



Catarina Inês Marques de Lucena

Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador : Paulo José Carrilho de Sousa Gil,
Professor Auxiliar, Universidade Nova de Lisboa

Co-orientador : Luís Filipe Figueira Brito Palma,
Professor Auxiliar, Universidade Nova de Lisboa

Júri:

Presidente: Luís Filipe Santos Gomes

Arguente: José António Barata de Oliveira

Vogais: Paulo José Carrilho de Sousa Gil
Luís Filipe Figueira Brito Palma



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2012

Aplicação de técnicas de controlo óptimo difuso em ambientes distribuídos

Copyright © Catarina Inês Marques de Lucena, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. A Faculdade de Ciências e Tecnologia e a Universidade

Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Aos meus pais, irmã e Marcelo

Agradecimentos

Em primeiro lugar gostaria de agradecer ao meu orientador, o Professor Doutor Paulo Gil pela disponibilidade que demonstrou ao longo da implementação deste trabalho.

Gostaria de agradecer também ao Professor Doutor Luís Brito Palma, co-orientador, que se mostrou sempre disponível para o esclarecimento de dúvidas quando estas surgiam.

Ao Departamento de Engenharia Electrotécnica da Universidade Nova de Lisboa, indispensável para a minha formação académica.

Gostaria de agradecer ao Edgar Silva pelo apoio prestado, não só a nível pessoal como a nível técnico, que contribuiu bastante para a finalização da Dissertação.

Aos meus amigos e familiares, em especial aos meus pais, que sempre apostaram em mim e que tornaram este momento possível.

Resumo

A presente dissertação centra-se na implementação de duas arquitecturas para sintonização de controladores PID (Proporcional-Integrativo-Derivativo) difusos onde o conceito de desempenho em malha fechada é explicitamente tido em conta. Com as metodologias propostas pretende-se automatizar a sintonização dos factores de escala de controladores difusos, pois até à data não existe ainda uma abordagem sistemática para este efeito.

Em ambas as arquitecturas a sintonização é feita através da solução de um problema de optimização com restrições, tanto nas acções de controlo como nos ganhos do controlador. A primeira arquitectura de sintonização consiste na solução do problema de optimização com base num modelo não linear do sistema e assumindo conhecido o sinal de referência. Na segunda arquitectura a optimização é feita em linha com base num modelo linear que será constantemente actualizado. Com a aplicação da segunda arquitectura pretende-se um melhor desempenho no que se refere ao controlo de sistemas não lineares variantes no tempo.

Dada a resiliência proporcionada pela optimização em linha dos factores de escala é possível, por esta via, acomodar o efeito da variação dos tempos de latência em sistemas distribuídos. Neste contexto, o presente trabalho propõe ainda a aplicação deste paradigma ao controlo remoto sobre redes de sensores sem fios.

Abstract

This Thesis focus on the implementation of two methodologies for tuning the gains of fuzzy PID (Proportional-Integral-Derivative) controllers where the concept of closed loop control system performance is explicitly taken into account. The main goal is to develop an automatic framework for tuning fuzzy controllers, since, currently, there is no systematic methodology to fill this gap.

In both approaches the fuzzy controller gains are found by solving a nonlinear optimization problem subject to a set of constraints on the controller gains, as well as, on control actions and outputs. The first problem concerns the case where the optimization is performed offline, considering the system dynamics described by a nonlinear model. The second one, focus the online optimization based on a linear model, which is recursively updated. In what the second methodology is concerned, it it intended to come with a conceptual solution to cope with nonlinear time varying systems.

Because of the resilience to changing dynamics provided by the online optimization of scale factors it is possible to accommodate the effect of varying latencies in distributed control systems. In this context, this work presents the application of this paradigm to networked control systems over wireless sensor and actuator networks.

Acrónimos

ARX *Auto Regressive with eXogenous input*

ARMAX *Auto Regressive Moving Average with eXogeneous input*

BFGS Brayden, Fletcher, Goldfarb & Shanno

BJ Box Jenkins

DFP Davidon, Fletcher & Powell

KKT Kharash Kuhn-Tucker

MIMO *Multi Input-Multi Output*

NNARX *Neural Network Auto Regressive with eXogenous input*

NG Negativo Grande

NM Negativo Médio

NP Negativo Pequeno

OE Output Error

PD Proporcional-Derivativo

PI Proporcional-Integral

PID Proporcional-Integral-Derivativo

PG Positivo Grande

PM Positivo Médio

PP Positivo Pequeno

RF Rádio Frequência

RSSF Rede de Sensores Sem Fios

SISO *Single Input-Single Output*

TCP/IP *Transmission Control Protocol/ Internet Protocol*

ZO Zero

Conteúdo

Acrónimos	xiii
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	2
1.3 Contribuições	2
1.4 Organização da Dissertação	3
2 Lógica difusa	7
2.1 Conjuntos difusos	8
2.1.1 Propriedades dos conjuntos difusos	10
2.1.2 Operações sobre conjuntos difusos	12
2.2 Raciocínio aproximado	14
2.2.1 Variáveis linguísticas	14
2.2.2 Proposições difusas	15
2.2.3 Proposições difusas <i>Se-Então</i>	16
2.3 Mecanismo de inferência	16
2.3.1 Método de inferência de Mamdani	18
3 Controladores difusos	21
3.1 Evolução histórica	22
3.2 Estrutura de um controlador difuso	22
3.2.1 Módulo de fuzificação	22
3.2.2 Base de Conhecimento	24
3.2.3 Mecanismo de inferência	27
3.2.4 Módulo de desfuzificação	27
3.3 Controladores PID difusos	28
3.3.1 Controlador PD difuso	29
3.3.2 Controlador PI difuso	30

3.3.3	Controlador PID difuso	32
4	Identificação de sistemas	35
4.1	Processo de Identificação	36
4.2	Métodos paramétricos	37
4.2.1	Modelo ARX	38
4.2.2	Modelo ARMAX	39
4.2.3	Modelo erro de saída	39
4.2.4	Modelo de Box Jenkins	40
4.2.5	Modelo no espaço de estado	40
4.3	Métodos dos mínimos quadráticos	41
4.4	Validação de modelos	42
4.4.1	Validação cruzada	42
4.4.2	Teste dos resíduos	42
4.4.3	Actualização recursiva	43
4.5	Identificação de sistemas não lineares	44
4.5.1	Modelos não lineares	44
4.5.2	Estruturas do tipo NNARX	45
5	Optimização	47
5.1	Optimização analítica e optimização numérica	48
5.2	Componentes do problema de optimização	48
5.3	Classificação dos problemas de optimização	49
5.4	Óptimo local e óptimo global	50
5.5	Métodos de optimização	51
5.5.1	Método de Newton para optimização sem restrições	52
5.5.2	Métodos Quasi-Newton para optimização sem restrições	52
5.6	Optimização sujeita a restrições	54
5.6.1	Extensão do método de quasi-nNwton para optimização com re- strições	55
6	Redes de sensores sem fios	57
6.1	Sensores sem fios	57
6.2	Redes de comunicação	59
6.2.1	Topologia de rede em árvore	59
6.2.2	Topologia de rede em estrela	59
6.2.3	Topologia de rede <i>mesh</i>	60
6.3	Áreas de aplicação das redes de sensores sem fios	61
6.4	Factores de influência no projecto de RSSFs	61
6.4.1	Tolerância a falhas	63
6.4.2	Escalabilidade	63
6.4.3	Custos de produção	63

6.5	Tipos de Comunicação	63
6.5.1	Rádio Frequência	64
6.6	Sistemas Operativos	65
7	Arquitecturas Propostas	67
7.1	PID difuso com optimização em diferido dos ganhos	68
7.2	PID difuso com optimização em linha dos ganhos	69
7.2.1	Computação paralela para optimização dos ganhos	69
7.2.2	Generalização do controlador a um sistema MIMO	73
8	Casos de Estudo	75
8.1	Descrição do sistema	75
8.2	Topologia de redes sem fios adoptada	76
8.3	Projecto do controlador PID difuso	77
8.4	PID difuso com sintonização de ganhos baseada em controladores convencionais	79
8.4.1	PID difuso com sintonização de ganhos baseada em controladores convencionais sobre placas de aquisição	79
8.4.2	PID difuso com sintonização de ganhos baseada em controladores convencionais sobre redes de sensores sem fios	79
8.5	Optimização do ganhos	81
8.6	PID difuso com optimização em diferido	82
8.6.1	PID difuso com optimização em diferido dos ganhos sobre placas de aquisição	83
8.6.2	PID difuso com optimização em diferido dos ganhos sobre redes de sensores sem fios	85
8.7	PID difuso com optimização em linha dos ganhos	85
8.7.1	PID difuso óptimo sobre placa de aquisição	87
8.7.2	PID difuso com optimização em linha dos ganhos sobre redes de sensores sem fios	90
8.8	Resultados comparativos	90
9	Conclusões e perspectivas futuras	95
9.1	Conclusões gerais	95
9.2	Perspectivas futuras	96
	Bibliografia	97

Lista de Figuras

1.1	Estrutura da Dissertação.	4
2.1	Conjunto crespo.	8
2.2	Conjunto difuso.	9
2.3	Propriedades de um conjunto difuso.	11
2.4	Operações sobre conjuntos clássicos utilizando diagramas de Veen.	12
2.5	Operações sobre conjuntos difusos contínuos.	14
2.6	Funções de pertença associadas a uma variável linguística.	16
3.1	Estrutura do controlador lógico difuso.	24
3.2	Grau de sobreposição e taxa de sobreposição	25
3.3	Método de inferência de Mamdani baseado em regras individuais.	28
3.4	Estrutura do controlador PD Difuso.	30
3.5	Estrutura do controlador PI Difuso.	31
3.6	Estrutura do controlador PID Difuso.	32
3.7	Estrutura do controlador PID Difuso híbrida.	33
4.1	Fluxograma ilustrativo do procedimento de refinamento de estrutura.	37
4.2	Modelo geral para identificação paramétrica.	38
4.3	Modelo ARX.	39
4.4	Modelo ARMAX.	39
4.5	Modelo OE.	39
4.6	Modelo BJ.	40
4.7	Esquema de blocos de uma estrutura NARX.	46
5.1	Exemplo de uma função difícil de otimizar.	50
6.1	Mote TelosB.	58
6.2	Topologia de rede em árvore.	60
6.3	Topologia de rede em estrela.	60

