de um mecanismo de controlo da qualidade de serviço da rede de comunicação que garanta que as alterações a efectuar não colocam em causa os requisitos de tempo-real das aplicações que utilizam o barramento. O protocolo FTT-CAN surge como a escolha óbvia porque inclui um mecanismo de gestão de QoS, a par de um mecanismo de admissão de mensagens que só admite alterações nas características do conjunto de mensagens existentes se essas não puserem em causa a escalonabilidade das mensagens anteriormente admitidas no barramento.

Para os ensaios apresentados seguidamente utilizou-se a arquitectura base A2, apresentada na figura 4.2.

A taxa de transmissão do barramento mantém o valor utilizado nos testes anteriores, 125 kbps. A duração do *Elementary Cycle* (EC) nesta implementação do protocolo FTT-CAN é igual a 20 ms e a duração máxima da janela síncrona é igual a 40% do EC (para mais detalhes sobre o protocolo consultar a secção 2.3.4). O escalonamento das mensagens é efectuado através do algoritmo *rate monotonic*. A implementação do protocolo FTT-CAN não inclui a janela assíncrona.

Nos testes efectuados utilizaram-se os sistemas S1 e S2 cujas funções de transferência foram apresentadas no início deste capítulo (ver tabela 4.4).

As características temporais do tráfego, de acordo com o protocolo FTT-CAN, têm que ser expressas em ECs. Deste modo os valores definidos para o período de amostragem mínimo e máximo da malha de controlo têm que ser múltiplos do valor do EC. Escolheu-se por isso $h_{\min} = 0.12$ s e $h_{\max} = 0.28$ s. A malha de controlo opera inicialmente com o valor mínimo do período de modo a apresentar o melhor desempenho de controlo possível.

A quantidade de tráfego no barramento é definida através da tabela que contém as características das mensagens síncronas que utilizam o barramento. Em todos os ensaios foi utilizado o mesmo conjunto base de mensagens. As características dessas mensagens são apresentadas na tabela 4.35 em ECs.

O conjunto de mensagem base é constituído por doze mensagens de carga para além das duas mensagens da malha de controlo (M1 e M2). Este conjunto de mensagens resulta numa utilização da janela síncrona igual a 99.8%.

Nesta implementação a ordem para a amostragem do sistema é enviada pelo nó *master* ao nó do sensor através da *trigger message*, de acordo com o período de amostragem da malha de controlo. Este modo de implementação faz com que a amostragem

Mensagem	ID local	Período	DLC	Offset	Deadline
M3	5	1	3	0	1
M4	7	2	6	0	2
M5	9	2	5	1	2
M6	10	4	5	0	4
M7	13	5	4	0	5
M8	14	8	7	0	8
M9	15	1	4	0	1
M10	16	8	1	0	8
M11	20	6	2	0	6
M12	25	6	2	0	6
M1	30	6	2	1	6
M2	35	6	2	2	6
M13	40	1	8	0	1
M14	43	2	6	0	2

Tabela 4.35: Características do conjunto de mensagens síncronas de base (período, *offset* e *deadline* em EC).

do sistema não seja afectada por nenhum tipo de incerteza temporal. A mensagem M1 é escalonada com *offset* de um EC para que o seu envio ocorra no EC seguinte ao da activação da tarefa de amostragem. A mensagem M1 contém a amostra da saída do sistema e é enviada do nó do sensor para o nó do controlador. A mensagem M2 contém o valor de controlo que deve ser aplicado ao sistema e é enviada do nó do controlador para o nó do actuador com um deslocamento de 2 ECs, ou seja 40 ms, em relação ao instante de amostragem, como explicado na secção 4.1.1.

De modo a não sobrecarregar o barramento com mais uma mensagem, a detecção da variação de h é feita localmente através da medição do intervalo de tempo que medeia entre cada TM que faz o *dispatch* da tarefa responsável pela amostragem do sistema.

Para cada uma das abordagens propostas efectuaram-se dois testes em que o valor do período de amostragem varia de acordo com as solicitações para admissão de novas mensagens no barramento. As características das mensagens que pedem a admissão ao sistema durante os ensaios são apresentadas na tabela 4.36.

Mensagem	ID local	Período	DLC	Offset	Deadline
M15	18	18	5	0	18
M16	46	16	6	0	16

Tabela 4.36: Mensagens admitidas em *runtime* (período, *offset* e *deadline* em EC).

Foram efectuados dois ensaios distintos designados por E1 e E2.

No ensaio E1 o *master* recebe um pedido para incluir a mensagem M15 para $t = 15$ s. Este pedido não pode ser aceite porque a utilização para o conjunto de mensagens total

excede a duração da janela síncrona. Face a este problema o gestor de QoS selecciona o valor de $h = 0.16$ s para as mensagens da malha de controlo. Esta alteração faz com que o conjunto total de mensagens seja de novo escalonável com uma utilização da janela síncrona igual a 99.8% e a mensagem M15 é admitida no sistema. Para $t = 25$ s, a mensagem M16 pede a sua admissão ao sistema. Uma vez mais o pedido não pode ser aceite e o gestor de QoS propõe a alteração do valor do período das mensagens de controlo para $h = 0.28$ s de modo a tornar o conjunto escalonável. A utilização da janela síncrona mantém-se nos 99.8%. No instante $t = 35$ s as mensagens M15 e M16 são retiradas do sistema e o valor inicial de h é repostado. Este padrão repete-se a partir do instante de tempo $t = 45$ s.

No ensaio E2 o *master* recebe em simultâneo o pedido de admissão das duas mensagens M15 e M16 e, para que o pedido seja aceite, o período das mensagens da malha de controlo é alterado directamente para $h = 0.28$ s. No instante $t = 35$ s as mensagens são retiradas do sistema e o valor inicial de h é repostado. Tal como no ensaio anterior, o padrão repete-se a partir de $t = 45$ s.

Foram também efectuados testes com h fixo no valor máximo e mínimo de modo a poder fazer-se uma análise comparativa do desempenho das várias soluções.

O valor de ISE e de ITSE foi calculado entre $t = 40$ s e $t = 78$ s, na metade final dos ensaios.

4.4.2 Abordagem estática

Nesta abordagem utilizam-se controladores adaptativos distribuídos por posicionamento de pólos cujos parâmetros são calculados *off-line*. Define-se um conjunto de valores possíveis para o período de amostragem da malha de controlo e projecta-se um controlador para cada valor possível do período de amostragem. Obtém-se, deste modo, um banco de controladores com parâmetros fixos. Durante a operação do sistema esses controladores são comutados entre si de acordo com o valor de h seleccionado a cada momento pelo *master*.

Esta abordagem destina-se a sistemas com pouca capacidade de processamento e em que seja possível saber antecipadamente as condições de operação do sistema, tanto em termos do tráfego como do comportamento do sistema a controlar.

Esta abordagem não é adequada a sistemas cujas características sejam variáveis ao longo do tempo ou a implementações que provoquem grandes alterações do atraso introduzido na malha durante a operação do sistema.

Na implementação apresentada os valores utilizados para o período de amostragem são iguais para os sistemas S1 e S2. O conjunto de valores possíveis para o período de amostragem é dado por

$$h \in \{0.12, 0.16, 0.20, 0.24, 0.28\} \quad (\text{s})$$

Sistema S1

Os parâmetros do modelo discreto do sistema, utilizados para o cálculo dos parâmetros dos controladores, são apresentados na tabela 4.37, para cada um dos valores possíveis de h . Estes valores foram obtidos em procedimentos de identificação individuais realizados para cada um dos valores de h .

h (s)	a_1	b_1	b_2
0.12	0.942	0.038	0.020
0.16	0.923	0.058	0.020
0.20	0.905	0.076	0.019
0.24	0.887	0.095	0.019
0.28	0.869	0.112	0.018

Tabela 4.37: Valor dos parâmetros do modelo discreto, abordagem estática, sistema S1.

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.74 e 4.75.

Os gráficos dos sinais de referência, de saída e de controlo não apresentam unidades no eixo vertical por mostrarem dados obtidos por simulação, não tendo por isso correspondência directa com nenhuma grandeza física.

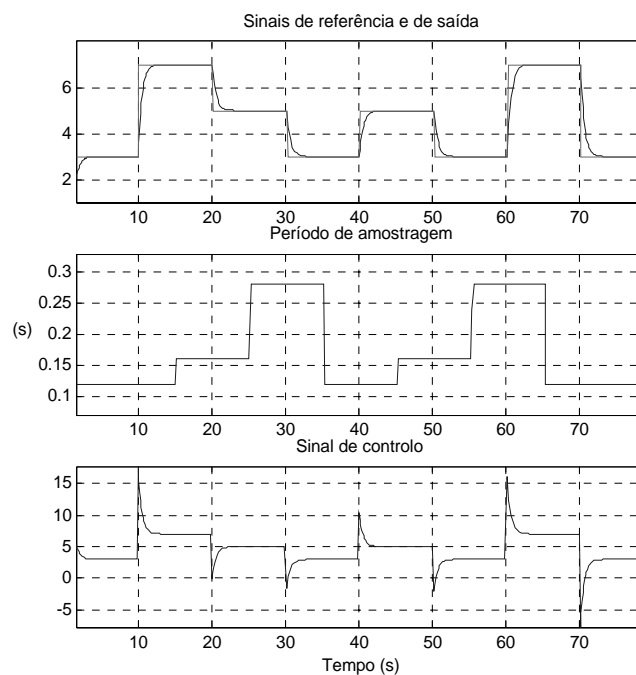


Figura 4.74: Sinais para o ensaio E1, abordagem estática, sistema S1.

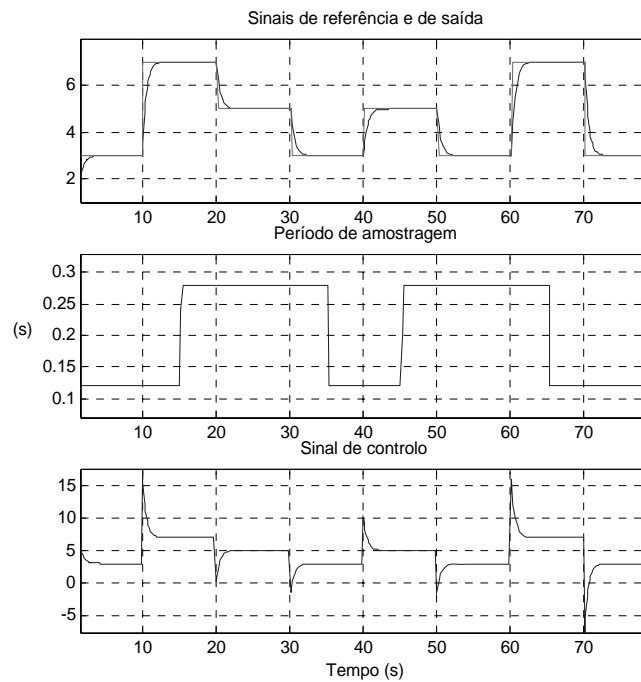


Figura 4.75: Sinais para o ensaio E2, abordagem estática, sistema S1.

A tabela 4.38 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Teste	$h = 0.12$ s	E1	E2	$h = 0.28$ s
ISE	13.8	15.7	16.0	17.8
ITSE	854	948	962	1070

Tabela 4.38: Valor de ISE e de ITSE para os ensaios da abordagem estática, sistema S1.

Para efectuar a comparação entre o desempenho de controlo recorreu-se uma vez mais ao cálculo da melhoria comparando o valor obtido para ISE (ou ITSE) nos diferentes ensaios utilizando a expressão 4.8:

$$M_I = \left(1 - \frac{I_E}{I_{h=fixo}}\right) * 100 \quad (4.8)$$

Nos comentários seguintes fala-se de degradação sempre que a melhoria obtida através de equação 4.8 apresenta um valor negativo.

Os valores obtidos para a melhoria comparando os índices ISE e ITSE são muito semelhantes.

O ensaio E1 apresenta uma melhoria do desempenho de controlo de 11% em relação ao ensaio com $h = 0.28$ s e uma degradação de 12% em relação ao ensaio com $h = 0.12$ s. O ensaio E2 apresenta uma melhoria do desempenho de controlo de 9% em relação ao ensaio com $h = 0.28$ s e uma degradação de 14% em relação ao ensaio com $h = 0.12$ s.

Os resultados mostram que a adaptação da taxa de amostragem durante a operação do sistema origina melhor desempenho de controlo quando comparada com a solução de manter o valor de h constante e igual a 0.28 s, que em termos de largura de banda teria efeitos semelhantes para a gestão do sistema. Essa melhoria é para este sistema de cerca de 10%. Por outro lado por comparação com o caso em que h é constante e igual a 0.12 s (melhor caso em termos do desempenho de controlo) a degradação que se verifica é de cerca de 13%. É este o preço a pagar ao nível da malha de controlo, para este sistema e para estas condições de operação, pela flexibilidade de operação do sistema global.

Para finalizar a análise desta abordagem efectuaram-se ainda ensaios em que o valor de h utilizado é diferente daquele para o qual o controlador foi projectado, no sentido de aferir se vale ou não a pena construir um banco de controladores para cada valor de h e proceder à sua comutação durante a operação do sistema.

Fizeram-se ensaios para o controlador projectado para $h = 0.12$ s com o valor de h igual a $\frac{h_{\min}+h_{\max}}{2}$ e a h_{\max} e para o controlador projectado para $h = 0.28$ s com o valor de h igual a h_{\min} e $\frac{h_{\min}+h_{\max}}{2}$, em que $h_{\min} = 0.12$ s e $h_{\max} = 0.28$ s. O valor de ISE obtido para estes ensaios é apresentado na tabela 4.39.

h (s)	0.12	0.20	0.28
Controlador projectado para $h = 0.12$ s	-	19.4	25.1
Controlador projectado para $h = 0.28$ s	12.0	14.5	-

Tabela 4.39: Valor de ISE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.

O valor de ITSE obtido para estes ensaios é apresentado na tabela 4.40.

h (s)	0.12	0.20	0.28
Controlador projectado para $h = 0.12$ s	-	1202	1562
Controlador projectado para $h = 0.28$ s	737	894	-

Tabela 4.40: Valor de ITSE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.

Verifica-se que para este sistema a utilização do controlador com parâmetros identificados para $h = 0.12$ s com valores de h elevados origina um desempenho de controlo pior do que aquele que se obtém quando se utiliza o controlador certo.

Para o caso do controlador cujos parâmetros foram identificados para $h = 0.28$ s, verifica-se que o valor de ISE (ou ITSE) obtido é menor do que o que se obtém com a utilização do controlador correcto. No entanto, quando $h = 0.12$ s, embora o desempenho de controlo seja melhor o sinal de saída apresenta *overshoot*, como se pode ver na figura 4.76.

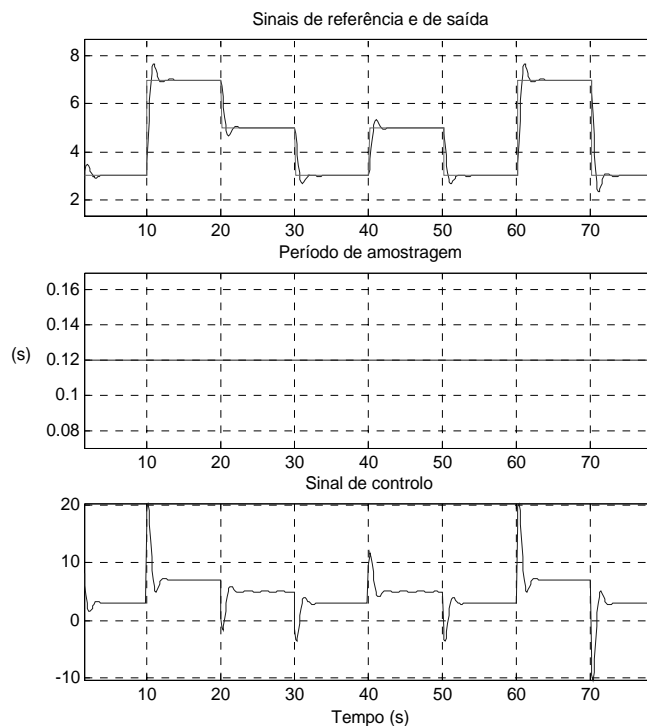


Figura 4.76: Sinais para o ensaio com $h = 0.12$ s utilizando os parâmetros identificados para $h = 0.28$ s, abordagem estática, sistema S1.

Verifica-se assim que, para este sistema e para estas condições de carga, pode também ser considerada a opção de utilizar um único controlador projectado para $h = 0.28$ s com comutação apenas do valor de h durante a operação do sistema.

Sistema S2

Os parâmetros do modelo discreto do sistema utilizados para o cálculo dos parâmetros dos controladores são apresentados na tabela 4.41, para cada um dos valores possíveis de h . Este valores foram obtidos em ensaios de identificação individuais realizados para cada um dos valores de h .

h (s)	a_1	a_1	b_1	b_2	b_3
0.12	-1.909	0.909	0.001	0.005	0.010
0.16	-1.905	0.905	0.003	0.008	0.001
0.20	-1.888	0.888	0.006	0.013	0.001
0.24	-1.871	0.871	0.009	0.017	0.001
0.28	-1.855	0.885	0.013	0.023	0.001

Tabela 4.41: Valor dos parâmetros do modelo discreto, abordagem estática, sistema S2.

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.77 e 4.78.

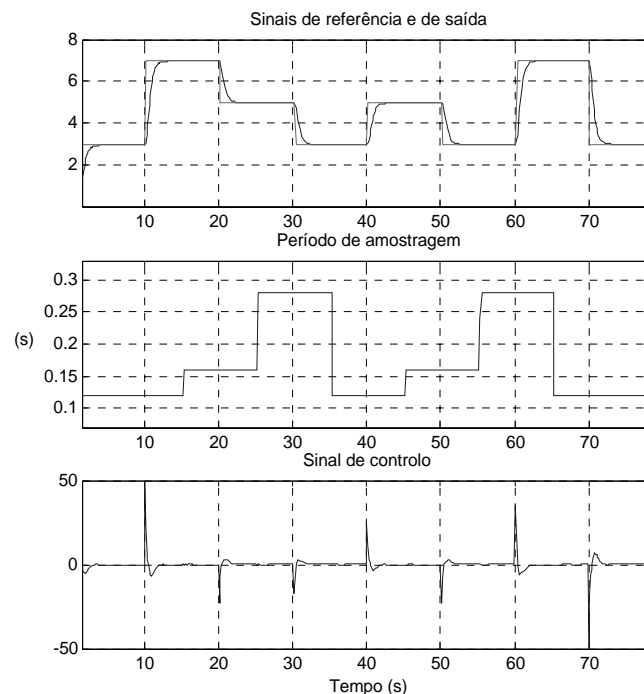


Figura 4.77: Sinais para o ensaio E1, abordagem estática, sistema S2.

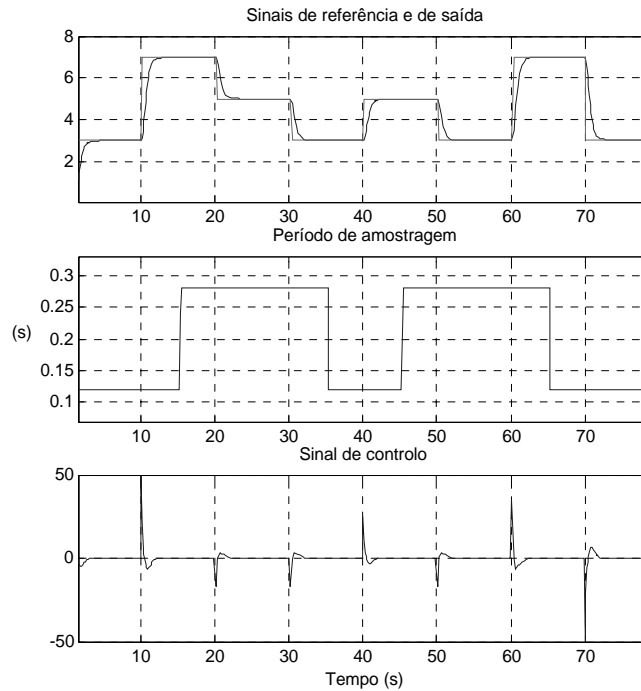


Figura 4.78: Sinais para o ensaio E2, abordagem estática, sistema S2.

A tabela 4.42 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Teste	$h = 0.12$ s	E1	E2	$h = 0.28$ s
ISE	21.3	23.0	23.3	25.0
ITSE	1312	1422	1429	1546

Tabela 4.42: Valor de ISE e de ITSE para os ensaios da abordagem estática, sistema S2.

A melhoria (ou degradação) para este conjunto de ensaios foi calculado através da equação 4.8, do modo referido anteriormente.

O ensaio E1 apresenta uma melhoria do desempenho de controlo de 8% em relação ao ensaio com $h = 0.28$ s e uma degradação de 8% em relação ao ensaio com $h = 0.12$ s. O ensaio E2 apresenta uma melhoria do desempenho de controlo de 7% em relação ao ensaio com $h = 0.28$ s e uma degradação de 9% em relação ao ensaio com $h = 0.12$ s.

Os resultados mostram que a adaptação da taxa de amostragem durante a operação do sistema origina melhor desempenho de controlo quando comparada com a solução de manter o valor de h constante e igual a 0.28 s. Essa melhoria é para este sistema de cerca de 8%. Por outro lado por comparação com o caso em que h é constante e

igual a 0.12 s (melhor caso em termos do desempenho de controlo) a degradação que se verifica é de 9%, para este sistema e para estas condições de operação.

Para finalizar a análise desta abordagem efectuaram-se ainda ensaios em que o valor de h utilizado é diferente daquele para o qual o controlador foi projectado, no sentido de aferir se vale ou não a pena construir um banco de controladores para cada valor de h e proceder à sua comutação durante a operação do sistema.

Fizeram-se ensaios para o controlador projectado para $h = 0.12$ s com o valor de h igual a $\frac{h_{\min}+h_{\max}}{2}$ e a h_{\max} e para o controlador projectado para $h = 0.28$ s com o valor de h igual a h_{\min} e $\frac{h_{\min}+h_{\max}}{2}$, em que $h_{\min} = 0.12$ s e $h_{\max} = 0.28$ s. O valor de ISE obtido para estes ensaios é apresentado na tabela 4.43.

h (s)	0.12	0.20	0.28
Controlador projectado para $h = 0.12$ s	-	39.2	592
Controlador projectado para $h = 0.28$ s	506	24.4	-

Tabela 4.43: Valor de ISE para os ensaios com h diferente do valor de projecto do controlador, sistema S2.

O valor de ITSE obtido para estes ensaios é apresentado na tabela 4.44.

h (s)	0.12	0.20	0.28
Controlador projectado para $h = 0.12$ s	-	2332	34229
Controlador projectado para $h = 0.28$ s	30248	1507	-

Tabela 4.44: Valor de ITSE para os ensaios com h diferente do valor de projecto do controlador, sistema S1.

Verifica-se que, para este sistema, a utilização de controladores com parâmetros identificados para $h = 0.12$ s ou para $h = 0.28$ s para valores de h distintos originam pior desempenho de controlo do que no caso em que se aplica a comutação dos controladores. Os resultados apresentam fortes oscilações ou *overshoot* significativo, dependendo dos casos.

4.4.3 Abordagem dinâmica

A abordagem dinâmica propõe a utilização de um único controlador adaptativo distribuído por posicionamento de pólos que utiliza, como é habitual, um único identificador. Ao longo da operação do sistema é permitida a alteração do valor de h e o identificador actua em tempo-real actualizando o valor dos parâmetros do modelo discreto do sistema de acordo com as alterações verificadas.

Esta abordagem destina-se a sistemas com alguma capacidade de processamento. Ela apresenta a grande vantagem de não requerer muito conhecimento prévio das condições de operação do sistema, uma vez que tem a capacidade de se adaptar às alterações introduzidas.

Esta abordagem proporciona tempos de comutação muito superiores à abordagem estática, quando se altera o valor de h até que o identificador possa convergir de novo.

Na implementação apresentada a gama de valores utilizados para o período de amostragem é igual para os sistemas S1 e S2. A gama de valores possíveis para o período de amostragem é dada por

$$h \in [0.12 \ 0.28] \text{ (s)}$$

O estimador utiliza o algoritmo RLS com esquecimento direccionado. O valor do factor de esquecimento é igual a 0.99.

Sistema S1

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.79 e 4.80.

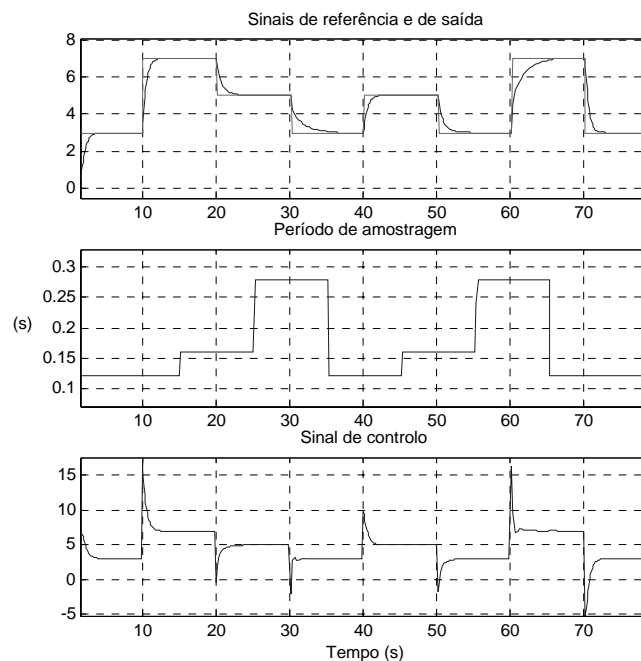


Figura 4.79: Sinais para o ensaio E1, abordagem dinâmica, sistema S1.

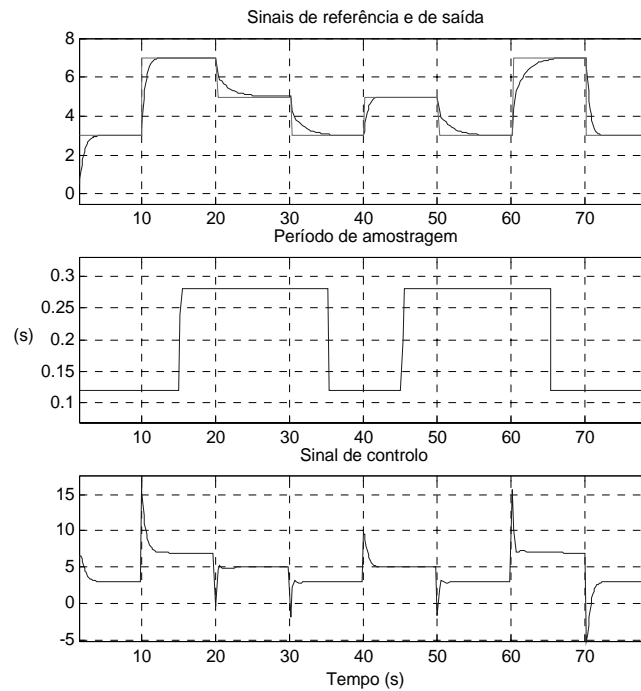


Figura 4.80: Sinais para o ensaio E2, abordagem dinâmica, sistema S1.

