Luís Miguel dos Santos Farinha

SISTEMA DE CONTROLO DISTRIBUÍDO BASEADO EM ETHERNET

Luís Miguel dos Santos Farinha

SISTEMA DE CONTROLO DISTRIBUÍDO BASEADO EM ETHERNET

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, 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.

Aos meus pais

O júri

Presidente	Prof. Dr. António Ferreira Pereira de Melo Professor Catedrático da Universidade de Aveiro
Vogais	Profª. Drª. Ana Luísa Lopes Antunes Professora Adjunta da Escola Superior de Tecnologia de Setúbal
	Prof. Dr. Alexandre Manuel Moutela Nunes da Mota Professor Associado da Universidade de Aveiro (Orientador)

Agradecimentos

Quero agradecer ao meu orientador Prof. Dr. Alexandre Mota pelo constante acompanhamento, assim como pelos valiosos conhecimentos científicos que esteve sempre disposto a transmitir.

Agradeço aos meus colegas de laboratório Nuno Marujo, Rui Sancho e José Santos pelas preciosas sugestões que foram transmitindo ao longo da execução de todo este trabalho.

Agradeço aos meus pais e familiares pela confiança que sempre depositaram em mim.

Agradeço à minha namorada Raquel por toda a paciência e apoio incondicional que sempre manifestou.

Agradeço a todos os meus amigos que sempre estiveram ao meu lado.

A todos, muito obrigado.

Palavras-chave

Sistemas de Controlo Distribuído, Controlo Digital, Controlo Adaptativo, Ethernet, Sistemas Embutidos

Resumo

A crescente competitividade entre as organizações industriais tem vindo a aumentar a rapidez com que são alterados os seus processos de produção, para além disso tem-se assistido, cada vez mais, a uma exigência da redução dos custos de produção. Para dar resposta a estas exigências surgiram os sistemas de controlo distribuído. Estes sistemas, que são de elevada flexibilidade e de baixo custo, poderão ainda apresentar custos mais baixos se as unidades computacionais utilizadas forem de recursos reduzidos, ou se for usada uma infra-estrutura de comunicação já existente, como a Ethernet.

No âmbito desta dissertação é proposto um sistema de controlo distribuído baseado em Ethernet, que poderá ser utilizado para controlar vários sistemas com propriedades distintas. São usados algoritmos de controlo adaptativo por posicionamento de pólos para proceder ao controlo dos exemplos de aplicações apresentados. No que diz respeito às unidades de processamento, é estudada a viabilidade de usar microcontroladores como unidades computacionais, tanto para executar os algoritmos de controlo, como para dar suporte às comunicações conjuntamente com um controlador Ethernet.

Keywords Distributed Control Systems, Digital Control, Adaptive Control, Ethernet, Embedded Systems

Abstract

The growing competitiveness among industrial organizations has been increasing the speed at which production processes are changed. Adding to this there is a crescent demand to decrease production costs. In order to respond to these demands distributed control systems have emerged. These, highly flexible and low cost systems, can be made even less expensive if reduced resources microcontrollers are used, or if an existent communication infrastructure, such as Ethernet, is used since these already exist in most organizations.

In this thesis an Ethernet based distributed control system is proposed, which is flexible enough to be used in systems with distinct properties. Pole placement adaptive control algorithms are used to control the analyzed application examples. In what is concerned to processing units, the viability of using microcontrollers as computing units, both to run control algorithms and support communications, together with an Ethernet controller, are evaluated.

Conteúdo

1 Introdução **1**

1.1 Enquadramento 1
1.2 Motivação 3
1.3 Pressupostos 3
1.4 Estrutura da dissertação 4

2 Conceitos fundamentais **5**

2.1 Sistemas de controlo digital 5
2.1.1 Não linearidades presentes nos sistemas de controlo digital . . . 6
2.1.2 Vantagens do controlo digital 7
2.1.3 Desvantagens do controlo digital 8
2.2 Sistemas de controlo distribuído 8
2.2.1 Implicações no uso dos sistemas de controlo distribuído 9
2.2.2 Propriedades 10
2.3 Barramentos de campo 10
2.4 Ethernet 11
2.4.1 Controlo de acesso ao meio na Ethernet 12
2.4.1.1 Considerações acerca do CSMA/CD 13
2.4.2 Vantagens apresentadas pela Ethernet para aplicações em redes de controlo distribuído 13
2.4.3 Redes Ethernet tempo-real 14
2.4.3.1 Ethernet Powerlink 14
2.4.3.2 FTT-Ethernet 15
2.4.4 Modelo TCP/IP 16
2.5 Sistemas embutidos 18
2.6 Sistemas tempo-real 19
2.6.1 Sistemas tempo-real *hard* e *soft* 19
2.6.2 Escalonamento 20

2.7 *Kernels* tempo-real 21

3 Sistema desenvolvido **23**
3.1 Descrição geral 23
3.2 Módulo de processamento MXS4 24
3.2.1 Unidade computacional 25
3.2.2 FreeRTOS 26
3.3 Comunicação entre os nós 26
3.3.1 PC 27
3.3.2 Módulo de processamento MXS4 27
3.3.2.1 Controlador Ethernet 27
3.3.2.2 *Stack* TCP/IP da Microchip 28
3.4 Nó controlador 29
3.5 Nó sensor 29
3.5.1 Módulo de interface analógico 30
3.5.2 Módulo de acondicionamento de sinal de termopares 30
3.6 Nó actuador 31
3.6.1 Módulo de interface analógico 32
3.7 Interface homem-máquina 32
3.7.1 Módulo *display*/teclado 32

4 Controlo adaptativo **35**
4.1 Identificação de sistemas 36
4.2 Método dos mínimos quadrados 36
4.2.1 Implementação não-recursiva 37
4.2.2 Implementação recursiva 37
4.2.2.1 Mínimos quadrados pesados com factor de esquecimento 38
4.2.3 Escolha do intervalo de amostragem 38
4.2.4 Sinal de entrada 39
4.3 Controlo adaptativo por posicionamento de pólos 39
4.4 Controlo adaptativo por posicionamento de pólos com tempo morto fracionário 40

5 Controlo distribuído de um sistema de 1ª ordem **43**
5.1 Modelação matemática do sistema 43
5.2 Controlador no PC 43
5.2.1 Algoritmo de controlo 44
5.2.2 Condições de teste 44

5.2.3 Resultados 45
5.3 Controlador no módulo MXS4 47
5.3.1 Algoritmo de controlo 48
5.3.2 Condições de teste 48
5.3.3 Resultados 48
5.4 Comparação entre as duas abordagens 50

6 Controlo distribuído de um processo térmico 51
6.1 Parte 1 51
6.1.1 Descrição do sistema de controlo distribuído 51
6.1.2 Algoritmos de controlo 52
6.1.3 Condições de teste 52
6.1.4 Resultados 53
6.1.4.1 Controlador adaptativo por posicionamento de pólos . 53
6.1.4.2 Controlador adaptativo por posicionamento de pólos com tempo morto fraccionário 55
6.2 Parte 2 56
6.2.1 Descrição geral - controlador no PC 56
6.2.1.1 Algoritmo de controlo 57
6.2.1.2 Condições de teste 57
6.2.1.3 Resultados 58
6.2.2 Descrição geral - controlador no módulo MXS4 59
6.2.2.1 Algoritmo de controlo 60
6.2.2.2 Condições de teste 60
6.2.2.3 Resultados 60

7 Conclusão e trabalho futuro 63
7.1 Conclusões 63
7.2 Trabalho futuro 64

A Multiplexer UART 65

B Módulos de interface de sinal (Nó sensor e actuador) 67

C Módulo de acondicionamento de sinal de termopares 69

D Módulo *display*/botões 71

E Esquema eléctrico do sistema de 1ª ordem 73

F Módulo de controlo de temperatura PCT 13 **75**

G Lista de acrónimos **87**

Bibliografia **89**

Lista de Figuras

1.1 Diagrama de blocos genérico de um sistema de controlo distribuído. . . 2

2.1 Sistema de controlo digital. 6
2.2 Sistema de controlo distribuído. 9
2.3 Trama Ethernet [1]. 11
2.4 Funcionamento do CSMA/CD. 13
2.5 Formato de um ciclo do Power Ethernet. 15
2.6 Ciclo elementar do FTT-Ethernet. 16
2.7 Modelo TCP/IP. 17

3.1 Diagrama de blocos geral do sistema desenvolvido. 23
3.2 Diagrama de blocos do módulo de processamento. 24
3.3 Módulo de processamento MXS4. 25
3.4 Módulo que contém o controlador Ethernet. 28
3.5 Fluxo de informação entre o nó controlador e os restantes nós. 29
3.6 Diagrama de blocos do módulo de interface analógico (nó sensor). . . . 30
3.7 Diagrama de blocos do módulo de acondicionamento de sinal dos termopares. 31
3.8 Diagrama de blocos do módulo de interface analógico (nó actuador). . . 32

4.1 Diagrama de blocos genérico de um controlador auto-ajustável [2]. . . . 35
4.2 Erro de estimação. 36

5.1 Diagrama de blocos geral do controlador no PC. 44
5.2 Sinais para o controlador no PC. 45
5.3 Evolução dos parâmetros estimados e da matriz de variância com o controlador no PC. 46
5.4 Controlador no módulo de processamento MXS4. 47
5.5 Sinais com o controlador na MXS4. 48

5.6 Evolução dos parâmetros estimados e da matriz de variância com o controlador na MXS4. 49

6.1 Diagrama de blocos do sistema de controlo [3]. 52
6.2 Sinais para o controlo do processo térmico com controlador PPST. . . . 54
6.3 Evolução dos parâmetros estimados e da matriz de variância para controlador PPST. 54
6.4 Sinais para o controlo do processo térmico com controlador PPST-TMF. 55
6.5 Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF. 55
6.6 Sistema totalmente distribuído 57
6.7 Sinais para o controlo do processo térmico com controlador PPST-TMF. Controlador no PC. 58
6.8 Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF. Controlador no PC. 58
6.9 Diagrama geral do sistema cuja unidade de processamento é um módulo MXS4. 60
6.10 Sinais para o controlo do processo térmico com controlador PPST-TMF. Controlador no módulo MXS4. 61
6.11 Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF. Controlador no módulo MXS4. 61

A.1 Esquema eléctrico do multiplexer UART. 65
A.2 Multiplexer UART. 66
A.3 *Jumpers* para selecção da porta de entrada. 66

B.1 Módulos de interface analógico. 67
B.2 Esquema eléctrico dos módulos de interface analógico. 68

C.1 Esquema eléctrico do módulo de acondicionamento de sinal de termopares. 69

D.1 Esquema eléctrico do módulo *display*/botões. 71
D.2 Módulo *display*/botões. 72

E.1 Esquema eléctrico do sistema de 1ª ordem. 73

F.1 Módulo PCT 13 75
F.2 Painel de interface 76
F.3 Sistema de controlo módulo PCT 13 78
F.4 Módulo de potência 78

F.5 Controlador da electroválvula 79
F.6 Circuito impresso placa electroválvula 80
F.7 Esquema da placa Electroválvula 80
F.8 Equipamento de instrumentação HP34970A 81
F.9 Bomba de água 81
F.10 Depósito de água 82
F.11 Resposta ao degrau 82
F.12 Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 50 cm^3/min 83
F.13 Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 100 cm^3/min 84
F.14 Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 200 cm^3/min 84
F.15 Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 280 cm^3/min 85
F.16 Identificação subsistema electroválvula 86

Lista de Tabelas

5.1 Parâmetros do controlador. 45
5.2 Parâmetros do sistema. 46
5.3 Parâmetros do sistema. 49

6.1 Parâmetros dos controladores usados para controlar o processo térmico. 53
6.2 Parâmetros do sistema para o controlador PPST. 54
6.3 Parâmetros do sistema para o controlador PPST-TMF. 56
6.4 Parâmetros do sistema para o controlador PPST-TMF. Controlador no PC. 59
6.5 Parâmetros do sistema para o controlador PPST-TMF. Controlador no módulo MXS4. 62

F.1 Constantes de tempo subsistema electroválvula 85

Capítulo 1

Introdução

No âmbito desta tese são explorados sistemas de controlo distribuído cuja rede de comunicações é uma rede Ethernet. É estudada a viabilidade da utilização de microcontroladores de recursos reduzidos, quando comparados com um PC, como nós de processamento do sistema de controlo. Estes microcontroladores, conjuntamente com um controlador Ethernet têm suporte para comunicações sobre esta tecnologia. Quanto à rede de comunicações é utilizada uma rede Ethernet também para avaliar o seu desempenho em sistemas lentos relativamente aos atrasos que esta infra-estrutura possa causar. Relativamente aos algoritmos de controlo serão abordadas duas técnicas de controlo adaptativo, com propriedades distintas. Estes algoritmos serão inicialmente testados num PC, mais concretamente no programa de cálculo MATLAB e posteriormente são implementados num microcontrolador. Este testes permitem a comparação entre o desempenho das duas abordagens.

1.1 Enquadramento

Com o aumento da complexidade dos sistemas industriais devido às constantes mudanças e exigências do mercado, surge a necessidade de criar sistemas escalonáveis e flexíveis de forma a providenciar uma adaptação rápida e de mais baixo custo. Para responder a este problema surgiram as redes de controlo distribuído, que são sistemas de controlo digital onde o ciclo de controlo é fechado através de uma rede tempo-real e no domínio digital. Estes sistemas na sua forma básica e conceptual são constituídos por sensor(es), actuador(es) e controlador(es), estando estes, como pode ser observado na Figura 1.1, interligados por uma rede/barramento de comunicações.

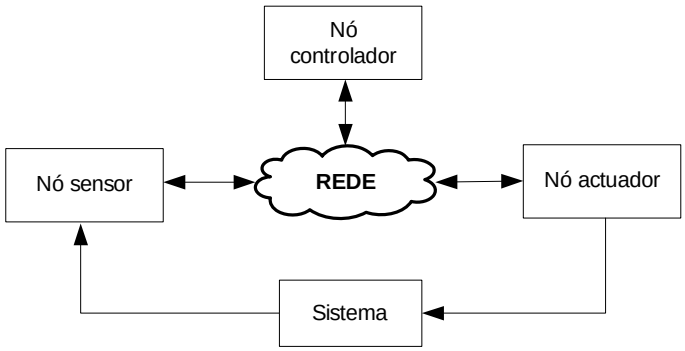


Figura 1.1: Diagrama de blocos genérico de um sistema de controlo distribuído.

O controlo digital surgiu na década de 50 do século passado em aplicações militares [2] e, não surgiu apenas com o objectivo de replicar directamente os controladores clássicos, como o PID, mas também providenciou o desenvolvimento de técnicas avançadas de controlo como a identificação de sistemas ou o controlo adaptativo. No caso concreto do controlo adaptativo, este surgiu na aviónica com o objectivo de controlar um determinado sistema para que este apresentasse um comportamento idêntico a um modelo teórico.

Os sistemas de controlo distribuído foram introduzidos em 1975 pela HoneyWell e pela Yokogawa que desenvolveram os sistemas TDC 2000 e o CENTUM respectivamente. Desde então os sistemas de controlo distribuído não têm parado de proliferar dada a massificação dos sistemas computacionais. Estes apresentam inúmeras vantagens das quais se destacam o facto de se tornar menos complexa a implementação e a manutenção, visto que se podem modificar apenas alguns dos seus blocos e não todo o sistema. Outra importante característica é o facto de toda a informação alusiva ao controlo se encontrar numa rede, facilitando assim a supervisão do processo a ser controlado. Com as características apresentadas estes sistemas apresentam alguns problemas, sendo estes o atraso na amostragem e na actuação, *jitter* na chegada da informação ou até mesmo perda de dados, tendo estes levado a vários estudos para compensar e caracterizar estas adversidades [4] [5].

Uma tecnologia largamente usada para dar suporte às comunicações em sistemas de controlo distribuídos são os barramentos de campo uma vez que estes, regra geral, garantem que as comunicações efectuadas sobre si são efectuadas em tempo-real. Quanto a aplicações, estes podem ser encontrados em automóveis, aviões, entre outros. Uma tecnologia que nos últimos anos tem sido alvo de muitos estudos e muitas adaptações para que possa providenciar garantias de tempo-real é a Ethernet, uma vez que na sua forma original não apresenta estas características.

A Ethernet surgiu em 1973 pelo investigador Robert Metcalfe na Xerox Parc, nos

Estados Unidos da América. Nesta altura existiam também dois grandes sistemas de comunicação entre PCs, que eram o TokenRing e o TokenBus, que rapidamente foram superados por esta tecnologia. A Ethernet quando surgiu suportava apenas 3Mbit/s, passando em 1983 a suportar ritmos de transmissão de 10Mbit/s. Com o aumento das tecnologias de informação e comunicação, estes ritmos revelam-se insuficientes, surgindo em 1998 a Fast-Ethernet que suportava ritmos de transmissão de 100Mbit/s.

Do ponto de vista temporal os sistemas de controlo distribuído, têm que operar em tempo-real para que o controlador funcione correctamente. A nível computacional uma ferramenta que é usada para garantir que as tarefas intervenientes no ciclo de controlo, cumpram as metas temporais, são os *kernels* de tempo-real. Estes são do tipo *multitasking* e são desenvolvidos para operar com sistemas de tempo real. Alguns exemplos de sistemas operativos tempo-real são o FreeRTOS, MicroC/OS-II [6], eCos [7] e RTAI [8].

1.2 Motivação

A Ethernet é uma rede de comunicação entre computadores essencialmente para transferência de tráfego não tempo-real. Dada a sua omnipresença, a sua largura de banda e o seu baixo custo, tornou-se inevitável a sua aplicação nas áreas de controlo distribuído. Esta na sua forma original poderá dar apenas suporte a sistemas de controlo sem restrições temporais críticas como o caso dos apresentados no âmbito deste trabalho.

Com o aparecimento de novas tecnologias de processadores que contêm periféricos e apresentam um custo reduzido, a implementação de algoritmos de controlo digital tornou-se mais simplificada e económica. Para além das características apresentadas, estes processadores apresentam já recursos computacionais suficientes para dar suporte a protocolos de comunicação sobre a tecnologia Ethernet, o que encoraja ainda mais o seu uso em sistemas de controlo distribuído baseados na mesma.

1.3 Pressupostos

Para o desenvolvimento deste trabalho partiu-se de um conjunto de pressupostos, os quais são apresentados a seguir.

- Considera-se que a incerteza nos instantes de amostragem, são razoavelmente pequenos, relativamente ao período de amostragem;
- Os atrasos provocados pela transmissão dos dados sobre a rede Ethernet, assim

como o tempo de processamento dos nós, serão igualmente pequenos relativamente ao período de amostragem;

- Por fim será considerado que a tecnologia Ethernet na sua forma fundamental, providencia qualidade de serviço suficiente para dar suporte aos sistemas de controlo distribuído apresentados.

1.4 Estrutura da dissertação

Esta dissertação encontra-se dividida em sete capítulos, que são seguidos dos Apêndices. No capítulo 2 são apresentados conceitos fundamentais que são a base de todo o sistema desenvolvido no âmbito deste trabalho, dos quais se destacam os sistemas de controlo digital distribuído e redes Ethernet.

No capítulo 3 será apresentado o sistema desenvolvido, assim como todas as suas características mais importantes. Serão também detalhados todos os aspectos construtivos relacionados com o mesmo.

O tema central do capítulo 4 é o controlo adaptativo. Neste serão apresentadas técnicas de identificação de sistemas, assim como duas técnicas de controlo adaptativo.

No capítulo 5 será apresentado um exemplo de aplicação do sistema desenvolvido, que é controlar um sistema de 1ª ordem baseado em electrónica. Serão também neste capítulo apresentados os respectivos resultados.

No que diz respeito ao capítulo 6, é apresentado outro exemplo de aplicação do sistema desenvolvido, que é controlar um processo térmico e uma comparação entre dois algoritmos de controlo neste mesmo sistema e nas mesmas condições. Neste mesmo capítulo serão apresentados os resultados respectivos.

No capítulo 7 serão apresentadas as conclusões finais assim como propostas de trabalho futuro.

Capítulo 2

Conceitos fundamentais

Neste capítulo serão introduzidos alguns conceitos fundamentais considerados importantes no desenvolvimento deste trabalho. No que diz respeito à teoria de controlo serão abordados temas tais como o controlo digital e sistemas de controlo distribuído. Do ponto de vista das comunicações, será feita uma pequena abordagem sobre Ethernet e serão apresentados alguns protocolos de comunicação de tempo-real baseados na mesma. Serão igualmente expostos alguns conceitos de tempo-real.

2.1 Sistemas de controlo digital

O controlo digital é uma área de controlo que tem origem no surgimento e proliferação dos processadores digitais. Com a massificação dos mesmos, os sistemas de controlo digital, têm tido tendência a substituir os sistemas de controlo analógico. Assim, os sistemas de controlo digital tornaram-se bastante comuns, estando presentes nos mais variados objectos que fazem parte do quotidiano, sendo muitos deles, inclusivamente, de elevada complexidade.

As técnicas de controlo digital permitem replicar controladores analógicos, como o PID e implementar algoritmos que só são possíveis no domínio digital, como por exemplo as técnicas de controlo baseadas em modelo, que englobam o controlo adaptativo. O suporte destes controladores é a identificação do modelo do sistema [9].

Um sistema de controlo digital , é constituído, na sua forma geral por um processador digital ou computador digital, sensor(es), actuador(es), ADC(s) e DAC(s), tal como ilustra a Figura 2.1.

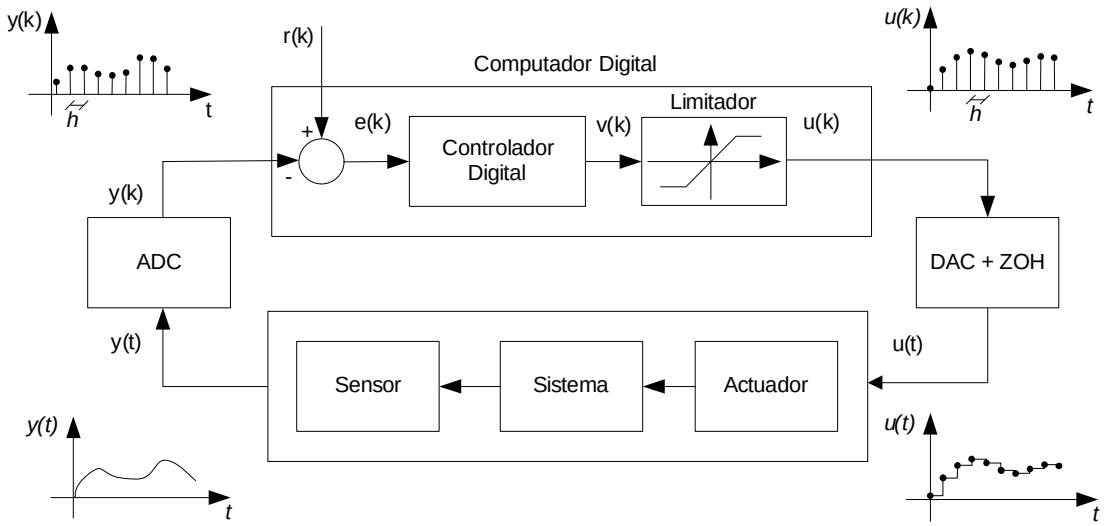


Figura 2.1: Sistema de controlo digital.

Nos sistemas de controlo digital, o sinal de saída do sistema $y(t)$ é amostrado com um determinado período de amostragem h, imposto pelo relógio do computador, originando desta forma o sinal $y(k)$. Esta conversão é efectuada por um conversor analógico-digital (ADC). No processo de amostragem, o controlador não tem qualquer percepção do estado do sistema durante o intervalo entre amostras. Posto isto, torna-se necessário escolher um período de amostragem adequado, para que o sinal no domínio digital contenha informação suficiente acerca do sinal analógico. O sinal de controlo no domínio digital $u(k)$, é convertido para o domínio analógico $u(t)$ através de um conversor digital analógico (DAC), que por meio do interpolador de ordem zero ZOH, mantém a mesma amplitude até à amostra seguinte.

Nos sistemas de controlo digital, o sinal de erro $e(k)$ é determinado no processador digital através de operações aritméticas. Isto possibilita que o sinal de referência $r(k)$ possa ser introduzido por exemplo através de um teclado.

2.1.1 Não linearidades presentes nos sistemas de controlo digital

Como pode ser observado nos sinais apresentados na Figura 2.1, existem não linearidades inerentes aos próprios sistemas de controlo digital. Estão presentes em alguns processos, tais como o de conversão do sinal analógico $y(t)$ para o sinal digital $y(k)$ e no processo de conversão do sinal de controlo digital $u(k)$ para o sinal de controlo analógico $u(t)$. Esta não linearidade deve-se ao facto da conversão de um sinal entre estes dois domínios, implicar a adição de mecanismos de adaptação de sinal, tais como

DACs, ADCs ou ZOH. Isto reflecte-se em alguns aspectos tais como a imposição do mesmo nível do sinal de controlo entre amostras imposto pelo ZOH, ou a limitação de resolução presente nos ADCs e nos DACs. A limitação do sinal de controlo $v(k)$ imposta pelo limitador, originando o sinal $u(k)$ apresenta-se igualmente como uma não linearidade presente tanto em sistemas de controlo analógico bem como digital. Esta limitação deve-se às restrições físicas do actuador, e deve ser tida em conta aquando do projecto do controlador. Contudo, apesar destas limitações, os sistemas de controlo digital apresentam inúmeras vantagens que facilmente justificam a sua utilização.

2.1.2 Vantagens do controlo digital

Neste ponto são apresentadas algumas das razões/vantagens que tornaram os sistemas de controlo digital tão utilizados, em detrimento dos sistemas de controlo analógico [10].