6.4	Topologia de rede <i>mesh</i>	61
6.5	Aplicações das redes de sensores sem fios.	62
7.1	Algoritmo de optimização dos ganhos do controlador.	68
7.2	Arquitectura geral do controlador com optimização em linha dos ganhos.	70
7.3	Rotina principal.	71
7.4	Rotina de optimização dos ganhos.	72
7.5	Horizonte de optimização.	73
7.6	Esquema simplificado da rotina principal para um sistema MIMO.	74
8.1	Sistema de três tanques.	76
8.2	Esquemático do processo industrial AMIRA DTS200.	76
8.3	<i>National Instrument USB-6009</i>	77
8.4	Arquitectura utilizada na rede de sensores sem fios (<i>single-hop</i>).	77
8.5	Funções de pertença de \tilde{e} e $\Delta\tilde{e}$	78
8.6	Funções de pertença de $\Delta\tilde{u}_{PI}$ e \tilde{u}_{PD}	78
8.7	Tanque 1 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre placas de aquisição.	80
8.8	Tanque 2 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre placas de aquisição.	80
8.9	Tanque 1 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre redes de sensores sem fios.	81
8.10	Tanque 2 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre redes de sensores sem fios.	82
8.11	Tanque 2 do sistema MIMO controlado por um controlador PID com optimização em diferido dos ganhos sobre placas de aquisição.	84
8.12	Tanque 1 do sistema MIMO controlado por um controlador PID com optimização em diferido dos ganhos sobre placas de aquisição.	84
8.13	Tanque 1 do sistema MIMO controlado por um controlador PID com optimização diferido dos ganhos sobre RSSFs.	85
8.14	Tanque 2 do sistema MIMO controlado por um controlador PID com optimização diferido dos ganhos sobre RSSFs.	86
8.15	Evolução do valor do critério de desempenho em ordem ao número de amostras do horizonte de optimização.	86
8.16	Tanque 1 do sistema MIMO controlado por um controlador PID com optimização em linha dos ganhos sobre placas de aquisição.	88
8.17	Tanque 2 do sistema MIMO controlado por um controlador PID com optimização em linha dos ganhos sobre placas de aquisição.	88

8.18	Evolução dos ganhos para o tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.	89
8.19	Evolução dos ganhos para o tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.	89
8.20	Tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre RSSFs.	90
8.21	Tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre RSSFs.	91
8.22	Evolução dos ganhos para o tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre uma rede de sensores sem fios.	91
8.23	Evolução dos ganhos para o tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre uma rede de sensores sem fios.	92

Lista de Tabelas

2.1	Funções de pertença mais comuns.	10
2.2	Operações sobre conjuntos difusos discretos.	13
2.3	Implicações difusas.	17
3.1	Breve história da tecnologia difusa.	23
3.2	Casos particulares da sobreposição de funções de pertença.	26
3.3	Base de regras genérica para o controlador PD Difuso.	30
3.4	Base de regras genérica para o controlador PI Difuso.	31
7.1	Características do processo de optimização em diferido.	69
7.2	Características do processo de optimização em em-linha.	73
7.3	Características dos paradigmas centralizado e descentralizado.	74
8.1	Configuração do processo e ambiente de controlo.	87
8.2	Medidas de desempenho dos controladores sobre placas de aquisição. . .	92
8.3	Medidas de desempenho dos controladores sobre RSSFs.	93

1

Introdução

Neste capítulo, como o próprio nome sugere, efectua-se formalmente a introdução ao trabalho desenvolvido, apresentando-se a motivação que presidiu à escolha do tema, os objectivos a serem alcançados e as contribuições mais relevantes.

1.1 Motivação

As técnicas de controlo Proporcional-Integral-Derivativo (PID) continuam a ser largamente utilizadas em processos industriais. A razão para a sua utilização deve-se essencialmente à reconhecida simplicidade e à existência de metodologias de sintonização dos correspondentes ganhos. No entanto, no caso de sistemas não lineares torna-se difícil satisfazer as condições de projecto em anel fechado utilizando controladores PID clássicos não-adaptativos.

As técnicas de controlo difusas são inerentemente abordagens não lineares, dado que incorporam três fontes principais de não linearidade, nomeadamente, a base de regras, o mecanismo de inferência e os módulos de fuzificação e desfuzificação.

Este paradigma de controlo baseado em lógica difusa tem provado ser uma abordagem efectiva no controlo de vários sistemas não lineares e tem vindo a ser sugerido como alternativa às técnicas de controlo convencionais [Fen06]. Estes controladores são conhecidos por apresentarem maior robustez, comparativamente aos controladores convencionais e o seu desempenho ser menos sensível a variações paramétricas do sistema ou a grandezas não modeladas [WTG96]. Para além disso, aplicações recentes dos controladores difusos têm mostrado um grande potencial no contexto de sistemas mal definidos

que podem ser convenientemente controlados por operadores humanos, sem o conhecimento explícito das dinâmicas do sistema [Lee90].

Apesar das grandes potencialidades relativamente à aplicação de controladores difusos em vários contextos, encontrar um conjunto de variáveis linguísticas, regras e factores de escala, e subsequentemente sintonizá-los, apresenta ainda um desafio que urge dar resposta, devido em grande medida à inexistência de uma abordagem sistemática. Quando se trata de sistema MIMO (Multi Input-Multi Output) o seu controlo é ainda mais exigente em virtude do efeito de acoplamento presente.

Na literatura existem múltiplas propostas para sintonização de controladores difusos. Estas incluem abordagens baseadas em heurísticas [MMC96], metodologias baseadas na pseudo-equivalência entre controladores PID difusos e controladores PID convencionais [Piv02] e as que se baseiam técnicas de computação evolutiva [HL09].

1.2 Objectivos

O principal objectivo deste trabalho consiste em desenvolver duas arquitecturas tendo em vista a sintonização dos factores de escala de controladores difusos. Uma das arquitecturas centra-se na obtenção dos ganhos do controlador em diferido, ou seja, obtidos previamente ao ciclo de controlo e outra em que os ganhos do controlador são obtidos em linha com o processo, apresentando o sistema de controlo um carácter adaptativo.

Em ambas as arquitecturas os ganhos são calculados através da solução de um problema de optimização não linear sujeita a restrições, nomeadamente, no que se refere aos factores de escala, acção de controlo e saídas, sendo a função de custo uma medida do desempenho do sistema em anel fechado.

1.3 Contribuições

A contribuição deste trabalho reside no desenvolvimento de duas arquitecturas que permitam o ajuste automático dos factores de escala para um controlador PID difuso em linha e em diferido. As características para efeito de sintonização do controlador PID, cujos factores de escala são obtidos em diferido são as seguintes:

- Obter um conjunto de factores de escala óptimos para o processo industrial, regras difusas previamente escolhidas, assim como variáveis linguísticas;
- Ser aplicável a sistemas MIMO;
- Ter em consideração o desempenho em malha fechada do sistema;

- Para além do desempenho, no que trata ao erro, ter em consideração os custos energéticos e o tempo de vida útil do sistema.

Quanto às características referentes ao processo de sintonização em linha do controlador PID, para além das referidas anteriormente, deverão ainda contemplar:

- Permitir manter um tempo de amostragem previamente definido independentemente do tempo de optimização dos ganhos;
- Uma resposta mais homogénea do sistema (sobreelevação e tempos de estabelecimento constantes), cujo resultado é de difícil obtenção para factores de escala fixos perante sistemas não lineares;
- Aumentar a robustez do controlador quando sujeito a alterações na dinâmica do sistema, tais como *outliers* e tempos de latência, característicos das redes de sensores sem fios.
- Permitir a distribuição computacional da optimização dos ganhos por diversas máquinas.

1.4 Organização da Dissertação

Esta Dissertação é organizada em nove capítulos, tal como se mostra no diagrama da Figura 1.1.

No capítulo 2 são introduzidos os conceitos básicos da lógica difusa e da teoria dos conjuntos difusos indispensáveis à presente dissertação. Nas duas primeiras secções é feita uma breve introdução à lógica difusa e aos conjuntos difusos, realçando as características que os distinguem dos conjuntos clássicos. Na secção 2.2 é apresentado o conceito de raciocínio aproximado, introduzindo os conceitos de variável linguística e proposições difusas. Por fim, na secção 2.3, é apresentado o conceito de mecanismo de inferência, particularizando para o mecanismo de inferência de Mamdani.

No capítulo 3, secção 3.1 é feita uma breve revisão histórica relativamente ao controlo difuso, sendo referidos os principais aspectos motivadores relativos à utilização de controladores difusos em alternativa aos controladores convencionais. Na secção 3.2 é apresentada a estrutura geral de um controlador difuso e descritos cada um dos seus componentes. Por fim, na secção 3.3, são descritos alguns controladores difusos, nomeadamente os controladores PD, PI e PID com inferência de Mamdani.

Na primeira parte do capítulo 4 são descritas as etapas envolvidas no processo de identificação de sistemas (secção 4.1), enfatizando os métodos paramétricos, nomeadamente os modelos ARX (secção 4.2). Na subsecção 4.3 descreve-se o processo de parametrização

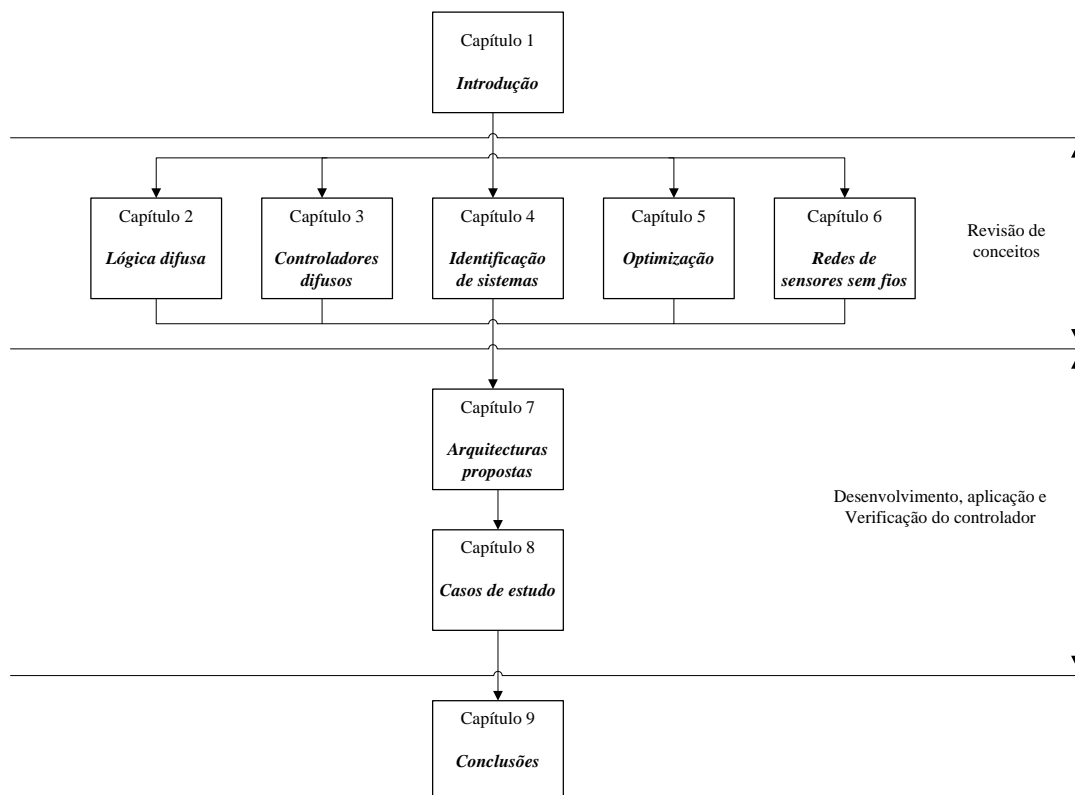


Figura 1.1: Estrutura da Dissertação.

com base no método dos mínimos quadráticos, apresentando-se na secção 4.4 os métodos de validação mais usuais. Por fim, no contexto dos sistemas não lineares, apresentam-se duas abordagens para acomodar este comportamento, nomeadamente, a actualização em linha dos parâmetros do modelo ARX e a utilização de estruturas de modelos não lineares para identificação de sistemas, em particular recorrendo a redes neuronais proactivas multicamada.