Os gráficos com a evolução dos parâmetros do modelo discreto do sistema são apresentados nas figuras 4.81 e 4.82.

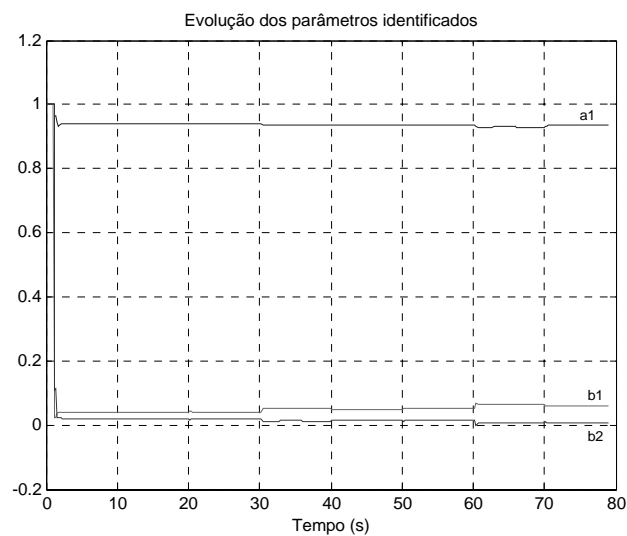


Figura 4.81: Evolução dos parâmetros identificados durante o ensaio E1, abordagem dinâmica, sistema S1.

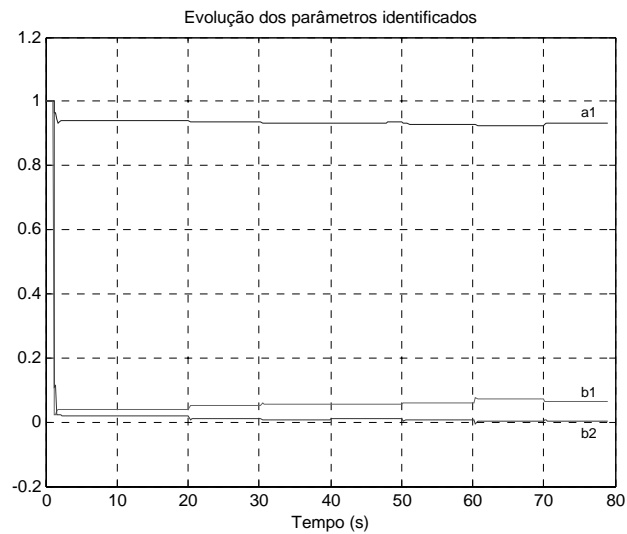


Figura 4.82: Evolução dos parâmetros identificados durante o ensaio E2, abordagem dinâmica, sistema S1.

Foram também realizados ensaios com h fixo igual a 0.12 s e 0.28 s para servirem de ponto de comparação.

A tabela 4.45 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Teste	$h = 0.12$ s	E1	E2	$h = 0.28$ s
ISE	13.9	17.9	18.7	17.3
ITSE	854	1102	1142	1070

Tabela 4.45: Valor de ISE e de ITSE para os ensaios da abordagem dinâmica, sistema S1.

Para esta abordagem o valor obtido para ISE (e para ITSE) é pior para ambos os ensaios E1 e E2 (em relação ao ensaio com h igual a 0.28 s), no entanto o sinal de saída não apresenta oscilações.

Nestes ensaios a degradação em relação ao melhor caso é de cerca de 30% e em relação ao pior caso é de 5%. Os valores da melhoria/ degradação obtidos para ISE e ITSE são semelhantes.

Este resultado era esperado pois de cada vez que há alteração do valor de h o identificador demora algum tempo até convergir para o novo valor dos parâmetros da função discreta do sistema. Em sistemas em que a alteração do valor de h seja lenta, em relação ao tempo que o identificador demora a convergir, essa desvantagem pode ser compensada com a melhor qualidade de controlo que se obtém para valores de h mais baixos.

Sistema S2

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.83 e 4.84.

Os gráficos com a evolução dos parâmetros do modelo discreto do sistema são apresentados nas figuras 4.85 e 4.86.

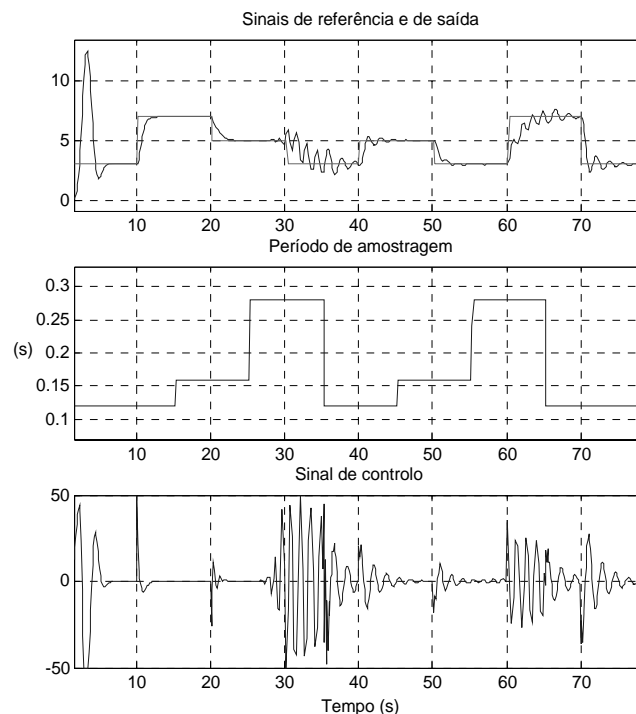


Figura 4.83: Sinais para o ensaio E1, abordagem dinâmica, sistema S2.

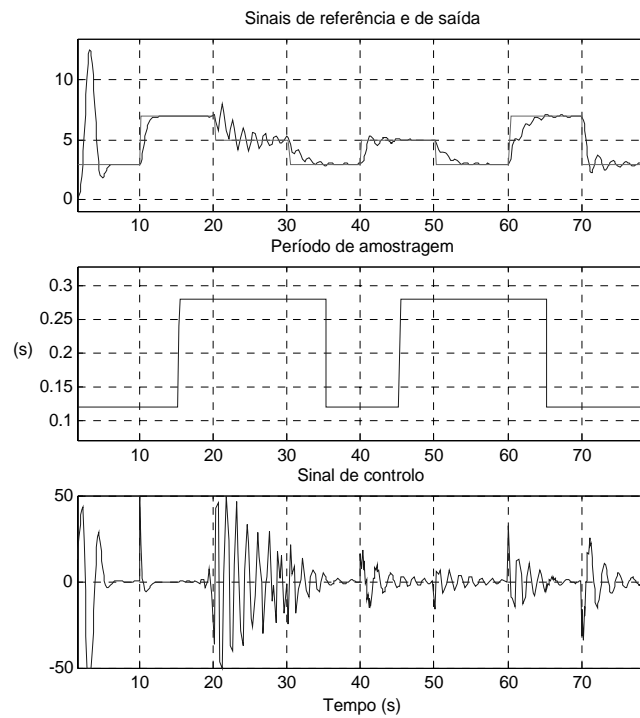


Figura 4.84: Sinais para o ensaio E2, abordagem dinâmica, sistema S2.

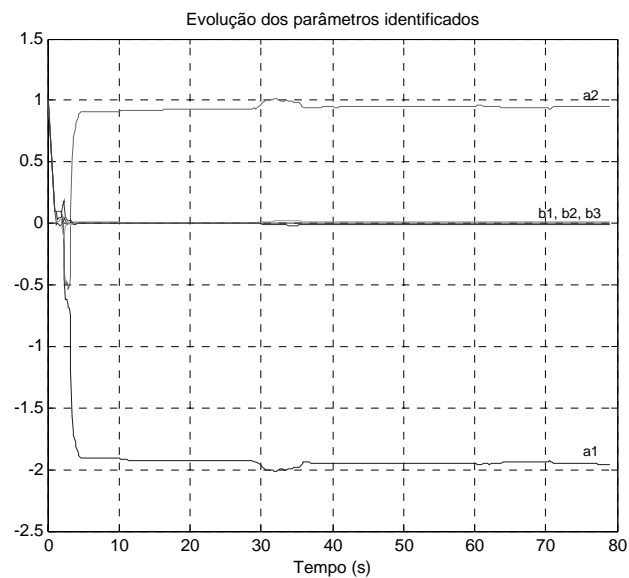


Figura 4.85: Evolução dos parâmetros identificados durante o ensaio E1, abordagem dinâmica, sistema S2.

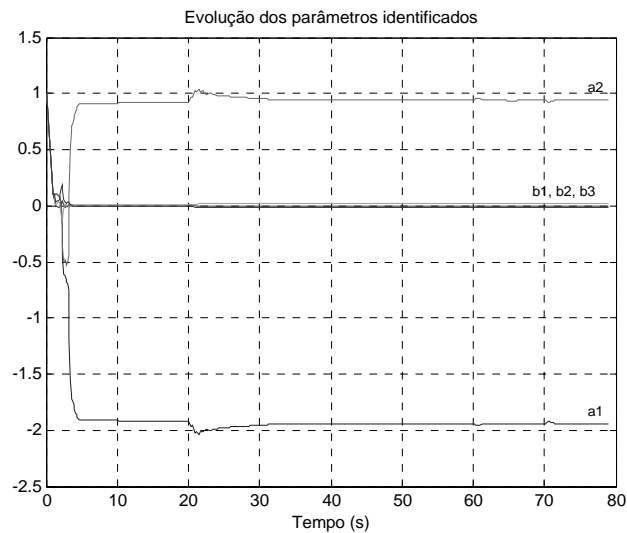


Figura 4.86: Evolução dos parâmetros identificados durante o ensaio E2, abordagem dinâmica, sistema S2.

Foram também realizados ensaios com h fixo igual a 0.12 s e 0.28 s para servirem de ponto de comparação.

A tabela 4.46 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Teste	$h = 0.12s$	E1	E2	$h = 0.28 s$
ISE	21.4	29.3	29.6	24.9
ITSE	1318	1804	1806	1540

Tabela 4.46: Valor de ISE e de ITSE para os ensaios da abordagem dinâmica, sistema S2.

Para esta abordagem o valor obtido para ISE (e ITSE) é pior para ambos os ensaios E1 e E2 (em relação ao ensaio com h igual a 0.28 s) e o sinal de saída apresenta oscilações.

Nestes ensaios a degradação em relação ao melhor caso é de 37% e em relação ao pior caso é de cerca de 18%. Os valores obtidos para a melhoria/ degradação são semelhantes utilizando os índices ISE ou ITSE.

Para este sistema verifica-se que o estimador precisa de mais tempo para convergir.

4.4.4 Abordagem híbrida

A abordagem híbrida pretende obviar os problemas da abordagem dinâmica, ou seja, melhorar o tempo de convergência dos estimadores quando há alteração do valor de h . Para esta abordagem propõe-se a utilização de um controlador adaptativo distribuído por posicionamento de pólos e de um banco de estimadores, um para cada valor de h . Para cada valor de h é mantida a informação relativa ao valor dos parâmetros θ e da matriz de covariância \mathbf{P} . Quando é feita a comutação entre diferentes valores de h o estimador é comutado passando a utilizar um conjunto de parâmetros θ e uma matriz de covariância \mathbf{P} específicos para cada valor de h . Deste modo o tempo de convergência do estimador é diminuído porque os cálculos têm por base valores de θ e de \mathbf{P} que estão de acordo com o valor de h .

Como foi explicado no capítulo 3 quando é feita a comutação do estimador o vector de regressores φ corresponde ao valor de h anterior e por isso a identificação só se inicia algumas amostras mais tarde quando já há informação actualizada para o vector de regressores. A espera é de duas amostras ou de três amostras consoante os sistemas sejam de 1ª ou de 2ª ordem, respectivamente.

A abordagem híbrida requer a mesma capacidade de processamento do que a abordagem dinâmica e maior quantidade de memória para armazenamento da informação relativa aos vários estimadores. Esta abordagem mantém a vantagem de se adaptar às alterações que o sistema venha a apresentar durante o seu funcionamento.

Sistema S1

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.87 e 4.88.

Os gráficos com a evolução dos parâmetros do modelo discreto do sistema são apresentados nas figuras 4.89 e 4.90.

São também apresentados os resultados para os ensaios com h fixo igual a 0.12 s e 0.28 s para servirem de ponto de comparação.

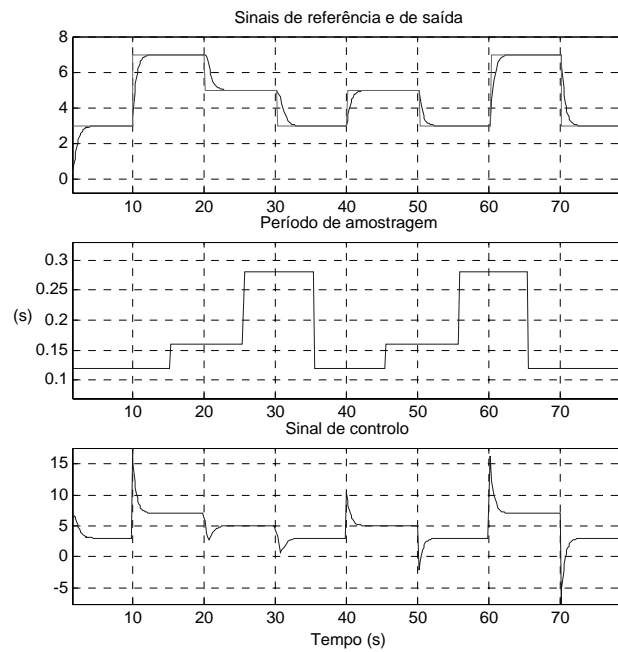


Figura 4.87: Sinais para o ensaio E1, abordagem híbrida, sistema S1.

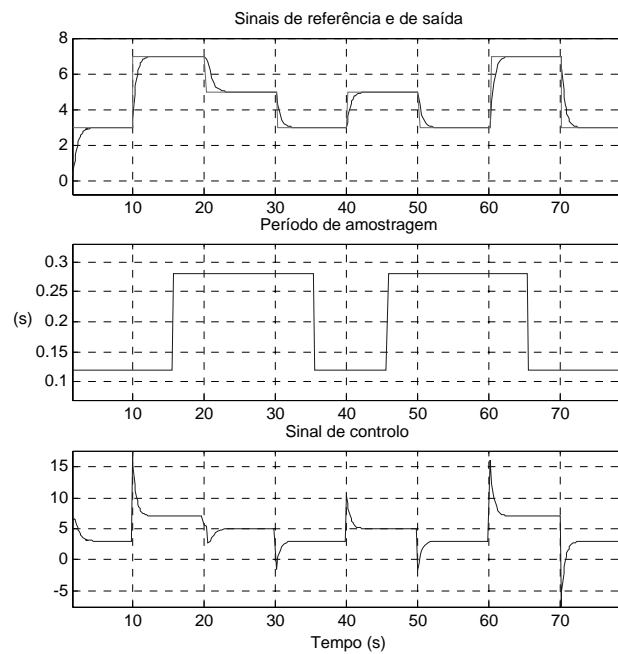


Figura 4.88: Sinais para o ensaio E2, abordagem híbrida, sistema S1.

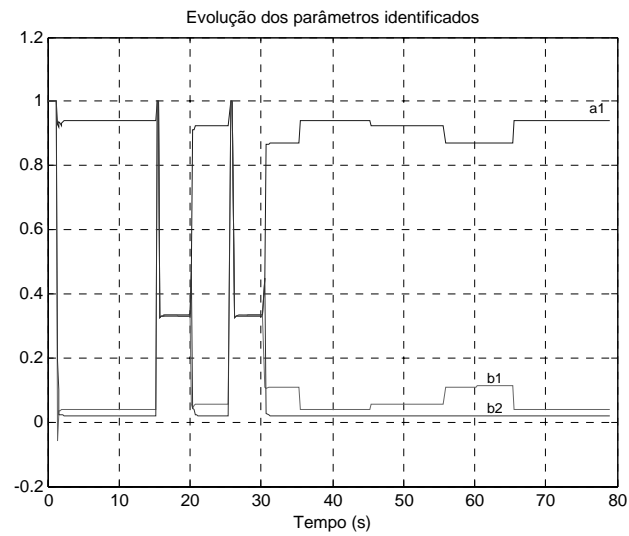


Figura 4.89: Evolução dos parâmetros identificados durante o ensaio E1, abordagem híbrida, sistema S1.

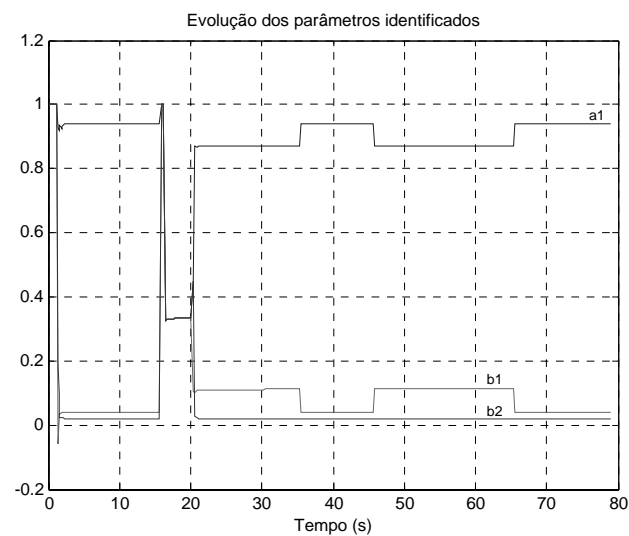


Figura 4.90: Evolução dos parâmetros identificados durante o ensaio E2, abordagem híbrida, sistema S1.

A tabela 4.47 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Como esperado para esta abordagem o valor obtido para ISE (e para ITSE) nos ensaios E1 e E2 é melhor do que o obtido para a abordagem dinâmica, sendo melhor do que para o ensaio com h fixo e igual a 0.28 s.

Teste	$h = 0.12$ s	E1	E2	$h = 0.28$ s
ISE	13.9	15.4	15.7	17.3
ITSE	854	944	960	1070

Tabela 4.47: Valor de ISE e de ITSE para os ensaios da abordagem híbrida, sistema S1.

Para o ensaio E1 a degradação em relação ao melhor caso é de 11% e a melhoria em relação ao pior caso é também de 11%. Para o ensaio E2 a degradação é de 13% e a melhoria de 9%. Os valores obtidos para a melhoria quando se utilizam os índices ISE ou ITSE são semelhantes.

Para esta abordagem a comutação de controladores é preferível em relação à utilização do valor fixo de h igual a 0.28 s.

Sistema S2

Os sinais obtidos para os ensaios E1 e E2 são apresentados nas figuras 4.91 e 4.92.

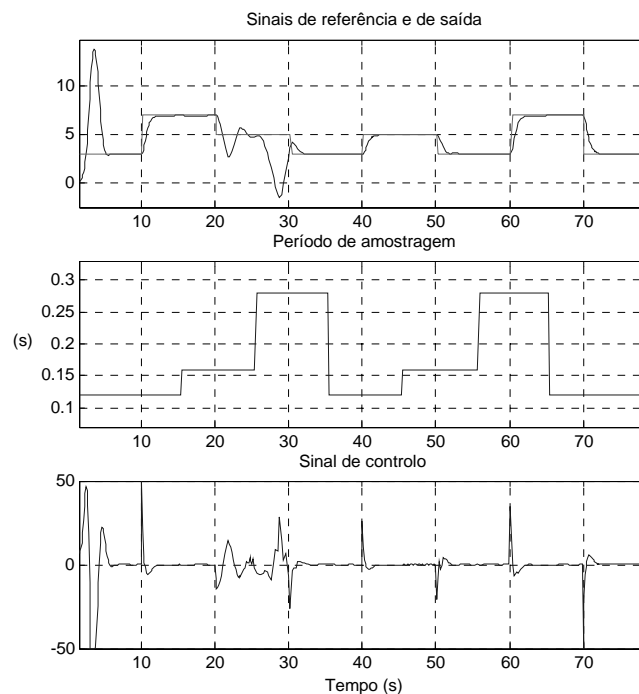


Figura 4.91: Sinais para o ensaio E1, abordagem híbrida, sistema S2.

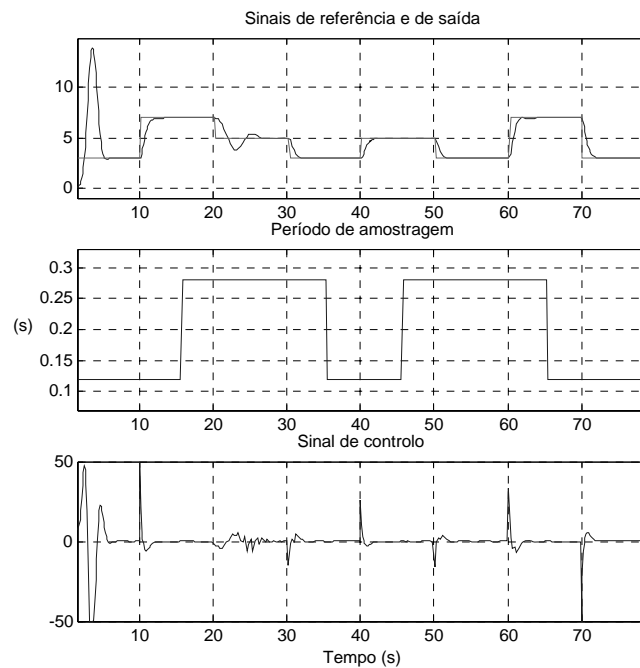


Figura 4.92: Sinais para o ensaio E2, abordagem híbrida, sistema S2.

Os gráficos com a evolução dos parâmetros do modelo discreto do sistema são apresentados nas figuras 4.93 e 4.94.

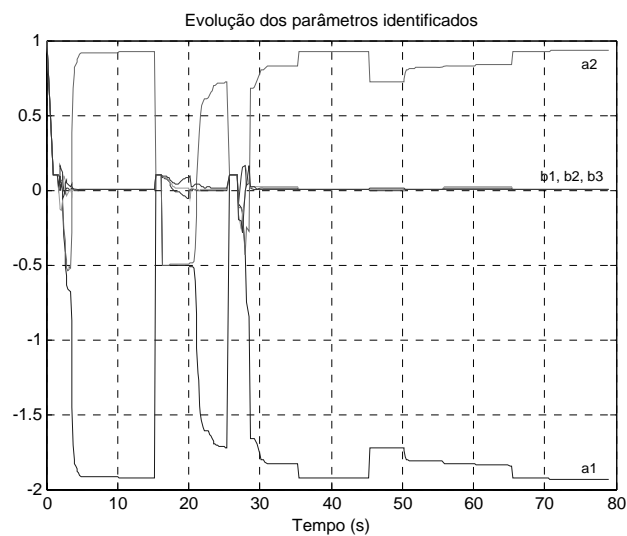


Figura 4.93: Evolução dos parâmetros identificados durante o ensaio E1, abordagem híbrida, sistema S2.

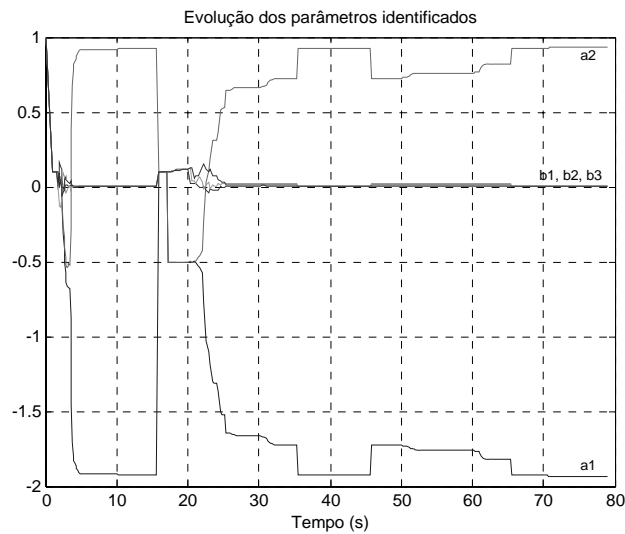


Figura 4.94: Evolução dos parâmetros identificados durante o ensaio E2, abordagem híbrida, sistema S2.

Apresentam-se também os resultados dos ensaios com h fixo igual a 0.12 s e 0.28 s.

A tabela 4.48 apresenta o valor de ISE e de ITSE para cada um dos ensaios realizados.

Teste	$h = 0.12$ s	E1	E2	$h = 0.28$ s
ISE	21.4	23.2	23.5	24.9
ITSE	1318	1427	1440	1540

Tabela 4.48: Valor de ISE e de ITSE para os ensaios da abordagem híbrida, sistema S2.

Como esperado para esta abordagem o valor obtido para ISE (e para ITSE) nos ensaios E1 e E2 é melhor do que o obtido para a abordagem dinâmica, sendo também melhor do que para o ensaio com h fixo e igual a 0.28 s.

Para o ensaio E1 a degradação em relação ao melhor caso é de 8% e a melhoria em relação ao pior caso é de 7%. Para o ensaio E2 a degradação é de 9% e a melhoria de 6%. Os resultados da melhoria/ degradação obtidos para ITSE são semelhantes.

Para esta abordagem a comutação de controladores resulta num melhor desempenho de controlo relativamente à utilização do valor fixo de h igual a 0.28 s e tal como esperado esta abordagem produz melhores resultados do que a abordagem dinâmica.

A melhoria do desempenho de controlo em relação ao valor que se obtém para h fixo e igual a 0.28 s depende directamente da forma como h varia ao longo do tempo.

A técnica de adaptação dinâmica da taxa de amostragem propõe-se alterar o valor de h em situações de sobrecarga pontual do barramento de modo a que o escalonador do sistema possa dispôr de mais largura de banda para escalonar os pedidos de envio

de mensagens pendentes. Como estas situações são esporádicas, a malha de controlo tenderá a funcionar durante a maior parte do tempo com valores de h menores do que 0.28 s.

A figura 4.95 apresenta o exemplo de um ensaio em que o valor de h se mantém igual a 0.16 s durante a maior parte do tempo, utilizando o sistema S1.