- **Precisão -** Apesar da conversão analógico/digital ser uma fonte de erro, torna-se muitas vezes preferível a utilização de sinais digitais dado que estes são mais imunes ao ruído do que os sinais analógicos. Isto deve-se ao facto de quando os sinais se encontram no domínio digital, deixarem de ser afectados do ruído. A regeneração dos sinais digitais não causa a sua deterioração, ao contrário dos sinais analógicos, tornando assim possível transmitir sinais digitais por distâncias que seriam impossíveis no domínio analógico;

- **Operações efectuadas -** No domínio digital, as operações efectuadas sobre estes sinais são apenas operações aritméticas, visto que os sinais digitais são interpretados sob a forma de números. Dada a fiabilidade dos processadores, o erro associado a este processo é negligível. Em contraste com os sistemas de controlo analógico, onde as operações são efectuadas por componentes que têm associado a si erros que podem influenciar de forma significativa os resultados obtidos;

- **Flexibilidade -** A alteração de sistemas de controlo analógico na maioria das vezes implica a alteração de *hardware*, e não permite a implementação de técnicas de controlo mais complexas. Por sua vez, os controladores digitais são implementados por *software* ou *firmware*, permitindo desta forma a alteração fácil dos parâmetros dos controladores, ou até mesmo de todo o controlador;

- **Custo -** A massificação dos circuitos digitais, conduziu a uma drástica queda do seu preço, tornando assim viável do ponto de vista económico a implementação

de controladores digitais em processadores.

2.1.3 Desvantagens do controlo digital

Os sistemas de controlo digital devida à sua estrutura, apresentam as desvantagens abaixo apresentadas, em relação aos sistemas de controlo analógico.

- **Efeito estroboscópico -** Uma vez que a saída do sistema é um sinal analógico, este terá de ser amostrado, com um determinado período. Durante este período, o controlador não tem percepção da saída do sistema, o que origina o efeito estroboscópico. Se este período não for devidamente escolhido, a saída do sistema pode degradar-se ou até mesmo tornar-se instável;

- **Latência** - Uma vez que o algoritmo de controlo é implementado num processador digital, o tempo de processamento deste, pode influenciar a performance do controlador e, por conseguinte, a saída do sistema. Este problema tem vindo a ser minimizado, uma vez que os processadores têm tendência a apresentar cada vez mais poder de cálculo.

2.2 Sistemas de controlo distribuído

Um sistema de controlo distribuído é um sistema de controlo digital, que consiste na distribuição geográfica dos sensor(es), actuador(es) e controlador(es), estando estes coordenados pela informação que trocam entre si através de uma rede de comunicações, ou barramento de campo [11]. Estes sistemas na literatura inglesa, são muitas vezes denominados *Networked Control Systems* (NCS), ou *Distributed Control Systems* (DCS). Os sistemas de controlo distribuído podem ser frequentemente encontrados em complexos industriais, aviões, automóveis, entre outros.

Estes sistemas, quando comparados com sistemas de controlo centralizado, apresentam uma grande flexibilidade. O facto de os seus componentes estarem interligados por uma rede, conduz a uma redução da quantidade de ligações necessárias, e consequentemente reduz o seu custo e complexidade de implementação. Aumenta a facilidade de incluir novos componentes no sistema de controlo, bem como a sua manutenção e diagnóstico. Uma vez que os dados intervenientes no controlo se encontram numa rede de comunicações, a monitorização dos processos a controlar torna-se simples, relativamente ao que acontece nos sistemas de controlo centralizado [12] [11]. A Figura 2.2 apresenta um sistema de controlo distribuído.

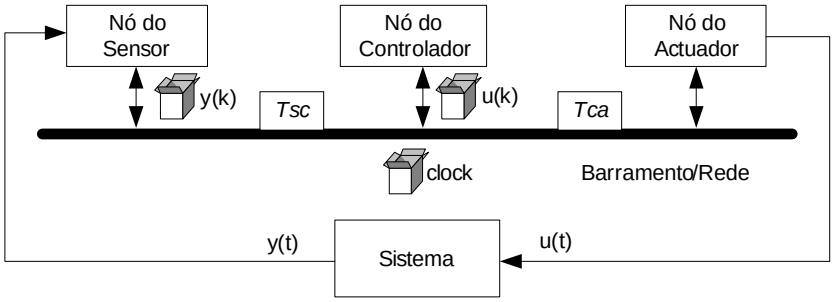


Figura 2.2: Sistema de controlo distribuído.

Nos sistemas de controlo distribuído, o nó sensor é responsável pela amostragem do sinal de saída do sistema $y(t)$, procedendo ao envio deste no domínio digital $y(k)$ para o controlador, através da rede de comunicações ou barramento de campo. Este encontra-se encapsulado numa trama ou num pacote, dependendo da rede/barramento em questão. Por sua vez o controlador aplica um algoritmo de controlo, determinando desta forma o sinal de controlo $u(k)$ a aplicar ao sistema. Este também é enviado para o actuador através da rede de comunicações. Por fim o actuador converte o sinal de controlo $u(k)$, no sinal de controlo no domínio analógico $u(t)$. Para que este sistema funcione correctamente, todos os nós têm que estar sincronizados. Este sincronismo é efectuado através do sinal de *clock* do sistema que pode ser gerado por qualquer um dos nós. Este sinal encontra-se igualmente sob a forma de trama ou pacote.

2.2.1 Implicações no uso dos sistemas de controlo distribuído

Dado que os sistemas de controlo distribuído têm os seus blocos interligados por uma rede de comunicações um problema inerente aos mesmos é o atraso na transmissão dos dados. Na Figura 2.2, estes são representados por *Tsc*, e *Tca*. O *Tcs* diz respeito ao atraso dos dados transmitidos entre o nó sensor e o nó controlador e o *Tca* ao atraso dos dados alusivos à comunicação entre o nó controlador e o nó actuador. Caso a soma destes tempos seja bastante inferior ao período de amostragem, estes podem ser desprezados, considerando deste modo que as comunicações entre os nós são efectuadas instantaneamente. Caso contrário, estas características não poderão ser desprezadas, podendo inclusivamente causar a degradação da performance do controlador, ou até mesmo conduzirem à instabilidade do sistema a controlar. Outra questão que também deverá ser tida em conta nestes sistemas é o *jitter* presente nos tempos *Tsc* e *Tca*. Este parâmetro representa as variações que estes tempos podem apresentar entre sucessivos envios de pacotes ou *frames*. Estas propriedades intrínsecas aos sistemas de controlo distribuído, são apresentadas em artigos tais como [13], [14] e [5]. O tipo de sistemas a

controlar no âmbito deste trabalho, possuem uma dinâmica suficientemente lenta para que estes efeitos possam ser desprezados.

Existe, adicionalmente, outra preocupação quanto às redes de controlo distribuído que é a perda de pacotes. A sua ocorrência poderá tornar-se igualmente factor de degradação, ou até mesmo de instabilização do sinal de saída do sistema a controlar [15].

2.2.2 Propriedades

Os sistemas de controlo distribuído podem ser avaliados com base nas propriedades apresentadas a seguir.

- **Agregabilidade** - Com a complexidade crescente dos sistemas torna-se importante que quando vários sistemas interajam, tenham um funcionamento idêntico ao seu funcionamento isolado. Um sistema diz-se que apresenta agregabilidade se não alterar as suas propriedades de funcionamento quando integrado em outro sistema;

- **Escalabilidade -** Entende-se por escalabilidade a capacidade de um sistema estar preparado para crescer de uma forma sustentada. Um sistema diz-se escalável se apresentar esta propriedade. O sucesso da Internet, deveu-se em grande parte a esta propriedade;

- **Fiabilidade -** A fiabilidade é o valor de confiança que pode ser depositado num determinado sistema, nomeadamente quando o seu mau funcionamento pode colocar em risco a integridade física de pessoas. Para que um determinado sistema apresente esta propriedade, terá de estar sempre disponível e apresentar um correcto funcionamento;

- **Interoperabilidade** - Com o crescimento do número de diferente sistemas, surgiu a necessidade de os interligar. Se um sistema se conseguir ligar a outros de uma forma transparente, diz-se que este é interoperável. Muitas vezes recorre-se a sistemas que permitem a interligação entre várias tecnologias, denominados de *gateways.*

2.3 Barramentos de campo

Com o interesse crescente de distribuir os elementos de um sistema de controlo, têm surgido nos últimos anos muitos protocolos de comunicação para dar resposta a esta

necessidade. Os protocolos de comunicação que dão suporte a este tipo de comunicações são os barramentos de campo, ou *Fieldbuses*, como são denominados na literatura inglesa. Uma propriedade básica que um protocolo de comunicação terá de ter para ser considerado um barramento de campo, é providenciar garantias de tempo-real nas comunicações.

Como anteriormente referido, existem nos dias de hoje inúmeros barramentos de campo. Alguns dos mais usados são o barramento CAN (*Controller Area Network*) que é largamente usado na indústria automóvel, o Profibus que é largamente usado na indústria, na interligação de autómatos, entre outros. Recentemente surgiu o FlexRay, igualmente direccionado para a indústria automóvel.

2.4 Ethernet

O termo Ethernet é usado para referenciar todas as tecnologias presentes nas redes de área-local (LANs) que são abrangidas pelo standard IEEE 802.3. Neste standard estão contempladas velocidades de 10-100-1000 Mbps para par trançado (*Twisted pair*) [16]. A escolha da velocidade depende das exigências das aplicações às quais se destina. A maioria dos sistemas *Embedded Ethernet* suportam velocidades de 10 Mbps. Se for uma rede terminal a velocidade mais usada é de 100 Mbps. Caso seja uma ligação principal a um servidor, é usual usar uma ligação de 1000 Mbps.

A Ethernet é uma tecnologia baseada em tramas ou *frames*, cujo tamanho está compreendido entre 72 e 1526 *bytes*. A estrutura desta está apresentada na Figura 2.3, sendo de seguida feita a descrição de cada um dos seus campos.

No que diz respeito ao endereçamento efectuado na Ethernet, este pode ser feito por *Broadcast*, *Unicast* e *Multicast*. No endereçamento *Broadcast*, todas as estações que se encontram no mesmo domínio de colisão processam a mensagem, ou seja a mensagem destina-se a todas as estações presentes no mesmo. No endereçamento *Unicast*, a mensagem transmitida destina-se apenas à estação cujo endereço corresponde ao campo de endereço de destino da trama Ethernet. Por fim, no endereçamento *Multicast*, a mensagem destina-se a um grupo de estações presentes na rede.

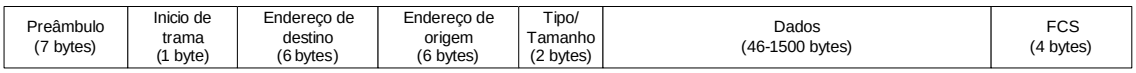


Figura 2.3: Trama Ethernet [1].

Descrição dos campos da trama Ethernet:

- **Preâmbulo** - Permite que os dispositivos se sincronizem;

- **Início de trama** - Este *byte* delimita o início da trama;
- **Endereço de destino** - Endereço físico do nó de destino;
- **Endereço de origem** - Endereço físico do nó de origem;
- **Tipo/Tamanho** - Identifica o protocolo de camada superior que se encontra no campo de dados. O número de *bytes* presentes no campo de dados é indicado pelo campo tamanho;
- **Dados** - É no campo de dados que se encontra a informação útil a ser transmitida. Caso o tamanho dos dados seja inferior a 46 *bytes*, é necessário proceder ao preenchimento deste, até perfazer este valor. Este preenchimento é denominado de *padding;*
- **FCS (*Frame Check Sequence*)** - Este campo contém o valor que é usado pelo destinatário da mensagem para determinar a integridade dos dados. Este valor é determinado através do algoritmo *Cyclic Redundancy Check* (CRC).

2.4.1 Controlo de acesso ao meio na Ethernet

O controlo de acesso ao meio é usado em qualquer rede de comunicações onde estejam presentes dois ou mais pontos que possam concorrer pelo canal de comunicação definindo a forma como os nós devem aceder ao canal. O controlo de acesso ao meio na Ethernet é feito através de uma técnica denominada na literatura inglesa de *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD). Este mecanismo de arbitragem pode ser decomposto em três partes:

- ***Carrier Sense*** - Quando um nó pretende transmitir, este tem a capacidade de verificar se o canal de comunicação se encontra livre ou ocupado;
- ***Multiple Access*** **-** Propriedade de vários nós poderem concorrer pelo canal de comunicação;
- ***Collision Detection*** **-** Para além das propriedades apresentadas acima, este algoritmo tem a capacidade de detectar colisões. Uma colisão dá-se quando duas, ou mais, estações tentam transmitir em simultâneo ou quando estão suficientemente distantes, de forma a não detectarem que o canal já se encontra ocupado devido ao tempo de propagação do sinal neste, começando a transmitir entretanto. Quando as estações intervenientes na transmissão detectam uma colisão, ambas param de transmitir e enviam um sinal de *jam*, para que todas as estações

detectem que ocorreu uma colisão. Depois deste processo, ambas as estações têm que retransmitir os dados. Para que a retransmissão não ocorra ao mesmo tempo, originando novamente uma colisão, as estações determinam um conjunto de tempos de espera discretos. Este conjunto é determinado através do algoritmo *Truncated Exponencial Back-off* . Dentro deste conjunto, é escolhido um valor de forma aleatória para tentar a retransmissão. O número máximo de tentativas de transmissão é 16.

A Figura 2.4 ilustra o funcionamento do CSMA/CD.

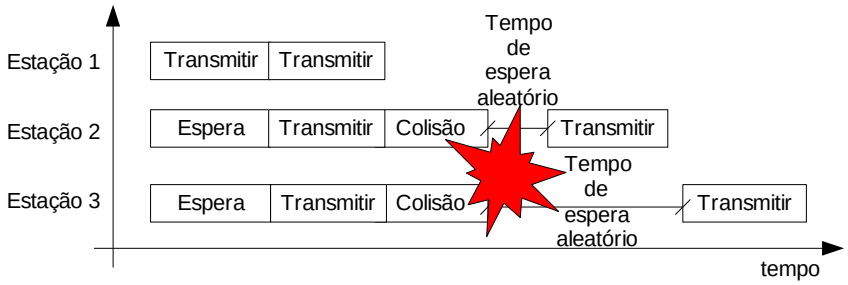


Figura 2.4: Funcionamento do CSMA/CD.

2.4.1.1 Considerações acerca do CSMA/CD

Com a técnica CSMA/CD usada na Ethernet torna-se difícil determinar a latência de uma determinada transmissão, dado que o instante em que ocorre a retransmissão em caso de colisão é determinado de forma aleatória, ao contrário do que acontece no CSMA/CA, usado nos barramentos CAN, onde as tramas de maior prioridade em caso de colisão continuam a ser transmitidas sem ter que ocorrer a sua retransmissão. Na Ethernet, ambas as transmissões são abortadas, tendo que voltar a concorrer pelo meio de comunicação.

Dadas as razões apresentadas, a Ethernet na sua forma original não é considerada um barramento para comunicações de tempo-real, uma vez que não preenche o requisito de determinismo no cálculo do tempo de transmissão de uma mensagem.

2.4.2 Vantagens apresentadas pela Ethernet para aplicações em redes de controlo distribuído

Apesar da Ethernet ter sido desenvolvida para comunicações entre computadores que não apresentavam requisitos temporais, esta tem sido largamente usada em comunicações com este tipo de requisitos. As razões para o uso desta tecnologia neste tipo

de ambientes, devem-se principalmente às suas características únicas, que são abaixo apresentadas [17].

- É uma tecnologia de custo reduzido uma vez que se encontra massificada;
- Fácil integração na Internet, uma vez que existem inúmeras *stacks* que disponibilizam os mais importantes protocolos presentes na Internet, nomeadamente suporte a *sockets*;
- A largura de banda apresentada pelos barramentos de campo convencionais é insuficiente para as mais recentes tecnologias de controlo baseados em multimédia, como por exemplo visão computacional;
- É uma tecnologia bem estudada e bem especificada, não apresentando desta forma grandes falhas do ponto de vista de compatibilidade.

2.4.3 Redes Ethernet tempo-real

Nesta secção serão apresentados alguns protocolos e técnicas de comunicação que conferem à Ethernet características de tempo-real [18]. Uma das técnicas usadas para conferir à Ethernet propriedades de tempo-real, é a alteração da camada de acesso ao meio. A alteração desta camada, implica a alteração do *firmware* dos controladores de Ethernet e não elimina as colisões mas sim facilita a análise de pior caso na ocorrência das mesmas.

Outra técnica que permite reduzir as colisões é usar *switches* que separam cada uma das suas portas num domínio de colisão. Ainda assim, como os *switches* convencionais só têm um *buffer*, não é possível atribuir diferentes prioridades ao tráfego, e obter o determinismo desejado. Neste domínio têm sido desenvolvidos *switches* com vários *buffers* de prioridades diferentes de forma a segmentarem o tráfego, dando maior prioridade ao tráfego tempo-real [19].

A técnica mais determinística é usar protocolos de camada superior que definem o instante em que são transmitidos os pacotes evitando as colisões. Nos próximos pontos são apresentados dois protocolos baseados nesta técnica.

2.4.3.1 Ethernet Powerlink

O Ethernet Powerlink é um protocolo que corre em cima da Fast-Ethernet com propriedades de tempo-real [20]. Este define ciclos de duração de $100\mu s$, suportando tráfego do tipo *time-trigger* (síncrono) e *event-trigger* (assíncrono), como pode ser observado

na Figura 2.5. Esta arquitectura é constituída por um árbitro que é chamado de *Managing Node* (MN), que funciona como o moderador de uma conversa, sendo os restantes nós denominados de *Controlled Nodes* (CN). O MN define os instantes temporais em que são efectuadas todas as transmissões, sendo desta forma o relógio do sistema.

O ciclo começa quando o MN envia um *Start-of-Cycle* (SoC), dando indicação aos CN do início da janela temporal de tráfego síncrono. Dentro desta janela, o MN envia a cada CN uma trama *unicast*, denominada de *Poll Request* (PReq), indicando que este pode proceder à transmissão dos dados. As tramas enviadas pelo CN são *multicast* e são denominadas de *Poll Response* (PRes). É dentro desta janela que se encontra todo o tráfego tempo-real síncrono.

A janela temporal assíncrona dentro do ciclo, é iniciada quando o MN envia um *Start of Asynchronous* (SoA). Os dados que são enviados dentro desta janela temporal, são dados que não têm propriedades tempo-real. Se um CN tiver várias tramas para enviar só pode enviar uma dentro de cada janela assíncrona.

Com as propriedades apresentadas, a implementação do Ethernet Powerlink não necessita de alterações ao nível do *hardware/firmware* do controlador, bem como no controlo de acesso ao meio, uma vez que as tramas Ethernet não sofrem alterações. Este protocolo apresenta um *jitter* baixo porque as mensagens são sincronizadas através dos MN. Do ponto de vista de utilização da largura de banda com informação útil este protocolo não é eficaz, visto que em média, é enviada uma trama de sincronização por cada trama com informação útil.

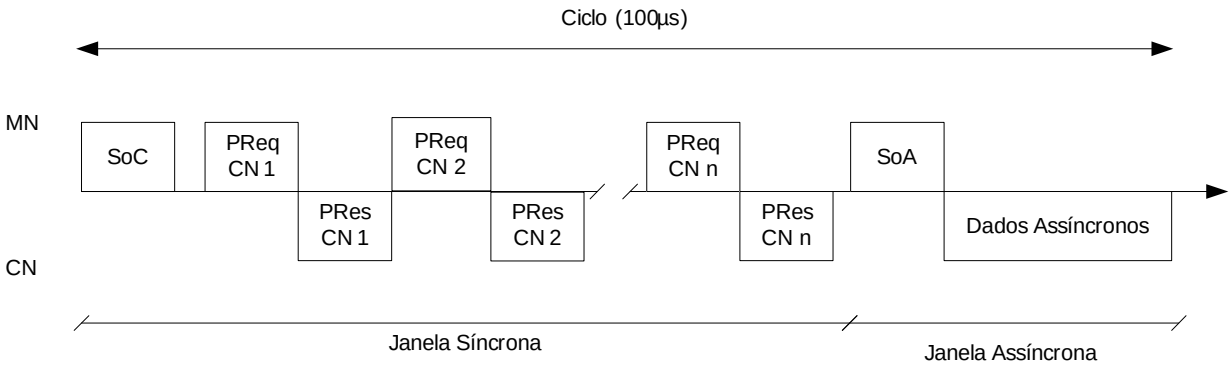


Figura 2.5: Formato de um ciclo do Power Ethernet.

2.4.3.2 FTT-Ethernet

O FTT-Ethernet, é um protocolo que confere à Ethernet propriedades tempo-real, desenvolvido na Universidade de Aveiro [18], utilizando uma política *Master-Multi Slave*. Tal como no Ethernet Powerlink, o FTT-Ethernet é constituído por um ciclo, que é

denominado de ciclo-elementar (EC) e, por sua vez, este é constituído por várias janelas que podem acomodar diferentes tipos de tráfego, mas geralmente são apenas usadas as janelas síncrona e assíncrona. Por sua vez cada janela é constituída por vários *time-slots*, que acomodam tráfego de fontes diferentes. Um exemplo do EC está apresentado na Figura 2.6.

Cada EC é iniciado por uma *Trigger-Message* (TM), que é enviada pelo *Master*. Esta mensagem é *broadcasted* e contém informação para que cada *Slave* possa determinar em que instante deve enviar os dados síncronos (SM), evitando desta forma colisões. Esta mensagem permite também que todos os nós se sincronizem. Os *Slaves* quando transmitem dados, fazem-no também por *broadcast*.

No que diz respeito ao tráfego assíncrono, o *Master* questiona aos *Slaves* se estes têm dados para transmitir. Os dados assíncronos são divididos em dois tipos, os dados assíncronos tempo-real, e assíncronos não tempo-real. Dentro destes dois tipos de dados, os primeiros têm maior prioridade, podendo estes, a título de exemplo, ser um sinal de alarme. No que diz respeito aos dados assíncronos não tempo-real, estes podem ser dados de um servidor FTP que não tem requisitos tempo-real.

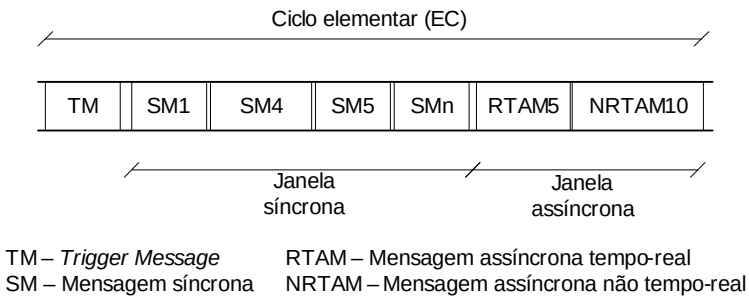


Figura 2.6: Ciclo elementar do FTT-Ethernet.

2.4.4 Modelo TCP/IP

De forma a organizar e a estruturar a tecnologia que suporta as redes de comunicações entre computadores surgiu o modelo TCP/IP baseado no modelo OSI, sendo este uma pilha protocolar onde se encontram todos os protocolos presentes hoje em dia nas comunicações entre computadores. Este modelo permite uma abstracção, por parte de quem usa apenas as suas camadas superiores. A título de exemplo, quem usa protocolos da camada de transporte não tem que ter conhecimento detalhado acerca das camadas de nível inferior. A Figura 2.7 apresenta o modelo TCP/IP [21].

Alguns dos protocolos contemplados neste modelo são, regra geral, suportados por *stacks* que se encontram presentes nos computadores pessoais para dar suporte a to-

das as comunicações efectuadas na Ethernet/Internet. Para além dos computadores pessoais, estas encontram-se hoje em dia em sistemas embutidos baseados em micro-controladores, para que estes suportem comunicações baseadas na tecnologia Ethernet, como por exemplo *sockets* TCP.

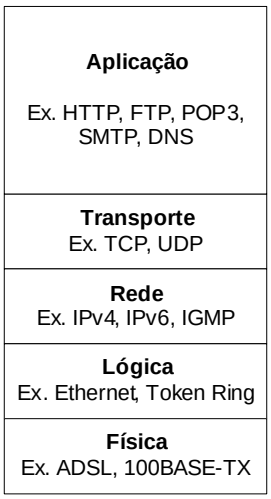


Figura 2.7: Modelo TCP/IP.

A descrição de cada uma das camadas do modelo TCP/IP é apresentada de seguida. São apresentados os detalhes de maior realce acerca de cada camada, assim como exemplos de protocolos presentes em cada uma delas.

- **Aplicação** - A camada de aplicação é a camada que disponibiliza os protocolos que mais perto se encontram dos serviços que estão disponíveis para o utilizador final na Ethernet/Internet. Dentro desta camada existem dois tipos de protocolos, os que fornecem directamente serviços ao utilizador e os que têm funções de suporte. Alguns dos protocolos que fornecem serviços directamente ao utilizador são o HTTP para disponibilização de conteúdo Web, o FTP para transferência de ficheiros e quanto ao serviço de E-mail, este pode ser suportado pelo POP3, SMTP ou IMAP. Do ponto de vista de suporte, existem o DHCP para que os dispositivos de rede possam adquirir um endereço IP de forma automática, o DNS para resolver os nomes em endereços IP, ou o SNMP para fazer a gestão da rede.

- **Transporte** - A principal função da camada de transporte é providenciar uma comunicação virtual ponto-a-ponto entre a camada de aplicação de dois dispositivos diferentes. Os dois principais protocolos presentes nesta camada são o TCP e o UDP. O TCP usa uma técnica de estabelecimento prévio de um canal virtual entre as duas aplicações que pretendem comunicar. Este procedimento de

estabelecimento de ligação é denominado de *three-way handshake*. O TCP é direccionado para comunicações que tolerem latência mas que não tolerem perdas de pacotes. O UDP usa uma filosofia de *best-effort*, sendo desta forma direccionado para comunicações que pretendam rapidez e que tolerem perdas de pacotes.