Na primeira secção do capítulo 5 é apresentada a distinção entre os métodos de optimização analíticos e os métodos de optimização numéricos. Em seguida, na secção 5.2 são definidos alguns conceitos associados à obtenção de uma solução óptima no contexto da presente dissertação. Na secção 5.3 são descritas algumas classificações que podem ser atribuídas ao processos de optimização, enquanto que na secção 5.5 é apresentada uma classificação para os métodos de optimização, dando ênfase aos métodos baseados no gradiente. Por fim, é apresentada a metodologia que permite aos métodos quasi-Newton lidarem com restrições de igualdade e de desigualdade.

No capítulo 6 apresentam-se as Redes de Sensores sem Fios (RSSFs) como meio de construção de uma rede de comunicações digitais. São referidas e caracterizadas as principais topologias de rede mais utilizadas (secção 6.2), as principais áreas de aplicação (ver

6.3) e os factores a ter em conta aquando do seu projecto 6.4. No âmbito desta dissertação, são caracterizados os nós sensores utilizados, o tipo de comunicação e o sistema operativo seleccionado.

Na secção 7.1 do capítulo 7 é proposta uma arquitectura em que os ganhos do controlador são calculados previamente ao ciclo de controlo, sendo mantidos ao longo deste ciclo. Na secção 7.2 é apresentada uma outra arquitectura para a qual se prevê adaptação dos sistema de controlo a alterações da dinâmica do sistema. Nesta abordagem os ganhos são actualizados permanentemente, acomodando por esta via variações da dinâmica do processo, ou no caso de sistemas de controlo sobre RSSFs o efeito de tempos de latência variáveis.

No capítulo 8 são apresentados os resultados para as duas arquitecturas, assim como as condições de projecto do controlador PID. É feito um estudo comparativo do desempenho dos controladores projectados em relação a um controlador difuso sintonizado recorrendo a baseada na pseudo-equivalência referente a um controlador PID convencional. É analisado também o desempenho dos controladores projectados sobre redes de sensores sem fios. Por fim apresentam-se os resultados na perspectiva de duas métricas de comparação, uma relacionada com o desempenho relativamente ao erro e outra relacionada com a variação da acção de controlo.

No capítulo 9 apresentam-se as principais conclusões relativas à presente dissertação e propõem-se algumas ideias com vista a dar continuidade ao trabalho desenvolvido.

2

Lógica difusa

A teoria dos conjuntos difusos desenvolvida inicialmente por Lofti A. Zadeh ¹ [Zad65], permite a representação e tratamento de informação sujeita a incertezas, imprecisões, ou mesmo algo vaga. Este tipo particular de conhecimento/informação é passível de ser encontrado nos mais variados contextos e ambientes, pelo que o paradigma em questão fornece uma base matemática inestimável de suporte ao seu tratamento, recorrendo para o efeito a mecanismos semânticos e cognitivos próprios dos seres humanos.

O principal objectivo da lógica difusa consiste na generalização da ideia representada pela teoria dos conjuntos ordinários, ou convencionais, aproximando-se da imprecisão e do aspecto vago do raciocínio humano [Kos92]. A lógica difusa pode ser entendida como um acervo de métodos e operações difusas que possibilitam a modelação de sistemas ou a implementação de controladores, sendo o conhecimento subjacente formulado fundamentalmente através de um conjunto de regras. O aspecto de maior realce desta metodologia é a possibilidade de capturar através de um modelo matemático conceitos intuitivos, tais como graus de satisfação, nível de conforto, entre outras etiquetas qualitativas. Neste sentido, a lógica difusa é uma lógica multivalor capaz de absorver informação vaga, normalmente descritas numa linguagem natural e convertê-la num formato numérico de fácil manipulação computacional, procurando, desta forma, modelar ou emular o modo impreciso do raciocínio humano. Além disso, torna possível apoiar os vários níveis de tomada de decisão, ou ainda dar forma ao condicionamento do comportamento de sistemas dinâmicos não-lineares.

¹Professor de Ciências da Computação na Universidade de Berkeley na Califórnia

2.1 Conjuntos difusos

A teoria dos conjuntos difusos tem como base a teoria clássica dos conjuntos, alicerçada na Álgebra Booleana desenvolvida por George Boole², que associou à Lógica dois estados de verdade e que passou a ser conhecida como Lógica de Boole.

Um conjunto é uma colecção de objectos que possuem uma determinada característica comum. Na lógica de Boole, um objecto possui apenas duas formas de se relacionar com um conjunto, designadamente, pertencendo ou não a esse conjunto. No domínio da teoria dos conjuntos difusos, os conjuntos booleanos, ou normais, são denominados conjuntos "crespos". Para um conjunto "crespo" C no universo X é possível definir a seguinte função característica $\mu_C : X \rightarrow \{0; 1\}$, isto é:

$$\mu_C(x) = \begin{cases} 1 & \text{se } x \in C \\ 0 & \text{se } x \notin C \end{cases} \quad (2.1)$$

Na Teoria dos conjuntos de Zadeh [Zad65], um conjunto difuso é uma colecção de elementos com características comuns. No entanto, estes elementos possuem uma miríade de maneiras de se relacionarem com um dado conjunto, além de pertencente ou não pertencente, pois estão associados através de um grau de pertença (μ) que varia entre 0 e 1. Tal permite uma suavização na transição entre as condições extremas de pertença e de não pertença, de tal forma que esta não ocorra abruptamente, mas sim de forma progressiva.

Exemplo 2.1

Tomando como exemplo o termo jovem, este refere-se a um conjunto cujos limites não estão bem definidos. Para o exemplo, considere-se apenas que são jovens os indivíduos até aos vinte e cinco anos. De acordo com a lógica Booleana, Figura 2.1, alguém com vinte e cinco anos é jovem, mas no instante em que complete vinte e seis anos deixará de o ser.

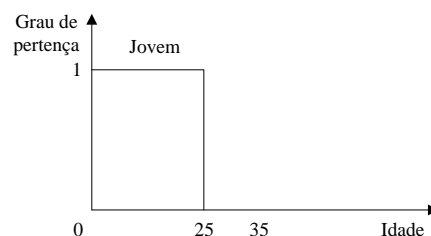


Figura 2.1: Conjunto crespo.

²(2 de Novembro de 1815 - 8 de Dezembro de 1864), foi um matemático e filósofo britânico, responsável pela criação da Álgebra Booleana, fundamental para o desenvolvimento da computação moderna

Na Figura 2.2 apresenta-se uma função característica que pode descrever de forma mais razoável o conceito subordinado à variável linguística Jovem, cujo grau de verdade (valor de pertença) varia de forma contínua entre 0 e 1.

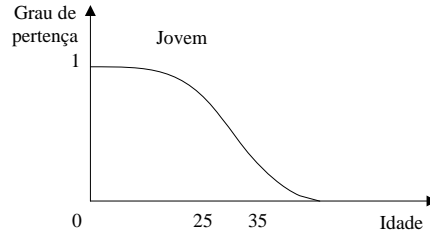


Figura 2.2: Conjunto difuso.

As funções de pertença são curvas que definem o mapeamento do espaço das entradas ($x \in X$) no espaço representado por valores de pertença (ou grau de pertença) compreendidos entre $[0; 1]$ (contradomínio), sendo o conjunto definido com base nesta função de pertença designado por conjunto difuso.

A função de pertença μ_D de um conjunto difuso D é uma função definida por:

$$\mu_D : X \rightarrow [0; 1] \quad (2.2)$$

logo, cada elemento x de X tem um grau de pertença $\mu_D \in [0; 1]$, sendo D completamente determinado pelo conjunto de pares ordenados,

$$\mu_D = \{(x, \mu_D(x)) : x \in X\} \quad (2.3)$$

Se o universo de discurso X for discreto, D é representado por:

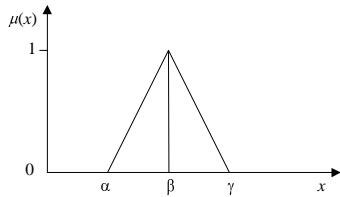
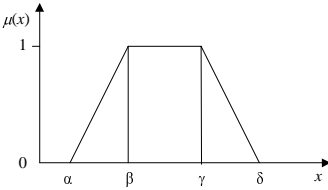
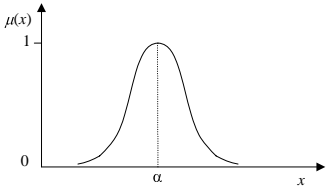
$$\sum_{x \in X} \mu_D(x)/x \quad (2.4)$$

No caso de X ser um universo de discurso contínuo, D é representado por:

$$\int_X \mu_D(x)/x \quad (2.5)$$

Uma função de pertença pode ser definida através da enumeração dos elementos do universo X , conjuntamente com os respectivos graus de pertença. Outra forma consiste na utilização de uma função geral que, para cada elemento x do universo X , forneça o seu grau de pertença $\mu_D(x)$. As funções de pertença mais comuns são do tipo triangular, trapezoidal e gaussiana. Estas funções de pertença encontram-se representadas na Tabela 2.1, assim como as respectivas equações características.

Tabela 2.1: Funções de pertença mais comuns.