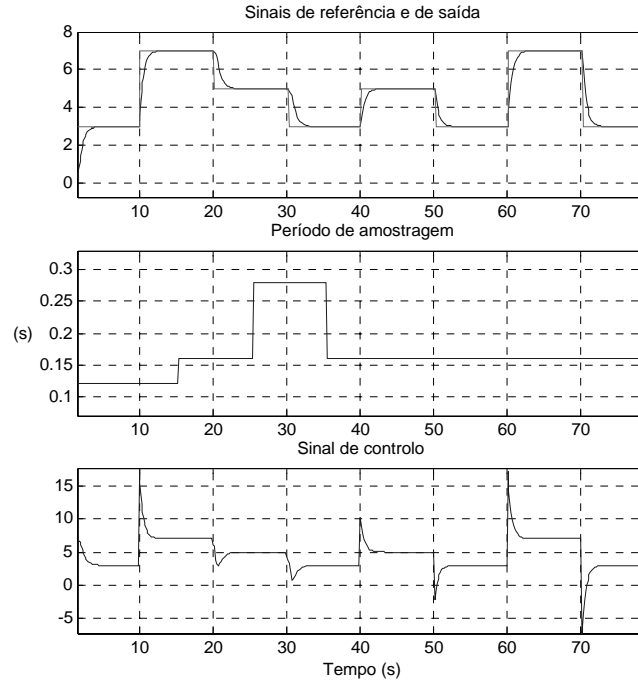


Figura 4.95: Sinais para o teste em que $h < 0.28$ s a maior parte do tempo, sistema S1.

Para este teste o valor de ISE é de 14.6 (ITSE igual a 904). Este valor representa uma degradação de apenas 5% em relação ao ensaio em que $h = 0.12$ s permitindo a flexibilidade suficiente para que o escalonador possa servir alguns pedidos pendentes. O mecanismo de QoS associado ao *master* do protocolo FTT-CAN permite a definição de políticas de gestão de qualidade onde pode também ser tida em conta a qualidade de controlo de cada malha.

Para esta abordagem efectuaram-se ainda mais dois testes complementares utilizando o sistema S1. Num dos testes as características do sistema a controlar variam a meio do ensaio (ensaio C1) e no outro verifica-se a existência de ruído (ensaio C2).

No primeiro caso a função de transferência do sistema a controlar varia de $H(s) = \frac{0.5}{s+0.5}$ para $H(s) = \frac{0.5}{s+1}$ para $t = 40$ s. A figura 4.96 apresenta os sinais obtidos para este teste e a figura 4.97 apresenta a evolução dos parâmetros do modelo discreto ao longo do tempo.

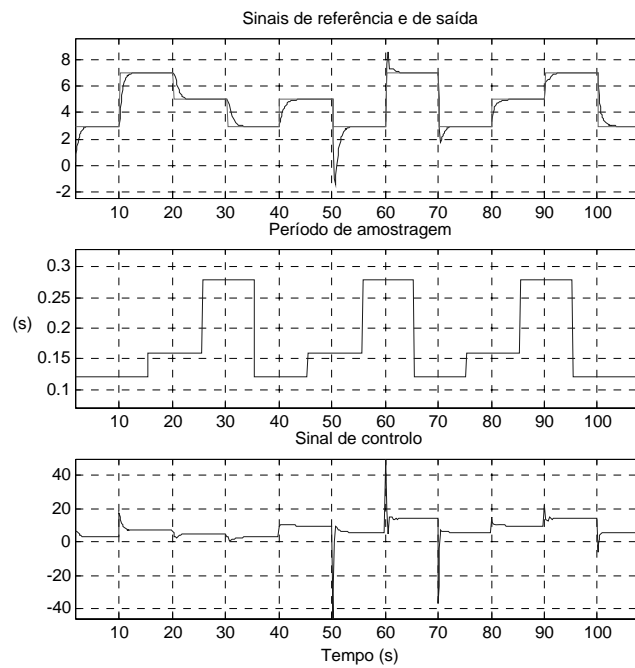


Figura 4.96: Sinais para o ensaio C1, abordagem híbrida, sistema S1.

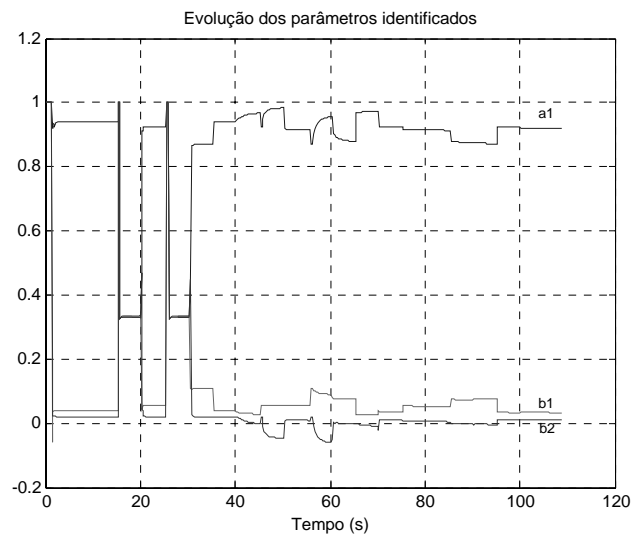


Figura 4.97: Evolução dos parâmetros do modelo discreto do sistema durante o ensaio C1, abordagem híbrida, sistema S1.

A análise das figuras permite concluir que, de acordo com o esperado, o controlador é capaz de se adaptar às novas características do sistema através do ajuste dos estimadores associados a cada valor de h utilizado. Verifica-se que há uma fase de transição durante a qual os estimadores se ajustam aos novos valores (entre $t = 40$ s e $t = 80$ s) e a partir daí o o sinal de saída segue o sinal de referência do modo usual. Verifica-se assim de modo prático a capacidade de adaptação característica da abordagem híbrida.

No teste C2 o sistema S1 é sujeito a ruído a partir de $t = 42$ s. O ruído introduzido é ruído do tipo branco com média aproximadamente nula e desvio padrão igual a 1.

Os sinais obtidos são apresentados na figura 4.98 e a evolução dos parâmetros da função discreta do sistema na figura 4.99.

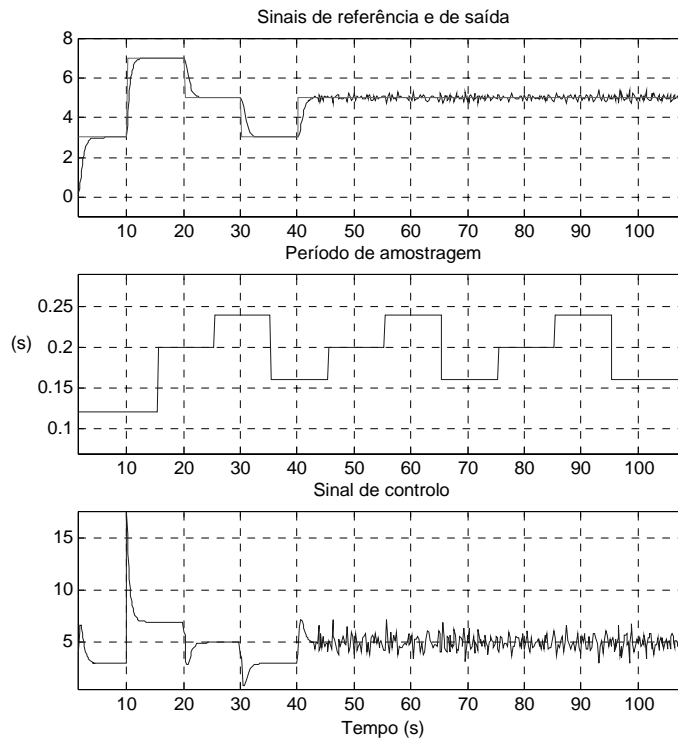


Figura 4.98: Sinais para o ensaio C2, abordagem híbrida, sistema S1.

Na presença de ruído, e tal como esperado, os parâmetros estimados apresentam oscilações. Apesar disso o controlador consegue seguir o sinal de referência e o processo de comutação do valor de h pode continuar a ser efectuado.

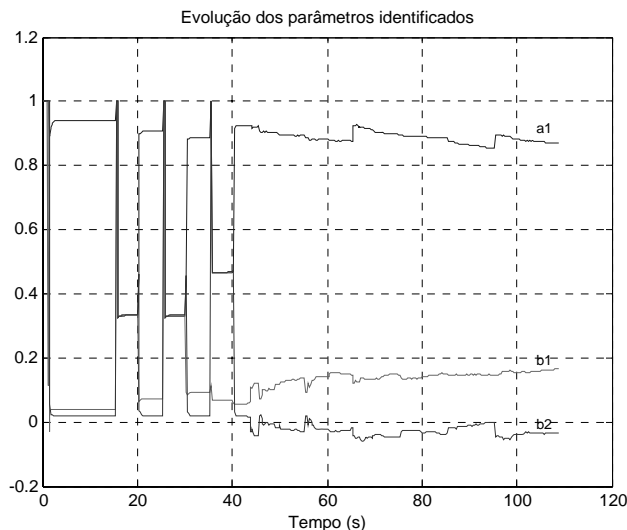


Figura 4.99: Evolução dos parâmetros do modelo discreto do sistema para o teste C2, abordagem híbrida, sistema S1.

4.4.5 Optimização local de malhas de controlo (resultados preliminares)

Para avaliar o interesse da comutação do período de amostragem das malhas de controlo em função do atraso médio entre a amostragem e a actuação a que essas malhas estão sujeitas, com o objectivo de melhorar o desempenho de controlo em malha fechada, realizaram-se alguns testes utilizando os sistemas S1 e S2 (ver secção 4.1.3).

Os controladores utilizados são controladores adaptativos distribuídos por posicionamento de pólos com as características apresentadas no início deste capítulo na secção 4.1.4.

Os testes são efectuados para dois valores distintos do período de amostragem $h = 0.16$ s e $h = 0.20$ s, valores escolhidos dentro da gama proposta por Åström e Wittenmark [98], já por várias vezes referida.

O padrão de atraso é seleccionado de modo a que o valor médio seja elevado em relação ao valor inicial do período de amostragem, que neste caso é $h_{inic} = 0.16$ s.

O histograma relativo ao padrão do atraso utilizado nos ensaios preliminares é apresentado na figura 4.100 e foi gerado com base na função *gamrnd* do MATLAB. Este padrão de carga é utilizado nos dois ensaios realizados.

O primeiro ensaio (L1) é realizado para $h_{inic} = 0.16$ s, com o padrão de carga apresentado na figura 4.100. Para este ensaio a relação R_{ha} é igual a 0.79. Os sinais obtidos para os sistemas S1 e S2 são apresentados, respectivamente, nas figuras 4.101 e 4.102.

No segundo ensaio (L2), o período de amostragem da malha é comutado para $h_{final} = 0.20$ s, sendo utilizado o mesmo padrão de carga.

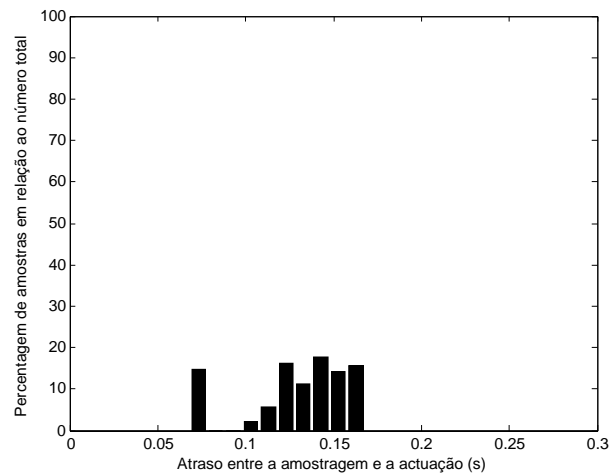


Figura 4.100: Atraso entre a amostragem e a actuação, padrão utilizado nos ensaios preliminares.

Nesta caso a relação entre o atraso médio e o valor do período de amostragem diminui para $R_{ha} = 0.64$. Os sinais obtidos para este ensaio para os sistemas S1 e S2 são apresentados, respectivamente, nas figuras 4.103 e 4.104.

Para avaliar o desempenho de controlo utilizaram-se os índices ISE e ITSE calculados entre os instantes $t = 8$ s e $t = 23$ s. Os resultados obtidos para os testes L1, com $R_{ha} = 0.79$, e L2, com $R_{ha} = 0.64$, são apresentados nas tabelas 4.49 e 4.50, respectivamente para os sistemas S1 e S2.

R_{ha}	0.79	0.64
ISE	4.0	3.3
ITSE	51	43

Tabela 4.49: Valor de ISE e de ITSE para os ensaios preliminares, sistema S1.

R_{ha}	0.79	0.64
ISE	5.4	5.3
ITSE	69	68

Tabela 4.50: Valor de ISE e de ITSE para os ensaios preliminares, sistema S2.

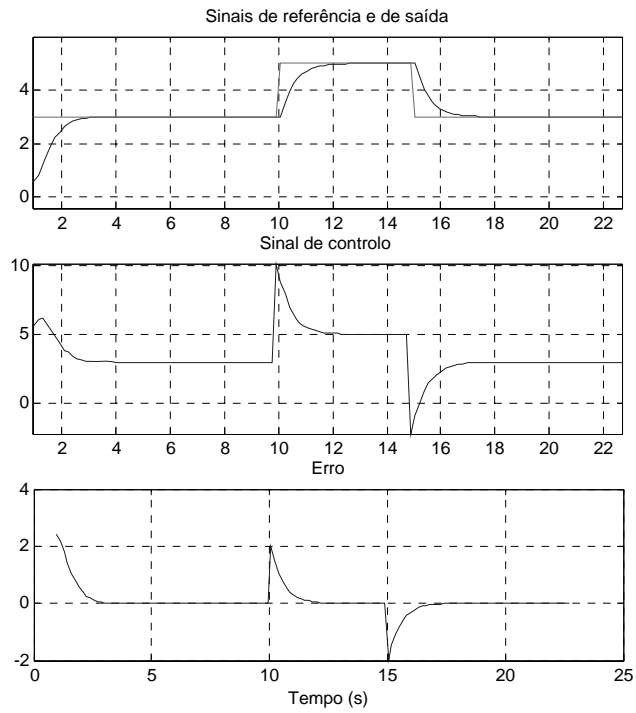


Figura 4.101: Sinais para o ensaio L1, sistema S1.

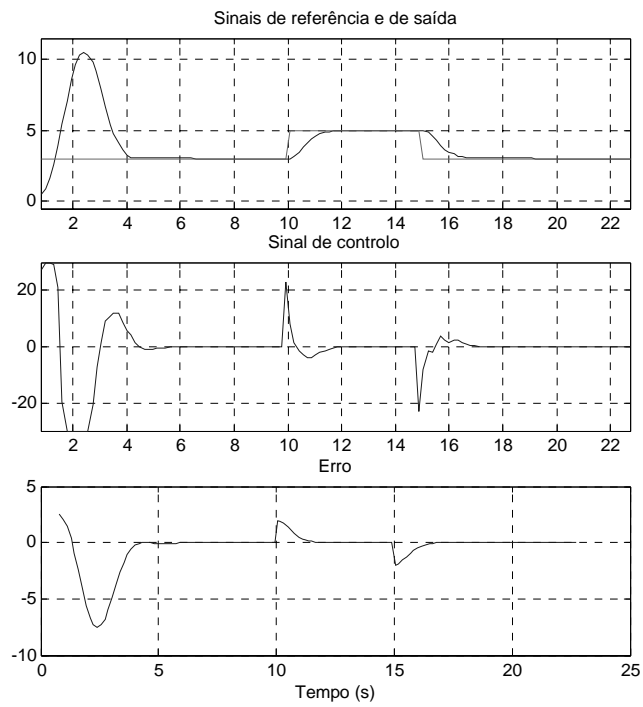


Figura 4.102: Sinais para o ensaio L1, sistema S2.

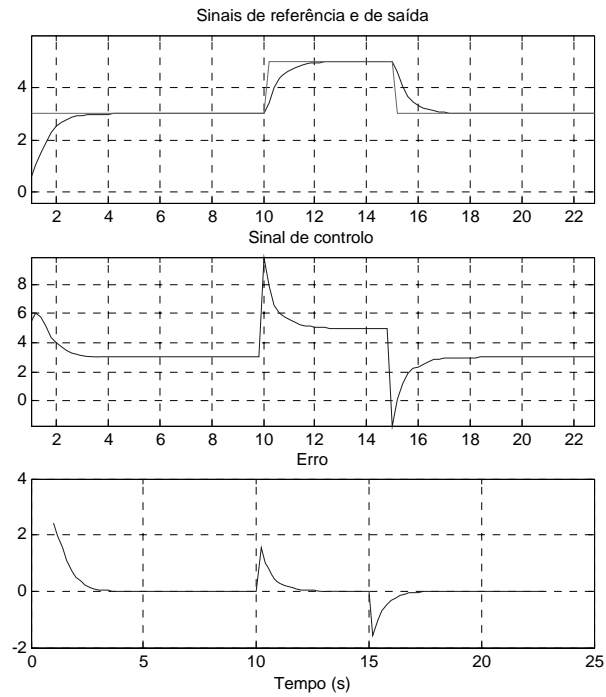


Figura 4.103: Sinais para o ensaio L2, sistema S1.

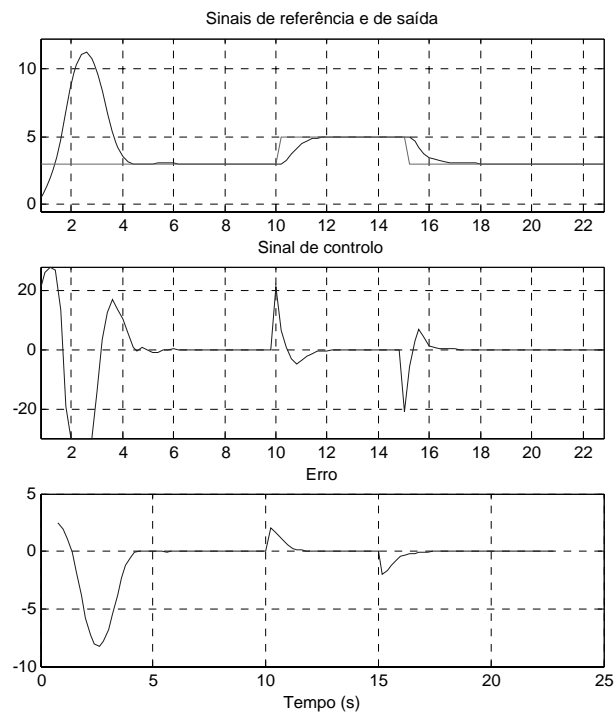


Figura 4.104: Sinais para o ensaio L2, sistema S2.

Os resultados mostram que, nos casos avaliados, o desempenho de controlo melhora com o aumento do período de amostragem das malhas de controlo. A melhoria é mais evidente para o sistema de 1ª ordem S1.

No entanto ficam ainda em aberto uma série de questões como a definição do limiar de R_{ha} a partir do qual deve ser feita a alteração do período de amostragem e como é que deve ser escolhido o novo valor do período de amostragem.

Os resultados obtidos para S2 mostram que, para este sistema e com os valores inicial e final de h utilizados, a alteração no desempenho de controlo não é muito significativa. Seria por isso interessante obter uma métrica que permitisse associar o benefício obtido para o desempenho de controlo em função do valor de h ou de R_{ha} de modo a que a alteração do período de amostragem seja efectuada apenas quando for relevante para a aplicação em causa.

4.5 Conclusão

Neste capítulo apresentaram-se os resultados obtidos nas simulações das implementações propostas para os controladores adaptativos distribuídos por posicionamento de pólos e da técnica de adaptação dinâmica da taxa de amostragem.

Primeiro descreveram-se as arquitecturas utilizadas para os sistemas de teste, os padrões de carga, os sistemas a controlar, os controladores e os critérios de avaliação do desempenho em malha fechada. Depois descreveu-se brevemente o simulador True-Time e a implementação das arquitecturas de teste utilizando este simulador. Por fim apresentaram-se os resultados das simulações e os comentários relevantes.

Os resultados obtidos para os controladores adaptativos distribuídos por posicionamento de pólos mostram que a modelação do atraso entre a amostragem e a actuação como tempo morto fraccionário, embora origine controladores mais lentos e mais complexos, devido à existência de um zero extra no modelo discreto, conduz a um melhor desempenho de controlo, especialmente nos casos em que o atraso provoca oscilações e instabilidade quando se utilizam os métodos de modelação “clássicos”.

Os resultados obtidos através da aplicação da técnica de adaptação dinâmica da taxa de amostragem confirmam a sua utilidade ao oferecerem alguma flexibilidade de operação ao sistema distribuído global onde a malha se insere, ainda que essa flexibilidade seja conseguida à custa de uma ligeira degradação do desempenho de controlo. Por outro lado a utilização desta técnica de adaptação de h permite obter melhor desempenho de controlo do que aquele que seria obtido utilizando um valor de h fixo mas elevado.

Por último apresentaram-se alguns resultados preliminares referentes à aplicação da técnica de adaptação dinâmica da taxa de amostragem para a optimização local do desempenho das malhas de controlo. Estes resultados mostram que essa aplicação pode ser interessante, embora permaneçam em aberto uma série de questões relativas à sua implementação.

Capítulo 5

Técnicas de controlo não-lineares para sistemas distribuídos

Nos capítulos anteriores foram propostas técnicas de controlo adaptativo que permitem o projecto de controladores distribuídos que têm em conta o atraso entre a amostragem e a actuação a que a malha está sujeita durante a sua operação. As propostas referidas têm por base as técnicas de controlo “clássicas” para o projecto de controladores adaptativos mas utilizam um modelo da função discreta do sistema que tem em conta o atraso na malha. As técnicas propostas permitem projectar o controlador distribuído de raiz.

Neste capítulo é proposta uma abordagem diferente para a melhoria do desempenho de controlo em sistemas de controlo distribuído. Esta abordagem propõe a inclusão na malha de controlo de um compensador que tem por objectivo diminuir a degradação do desempenho provocada pelos atrasos existentes na malha devidos ao processo de distribuição e à sua implementação. A acção do compensador baseia-se no valor do atraso que afecta a malha de controlo a cada instante. Esta técnica foi designada por compensador de atraso.

O princípio genérico de actuação do compensador de atraso é aqui descrito, assim com a sua implementação utilizando técnicas de modelação não-lineares como a lógica difusa e as redes neuronais. O compensador de atraso foi testado recorrendo ao simulador TrueTime. As condições de realização dos testes, embora semelhantes às utilizadas no capítulo 4, são também descritas e os resultados obtidos são apresentados.

Finalmente é efectuada uma comparação entre o desempenho das técnicas de controlo lineares e não-lineares propostas neste trabalho.

Ao longo do texto utiliza-se por vezes a designação de atraso para referir o atraso existente na malha de controlo entre o instante de amostragem e o instante de actuação, cuja origem foi analisada no capítulo 2.

5.1 O princípio do compensador de atraso

A técnica do compensador de atraso propõe a inclusão de um bloco de compensação num sistema de controlo distribuído em que o controlador foi projectado sem ter em conta o efeito do atraso existente na malha de controlo. O bloco de compensação actua com base no valor do atraso que afecta a malha a cada ciclo de controlo mas pode ter outras entradas que sejam consideradas relevantes.

O princípio de acção do compensador de atraso tem por base o esquema do Controlador Aditivo (*Additive Feedforward Controller* - AFC) [137], cujo princípio de operação consiste na adição de um modelo inverso a uma malha de controlo realimentada cujo comportamento não é satisfatório [138]. A estrutura utilizada para o controlo aditivo é apresentada na figura 5.1.

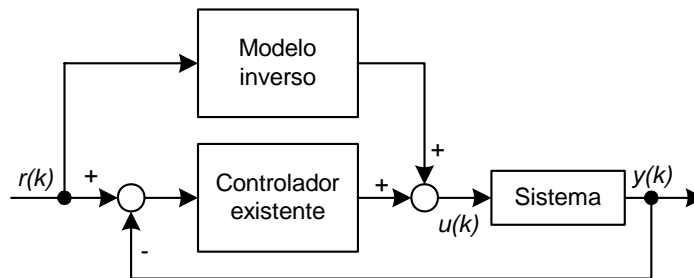


Figura 5.1: Estrutura utilizada para o controlo aditivo.

Neste esquema de controlo se o modelo inverso for de boa qualidade a saída do sistema aproxima-se do valor do sinal de referência, fazendo com que o sinal de entrada do controlador existente se aproxime de zero e conseqüentemente com que a sua contribuição seja pequena. No entanto o controlador existente pode ser útil devido por exemplo ao efeito da realimentação [138]. Esta técnica apresenta as seguintes vantagens: a recolha de informação pode ser feita sem abrir a malha de controlo existente, evitando a paragem de funcionamento do sistema e facilitando o acesso a dados de boa qualidade (para a elaboração do modelo inverso); não existe a necessidade de abrir a malha de controlo para fazer a introdução do novo controlador; permite determinar de forma clara a contribuição de cada um dos blocos [138].

O princípio do compensador de atraso [139] assemelha-se à estrutura do controlador aditivo na medida em que o bloco de compensação é colocado em paralelo com o controlador existente, sendo possível determinar a contribuição individual de cada um deles. No entanto, neste caso não é suposto o compensador substituir o controlador existente, como acontece no caso do controlador aditivo se o modelo inverso for de boa qualidade [138].

O diagrama de blocos do princípio do compensador de atraso é apresentado na figura 5.2.

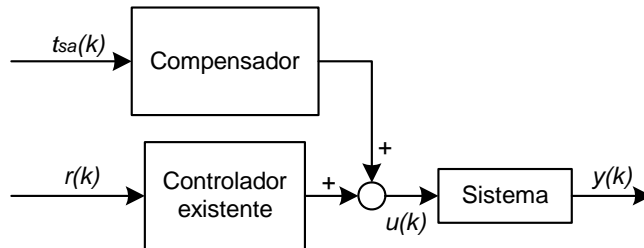


Figura 5.2: Diagrama de blocos do princípio do compensador de atraso.

Tal como foi referido, o compensador de atraso actua com base no conhecimento do atraso variável entre o instante de amostragem e o instante de actuação $t_{sa}(k)$, podendo, se necessário, ter outras entradas. A acção do compensador tem por objectivo corrigir o sinal de controlo de modo a compensar a degradação do desempenho provocada pelo atraso. O controlador utilizado neste esquema pode ser qualquer tipo de controlador realimentado.

Esta estrutura pode ser aplicada a qualquer sistema de controlo distribuído desde que o atraso entre o instante de amostragem e o instante de actuação possa ser conhecido em cada iteração de controlo.

A determinação do atraso pode ser feita através de medições durante a operação do sistema ou, se houver conhecimento suficiente sobre o sistema e as suas condições de operação, através de cálculos anteriores à operação do sistema. A solução de efectuar a medição durante a operação do sistema permite maior flexibilidade na gestão do mesmo, embora necessite da existência de uma base de tempo global acessível em cada nó do sistema distribuído. Esta solução pode também condicionar a arquitectura do sistema distribuído levando a que o actuador seja colocado no mesmo nó de processamento que o controlador de modo a que o atraso medido entre a amostragem e a actuação que afecta a malha em cada iteração seja o mais exacto possível.

O compensador pode ser implementado utilizando diferentes técnicas de modelação. Nesta tese apresentam-se implementações do compensador utilizando lógica difusa e redes neuronais.