- **Rede** - O objectivo da camada de rede é escolher o melhor caminho que os pacotes percorrem quando são transmitidos pela rede. O encaminhamento destes entre as várias redes é efectuado nesta camada e o protocolo mais usado é o IP. Este fornece um conjunto de primitivas que permitem que os pacotes cheguem tanto à rede como ao dispositivo de destino de forma correcta. Outros protocolos desta camada são o ICMP, que é usado para fazer testes sobre a rede, nomeadamente, testes de conectividade.

- **Lógica** - Quando chegam dados a esta camada de outras de nível superior antes de estes serem transmitidos é necessário verificar se o meio de transmissão está livre, sendo essa gestão efectuada nesta camada através de políticas de acesso ao meio. É nesta camada que se enquadra a Ethernet.

- **Física** - Esta é a camada de nível mais baixo. Trata dos sinais eléctricos que são transmitidos e define a codificação do canal que é usada, o *bit rate*. Define também os conectores, os cabos, entre outros. Alguns exemplos de normas de camada física, são Wi-Fi, ADSL, entre outros.

2.5 Sistemas embutidos

Um sistema embutido ou *embedded system* é um sistema desenvolvido para uma aplicação específica, não sendo concebido para ser programado pelo utilizador final como os computadores de uso geral [22]. Um sistema embutido possui, frequentemente, características de tempo-real. Com estas propriedades, estes sistemas tornam-se mais económicos, visto que visam apenas cobrir um único propósito, eliminando recursos desnecessários. Dada a objectividade destes sistemas, estes podem ser optimizados, o que acresce a sua fiabilidade e a sua performance. Os sistemas embutidos fazem parte do nosso quotidiano podendo estar presentes em leitores de MP3, telemóveis ou até mesmo no microondas. Por outro lado os sistemas embutidos podem estar presentes em sistemas muito complexos como controladores de linhas de produção industriais, controladores dos sistemas de navegação dos aviões ou controlador de tracção dos automóveis.

Estes sistemas do ponto de vista computacional podem variar desde um simples microcontrolador com uma arquitectura de 8 *bits*, até sistemas com múltiplas unidades de processamento de 64 *bits*. Podem ser processadores digitais de sinal DSPs, que são direccionados para processamento de sinal. Na área dos sistemas embutidos, tem sido cada vez mais constante a utilização de sistemas do tipo *field-programmable gate array* (FPGA), que são semicondutores que podem ser reconfigurados depois do fabrico.

Em 2002 a Microchip Technology Inc., introduziu um novo conceito nesta área, quando lançou os DSCs. Estes são um híbrido entre os microcontroladores e os DSPs, uma vez que têm periféricos tal como os microcontroladores e contêm características computacionais que podem ser encontradas em DSPs.

2.6 Sistemas tempo-real

No nosso quotidiano cruzamo-nos e interagimos frequentemente com sistemas tempo-real, quando, por exemplo, usamos o telemóvel, ligamos o DVD, ou andamos de automóvel. Segundo Kopetz [23], um sistema tempo-real para além de ter que gerar resultados logicamente correctos, terá também de cumprir metas temporais, para o correcto funcionamento do sistema a controlar. Outro requisito imposto em sistemas de tempo-real é a fiabilidade, que consiste no grau de confiança com o qual podemos depender de um determinado sistema. Um erro comum quando se pensa em tempo-real é associar este a rapidez, sendo na realidade a dinâmica imposta pelo sistema a controlar. Por exemplo se for considerada uma simulação científica num super-computador, esta é extremamente rápida, podendo estar desta forma o sistema a ser simulado com uma dinâmica mais rápida que a sua dinâmica real.

Para além dos sistemas apresentados, os sistemas de tempo-real, estão fortemente presentes na indústria, na aviónica, entre outros.

2.6.1 Sistemas tempo-real *hard* e *soft*

Existem dois tipos de sistemas de tempo real, sendo estes *hard* e *soft.* Os sistemas do tipo *hard* diferem dos de tipo *soft* em vários aspectos, sendo o mais importante o impacto que tem o instante em que são produzidos os resultados. Num sistema do tipo *hard*, se falhar a *deadline*, o sistema falha completamente, podendo causar não só danos materiais como também pôr em risco vidas humanas. Um exemplo deste tipo de sistemas pode ser o controlo de um reactor nuclear. No caso dos sistemas do tipo *soft*, estes são tolerantes a falhas, uma vez que o incumprimento de uma *deadline* apenas degrada a qualidade de serviço providenciada pelo sistema. A título de exemplo um

sistema deste tipo pode ser um sistema que dá suporte a *stream* de vídeo.

Nos sistemas do tipo *hard*, os picos de carga têm que estar bem definidos. Para isto ser conseguido é necessário que o projectista conheça bem o sistema e que tenha capacidade de prever o seu comportamento em inúmeras situações. Para os sistemas do tipo *soft*, ao contrário dos sistemas do tipo *hard*, é apenas necessário garantir que o sistema tenha uma boa performance durante a maioria do tempo de funcionamento, garantindo assim uma determinada qualidade de serviço e mais baixo custo no seu desenvolvimento. Segundo Kopetz [23] um outro aspecto muito importante nos sistemas de tempo-real é a recuperação dos sistemas em caso de falhas, onde sistemas do tipo *hard,* têm que ser capazes de detectar a ocorrência de um erro e conseguir recuperar. No caso de sistemas do tipo *soft*, se ocorrer um erro, para o sistema recuperar é necessária, na maioria das vezes, a intervenção humana.

2.6.2 Escalonamento

Na maioria das aplicações de tempo-real, existem várias tarefas a concorrer pelo mesmo recurso, o que é muito frequente nos sistemas computacionais onde várias tarefas concorrem pelo processador. Para que os sistemas operem correctamente, é necessário estabelecer precedências na execução das tarefas. As técnicas que são usadas para definir esta ordem, são denominadas técnicas ou algoritmos de escalonamento [24].

No universo das tarefas existentes num sistema de tempo-real, podem existir tarefas periódicas e esporádicas. As tarefas periódicas são activadas regularmente, podendo a título de exemplo ser o controlo de injecção de combustível de um automóvel. Quanto às tarefas esporádicas, como o próprio nome indica são activadas esporadicamente, como acontece por exemplo na activação de um alarme. Abaixo estão apresentados os vários tipos de algoritmos de escalonamento.

- **Preemptivo ou Não-preemptivo**: Caso uma tarefa que se encontre a ser executada possa ou não ser, respectivamente, interrompida por uma de mais alta prioridade. O não uso de preempção apresenta a vantagem de não existirem mudanças de contexto entre as tarefas. Por sua vez, cria maior latência nas tarefas de maior prioridade. O uso de preempção cria *overhead* nas mudanças de contexto, mas reduz latência nas tarefas de maior prioridade.

- ***On-line* ou *Off-line*:** A grande diferença entre o escalonamento *on-line* e *off-line*, assenta no facto deste ser feito durante a execução do programa ou antes do programa entrar em funcionamento, respectivamente. O escalonamento *on-line*, apresenta a facilidade de albergar uma nova tarefa, mas apresenta *overhead* no

escalonamento das tarefas. Por sua vez o escalonamento *offline*, não apresenta *overhead* mas é mais difícil analisar o escalonamento aquando da inserção de uma nova tarefa.

- **Dinâmico ou Estático:** Caso a prioridade das tarefas possa ou não, respectivamente, ser alterada durante a execução de um programa. Os algoritmos de escalonamento dinâmicos, apresentam maior flexibilidade, e maior aproveitamento dos recursos que os algoritmos estáticos. Por sua vez, exigem maior carga computacional.

2.7 *Kernels* tempo-real

Os *kernels* tempo-real são um tipo de *kernel* multitarefa, que dão suporte a sistemas tempo-real. Os *kernels* tempo-real facilitam a gestão de um sistema de tempo-real, mas por si só não garantem o cumprimento das *deadlines* das tarefas. Para analisar se um conjunto de tarefas cumpre a *deadline* é necessário aplicar a análise de escalonamento alusiva ao algoritmo usado pelo *kernel*.

Quanto à gestão de tarefas, um *kernel* de tempo-real deve ser capaz de as criar, terminar, escalonar, suspender e proceder a mudanças de contexto, entre outros. Quanto às interrupções, este também tem de ser capaz de fazer a sua gestão e terá de ter políticas de atendimento às mesmas, para garantir que estas sejam tratadas e que todas as tarefas continuem a cumprir a sua *deadline*. Um *kernel* de tempo-real, terá também de ter ferramentas de sincronização, nomeadamente semáforos binários e semáforos de contagem, para garantir a exclusão mútua no acesso a recursos partilhados.

Capítulo 3

Sistema desenvolvido

Este presente capítulo apresenta o sistema desenvolvido no âmbito deste trabalho. Este servirá de base para as aplicações apresentadas nos capítulos 5 e 6, podendo para cada caso específico sofrer adaptações.

Inicialmente será apresentada uma descrição geral do sistema desenvolvido, assim como as suas características mais importantes. Seguidamente será apresentada a forma como são efectuadas as comunicações entre os nós do sistema e, por fim será apresentado todo o *hardware* que poderá ser usado em cada nó, assim como os detalhes mais importantes de cada um deles.

3.1 Descrição geral

O sistema desenvolvido no âmbito deste trabalho é um sistema de controlo distribuído baseado em Ethernet. O diagrama de blocos que o representa é apresentado na Figura 3.1.

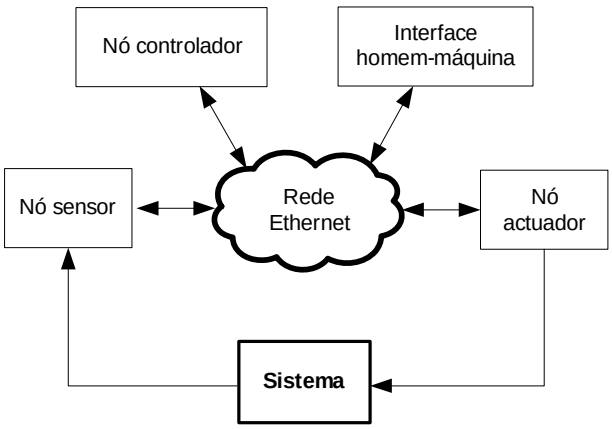


Figura 3.1: Diagrama de blocos geral do sistema desenvolvido.

Todos os nós do sistema são constituídos por uma unidade computacional, podendo esta ser um PC ou um módulo de processamento denominado de MXS4, com recursos computacionais mais reduzidos.

Os nós sensor e actuador poderão em algumas situações coexistir no mesmo nó, dependendo da aplicação à qual o sistema desenvolvido se destina. Da mesma forma os nós controlador e interface homem-máquina poderão também coexistir no mesmo nó.

3.2 Módulo de processamento MXS4

O módulo de processamento MXS4 foi desenvolvido por uma entidade exterior (Micro I/O), sendo desta forma os detalhes técnicos, nomeadamente esquema e placa de circuito impresso, confidenciais. A Figura 3.2 apresenta o diagrama de blocos deste módulo.

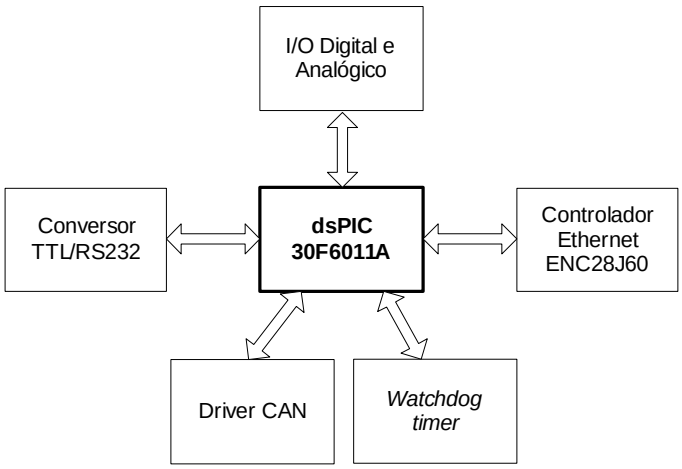


Figura 3.2: Diagrama de blocos do módulo de processamento.

O elemento central deste módulo é a unidade computacional, um dsPIC 30F6011A, cujas características técnicas serão apresentadas posteriormente. É neste elemento que são efectuadas todas as operações lógicas e aritméticas, assim como toda a gestão dos periféricos. O módulo é constituído também por uma extensão que permite o acesso a alguns dos portos de entrada/saída digitais, bem como aos portos de entrada analógicos presentes no dsPIC. Possui também um conversor TTL/RS232C que tem a cargo converter sinais TTL alusivos a uma comunicação série em sinais definidos na norma RS232C. Contém um driver CAN que tem a cargo converter um sinal TTL num sinal diferencial presente numa gama entre os -2.5 Volt e os 2.5 Volt. Ainda no que diz respeito às comunicações, este módulo contém um controlador Ethernet ENC28J60

que é responsável por todas as comunicações Ethernet efectuadas pelo dsPIC, sendo este apresentado posteriormente. Por fim, o módulo de processamento contém ainda um *watchdog timer* que tem a cargo reiniciar automaticamente o dsPIC em caso de falha do programa.

A Figura 3.3 apresenta uma fotografia do módulo de processamento MXS4.

Figura 3.3: Módulo de processamento MXS4.

3.2.1 Unidade computacional

A unidade computacional usada é, tal como referido, um dsPIC 30F6011A, da Microchip Technology Inc. que tem uma arquitectura de 16 *bits* [25], sendo as restantes características apresentadas de seguida.

- A frequência de operação do dsPIC é de 80MHz;
- 16 canais de ADC, com resolução de 12 *bits*;
- 5 *timers* de 16 *bits* cada, podendo 4 destes serem agrupados em 2, formando assim 2 *timers* de 32 *bits*;
- Controladores de comunicações: UART, SPI, CAN e I2C;
- 312 *KBytes* para o *program memory;*
- 6 *KBytes* de RAM;

- 2 *KBytes* de EEPROM;
- 2 acumuladores de 40 *bits* cada;
- Multiplicador por *hardware* de 17 *bits*;

Para proceder ao carregamento do *firmware* para este elemento, foi inserido um *bootloader* denominado de *tiny bootloader* usando a porta série para o efeito. Este permite o carregamento de *firmware* sem necessidade de programador por *hardware*. Uma vez que são usados vários módulos de processamento MXS4 em simultâneo, foi desenvolvido um *multiplexer* UART que permite o carregamento de *firmware* seleccionando apenas a porta que interliga ao módulo MXS4 a programar. Os detalhes desta ferramenta são apresentados no Apêndice A.

Para possibilitar a execução de múltiplas tarefas, foi usado o *kernel* de tempo-real FreeRTOS que é apresentado a seguir.

3.2.2 FreeRTOS

O FreeRTOS é um *kernel* de prioridades fixas implementado não só para dsPICs mas também para outros microcontroladores de outros fabricantes. A escolha deste *kernel* deveu-se ao facto de ser *freeware*, *open-source* e de ser compatível com o compilador *MPLAB Compiler C30* versão estudante.

Este *kernel* usa uma base temporal configurável, chamada de *tick*, para efectuar toda a gestão temporal das tarefas. O período do *tick* usado foi 1ms e é gerado através de um *timer* do dsPIC. Outro parâmetro importante que é igualmente configurável, é o número de prioridades diferentes que podem ser atribuídas a cada tarefa. Em todos os dsPICs usados é possível definir 3 prioridades diferentes para as tarefas. Todos estes parâmetros são configurados através do ficheiro *FreeRTOSConfig.h*. Toda a informação alusiva a esta ferramenta pode ser encontrada em [26].

3.3 Comunicação entre os nós

A comunicação entre os nós é efectuada através de *sockets* TCP. A escolha destes em detrimento de *sockets* UDP deve-se ao facto de implementarem mecanismos que garantem a entrega dos dados. Em contrapartida exigem uma maior carga de processamento, maior largura de banda e maior latência na transmissão dos dados. Estas desvantagens no contexto do sistema desenvolvido podem ser desprezadas uma vez que cada nó contém recursos computacionais capazes de os suportar, assim como os sistemas a controlar serão lentos relativamente aos atrasos que estes possam adicionar.

De seguida é apresentada a forma como estes são suportados tanto no caso em que a unidade computacional de cada nó é um PC assim como um módulo de processamento MXS4.

3.3.1 PC

Quando a unidade computacional presente num nó é um PC, os *sockets* TCP são implementados em linguagem JAVA. No caso concreto em que é necessário usar o MATLAB tanto para executar o algoritmo de controlo como para visualização de resultados, os *sockets* que dão suporte à comunicação foram desenvolvidos no âmbito do trabalho [3].

3.3.2 Módulo de processamento MXS4

Os *sockets* TCP no módulo de processamento MXS4 são suportados por um controlador Ethernet ENC28J60 que implementa as camadas física e lógica do modelo TCP/IP e pela *stack* TCP/IP da Microchip Technology Inc. que implementa as camadas superiores também do modelo TCP/IP. Estes dois módulos são apresentados de seguida.

3.3.2.1 Controlador Ethernet

O controlador Ethernet usado troca informação com o dsPIC através de um barramento SPI [27]. As características mais relevantes deste são:

- Tem integrado a camada física e a sub-camada *Medium Access Control*, fazendo esta última parte da camada lógica do modelo TCP/IP;
- Tem uma porta 10 Base-T que suporta ritmos de transmissão de 10 Mbits/s;
- É compatível com redes 10/100/1000 Base-T, uma vez que os controladores que suportam velocidades mais elevadas, se adaptam aos que suportam apenas velocidades mais reduzidas;
- Suporta *Half* e *Full-Duplex*;
- Suporta pacotes *Unicast*, *Multicast* e *Broadcast*;
- Calcula o CRC das tramas Ethernet, bem como preenche o campo *padding*, se necessário.

A Figura 3.4 apresenta uma fotografia do módulo que contém o controlador apresentado. É de salientar que este módulo faz parte do módulo de processamento MXS4.

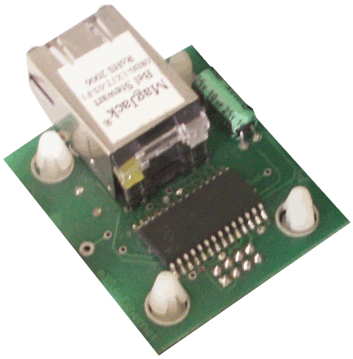


Figura 3.4: Módulo que contém o controlador Ethernet.

3.3.2.2 *Stack* TCP/IP da Microchip

A *stack* aqui apresentada é livre, *open source* e é suportada por todas as famílias de entidades computacionais da Microchip, o que lhe confere portabilidade entre dispositivos deste fabricante. Outra vantagem relativamente ao uso desta é o facto de se encontrar implementada em C para o compilador MPLAB *Compiler* C30 versão estudante, que é igualmente livre.

Para além dos *sockets* TCP usados para comunicação entre os nós, foi também usado o protocolo ICMP para testar a conectividade com os nós e o ARP para resolução de endereços MAC. Adicionalmente a *stack* apresentada disponibiliza outros protocolos, podendo a informação relativa a estes ser consultada no documento [28].

Configuração

Uma vez que a *stack* referida é implementada por forma a ser suportada por várias famílias de unidades computacionais da Microchip Technology Inc. assim como suporta vários protocolos, é necessário proceder à sua configuração. Esta configuração é efectuada através de dois ficheiros que são o *HardwareProfile.h* e o *TCPIPConfig.h*. É no *HardwareProfile.h* que são definidos todos os aspectos específicos relacionados com o *hardware,* nomeadamente frequência do *clock* a que opera a unidade computacional, assim como toda a configuração da comunicação entre a unidade computacional e o controlador Ethernet. No que diz respeito ao *TCPIPConfig.h* é neste ficheiro que são configurados todos os endereços que serão usados pela *stack*. Estes são o endereço MAC, IP, máscara de rede, endereço da *gateway*, entre outros. No que diz respeito aos endereços MAC e IP estes terão de ser atribuídos de forma cuidada dado que numa rede não podem coexistir diapositivos com o mesmo endereço MAC nem IP. Os três *bytes* mais significativos dos endereços MAC são atribuídos para que seja possível através destes determinar qual o fabricante do controlador, sendo os dos controladores da

Microchip Technology Inc. 00-04-A3, ficando a escolha dos restantes a cargo de quem projecta o sistema. O ficheiro de configuração *TCPIPConfig.h* permite ainda a escolha dos serviços que a *stack* irá disponibilizar assim como a configuração dos mesmos.

3.4 Nó controlador

O nó controlador presente no sistema desenvolvido é constituído apenas por um PC ou por um módulo de processamento MXS4 uma vez que este tem a cargo apenas executar o algoritmo de controlo.

Este nó para obter uma amostra alusiva ao sinal de saída do sistema efectua um pedido ao nó sensor. Quando este procede ao seu envio, o nó controlador procede à execução do algoritmo de controlo e, quando chega ao fim desta operação, envia o sinal de controlo para o nó actuador.

Uma vez que os nós sensor e actuador apenas se limitam a responder a comandos enviados por este, o pedido de uma amostra ao nó sensor pode ser considerado a marca temporal do sistema (*clock*), uma vez que dá início ao ciclo de controlo.

A Figura 3.5 apresenta o fluxo de informação entre o nó aqui apresentado e os restantes.

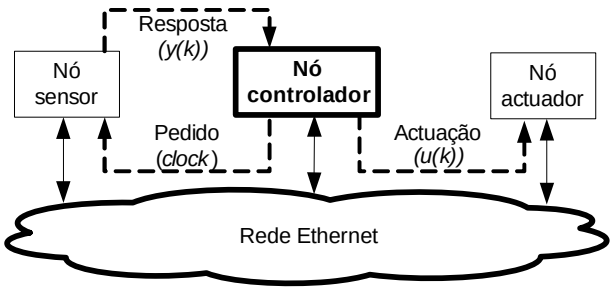


Figura 3.5: Fluxo de informação entre o nó controlador e os restantes nós.

3.5 Nó sensor

O nó sensor é responsável pela recolha de amostras alusivas à saída do sistema a controlar e proceder ao seu envio para o nó controlador, quando este lhe envia um pedido. Este módulo poderá ser constituído por um módulo de processamento MXS4 ou por um PC e por um módulo de acondicionamento de sinal. Este último poderá ser um módulo de interface analógico que visa aplicar um *offset* a um sinal de modo que este só apresente níveis de tensão positivos, ou um módulo de acondicionamento de sinal para termopares do tipo K, encontrando-se ambos descritos de seguida.

3.5.1 Módulo de interface analógico

Este módulo surge com a limitação dos ADCs do dsPIC apenas efectuarem leituras de tensões positivas. Desta forma este módulo converte um sinal presente numa gama de tensão entre os -2.5 e os 2.5 Volt num sinal presente numa gama entre os 0 e os 5 Volt, ou seja aplica um *offset* a um sinal de entrada. O diagrama de blocos deste módulo é apresentado na Figura 3.6. Este coexiste na mesma placa de circuito impresso de um módulo de actuação que será apresentado na sub-secção respectiva. O esquema eléctrico de ambos os módulos e a sua fotografia são apresentados no Apêndice B.

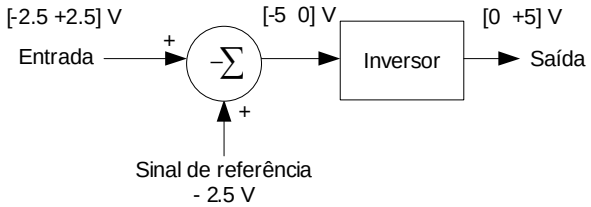


Figura 3.6: Diagrama de blocos do módulo de interface analógico (nó sensor).

Como pode ser observado no diagrama de blocos, o sinal de entrada após passar por um somador inversor dá origem a um sinal intermédio com uma gama de tensão entre os -5 e os 0 Volt. Por fim este após ser invertido encontra-se em condições de ser lido pelo ADC do dsPIC, uma vez que se encontra numa gama de tensão entre os 0 e os 5 Volt.

3.5.2 Módulo de acondicionamento de sinal de termopares

O módulo de acondicionamento de sinal de termopares visa tratar o sinal proveniente de um termopar do tipo K, para que este possa ser lido por um ADC do dsPIC. Um termopar é uma junção entre dois metais que apresenta entre estes uma diferença de potencial em função da temperatura. Estes, regra geral, são usados como sensores para medir temperaturas elevadas, tendo aplicações em monitorização da temperatura de fornos, caldeiras, entre outros. No caso concreto do termopar do tipo K, este é constituído por *chromel* e *alumel*, podendo ser usado para medir temperaturas que podem variar aproximadamente entre os 0ºC e os 1300ºC, apresentando uma sensibilidade de $40\mu V/^oC$ para 0^oC [29].

Caso se pretenda medir a diferença de potencial entre os terminais de um termopar à temperatura ambiente o valor medido será zero. Isto deve-se à existência de uma outra junção formada pelas pontas de prova do equipamento de medida e os terminais do termopar uma vez que são constituídos por metais diferentes, sendo esta denomi-

nada de junção fria, compensando o seu efeito. Desta forma, é como se de duas fontes de tensão opostas se tratasse anulando-se mutuamente. Por forma a resolver este problema, usa-se uma técnica denominada de compensação de junção fria que consiste em determinar a queda de tensão provocada pela junção fria. No caso concreto deste módulo, a compensação é efectuada através de um AD595 [30]. A escolha deste componente deveu-se ao facto de proceder à compensação de junção fria, apresentar grande imunidade ao ruído e amplificar o sinal proveniente do termopar, uma vez que este apresenta tensões bastante baixas aos seus terminais.