Função de pertença	Caracterização
<p style="text-align: center;">Triangular</p> 	$\Lambda : X \rightarrow [0; 1]$ $\Lambda(x, \alpha, \beta, \gamma) = \begin{cases} 0 & x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \alpha \leq x \leq \beta \\ \frac{\gamma-x}{\gamma-\beta} & \beta \leq x \leq \gamma \\ 0 & x > \gamma \end{cases}$
<p style="text-align: center;">Trapezoidal</p> 	$\Pi : X \rightarrow [0; 1]$ $\Pi(x, \alpha, \beta, \gamma, \delta) = \begin{cases} 0 & x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \alpha \leq x \leq \beta \\ 1 & \beta \leq x \leq \gamma \\ \frac{\delta-x}{\delta-\gamma} & \gamma \leq x \leq \delta \\ 0 & x > \delta \end{cases}$
<p style="text-align: center;">Radial</p> 	$\Omega : X \rightarrow [0; 1]$ $\Omega(x, \alpha, \beta) = \exp\left(\frac{-x(x-\alpha)^2}{2\beta^2}\right)$

2.1.1 Propriedades dos conjuntos difusos

Os conjuntos difusos apresentam três propriedades: *suporte*, *altura* e *núcleo*. Na Figura 2.3 encontra-se representado um conjunto difuso com as suas propriedades específicas.

Uma outra propriedade importante dos conjuntos difusos, em particular no contexto dos sistemas de controlo, refere-se à propriedade de convexidade. Um dado conjunto difuso A é convexo se a sua função de pertença é crescente, decrescente ou em forma de sino, ou seja:

$$\forall x_1, x_2 \in X \forall \lambda \in [0; 1] : \mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2)) \quad (2.6)$$

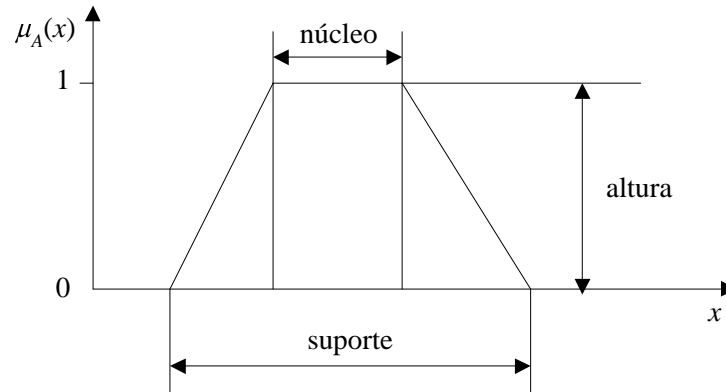


Figura 2.3: Propriedades de um conjunto difuso.

Suporte

Seja A um conjunto difuso expresso no universo X , o suporte do conjunto difuso A é definido por:

$$\text{suporte}(A) = \{x \in X : \mu_A(x) > 0\} \quad (2.7)$$

Tratando-se de um conjunto difuso satisfazendo a propriedade de convexidade, o seu suporte é um intervalo, também designado por *largura*. A *largura* de um conjunto convexo é definida por:

$$\text{largura}(A) = \sup(\text{suporte}(A)) - \inf(\text{suporte}(A)) \quad (2.8)$$

Se o conjunto $\text{suporte}(A)$ é limitado, como é usual no controlo difuso, o supremo (sup) e o ínfimo (inf) podem ser substituídos por máximo (max) e mínimo (min).

Altura

A *altura* de um conjunto difuso A é definida por:

$$\text{altura}(A) = \sup_{x \in X} \mu_A(x) \quad (2.9)$$

Núcleo

O *núcleo* de um conjunto difuso A é dado por:

$$\text{núcleo}(A) = \{x \in X : \mu_A(x) = 1\} \quad (2.10)$$

2.1.2 Operações sobre conjuntos difusos

É possível construir conjuntos a partir de dois ou mais conjuntos disponíveis, aplicando sobre eles, operações básicas, tais como o complemento, intercepção e união. É comum, na lógica clássica, utilizar-se o diagrama de Venn para representar o conjunto universo X e os conjuntos formados a partir dos seus elementos. Na Figura 2.4 a parte mais escura ilustra as operações efectuadas sobre os conjunto A e B no universo X .

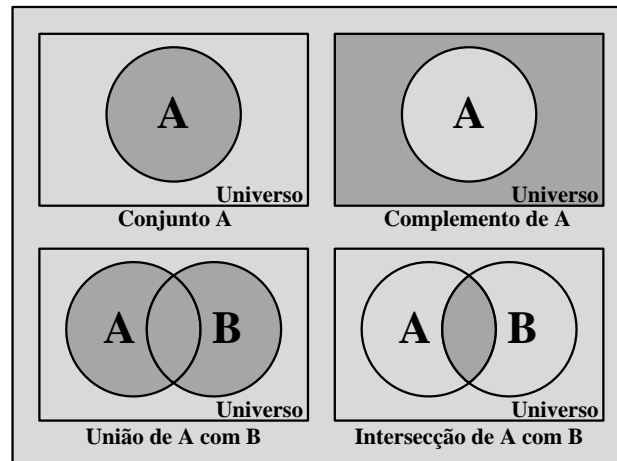


Figura 2.4: Operações sobre conjuntos clássicos utilizando diagramas de Veen.

Algumas das noções da teoria clássica de conjuntos podem ser estendidas aos conjuntos difusos, tais como a noção de igualdade e de subconjunto. Sejam A e B subconjuntos difusos definidos no universo X :

Dois subconjuntos são iguais ($A = B$) sse

$$\forall_{x \in X} : \mu_A(x) = \mu_B(x) \quad (2.11)$$

A é um *subconjunto* de B ($A \subseteq B$) sse

$$\forall_{x \in X} : \mu_A(x) \leq \mu_B(x) \quad (2.12)$$

Enquanto que na teoria clássica de conjuntos operações, tais como a união, intersecção e complemento são definidas sem qualquer ambiguidade, tal não se verifica no âmbito dos conjuntos difusos.

Operadores de Zadeh

Zadeh propôs os seguintes operadores para as operações de união, intersecção e complemento de conjuntos difusos [Zad73]:

Intersecção de conjuntos

$$\forall x \in X : \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.13)$$

União de conjuntos

$$\forall x \in X : \mu_{A \cup B}(x) = \max(\mu_A, \mu_B(x)) \quad (2.14)$$

Complemento de um conjunto

$$\forall x \in X : \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.15)$$

Exemplo 2.2

Sejam A e B dois conjuntos discretos com as seguintes funções de pertença definidas no universo discreto X :

$$X = \{1; 2; 3; 4\} \quad \mu_A = \{0, 0; 0, 3; 0, 7; 1, 0\} \quad \mu_B = \{0, 2; 0, 7; 1, 0; 0, 8\}$$

As operações sobre os conjuntos difusos têm como resultado os valores presentes na Tabela 2.2.

Tabela 2.2: Operações sobre conjuntos difusos discretos.

Operação difusa	Resultado
$\mu_{A \cap B}$	$\{0, 0; 0, 3; 0, 7; 0, 8\}$
$\mu_{A \cup B}$	$\{0, 2; 0, 7; 1, 0; 1, 0\}$
$\mu_{\bar{A}}$	$\{1, 0; 0, 7; 0, 3; 0, 0\}$

Exemplo 2.3

Sejam A e B dois conjuntos contínuos, as operações de *interceção*, *união* e *complemento* encontram-se ilustradas na Figura 2.5.

Operadores de Yegar

Como outro exemplo de operações sobre conjuntos difusos, apresenta-se seguidamente as operações de Yegar [KF87]:

Intersecção de conjuntos

$$\forall x \in X : \mu_{A \cap B}(x) = \min[(\mu_A(x)^w, \mu_B(x)^w)^{\frac{1}{w}}] \quad (2.16)$$

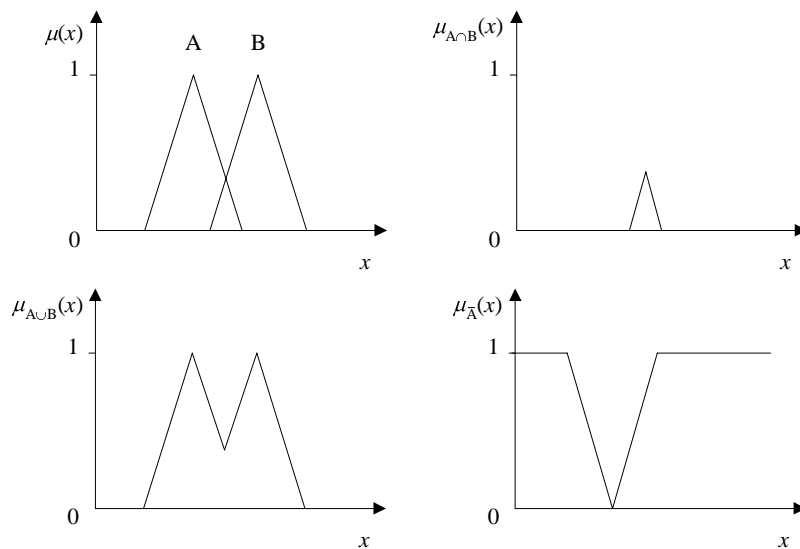


Figura 2.5: Operações sobre conjuntos difusos contínuos.

União de conjuntos

$$\forall_{x \in X} : \mu_{A \cup B}(x) = \max(\mu_A, \mu_B(x)) \quad (2.17)$$

Em ambas as expressões assume-se que as funções de pertença são funções de $x \in X$, e $w \in (0, \infty)$. Pode-se demonstrar que estas definições tendem para as apresentadas por Zadeh quando $w \rightarrow \infty$. É possível verificar que uma das intenções destas definições, é permitir que ambas as funções de pertença tenham influência no resultado final, ao contrário do que se passava com as definições apresentadas por Zadeh. No entanto, e apesar de eventuais benefícios que daí possam advir, tem de se ter em conta que as operações definidas por Yager são computacionalmente mais pesadas, o que as pode tornar inapropriadas para utilização em sistemas em que o tempo de resposta seja um factor determinante.

2.2 Raciocínio aproximado

No raciocínio aproximado a consequência de um dado conjunto de proposições difusas depende essencialmente do significado ligado a estas proposições. Deste modo, a inferência é determinada recorrendo a conjuntos difusos que representem o significado destas proposições [DHR93].