A implementação das técnicas descritas originou até agora a publicação do artigo [139] e a aceitação dos seguintes [140] e [141].

5.1.1 Compensador difuso

O diagrama de blocos da implementação do compensador de atraso utilizando lógica difusa é apresentado na figura 5.3.

O controlador existente é um controlador por posicionamento de pólos “clássico” (PP) (ver secção 3.5.2).

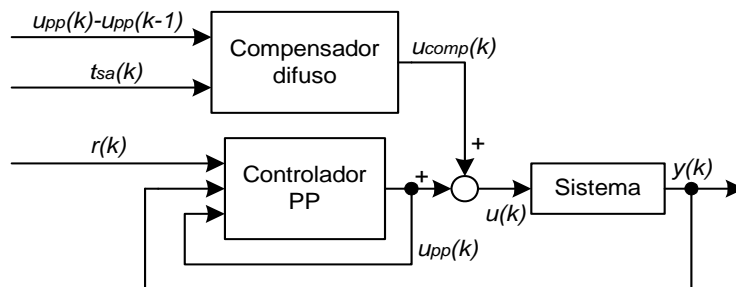


Figura 5.3: Diagrama de blocos da implementação do compensador de atraso utilizando lógica difusa.

A compensação proposta pelo módulo difuso pretende contrariar o efeito que o atraso entre a amostragem e a actuação tem sobre o desempenho em malha fechada do sistema, impedindo a degradação que este introduz.

Quando não existe atraso na malha de controlo o sinal de saída é aplicado ao sistema durante o tempo correspondente ao período de amostragem. Se existir atraso na malha o sinal de saída é aplicado durante menos tempo ao sistema e por isso o seu efeito na saída não é igual ao que seria desejado. O compensador tenta obviar esta situação aumentando ou diminuindo o sinal de controlo de modo a que a saída do sistema seja igual à que se produz quando não há atraso, de acordo com o valor do atraso que afecta o sistema e com a variação do sinal de controlo.

A implementação difusa do compensador de atraso tem duas entradas: o atraso entre o momento de amostragem e o momento de actuação ($t_{sa}(k)$) e a diferença entre o valor actual e o valor anterior do sinal de saída gerado pelo controlador PP ($u_{pp}(k) - u_{pp}(k-1)$). Esta última entrada é utilizada para determinar o sinal e a magnitude da saída do compensador. Se o sinal de controlo estiver a aumentar a saída do compensador deve ser positiva, se estiver a diminuir a saída do compensador deve ser negativa.

A saída do compensador ($u_{comp}(k)$) é adicionada à saída do controlador PP ($u_{pp}(k)$) de acordo com o princípio de compensação já descrito.

O compensador difuso foi implementado utilizando uma função difusa de Mamdani [142] com duas entradas (atraso e diferença entre os valores anteriores do sinal de saída do controlador PP) e uma saída, e com seis regras.

As funções de pertença têm a forma de sino e as entradas e saídas podem ter os valores seguintes: o atraso (t_{sa}) pode ser “pequeno”, “médio” ou “elevado”; a diferença entre os valores anteriores do sinal de controlo do controlador PP (difupp) pode ser “nulo”, “positivo” ou “negativo”; a saída do compensador (ucomp) pode ser “nulo”, “positivo” ou “negativo”.

As funções de pertença correspondentes a cada uma das entradas e à saída são apresentadas nas figuras 5.4, 5.5 e 5.6.

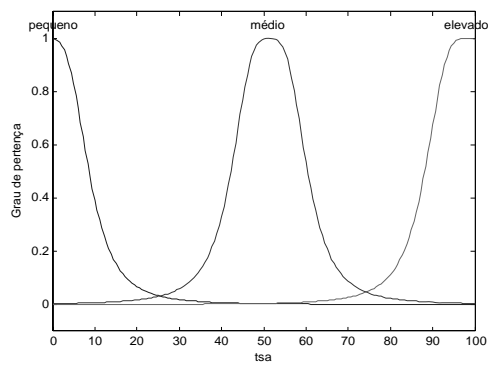


Figura 5.4: Função de pertinência da entrada tsa.

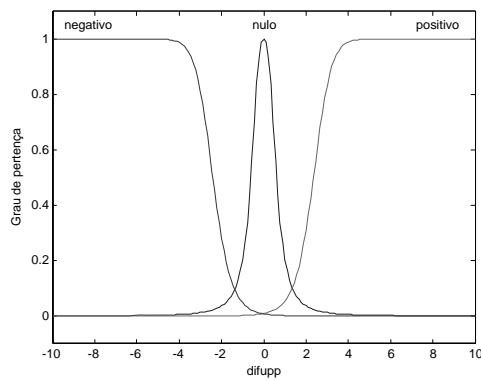


Figura 5.5: Função de pertinência da entrada difupp.

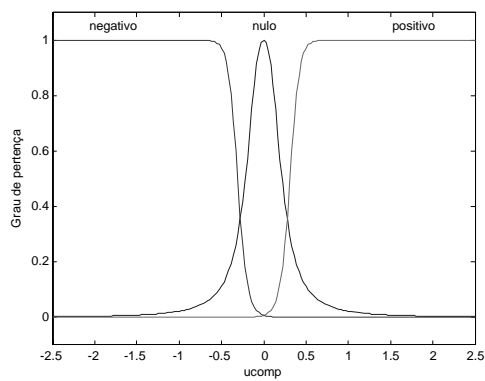


Figura 5.6: Função de pertinência da saída ucomp.

As regras podem resumir-se do seguinte modo:

1. se a diferença entre os valores anteriores do sinal de controlo do controlador PP for “nula” então a saída do compensador é “nula”.
2. se o atraso for “pequeno” a saída do compensador é “nula”.
3. se o atraso é “médio” ou “elevado” a saída do controlador será “média” com o sinal dado pela diferença entre os valores anteriores do sinal de controlo do controlador PP: se o sinal de controlo do controlador PP está a diminuir a saída será “negativa”, se o sinal de controlador PP estiver a aumentar a saída será “positiva”.

A superfície resultante do modelo difuso desenvolvido é apresentada na figura 5.7.

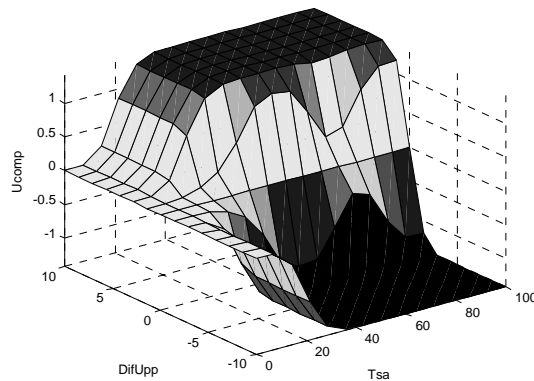


Figura 5.7: Superfície resultante do modelo difuso implementado para o compensador difuso.

Os resultados obtidos com a utilização desta estratégia de controlo são apresentados na secção 5.2.2.

5.1.2 Compensador neuronal

Os resultados obtidos com o compensador difuso, como será visível mais adiante, mostram que o desempenho de controlo é melhor do que aquele que se obtém sem o compensador mas levam também à constatação de que a aproximação que serviu de base à definição das suas regras não permitem descrever de forma exacta os efeitos do atraso na malha de controlo.

Numa tentativa de obter uma melhor modelação do efeito do atraso optou-se por outra solução não-linear recorrendo a uma rede neuronal para a implementação do compensador.

O modelo utilizado para a implementação do compensador neuronal foi desenvolvido utilizando uma metodologia semelhante à que se aplica para a construção de modelos dinâmicos não-lineares. No caso presente pretende-se modelar o efeito do atraso sobre o sistema e não o comportamento global do sistema quando sujeito a atrasos.

A informação disponível para o treino do modelo corresponde aos valores da saída do sistema sujeito a atrasos ($y_a(k)$) ou sem atrasos ($y(k)$) e ao valor do atraso existente na malha. O objectivo é o de produzir um modelo que possa compensar o efeito do atraso de modo a corrigir o sinal de controlo aplicado ao sistema evitando a degradação do desempenho de controlo.

Para obter o modelo consideraram-se duas hipóteses: calcular o erro no sinal de saída $e_y(k) = y(k) - y_a(k)$ e reportá-lo à entrada através de um modelo inverso ou, alternativamente, calcular a entrada equivalente num sistema sem atraso à saída obtida para o sistema com atraso $y_a(k)$.

A primeira solução só seria válida se o sistema fosse linear, visto que para um sistema não-linear não há garantias de que o erro na saída possa ser reportado para a entrada uma vez que a gama de valores do erro é muito diferente da gama de valores do sinal de saída. A utilização desta solução poderia resultar numa distorção no valor de entrada devida à não-linearidade. Optou-se por isso pela segunda alternativa, que também utiliza um modelo inverso.

O diagrama de blocos do método utilizado para calcular a saída do modelo é apresentado na figura 5.8.

O compensador neuronal tem cinco entradas: o valor anterior da sua saída, as duas amostras anteriores da saída do controlador PP e os dois valores anteriores do atraso.

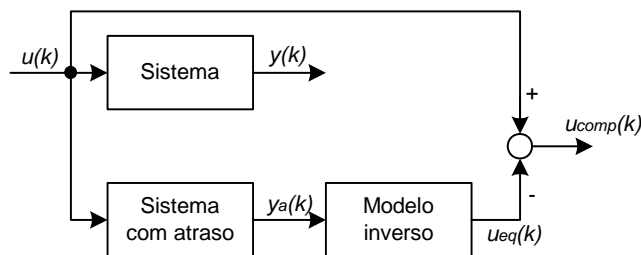


Figura 5.8: Diagrama de blocos do método utilizado para calcular a saída do modelo neuronal.

Utilizando o método apresentado na figura 5.8 e fazendo a análise dos regressores obteve-se o modelo representado na figura 5.9.

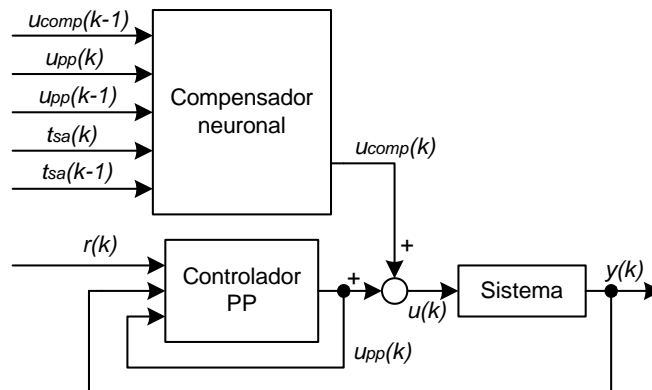


Figura 5.9: Diagrama de blocos da implementação do compensador de atraso utilizando redes neurais.

O modelo é composto por dez neurónios na camada escondida que têm como função de activação a tangente hiperbólica e por um neurónio na camada de saída com função de activação linear. O modelo foi treinado durante quinze mil iterações com o algoritmo Levenberg-Marquardt [143] e [144].

5.2 Implementações do compensador de atraso

Nesta secção apresentam-se as condições de teste e os resultados obtidos nas implementações do compensador de atraso utilizando o compensador difuso e o compensador neuronal.

5.2.1 Condições de teste

A acção do compensador de atraso é baseada no conhecimento do valor do atraso entre o instante de amostragem e o instante de actuação. Nas implementações efectuadas esse atraso é medido durante a operação do sistema e por isso a actuação tem que ser efectuada no mesmo nó de processamento em que for colocado o controlador. Deste modo a arquitectura do sistema de controlo não é totalmente distribuída como nos exemplos apresentados no capítulo 4. A arquitectura utilizada para a implementação do compensador de atraso é apresentada na figura 5.10.

Nesta arquitectura existe apenas uma mensagem (M1) associada à malha de controlo, cujas características são apresentadas na tabela 5.1. O envio desta mensagem é efectuado na sequência da activação da tarefa que faz a amostragem do sistema no nó do sensor, sendo por isso *event-triggered*.

O controlador existente é um controlador por posicionamento de pólos “clássico” (PP), ou seja é baseado num modelo discreto do sistema que não tem em conta o atraso entre o instante de amostragem e o instante de actuação.

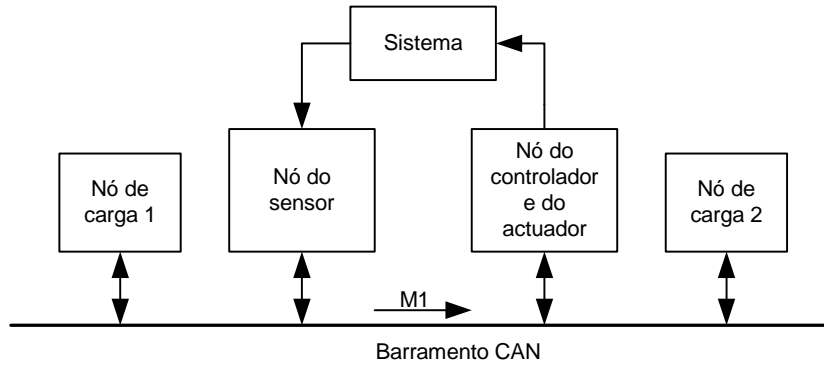


Figura 5.10: Diagrama de blocos do sistema utilizado para o teste do compensador de atraso.

Mensagem	Identificador	DLC	Período
M1	130	1	$\simeq h$

Tabela 5.1: Características da mensagem da malha de controlo.

No teste das várias implementações do compensador de atraso utilizaram-se os padrões de carga P1, P2 e P3, apresentados na secção 4.1.2.

Nos testes efectuados utilizou-se o sistema S1.

Nesta implementação os parâmetros deste controlador são calculados antecipadamente com base em ensaios para identificação do sistema e mantêm-se constantes ao longo da operação da malha de controlo, ao contrário do que acontece em alguns dos ensaios do capítulo 4, em que os parâmetros variam ao longo do ensaio de acordo com os parâmetros identificados para a função de transferência discreta do sistema durante a operação da malha.

O modelo utilizado para a função discreta do sistema, que tal como referido, não tem em conta o atraso entre a amostragem e a actuação, é dado pela equação seguinte (ver secção 3.3.3):

$$G(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}$$

O posicionamento do pólo em malha fechada é igual ao que foi utilizado nos ensaios do capítulo 4 para o sistema S1, ou seja, o pólo é colocado em -2 e o observador em -4.

Para cada uma das implementações do compensador de atraso foram efectuados testes com e sem o compensador incluído no sistema de modo a poder aferir-se a sua acção efectiva.

Para além dos testes efectuados com os padrões de carga P1, P2 e P3, efectuou-se ainda um ensaio sem mensagens de carga (ensaio de referência).

O atraso entre a amostragem e a actuação obtido para o ensaio de referência é igual a 4 ms. Este é o valor mínimo devido aos atrasos de processamento e de acesso ao barramento.

Os resultados de controlo obtidos para o sistema S1, sem o compensador na malha, ou seja apenas com o controlador PP, são apresentados nas figuras 5.12 a 5.15, para o teste de referência e para os testes com os padrões de carga P1 a P3, respectivamente.

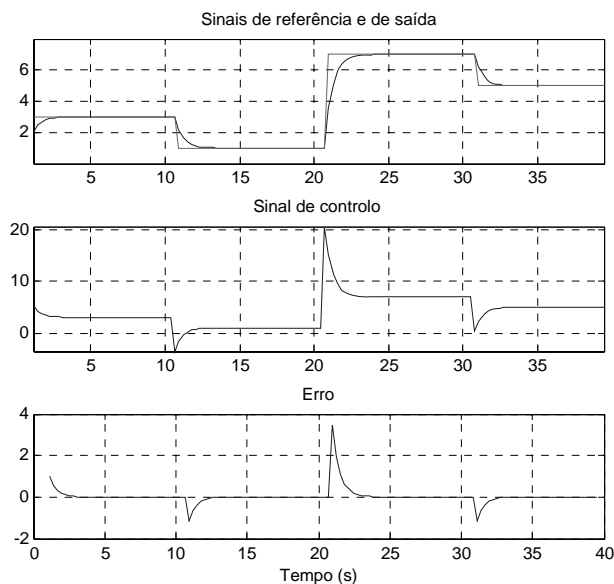


Figura 5.12: Sinais para o teste do compensador de atraso (sem o compensador), teste de referência.

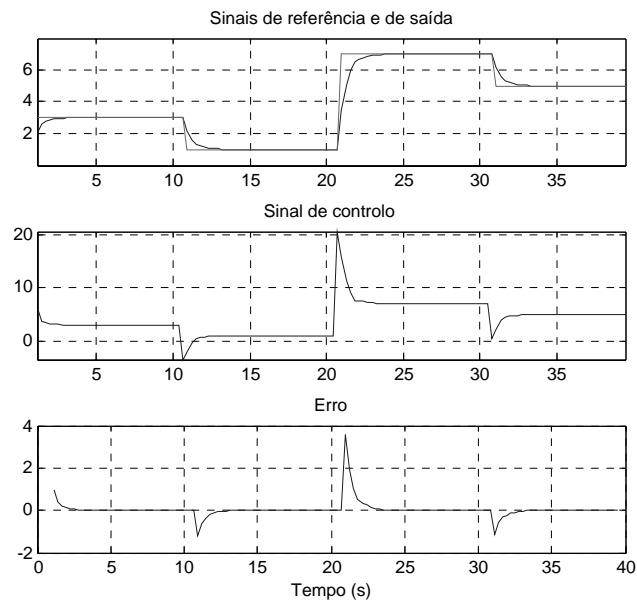


Figura 5.13: Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P1.

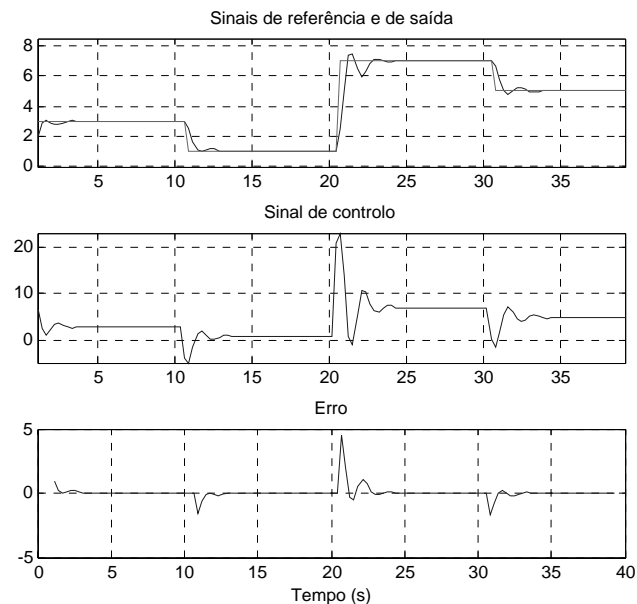


Figura 5.14: Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P2.

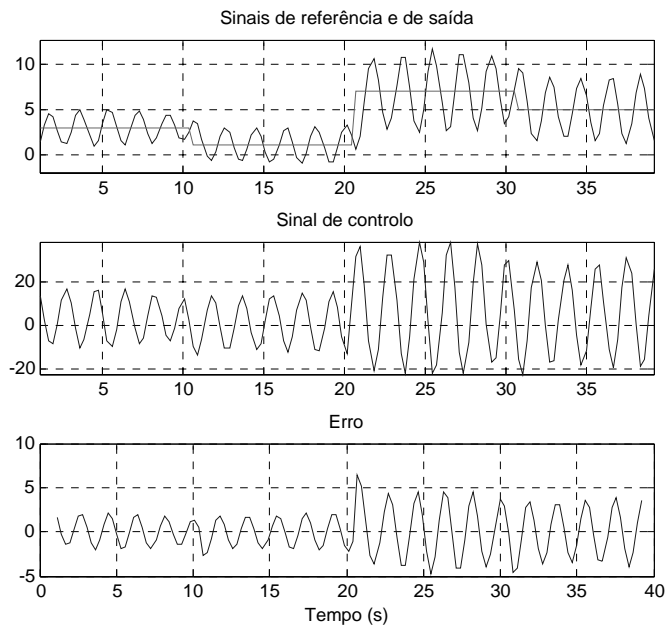


Figura 5.15: Sinais para o teste do compensador de atraso (sem o compensador), para o padrão de carga P3.

O valor de ISE e de ITSE obtidos para estes testes é apresentado na tabela 5.3.

Teste	Referência	P1	P2	P3
ISE	18.3	18.5	21.5	205.0
ITSE	372	374	434	5265

Tabela 5.3: Valor de ISE e de ITSE para os testes do compensador de atraso (sem compensador).

Os resultados obtidos mostram que, tal como esperado, o atraso introduzido na malha de controlo degrada o desempenho de controlo. O padrão de carga P1 provoca uma degradação muito pequena do desempenho de controlo. No teste com o padrão de carga P2 a saída do sistema começa a apresentar oscilações ligeiras quando há alterações no sinal de referência e para o padrão de carga P3 a saída apresenta grandes oscilações ao longo de todo o teste.

5.2.2 Implementação difusa

As figuras 5.16 e 5.17 a 5.22 e 5.23 apresentam, respectivamente, os sinais para o ensaios de referência e para os ensaios com os padrões de carga P1 a P3 e a contribuição do controlador PP e do compensador difuso para o ensaio respectivo.

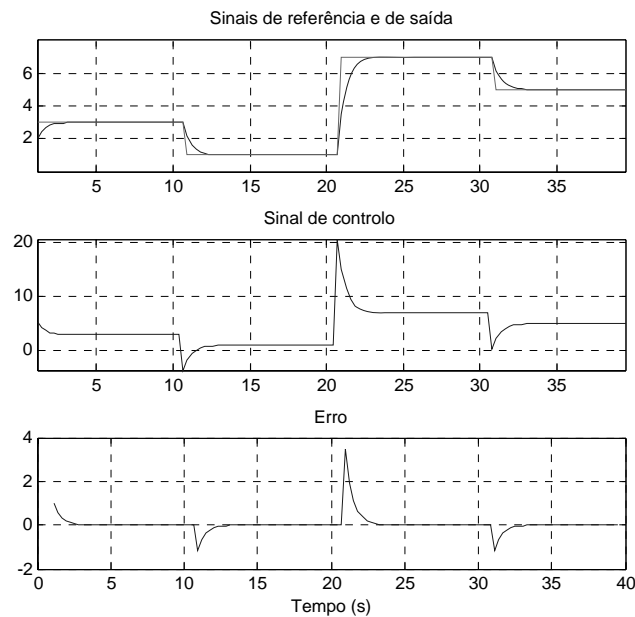


Figura 5.16: Sinais para o teste com o compensador difuso, ensaio de referência.

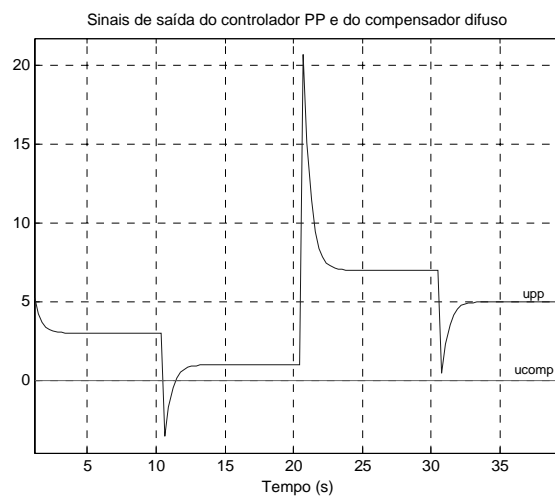


Figura 5.17: Contribuição do controlador PP e do compensador difuso durante o ensaio de referência.

Tal como seria de esperar a contribuição do compensador difuso durante o ensaio de referência é nula.

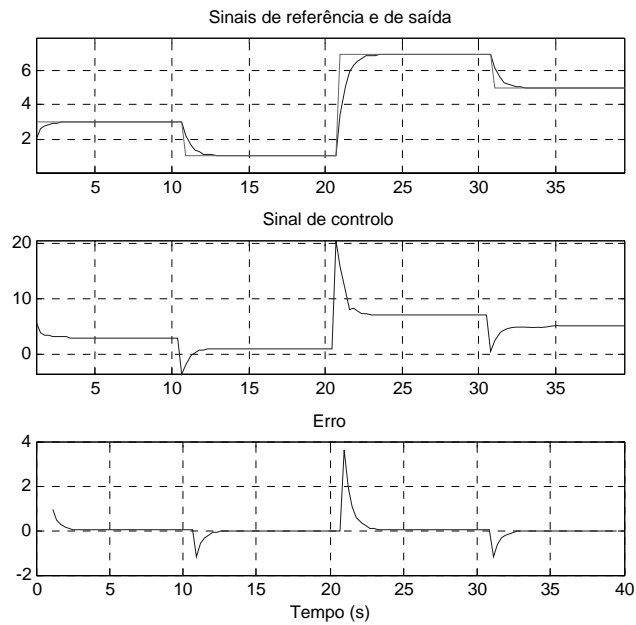


Figura 5.18: Sinais para o teste com o compensador difuso, para o padrão de carga P1.

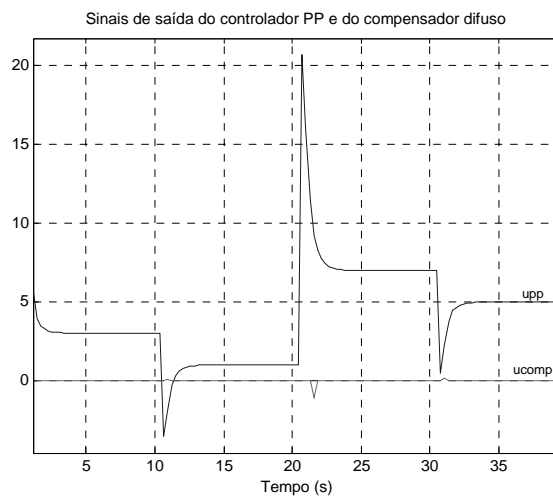


Figura 5.19: Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P1.

No ensaio com o padrão de carga P1 o atraso introduzido na malha de controlo é

relativamente pequeno (quando comparado com o valor de h) pelo que a contribuição do compensador difuso é também ela quase nula.

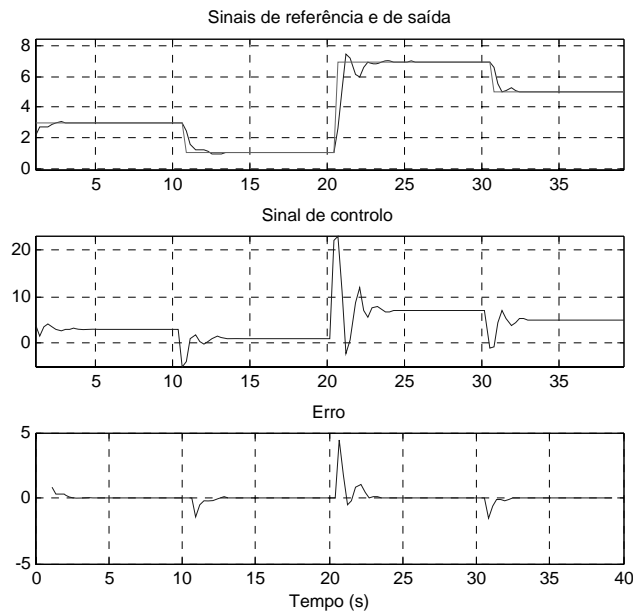


Figura 5.20: Sinais para o teste com o compensador difuso, para o padrão de carga P2.