O diagrama de blocos deste módulo é apresentado na Figura 3.7, e o seu esquema é apresentado no Apêndice C.

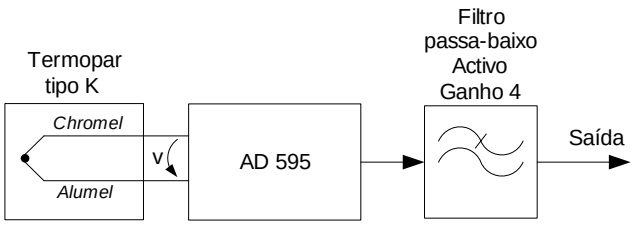


Figura 3.7: Diagrama de blocos do módulo de acondicionamento de sinal dos termopares.

Como pode ser ainda observado na Figura 3.7, este módulo é ainda constituído por um filtro passa baixo activo de ganho 4 que visa filtrar o ruído que possa estar presente à saída do AD595, assim como proceder a uma amplificação adicional.

A tensão de saída deste módulo em função da temperatura presente aos terminais do termopar é dada pela expressão 3.1.

$$Saída = (0.01 \times T(^oC)) \times 4 \quad [V] \tag{3.1}$$

3.6 Nó actuador

O nó actuador é responsável por aplicar o sinal de controlo ao sistema a controlar. Quanto ao processamento este nó pode ser constituído por um PC ou por um módulo de processamento MXS4. Do ponto de vista de electrónica de actuação este nó poderá ser constituído por um módulo de interface analógico que visa converter um sinal de (*Pulse Width Modulation*) PWM num sinal contínuo para ser aplicado ao sistema a controlar, ou poderá ser um controlador de ciclos de rede que permite controlar a potência entregue a uma carga AC. Este último é apresentado no Apêndice F.

3.6.1 Módulo de interface analógico

Este módulo é o que coexiste na mesma placa de circuito impresso do módulo apresentado na secção 3.5.1. Dada a inexistência de uma saída analógica por parte do dsPIC usado, surgiu este módulo que converte um sinal de PWM num sinal contínuo com uma tensão entre os -2.5 e os 2.5 Volt. O sinal proveniente desta conversão será o sinal de controlo aplicado ao sistema a controlar. O diagrama de blocos referente a este módulo é apresentado na Figura 3.8.

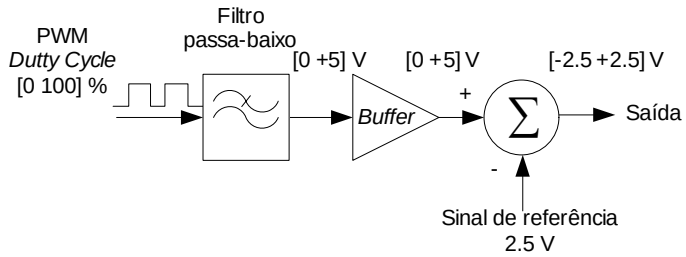


Figura 3.8: Diagrama de blocos do módulo de interface analógico (nó actuador).

A frequência do sinal de PWM inserido neste módulo é de 20KHz. Este ao passar no filtro origina um sinal contínuo que varia entre os 0 e os 5 Volt quando se aplica um sinal de PWM com um *dutty-cycle* entre os 0% e os 100% respectivamente. Para que este sinal possa ser usado externamente é necessário colocar um *buffer* para reduzir a impedância de saída. O sinal à saída do *buffer* após ser somado com um sinal de referência de -2.5 Volt, origina por fim o sinal cuja gama de tensão se encontra compreendida entre os -2.5 e os 2.5 Volt.

3.7 Interface homem-máquina

A interface homem-máquina permite a visualização de alguns parâmetros do sistema, assim como a sua configuração. No sistema desenvolvido esta interface poderá ser efectuada através do PC, mais concretamente através do programa MATLAB, ou através do módulo de processamento MXS4 conjuntamente com um módulo desenvolvido para o efeito. Este módulo é constituído por um *display* e um teclado e é apresentado de seguida.

3.7.1 Módulo *display*/teclado

Este módulo é constituído por um teclado com 5 botões e um *display* LCD com 2 linhas e 20 colunas. O Apêndice D apresenta o esquema eléctrico e a fotografia deste módulo.

Os botões que constituem o teclado, são botões de pressão que apresentam sinais com níveis lógicos *high* e *low* aos seus terminais quando se encontram pressionados ou libertos respectivamente. Desta forma, estes sinais podem ser lidos directamente pelo dsPIC, como sinais digitais, sem qualquer tipo de acondicionamento de sinal adicional.

No que diz respeito ao *display* LCD, este possui um controlador próprio. A comunicação com o controlador é feita através de dois barramentos: um barramento de 8 *bit*, bidireccional, que transporta os dados (caracteres do código ASCII) e instruções que permitem a configuração do *display* e um barramento de 3 *bit* que transporta sinais de controlo. Como parte integrante do controlador que o *display* utiliza, existe uma tabela de caracteres pré-programados disponíveis, que se encontram numa memória não-volátil. Tal como referido, este é constituído por 2 linhas e 20 colunas, formando desta forma uma matriz. Para escolher a posição onde se pretende apresentar um determinado carácter, o controlador do *display* contém uma memória volátil onde a cada endereço corresponde uma posição presente na matriz que este representa.

Capítulo 4

Controlo adaptativo

O controlo adaptativo consiste num conjunto de técnicas usadas para ajustar parâmetros de controladores para que estes mantenham uma boa performance quando o sistema a controlar apresenta pequenos desvios ao longo do tempo [9].

A Figura 4.1 apresenta o diagrama de blocos genérico de um sistema de controlo adaptativo.

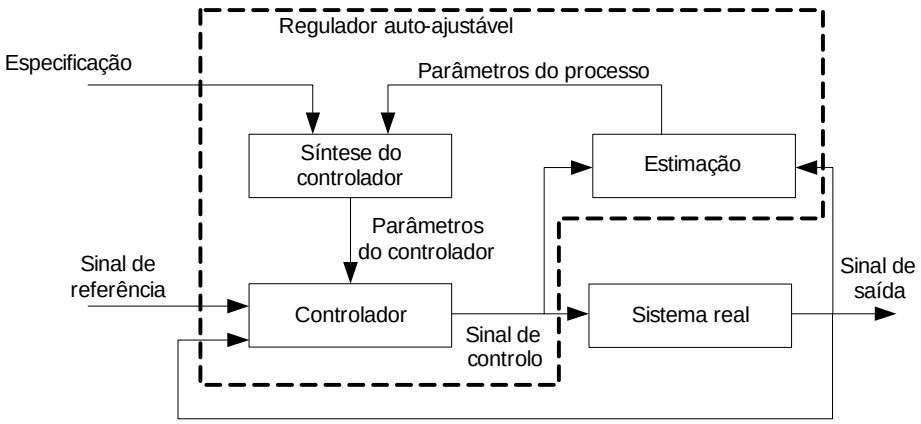


Figura 4.1: Diagrama de blocos genérico de um controlador auto-ajustável [2].

Os parâmetros do controlador são determinados com base no valor estimado dos parâmetros do sistema e nas especificações impostas pela aplicação. Estes parâmetros são determinados pelo bloco de síntese do controlador. O estimador usado terá de estimar os parâmetros em tempo-real, sendo para o efeito apresentados neste capítulo dois estimadores diferentes. A grande vantagem deste método é a flexibilidade que apresenta, visto que podem ser usadas diferentes estratégias para a síntese do controlador e do estimador.

4.1 Identificação de sistemas

A identificação de um sistema consiste em determinar um modelo matemático que o represente, com base na sua resposta a vários sinais de entrada. Este modelo terá que representar os aspectos mais importantes relativamente à dinâmica do sistema a identificar.

A identificação do modelo do sistema deve ser feita de tal forma que o erro de estimação seja mínimo. Esse erro é a diferença entre a saída do sistema real e a saída do modelo estimado. A Figura 4.2 apresenta o diagrama de blocos alusivo à determinação do erro de estimação.

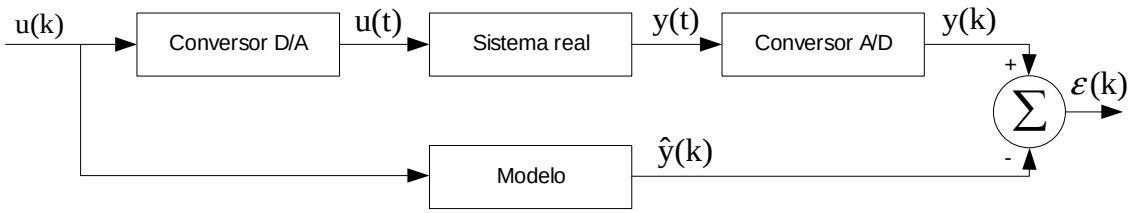


Figura 4.2: Erro de estimação.

Uma das técnicas para proceder à identificação é encontrar uma função que minimize a função de custo do quadrado do erro, encontrando-se esta apresentada na expressão 4.1.

$$J = J(y, \hat{y}) = \int (y(k) - y(\hat{k}))^2 dk = \int \varepsilon^2(k) dk \tag{4.1}$$

Dentro desta técnica existem vários métodos, dos quais será apresentado o método dos mínimos quadrados, ou *Least Squares* como é designado na literatura inglesa. Dentro deste método será apresentada uma implementação não-recursiva e uma recursiva.

4.2 Método dos mínimos quadrados

Este método foi apresentado a primeira vez por Karl Friedrich Gauss em 1795, sendo usado por este para a resolução de problemas de astronomia [31]. Desde então este método tem sido aplicado a inúmeras áreas tais como matemática e engenharia. Dentro desta última teve um enorme sucesso na estimação de parâmetros de sistemas para aplicações em controlo automático. Dentro deste método são apresentadas de seguida duas implementações.

4.2.1 Implementação não-recursiva

A implementação não recursiva do método dos mínimos quadrados permite que sejam identificados os parâmetros de um determinado sistema num modo *off-line*. Para que o sistema seja bem identificado é necessário que a estrutura do modelo capaz de aproximar o seu comportamento seja bem conhecida..

A saída de um determinado sistema é dada pela expressão 4.2.

$$y(k) = \varphi_1(k)\theta_1 + \varphi_2(k)\theta_2 + ... + \varphi_n(k)\theta_n \Rightarrow y(k) = \varphi^T(k)\theta \tag{4.2}$$

Desta forma o erro de estimação pode ser dado pela expressão 4.3.

$$\varepsilon(i) = y(i) - \varphi^T(k)\hat{\theta} \tag{4.3}$$

Caso a ordem do sistema seja conhecida, a função de custo do quadrado do erro pode ser representada por:

$$J = J(\hat{\theta}, N) = \frac{1}{2}\sum_{i=n}^{N}\varepsilon^2(i) \tag{4.4}$$

O mínimo da função de custo do erro quadrático (J) pode ser obtido através da obtenção de um nulo da primeira derivada relativa a $\hat{\theta}$. A dedução relativa a este procedimento poderá ser consultada em [31]. Depois da obtenção deste mínimo, encontra-se a expressão dos parâmetros estimados, onde a matriz Φ representa a matriz dos dados, e a matriz Y representa as amostras da saída do sistema.

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y \Rightarrow \Phi = \Phi(k) = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ ... \\ \varphi^T(N) \end{bmatrix} \tag{4.5}$$

4.2.2 Implementação recursiva

A implementação recursiva é adequada para proceder à identificação de sistemas em tempo-real tendo aplicações em algoritmos de controlo que apresentem este requisito, como o caso do controlo adaptativo. Este método, na sua forma original, é adequado essencialmente a sistemas invariáveis no tempo dado que todas as amostras têm o mesmo peso ao longo do tempo. Para que o identificador possa identificar o modelo de um sistema cujos parâmetros possam variar, surgiu uma técnica que é acrescida de um factor de esquecimento, sendo esta apresentada na subsecção 4.2.2.1. As equações

que permitem obter os parâmetros do sistema são [2] [31]:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\varepsilon(k)$$

$$\varepsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$$

$$K(k) = \frac{P(k-1)\varphi(k)}{1+\varphi^T(k)P(k-1)\varphi(k)}$$

$$P(k) = (I - K(k)\varphi^T(k))P(k-1)$$

4.2.2.1 Mínimos quadrados pesados com factor de esquecimento

Nas implementações apresentadas nos pontos anteriores todas as amostras têm o mesmo peso durante o processo de identificação do sistema. Caso o sistema seja variante no tempo faz mais sentido atribuir maior peso às amostras recentes, uma vez que contêm maior informação acerca do estado actual do sistema do que amostras mais antigas. Desta forma surgiu o método dos mínimos quadrados pesados com factor de esquecimento. O factor de esquecimento, ou λ, é um parâmetro que indica o peso atribuído às amostras mais antigas e, o valor deste é ligeiramente inferior à unidade (regra geral entre 0.95 e 0.999). Este factor deve tomar valores muito próximos da unidade para sistemas cujos parâmetros apresentem pequenas variações ao longo do tempo e valores mais próximos de 0.95 caso contrário. As expressões para determinar os parâmetros do sistema com esta implementação podem ser consultadas em [2] ou [31].

4.2.3 Escolha do intervalo de amostragem

Quando se opera com sinais que necessitam de ser representados no domínio digital, é necessário definir critérios relativamente à escolha do intervalo de amostragem. Num sistema de processamento digital de sinal, e segundo o critério de Nyquist, a frequência mínima de amostragem deverá ser duas vezes a frequência máxima do sinal a amostrar. No que diz respeito ao critério de escolha do período de amostragem em sistemas de controlo digital, esta deve estar relacionada com a largura de banda do sistema em malha fechada a controlar e, consequentemente com o seu tempo de subida igualmente em malha fechada. Regra geral, um critério usado, é obter 4 a 10 amostras durante o tempo de subida do sistema [31]. Ao contrário do que a nossa intuição nos possa levar a admitir, amostrar um sinal de saída de um sistema a controlar acima dos valores apresentados pode conduzir a maus resultados [31].

4.2.4 Sinal de entrada

Um aspecto muito importante para a identificação de sistemas é o sinal que deve ser aplicado ao sistema a identificar para que os parâmetros sejam bem estimados. Antes de se proceder à identificação do sistema através de uma técnica de identificação apresentada, é boa política aplicar um degrau a este por forma a determinar parâmetros tais como tempo de subida em malha aberta, valor máximo que a saída deste atinge, existência de tempo morto, entre outros, permitindo logo à partida escolher parâmetros tais como o intervalo de amostragem, valores mínimo e máximo que a saída do sistema pode atingir, entre outros.

Quanto ao sinal a aplicar ao sistema para proceder à sua identificação, este deve ter uma largura de banda grande para que o identificador determine com exactidão o modelo do sistema. Um sinal que preenche este requisito é o *pseuedo-random binary sequence* (prbs), que pode ser considerado o homólogo do ruído branco dos sinais digitais.

4.3 Controlo adaptativo por posicionamento de pólos

O controlo adaptativo por posicionamento de pólos, ou PPST, é um algoritmo do tipo auto-ajustável. O objectivo deste algoritmo é proceder à síntese do controlador por forma a que o sistema apresente um comportamento desejado em malha fechada, possibilitando desta forma o controlo de sistemas que apresentem variações ao longo do tempo uma vez que se adapta a estas.

O identificador usado neste algoritmo é o apresentado na subsecção 4.2.2.1.

A expressão 4.6 apresenta a função de transferência discreta para um sistema de 1ª ordem.

$$G(q) = \frac{b}{q + a} \tag{4.6}$$

A função de controlo para um sistema de 1ª ordem é apresentada na expressão 4.7, podendo a dedução desta ser consultada em [2].

$$u(k) = u(k-1) + t_0 u_c(k) + t_1 u_c(k-1) - s_0 y(k) - s_1 y(k-1) \tag{4.7}$$

Os parâmetros que se encontram a multiplicar pelos sinais do sistema, são denominados de parâmetros do regulador. As expressões que permitem a obtenção destes

são:

$$t_0 = \frac{b_m}{b}$$

$$b_m = 1 + a_m$$

$$t_1 = \frac{b_m a_o}{b}$$

$$s_0 = \frac{1-a+a_m+a_o}{b}$$

$$s_1 = \frac{a_m a_o - a}{b}$$

O parâmetro b_m determina-se de forma a que o ganho estacionário em malha fechada seja unitário. Quanto aos parâmetros a e b estes são substituídos em tempo-real pelos parâmetros estimados $\hat{a}$ e $\hat{b}$.

O parâmetro a_m é determinado através da expressão 4.8, onde o h corresponde ao período de amostragem e o α_m corresponde ao pólo pretendido em malha fechada no domínio contínuo.

$$a_m = -e^{-h.\alpha_m} \tag{4.8}$$

No que diz respeito ao parâmetro a_o este é determinado através da expressão 4.9 e corresponde ao pólo do observador. O valor deste deverá ter uma dinâmica mais rápida que a dinâmica do sistema em malha fechada.

$$a_o = -e^{-h.\alpha_o} \tag{4.9}$$

4.4 Controlo adaptativo por posicionamento de pólos com tempo morto fraccionário

O termo tempo morto fraccionário vem do facto de um sistema apresentar tempo morto cuja duração é inferior ao período de amostragem, ou no caso dos sistemas de controlo distribuído onde a transmissão dos dados pode sofrer atrasos significativos.

O controlo adaptativo por posicionamento de pólos com tempo morto fraccionário, ou PPST-TMF, tem a mesma filosofia de funcionamento que o PPST, só que considera mais um parâmetro na função de transferência discreta que representa o tempo morto. O identificador usado neste algoritmo é o apresentado na subsecção 4.2.2.1. A expressão 4.10 apresenta a função de transferência genérica para um sistema de 1ª ordem com tempo morto fraccionário.

$$G(q) = \frac{b_1 q + b_2}{q(q + a_1)} \tag{4.10}$$

A função de controlo para um sistema de 1ª ordem é apresentada na expressão 4.11, podendo a dedução desta ser igualmente consultada em [2].

$$u(k) = -(r_1 - 1)u(k-1) + r_1 u(k-2) + b_m u_c(k) + a_o b_m u_c(k-1) - s_0 y(k) - s_1 y(k-1) \tag{4.11}$$

Os parâmetros do regulador, para este controlador, são:

$$b_m = 1 + a_m$$

$$s_0 = \frac{P_1}{P_2}$$

$$s_1 = \frac{a_1(a_m + a_o) + a_1 - a_1^2 - b_1 s_0 a_1}{b_2}$$

com

$$P_0 = a_2 b_2 - b_2 + a_1 b_1$$

$$P_1 = b_2(a_m a_o + a_1) - P_0(1 - a_1 + a_m + a_o)$$

$$P_2 = b_2^2 - b_1 P_0$$

Os parâmetros a_1, b_1e b_2 correspondem aos parâmetros do sistema apresentados na expressão 4.10, sendo neste caso substituídos em tempo-real pelos parâmetros estimados $\hat{a}_1$, $\hat{b}_1$ e $\hat{b}_2$.

Quanto aos parâmetros a_m e a_o estes são determinados através das expressões 4.8 e 4.9 respectivamente.

Capítulo 5

Controlo distribuído de um sistema de 1ª ordem

Neste capítulo serão detalhados os procedimentos inerentes à utilização do sistema desenvolvido para controlo de um sistema de 1ª ordem. O controlo será efectuado usando duas abordagens distintas. Na primeira o controlador será implementado num computador usando o MATLAB e, numa segunda abordagem o controlador será implementado num módulo de processamento MXS4. Em ambos os casos os nós encontram-se interligados por uma rede Ethernet cuja velocidade é 10 Mbps.

Por fim serão comparados os resultados para verificar a viabilidade de implementar algoritmos de controlo de média complexidade, como o PPST, no dsPIC usado.

5.1 Modelação matemática do sistema

O sistema a controlar é um sistema de 1ª ordem implementado em electrónica discreta, cujo esquema é apresentado no Apêndice E. Na expressão 5.1 é apresentada a função de transferência do sistema em malha aberta no domínio contínuo.

$$G(s) = \frac{1}{s + 0.896} \tag{5.1}$$

5.2 Controlador no PC

Nesta secção será apresentada a abordagem em que o nó controlador é um PC e onde o algoritmo de controlo e executado no MATLAB. A Figura 5.1 apresenta o diagrama de blocos correspondente.

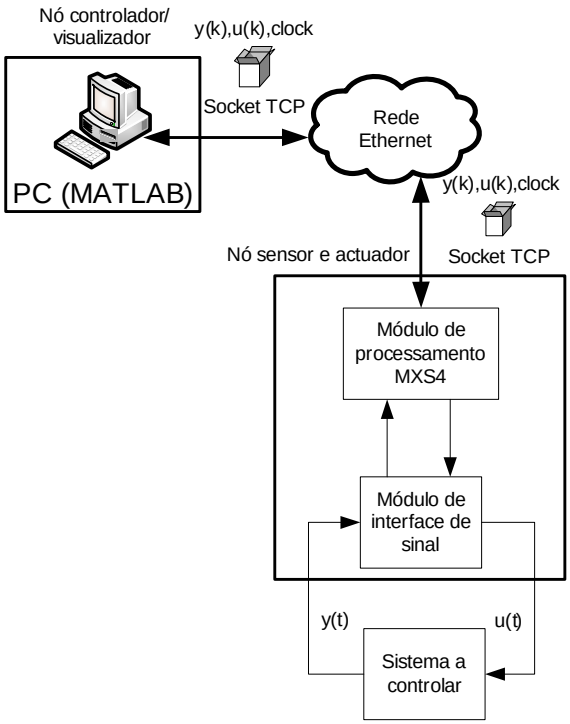


Figura 5.1: Diagrama de blocos geral do controlador no PC.

Como pode ser observado no diagrama de blocos, o módulo responsável tanto pelo acondicionamento do sinal de saída do sistema, como por aplicar o sinal de controlo a este é o módulo apresentado nas secções 3.5.1 e 3.6.1 respectivamente. Interligado a este módulo encontra-se um módulo de processamento MXS4, dando desta forma origem ao nó sensor/actuador.

No que diz respeito à troca de sinais de controlo, saída do sistema e *clock*, entre os nós, esta é efectuada através de *sockets* TCP, tal como apresentado no diagrama de blocos.

5.2.1 Algoritmo de controlo

O algoritmo de controlo usado é o algoritmo de controlo adaptativo por posicionamento de pólos, cuja descrição teórica é apresentada na secção 4.3.

5.2.2 Condições de teste

O sinal de referência usado é um *prbs*, podendo os seus patamares variar entre os -2.5 e os 2.5 Volt. No que diz respeito à rede de comunicações, esta é uma rede Ethernet dedicada.

Na Tabela 5.1 são apresentados os parâmetros do controlador.

Parâmetro do controlador	**Valor**
Período de amostragem (h)	0.2 s
Pólo em malha fechada (α_m)	1.8 rad/s
Pólo do observador (α_0)	2.0 rad/s
Factor de esquecimento (λ)	0.98
Valor inicial da matriz de covariância ($P0$)	100

Tabela 5.1: Parâmetros do controlador.

Uma vez que o tempo de subida teórico apresentado pelo sistema é de 9.85s, um período de amostragem de 0.2s preenche os requisitos apresentados na Subsecção 4.2.3, uma vez serão obtidas aproximadamente 10 amostras durante este tempo.

5.2.3 Resultados

Neste ponto serão apresentados os resultados relativos ao controlador no PC, cujas condições de teste foram apresentadas no ponto anterior.

Na Figura 5.2 são apresentados os gráficos relativos aos sinais intervenientes no controlo do sistema.

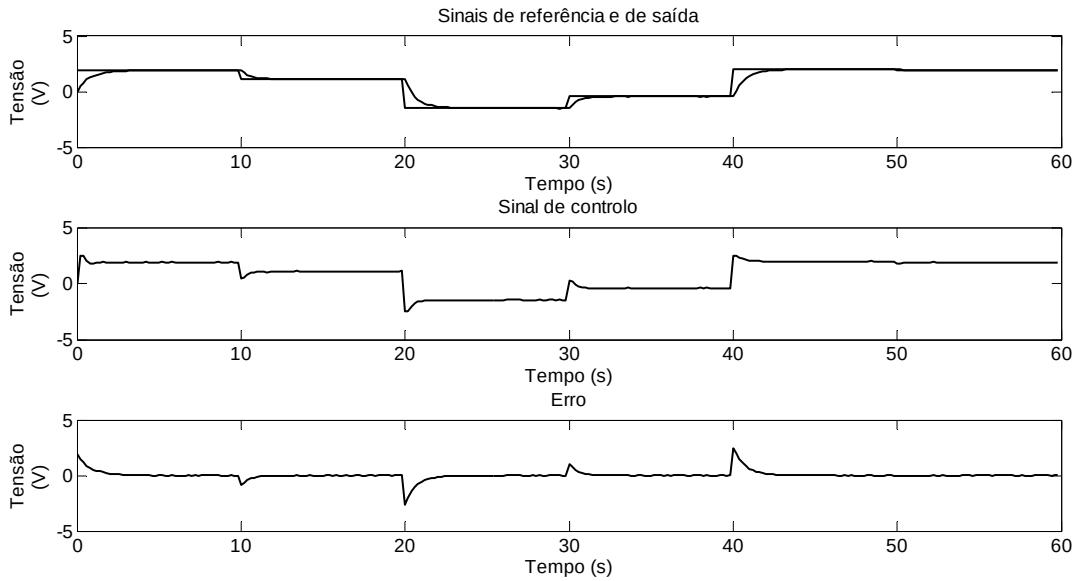


Figura 5.2: Sinais para o controlador no PC.

Como pode ser observado na figura anterior, o sinal de saída no estado estacionário de cada patamar apresenta um erro quase nulo e no regime transitório apresenta erro

devido à dinâmica do sistema. Este facto pode ser verificado tanto por comparação directa entre o sinal de referência e o sinal de saída como através do sinal de erro. No que diz respeito ao sinal de controlo este é bastante estável no estado estacionário, apresentando no regime transitório variações mais bruscas por forma a impor o pólo pretendido em malha fechada.