2.2.1 Variáveis linguísticas

Sendo a teoria dos conjuntos difusos capaz de tratar e modelar contextos complexos e imprecisos, esta permite também a utilização de variáveis não numéricas e imprecisas. As

variáveis linguísticas são variáveis cujos valores são palavras em linguagem natural representadas em conjuntos difusos e consistem na unidade fundamental de conhecimento no âmbito do raciocínio aproximado. Zadeh [Zad73] considera:

" *By a linguistic variable we mean a variable whose values are words or sentences in a natural or artificial language. For example, Age is a linguistic variable if its values are linguistic rather than numerical, i.e., young, not young, very young, quite young, old, not very old and not very young, etc., rather than 20, 21, 22, 23, ...* "

Formalmente, uma variável linguística é definida pelo quadripleto (X, LX, X, M) em que:

- X é o nome simbólico da variável linguística;
- LX é o conjunto dos valores (termos) linguísticos que a variável X pode tomar;
- X é o universo de discurso;
- M é a regra semântica que associa a cada valor X o seu significado $M(X)$, num subconjunto de X .

Exemplo 2.4

Considere-se no caso de um controlador difuso a variável linguística erro. A variável é definida pelo "quadripleto (E, LE, E, M_E) , em que:

- E , variável linguística erro;
- $LE, \{NG, NM, NP, ZO, PP, PM, PG\}$ o conjunto de termos linguísticos ou conjuntos de referência da variável linguística erro;
- $E, [-1, 0; 1, 0]$ o universo de discurso;
- M_E , significado de cada termo linguístico da variável linguística erro, através da função de pertinência associada a cada um dos termos³ (ver Figura 2.6)

2.2.2 Proposições difusas

O raciocínio aproximado é utilizado para representar e estabelecer relações com base em conhecimento expresso através de premissas simples, representadas na forma de linguagem natural, por exemplo:

³NG = Negativo Grande, NM = Negativo Médio, NP = Negativo Pequeno, ZO = Zero, PP = Positivo Pequeno, PM = Positivo Médio; PG = Positivo Grande

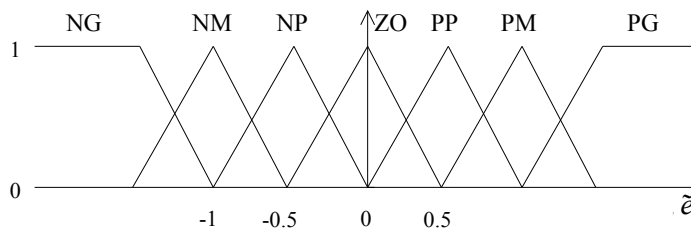


Figura 2.6: Funções de pertinência associadas a uma variável linguística.

" Erro tem o valor *Negativo Grande* "

A tradução simbólica desta expressão em termos de variáveis linguísticas é a seguinte:

1. O símbolo E é escolhido para indicar a variável linguística erro;
2. O símbolo NG é escolhido para indicar o valor particular do erro, negativo grande;
3. A expressão em linguagem natural é reescrita do seguinte modo:

"Erro tem a propriedade de ser Negativo Grande"

4. A representação simbólica da expressão anterior toma a seguinte forma:

E é NG

Com base na noção de proposições difusas e nas conectividades *e*, *ou*, *não* e *Se-Então* é possível construir proposições difusas mais complexas.

2.2.3 Proposições difusas *Se-Então*

Nesta subsecção apresenta-se um conjunto de relações difusas que podem ser usadas para representar o significado da regra:

$$\text{Se } X \text{ é } A \text{ então } Y \text{ é } B \quad (2.18)$$

A maioria das relações difusas são geralmente derivadas a partir da lógica multivalor. Na Tabela 2.3 são apresentadas algumas relações que traduzem as implicações difusas mais utilizadas na representação do significado da regra caracterizada pela equação (2.18)[R.95]

2.3 Mecanismo de inferência

Na secção anterior foi abordada a representação do significado de uma regra *Se – Então*. Nesta secção considera-se a forma de representação de múltiplas regras utilizando o método de inferência de Mamdani, pois é o mais utilizado no contexto do controlo difuso. Este método de inferência recorre à implicação de Mamdani para representar o

Tabela 2.3: Implicações difusas.

Implicação	$I(a,b)$
Lukasiewicz	$\min(1 - a + b, 1)$
Zadeh	$\min(1 - a, \min(a, b))$
Kleene	$\max(1 - a, b)$
Reichenbach	$1 - a + ab$
Gödel	$\begin{cases} a & \geq b \\ 1 & \text{c.c.} \end{cases}$
Dubois e Prade	$\begin{cases} 1 - a & b = 0 \\ b & a = 1 \\ 1 & \text{c.c.} \end{cases}$
Goguen	$\begin{cases} 1 & a = 0 \\ \min(\frac{b}{a}, 1) & \text{c.c.} \end{cases}$
Gaines	$\begin{cases} 1 & a \leq b \\ 0 & \text{c.c.} \end{cases}$
Wu1	$\begin{cases} 1 & a < b \\ \min(1 - a, b) & \text{c.c.} \end{cases}$
Wu2	$\begin{cases} 0 & a < b \\ b & \text{c.c.} \end{cases}$
Yager	b^a
Willmott	$\min\left(\max(1 - a, b), \max\left(a, 1 - b, \min(b, 1 - a)\right)\right)$
Mamdani	$\min(a, b)$
Larsen	ab

significado de uma regra *Se – Então*.

O motor de inferência é responsável pela combinação dos dados de entrada, já no formato difuso, com as regras existentes, as quais, trabalhando sobre regras de produção, descrevem o processo de tal forma que se obtenha, por meio de inferências particulares, o valor de saída correspondente[KY95].

Existem diferentes classificações no que se refere aos motores de inferência de um sistema difuso. Driankov propõe a classificação em dois grandes grupos [DHR93]: as máquinas de inferência baseadas em composição e as baseadas em regras individuais. No primeiro caso combinam-se todas as regras da base e aplica-se uma única inferência, enquanto que no segundo caso é efectuada a inferência regra a regra e são aplicadas normas-*T* no final deste processo para obtenção do valor único de saída.

Klir [KY95] classifica os motores de inferência de acordo com os cálculos efectuados sobre a base de regras. Existem os motores de inferência dirigidos aos dados, em que os antecedentes da regra são os valores de entrada do motor e a saída corresponde ao consequente da regra. Nos motores de inferência dirigidos aos antecedentes da regra os valores de entrada do motor de inferência são os consequentes da própria regra.

2.3.1 Método de inferência de Mamdani

Num conjunto de n regras, cada regra é representada simbolicamente da seguinte forma:

$$\text{Se } E \text{ é } LE^{(k)} \text{ então } U \text{ é } LU^{(k)}$$

em que, $LE^{(k)}$ e $LU^{(k)}$ são respectivamente os valores linguísticos de E e de U na regra k .

A inferência de Mamdani é definida da seguinte forma:

$$\forall_k : R_m^{(k)} = \int_{E \times U} \min(\mu_{LE^{(k)}}(e), \mu_{LU^{(k)}}(u)) / (e, u) \quad (2.19)$$

A função de pertença da entrada fuzificada, e^* é:

$$\forall_e : \mu^*(e) = \begin{cases} 1 & e = e^* \\ 0 & \text{outros} \end{cases} \quad (2.20)$$

O resultado de todo o conjunto de regras disparadas é dado por:

$$R_m = \bigcup_{k=1}^n R_m^{(k)} \quad (2.21)$$

o que significa que:

$$\forall_{e,u} : \mu_{R_m}(e, u) = \max_k \mu_{R_m^k}(e, u) = \max_k \min(\mu_{LE^{(k)}}(e), \mu_{LU^{(k)}}(u)) \quad (2.22)$$

O disparo de um conjunto de regras pode ser expresso por:

$$U - \mu^* o R_m \quad (2.23)$$

ou seja,

$$\forall_u : \mu_U(u) - \max \min(\mu_{LE^{(k)}}(e^*), \mu_{LU^{(k)}}(u)) \quad (2.24)$$

Além do método de inferência de Mamdani, também designado por método de inferência max – min, existem outros métodos igualmente utilizados no contexto do controlo difuso, nomeadamente, o método de inferência max – prod e o método de inferência

sum – prod. A designação destes métodos está intimamente relacionada com o tipo de operadores utilizados para representar a ligação *ou* (*or*) na agregação das regras e para a implicação *então* (*then*) de cada regra. O método de inferência max – prod difere do método de inferência max – min pelo facto de utilizar o produto para representar a implicação *então*.

3

Controladores difusos

A aplicação das técnicas de controlo convencionais pressupõe a existência de modelos matemáticos, na generalidade lineares, que reproduzam a dinâmica do sistema em questão. Em sistemas mais complexos, como é comum encontrar-se na indústria de processos, os modelos matemáticos apresentam na generalidade uma complexidade elevada e em muitas situações a sua obtenção é mesmo impossível. Frequentemente, mesmo sendo possível encontrar um modelo matemático, este revela-se altamente não-linear, variante no tempo e com incertezas associadas. Apesar das limitações apontadas, o controlo desses processos é, nalguns casos realizável através de operadores humanos experientes [Lee90]. Neste enquadramento, através da lógica difusa como meio de exprimir o comportamento de um operador humano materializa-se o conceito de controlo difuso.

Uma das vantagens da utilização de controladores difusos reside precisamente na possibilidade de expressar e codificar por meio de regras, o conhecimento/comportamento humano, com a vantagem de estes não sofrerem de problemas de cansaço, esquecimento ou distração, entre outros, o que potencia um melhor desempenho global do que o obtido através de operadores humanos. Outras vantagens dizem respeito à facilidade de incorporação de múltiplos objectivos, tais como, segurança, poupança de energia, estabilidade de operação, entre outros, bem como a possibilidade de utilização de variáveis não quantitativas, por exemplo, descrições qualitativas, as quais não podem ser utilizadas explicitamente através de técnicas de controlo clássico [MS85].

No que diz respeito ao esforço de desenvolvimento e custos de manutenção deste paradigma, estes são substancialmente menores, já que uma parte significativa do desenvolvimento associado à generalidade dos controladores passa pela obtenção de um modelo adequado

do processo a controlar. No caso de sistemas de elevada complexidade a obtenção de um bom modelo não é uma tarefa fácil, consumindo inevitavelmente elevados recursos. Pelo contrário, as técnicas de controlo difuso não requerem um modelo matemático do processo (desde que exista conhecimento suficiente acerca do processo que possa ser expresso de forma linguística), o que reduz substancialmente o tempo de desenvolvimento e respectivos custos.