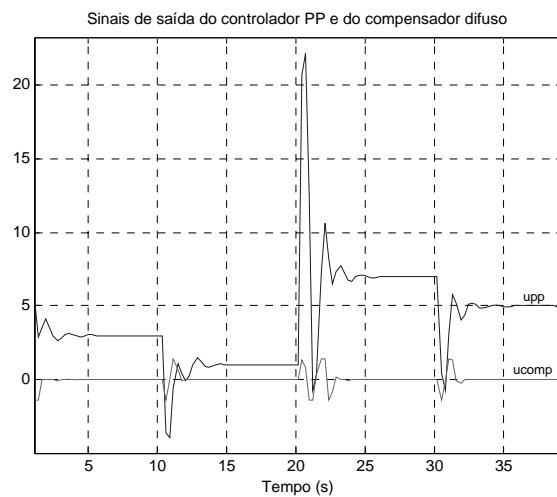


Figura 5.21: Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P2.

Para o ensaio com o padrão de carga P2 o compensador difuso apresenta uma

contribuição visível.

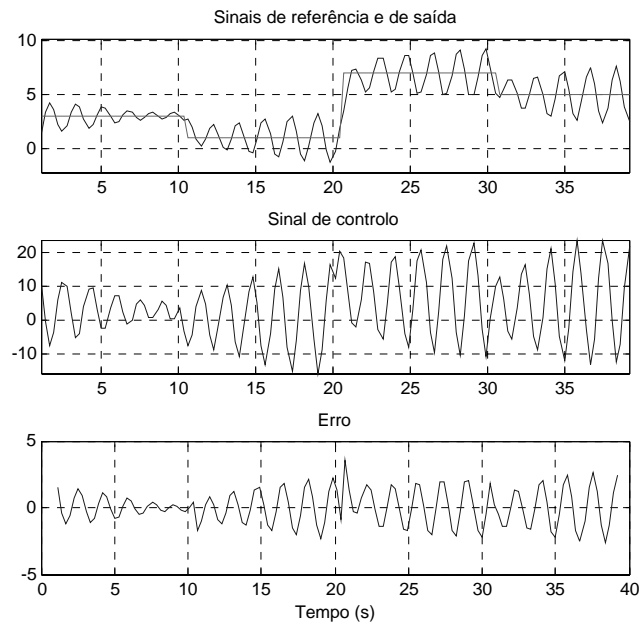


Figura 5.22: Sinais para o teste com o compensador difuso, para o padrão de carga P3.

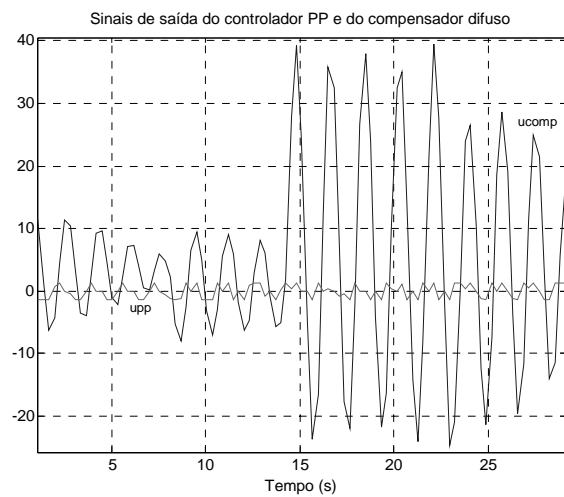


Figura 5.23: Contribuição do controlador PP e do compensador difuso durante o ensaio com o padrão de carga P3.

No ensaio com o padrão de carga P3 a contribuição do compensador difuso é visível mas não é suficiente para impedir a saída do sistema de oscilar, embora a amplitude da oscilação diminua.

O valor de ISE e de ITSE obtidos para estes testes é apresentado na tabela 5.4.

Teste	Referência	P1	P2	P3
ISE	18.3	18.5	20.8	67.3
ITSE	372	375	418	1686

Tabela 5.4: Valor de ISE e de ITSE para os testes com o compensador difuso.

A melhoria obtida no desempenho de controlo com a introdução do compensador de atraso na malha é apresentada na tabela 5.5.

Teste	Referência	P1	P2	P3
M_{ISE}	0%	0%	+3%	+67%
M_{ITSE}	0%	0%	+4%	+68%

Tabela 5.5: Melhoria obtida com o compensador difuso.

Este valor foi calculado através da equação seguinte

$$M_I = \left(1 - \frac{I_{ccomp}}{I_{scomp}}\right) * 100 \quad (5.2)$$

A introdução do compensador difuso no sistema permite melhorar o desempenho de controlo nos ensaios com os padrões de carga P2 e P3, no entanto ele não é capaz de eliminar na totalidade os efeitos do atraso no sistema de controlo. Para o padrão de carga P1 os resultados para ISE (ou ITSE) são praticamente iguais com e sem o compensador. Verifica-se que para o padrão de carga P2 o sinal de saída continua a apresentar ligeiras oscilações e para o padrão de carga P3 o compensador, embora diminua a amplitude das oscilações, não consegue fazer com que o sinal de saída siga de perto o sinal de referência.

Conclui-se por isso que a modelação proposta pelo compensador difuso não consegue replicar totalmente o efeito real do atraso na malha de controlo.

5.2.3 Implementação neuronal

As figuras 5.24 e 5.25 a 5.30 e 5.31 apresentam, respectivamente, os sinais para o ensaio de referência e para os ensaios com os padrões de carga P1 a P3 e a contribuição do controlador PP e do compensador neuronal para o ensaio respectivo.

O valor de ISE e de ITSE obtido para estes testes é apresentado na tabela 5.6.

Teste	Referência	P1	P2	P3
ISE	18.2	18.2	19.2	30.1
ITSE	369	375	378	504

Tabela 5.6: Valor de ISE e ITSE para os testes com o compensador neuronal.

A melhoria obtida no desempenho de controlo em malha fechada com a introdução do compensador de atraso é apresentada na tabela 5.7. Este valor foi calculado através da equação 5.2.

Teste	Referência	P1	P2	P3
M_{ISE}	+1%	+2%	+12%	+88%
M_{ITSE}	+1%	+1%	+13%	+90%

Tabela 5.7: Melhoria obtida para o compensador neuronal.

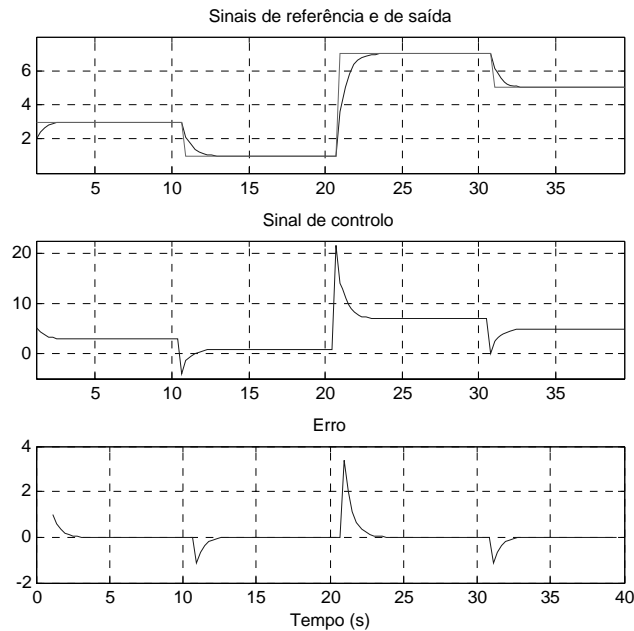


Figura 5.24: Sinais para o teste com o compensador neuronal, ensaio de referência.

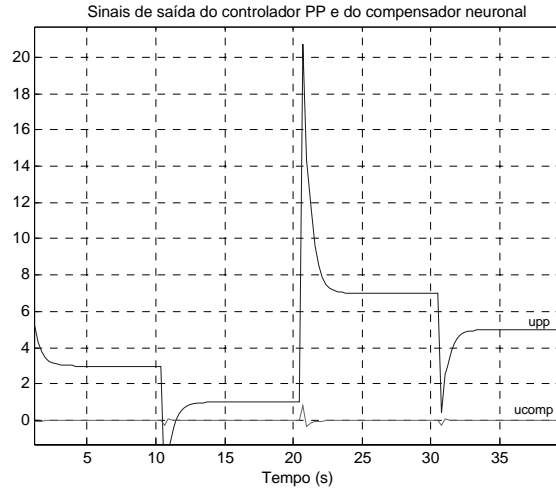


Figura 5.25: Contribuição do controlador PP e do compensador neuronal durante o ensaio de referência.

Durante o ensaio de referência o compensador neuronal tem um contributo nulo durante a maior parte do tempo. No entanto, e embora isso não fosse expectável, existem instantes em que a saída do compensador neuronal não é nula.

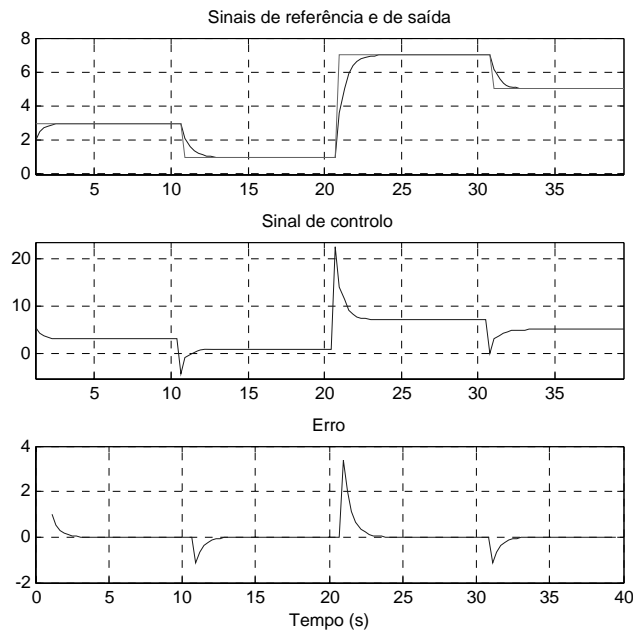


Figura 5.26: Sinais para o teste com o compensador neuronal, para o padrão de carga P1.

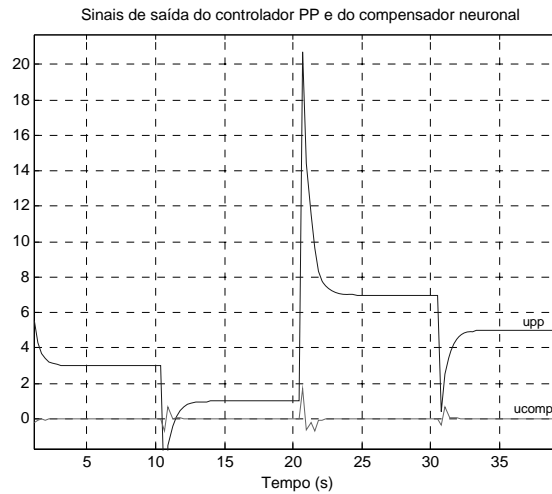


Figura 5.27: Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P1.

Durante o ensaio com o padrão de carga P1 e, apesar do atraso na malha de controle ser considerado pequeno (por comparação com o valor de h), o compensador neuronal propõe a sua contribuição em certos instantes.

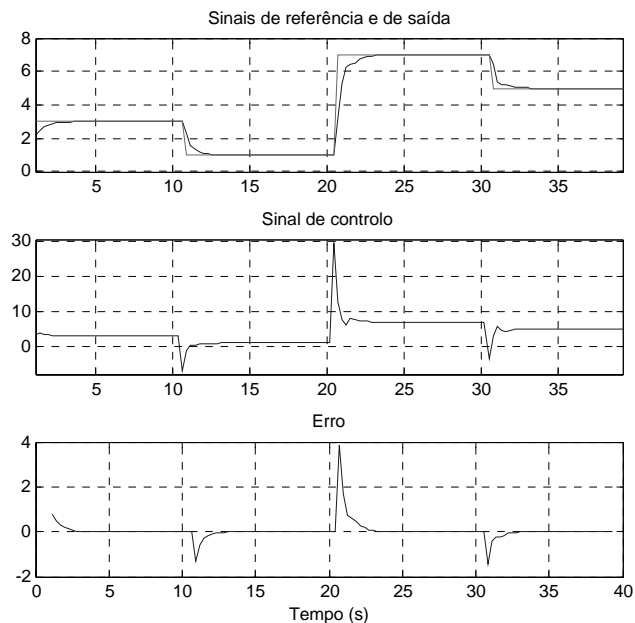


Figura 5.28: Sinais para o teste com o compensador neuronal, para o padrão de carga P2.

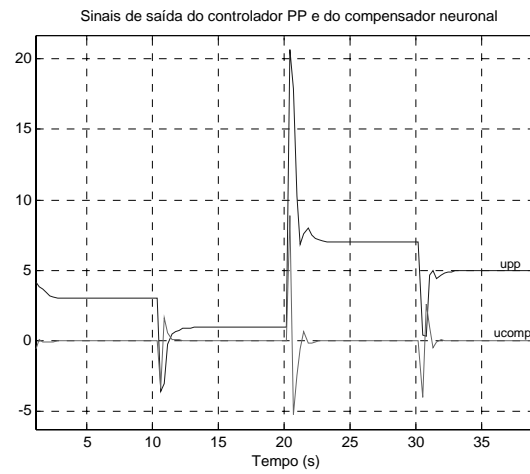


Figura 5.29: Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P2.

Para o ensaio com o padrão de carga P2 a contribuição dada pelo compensador neuronal é claramente visível e faz com que o sinal de saída do sistema deixe de apresentar *overshoot*, o que se verifica quando o compensador não é utilizado.

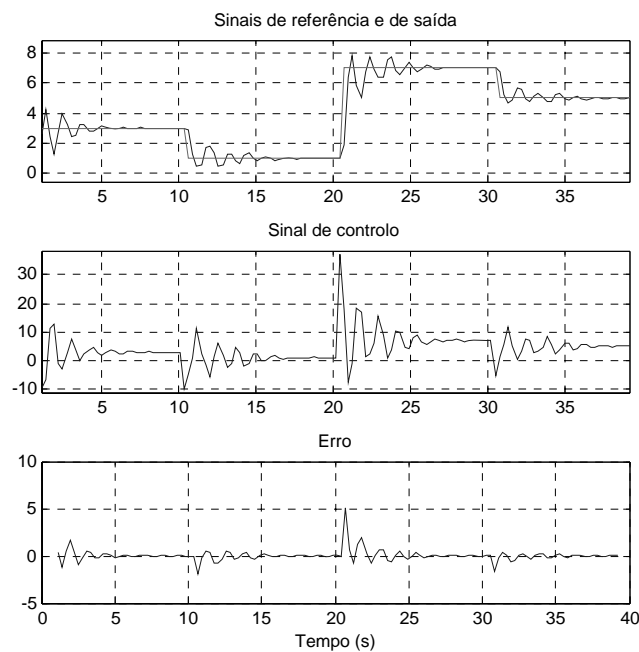


Figura 5.30: Sinais para o teste com o compensador neuronal, para o padrão de carga P3.

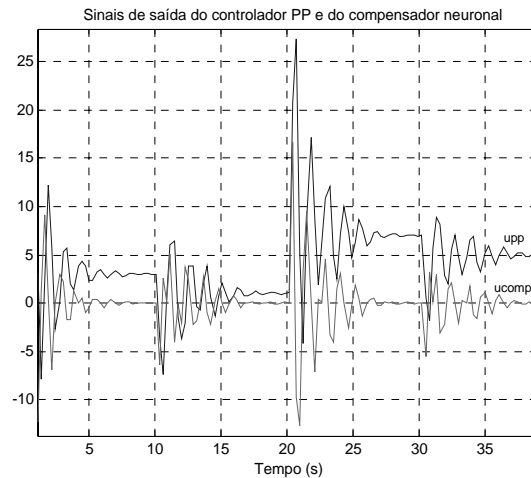


Figura 5.31: Contribuição do controlador PP e do compensador neuronal durante o ensaio com o padrão de carga P3.

No ensaio com o padrão de carga P3 o compensador tem uma intervenção determinante pois permite que o sinal de saída se aproxime do sinal de referência, embora apresente um tempo de estabilização longo, durante o qual a saída do sistema apresenta oscilações de amplitude decrescente. No entanto sem o compensador a saída do sistema apresenta fortes oscilações e não consegue seguir o sinal de referência de modo apropriado.

Os resultados mostram que o compensador neuronal permite obter melhor desempenho de controlo do que o compensador difuso, sendo por isso capaz de modelar melhor o efeito dos atrasos na malha de controlo. Para o teste com o padrão de carga P2 o compensador é capaz de prevenir o aparecimento de oscilação na saída e no caso do padrão de carga P3 as oscilações deixam de ser permanentes e passam a ser transitórias.

5.3 Comparação de desempenho entre as técnicas lineares e não-lineares

Comparando o desempenho de controlo obtido através da utilização do compensador de atraso, nas versões difusa e neuronal, com aquele que se obtém através da utilização de controladores distribuídos por posicionamento de pólos conclui-se que este último obtém melhores resultados nos casos em que o atraso introduzido é maior e produz por isso efeitos negativos mais significativos no desempenho de controlo. Neste caso o sinal de saída do sistema não apresenta oscilações, ao contrário do que acontece com o compensador de atraso que não é capaz de modelar por completo o efeito do atraso.

Quando o atraso introduzido na malha é pequeno o compensador de atraso é capaz de melhorar o desempenho de controlo, ao contrário do controlador distribuído por posicionamento de pólos que pelas suas características intrínsecas acaba por demorar

mais tempo a manifestar o seu efeito sobre o sinal de saída, fazendo com que o valor de ISE e de ITSE não seja melhor do que o da versão sem compensador.

A comparação entre os resultados obtidos com o compensador difuso e com o controlador adaptativo por posicionamento de pólos resultou na aceitação para publicação do artigo [145].

5.4 Conclusão

Neste capítulo apresentou-se o princípio do compensador de atraso, que consiste na introdução de um compensador numa malha cujo funcionamento se baseia num controlador que não tem em conta o efeito do atraso variável entre a amostragem e a actuação. O compensador é incluído na malha com o objectivo de minimizar o efeito dos atrasos variáveis na malha de controlo. O funcionamento do compensador de atraso baseia-se no conhecimento do atraso que afecta cada iteração da malha de controlo e por isso pode impôr restrições à implementação do sistema.

O atraso pode ser determinado através de medições efectuadas durante a operação do sistema, o que conduz a implementações mais flexíveis em que o tráfego no barramento pode ser alterado de acordo com as necessidades ao longo do tempo, ou através de cálculos efectuados antes do projecto do sistema baseados no conhecimento detalhado das políticas de escalonamento e das condições de operação. Esta opção conduz a sistemas mais rígidos em que todo o tráfego é previsto *a priori*. A medição do atraso durante a operação do sistema obriga a que o actuador seja implementado no mesmo nó do controlador de modo a que a medição seja correcta.

O compensador de atraso pode ser implementado recorrendo a diferentes técnicas ou tipos de modelos. Neste capítulo apresentaram-se implementações recorrendo a lógica difusa e a redes neuronais.

Os resultados obtidos mostram que ambas as implementações conseguem melhorar o desempenho de controlo, mas que a implementação difusa não é capaz de modelar de modo eficaz o efeito do atraso na malha, quando esse atraso é muito elevado. Por outro lado a implementação neuronal consegue melhores resultados, embora não seja capaz de eliminar por completo os efeitos do atraso nos casos em que esse atraso é grande.

Conclui-se por isso que, as implementações efectuadas permitiram melhorar o desempenho de controlo, embora nenhuma delas consiga modelar completamente o efeito do atraso na malha de controlo ao longo de toda a gama de valores possíveis para o atraso. A modelação é boa para atrasos mais pequenos e pior para valores de atraso maiores.

Finalmente apresentou-se uma comparação entre o desempenho obtido pelas técnicas de controlo adaptativo distribuído e o compensador de atraso. Para atrasos pequenos o compensador de atraso consegue melhores resultados, enquanto que para atrasos grandes o controlador distribuído por posicionamento de pólos supera o compensador de atraso.

Capítulo 6

Conclusões e trabalho futuro

Este capítulo finaliza este documento e nele se expõem as principais conclusões a retirar do trabalho efectuado. Após a apresentação das conclusões passam a enunciar-se as propostas de trabalho futuro.

6.1 Conclusões

Nesta tese são propostas técnicas de controlo lineares e não-lineares para sistemas de controlo distribuído em que o atraso entre a amostragem e a actuação varia de iteração para iteração. Os principais contributos são:

1. A utilização sistemática da modelação do tempo morto fracionário nos modelos discretos do sistema, que permite uma melhor identificação dos parâmetros da função discreta, e a sua utilização no projecto de controladores auto-ajustáveis, que conduz à obtenção de melhor desempenho de controlo quando o atraso introduzido produz efeitos nefastos sobre o sistema de controlo. Este tipo de modelação tem a vantagem de não necessitar da medição do atraso introduzido na malha e por isso proporciona implementações totalmente distribuídas do sistema de controlo.
2. A técnica de comutação de controladores auto-ajustáveis possibilita a adaptação do controlador a novas condições de operação durante o funcionamento do sistema. O critério utilizado para a comutação dos controladores pode ser interno ou externo à malha visando, respectivamente, a melhoria local do desempenho da malha ou a melhoria da qualidade de serviço do sistema distribuído.
3. A técnica de adaptação dinâmica do período de amostragem, que consiste na alteração do período de amostragem de uma dada malha de controlo. Esta técnica permite tanto a optimização do funcionamento do sistema distribuído global, por exemplo em caso de sobrecarga da rede, como a optimização do desempenho de uma malha de controlo local. Para a implementação desta técnica foram propostas três abordagens: estática, dinâmica e híbrida. A opção por cada uma

destas abordagens deve ter por base as características dos sistemas a que se destinam. A abordagem estática utiliza um conjunto de controladores auto-ajustáveis com parâmetros fixos que são guardados em memória. Esses parâmetros são calculados antecipadamente com base no conhecimento das condições de operação do sistema. Esta abordagem tem a vantagem de proporcionar uma comutação rápida entre controladores durante a operação do sistema e não requer muita capacidade de processamento. É necessário algum espaço de memória para guardar os parâmetros dos controladores. A abordagem estática apresenta contudo a desvantagem de ser pouco flexível e de necessitar do conhecimento das condições de operação do sistema para que o desempenho dos controladores seja bom. A abordagem dinâmica utiliza controladores auto-ajustáveis com parâmetros calculados durante a operação do sistema, com base na identificação dos parâmetros do modelo discreto do sistema. Nesta abordagem a convergência do estimador é mais demorada, quando há alteração do valor de h , e é necessária alguma capacidade de processamento para proceder à identificação do sistema e ao cálculo dos parâmetros do controlador. Ela tem a vantagem de ser muito flexível, uma vez que o controlador se vai adaptando às condições de operação que forem surgindo e de não ser necessária memória extra. Finalmente a abordagem híbrida permite aproveitar o melhor de cada uma das estratégias anteriores. Neste caso os parâmetros dos controladores auto-ajustáveis são também calculados durante a operação do sistema e a identificação dos parâmetros da função discreta é feita do mesmo modo, contudo a informação relativa ao estimador de cada controlador é mantida em memória de modo a tornar a convergência do estimador mais rápida.

4. O princípio da compensação do atraso propõe a inclusão na malha de controlo de um bloco de compensação com o objectivo de minimizar a degradação do desempenho de controlo introduzido pelos atrasos variáveis entre a amostragem e a actuação. Esta estratégia tem a vantagem de poder ser utilizada sobre um controlador já existente que não tem em conta os atrasos. Para a sua implementação é necessário o conhecimento do atraso entre a amostragem e a actuação em cada iteração. O atraso pode ser determinado de vários modos, dependendo do sistema em causa. Nos sistemas menos flexíveis, em que as condições de operação são bem conhecidas *a priori*, pode fazer-se uma análise antecipada dos atrasos induzidos na malha, tendo em conta a política de escalonamento utilizada e os tempos de acesso ao MAC e de processamento. Para uma abordagem mais genérica e flexível pode proceder-se à medição do atraso durante a operação do sistema. A estratégia da compensação do atraso pode ser implementada recorrendo a diferentes técnicas de modelação, como por exemplo lógica difusa ou redes neuronais.

As técnicas propostas foram validadas através da simulação de vários sistemas utilizando o simulador TrueTime e os resultados obtidos permitem, nestas condições, atestar a sua validade.

Conclui-se assim, em resposta à pergunta de investigação formulada no início deste documento, que é possível utilizar técnicas de controlo “clássicas” para controlo distri-

buído, desde que se utilizem técnicas de modelação que tenham em conta o efeito do atraso introduzido na malha de controlo pela implementação distribuída, sendo esta a tese defendida neste trabalho.

6.2 Trabalho futuro

No seguimento do trabalho efectuado propõem-se os seguintes desenvolvimentos:

1. Implementação da técnica de adaptação dinâmica da taxa de amostragem para optimização local das malhas de controlo. Esta aplicação não foi ainda implementada, pelo que é importante proceder à sua validação. Os resultados preliminares mostram o interesse da sua aplicação mas apontam para a necessidade de um estudo mais aprofundado do impacto da alteração de h e de R_{ha} sobre o desempenho de controlo nos controladores adaptativos distribuídos por posicionamento de pólos.
2. Estudo das condições de comutação entre controladores decorrentes da utilização da técnica de adaptação dinâmica da taxa de amostragem.
3. Implementação da técnica de adaptação dinâmica da taxa de amostragem num sistema de teste utilizando componentes reais de modo a evidenciar o seu interesse prático e a sua aplicabilidade. O sistema de teste será baseado em micro-controladores PIC com controlador CAN integrado para facilitar a gestão das comunicações através do barramento.
4. Comparação das necessidades de *hardware* de cada uma das abordagens propostas para a técnica de adaptação dinâmica da taxa de amostragem.
5. Desenvolvimento de modelos para o compensador neuronal para sistemas de 2ª ordem.
6. Implementação do compensador de atraso utilizando técnicas neuro-difusas¹.

¹O artigo "A neuro-fuzzy delay compensator for distributed control systems",

Ana Antunes, Fernando Dias, José Vieira, Alexandre Mota, com os primeiros resultados da aplicação desta técnica, foi recentemente aceite para publicação na conferência 13th IEEE International Conference on Emerging Technologies and Factory Automation.