O erro presente nos instantes iniciais deve-se ao facto dos parâmetros do sistema ainda não terem convergido uma vez o estimador ainda não continha informação suficiente acerca do sistema. A evolução destes ao longo do tempo é apresentada no gráfico respectivo, presente na Figura 5.3.

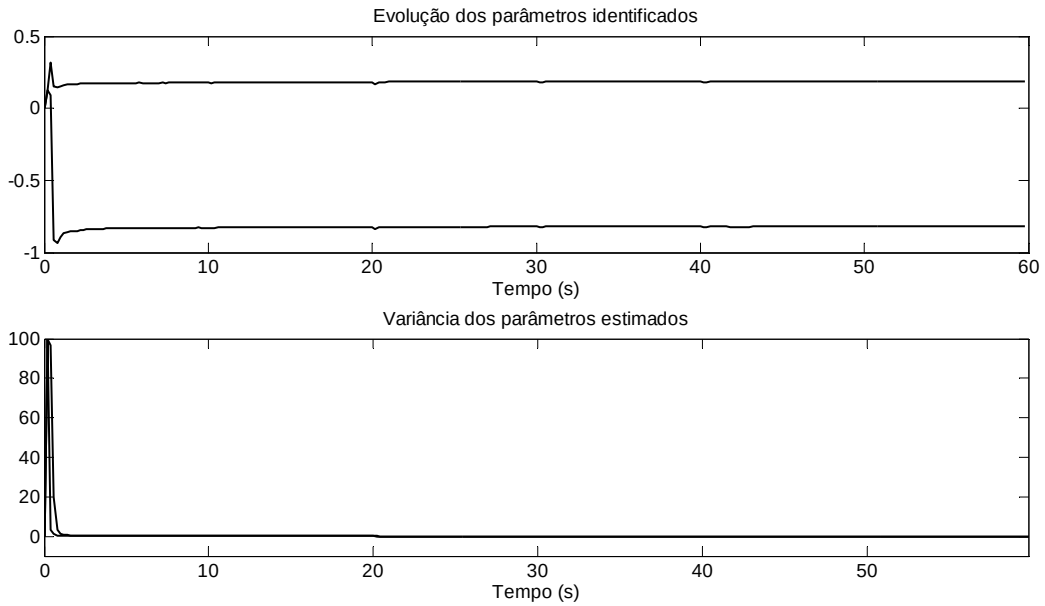


Figura 5.3: Evolução dos parâmetros estimados e da matriz de variância com o controlador no PC.

Após este período inicial os parâmetros convergem, sendo o valor destes apresentados na Tabela 5.2. Esta estabilização pode ser igualmente verificada através da convergência dos valores da matriz de covariância para zero, que pode ser observada no gráfico respectivo apresentado na Figura 5.3, uma vez que este comportamento representa a estabilização dos parâmetros estimados.

Parâmetro	**Valor**
$\hat{a}$	-0.8193
$\hat{b}$	0.1861

Tabela 5.2: Parâmetros do sistema.

Quanto ao tempo entre o instante em que é pedida uma amostra e o instante de actuação (t_{sa}) este apresenta um valor médio de 37ms, sendo este 18.5% do intervalo de amostragem. A degradação do desempenho do controlador só começa a ser sentida quando este valor é superior a 20%, logo não justifica o uso do controlador PPST-TMF [2].

5.3 Controlador no módulo MXS4

Nesta secção é apresentada a abordagem onde o nó controlador é constituído por um módulo de processamento MXS4. A Figura 5.4 apresenta o diagrama de blocos referente a esta abordagem.

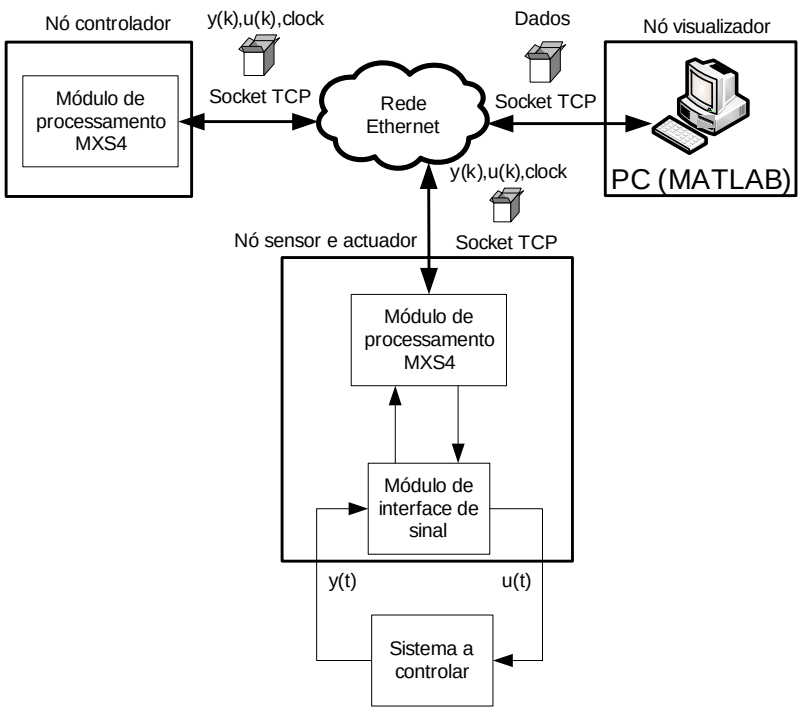


Figura 5.4: Controlador no módulo de processamento MXS4.

O nó sensor/actuador é idêntico ao usado na abordagem anterior. Quanto ao nó controlador este é constituído por um módulo de processamento MXS4. Tal como referido no Capítulo 3, é utilizado neste módulo o *kernel* FreeRTOS para fornecer a marca temporal ao algoritmo de controlo. Tal como na abordagem anterior, a troca de informação entre os nós será efectuada através *sockets* TCP. Para visualizar os resultados,

estes a cada ciclo de controlo são enviados para o nó visualizador que é constituído por um PC, onde o MATLAB neste procede à sua recolha e à sua apresentação.

5.3.1 Algoritmo de controlo

O algoritmo de controlo usado é o algoritmo de controlo adaptativo por posicionamento de pólos idêntico ao usado na abordagem anterior.

5.3.2 Condições de teste

As condições deste teste são idênticas às da abordagem anterior, encontrando-se apresentadas na Subsecção 5.2.2. Par tornar mais fácil a análise dos resultados foi usado o mesmo sinal de referência *prbs.*

5.3.3 Resultados

Neste ponto serão apresentados os resultados relativos ao controlador no módulo de processamento MXS4.

Na Figura 5.5 são apresentados os gráficos relativos aos sinais intervenientes no controlo do sistema.

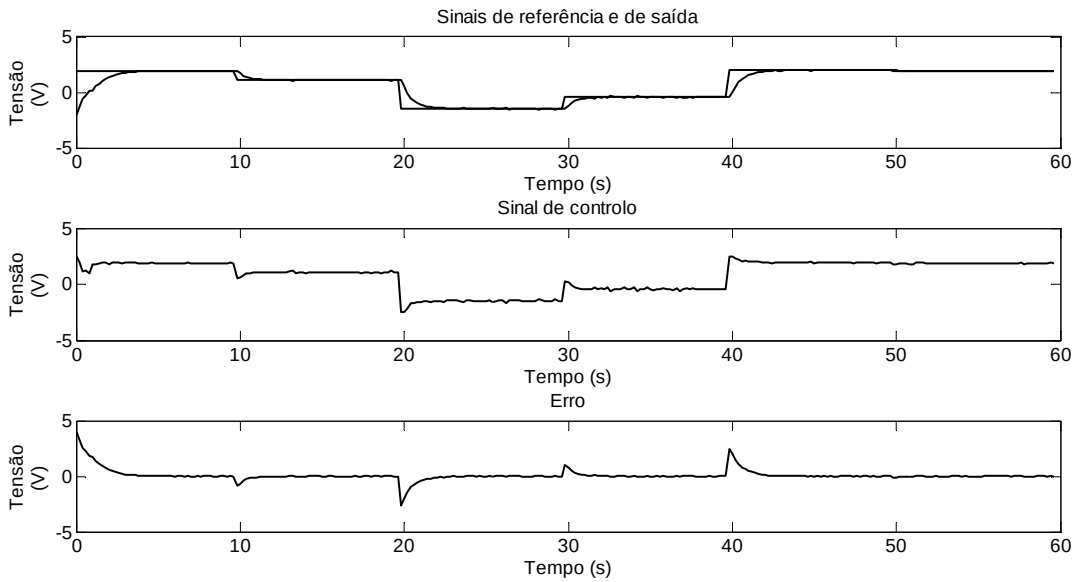


Figura 5.5: Sinais com o controlador na MXS4.

Como pode ser observado na figura anterior, o sinal de saída no estado estacionário de cada patamar apresenta um erro quase nulo e no regime transitório apresenta erro

devido à dinâmica do sistema. Este facto pode ser verificado tanto por comparação directa entre o sinal de referência e o sinal de saída como através do sinal de erro. No que diz respeito ao sinal de controlo este é bastante estável no estado estacionário, apresentando no regime transitório variações mais bruscas por forma a impor o pólo pretendido em malha fechada.

O erro presente nos instantes iniciais deve-se aos parâmetros do sistema ainda não terem convergido uma vez o estimador ainda não continha informação suficiente acerca do sistema. A evolução destes ao longo do tempo é apresentada no gráfico respectivo, presente na Figura 5.6.

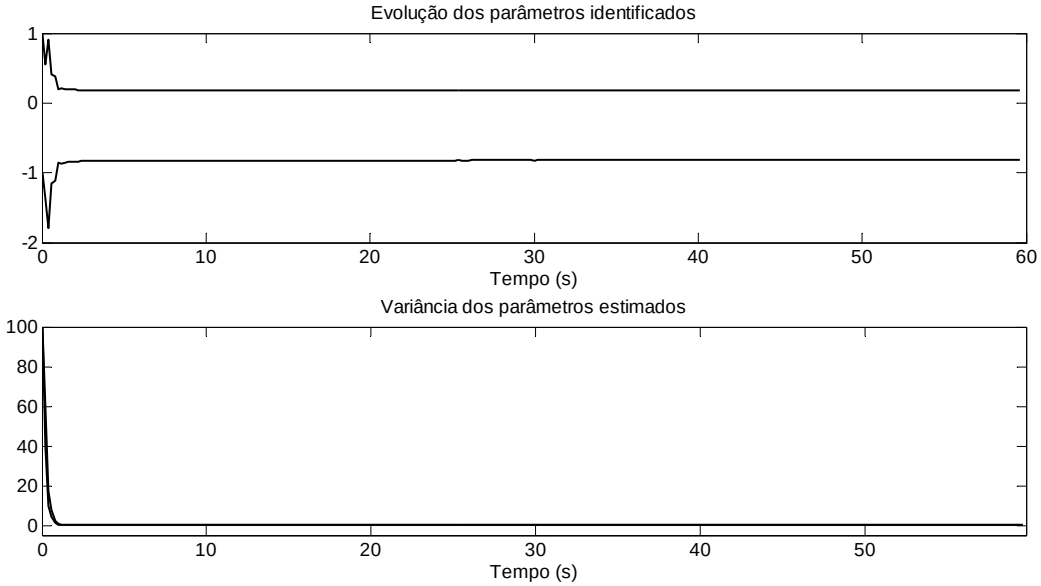


Figura 5.6: Evolução dos parâmetros estimados e da matriz de variância com o controlador na MXS4.

Após este período inicial os parâmetros convergem, sendo o valor destes apresentados na Tabela 5.3. Esta estabilização pode ser igualmente verificada através da convergência dos valores da matriz de covariância para zero, que pode ser observada no gráfico respectivo apresentado na Figura 5.6, uma vez que este comportamento representa a estabilização dos parâmetros estimados.

Parâmetro	**Valor**
$\hat{a}$	-0.8177
$\hat{b}$	0.1881

Tabela 5.3: Parâmetros do sistema.

Quanto ao t_{sa} este apresenta um valor médio de 32ms, sendo este 16% do intervalo de amostragem.

5.4 Comparação entre as duas abordagens

Comparando os resultados de ambas as abordagens conclui-se que é viável usar o dsPIC 30F6011A contido no módulo de processamento MXS4 para executar algoritmos de controlo adaptativo por posicionamento de pólos, assim como é viável usar o *kernel* de tempo-real FreeRTOS para fornecer a marca temporal ao sistema. Os factores que levaram a estas conclusões são:

- O erro apresentado por ambas as abordagens é muito semelhante, assim como o sinal de controlo;
- Os parâmetros do sistema identificados são bastante semelhantes e convergem com a mesma rapidez.
- Os valores de covariância dos parâmetros identificados convergem para zero, o que é um indicador de que ambos os estimadores se encontram a operar correctamente;
- Relativamente ao valor de t_{sa}, na segunda abordagem este é ligeiramente menor que na primeira, o que melhora ainda o controlo do sistema.

Capítulo 6

Controlo distribuído de um processo térmico

Este capítulo encontra-se dividido em duas partes, visando cobrir dois objectivos distintos. Na primeira parte será apresentado o comportamento dos algoritmos de controlo PPST e PPST-TMF num sistema com tempo morto como o processo térmico usado. Na segunda parte o objecto de estudo será verificar, tal como no capítulo anterior, a viabilidade de usar o módulo de processamento MXS4 como unidade computacional do nó controlador do sistema de controlo distribuído, tornando-se um segundo exemplo de aplicação do sistema desenvolvido.

6.1 Parte 1

Tal como referido será apresentado nesta parte o comportamento dos algoritmos PPST e PPST-TMF, quando usados para controlar um sistema com tempo morto fraccionáro, como o processo térmico usado.

6.1.1 Descrição do sistema de controlo distribuído

O sistema distribuído aqui apresentado é um sistema parcialmente distribuído, uma vez que os nós sensor e actuador coexistem no mesmo nó e os nós actuador e visualizador coexistem noutro. Quanto às unidades de processamento estas são uma MINI-ITX e um PC respectivamente, sendo a primeira um PC com recursos computacionais mais reduzidos. Quanto aos algoritmos de controlo, estes são executados no MATLAB.

O sistema aqui apresentado tem o objectivo de controlar a temperatura de saída do módulo PCT-13, cuja caracterização se encontra apresentada no Apêndice F. Como apresentado no Apêndice, este módulo é constituído por uma electroválvula que regula o

caudal de água quente e, por conseguinte, é possível através da variação deste controlar a temperatura da água de saída. O diagrama de blocos do sistema usado é apresentado na Figura 6.1.

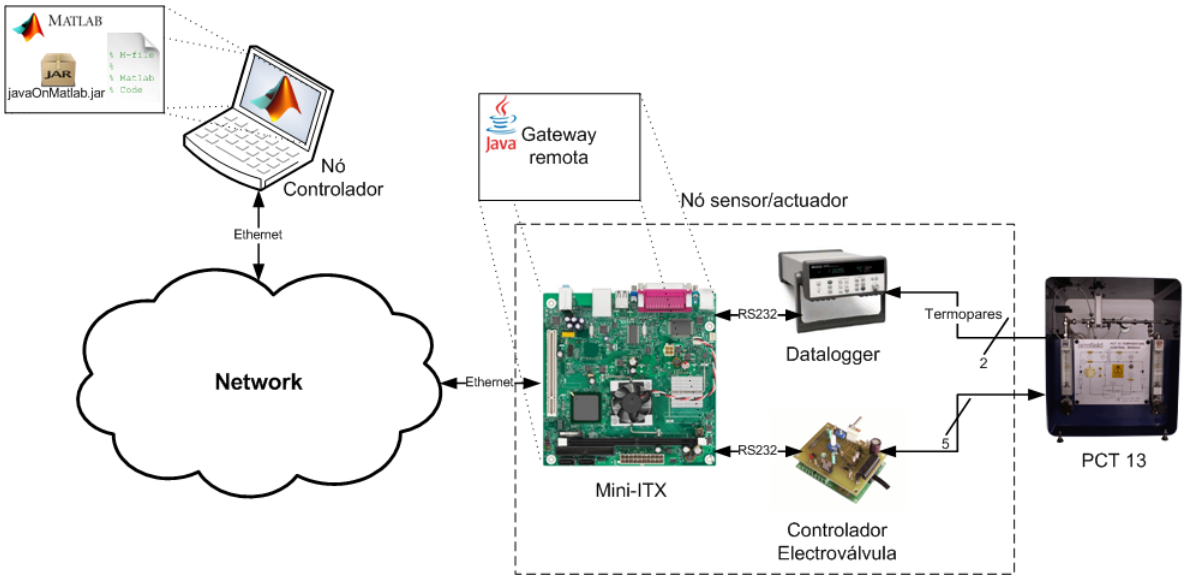


Figura 6.1: Diagrama de blocos do sistema de controlo [3].

Para controlar a posição da electroválvula é usada uma placa desenvolvida para o efeito, e para recolher amostras do valor da temperatura é usado um *datalogger*.

A arquitectura deste sistema foi desenvolvida no âmbito do trabalho [3].

6.1.2 Algoritmos de controlo

Para controlar o sistema apresentado é usado o algoritmo de controlo adaptativo por posicionamento de pólos, com e sem tempo morto fraccionário. O objectivo de usar estes dois algoritmos de controlo, tal como referido, é comparar o comportamento de ambos no controlo de um sistema que apresenta tempo morto fraccionário. O tempo morto que apresenta o subsistema electroválvula é de aproximadamente 4s, tal como referido no Apêndice F. As descrições teóricas destes algoritmos são apresentadas nas Subsecções 4.3 e 4.4 respectivamente.

6.1.3 Condições de teste

Antes de proceder a qualquer teste, foi necessário aquecer a água do subsistema depósito até uma temperatura de aproximadamente 85ºC, ficando o controlo da temperatura desta a cargo de um termostáto presente no sistema PCT13. No que diz respeito ao caudal de entrada de água fria este foi regulado para 280 cm^3/min. O sinal de referência

aplicado foi também um *prbs* cujo níveis mínimos e máximos são, respectivamente 45ºC e 55ºC.

Quanto aos parâmetros de ambos os controladores, estes são apresentados na Tabela 6.1.

Parâmetro do controlador	**Valor**
Período de amostragem (h)	10 s
Pólo em malha fechada (α_m)	0.05 rad/s
Pólo do observador (α_0)	0.1 rad/s
Factor de esquecimento (λ)	0.99
Valor inicial da matriz de covariância ($P0$)	10

Tabela 6.1: Parâmetros dos controladores usados para controlar o processo térmico.

6.1.4 Resultados

Nesta subsecção serão apresentados os resultados relativos aos dois algoritmos de controlo usados.

6.1.4.1 Controlador adaptativo por posicionamento de pólos

Nesta subsecção são apresentados os resultados relativos ao controlo do subsistema electroválvula, tendo este sido submetido às condições de teste apresentadas. O controlador usado neste teste foi o PPST.

A Figura 6.2 apresenta os sinais intervenientes no controlo do sistema apresentado.

Como pode ser observado nos sinais apresentados, a saída do sistema apresenta oscilação em alguns patamares, nunca chegando a estabilizar nos mesmos. Este facto deve-se ao sinal de controlo não ser bem determinado uma vez que sistema possui tempo morto fraccionário de aproximadamente 4s e o algoritmo de controlo usado não está preparado para lidar com esta característica, uma vez que representa 40% do período de amostragem.

No que diz respeito aos parâmetros estimados, estes inicialmente apresentam grandes variações, tal como pode ser observado na Figura 6.3, uma vez que o estimador ainda não contém informação suficiente acerca do sistema. Após esta fase, estes convergem para os valores apresentados na Tabela 6.2. A partir do gráfico da variância apresentado na Figura 6.3 é possível também verificar que os parâmetros estabilizam, dado que a convergência da matriz de variância é indicador de estabilização dos parâmetros identificados.

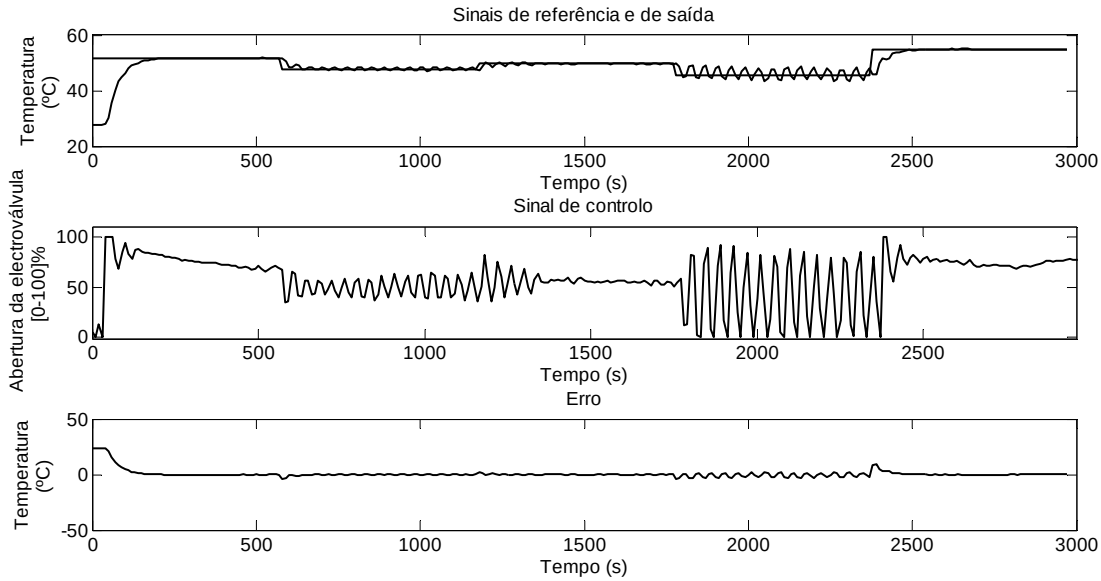


Figura 6.2: Sinais para o controlo do processo térmico com controlador PPST.

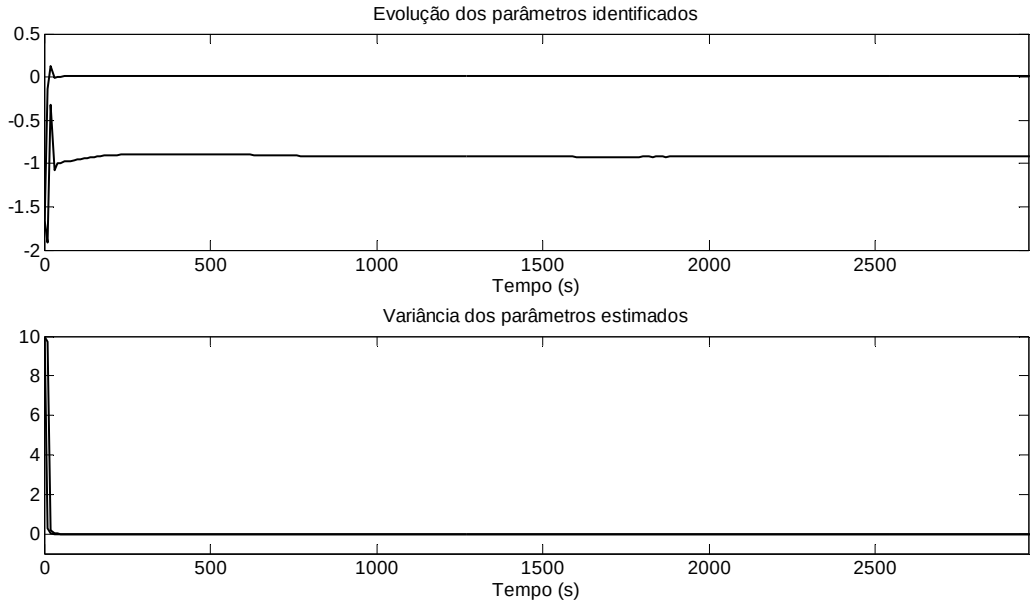


Figura 6.3: Evolução dos parâmetros estimados e da matriz de variância para controlador PPST.

Parâmetro	**Valor**
$\hat{a}$	-0.9168
$\hat{b}$	0.0116

Tabela 6.2: Parâmetros do sistema para o controlador PPST.

6.1.4.2 Controlador adaptativo por posicionamento de pólos com tempo morto fraccionário

Nesta subsecção são apresentados os resultados relativos ao controlo do subsistema electroválvula, usando o controlador PPST-TMF.

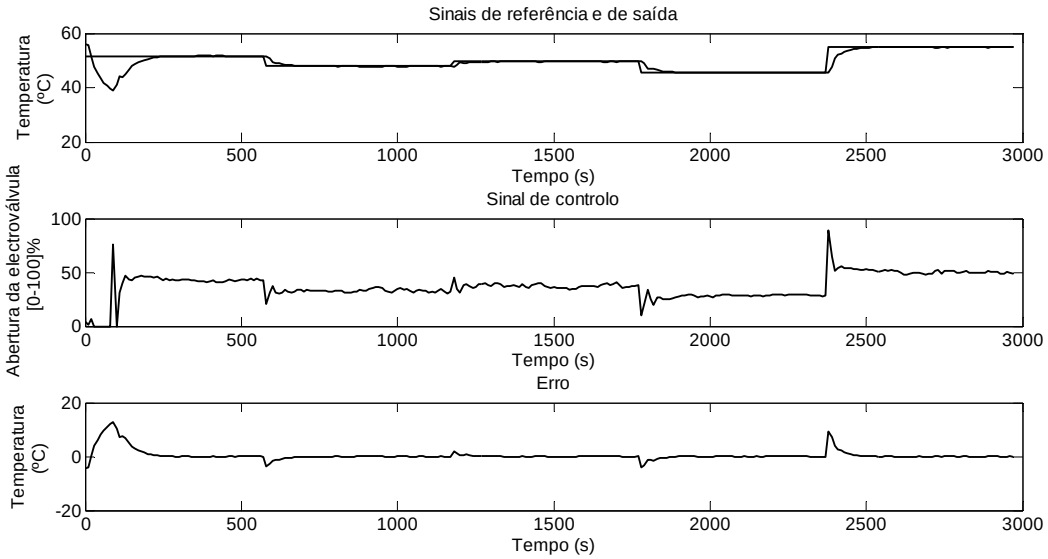


Figura 6.4: Sinais para o controlo do processo térmico com controlador PPST-TMF.