3.1 Evolução histórica

O primeiro trabalho a propor a aplicação da teoria dos conjuntos difusos a problemas de controlo surgiu em 1972 por Chang e Zadeh [CZ72]. Contudo, foi um artigo de Zadeh publicado em 1973 que formulou a teoria básica na qual assentam as técnicas de controlo difuso [Zad73]. A primeira verdadeira aplicação no domínio do controlo difuso surgiu por Mamdani [Mam74] para efeito do controlo de um pequeno engenho a vapor. Desde então, este paradigma de controlo foi aplicado numa miríade de contextos, nomeadamente, no controlo automático, modelação qualitativa, reconhecimento de padrões, processamento de informação, inteligência artificial, sistemas de decisão, gestão, finanças, entre outros [PH11].

A seguinte tabela, proposta por Leonid Reznik (Tabela 3.1), sintetiza a evolução história da lógica difusa [Rez97].

3.2 Estrutura de um controlador difuso

Um controlador lógico difuso é constituído por quatro módulos principais, designadamente, o módulo de fuzificação ou fuzificador, a base de conhecimento, o mecanismo de inferência e o módulo de desfuzificação ou desfuzificador [Lee90]. A estrutura do controlador difuso encontra-se representada esquematicamente na Figura 3.1.

Nas subsecções seguintes será feita uma descrição mais pormenorizada de cada um dos módulos em questão.

3.2.1 Módulo de fuzificação

O módulo de fuzificação converte um valor real ou crespo num valor difuso, de modo a este ser compatível com a representação interna do controlador. Se o domínio de discurso não estiver normalizado, o módulo de fuzificação deverá também promover uma transformação de escala, num universo de discurso normalizado. Existem dois tipos principais de operadores de fuzificação, designadamente, o *singleton* e o *nonsingleton*. Quando o valor a fuzificar apresenta um valor numérico, o que acontece normalmente no domínio

Tabela 3.1: Breve história da tecnologia difusa.

1965	Teoria dos conjuntos difusos por Lofti Zadeh (EUA)
1972	Primeiro trabalho em grupo sobre sistemas difusos por Toshiro Terano (Japão)
1973	Publicação sobre algoritmos difusos por Zadeh (EUA)
1974	Controlo de um engenho de vapor por Ebrahim Mamdani
1977	Primeiro sistema especialista para avaliação de empréstimos por Hans Zimmermann (Alemanha) Sistema de controlo <i>Subway Sendai Transportation</i> (Japão)
1980	Controlo de fornos de cimento por L.Smith & Co. e Lauritz P. Holmbland (Dinamarca) (primeira aplicação permanente) Programa de Xadrez e Gamão difuso por Hans Berliner (EUA)
1984	Controlo de um sistema de tratamento de águas (injecção química) (Japão)
1985	Primeiro navio com controlo automático difuso desenvolvido por Masaki Togai e Hiroykyke Watanabe (EUA)
1986	Primeiro sistema especialista difuso para diagnosticar doenças (Japão)
1987	Controlador de manivelas Escavação de túneis Aterragem automática de uma aeronave Segunda conferencia IFSA (Tokio) Togai InfraLogic Inc (primeira companhia de lógica difusa) (USA)
1988	Controlo de fornos por Yokagawa Primeira venda de controladores difusos dedicados (Japão)
1990	<i>Laboratory for International Fuzzy Engineering Research</i> (LIFE) (Japão)
1991	Primeira aplicação em aparelhos de televisão pela Sony (Japão) Olho electrónico difuso por Fujitsu (Japão) <i>Fuzzy Logic System Institute</i> (FLSI) por Takashi Yamakawa (Japão) <i>Intelligent Systems Control Laboratory</i> (Siemens, Alemanha)
+1992	Inúmeros eventos, invenções e projectos

do controlo difuso, a representação difusa desse valor é efectuada através do fuzificador *singleton* [Lee90], ou seja:

$$\mu_A(x_0) = \begin{cases} 1 & \text{se } x = x_0 \\ 0 & \text{outros} \end{cases} \quad (3.1)$$

Uma das características intrínsecas a este operador é que não introduz imprecisão no processo de fuzificação, uma vez que o valor crespo x_0 é convertido no conjunto difuso A cuja função de pertença $\mu_A(x)$ é igual a zero excepto em x_0 , onde toma o valor 1. Refira-se ainda que esta estratégia de fuzificação é de fácil implementação.

Outra particularidade deste tipo de fuzificador é ser adequado em situações onde existe uma certeza absoluta nos dados. No entanto, quando tal não se verifica, por exemplo,

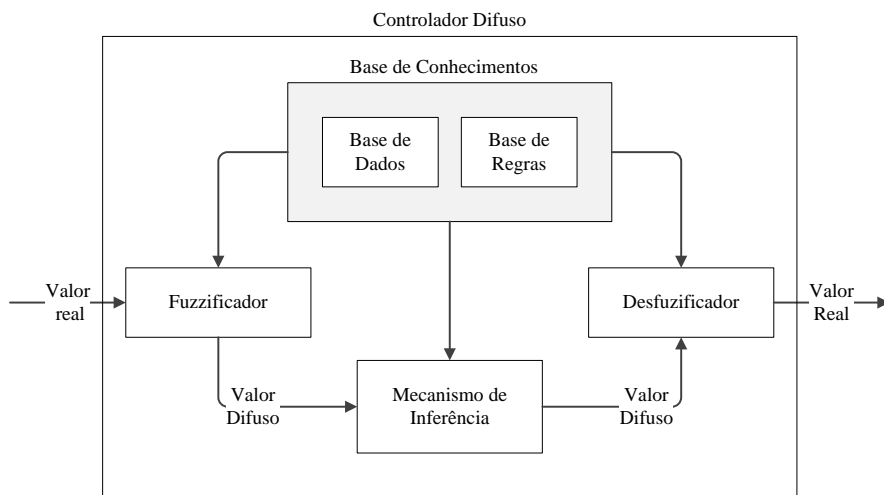


Figura 3.1: Estrutura do controlador lógico difuso.

quando existem dados corrompidos por ruído ou *outliers* nas leituras, é aconselhável recorrer-se a outro tipo de fuzzificador, como por exemplo um fuzzificador *nonsingleton*.

3.2.2 Base de Conhecimento

A Base de Conhecimento é constituída por dois módulos: a Base de Dados e a Base de Regras. A Base de Dados congrega toda a informação necessária para o correcto funcionamento do módulo de fuzzificação, da Base de Regras e do módulo de desfuzzificação. Esta informação inclui as funções de pertença, que representam o significado dos valores linguísticos das variáveis do controlador, bem como os domínios físicos e normalizados, juntamente com os factores de escala.

Base de Regras

A Base de Regras define inequivocamente a estratégia de controlo a implementar, concentrando o conhecimento que se pretende ver incorporado no controlador e tendo em conta o processo a controlar [Lee90, Ber88]. Tipicamente as regras são da seguinte forma:

Se <estado do processo> Então <saída do controlador>

A parte do "Se <>" de uma regra é designada por antecedente e apresenta-se como uma descrição do estado do processo em termos de uma combinação lógica de proposições difusas atómicas. A parte subordinada ao "Então <>", designa-se por conseqüente e traduz a descrição da saída do sistema difuso em termos de combinações lógicas de proposições difusas. A semântica correspondente à formalização de uma regra toma assim a seguinte forma:

Se <antecedente> Então <consequente>

Estas proposições determinam os valores linguísticos que as variáveis de saída do controlador tomam, sempre que haja uma correspondência, pelo menos para um certo grau, do estado do processo no antecedente da regra. As etapas envolvidas na construção da base de regras de um controlador difuso são as seguintes:

- Escolha das variáveis linguísticas de entrada e saída do controlador;
- Escolha do conjunto de termos para as variáveis linguísticas;
- Síntese do conjunto de regras.

Base de Dados

A Base de Dados é composta pelas funções de pertença e pelos factores de escala (ganhos do controlador) de um controlador difuso. As propriedades das respectivas funções de pertença que descrevem os termos linguísticos afectam fortemente o desempenho do controlador.

Sobreposição e simetria

O grau de sobreposição entre duas funções de pertença é definido como sendo o grau de pertença correspondente ao ponto do universo de discurso em que se intersectam, sendo este um valor compreendido entre 0 e 1. Como duas funções de pertença adjacentes podem intersectar-se em mais do que um ponto, define-se taxa de sobreposição como sendo o número de intersecções que ocorre (ver Figura 3.2).

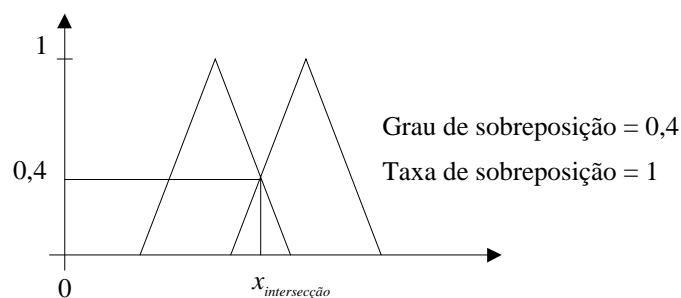
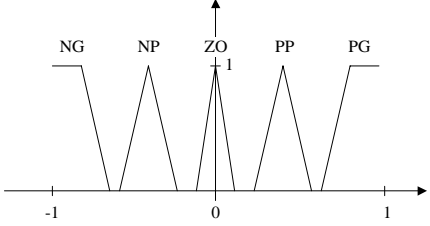
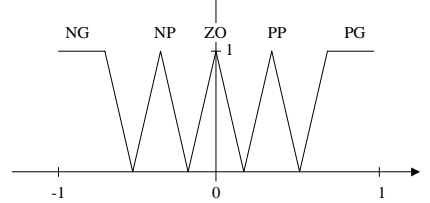


Figura 3.2: Grau de sobreposição e taxa de sobreposição

Existem, contudo, alguns casos particulares no que se refere à sobreposição das funções de pertença, nomeadamente quando a taxa de sobreposição é zero ou quando o grau de sobreposição é zero. As situações particulares descritas, assim como as suas implicações em termos de controlo de processos encontram-se discriminadas na Tabela 3.2.

Relativamente ao valor considerado adequado para o grau de sobreposição Kosko [Kos92] recomenda um valor mínimo de 0,25 e 0,5 como valor adequado, enquanto Oliveira [Oli96] mostrou que 0,5 se apresenta como um valor óptimo para o grau de sobreposição.

Tabela 3.2: Casos particulares da sobreposição de funções de pertinência.