Bibliografia

- [1] K. Arzen, A. Cervin, and D. Henriksson, “Implementation-aware embedded control systems,” *Handbook of networked and embedded control systems*, 2005.
- [2] “CAN specification version 2.0,” Tech. Rep., Robert Bosch GmbH, Stuttgart, Germany, 1991.
- [3] Feng-Li Lian, *Analysis, Design, Modeling, and Control of Networked Control Systems*, Ph.D. thesis, University of Michigan, 2001.
- [4] P.M. Colom, *Analysis and Design of Real-Time Control Systems with Varying Control Timing Constraints*, Ph.D. thesis, Universitat Politècnica de Catalunya, 2002.
- [5] A. Cervin, *Integrated Control and Real-Time Scheduling*, Ph.D. thesis, Lund Institute of Technology, 2003.
- [6] L.-S. Hu, T. Bai, P. Shi, and Z. Wu, “Sampled-data control of networked linear control systems,” *Automatica*, vol. 43, pp. 903–911, 2007.
- [7] K. Ji and W. Kim, “Real-time control of networked control systems via ethernet,” *International Journal of Control, Automation, and Systems*, vol. 3, pp. 591–600, 2005.
- [8] Y. Jianyong, Y. Shimin, and W. Haiqing, “Survey on the performance analysis of networked control systems,” *Proc. of the 2004 International Conference on Systems, Man and Cybernetics*, pp. 5068–5073, 2004.
- [9] J. Ferreira, P. Pedreiras, L. Almeida, and J.A. Fonseca, “The FTT-CAN protocol for flexibility in safety-critical systems,” *IEEE Micro*, pp. 46–55, 2002.
- [10] A. Ray, “Network access protocols for real-time distributed control systems,” *IEEE Transactions on industry applications*, vol. 24, pp. 897–904, 1988.
- [11] Y.-C. Tian and D. Levy, “Compensation for control packet dropout in networked control systems,” *Information Sciences*, vol. 178, pp. 1263–1278, 2008.
- [12] K. Ji, W.-J. Kim, and A. Ambike, “Control strategies for distributed real-time control with time delays and packets losses,” *Proc. of the ASME International Mechanical Engineering Congress and Exposition*, 2004.

- [13] Q. Ling and M.D. Lemmon, "Soft real-time scheduling of networked control systems with dropouts governed by a markov chain," *Proc. of the American Control Conference*, vol. 6, pp. 4845–4850, 2003.
- [14] W. Zang, M.S. Branicky, and S.M. Philips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, pp. 84–99, 2001.
- [15] J. Nilsson, *Real-Time Control Systems with Delays*, Ph.D. thesis, Dept. of Automatic Control Lund, 1998.
- [16] Y. Halevi and A. Ray, "Integrated communication and control systems: Part i - analysis," *ASME Journal of dynamic system measurements and control*, vol. 110, pp. 367–373, 1988.
- [17] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *IFAC Control engineering practice*, vol. 11, pp. 1099–1111, 2003.
- [18] J. W. S. Liu, *Real-Time Systems*, Prentice Hall, 2000.
- [19] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
- [20] G. Buttazzo, G. Lipari, and L. Abeni, "Elastic task model for adaptive rate control," *Proc. of the 19th IEEE Real-Time Systems Symposium*, pp. 286–295, 1998.
- [21] G. Buttazzo and L. Abeni, "Adaptive rate control through elastic scheduling," *Proc. of the 39th IEEE Conference on Decision and Control*, pp. 4883–4888, 2000.
- [22] G. Buttazzo and L. Abeni, "Adaptive workload management through elastic scheduling," *Real-time systems*, vol. 23, pp. 7–24, 2002.
- [23] G. C. Buttazzo, *Hard Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications*, Kluwer Academic Publishers, 1997.
- [24] A. Mohammadi and S.G. Akl, "Scheduling algorithms for real-time systems," Tech. Rep. 2005-499, School of Computing Queen's University, 2005.
- [25] C.D. Locke, "Software architecture for hard real-time applications: Cyclic vs. fixed priority executives," *Real-Time Systems*, vol. 4, pp. 37–57, 1992.
- [26] G. Fohler, "Joint scheduling of distributed complex periodic and hard aperiodic tasks in statically scheduled systems," *IEEE Real-Time Systems Symposium*, 1995.
- [27] C. J. Lui and J. W. Layland, "Scheduling algorithms for multiprocessing in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

- [28] J. Lehoczky, L. Sha, and J. Strosnider, “Enhanced aperiodic responsiveness in hard-real time environment,” *Proc. of the 8th IEEE Real-Time Systems Symposium*, 1987.
- [29] B. Sprunt, L. Sha, and J. Lehoczky, “Aperiodic task scheduling for hard real-time systems,” *Real-time systems*, 1989.
- [30] L. Abeni and Giorgio Buttazzo, “Integrating multimedia applications in hard real-time systems,” *Proc. of IEEE Real-Time System Symposium*, pp. 4–13, 1998.
- [31] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Kluwer Academic Publishers, 1997.
- [32] J.Y.T. Leung and J. Whitehead, “On the complexity of fixed-priority scheduling of periodic, real-time tasks,” *Performance Evaluation*, vol. 2:4, pp. 237–250, 1982.
- [33] G.C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, Springer, 2004.
- [34] M. Joseph and P. Pandya, “Finding response times in a real-time system,” *The computer journal*, vol. 29:5, pp. 390–395, 1986.
- [35] O. Redell and M. Sanfridson, “Exact best-case response time analysis of fixed priority scheduled tasks,” *Proc. of the 14th Euromicro Conference on Real-Time systems*, 2002.
- [36] G.C. Buttazzo, “Rate monotonic vs. EDF: Judgment day,” *Real-Time Systems*, vol. 29, pp. 5–26, 2005.
- [37] A. Burns and A. Wellings, *Real-Time Systems and Programming Languages*, Addison-Wesley, 3rd edition, 2001.
- [38] J.A. Stankovic, M. Spuri, K.Ramamritham, and G.C. Buttazzo, *Deadline Scheduling for Real-Time Systems - EDF and Related Algorithms*, Kluwer Academic Publishers, 1998.
- [39] K. Alberts and F. Slomka, “Efficient feasibility analysis for real-time systems with EDF scheduling,” *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 492–497, 2005.
- [40] L. Abeni and G.C. Buttazzo, “Resource reservation in dynamic real-time systems,” *Real-time systems*, vol. 27, pp. 123–167, 2004.
- [41] G. Buttazzo and E. Bini, “Optimal dimensioning of a constant bandwidth server,” *Proc. of the 27th IEEE International Real-Time Systems Symposium*, pp. 169–177, 2006.

- [42] H. Schumny, “Fieldbuses in measurement and control,” *Computer Standards and Interfaces*, vol. 19, pp. 295–304, 1998.
- [43] R. Patzke, “Fieldbus basics,” *Computer Standards and Interfaces*, vol. 19, pp. 275–293, 1998.
- [44] L. Almeida, *Flexibility and Timeliness in Fieldbus-Based Real-Time Systems*, Ph.D. thesis, Dept. de Electrónica e Telecomunicações, Universidade de Aveiro, 1999.
- [45] P. Pedreiras, *Supporting Flexible Real-Time Communication on Distributed Systems*, Ph.D. thesis, Dept. de Electrónica e Telecomunicações, Universidade de Aveiro, 2003.
- [46] J. Tsai, Y. Bi, S. Yang, and R. Smith, *Distributed Real-Time Systems: Monitoring, Visualization, Debugging and Analysis*, John Wiley and Sons, 1996.
- [47] W. Lawrenz, *CAN System Engineering: From Theory to Practical Applications*, Springer-Verlag, 1997.
- [48] H. Kopetz, “Should responsive systems be event-triggered or time-triggered?,” *IEICE Transactions on information and systems*, vol. 11, pp. 1325–1332, 1993.
- [49] L. Almeida, P. Pedreiras, and J.A. Fonseca, “The FTT-CAN protocol: Why and how,” *IEEE Transactions on Industrial Electronics*, vol. 49, pp. 1189–1201, 2002.
- [50] P. Veríssimo and L. Rodrigues, *Distributed Systems for System Architects*, Kluwer Academic Publishers, 2001.
- [51] J.-P. Thomesse, “Time and industrial local area networks,” *Pro. of COMPEURO’93*, 1993.
- [52] J.P. Thomesse, “A review of the fieldbuses,” *Annual Reviews in Control*, vol. 22, pp. 35–45, 1998.
- [53] K. Tindell and A. Burns, “Guaranteed message latencies for distributed safety-critical hard real-time control networks,” Tech. Rep. YCS 94-229, Dept. Computer Science, University of York, 1994.
- [54] K. Tindell, A. Burns, and J. Wellings, “Calculating controller area network message response times,” *IFAC Control Engineering Practice*, vol. 3, pp. 1163–1168, 1995.
- [55] “European standard EN 50170: Fieldbus: Vol. 1: P-net; vol. 2: PROFIBUS; vol. 3: WorldFIP,” Tech. Rep. EN50170, CENELEC, European Committee for Electrotechnical Standardisation, 1996.

- [56] L. Sha and S. Sathaye, "A systematic approach to designing distributed real-time systems," *Computer*, pp. 68–78, 1993.
- [57] S. Kolla, D. Border, and E. Mayer, "Fieldbus networks for control system implementations," *Proc. of the Electrical Insulation Conference and Electrical Manufacturing and Coil Winding Technology Conference*, pp. 493–498, 2003.
- [58] A.V. Scott and W.J. Buchanan, "Truly distributed control systems using fieldbus technology," *Proc. of the 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, pp. 165–173, 2000.
- [59] G. Schickhuber and O. McCarthy, "Distributed fieldbus and control network systems," *Computing and Control Engineering Journal*, pp. 21–32, 1997.
- [60] "Road vehicles- interchange of digital information- controller area network (CAN) for high-speed communication," Tech. Rep. ISO-11898, ISO, 1993.
- [61] "Road vehicles - low-speed serial data communication - part 2: Low-speed controller area network (CAN)," Tech. Rep. ISO-11519-2, ISO, 1994.
- [62] J. Rufino, P. Veríssimo, G. Arroz, C. Almeida, and L. Rodrigues, "Fault-tolerant broadcast in CAN.," *Digest of papers, 29th International Symposium on Fault Tolerant Computer Systems*, pp. 150–159, 1998.
- [63] J. Ferreira, *Fault-Tolerance in Flexible Real-Time Communication Systems*, Ph.D. thesis, Departamento de Electrónica e Telecomunicações da Universidade de Aveiro., 2005.
- [64] J. Rufino, N. Pedroso, J. Monteiro, P. Veríssimo, and G. Arroz, "Hardware support for CAN fault-tolerant communication," *Proc. of the 5th IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 263–266, 1998.
- [65] "IEC international standard 61158: Fieldbus standard for use in industrial control systems- type 1: Existing IEC TS61158 parts3-6 (foundation fieldbus h1); type 2: ControlNet; type 3: PROFIBUS; type 4: P-net; type 5: Fieldbus foundation HSE; type 6: SwiftNet; type 7: WorldFIP; type 8: Interbus-s," Tech. Rep. IEC61158, IEC, International Electrotechnical Committee, 2000.
- [66] J. Weigmann and G. Kilian, *Decentralization with PROFIBUS-DP*, Siemens, 2000.
- [67] J.A. Fonseca, J. Ferreira, M. Calha, P. Pedreiras, and L. Almeida, "Issues on task dispatching and master replication in FTT-CAN," *IEEE AFRICON 6th Africon Conference in Africa*, vol. 1, pp. 221–226, 2002.
- [68] P. Pedreiras and L. Almeida, "The flexible time-triggered (FTT) paradigm: An approach to QoS management in distributed real-time systems," *Proc. of the IEEE International Parallel and distributed processing symposium*, vol. 22-26, 2003.

- [69] L.G. Bushnell, "Guest editorial: Networks and control," *IEEE Control System Magazine*, vol. 21, pp. 22–23, 2001.
- [70] P. Antsaklis and J. Baillieul, "Guest editorial: Special issue on networked control systems," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1421–1423, 2004.
- [71] P. Antsaklis and J. Baillieul, "Special issue on technology of networked control systems," *Proceedings of the IEEE*, vol. 95, pp. 5–8, 2007.
- [72] Y. Halevi and A. Ray, "Integrated communication and control systems: Part II - design considerations," *ASME Journal of Dynamic Systems, measurement and control*, vol. 110, pp. 370–377, 1988.
- [73] B. Wittenmark, J. Nilsson, and M. Thorgren, "Timing problems in real-time control systems," *Proc. of the American Control Conference*, 1995.
- [74] M. Thorgren, "Fundamentals of implementing real-time control applications in distributed computer systems," *Journal of real-time systems*, vol. 14, pp. 219–250, 1998.
- [75] M. Sanfridson, "Timing problems in distributed real-time computer control systems," Tech. Rep. ISSN 1400-1179, Royal Institute of Technology, 2000.
- [76] A. Stothert and I.M. McLeod, "Effect of timing jitter on distributed computer control system performance," *Proc. of the 15th IFAC Workshop on distributed computer control systems*, 1998.
- [77] N. Andreff, "Robustness to jitter in real-time systems," Tech. Rep. ISRN LUTFD2/TFRT-5507-SE, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1994.
- [78] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, "How does control timing affect performance?," *IEEE Control Systems magazine*, vol. 23, pp. 16–30, 2003.
- [79] D. Henriksson, A. Cervin, and K.E. Arzen, "TrueTime: Real-time control system simulation using MATLAB/Simulink," *Proc. of the Nordic MATLAB Conference*, 2003.
- [80] A. Cervin, "Using jitterbug to derive control loop timing requirements," *Proc. of CERTS'03- Co-design of embedded real-time systems workshop*, 2003.
- [81] G.P. Liu, D. Rees, and S.C. Chai, "Design and practical implementation of networked control systems," *Proc. of the IEEE Networking, sensing and control*, pp. 336–341, 2005.

- [82] R. Luck, *Observability and Delay Compensation of Integrated Communication and Control Systems*, Ph.D. thesis, Dept. of mechanical engineering, Pensilvania State University, USA., 1989.
- [83] R. Luck and A. Ray, “An observer-based compensator for distributed delays,” *Automatica*, vol. 26, pp. 903–908, 1990.
- [84] R. Luck and A. Ray, “Experimental verification of a delay compensation algorithm for integrated communication and control systems,” *International Journal of Control*, vol. 59, pp. 1357–1372, 1994.
- [85] M. Thorngren, *Modelling and Design of Distributed Real-Time Control Applications*, Ph.D. thesis, Dept. of Machine Design, The Royal Institute of Technology, Stockholm, Sweden, 1995.
- [86] K. Shin and X. Cui, “Computing time delay and its effects on real-time control systems,” *IEEE Transactions on control systems technology*, vol. 3, pp. 218–224, 1995.
- [87] F. Goktas, *Distributed Control of Systems over Communication Networks*, Ph.D. thesis, University of Pensilvania, U.S.A., 2000.
- [88] N.B. Almutairi, M.-Y. Chow, and Y. Tipsuwan, “Networked-based controlled DC motor with fuzzy compensation,” *Proc. of the 27th annual conference of the IEEE Industrial Electronics Society*, vol. 3, pp. 1844–1849, 2001.
- [89] K.C. Lee, S. Lee, and M.H. Lee, “Remote fuzzy logic control of networked control system via profibus-DP,” *IEEE Transactions on Industrial Electronics*, vol. 50, pp. 784–792, 2003.
- [90] S.H. Hong, “Scheduling algorithm of data sampling times in the integrated communication and control systems,” *IEEE Transactions on Control Systems Technology*, vol. 3, pp. 225–230, 1995.
- [91] D. Seto, J.P. Lehoczky, L. Sha, and K.G. Shin, “On task schedulability in real-time control systems,” *Proc. of the IEEE Real-time systems symposium*, pp. 13–21, 1996.
- [92] K.G. Shin and C.L. Meissner, “Adaptation and graceful degradation of control system performance by task reallocation and period adjustment,” *Proc. of the 11th Euromicro Conference on real-time systems*, pp. 29–36, 1999.
- [93] G. Buttazzo, M. Velasco, P. Marti, and G. Fohler, “Managing quality-of-control performance under overload conditions,” *Proc. of the IEEE Euromicro Conference on Real-time systems*, pp. 53–60, 2004.
- [94] D. Henriksson and A. Cervin, “Optimal on-line sampling period assignment of real-time control tasks based on plant state information,” *Proc. of the 44th*

- IEEE Conference on Decision and Control and European Control Conference*, pp. 4469–4472, 2005.
- [95] R. Castane, P. Marti, M. Velasco, A. Cervin, and D. Henriksson, “Resource management for control tasks based on the transient dynamics of closed-loop systems,” *Proc. of the 18th Euromicro conference on real-time systems*, 2006.
- [96] K. J. Astrom and B. Wittenmark, *Adaptive Control, 2nd Edition*, Addison-Wesley, 1995.
- [97] P.E. Wellstead and M.B. Zarrop, *Self-Tuning Systems: Control and Signal Processing*, John Wiley and Sons, 1991.
- [98] K. J. Astrom and B. Wittenmark, *Computer Controlled Systems: Theory and Design, 3rd Edition*, Prentice Hall, 1997.
- [99] A. M. Mota, *Identificação de Sistemas*, DETUA, 2000.
- [100] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, 1987.
- [101] C.H. Houpis and G.B. Lamont, *Digital Control Systems: Theory, Hardware, Software*, McGraw-Hill Int. Editions, 1992.
- [102] K. Ogata, *Digital Control Systems, 2nd Edition*, Saunders College Publishing, 1992.
- [103] U. Holmberg, *Digital Control*, 2003.
- [104] K. Mondgalya, *Digital Control*, John Wiley and Sons, 2007.
- [105] L. Ljung and T. Glad, *Modeling of Dynamic Systems*, Prentice Hall, 1994.
- [106] B. Widrow and E. Walach, *Adaptive Inverse Control*, Prentice Hall, 1996.
- [107] Tore Hägglund, *New Estimation Techniques for Adaptive Control*, Ph.D. thesis, Lund Institute of Technology, 1983.
- [108] Urban Forssell and Lennart Ljung, “Closed-loop identification revisited,” *Automatica*, vol. 35, pp. 1215–1241, 1999.
- [109] R. Vinter, *Optimal Control (Systems and Control)*, Birkhauser Verlag AG, 2000.
- [110] Edited by V. J. VanDoren, *Techniques for Adaptive Control*, Butterworth-Heinemann, 2003.
- [111] C. Mothadi D.W. Clarke and P.S. Tuffs, “Generalized predictive control. part I- the basic algorithm,” *Automatica*, vol. 23, pp. 137–148, 1987.
- [112] C. Mothadi D. W. Clarke and P.S. Tuffs, “Generalized predictive control. part II- extensions and interpretations,” *Automatica*, vol. 23, pp. 149–160, 1987.

- [113] D.W. Clarke and C. Mothadi, "Properties of generalized predictive control," *Automatica*, vol. 25, pp. 859–875, 1989.
- [114] E. F. Camacho, *Model Predictive Control*, Springer-Verlag, 1999.
- [115] D. Soloway, J. Shi, and A. Kelkar, "GPC-based stable reconfigurable control," *Proc. of ASME Int. Mechanical Engineering Congress*, 2003.
- [116] J.M. Lemos, "Structure selection in predictive adaptive control: A maximum entropy interpretation," *Proc. 10th Mediterranean Conference on Control and Automation*, 2002.
- [117] G. Menga and E. Mosca, *MUSMAR: Multivariable Adaptive Regulators Based on Multistep Cost Functionals*. In *Advances in Control*, D.G. Lainiotis and N.S. Tzannes (eds). D. Reidel, Dordrecht, 1980.
- [118] E. Mosca and G. Zappa, "MUSMAR: Basic convergence and consistency properties," *Lecture notes in control*, vol. 28, pp. 189–199, 1980.
- [119] L. Ljung K.J. Astrom, U. Borisson and B. Wittenmark, "Theory and applications of self-tuning regulators," *Automatica*, vol. 13, pp. 457–476, 1977.
- [120] M.J. Grimble, "Implicit and explicit LQG self-tuning controllers," *Automatica*, vol. 20, pp. 661–669, 1984.
- [121] P.P. Kanjilal D.W. Clarke and C. Mohtadi, "A generalized LQG approach to self-tuning control. part i- aspects of design.," *Int. Journal of Control*, vol. 41, pp. 1509–1523, 1985.
- [122] P.P. Kanjilal D.W. Clarke and C. Mothadi, "A generalized LQG approach to self-tuning control. part II- implementation and simulation.," *Int. Journal of control*, vol. 41, pp. 1525–1544, 1985.
- [123] A. M. Mota, *Controlo Adapativo*, DETUA, 1997.
- [124] A. Antunes and A.M. Mota, "Control performance of a real-time adaptive distributed control system under jitter conditions," *Proc. of the Control 2004 Conference*, 2004.
- [125] A. Antunes and F.M. Dias, "Influence of the sampling period in the performance of a real-time distributed system under jitter conditions," *WSEAS Transactions on communications*, vol. 3, pp. 248–253, 2004.
- [126] A.M. Mota and J.A. Fonseca, "Dealing with jitter in system modeling and identification," *Proc. of the Control 2000 Conference*, 2000.
- [127] P. Fonseca A.M. Mota and J.A. Fonseca, "Systems modelling and identification in CAN based distributed control systems," *Proc. of the IFAC Distributed Computer Control Systems*, 2000.

- [128] A. Antunes, F.M. Dias, and A. Mota, “Fractional dead-time modelling for variable sampling to actuation delay compensation in distributed real-time control systems,” *International Transactions on Computer Science and Engineering, GESTS*, vol. 24, pp. 9–18, 2005.
- [129] A. Antunes, F.M. Dias, and A. Mota, “Improving the control performance of a real-time control system under variable sampling to actuation delay,” *Proc. of the 25th International Conference on Modelling, Identification and Control*, 2006.
- [130] A. Antunes, P. Pedreiras, and A. Mota, “Adapting the sampling period of a real-time adaptive distributed controller to the bus load,” *Proc. of the 10th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2005.
- [131] A. Antunes, P. Pedreiras, L. Almeida, and A. Mota, “Dynamic rate adaptation in distributed computer control systems,” *Proc. of the ANIPLA 2006- Methodologies for Emerging Technologies in Automation*, 2006.
- [132] A. Antunes, P. Pedreiras, L. Almeida, and A. Mota, “Improving operational flexibility in distributed control systems: The dynamic rate adaptation technique,” *Automazione e Strumentazione- Ellectronica Industriale*, vol. LV-2, 2007.
- [133] A. Antunes, P. Pedreiras, L. Almeida, and A. Mota, “Dynamic rate and control adaptation in networked control systems,” *Proc. of the 5th IEEE International Conference on Industrial Informatics*, vol. 2, pp. 841–846, 2007.
- [134] A. Antunes, P. Pedreiras, L. Almeida, and A. Mota, “The dynamic rate adaptation technique for distributed computer control systems,” *Modelling and control of discrete production systems, ISA/O3NEIDA*, 2008.
- [135] A. M. Mota, *Controlo Adaptativo: Um Caso Concreto*, Ph.D. thesis, Universidade de Aveiro, 1992.
- [136] P. Fonseca, *Modelisation et Validation Des Algorithmes Non-Deterministes de Synchronisation Des Horloges*, Ph.D. thesis, Institut National Polytechnique de Lorraine e Universidade de Aveiro, 1999.
- [137] O. Sorensen, *Neural Networks in Control Applications*, Ph.D. thesis, Dept. of control engineering, Institute of electronic systems, Aalborg university, Denmark., 1994.
- [138] F.M. Dias, *Técnicas de Controlo Não-Linear Baseadas Em Redes Neurais: Do Algoritmo À Implementação*, Ph.D. thesis, Dept. de Electrónica e Telecomunicações, Universidade de Aveiro, Portugal, 2005.
- [139] A. Antunes, F.M. Dias, J. Vieira, and A. Mota, “Delay compensator: An approach to reduce the variable sampling to actuation delay effect in distributed real-time control systems,” *Proc. of the 11th IEEE Conference on Emerging Technologies and Factory Automation*, pp. 531–536, 2006.

- [140] A. Antunes, F.M. Dias, and A. Mota, “A neural model for delay correction in a distributed control system,” *Proc. of the 18th International Conference on Artificial Neural Networks*, 2008.
- [141] A. Antunes, F. M. Dias, and A. Mota, “A neural network delay compensator for networked control systems,” *Proc. of the 11th IEEE International Conference on Emerging Technologies and Factory Automation*, 2008.
- [142] E.H. Mamdani and S. Assilian, “An experimental in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, vol. 7, pp. 1–13, 1975.
- [143] K. Levenberg, “A method for the solution of certain problems in least squares,” *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.
- [144] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *SIAM Journal on applied mathematics*, vol. 11, pp. 431–441, 1963.
- [145] A. Antunes, F. M. Dias, and A. Mota, “Comparing the delay compensator approach with fractional dead-time modeling in distributed real-time control systems,” *Proc. of the 8th Portuguese Conference on Automatic Control- CONTROLLO 2008*, 2008.
- [146] D. Henriksson and A. Cervin, “TrueTime 1.13- reference manual,” Tech. Rep., Department of Automatic Control, Lund Institute of Technology, 2003.

Anexo A - Dedução dos parâmetros do regulador

Este anexo apresenta a dedução dos parâmetros de controladores por posicionamento de pólos para sistemas de 1ª e de 2ª ordem, com e sem modelação do tempo morto fraccionário, utilizando o método de resolução directa da equação de Diophantine.

Posicionamento de pólos sem modelação do tempo morto fraccionário

Sistemas de 1ª ordem Considera-se que a função de transferência discreta do sistema de 1ª ordem é dada por

$$G(q) = \frac{b}{q + a}$$

ou seja os polinómios $A(q)$ e $B(q)$ são

$$\begin{aligned} A(q) &= q + a \\ B(q) &= b \end{aligned}$$

O polinómio que define o comportamento do sistema em malha fechada é igual a

$$A_m(q) = q + a_m$$

O polinómio observador é dado por

$$A_o(q) = q + a_o$$

A equação de Diophantine é dada por

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= q - 1 \\ S(q) &= s_0q + s_1 \end{aligned}$$

Substituindo todos os polinômios na equação de Diophantine obtém-se:

$$(q + a_m)(q + a_o) = (q + a)(q - 1) + b(s_0q + s_1)$$

Desenvolvendo e agrupando os coeficientes das potências de igual valor

$$q^2 + (a_o + a_m)q + a_o a_m = q^2 + (a - 1 + bs_0)q + (bs_1 - a)$$

Igualando os coeficientes da mesma ordem obtém-se o sistema de duas equações a duas incógnitas

$$\begin{cases} a_o + a_m = a - 1 + bs_0 \\ a_o a_m = bs_1 - a \end{cases}$$

de onde se retiram as expressões para os parâmetros s_0 e s_1

$$\begin{cases} s_0 = \frac{1-a+a_m+a_o}{b} \\ s_1 = \frac{a_m a_o - a}{b} \end{cases}$$

Os polinômios $R(q)$ e $S(q)$ são dados por

$$\begin{aligned} R(q) &= q - 1 \\ S(q) &= \frac{1 - a + a_m + a_o}{b}q + \frac{a_m a_o - a}{b} \end{aligned}$$

O polinômio $T(q)$ é dado pela expressão

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

Para a determinação deste polinômio é necessário fazer considerações adicionais.