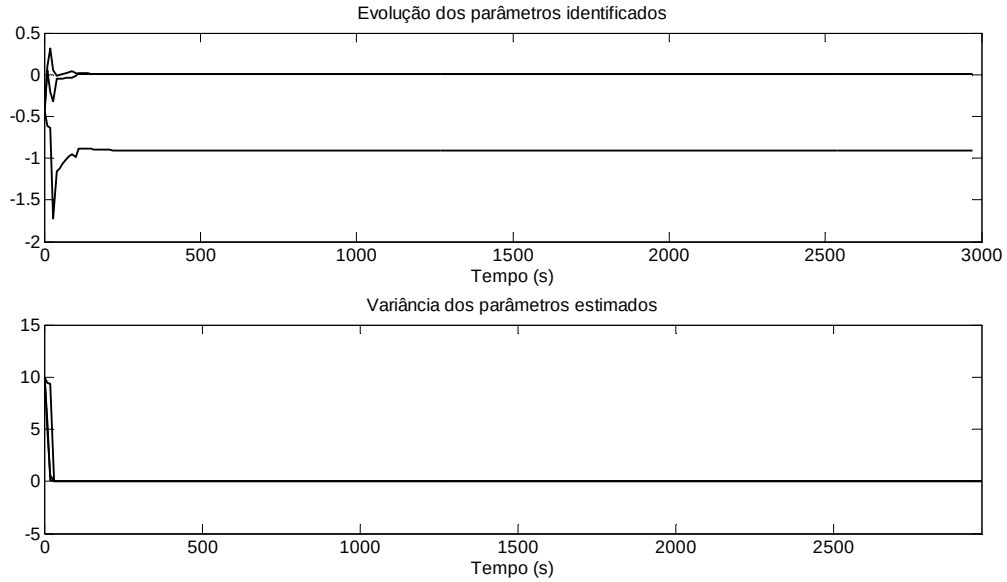


Figura 6.5: Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF.

A Figura 6.4 apresenta os sinais intervenientes no controlo do processo apresentado.

O erro presente nos instantes iniciais deve-se ao facto dos parâmetros do sistema ainda não terem convergido uma vez o estimador ainda não continha informação suficiente acerca do sistema. A evolução destes ao longo do tempo é apresentada no gráfico respectivo, presente na Figura 6.5.

Após este período inicial os parâmetros convergem, sendo o valor destes apresentados na Tabela 6.3. Esta estabilização pode ser igualmente verificada através da convergência dos valores da variância dos parâmetros estimados para zero, uma vez que este comportamento representa a estabilização dos parâmetros estimados. A evolução da variância pode ser observada no gráfico respectivo apresentado também na Figura 6.5.

Parâmetro	**Valor**
$\hat{a}_1$	-0.9100
$\hat{b}_1$	0.0066
$\hat{b}_2$	0.0088

Tabela 6.3: Parâmetros do sistema para o controlador PPST-TMF.

Comparando os resultados obtidos em ambos os controladores, conclui-se que em sistemas com tempo morto fraccionário grande relativamente ao período de amostragem, o algoritmo PPST não apresenta resultados satisfatórios quando comparado com o algoritmo PPST-TMF. Desta forma justifica-se o uso deste último no controlo deste tipo de sistemas.

6.2 Parte 2

Aqui serão apresentados dois sistemas completamente distribuídos para controlar o subsistema electroválvula já apresentado. A diferença entre ambos é o nó controlador, onde no primeiro é um PC e o algoritmo é executado no MATLAB e, no segundo é um módulo de processamento MXS4.

6.2.1 Descrição geral - controlador no PC

O sistema aqui apresentado é um sistema de controlo completamente distribuído. As unidades de processamento são um módulo de processamento MXS4 para os nós sensor e actuador e, um PC para os nós controlador e visualizador. Quanto ao algoritmo de controlo, este é executado no MATLAB. Este é um exemplo de aplicação do sistema

desenvolvido e visa controlar o mesmo sistema apresentado em 6.1.1. O diagrama de blocos que representa este exemplo de aplicação é apresentado na Figura 6.6.

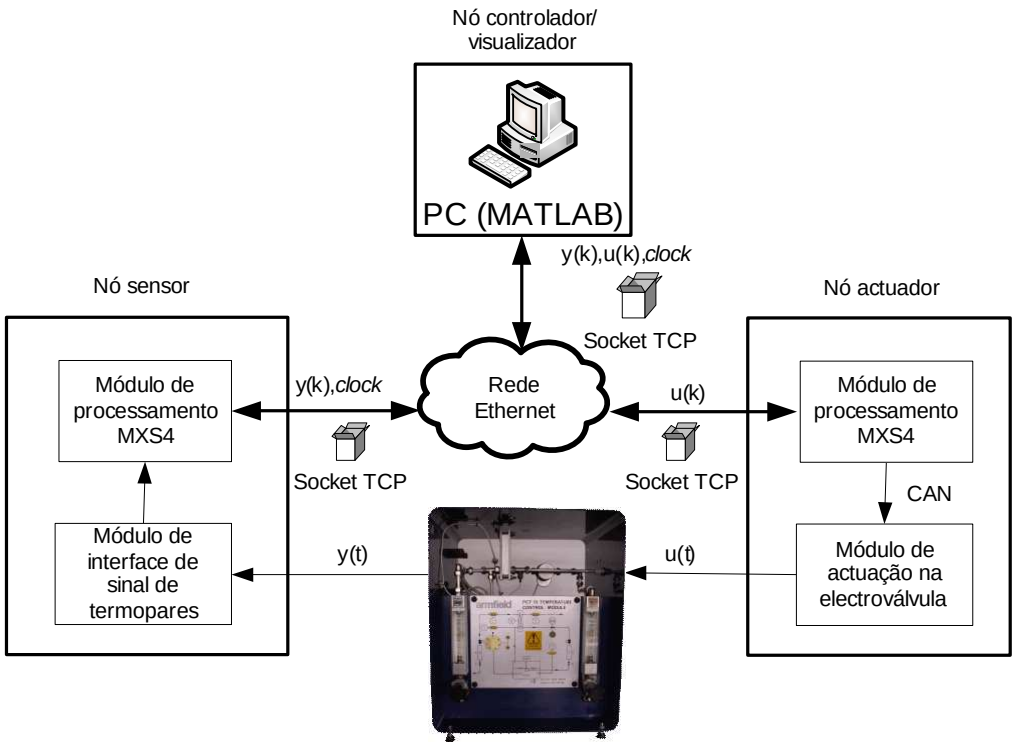


Figura 6.6: Sistema totalmente distribuído

Como pode ser observado no diagrama de blocos, o nó sensor para além de um módulo MXS4, é também constituído por um módulo de acondicionamento de sinal de termopares que visa tratar o sinal proveniente destes para que possa ser lido por um ADC do dsPIC. Quanto ao nó actuador, este tem presente um módulo de processamento MXS4 que opera como uma interface Ethernet/CAN que é responsável pelo envio do sinal de controlo através de um barramento CAN para o módulo de actuação na electroválvula.

6.2.1.1 Algoritmo de controlo

Para controlar o sistema apresentado foi usado o algoritmo de controlo PPST-TMF, uma vez que foi o que apresentou melhores resultados, relativamente ao algoritmo de controlo PPST.

6.2.1.2 Condições de teste

As condições de teste usadas foram idênticas às apresentadas na subsecção 6.1.3.

6.2.1.3 Resultados

Nesta subsecção são apresentados os resultados relativos a este teste, sendo apresentado na Figura 6.7 os sinais resultantes deste.

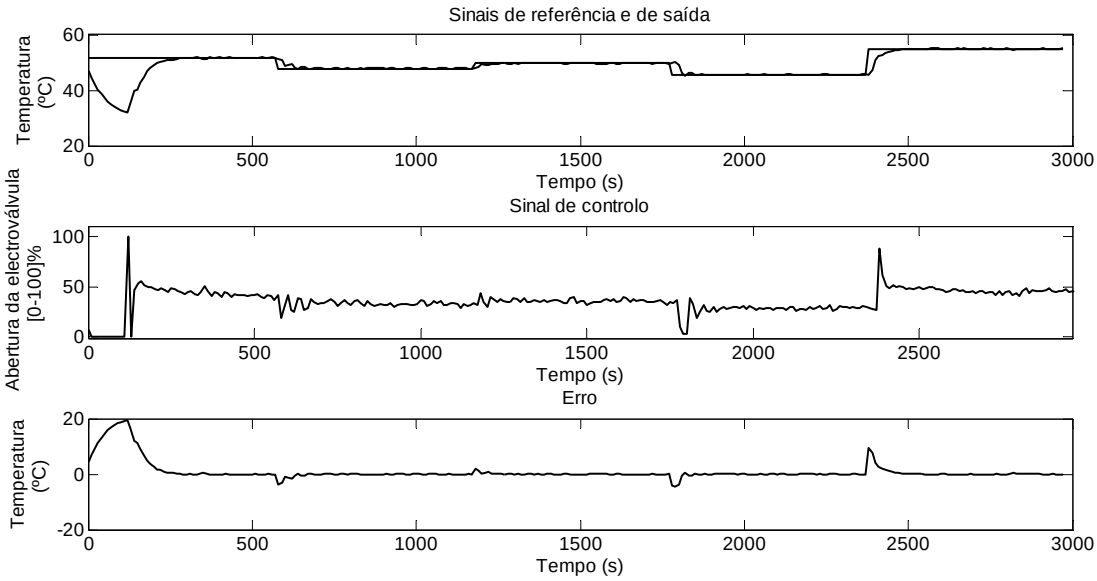


Figura 6.7: Sinais para o controlo do processo térmico com controlador PPST-TMF. Controlador no PC.

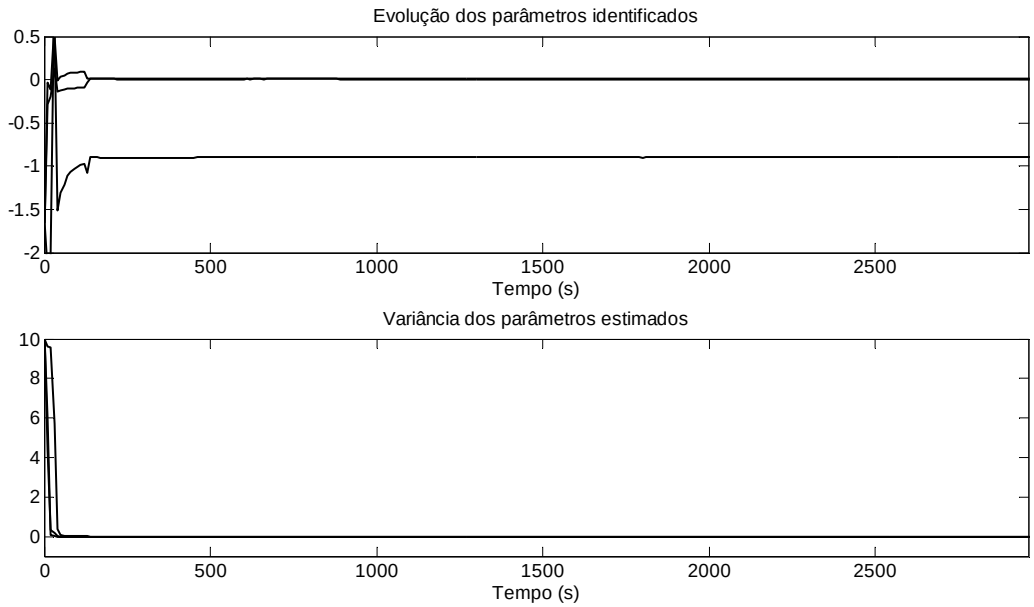


Figura 6.8: Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF. Controlador no PC.

Como pode ser observado no sinal de saída do sistema, este apresenta um comportamento idêntico ao apresentado na Figura 6.4, apresentando só um pouco mais de ruído que se deve ao facto do circuito de acondicionamento de sinal de termopares não ter a mesma capacidade de rejeição ao ruído que o *datalogger*. Quanto aos parâmetros estimados, a evolução destes é apresentada na Figura 6.8, assim como a evolução da variância dos mesmos.

Como pode ser observado, os parâmetros convergem com a mesma rapidez que os apresentados na Figura 6.5, assim como a variância destes apresenta o mesmo comportamento. A Tabela 5.3 apresenta os valores dos parâmetros identificados.

Parâmetro	**Valor**
$\hat{a}_1$	-0.8951
$\hat{b}_1$	0.0068
$\hat{b}_2$	0.0117

Tabela 6.4: Parâmetros do sistema para o controlador PPST-TMF. Controlador no PC.

As variações apresentadas entre estes parâmetros e os apresentados na sub-secção 6.4.1.2 devem-se a pequenas variações no caudal de entrada, na temperatura do subsistema depósito do módulo PCT13, variações da temperatura ambiente, entre outros.

O tempo de execução do ciclo de controlo completo é em média de 0.339s, ficando muito abaixo do período de amostragem. A grande contribuição para este valor é a leitura do sinal de saída através de um ADC do dsPIC presente no módulo MXS4. Este valor quando comparado com o tempo morto apresentado pelo sistema pode ser desprezado.

6.2.2 Descrição geral - controlador no módulo MXS4

O sistema aqui usado apenas difere do apresentado em 6.2.1 no nó controlador e no visualizador. O nó controlador neste caso é um módulo de processamento MXS4, e tal com já referido, o *kernel* FreeRTOS é responsável for fornecer a marca temporal. Quanto ao nó visualizador, é um PC, que executa o MATLAB para visualização dos gráficos. O diagrama de blocos que representa esta abordagem é apresentado na Figura 6.9.

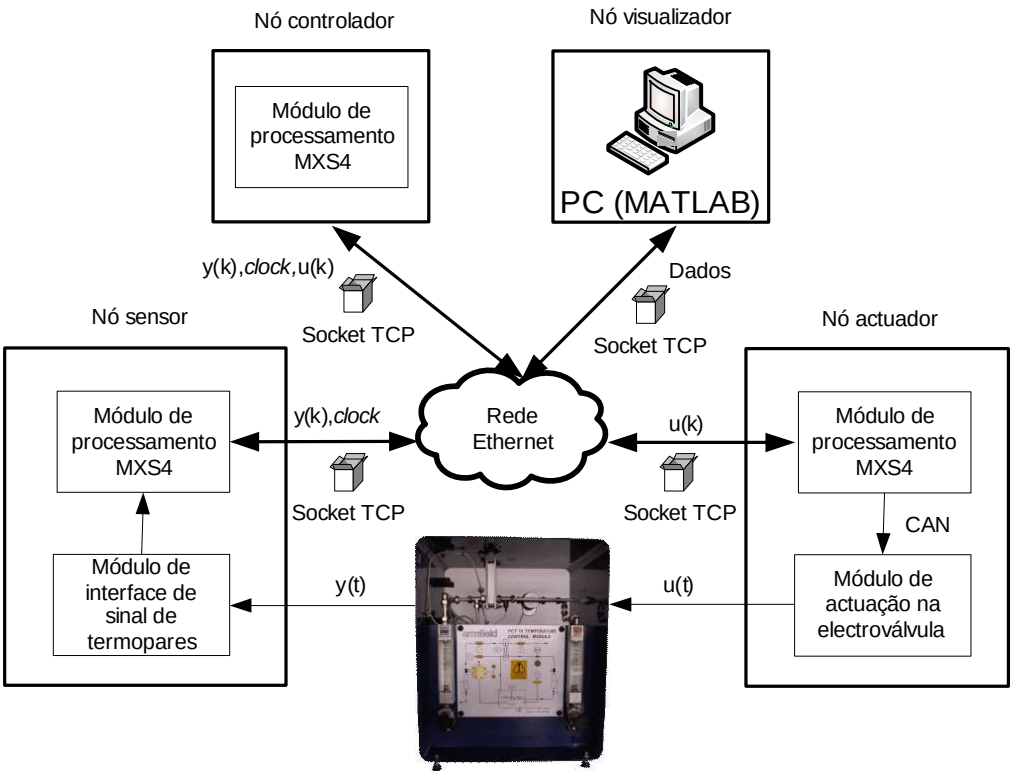


Figura 6.9: Diagrama geral do sistema cuja unidade de processamento é um módulo MXS4.

6.2.2.1 Algoritmo de controlo

O algoritmo de controlo usado é idêntico ao usado na abordagem anterior.

6.2.2.2 Condições de teste

As condições de teste usadas foram as apresentadas na subsecção 6.1.3.

6.2.2.3 Resultados

Nesta subsecção são apresentados os resultados relativos a este teste, sendo apresentado na Figura 6.10 os sinais resultantes deste.

Como pode ser observado nos sinais apresentados, estes são muito idênticos aos apresentados na subsecção 6.2.1.3. As pequenas variações que estes apresentam devem-se igualmente ao módulo de acondicionamento de sinal do termopar. Quanto aos parâmetros estimados estes após convergirem mantêm-se em média nos valores apresentados na Tabela 6.5. A evolução destes assim como a evolução da sua variância são apresentadas nos gráficos presentes na Figura 6.11.

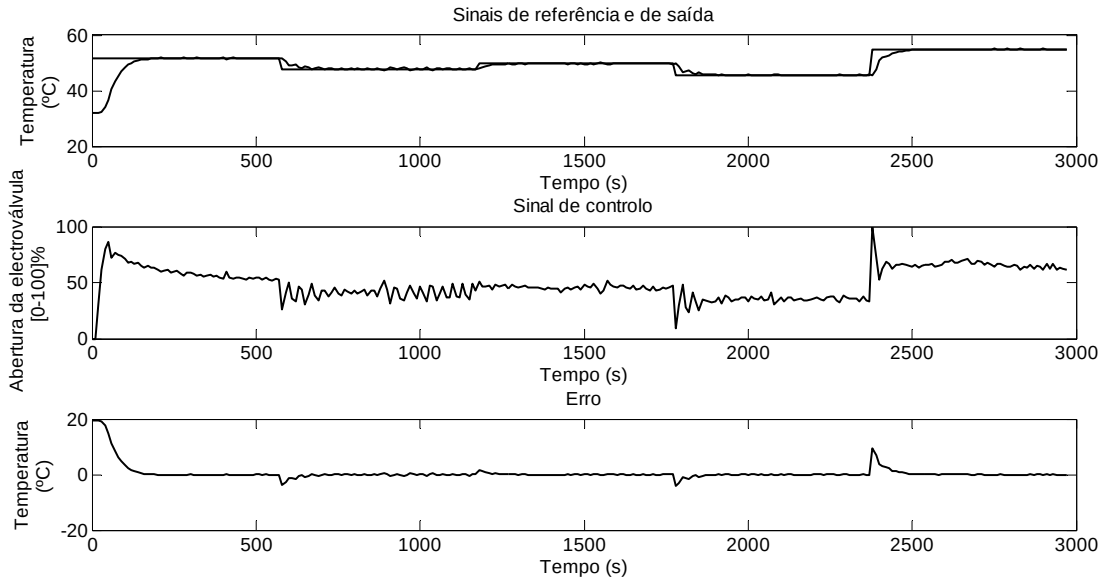


Figura 6.10: Sinais para o controlo do processo térmico com controlador PPST-TMF. Controlador no módulo MXS4.

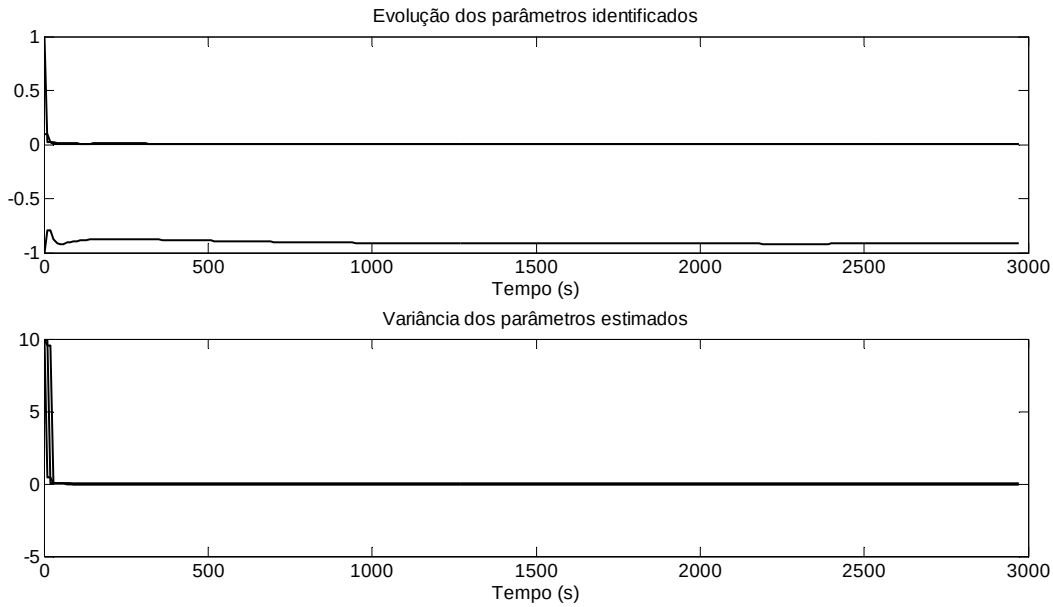


Figura 6.11: Evolução dos parâmetros estimados e da matriz de variância para controlador PPST-TMF. Controlador no módulo MXS4.

Parâmetro	**Valor**
$\hat{a}_1$	-0.9078
$\hat{b}_1$	0.0061
$\hat{b}_2$	0.0080

Tabela 6.5: Parâmetros do sistema para o controlador PPST-TMF. Controlador no módulo MXS4.

Os valores dos parâmetros apresentam pequenas variações relativamente aos anteriores pelas mesmas razões apresentadas na subsecção 6.2.1.3.

O tempo de execução do ciclo de controlo é em média 0.301s, podendo igualmente ser desprezado quando comparado com o período de amostragem e com o tempo morto. Através dos resultados apresentados para esta abordagem, é possível concluir que é viável usar o módulo de processamento MXS4 como nó controlador para controlo do sistema apresentado.

Capítulo 7

Conclusão e trabalho futuro

Neste capítulo é feita uma apreciação global do trabalho realizado, tendo em conta os resultados obtidos. No final deste capítulo são apresentadas propostas de trabalho futuro.

7.1 Conclusões

O sistema desenvolvido, apresentado no Capítulo 3, permitiu ser aplicado no controlo de dois sistemas distintos. No controlo do sistema de 1ª ordem, apresentado no capítulo 5, verificou-se um excelente desempenho do controlador tanto no MATLAB como no módulo de processamento MXS4. Relativamente aos resultados obtidos em ambas as abordagens, estes são muito idênticos o que viabiliza o uso tanto do dsPIC presente no módulo de processamento, como do *kernel* de tempo-real FreeRTOS em nós do sistema de controlo distribuído. Relativamente ao uso da Ethernet como canal de comunicação entre os nós, esta apresentou um bom desempenho, o que leva a concluir que poderá ser usada em sistemas de controlo distribuído de tempo-real do tipo *soft*. Quanto ao algoritmo de controlo adaptativo por posicionamento de pólos usado, este pode ser usado em sistemas de controlo distribuído que não apresentem um valor de t_{sa} relativamente grande.

Quanto ao controlo de um processo térmico, apresentado no Capítulo 6, surge a necessidade de usar um controlador adaptativo por posicionamento de pólos com tempo morto fraccionário, uma vez que o sistema apresenta um tempo morto de 40% relativamente ao período de amostragem. Quanto à viabilidade do uso da rede Ethernet como infra-estrutura de comunicações no controlo deste processo conclui-se que é também viável o seu uso. Quanto ao dsPIC, este apresentou igualmente bons desempenhos, o que mais uma vez viabiliza o seu uso em nós de sistemas de controlo distribuído.Para

que o identificador possa identificar o modelo de um sistema cujos parâmetros possam variar, surgiu uma técnica que é acrescida de um factor de esquecimento, sendo esta apresenta

7.2 Trabalho futuro

Relativamente aos testes efectuados, poderão ser feitos testes aos algoritmos de controlo em situações que a rede Ethernet se encontra sobrecarregada, para dimensionar o impacto desta situação, visto que os testes efectuados foram em situações normais de operação da rede.

Do ponto de vista de exploração do sistema desenvolvido, poderão ser criados outros tipos de sistemas baseados em electrónica discreta, como o usado no Capítulo 5 e, desta forma é possível testar facilmente novos algorimos de controlo em sistemas de controlo distribuído baseados em Ethernet.

No que diz respeito à Ethernet, poderá ser explorado o uso de protocolos com propriedades tempo-real baseados nesta e, desta forma atribuir propriedades temporais mais restritas ao sistema desenvolvido.

Apêndice A

Multiplexer UART

Luís Farinha, Rui Sancho
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
Maio 2009

O multiplexer UART foi uma ferramenta desenvolvida para carregar o *firmware* para um determinado módulo de processamento seleccionando apenas, através de um botão, a porta ao qual este está ligado.