Caso particular	Implicação no sistema de controlo
<p data-bbox="284 434 624 465">Taxa de sobreposição zero</p> 	<p data-bbox="715 465 1289 676">Existem valores crespos de entrada para os quais nenhuma regra disparará. Logo, não é calculado nenhum valor para a saída do controlador, surgindo uma zona morta ou zona sem intervenção do controlador, o que cria descontinuidades na acção de controlo</p>
<p data-bbox="284 846 624 878">Grau de sobreposição zero</p> 	<p data-bbox="715 967 1225 999">Apenas dispara uma regra de cada vez.</p>

No que se refere à taxa de sobreposição entre as funções de pertinência, esta deve ser igual a 1, o que se traduz na sua simetria. Segundo Kosko [Kos92] estes valores para o grau de sobreposição e para a taxa de sobreposição, respectivamente 0,5 e 1, permitem obter uma sobrelevação significativamente menor e um tempo de subida mais rápido.

Largura

Outra propriedade importante das funções de pertinência diz respeito à largura. Nos controladores difusos é frequente encontrar funções de pertinência com larguras diferentes para diferentes intervalos do universo de discurso. As funções de pertinência com maior largura tornam o sistema de controlo menos preciso, sendo preferível a selecção de funções de pertinência com largura comparativamente mais reduzida, para que se obtenha uma granularidade mais fina. Contudo, para que o universo de discurso seja abrangido por completo, tal implica a incorporação de um número superior de termos linguísticos, aumentando inevitavelmente a complexidade do controlador difuso.

3.2.3 Mecanismo de inferência

A função básica do motor de inferência consiste em calcular o valor global da saída do controlador considerando as contribuições individuais de cada regra. Cada uma destas contribuições representa o valor da saída correspondente ao disparo de uma regra, isoladamente. As saídas do módulo de fuzificação, representando os valores crespos concorrentes das variáveis de estado do processo, são comparadas com os antecedentes de cada regra de forma a determinar a intensidade com que a regra dispara. Com base na agregação desta informação calcula-se o valor difuso de saída do controlador.

Tal como foi referido na secção 2.3, segundo Driankov, existem dois tipos de fundamentais de heurísticas implementáveis no motor de inferência, designadamente, a inferência baseada na composição de regras e a inferência baseada em regras individuais, sendo a primeira a mais utilizada dado ser computacionalmente menos exigente em termos de memória requerida. Na Figura 3.3 apresenta-se o funcionamento da inferência baseada em regras individuais de Mamdani [Lee90], encontrando-se representadas duas regras descritas por termos linguísticos associados aos conjuntos difusos A_1, B_1, C_1 e A_2, B_2, C_2 . Sejam e_0 e Δe_0 os valores das variáveis de entrada do controlador, por exemplo, o erro e a variação do erro no instante inicial. Para os antecedentes de cada regra existe um grau de sobreposição entre as entradas crespas e_0 e Δe_0 e os conjuntos difusos, vd. Figura 3.3. Seguidamente é utilizado o mecanismo de inferência de Mandani (max – min) de modo a obter-se o valor global da saída do controlador. Considere-se então as seguintes regras:

$$R1 : \text{ Se } E \text{ é } A_1 \text{ e } \Delta E \text{ é } B_1 \text{ então } \Delta U \text{ é } C_1 \quad (3.2)$$

$$R2 : \text{ Se } E \text{ é } A_2 \text{ e } \Delta E \text{ é } B_2 \text{ então } \Delta U \text{ é } C_2 \quad (3.3)$$

em que $E, \Delta E$ e ΔU são os termos linguísticos associados às variáveis erro, variação do erro e variação da acção de controlo.

Finalmente, para se obter um valor crespo (real) como saída do controlador, ter-se-á de recorrer a um dos métodos de desfuzificação aplicável à correspondente saída difusa do sistema difuso.

3.2.4 Módulo de desfuzificação

O processo de desfuzificação desempenha um papel fulcral na realização de controladores difusos, pois fornece o meio de comunicação entre a representação difusa do controlador e mundo crespo dos sistemas dinâmicos. A desfuzificação, em si, consiste na selecção de um valor crespo a partir da saída difusa do controlador.

Os métodos de desfuzificação mais utilizados são:

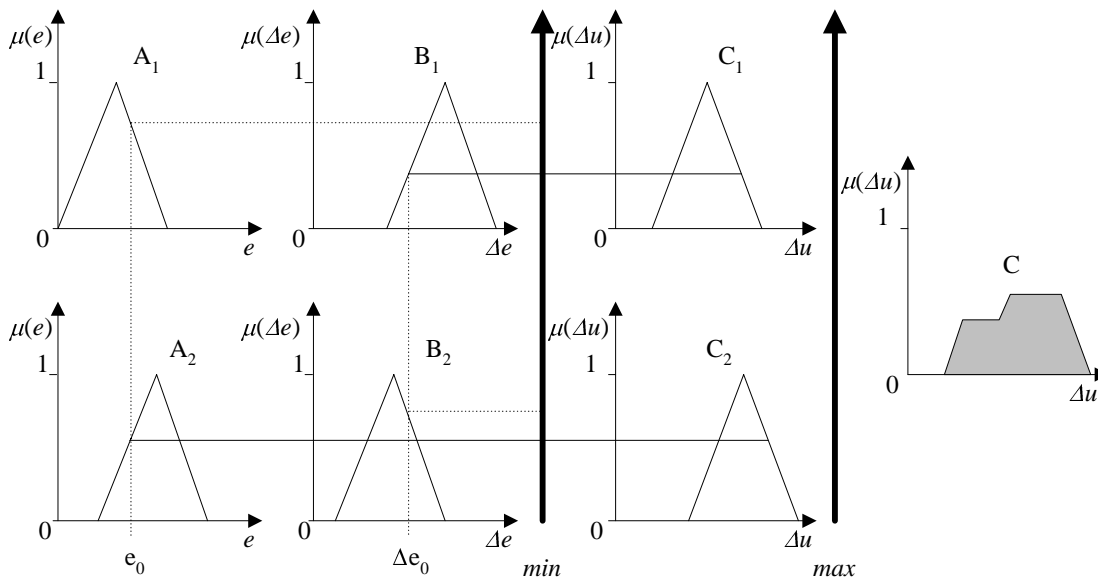


Figura 3.3: Método de inferência de Mamdani baseado em regras individuais.

- **Centroide:** o valor crespo desfuzificado de um conjunto difuso C é dado por:

$$Z_0 = \frac{\sum_W zC(z)dz}{\sum_W C(z)dz} \quad (3.4)$$

- **Primeiro dos máximos:** o valor crespo desfuzificado de um conjunto difuso C é obtido segundo:

$$Z_o = \min z|C(z) = \max_w C(w) \quad (3.5)$$

- **Método da média dos máximos:** valor crespo desfuzificado de um conjunto difuso C é definido como a média de todos os valores do universo de discurso, ou seja:

$$Z_o = \frac{1}{N} \sum_{j=1}^N z_j \quad (3.6)$$

3.3 Controladores PID difusos

Os controladores difusos diferem dos controladores convencionais na medida em que utilizam conhecimento expresso de uma forma natural expresso através de um conjunto de regras e variáveis linguísticas, cujos valores são descritos por conjuntos difusos.

A ideia básica subjacente ao controlador PID (Proporcional-Integral-Derivativo) convencional é escolher a lei de controle considerando o erro, o integral do erro e a derivada do

erro em ordem ao tempo de acordo com a equação (3.7).

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t) \quad (3.7)$$

em que K_p é o ganho proporcional, K_i o ganho integral e K_d o ganho derivativo do controlador. Para se obter a versão incremental do controlador PID convencional determina-se em primeiro lugar a derivada da equação 3.7, obtendo-se:

$$\dot{u}(t) = K_p \dot{e}(t) + K_i e(t) + K_d \ddot{e}(t) \quad (3.8)$$

Quanto à correspondente versão discreta, esta pode ser aproximada pela seguinte expressão:

$$\Delta u(k) = K_p \Delta e_k + K_i e_k + K_d \Delta^2 e_k \quad (3.9)$$

Esta secção aborda o projecto de controladores difusos com base no método de inferência de Mamdani, nomeadamente, o controlador PD (Proporcional-Derivativo), PI (Proporcional-Integral) e PID.

Existem dois tipos básicos de controladores de Mamdani que utilizam como variáveis de entrada o erro e a variação do erro, designadamente, o controlador PD difuso e o controlador PI, que disponibilizam respectivamente a acção de controlo e o seu incremento. Relativamente ao controlador PID de Mamdani, este pode ser obtido com base na agregação das saídas dos controladores PD e PI.

3.3.1 Controlador PD difuso

O controlador PD difuso utiliza como variáveis linguísticas de entrada o erro e a variação do erro e apresenta como variável linguística de saída a acção de controlo. A lei de controlo pode ser expressa por:

$$u_k = f(e_k, \Delta e_k) \quad (3.10)$$

em que $f(\cdot)$ representa o mecanismo interno do controlo difuso que permite transformar as variáveis de entrada em variáveis de saída.

A estrutura do controlador PD difuso encontra-se na Figura 3.4, onde K_e e $K_{\Delta e}$ são os factores de escala associados às entradas e K_u o factor de escala afecto à saída.

Os factores de escala têm um papel idêntico ao dos ganhos nos controladores convencionais, sendo de grande importância no que diz respeito ao desempenho e estabilidade do controlador, isto é, por vezes podem ser fonte de instabilidade, oscilações e deterioração dos efeitos de amortecimento.

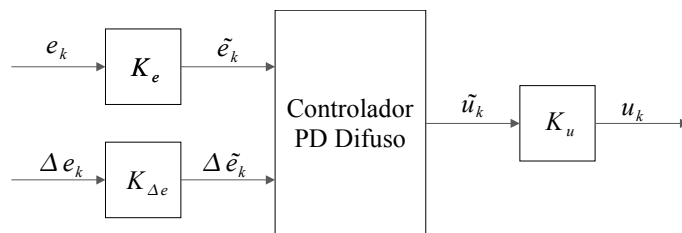


Figura 3.4: Estrutura do controlador PD Difuso.

As regras do controlador apresentam a seguinte forma:

$$\text{Regra } i: \text{ Se } E \text{ é } LE^{(i)} \text{ e } \Delta E \text{ é } L\Delta E^{(i)} \text{ então } U \text{ é } LU^{(i)} \quad (3.11)$$

em que $LE^{(i)}$, $L\Delta E^{(i)}$ e $LU^{(i)}$ são os valores linguísticos, respectivamente, das variáveis E , ΔE e U referentes à regra i . Na tabela 3.3 é apresentada uma base de regras típica do controlador PD difuso [LG96]. Para o caso apresentado o número de variáveis linguísticas associadas às variáveis do controlador (erro, variação do erro e acção de controlo) é sete.