Como $B(q)$ é uma constante, não existe nenhuma limitação para a escolha do polinômio $B_m(q)$. Considere-se:

$$B_m(q) = b_m$$

O valor de b_m é normalmente escolhido de modo a que o modelo de malha fechada tenha ganho estático unitário, ou seja:

$$\lim_{q \rightarrow 1} \frac{B_m(q)}{A_m(q)} = 1$$

assim, obtém-se para $B_m(q)$

$$B_m(q) = b_m = 1 + a_m$$

Substituindo na expressão de $T(q)$ obtém-se:

$$T(q) = t_0q + t_1 = \frac{b_m}{b}q + \frac{b_m a_o}{b}$$

A função de controlo é dada por:

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

Substituindo os polinómios obtidos anteriormente na função de controlo obtém-se:

$$(q - 1)u(k) = (t_0q + t_1)u_c(k) - (s_0q + s_1)y(k)$$

multiplicando ambos os lados da equação por q^{-1} resulta:

$$q^{-1}(q - 1)u(k) = q^{-1}(t_0q + t_1)u_c(k) - q^{-1}(s_0q + s_1)y(k)$$

ou seja:

$$(1 - q^{-1})u(k) = (t_0 + t_1q^{-1})u_c(k) - (s_0 + s_1q^{-1})y(k)$$

desenvolvendo obtém-se:

$$u(k) - u(k - 1) = t_0u_c(k) + t_1u_c(k - 1) - s_0y(k) - s_1y(k - 1)$$

de onde resulta finalmente a expressão de $u(k)$

$$u(k) = u(k - 1) + t_0u_c(k) + t_1u_c(k - 1) - s_0y(k) - s_1y(k - 1)$$

onde

$$\begin{cases} t_0 = \frac{b_m}{b} \\ b_m = 1 + a_m \\ t_1 = \frac{b_m a_o}{b} \\ s_0 = \frac{1 - a + a_m + a_o}{b} \\ s_1 = \frac{a_m a_o - a}{b} \end{cases}$$

Os parâmetros a e b são substituídos pelos valores estimados em tempo-real \hat{a} e \hat{b} .

Sistemas de 2ª ordem Considera-se que a função de transferência discreta do sistema de 2ª ordem é dada por

$$G(q) = \frac{b_1q + b_2}{q^2 + a_1q + a_2}$$

ou seja os polinómios $A(q)$ e $B(q)$ são

$$\begin{aligned} A(q) &= q^2 + a_1q + a_2 \\ B(q) &= b_1q + b_2 \end{aligned}$$

O polinómio que define o comportamento do sistema em malha fechada é igual a:

$$A_m(q) = q^2 + a_{m1}q + a_{m2}$$

O polinómio observador é dado por:

$$A_o(q) = q^2 + a_{o1}q + a_{o2}$$

A equação de Diophantine é dada por

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= (q-1)(q+r_1) = q^2 + (r_1-1)q - r_1 \\ S(q) &= s_0q^2 + s_1q + s_2 \end{aligned}$$

Substituindo todos os polinómios na equação de Diophantine obtém-se:

$$(q^2 + a_{m1}q + a_{m2})(q^2 + a_{o1}q + a_{o2}) = (q^2 + a_1q + a_2)(q^2 + (r_1-1)q - r_1) + (b_1q + b_2)(s_0q^2 + s_1q + s_2)$$

desenvolvendo e agrupando os coeficientes das potências de igual valor

$$\begin{aligned} q^4 + (a_{o1} + a_{m1})q^3 + (a_{o2} + a_{m1}a_{o1} + a_{m2})q^2 + (a_{m1}a_{o2} + a_{m2}a_{o1})q + (a_{m2}a_{o2}) = \\ = q^4 + ((r_1-1) + a_1 + b_1s_0)q^3 + (a_1(r_1-1) - r_1 + a_2 + b_1s_1 + b_2s_0)q^2 + \dots \\ (a_2(r_1-1) - a_1r_1 + b_1s_2 + b_2s_1)q + (b_2s_2 - a_2r_1) \end{aligned}$$

igualando os coeficientes da mesma ordem obtém-se o sistema de quatro equações a quatro incógnitas:

$$\begin{cases} a_{o1} + a_{m1} = (r_1 - 1) + a_1 + b_1s_0 \\ a_{o2} + a_{m1}a_{o1} + a_{m2} = a_1(r_1 - 1) - r_1 + a_2 + b_1s_1 + b_2s_0 \\ a_{m1}a_{o2} + a_{m2}a_{o1} = a_2(r_1 - 1) - a_1r_1 + b_1s_2 + b_2s_1 \\ a_{m2}a_{o2} = b_2s_2 - a_2r_1 \end{cases}$$

Resolvendo a 1ª equação em ordem a r_1 e substituindo na 2ª, na 3ª e na 4ª equações obtém-se:

$$\begin{cases} r_1 = a_{o1} + a_{m1} - a_1 + 1 - b_1s_0 \\ (b_2 - a_1b_1 + b_1)s_0 = a_2 + a_1P_0 - a_{m1} - a_{m2} - a_{o2} - a_{o1}(1 + a_{m1}) - 1 - b_1s_1 \\ a_2b_1s_0 - b_1s_2 - b_2s_1 - a_1b_1s_0 = a_2P_0 - a_2 - a_1P_0 - a_{m1}a_{o2} - a_{m2}a_{o1} \\ b_2s_2 - a_2P_0 + a_2b_1s_0 = a_{m2}a_{o2} \end{cases}$$

substituindo s_0 na 3ª equação obtém-se:

$$\begin{cases} r_1 = P_0 - b_1 s_0 \\ s_0 = -\frac{P_1 + b_1 s_1}{P_2} \\ P_3 = -a_2 b_1 \left(\frac{P_1 + b_1 s_1}{P_2}\right) - b_1 s_2 - b_2 s_1 + a_1 b_1 \left(\frac{P_1 + b_1 s_1}{P_2}\right) \\ - \end{cases}$$

com

$$\begin{cases} P_0 = a_{o1} + a_{m1} - a_1 + 1 \\ P_1 = a_2 + a_1 P_0 - a_{m1} - a_{m2} - a_{o2} - a_{o1}(1 + a_{m1}) - 1 \\ P_2 = b_2 + (1 - a_1)b_1 \\ P_3 = a_2(P_0 - 1) - a_1 P_0 - a_{m1} a_{o2} - a_{m2} a_{o1} \end{cases}$$

desenvolvendo

$$\begin{cases} - \\ - \\ b_1 P_2 s_2 = -P_3 P_2 + P_1(a_1 b_1 - a_2 b_1) + s_1(a_1 b_1^2 - P_2 b_2 - a_2 b_1^2) \\ - \end{cases}$$

resolvendo em ordem a s_2 e substituindo s_2 na 4ª equação obtém-se

$$\begin{cases} - \\ - \\ s_2 = \frac{P_3 + s_1(P_4 b_1 - P_2 b_2)}{b_1 P_2} \\ b_2 \left(\frac{P_3 + s_1(P_4 b_1 - P_2 b_2)}{b_1 P_2}\right) - a_2 P_0 - a_2 b_1 \left(\frac{P_1 + b_1 s_1}{P_2}\right) = a_{m2} a_{o2} \end{cases}$$

com

$$\begin{cases} P_4 = b_1(a_1 - a_2) \\ P_5 = -P_3 P_2 + P_1 P_4 \end{cases}$$

desenvolvendo obtém-se:

$$\begin{cases} - \\ - \\ - \\ P_2 b_1(-a_2 P_0 - a_{m2} a_{o2}) = -b_2 P_5 - b_2(P_4 b_1 - P_2 b_2)s_1 + a_1 b_1^2 P_1 + a_2 b_1^3 s_1 \end{cases}$$

de onde resulta finalmente

$$\begin{cases} r_1 = P_0 - b_1 s_0 \\ s_0 = -\frac{P_1 + b_1 s_1}{P_2} \\ s_2 = \frac{P_3 + s_1(P_4 b_1 - P_2 b_2)}{b_1 P_2} \\ s_1 = -\frac{b_1(P_6 P_2 - a_1 b_1 P_1) + b_2 P_5}{b_2(P_4 b_1 - P_2 b_2) - a_2 b_1^3} \end{cases}$$

com

$$\begin{cases} P_0 = a_{o1} + a_{m1} - a_1 + 1 \\ P_1 = a_2 + a_1 P_0 - a_{m1} - a_{m2} - a_{o2} - a_{o1}(1 + a_{m1}) - 1 \\ P_2 = b_2 + (1 - a_1)b_1 \\ P_3 = a_2(P_0 - 1) - a_1 P_0 - a_{m1}a_{o2} - a_{m2}a_{o1} \\ P_4 = b_1(a_1 - a_2) \\ P_5 = -P_3 P_2 + P_1 P_4 \\ P_6 = -a_2 P_0 - a_{m2}a_{o2} \end{cases}$$

Os polinómios $R(q)$ e $S(q)$ são dados por

$$\begin{aligned} R(q) &= q - 1 \\ S(q) &= \frac{1 - a + a_m + a_o}{b}q + \frac{a_m a_o - a}{b} \end{aligned}$$

O polinómio $T(q)$ que é dado pela expressão

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

obtendo-se para b_m

$$b_m = \left(\frac{1 + a_{m1} + a_{m2}}{b_1 + b_2} \right)$$

Substituindo na expressão de $T(q)$ obtém-se:

$$T(q) = (q^2 + a_{o1}q + a_{o2})b_m$$

A função de controlo é dada por:

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

Substituindo os polinómios obtidos anteriormente na função de controlo obtém-se:

$$(q^2 + (r_1 - 1)q - r_1)u(k) = b_m(q^2 + a_{o1}q + a_{o2})u_c(k) - (s_0q^2 + s_1q + s_2)y(k)$$

multiplicando ambos os lados da equação por q^{-2}

$$(1 + (r_1 - 1)q^{-1} - r_1q^{-2})u(k) = b_m(1 + a_{o1}q^{-1} + a_{o2}q^{-2})u_c(k) - (s_0 + s_1q^{-1} + s_2q^{-2})y(k)$$

desenvolvendo obtém-se:

$$\begin{aligned} u(k) + (r_1 - 1)u(k - 1) - r_1u(k - 2) &= b_m(u_c(k) + a_{o1}u_c(k - 1) + a_{o2}u_c(k - 2)) \\ &\dots - s_0y(k) - s_1y(k - 1) - s_2y(k - 2) \end{aligned}$$

de onde resulta finalmente a expressão de $u(k)$

$$u(k) = b_m(u_c(k) + a_{o1}u_c(k-1) + a_{o2}u_c(k-2)) - s_0y(k) - s_1y(k-1) - s_2y(k-2) \\ \dots - (r_1 - 1)u(k-1) + r_1u(k-2)$$

onde

$$\begin{cases} r_1 = P_0 - b_1s_0 \\ s_0 = -\frac{P_1 + b_1s_1}{P_2} \\ s_1 = -\frac{b_1(P_6P_2 - P_1a_2b_1) + b_2P_5}{b_2(P_4b_1 - P_2b_2) - a_2b_1^3} \\ s_2 = \frac{P_5 + (P_4b_1 - P_2b_2)s_1}{P_2b_1} \\ b_m = \frac{1 + a_{m1} + a_{m2}}{b_1 + b_2} \end{cases}$$

com

$$\begin{cases} P_0 = a_{m1} + a_{o1} + 1 - a_1 \\ P_1 = a_2 + a_1P_0 - a_{m1} - a_{m2} - a_{o2} - a_{o1}(1 + a_{m1}) - 1 \\ P_2 = b_1(1 - a_1) + b_2 \\ P_3 = a_2(P_0 - 1) - a_1P_0 - a_{m1}a_{o2} - a_{m2}a_{o1} \\ P_4 = b_1(a_1 - a_2) \\ P_5 = -P_3P_2 + P_4P_1 \\ P_6 = -a_2P_0 - a_{m2}a_{o2} \end{cases}$$

Os parâmetros a_1 , a_2 , b_1 e b_2 são substituídos pelos valores estimados pelo controlador adaptativo em tempo-real.

Posicionamento de pólos com modelação do tempo morto fraccionário

Sistemas de 1ª ordem Considera-se que a função de transferência discreta do sistema de 1ª ordem é dada por

$$G(q) = \frac{b_1q + b_2}{q(q + a_1)}$$

Ao denominador da função $G(q)$ foi acrescentado um pólo na origem para garantir a causalidade da função [135].

Os polinómios $A(q)$ e $B(q)$ são

$$\begin{aligned} A(q) &= q(q + a_1) \\ B(q) &= b_1q + b_2 \end{aligned}$$

O polinómio que define o comportamento do sistema em malha fechada é igual a

$$A_m(q) = q(q + a_m)$$

O polinómio observador é dado por:

$$A_o(q) = q(q + a_o)$$

A equação de Diophantine é dada por:

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= (q - 1)(q + r) \\ S(q) &= q(s_0q + s_1) \end{aligned}$$

Substituindo todos os polinómios na equação de Diophantine obtém-se:

$$q(q + a_m)q(q + a_o) = q(q + a_1)(q - 1)(q + r) + (b_1q + b_2)q(s_0q + s_1)$$

Desenvolvendo e agrupando os coeficientes das potências de igual valor

$$q^4 + q^3(a_o + a_m) + q^2 a_o a_m = q^4 + q^3(r - 1 + a_1 + b_1 s_0) + q^2(a_1(r - 1) - r + b_1 s_1 + b_2 s_0) + q(s_1 b_2 - r a_1)$$

Igualando os coeficientes da mesma ordem obtém-se o sistema de três equações a três incógnitas

$$\begin{cases} a_o + a_m = r - 1 + a_1 + b_1 s_0 \\ a_o a_m = a_1(r - 1) - r + b_1 s_1 + b_2 s_0 \\ 0 = s_1 b_2 - r a_1 \end{cases}$$

Desenvolvendo a 3ª equação em ordem a r e substituindo nas outras equações

$$\begin{cases} a_o + a_m = \frac{s_1 b_2}{a_1} - 1 + a_1 + b_1 s_0 \\ a_o a_m = a_1 \frac{s_1 b_2}{a_1} - a_1 - \frac{s_1 b_2}{a_1} + b_1 s_1 + b_2 s_0 \\ r = \frac{s_1 b_2}{a_1} \end{cases}$$

$$\begin{cases} s_1 = \frac{a_1(a_o + a_m) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2} \\ a_o a_m a_1 = a_1 s_1 b_2 - a_1^2 - s_1 b_2 + a_1 b_1 s_1 + a_1 b_2 s_0 \\ - \end{cases}$$

ordenando os termos da 2ª equação e substituindo s_1

$$\begin{cases} - \\ \left(\frac{a_1(a_o + a_m) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2} \right) (a_1 b_1 + a_1 b_2 - b_2) + a_1 b_2 s_0 = a_o a_m a_1 + a_1^2 \\ - \end{cases}$$

de onde se retira finalmente que

$$\begin{cases} b_m = 1 + a_m \\ r = \frac{s_1 b_2}{a_1} \\ s_0 = \frac{P_1}{P_2} \\ s_1 = \frac{a_1(a_m + a_o) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2} \end{cases}$$

com

$$\begin{cases} P_0 = a_2 b_2 - b_2 + a_1 b_1 \\ P_1 = b_2(a_m a_o + a_1) - P_0(1 - a_1 + a_m + a_o) \\ P_2 = b_2^2 - b_1 P_0 \end{cases}$$

O polinómio $T(q)$ que é dado pela expressão

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

obtendo-se para b_m

$$b_m = 1 + a_m$$

Substituindo na expressão de $T(q)$ obtém-se:

$$T(q) = q(q + a_o)b_m$$

A função de controlo é dada por:

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

Substituindo os polinómios obtidos anteriormente na função de controlo obtém-se:

$$(q^2 + (r_1 - 1)q - r_1)u(k) = (q^2 + a_o q)b_m u_c(k) - (s_0 q^2 + s_1 q)y(k)$$

multiplicando ambos os lados da equação por q^{-2}

$$(1 + (r_1 - 1)q^{-1} - r_1 q^{-2})u(k) = (1 + a_o q^{-1})b_m u_c(k) - (s_0 + s_1 q^{-1})y(k)$$

desenvolvendo obtém-se:

$$u(k) + (r_1 - 1)u(k - 1) - r_1 u(k - 2) = b_m u_c(k) + a_0 b_m u_c(k - 1) - s_0 y(k) - s_1 y(k - 1)$$

de onde se obtém finalmente a expressão de $u(k)$

$$u(k) = -(r_1 - 1)u(k - 1) + r_1 u(k - 2) + b_m u_c(k) + a_0 b_m u_c(k - 1) - s_0 y(k) - s_1 y(k - 1)$$

onde

$$\begin{cases} b_m = 1 + a_m \\ r_1 = \frac{s_1 b_2}{a_1} \\ s_0 = \frac{P_1}{P_2} \\ s_1 = \frac{a_1(a_m + a_o) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2} \end{cases}$$

com

$$\begin{cases} P_0 = a_2 b_2 - b_2 + a_1 b_1 \\ P_1 = b_2(a_m a_o + a_1) - P_0(1 - a_1 + a_m + a_o) \\ P_2 = b_2^2 - b_1 P_0 \end{cases}$$

Os parâmetros a_1 , b_1 e b_2 são substituídos pelos valores estimados em tempo-real pelo controlador adaptativo.

Sistemas de 2^a ordem Para um sistema de 2^a ordem genérico, com modelação do tempo morto fraccionário, cuja função de transferência discreta é dada por:

$$G(q) = \frac{b_1 q^2 + b_2 q + b_3}{q(q^2 + a_1 q + a_2)}$$

Ao denominador da função $G(q)$ foi acrescentado um pólo na origem para garantir a causalidade da função [135]

Os polinómios $A(q)$ e $B(q)$ são representados por

$$\begin{aligned} A(q) &= q(q^2 + a_1 q + a_2) \\ B(q) &= b_1 q^2 + b_2 q + b_3 \end{aligned}$$

Os polinómios que definem o comportamento em malha fechada e o polinómio observador são dados por:

$$A_m(q) = q(q^2 + a_{m1} q + a_{m2})$$

$$A_o(q) = q(q^2 + a_{o1} q + a_{o2})$$

De acordo com os critérios de causalidade e de rejeição das perturbações, escolhem-se os polinómios $R(q)$ e $S(q)$ do seguinte modo

$$\begin{aligned} R(q) &= (q - 1)(r_0 q^2 + r_1 q + r_2) \\ S(q) &= q(s_0 q^2 + s_1 q + s_2) \end{aligned}$$

A equação de Diophantine é dada por:

$$A_m(q)A_o(q) = A(q)R(q) + B(q)S(q)$$

Substituindo todos os polinômios na equação de Diophantine obtém-se:

$$q(q^2 + a_{m1}q + a_{m2})q(q^2 + a_{o1}q + a_{o2}) = q(q^2 + a_1q + a_2)(q-1)(r_0q^2 + r_1q + r_2) + \dots (b_1q^2 + b_2q + b_3)q(s_0q^2 + s_1q + s_2)$$

Desenvolvendo e agrupando os coeficientes das potências de igual valor

$$\begin{aligned} q^6 + (a_{o1} + a_{m1})q^5 + (a_{o2} + a_{m1}a_{o1} + a_{m2})q^4 + (a_{m1}a_{o2} + a_{m2}a_{o1})q^3 + (a_{m2}a_{o2})q^2 = \\ = r_0q^6 + (r_1 - r_0 + a_1r_0 + b_1s_0)q^5 + \dots \\ (r_2 - r_1 + a_1r_1 - a_1r_0 + a_2r_1 + b_1s_1 + b_2s_0)q^4 + \dots \\ (-r_2 + a_1r_2 - a_1r_1 + a_2r_1 - a_2r_0 + b_1s_2 + b_2s_1 + b_3s_0)q^3 + \dots \\ (-a_1r_2 + a_2r_2 - a_2r_1 + b_2s_2 + b_3s_1)q^2 + (b_3s_2 - a_2r_2)q \end{aligned}$$

Igualando os coeficientes da mesma ordem obtém-se o sistema de cinco equações a cinco incógnitas

$$\begin{cases} r_0 = 1 \\ a_{o1} + a_{m1} = r_1 - r_0 + a_1r_0 + b_1s_0 \\ a_{o2} + a_{m1}a_{o1} + a_{m2} = r_2 - r_1 + a_1r_1 - a_1r_0 + a_2r_0 + b_1s_1 + b_2s_0 \\ a_{m1}a_{o2} + a_{m2}a_{o1} = -r_2 + a_1r_2 - a_1r_1 + a_2r_1 - a_2r_0 + b_1s_2 + b_2s_1 + b_3s_0 \\ a_{m2}a_{o2} = -a_1r_2 + a_2r_2 - a_2r_1 + b_2s_2 + b_3s_1 \\ 0 = b_3s_2 - a_2r_2 \end{cases}$$

Resolvendo a 2ª equação em ordem a r_1 e fazendo alguns agrupamentos de constantes obtém-se:

$$\begin{cases} - \\ r_1 = X_1 - b_1s_0 \\ r_2 + r_1(a_1 - 1) + b_1s_1 + b_2s_0 = X_2 \\ r_2(a_1 - 1) + r_1(a_2 - a_1) + b_1s_2 + b_2s_1 + b_3s_0 = X_3 \\ r_2(a_2 - a_1) - a_2r_1 + b_2s_2 + b_3s_1 = X_4 \\ 0 = b_3s_2 - a_2r_2 \end{cases}$$

com

$$\begin{cases} X_1 = a_{o1} + a_{m1} + 1 - a_1 \\ X_2 = a_{o2} + a_{m1}a_{o1} + a_{m2} + a_1 - a_2 \\ X_3 = a_{m1}a_{o2} + a_{m2}a_{o1} + a_2 \\ X_4 = a_{m2}a_{o2} \end{cases}$$

Resolvendo a 5ª equação em ordem a r_2 e fazendo $X_5 = a_2 - a_1$ e $X_6 = a_1 - 1$ obtém-se:

$$\begin{cases} - \\ - \\ r_2 + r_1 X_6 + b_1 s_1 + b_2 s_0 = X_2 \\ r_2 X_6 + r_1 X_5 + b_1 s_2 + b_2 s_1 + b_3 s_0 = X_3 \\ r_2 X_5 - a_2 r_1 + b_2 s_2 + b_3 s_1 = X_4 \\ r_2 = \frac{b_3}{a_2} s_2 \end{cases}$$

Substituindo r_1 e r_2 nas outras equações

$$\begin{cases} - \\ - \\ \frac{b_3}{a_2} s_2 + (X_1 - b_1 s_0) X_6 + b_1 s_1 + b_2 s_0 = X_2 \\ \frac{b_3}{a_2} s_2 X_6 + (X_1 - b_1 s_0) X_5 + b_1 s_2 + b_2 s_1 + b_3 s_0 = X_3 \\ \frac{b_3}{a_2} s_2 X_5 - a_2 (X_1 - b_1 s_0) + b_2 s_2 + b_3 s_1 = X_4 \\ - \end{cases}$$

desenvolvendo

$$\begin{cases} - \\ - \\ \frac{b_3}{a_2} s_2 + X_1 X_6 - b_1 s_0 X_6 + b_1 s_1 + b_2 s_0 = X_2 \\ \frac{b_3}{a_2} s_2 X_6 + X_1 X_5 - b_1 s_0 X_5 + b_1 s_2 + b_2 s_1 + b_3 s_0 = X_3 \\ \frac{b_3}{a_2} s_2 X_5 - a_2 X_1 + a_2 b_1 s_0 + b_2 s_2 + b_3 s_1 = X_4 \\ - \end{cases}$$

$$\begin{cases} - \\ - \\ s_0 = \frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{b_2 - b_1 X_6} \\ \left(\frac{b_3}{a_2} X_6 + b_1\right) s_2 + (b_3 - b_1 X_5) s_0 + b_2 s_1 = X_3 - X_1 X_5 \\ \left(b_2 + \frac{b_3}{a_2} X_5\right) s_2 + a_2 b_1 s_0 + b_3 s_1 = X_4 + a_2 X_1 \\ - \end{cases}$$

Substituindo s_0 nas outras equações

$$\begin{cases} - \\ - \\ s_0 = \frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{b_2 - b_1 X_6} \\ \left(\frac{b_3}{a_2} X_6 + b_1\right) s_2 + (b_3 - b_1 X_5) \left(\frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{b_2 - b_1 X_6}\right) + b_2 s_1 = X_3 - X_1 X_5 \\ \left(b_2 + \frac{b_3}{a_2} X_5\right) s_2 + a_2 b_1 \left(\frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{b_2 - b_1 X_6}\right) + b_3 s_1 = X_4 + a_2 X_1 \\ - \end{cases}$$

desenvolvendo

$$\left\{ \begin{array}{l} - \\ - \\ - \\ (b_3 X_6 + b_1) s_2 + \left(\frac{X_5 b_1 \frac{b_3}{a_2} - \frac{b_3^2}{a_2}}{b_2 - X_6 b_1} \right) s_2 + \left(\frac{X_5 b_1^2 - b_1 b_3}{b_2 - X_6 b_1} \right) s_1 + \frac{(b_3 - b_1 X_5)(X_2 - X_1 X_6)}{b_2 - X_6 b_1} + b_2 s_1 = \\ = X_3 - X_1 X_5 \\ (b_2 + \frac{b_3}{a_2} X_5 - \frac{a_2 b_1 \frac{b_3}{a_2}}{b_2 - b_1 X_6}) s_2 + (b_3 - \frac{a_2 b_1^2}{b_2 - b_1 X_6}) s_1 + \left(\frac{a_2 b_1 X_2 - a_2 b_1 X_1 X_6}{b_2 - b_1 X_6} \right) = X_4 + a_2 X_1 \\ - \end{array} \right.$$

$$\left\{ \begin{array}{l} - \\ - \\ - \\ (b_3 X_6 + b_1 + \frac{X_5 b_1 \frac{b_3}{a_2} - \frac{b_3^2}{a_2}}{b_2 - X_6 b_1}) s_2 + \left(\frac{X_5 b_1^2 - b_1 b_3}{b_2 - X_6 b_1} + b_2 \right) s_1 + \frac{b_3 X_2 - b_3 X_1 X_6 - b_1 X_5 X_2 + b_1 X_5 X_1 X_6}{b_2 - X_6 b_1} = \\ = X_3 - X_1 X_5 \\ (b_2 + \frac{b_3}{a_2} X_5) s_2 + a_2 b_1 \left(\frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{b_2 - b_1 X_6} \right) + b_3 s_1 = X_4 + a_2 X_1 \\ - \end{array} \right.$$