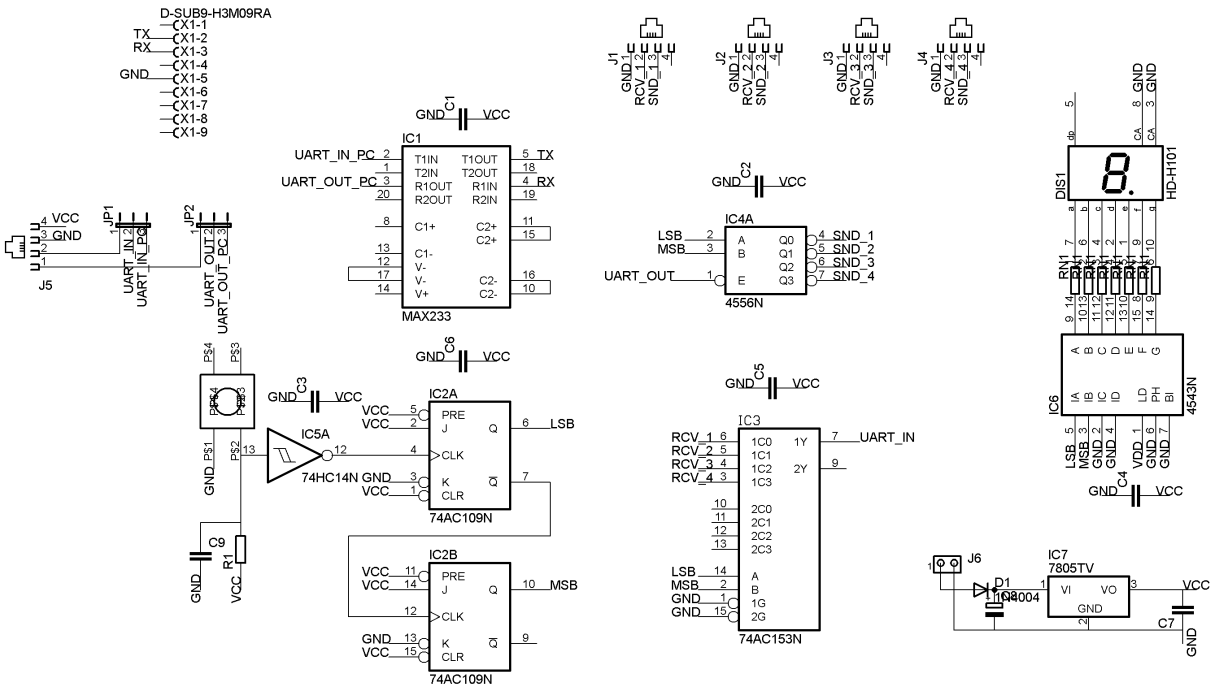


Figura A.1: Esquema eléctrico do multiplexer UART.

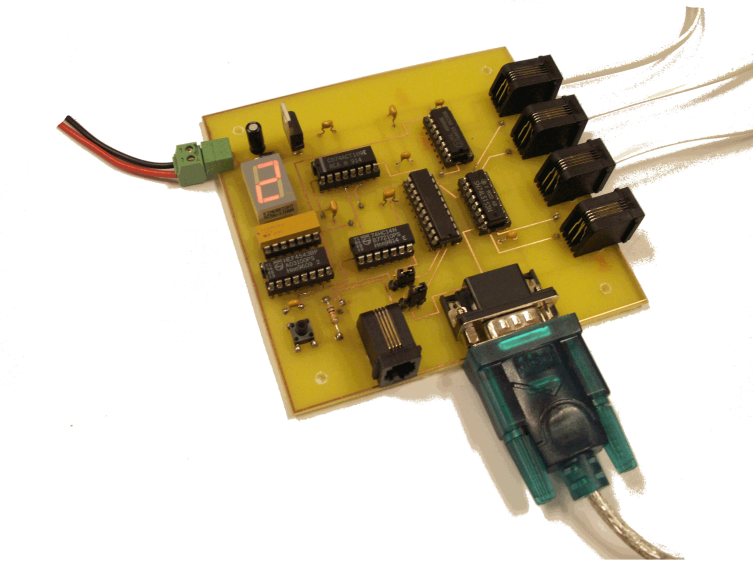


Figura A.2: Multiplexer UART.

Esta ferramenta é constituída por um botão que permite a selecção de uma das 4 portas de saída que se pretende usar, sendo esta selecção feita de uma forma incremental. O número da porta usada é apresentado no *display* de 7 segmentos. Este módulo tem duas portas de entrada distintas, sendo uma UART e uma RS-232C o que acresce a sua flexibilidade. A sua selecção é efectuada através de dois *jumpers* presentes entre as mesmas. A Figura A.3 apresenta a posição em que estes se devem encontrar para cada porta.

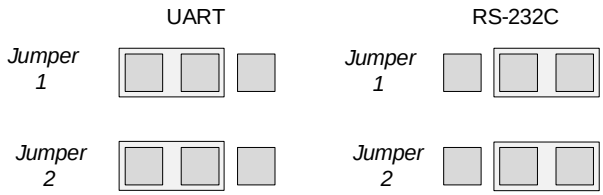


Figura A.3: *Jumpers* para selecção da porta de entrada.

Apêndice B

Módulos de interface de sinal (Nó sensor e actuador)

Luís Farinha, Rui Sancho
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
Maio 2009

Este apêndice apresenta o esquema eléctrico assim como uma fotografia do módulo de interface analógico presente no nó sensor e no nó actuador. O esquema eléctrico encontra-se na Figura B.2 e a Figura B.1 apresenta uma fotografia da placa de circuito impresso.

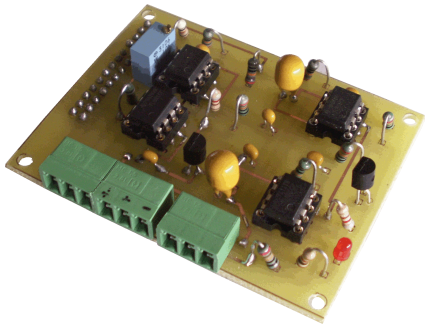


Figura B.1: Módulos de interface analógico.

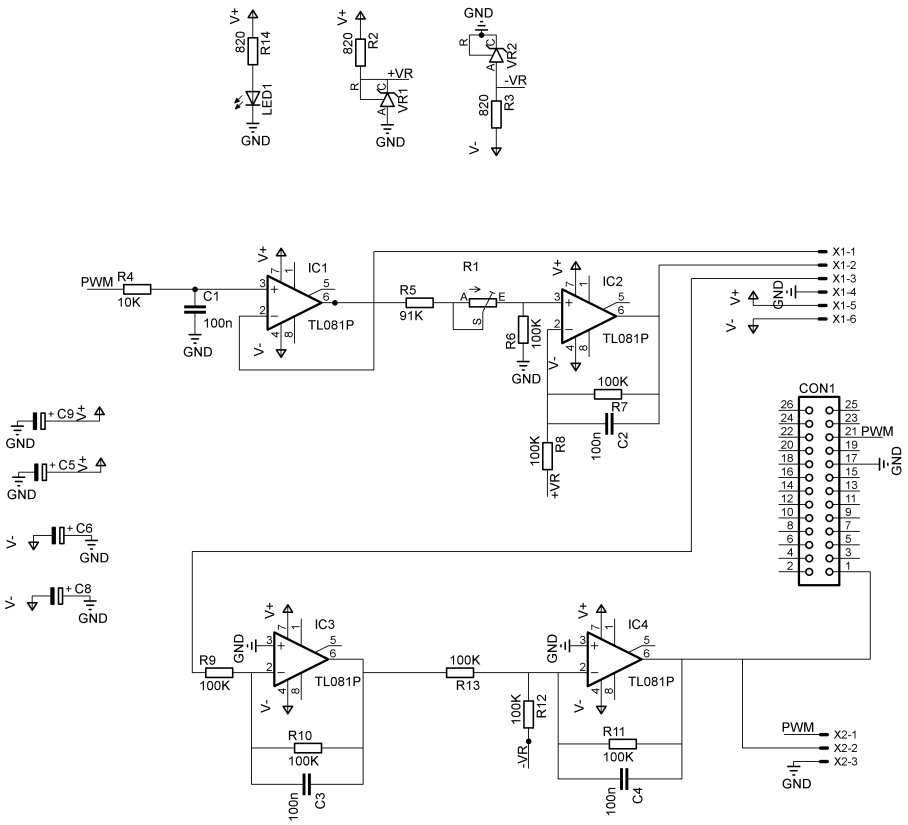


Figura B.2: Esquema eléctrico dos módulos de interface analógico.

Apêndice C

Módulo de acondicionamento de sinal de termopares

Luís Farinha, Rui Sancho
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
Maio 2009

Este apêndice apresenta o esquema eléctrico do módulo de acondicionamento de sinal de termopares.

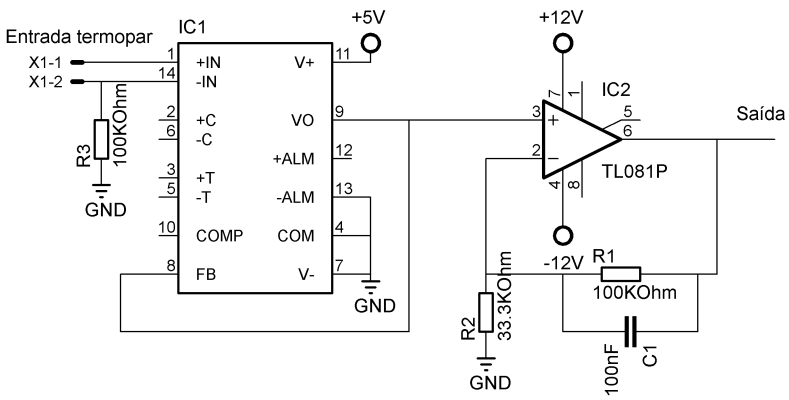


Figura C.1: Esquema eléctrico do módulo de acondicionamento de sinal de termopares.

Apêndice D

Módulo *display*/botões

Este apêndice apresenta o esquema eléctrico assim como uma fotografia do módulo *display*/botões. O esquema eléctrico encontra-se na Figura D.1 e a sua fotografia é apresentada na Figura D.2.

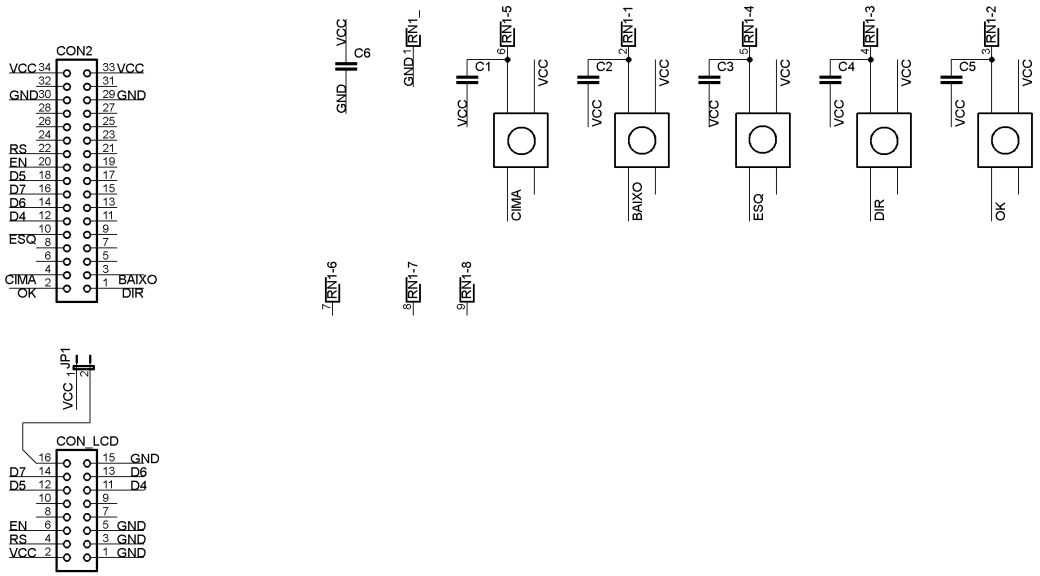


Figura D.1: Esquema eléctrico do módulo *display*/botões.

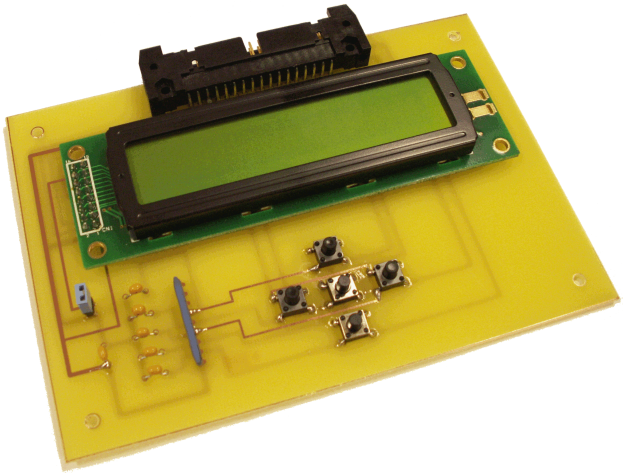


Figura D.2: Módulo *display*/botões.

Apêndice E

Esquema eléctrico do sistema de 1ª ordem

Luís Farinha, Rui Sancho
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
Maio 2009

Neste apêndice é apresentado o esquema eléctrico do sistema de 1ª ordem usado no Capítulo 5.

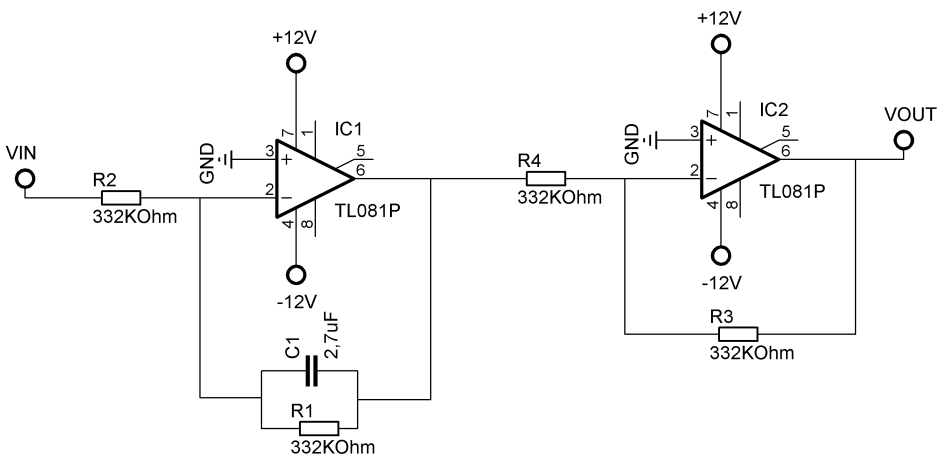


Figura E.1: Esquema eléctrico do sistema de 1ª ordem.

Apêndice F

Módulo de controlo de temperatura PCT 13

Luís Farinha, Nuno Marujo, Rui Sancho
Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro
Maio 2009

(a) Visão frontal (b) Visão traseira

Figura F.1: Módulo PCT 13

O módulo de controlo de temperatura PCT 13 é um equipamento destinado ao ensino e investigação, desenvolvido pela Armfield, que faz parte de uma alargada gama de produtos que a mesma possui e que visam demonstrar processos de medição e controlo.

Este módulo tem por objectivo o controlo de um processo térmico, que consiste no aquecimento de um caudal de água de entrada até a uma temperatura desejada. Para este efeito, o módulo possui um pequeno depósito de água, situado à retaguarda, que contém no seu interior uma resistência. Quando se aplica uma corrente à resistência, esta dissipa energia aquecendo, assim, a água contida no depósito. No topo do depósito está situada uma bomba que faz circular a água contida no mesmo, através de um circuito fechado. O caudal que circula neste pode ser regulado por uma válvula manual ou por uma electroválvula.

Na retaguarda do módulo está situado um tubo por onde pode ser injectado um caudal de água fria. Este caudal pode, igualmente, ser regulado por uma válvula manual. Quando injectada, a água flui por um circuito que passa paralelamente ao circuito fechado de água quente através de um permutador, saindo finalmente por um tubo na lateral do módulo.

O permutador é um elemento chave do sistema, propiciando a transferência de energia entre o caudal de água quente e o caudal de água fria. É deste modo que o caudal de água fria é aquecido à temperatura de saída desejada.

O módulo inclui um controlo ON/OFF que permite controlar a temperatura da água no depósito, sendo necessário, para o efeito, regular um botão que se encontra na caixa à retaguarda.

Painel de interface ao módulo PCT 13

O módulo de controlo de temperatura PCT 13 possui um painel de interface, cuja imagem está apresentada na Figura F.2.

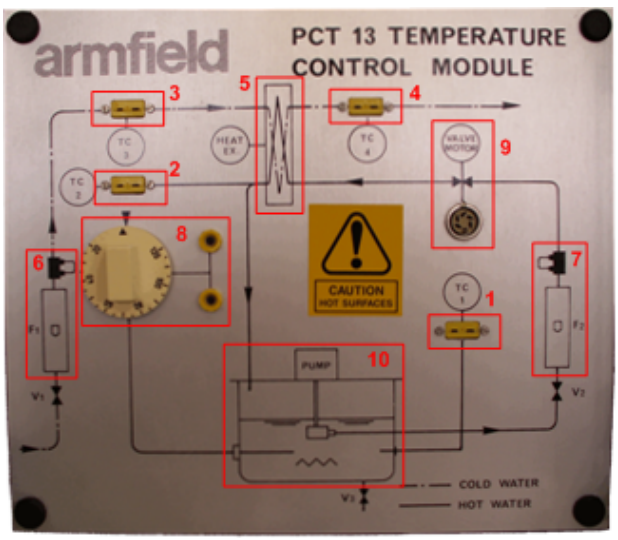


Figura F.2: Painel de interface

- **Termopares** - Para a aquisição da temperatura nos diferentes pontos do PCT 13, são usados 4 termopares do tipo K (1, 2, 3 e 4). O termopar 1 permite efectuar a leitura da temperatura presente no depósito de água quente; o termopar 2 permite efectuar a leitura da temperatura do circuito fechado de água quente, após a passagem pelo permutador; os termopares 3 e 4 permitem efectuar a leitura da temperatura da água de entrada e de saída, respectivamente.

- **Permutador** - O módulo possui um permutador (5) que, como já foi referido anteriormente, faz a transferência de energia da água quente para a água fria, resultando assim na saída a temperatura da água desejada.

- **Depósito de água quente -** A temperatura da água no circuito fechado é regulada com base na temperatura presente no depósito (10). Este possui internamente uma resistência pela qual, a água é aquecida. A temperatura da água pode ser controlada por um termostáto (8), ou então, por controlo de potência baseado na técnica de contagem de impulsos de rede.

- **Termostáto** - O termostáto (8), como foi referido, possibilita controlar de forma ON/OFF a temperatura no depósito de água quente.

- **Caudal de água** - Existem dois caudais presentes no PCT 13 que influenciam directamente a temperatura de saída: um para a entrada de água fria (6) e outro para a água quente (7). O primeiro caudal indica o volume de água de entrada no PCT 13 e é controlado de forma manual; o segundo indica o volume de água que circula no circuito fechado, onde pode ser controlado de duas formas: de forma manual ou através de uma electroválvula (9).

- **Electroválvula** - A electroválvula (9) permite determinar o volume de água que circula no circuito fechado, possibilitando, com base na sua posição, controlar a temperatura da água de saída.

Controlo do módulo PCT 13

Por forma a controlar o módulo PCT 13 foi necessária a adição de alguns módulos ao sistema, seguindo a estrutura presente na Figura F.3.

A temperatura de saída é influenciada por diversas variáveis, tais como a temperatura no depósito, o caudal de entrada e a posição da electroválvula. O sistema permite a medição da temperatura em diversos pontos através dos termopares que disponibiliza. Deste modo, várias filosofias de controlo podem ser adoptadas.

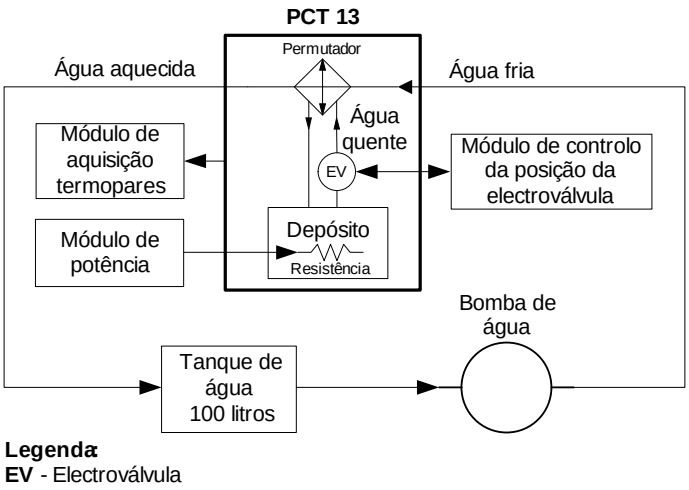


Figura F.3: Sistema de controlo módulo PCT 13

Os módulos adicionados ao sistema visam o controlo da temperatura da água no depósito, da posição da electroválvula e da aquisição de sinal dos termopares para monitorização do sistema e foram utilizados para efeitos de caracterização da dinâmica do mesmo.

Módulo de potência

O módulo de potência permite o controlo da potência fornecida à resistência de aquecimento, presente no depósito, numa determinada janela temporal. O módulo utilizado é apresentado na Figura F.4.

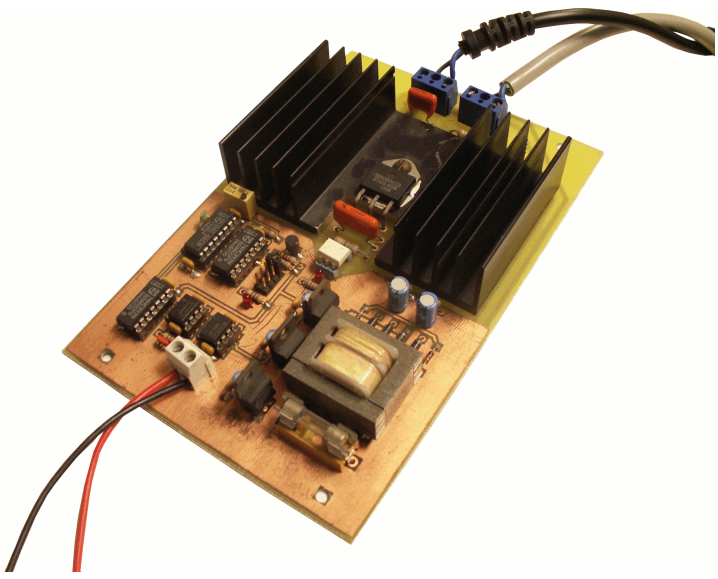


Figura F.4: Módulo de potência

Esta placa aplica um controlo por ciclos de rede, ou seja, numa determinada janela temporal define em quantos ciclos de rede é aplicada potência à resistência. Desta forma, consegue definir uma potência média durante esse intervalo.

Por forma a calcular a potência máxima que pode ser fornecida à resistência (caso que acontece quando conduz todos os ciclos) foi necessário medir o valor da resistência. Esta apresenta um valor de $57,2\Omega$, o que implica uma potência dissipada de aproximadamente 1000W.

Módulo de controlo da electroválvula

O módulo de controlo da electroválvula permite controlar o posicionamento da mesma, e consequentemente a temperatura da água de saída. Este módulo é apresentado na Figura F.5.

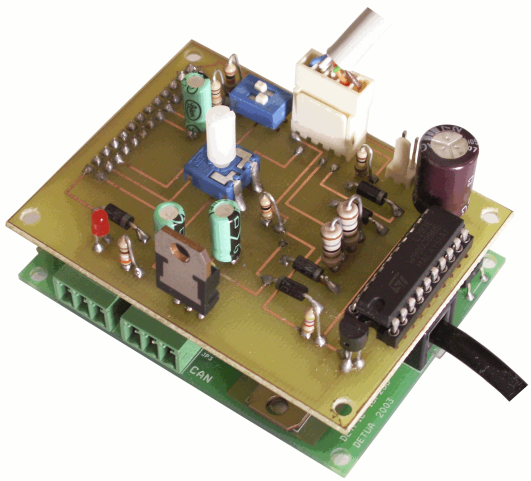
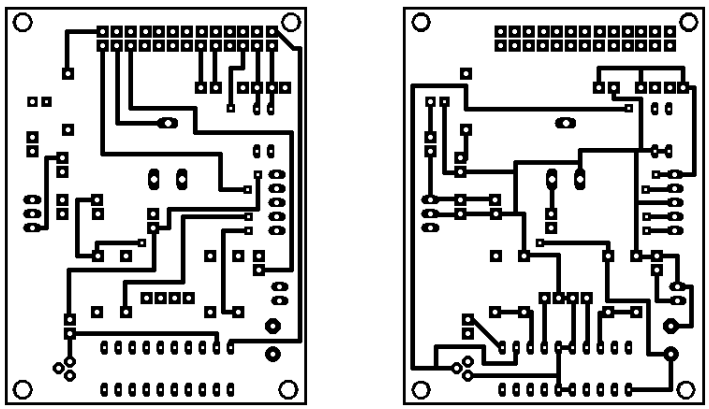


Figura F.5: Controlador da electroválvula

Este módulo possui quatro interfaces distintas. Os acessos podem ser feitos via RS232, barramento CAN, potenciómetro manual ou potenciómetro externo. O controlo da electroválvula é feito, essencialmente, através de um microcontrolador e de uma ponte-H. Este módulo é composto por duas placas, uma placa DETPIC desenvolvida pelo Departamento Electrónica, Telecomunicações e Informática da Universidade de Aveiro e uma placa de expansão, desenvolvida no âmbito desta tese, cujo projecto resultou no esquema de circuito impresso apresentado na Figura F.6.



(a) Frente (b) Traseira

Figura F.6: Circuito impresso placa electroválvula

O esquemático correspondente a esta placa é apresentado na Figura F.7.

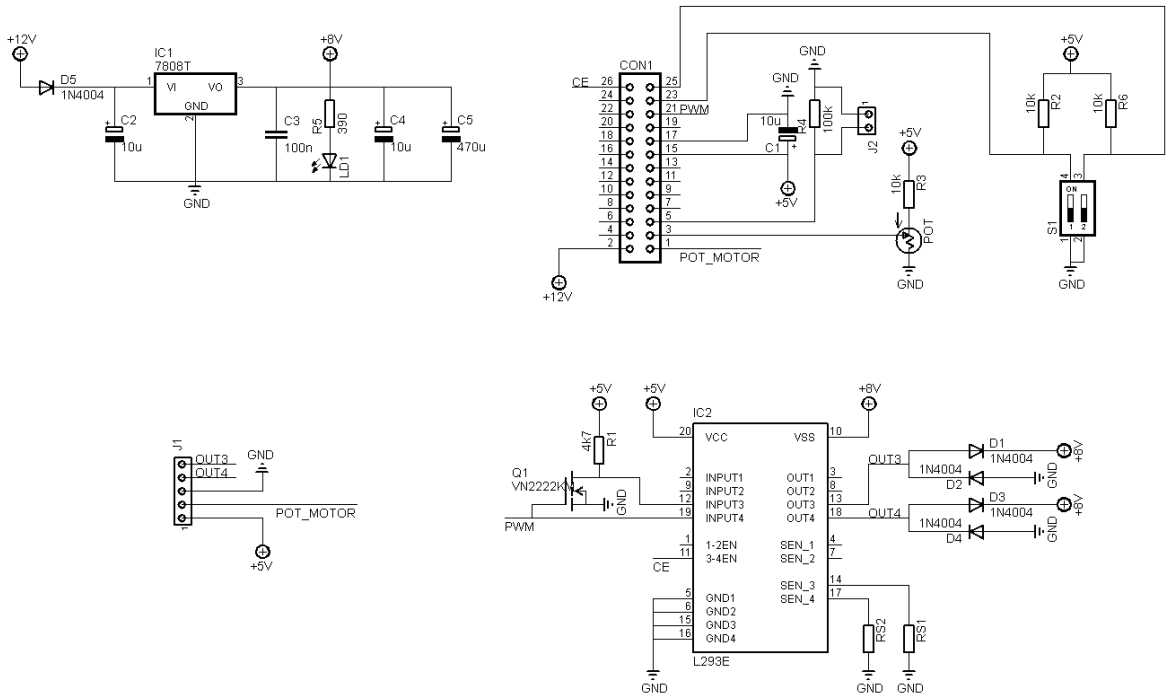


Figura F.7: Esquema da placa Electroválvula

Módulo de aquisição termopares

Para o módulo de aquisição dos termopares é necessário um equipamento de instrumentação que acondicione o sinal. Para efeitos de caracterização do sistema optou-se pela utilização do equipamento de instrumentação HP34970A da Hewlett-Packard, que possibilita a interface com termopares do tipo K, fazendo a amplificação do sinal e a

compensação de junção fria. Esta opção possui ainda a vantagem deste aparelho suportar a norma SCPI, o que permite que se possa monitorizar o estado dos termopares no PC via RS232. Este equipamento é apresentado na Figura F.8.

Figura F.8: Equipamento de instrumentação HP34970A

Bomba de água

Para a injecção de água no sistema foi necessário a utilização de uma bomba, que bombeasse a água armazenada num depósito para o módulo PCT 13. A bomba utilizada é apresentada na Figura F.9.

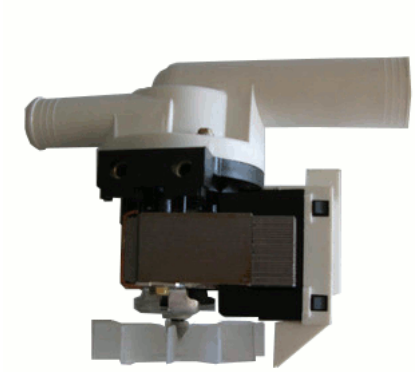


Figura F.9: Bomba de água

Depósito de água

Por forma a que fosse possível a execução dos testes necessários sem um gasto despropositado de água, utilizou-se um depósito de 100 litros, do qual a água era bombeada e para o qual era reposta. O depósito é apresentado na Figura F.10.

Figura F.10: Depósito de água

Caracterização do sistema

O sistema apresentado pode ser separado em dois subsistemas e, por conseguinte, podem ser efectuadas duas caracterizações distintas. Numa primeira abordagem será caracterizado o subsistema depósito onde será apresentada a sua resposta ao degrau. Numa fase posterior será apresentada a resposta ao degrau da electroválvula.

Subsistema depósito

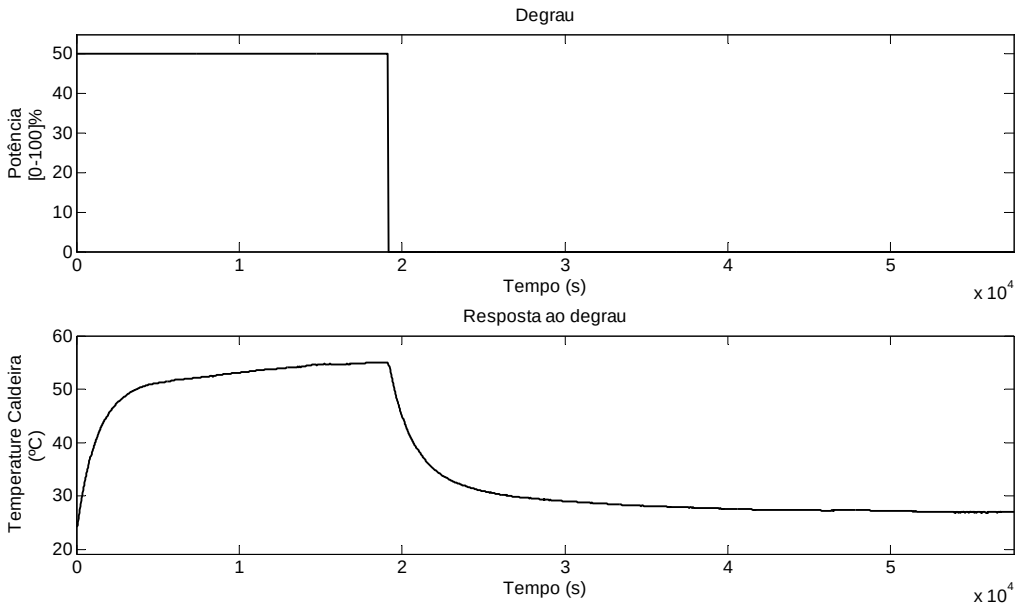


Figura F.11: Resposta ao degrau

Para a caracterização do subsistema depósito utilizou-se o módulo de potência anteriormente apresentado. Partindo da temperatura ambiente, aplicou-se um degrau com 50% de potência à resistência interna do depósito e monitorizou-se a temperatura do

depósito. Quando esta estabilizou, deixou-se de aplicar potência à resistência, deixando assim que a água arrefecesse. Esta experiência permitiu obter a constante de tempo de aquecimento e arrefecimento do subsistema depósito. Os resultados obtidos são apresentados na Figura F.11.

Através dos resultados obtidos concluiu-se que as constantes de tempo de aquecimento e arrefecimento são aproximadamente de 1650s e 3400s, respectivamente.

Subsistema electroválvula

O subsistema electroválvula consiste no controlo do caudal de água quente através do posicionamento da electroválvula, por forma a que a água aquecida atinja uma determinada temperatura.

Para caracterizar o subsistema electroválvula regulou-se a temperatura da água quente para um valor de 75ºC. O controlo desta temperatura fica a cargo do termostáto presente na caixa à retaguarda do módulo PCT 13. Após esta temperatura se encontrar relativamente estável, obtém-se a resposta ao degrau aplicado na electroválvula para caudais de entrada de 50, 100, 200 e 280 cm^3/min. Através desta experiência é possível determinar quais as constantes de tempo associadas ao subsistema electroválvula, assim como o valor do tempo morto, caso este o apresente. Os gráficos seguintes apresentam a resposta ao degrau para os vários caudais.

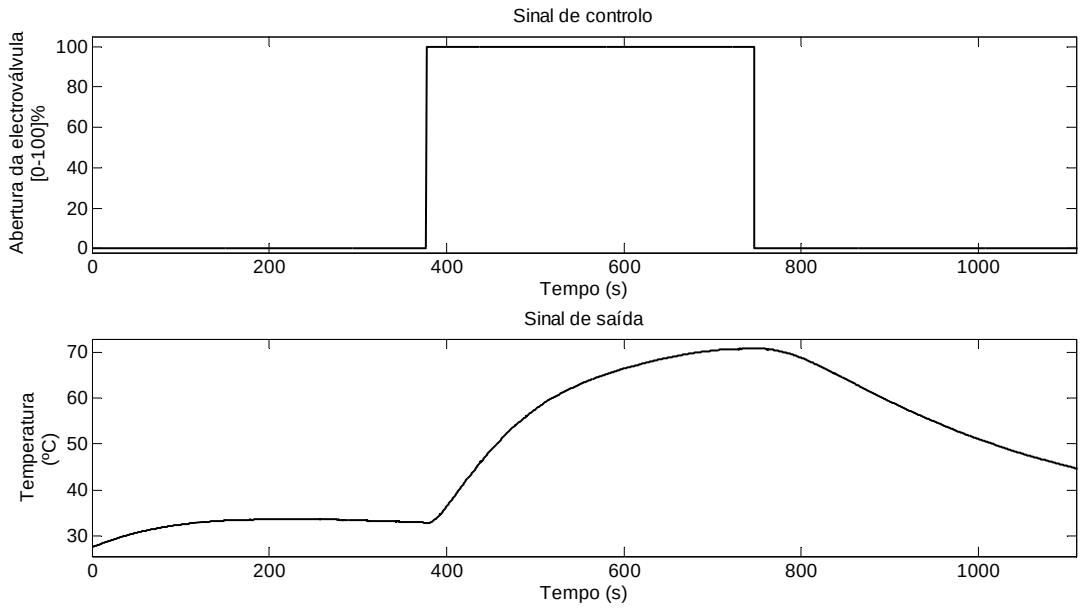


Figura F.12: Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 50 cm^3/min

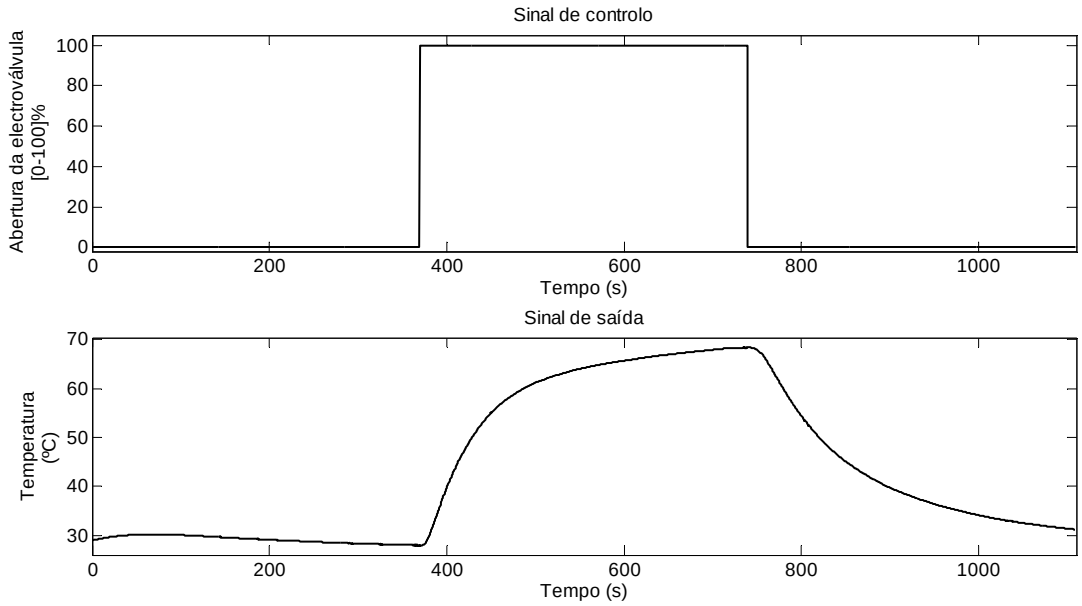


Figura F.13: Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 100 cm^3/min

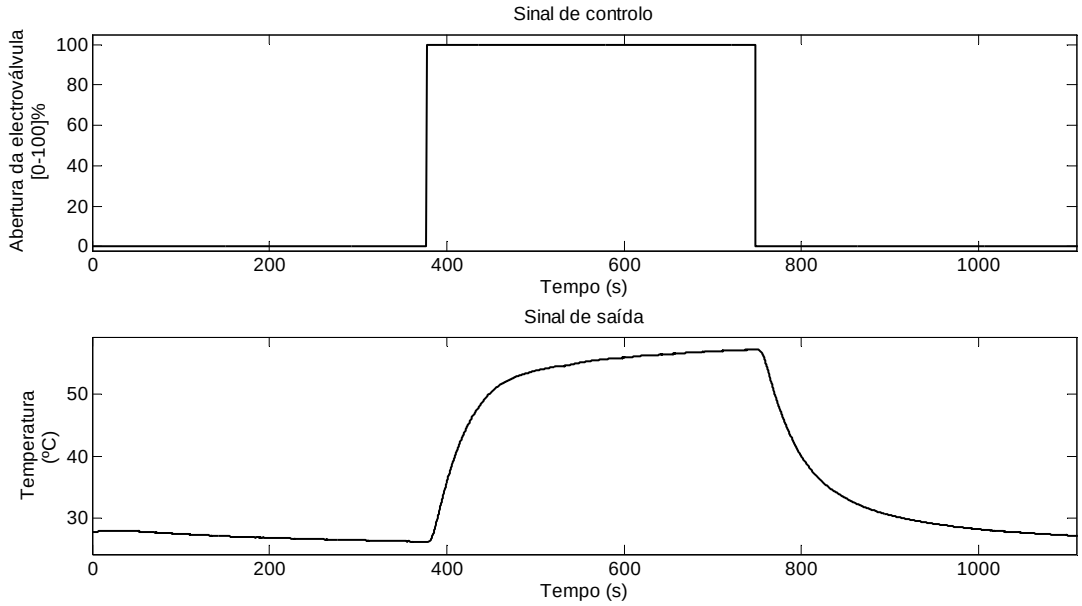


Figura F.14: Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 200 cm^3/min

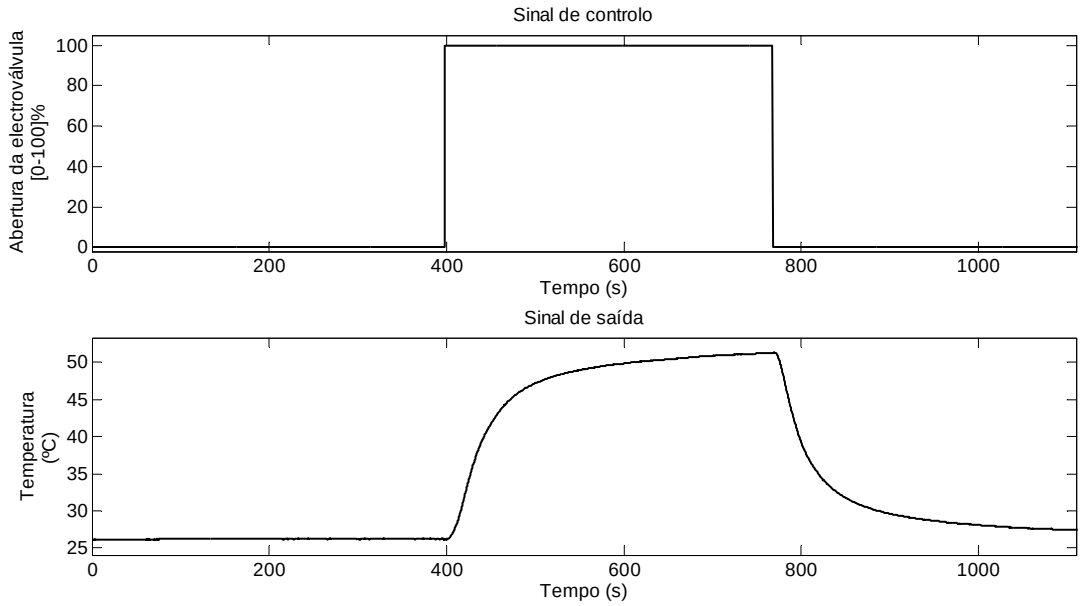


Figura F.15: Resposta ao degrau do subsistema electroválvula para um caudal de entrada de 280 cm^3/min

Análise dos resultados

Após uma análise dos resultados apresentados anteriormente, conclui-se que o tempo morto apresentado pelo subsistema electroválvula, ou seja o tempo que este demora a reagir a uma excitação externa, é de aproximadamente 4s. No que diz respeito aos tempos de subida, que correspondem ao tempo que o sinal de saída demora a atingir 63% do seu valor máximo quando excitado por um degrau, estes são apresentados na tabela F.1 juntamente com os tempos de descida.

Caudal (cm^3/min)	**Tempo de subida (s)**	**Tempo de descida (s)**
50	120.8	202.9
100	68.8	111.4
200	48.5	57.3
280	50	43.1

Tabela F.1: Constantes de tempo subsistema electroválvula

Identificação do subsistema electroválvula

Para a identificação do subsistema electroválvula gerou-se um sinal PRBS *(Pseudo-Random Binary Sequence)* que foi utilizado como sinal de controlo de abertura da electroválvula. O período de amostragem utilizado foi de 10s. Monitorizou-se a temperatura da água de saída do sistema e no fim do teste usaram-se estes valores para a

obtenção de um modelo matemático do sistema. Para este efeito utilizou-se o método dos mínimos quadrados para um sistema de 1ª ordem.

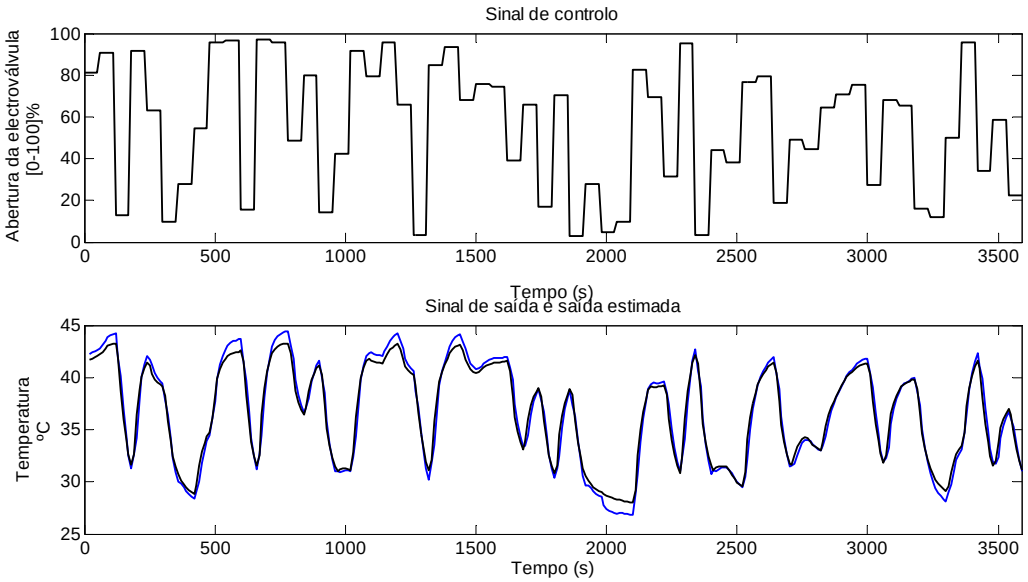


Figura F.16: Identificação subsistema electroválvula

Na Figura F.16 pode ser observado o sinal de controlo utilizado, bem como a saída do sistema e a saída estimada obtida usando o modelo matemático estimado. Verifica-se que estes dois sinais se assemelham muito, o que indica que o modelo do sistema é bastante preciso. O modelo matemático obtido é apresentado na expressão F.1.

$$G(z) = \frac{\hat{\theta}_2 z^{-1} + \hat{\theta}_3 z^{-2}}{1 + \hat{\theta}_1 z^{-1}} = \frac{0.0057 z^{-1} + 0.0017 z^{-2}}{1 - 0.9430 z^{-1}} \tag{F.1}$$

Apêndice G

Lista de acrónimos

ADC	*Analogue Digital Converter*
CRC	*Cyclic Redundancy Check*
CSMA/CD	*Carrier Sense Multiple Access with Collision Avoidance*
CSMA/CA	*Carrier Sense Multiple Access with Collision Detection*
DAC	*Digital Analogue Converter*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
DSC	*Digital Signal Controller*
DSP	*Digital Signal Processors*
FTP	*File Transfer Protocol*
HTTP	*Hypertext Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IMAP	*Internet Message Access Protocol*
IP	*Internet Protocol*
PID	*Proportional Integral Differetial*
POP3	*Post Office Protocol version 3*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*

TCP	*Transmission Control Protocol*
TTL	*Transistor-Transistor Logic*
UDP	*User Datagram Protocol*
ZOH	*Zero-Order Hold*

Bibliografia

[1] Charles Spurgeon. *Ethernet the definitive guide*. O'Reilly, Cambridge, MA, 2000.

[2] Ana Antunes. *Algoritmos de Controlo Distribuído em Sistemas Baseados em Microprocessadores*. PhD thesis, Universidade de Aveiro, 2008.

[3] Nuno Marujo. Sistema distribuído de aquisição e controlo baseado em linguagem java. Master's thesis, Universidade de Aveiro, Aveiro, 2009.

[4] Jun Wang, Yanyun Cheng, Chao Niu, and Hongmei Xiao. A compensation algorithm of ethernet time delay based on model predictive control. In *Proc. IEEE International Conference on Industrial Technology ICIT 2008*, pages 1–3, 21–24 April 2008.

[5] Cai Hua, Fan Weihua, Mu Jingqing, and Hu Weili. Study of the networked control system with long delay. In *Proc. IEEE International Conference on Control and Automation ICCA 2007*, pages 1194–1197, May 30 2007–June 1 2007.

[6] Jean J. Labrosse. *MicroC/OS-II: The Real-Time Kernel*. CMP Books, 2002.

[7] Anthony J. Massa. *Embedded Software Development with eCos*. Prentice Hall PTR, 2002.

[8] Doug Abbott. *Linux for embedded and real-time applications*. Newnes, Burlington, MA, 2006.

[9] Gianluca Zito Ioan D. Landau. *Digital Control Systems: Design, Identification and Implementation*. Springer, 2006.

[10] M. Sami Fadali. *Digital Control Engineering: Analysis and Design*. Academic Press, 2009.

[11] Fei-Yue Wang. *Networked Control Systems: Theory and Applications*. Springer, 2008.

[12] Feng-Li Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks: Ethernet, controlnet, and devicenet. *IEEE Control Systems Magazine*, 21(1):66–83, Feb. 2001.

[13] Yu Jianyong, Yu Shimin, and Wang Haiqing. Survey on the performance analysis of networked control systems. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 5068–5073, 10–13 Oct. 2004.

[14] Wei Zhang, M. S. Branicky, and S. M. Phillips. Stability of networked control systems. 21(1):84–99, Feb. 2001.

[15] B. Azimi-Sadjadi. Stability of networked control systems in the presence of packet losses. In *Proc. 42nd IEEE Conference on Decision and Control*, volume 1, pages 676–681, 9–12 Dec. 2003.

[16] Inc Cisco Systems. *Internetworking Technologies Handbook*. Cisco Press, New York, 2000.

[17] Paulo Pedreiras, Luís Almeida, and José Alberto Fonseca. The quest for real-time behavior in ethernet. In *The Industrial Information Technology Handbook*, pages 1–14. 2005.

[18] P. Pedreiras, L. Almeida, and P. Gai. The ftt-ethernet protocol: merging flexibility, timeliness and efficiency. In *Proc. 14th Euromicro Conference on Real-Time Systems*, pages 134–142, 19–21 June 2002.

[19] R. Marau, L. Almeida, and P. Pedreiras. Enhancing real-time communication over cots ethernet switches. In *Proc. IEEE International Workshop on Factory Communication Systems*, pages 295–302, June 27, 2006.

[20] Ethernet powerlink - communication profile specification. Technical report, EPSG - Ethernet POWERLINK Standardisation Group, 2008.

[21] Gilbert Held. *Ethernet Networks Design, Implementation, Operation, Management*. Wiley, New York, 4 edition, 2003.

[22] Steve Heath. *Embedded systems design*. Newnes, Oxford, 2003.

[23] Hermann Kopetz. *Real-time systems design principles for distributed embedded applications*. Kluwer Academic, Boston, 1997.

[24] Albert M. K. Cheng. *Real-time systems scheduling, analysis, and verification*. Wiley-Interscience, Hoboken, NJ, 2002.

[25] Microchip Technology Inc. *dsPIC30F6011A/6012A/6013A/6014A - Data Sheet*, 2008.

[26] http://www.freertos.org/ (12/07/2009).

[27] Microchip Technology Inc. *ENC28J60 - Data Sheet*, 2008.

[28] Nilesh Rajbharti. *The Microchip TCP/IP Stack - AN833*. Microchip Technology Inc., 2008.

[29] P.P.L. Regtien. *Electronic instrumentation*. VSSD, 2 edition, 2005.

[30] Monolithic thermocouple amplifiers with cold junction compensation-ad594/ad595. Technical report, Analog Devices, 1999.

[31] Alexandre M. Mota. Identificação de sistemas, 2000.