Fazendo $X_7 = b_2 - b_1 X_6$ e resolvendo a 1ª equação em ordem a s_2 obtém-se:

$$\left\{ \begin{array}{l} - \\ - \\ - \\ s_2 = - \frac{\left(\frac{X_5 b_1^2 - b_1 b_3}{X_7} + b_2 \right) s_1 + \frac{b_3 X_2 - b_3 X_1 X_6 - b_1 X_5 X_2 + b_1 X_5 X_1 X_6}{X_7} - X_3 + X_1 X_5}{\left(\frac{b_3}{a_2} X_6 + b_1 + \frac{X_5 b_1 \frac{b_3}{a_2} - \frac{b_3^2}{a_2}}{X_7} \right)} \\ (b_2 + \frac{b_3}{a_2} X_5 - \frac{b_1 b_3}{X_7}) s_2 + \left(b_3 - \frac{a_2 b_1^2}{X_7} \right) s_1 + \frac{a_2 b_1 X_2 - a_2 b_1 X_1 X_6}{X_7} - X_4 - a_2 X_1 = 0 \\ - \end{array} \right.$$

Fazendo

$$\left\{ \begin{array}{l} X_8 = \frac{X_5 b_1^2 - b_1 b_3}{X_7} + b_2 \\ X_9 = \frac{b_3 X_2 - b_3 X_1 X_6 - b_1 X_5 X_2 + b_1 X_5 X_1 X_6}{X_7} - X_3 + X_1 X_5 \\ X_{10} = \frac{b_3}{a_2} X_6 + b_1 + \frac{X_5 b_1 \frac{b_3}{a_2} - \frac{b_3^2}{a_2}}{X_7} \\ X_{11} = b_2 + \frac{b_3}{a_2} X_5 - \frac{b_1 b_3}{X_7} \\ X_{12} = b_3 - \frac{a_2 b_1^2}{X_7} \\ X_{13} = \frac{a_2 b_1 X_2 - a_2 b_1 X_1 X_6}{X_7} - X_4 - a_2 X_1 \end{array} \right.$$

e substituindo nas equações obtém-se:

$$\left\{ \begin{array}{l} - \\ - \\ - \\ s_2 = -\frac{X_8 s_1 + X_9}{X_{10}} \\ X_{11} s_2 + X_{12} s_1 + X_{13} = 0 \\ - \end{array} \right.$$

Substituindo s_2 na última equação

$$\left\{ \begin{array}{l} - \\ - \\ - \\ s_2 = -\frac{X_8 s_1 + X_9}{X_{10}} \\ -X_{11} X_8 s_1 - X_{11} X_9 + X_{10} X_{12} s_1 + X_{10} X_{13} = 0 \\ - \end{array} \right.$$

de onde finalmente se retira a expressão de s_1

$$\left\{ \begin{array}{l} - \\ - \\ - \\ s_1 = \frac{X_{11} X_9 - X_{10} X_{13}}{X_{10} X_{12} - X_{11} X_8} \\ - \end{array} \right.$$

obtendo-se finalmente

$$\left\{ \begin{array}{l} r_1 = X_1 - b_1 s_0 \\ r_2 = \frac{b_3}{a_2} s_2 \\ s_2 = -\frac{X_8 s_1 + X_9}{X_{10}} \\ s_0 = \frac{X_2 - X_1 X_6 - b_1 s_1 - \frac{b_3}{a_2} s_2}{X_7} \\ s_1 = \frac{X_{11} X_9 - X_{10} X_{13}}{X_{10} X_{12} - X_{11} X_8} \end{array} \right.$$

com

$$\left\{ \begin{array}{l} X_1 = a_{o1} + a_{m1} + 1 - a_1 \\ X_2 = a_{o2} + a_{m1}a_{o1} + a_{m2} + a_1 - a_2 \\ X_3 = a_{m1}a_{o2} + a_{m2}a_{o1} + a_2 \\ X_4 = a_{m2}a_{o2} \\ X_5 = a_2 - a_1 \\ X_6 = a_1 - 1 \\ X_7 = b_2 - b_1X_6 \\ X_8 = \frac{X_5b_1^2 - b_1b_3}{X_7} + b_2 \\ X_9 = \frac{b_3X_2 - b_3X_1X_6 - b_1X_5X_2 + b_1X_5X_1X_6}{X_7} - X_3 + X_1X_5 \\ X_{10} = \frac{b_3}{a_2}X_6 + b_1 + \frac{X_5b_1\frac{b_3}{a_2} - \frac{b_3^2}{a_2}}{X_7} \\ X_{11} = b_2 + \frac{b_3}{a_2}X_5 - \frac{b_1b_3}{X_7} \\ X_{12} = b_3 - \frac{a_2b_1^2}{X_7} \\ X_{13} = \frac{a_2b_1X_2 - a_2b_1X_1X_6}{X_7} - X_4 - a_2X_1 \end{array} \right.$$

O polinómio $T(q)$ que é dado pela expressão

$$T(q) = \frac{B_m(q)A_o(q)}{B(q)}$$

obtendo-se para b_m

$$b_m = \frac{1 + a_{m1} + a_{m2}}{b_1 + b_2 + b_3}$$

Substituindo na expressão de $T(q)$ obtém-se:

$$T(q) = A_o(q)b_m = (q^3 + a_{o1}q^2 + a_{o2}q)b_m$$

A função de controlo é dada por:

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$

Substituindo os polinómios obtidos anteriormente na função de controlo obtém-se:

$$(q-1)(r_0q^2 + r_1q + r_2)u(k) = b_m(q^3 + a_{o1}q^2 + a_{o2}q)u_c(k) - (s_0q^3 + s_1q^2 + s_2q)y(k)$$

desenvolvendo

$$(q^3 + q^2(r_1 - 1) + q(r_2 - r_1) - r_2)u(k) = b_m(q^3 + a_{o1}q^2 + a_{o2}q)u_c(k) - (s_0q^3 + s_1q^2 + s_2q)y(k)$$

multiplicando ambos os lados por q^{-3}

$$(1 + q^{-1}(r_1 - 1) + q^{-2}(r_2 - r_1) - q^{-3}r_2)u(k) = b_m(1 + a_{o1}q^{-1} + a_{o2}q^{-2})u_c(k) - \dots \\ (s_0 + s_1q^{-1} + s_2q^{-2})y(k)$$

de onde se obtém a expressão final da função de controle

$$u(k) = (1 - r_1)u(k - 1) + (r_2 - r_1)u(k - 2) + r_3u(k - 3) - s_0y(k) - s_1y(k - 1) - \dots - s_2y(k - 2) + b_m(u_c(k) + a_{o1}u_c(k - 1) + a_{o2}u_c(k - 2))$$

onde

$$\left\{ \begin{array}{l} r_0 = 1 \\ r_1 = X_1 - b_1s_0 \\ r_2 = \frac{b_3}{a_2}s_2 \\ s_0 = \frac{X_2 - X_1X_6 - \frac{b_3}{a_2}s_2 - b_1s_1}{X_7} \\ s_1 = \frac{X_{11}X_9 - X_{13}X_{10}}{X_{10}X_{12} - X_{11}X_8} \\ s_2 = -\frac{X_8s_1 + X_9}{X_{10}} \\ b_m = \frac{1 + a_{m1} + a_{m2}}{b_1 + b_2 + b_3} \end{array} \right.$$

com

$$\left\{ \begin{array}{l} X_1 = a_{o1} + a_{m1} + 1 - a_1 \\ X_2 = a_{o2} + a_{o1}a_{m1} + a_{m2} + a_1 - a_2 \\ X_3 = a_{m1}a_{o2} + a_{o1}a_{m2} + a_2 \\ X_4 = a_{m2}a_{o2} \\ X_5 = a_2 - a_1 \\ X_6 = a_1 - 1 \\ X_7 = b_2 - X_6b_1 \\ X_8 = \frac{X_5b_1^2 - b_1b_3}{X_7} + b_2 \\ X_9 = \frac{b_3X_2 - b_3X_1X_6 - X_5b_1X_2 + X_5b_1X_1X_6}{X_7} - X_3 + X_5X_1 \\ X_{10} = b_1 + \frac{b_3}{a_2}(X_6 + \frac{X_5b_1 - b_3}{X_7}) \\ X_{11} = \frac{b_3}{a_2}X_5 + b_2 - \frac{b_1}{X_7}b_3 \\ X_{12} = b_3 - \frac{a_2b_1^2}{X_7} \\ X_{13} = \frac{a_1b_1X_2 - a_2b_1X_1X_6}{X_7} - X_4 - a_2X_1 \end{array} \right.$$

Os parâmetros a_1 , a_2 , b_1 , b_2 e b_3 são substituídos pelos valores estimados em tempo-real pelo controlador adaptativo.

Anexo B - O simulador TrueTime

O simulador TrueTime é um simulador para MATLAB/ Simulink que permite a simulação de sistemas de tempo-real distribuídos [78]. Este simulador facilita a simulação simultânea da execução de tarefas de controlo em *kernels* de tempo-real, transmissões através de uma rede de comunicação e sistemas com dinâmicas contínuas.

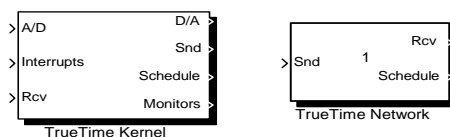
Nos ensaios executados neste trabalho utilizou-se a versão TrueTime 1.13 com uma actualização fornecida pelos autores que permite fazer o *broadcast* das mensagens. A versão TrueTime 1.2 disponibilizada em Outubro de 2004 já inclui esta actualização. Em Junho de 2008 ficou disponível a versão 1.6.

A biblioteca TrueTime é constituída por dois blocos: o bloco TrueTime Kernel, que representa um *kernel* de tempo-real e o bloco TrueTime Network que representa a rede de comunicação. Os blocos da biblioteca TrueTime são apresentados na figura 1 [146].

Os blocos TrueTime utilizam-se do mesmo modo que os blocos Simulink. No entanto é necessário efectuar a sua inicialização e criar as tarefas, rotinas de serviço à interrupção, temporizadores e eventos a eles associados antes de se efectuar a simulação do sistema.

Bloco TrueTime Kernel

A inicialização do bloco TrueTime Kernel é feita através de um ficheiro que contém o código de inicialização no qual se especifica o número de entradas e de saídas do bloco, se define a política de escalonamento a utilizar e onde se criam as tarefas, rotinas de serviço à interrupção, eventos e monitores associados ao *kernel*. A função de código associada a cada tarefa e a cada rotina de serviço à interrupção é colocada num ficheiro



TrueTime Block Library 1.13
Copyright (c) 2003 Dan Henriksson and Anton Cervin
Department of Automatic Control, Lund University, Sweden
Please direct questions and bug reports to: truetime@control.lth.se

Figura 1: Biblioteca de blocos TrueTime.

específico.

O código de inicialização e as funções de código associadas às tarefas e às rotinas de serviço à interrupção podem ser escritos em MATLAB ou em C++.

Uma função de código está dividida em vários segmentos que são executados pelo *kernel* pela ordem em que são definidos. A execução do código definido pelo utilizador é efectuada no início de cada segmento. A função de código deve retornar o tempo de execução de cada segmento.

Cada segmento de código é constituído por código MATLAB ou C++ e por comandos TrueTime.

O bloco TrueTime Kernel tem dois parâmetros associados: o nome do ficheiro que contém o código de inicialização e um parâmetro utilizado para indicar se a execução das funções de código é *fail-safe* (consultar os detalhes em [146]).

Bloco TrueTime Network

O bloco TrueTime Network permite a simulação de seis tipos de estratégias de controlo de acesso ao meio: CSMA/CD (por exemplo Ethernet), CSMA/CA (por exemplo CAN), *round-robin* (por exemplo Token Bus), FDMA e TDMA (por exemplo TTP). O atraso de propagação é ignorado. Este bloco é inicializado através da janela de interface do bloco. Os parâmetros comuns a todos os modelos são os seguintes: número de nós, taxa de transmissão, atraso pré-processamento, atraso pós-processamento, tamanho mínimo e máximo das tramas, número de *bytes* de *overhead* em cada trama e probabilidade de perda de informação durante a transmissão.

Tarefas

As tarefas podem ser periódicas ou aperiódicas. As tarefas aperiódicas são executadas através da criação de instâncias (*jobs*). Todas as instâncias em espera são inseridas numa fila de instâncias ordenadas pelo seu tempo de *release*. A cada tarefa periódica é associado um temporizador interno que cria a instância da tarefa.

Cada tarefa caracteriza-se pela função de código a ela associada e por um conjunto de atributos. Os atributos estáticos de uma tarefa são o *deadline* relativo, a prioridade, o tempo de execução no pior caso e o período (se a tarefa for periódica). O valor associado a estes atributos mantém-se constante ao longo de toda a simulação, a menos que sejam explicitamente alterados através dos comandos do tipo *ttSetX*, em que X representa o atributo. As tarefas têm também atributos dinâmicos associados. Os atributos dinâmicos são actualizados pelo *kernel* durante a simulação. Esses atributos incluem o *deadline* absoluto, o tempo de *release*, a quota temporal de execução e o tempo de execução restante. A maioria destes atributos pode também ser alterada pelo utilizador durante a simulação.

É possível associar a cada tarefa duas rotinas de serviço à interrupção. Uma das rotinas associada à situação de violação do *deadline* da tarefa e outra associada à violação do tempo de execução no pior caso. A associação dessas rotinas às tarefas é efectuada com os comandos *ttAttachDLHandler* e *ttAttachWCETHandler*.

É também possível associar pequenos pedaços de código designados por *hooks* a cada tarefa. Esses pedaços de código são executados em diferentes fases da simulação, associados aos momentos em que a tarefa fica preparada para executar, começa a

Comando	Descrição
ttInitKernel	Inicializa a <i>kernel</i> TrueTime.
ttInitNetwork	Inicializa a interface com a rede TrueTime.
ttCreatePeriodicTask	Cria uma tarefa TrueTime periódica.
ttCreateTask	Cria uma tarefa TrueTime.
ttCreateInterruptHandler	Cria um serviço à interrupção TrueTime.
ttCreateExternalTrigger	Associa um serviço à interrupção TrueTime com um canal de interrupção externo.
ttCreateMonitor	Cria um monitor TrueTime.
ttCreateEvent	Cria um evento TrueTime.
ttCreateMailbox	Cria uma caixa de correio TrueTime para a comunicação entre tarefas.
ttNoSchedule	Impede o escalonamento para uma determinada tarefa ou serviço à interrupção.
ttNonPreemptable	Impede que uma tarefa seja interrompida depois de iniciar a sua execução.
ttAttachDLHandler	Associa um <i>deadline overrun handler</i> a uma tarefa.
ttAttachWCETHandler	Associa um <i>worst-case execution time overrun handler</i> a uma tarefa.
ttAttachPrioFcn (para C++)	Associa uma função arbitrária de prioridade a um <i>kernel</i> .
ttAttachHook (para C++)	Associa um <i>run-time hook</i> a uma tarefa.

Tabela 1: Comandos utilizados para criar e inicializar objectos TrueTime.

execução, é suspensa, retoma a execução ou termina (consultar os detalhes em [146]). Esta característica facilita a implementação de políticas de escalonamento dinâmicas arbitrárias definidas pelo utilizador.

Comandos

Os comandos TrueTime podem ser divididos em três conjuntos [146]: comandos utilizados para criar e inicializar objectos TrueTime, comandos utilizados para definir e determinar os atributos das tarefas e primitivas de tempo-real. As listas de comandos são apresentadas nas tabelas 1, 2 e 3.

Exemplos

O código da função de inicialização do bloco TrueTime Kernel que implementa o controlador adaptativo por posicionamento de pólos (versão “clássica”) na arquitectura A1 (ver capítulo 4) é apresentado no exemplo 1 e a função de código que implementa o controlador respectivo é apresentada no exemplo 2. O código apresentado foi desenvolvido em MATLAB.

Comando	Descrição
ttSetDeadline	Atribui um <i>deadline</i> relativo a uma tarefa.
ttSetAbsDeadline	Atribui um <i>deadline</i> absoluto à instância de uma tarefa.
ttSetPriority	Atribui uma prioridade a uma tarefa.
ttSetPeriod	Atribui um período a uma tarefa periódica.
ttSetBudget	Atribui uma quota temporal de execução a uma tarefa.
ttSetWCET	Atribui um tempo de execução no pior caso a uma tarefa.
ttGetRelease	Determina o <i>release time</i> de uma tarefa.
ttGetDeadline	Determina o <i>deadline</i> relativo de uma tarefa.
ttGetAbsDeadline	Determina o <i>deadline</i> absoluto da instância de uma tarefa.
ttGetPriority	Determina a prioridade de uma tarefa.
ttGetPeriod	Determina o período de uma tarefa periódica.
ttGetBudget	Determina a quota temporal de execução de uma tarefa.
ttGetWCET	Determina o tempo de execução no pior caso de uma tarefa.

Tabela 2: Comandos utilizados para atribuir e determinar os atributos das tarefas.

Exemplo 1

```

% Função de inicialização para o controlador de 1ª ordem sem
% tempo morto fraccionário
function adapini
% InitKernel: n° entradas, n° saídas, prioridade fixa
ttInitKernel(1, 0, 'prioFP');
% Inicialização dos parâmetros do identificador
para = 2;
dctrl.npar = para;           % Número de parâmetros
dctrl.ForgFactor = 0.99;    % Factor de Esquecimento
PM0 = 10;                   % Covariância Inicial
dctrl.PM = PM0 * eye(para);
dctrl.LongPM = zeros(fim, para);
th0 = 1;
dctrl.th = th0 * ones(1, para);
dctrl.LongTheta = th0 * ones(fim, para);
dctrl.LongParams = zeros(fim, para);
dctrl.PError = zeros(1, para);
dctrl.Phi = zeros(1, para);
dctrl.k = 2;
dctrl.y = zeros(1, fim);

```

Comando	Descrição
ttCreateJob	Cria a instância de uma tarefa TrueTime.
ttKillJob	Acaba com a execução da instância de uma tarefa.
ttEnterMonitor	Tentativa de entrar num monitor.
ttExitMonitor	Saída de um monitor.
ttWait	Espera por um evento.
ttNotifyAll	Notifica todas as tarefas à espera de um evento.
ttTryFetch	Procura uma mensagem numa caixa de correio.
ttTryPost	Coloca uma mensagem numa caixa de correio.
ttCreateTimer	Cria um temporizador <i>one-shot</i> e associa-lhe uma rotina de serviço.
ttCreatePeriodicTimer	Cria um temporizador periódico e associa-lhe uma rotina de serviço.
ttRemoveTimer	Termina a execução de um determinado temporizador.
ttCurrentTime	Determina o tempo de execução actual.
ttSleepUntil	Adormece uma tarefa até determinado instante temporal.
ttSleep	Adormece uma tarefa durante um certo tempo.
ttAnalogIn	Lê um valor a partir de um canal analógico de entrada.
ttAnalogOut	Escreve um valor num canal analógico de saída.
ttSetNextSement	Atribui o próximo segmento de código a ser executado na função de código.
ttInvokingTask	Determina o nome da tarefa que invocou um dado serviço à interrupção.
ttCallBlockSystem	Chama um diagrama de blocos Simulink a partir de uma função de código.
ttSendMsg	Envia uma mensagem através da rede.
ttGetMsg	Determina a mensagem recebida através da rede.

Tabela 3: Primitivas de tempo-real.

```

dctrl.u = zeros(1,fim);
dctrl.r = zeros(1,fim);
dctrl.id = id_act;
dctrl.tex=zeros(1,fim);
% Inicialização dos parâmetros do regulador
dctrl.h = h;           % Intervalo de Amostragem
alphan = 2;           %Pólo em malha fechada
alphaobs = 4;         %Pólo do observador
dctrl.aobs = exp(-dctrl.h*alphaobs);
dctrl.am = exp(-dctrl.h * alphan);
dctrl.bm = 1 - dctrl.am;
dctrl.UHigh = 50;     % Limite Superior do Sinal de Controlo
dctrl.ULow = -50;    % Limite Inferior do Sinal de Controlo

```

```

% Tarefa do controlador
% CreateTask: nome tarefa, deadline, prioridade, nome do ficheiro da
% função de código, estrutura de dados interna
ttCreateTask('adap_task', dctrl.h, 10.0, 'adap1ord', dctrl);
% CreateJob: momento de release, nome da tarefa
ttCreateJob(0.0,'adap_task');
% Inicialização da rede
% CreateInterruptHandler: nome do handler, prioridade, nome do
% ficheiro com a função de código
ttCreateInterruptHandler('msgRcv', 1, 'msgRcvHandler');
% InitNetwork: número do nó, nome do handler
ttInitNetwork(4, 'msgRcv');
% CreateEvent: nome do evento
ttCreateEvent('packet');

```

A título de exemplo apresenta-se também a função de código utilizada para implementar a tarefa de controlo *adap_task* do exemplo anterior.

Exemplo 2

```

% Controlador de 1ª ordem sem tempo morto fraccionário
function [exectime, ctrld] = adap1ord(seg, ctrld)
switch seg,
case 1,
% Wait: nome do evento
ttWait('packet');
exectime = 0;
case 2,
ctrld.y(ctrld.k) = ttGetMsg;
ctrld.r(ctrld.k) = ttAnalogIn(1);
% Estimação dos parâmetros do sistema
[ctrld.th, ctrld.PM, ctrld.PError(ctrld.k)]=
rlsdf(ctrld.th,ctrld.Phi,ctrld.PM,ctrld.y(ctrld.k),ctrld.npar,...
...ctrld.ForgFactor);
for i= 1: ctrld.npar
ctrld.LongTheta(ctrld.k, i) = ctrld.th(i);
ctrld.LongPM(ctrld.k, i) = ctrld.PM(i,i);
end
% Cálculo dos parâmetros do regulador
s0 = (1 - ctrld.th(1)- ctrld.aobs - ctrld.am) / ctrld.th(2);
s1 = (ctrld.aobs * ctrld.am + ctrld.th(1)) / ctrld.th(2);
t0 = ctrld.bm / ctrld.th(2);
ctrld.LongParams(ctrld.k,1)=s0;
ctrld.LongParams(ctrld.k,2)=s1;
ctrld.LongParams(ctrld.k,3)=t0;
% Cálculo do sinal de controlo com limitação

```

```

ctrld.u(ctrld.k) = t0 * (ctrld.r(ctrld.k)...
...- ctrld.aobs * ctrld.r(ctrld.k-1)) - s0 * ctrld.y(ctrld.k)
...-s1 * ctrld.y(ctrld.k-1) + ctrld.u(ctrld.k-1);
if ctrld.u(ctrld.k) < ctrld.ULow
ctrld.u(ctrld.k) = ctrld.ULow;
end
if ctrld.u(ctrld.k) > ctrld.UHigh
ctrld.u(ctrld.k) = ctrld.UHigh;
end
% SendMsg: n° do nó receptor, conteúdo da mensagem, tamanho
% da mensagem, prioridade da mensagem
ttSendMsg(5, ctrld.u(ctrld.k), 2, ctrld.id);
% Inovações do regressor
ctrld.Phi(1) = -ctrld.y(ctrld.k);
ctrld.Phi(2) = ctrld.u(ctrld.k);
ctrld.k = ctrld.k+1;
% SetNextSegment: número do segmento
ttSetNextSegment(1);          % loop
exectime = 0.005;
end

```


Anexo C - Lista de Abreviaturas

A/D	Conversor Analógico-Digital
AFC	<i>Additive Feedforward Controller</i>
AM	<i>Asynchronous Message</i>
AML	<i>Approximate Maximum Likelihood</i>
ARMAX	<i>AutoRegressive Moving Average with eXogenous signal</i>
ARIMAX	<i>AutoRegressive Integrating Moving Average with eXogenous signal</i>
ARX	<i>AutoRegressive with eXogenous signal</i>

CAD	<i>Control And Data</i>
CAN	<i>Controller Area Network</i>
CBS	<i>Constant Bandwidth Server</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
CSMA/NBA	<i>Carrier Sense Multiple Access with Non-destructive Bitwise Arbitration</i>

D	<i>Deadline</i>
D/A	Conversor Digital-Analógico
DLC	<i>Data Length Code</i>
DS	<i>Deferrable Server</i>
DM	<i>Deadline-Monotonic</i>

EC	<i>Elementary Cycle</i>
ECU	<i>Electronic Control Unit</i>
EDF	<i>Earliest Deadline First</i>
EPA	<i>Enhanced Performance Architecture</i>

FCS	<i>Frame Check Sequence</i>
FDL	<i>Fieldbus Data Link</i>
FDMA	<i>Frequency Division Multiple Access</i>
FES	<i>Frame End Sequence</i>
FIR	<i>Finite Impulse Response</i>
FP	<i>Fixed Priority</i>
FSS	<i>Frame Start Sequence</i>
FTT	<i>Flexible Time-Triggered</i>
FTT-CAN	<i>Flexible Time-Triggered on CAN</i>
GPC	<i>Generalized Predictive Control</i>
IDE	<i>IDentifier Extension</i>
ISE	<i>Integral of the Square Error</i>
ITSE	<i>Integral of the Time-weighted Square Error</i>
LMS	<i>Least Mean Square</i>
LQG	<i>Linear Quadratic Gaussian</i>
LS	<i>Least Squares</i>
LSW	<i>Maximum Length of the Synchronous Window</i>
MAC	<i>Medium Access Control</i>
MUSMAR	<i>MULTiStep Multivariable Adaptive Regulator</i>
NCS	<i>Networked Control System</i>
NRZ	<i>Non-Return-to-Zero</i>
OE	<i>Output Error</i>
OSI	<i>Open Systems Interconnection</i>
PDC	<i>Produtor-Distribuidor-Consumidor</i>
PES	<i>Priority-Exchange Server</i>
PI	<i>Proporcional Integral</i>
PLC	<i>Programmable Logic Controller</i>
PP	<i>Posicionamento de pólos</i>
PSA	<i>PSA Peugeot Citroën</i>
QoC	<i>Quality of Control</i>
QoS	<i>Quality of Service</i>
REC	<i>Reception Error Count</i>
RLS	<i>Recursive Least Squares</i>
RM	<i>Rate-Monotonic</i>
RTR	<i>Remote Transfer Request</i>

SM	<i>Synchronous Message</i>
SRR	<i>Substitute Remote Request</i>
SRT	<i>Synchronous Requirements Table</i>
SS	<i>Sporadic Server</i>
STR	<i>Self-Tuning Regulator</i>
TDMA	<i>Time Division Multiple Access</i>
TEC	<i>Transmission Error Count</i>
TM	<i>Trigger Message</i>
TMF	Tempo Morto Fraccionário
TTP	<i>Time-Triggered Protocol</i>
UCP	Unidade Central de Processamento
WCET	<i>Worst-Case Execution Time</i>
ZOH	<i>Zero-Order Hold</i>