Tabela 3.3: Base de regras genérica para o controlador PD Difuso.

e	Δe	NG	NM	NP	ZO	PP	PM	PG
NG		<i>NG</i>	<i>NG</i>	<i>NG</i>	<i>NG</i>	<i>NM</i>	<i>NP</i>	<i>NP</i>
NM		<i>NG</i>	<i>NG</i>	<i>NG</i>	<i>NM</i>	<i>NP</i>	<i>NP</i>	<i>NP</i>
NP		<i>NG</i>	<i>NG</i>	<i>NM</i>	<i>NP</i>	<i>NP</i>	<i>NP</i>	<i>NP</i>
ZO		<i>NP</i>	<i>NP</i>	<i>NP</i>	<i>ZO</i>	<i>PP</i>	<i>PP</i>	<i>PP</i>
PP		<i>PP</i>	<i>PP</i>	<i>PP</i>	<i>PP</i>	<i>PM</i>	<i>PG</i>	<i>PG</i>
PM		<i>PP</i>	<i>PP</i>	<i>PP</i>	<i>PM</i>	<i>PG</i>	<i>PG</i>	<i>PG</i>
PG		<i>PP</i>	<i>PP</i>	<i>PM</i>	<i>PG</i>	<i>PG</i>	<i>PG</i>	<i>PG</i>

3.3.2 Controlador PI difuso

O controlador PI difuso é caracterizado por ter como variáveis de entrada o erro e a variação do erro, e como variáveis de saída o incremento da acção de controlo. A obtenção da acção de controlo correspondente ao controlador PI é calculada com base no operador somador, podendo esta ser expressa do seguinte modo:

$$\Delta u_k = f(e_k, \Delta e_k) \quad (3.12)$$

em que $f(\cdot)$ representa o mecanismo interno do controlador.

A estrutura do controlador PI difuso encontra-se na Figura 3.5. K_e e $K_{\Delta e}$ são os factores de escala associados às entradas e $K_{\Delta u}$ o factor de escala associado à saída crespça.

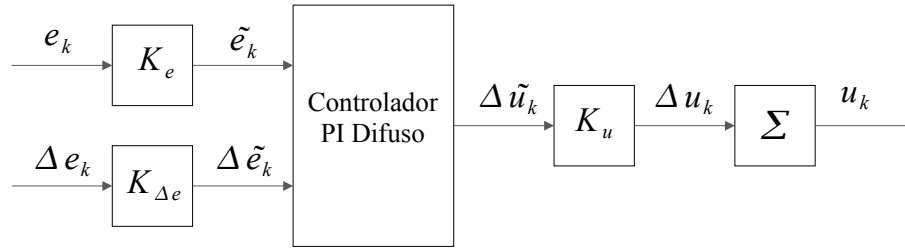


Figura 3.5: Estrutura do controlador PI Difuso.

As regras do controlador apresentam a seguinte forma:

$$\text{Regra } i: \text{ Se } E \text{ é } LE^{(i)} \text{ e } \Delta E \text{ é } L\Delta E^{(i)} \text{ então } \Delta U \text{ é } \Delta LU^{(i)} \quad (3.13)$$

em que $LE^{(i)}$, $L\Delta E^{(i)}$ e $\Delta LU^{(i)}$ são os valores linguísticos, respectivamente, das variáveis E , ΔE e ΔU na regra i . Na tabela 3.4 é apresentada uma base de regras típica de um controlador PI difuso [LG96]. Em relação ao caso apresentado consideraram-se 7 variáveis linguísticas para as variáveis do controlador (erro, variação do erro e acção de controlo).

Tabela 3.4: Base de regras genérica para o controlador PI Difuso.

$e \Delta e$	NG	NM	NP	ZO	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	ZO
NM	NG	NG	NG	NM	NP	ZO	PP
NP	NG	NG	NM	NP	ZO	PP	PM
ZO	NG	NM	NP	ZO	PP	PM	PG
PP	NM	NP	ZO	PP	PM	PG	PG
PM	NP	ZE	PP	PM	PG	PG	PG
PG	ZO	PP	PM	PG	PG	PG	PG

O mecanismo de inferência no caso dos controladores PI e PD funcionam de forma semelhante, residindo a única diferença na variável de saída. A selecção do tipo de controlo a utilizar é feita de acordo com o seu comportamento em regime transitório e estacionário. Em particular, utilizando um controlador PD é extremamente difícil eliminar o erro em regime estacionário, enquanto que utilizando um controlador PI não é possível obter um bom desempenho em regime transitório para sistemas de ordem elevada, em virtude da operação interna de integração.

No que se refere à base de regras dos controladores PD e PI, a sua principal diferença reside na linha de comutação da variável de saída entre valores positivos e negativos. No caso da base de regras do controlador PI difuso a linha de comutação é diagonal ($E = \Delta E$), enquanto que para o controlador PD tal ocorre para $E = ZO$. Estas linhas

encontram-se sombreadas nas Tabelas 3.3 e 3.4.

3.3.3 Controlador PID difuso

O controlador PID difuso pode ser construído adicionando uma terceira variável linguística, para além do erro e da variação do erro, que traduz a segunda diferença do erro, com a variável linguística de saída consistindo na variação da acção de controlo. A lei de controlo do controlador PID difuso pode ser expressa por:

$$\Delta u_k = f(e_k, \Delta e_k, \Delta^2 e_k) \quad (3.14)$$

em que $f(\cdot)$ representa o mecanismo interno do controlador difuso.

No que se refere à topologia do controlador PID difuso esta tem a estrutura apresentada na Figura 3.6, onde K_e , $K_{\Delta e}$ e $K_{\Delta^2 e}$ são os factores de escala associados às entradas e $K_{\Delta u}$ o factor de escala afecto à saída.

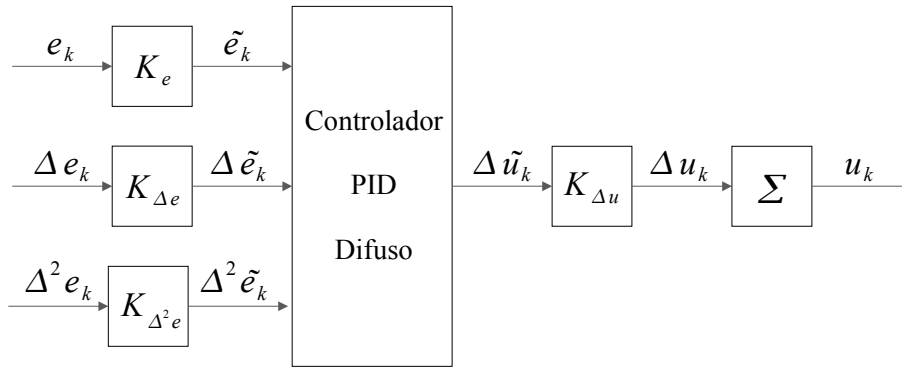


Figura 3.6: Estrutura do controlador PID Difuso.

As regras do controlador PID apresentam a seguinte forma:

$$\text{Regra } i: \text{ Se } E \text{ é } LE^{(i)}, \Delta E \text{ é } L\Delta E^{(i)} \text{ e } \Delta^2 E \text{ é } L\Delta^2 E^{(i)} \text{ então } \Delta U \text{ é } \Delta LU^{(i)} \quad (3.15)$$

em que $LE^{(i)}$, $L\Delta E^{(i)}$, $L\Delta^2 E^{(i)}$ e $\Delta LU^{(i)}$ são os valores linguísticos, respectivamente, das variáveis E , ΔE , $\Delta^2 E$ e ΔU na regra i . Para este controlador, a existência de três entradas faz com que a base de regras se expanda enormemente (n^3 regras, em que n é o número de termos linguísticos), tornando algo difícil o projecto e a interpretabilidade do controlador. Por outro lado, dado que um controlador difuso pretende representar o conhecimento de um operador experiente, é pouco realista esperar que este operador consiga determinar regras de actuação relativas à segunda ordem do erro ou superiores.

Por este facto, usualmente utiliza-se uma abordagem alternativa consistindo na agregação das saídas dos controladores PD e PI, ou seja:

$$u_k^{PID} = u_k^{PI} + u_k^{PD} \quad (3.16)$$

em que u_k^{PI} corresponde à saída disponibilizada por um controlador de velocidade do tipo PI difuso e u_k^{PD} a saída associada ao controlador de posição do tipo PD difuso. A utilização deste controlador tem como principal vantagem a redução de um sistema de regras tridimensional a um sistema de regras bidimensional, facilitando a tarefa de projecto e interpretação. A estrutura deste controlador encontra-se representada na Figura 3.7.

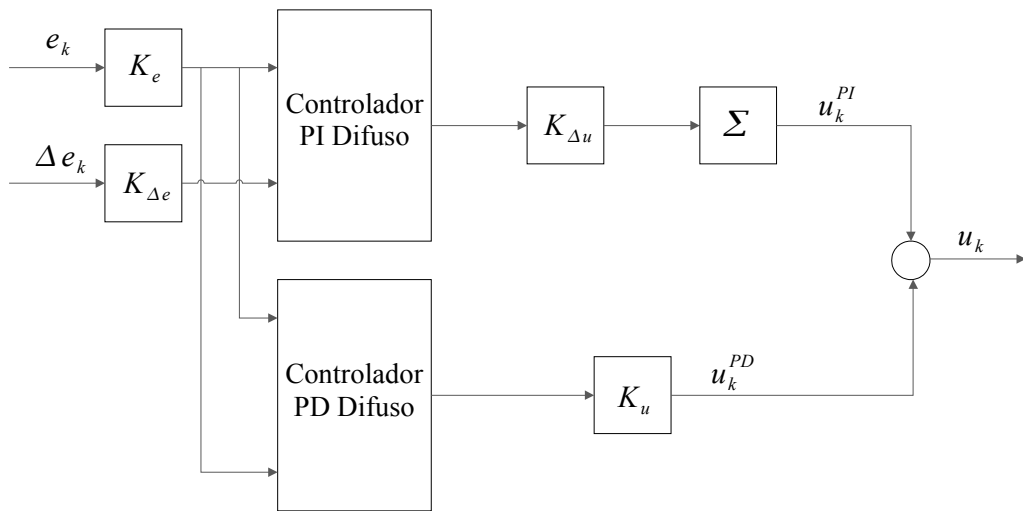


Figura 3.7: Estrutura do controlador PID Difuso híbrida.

4

Identificação de sistemas

A identificação de sistemas foi inicialmente utilizada para desenvolver modelos matemáticos lineares para os sistemas em estudo [BJ90]. No entanto, os sistemas reais são geralmente não lineares e a representação linear deste tipo de sistemas apresenta as suas naturais limitações. Um modelo linear permite descrever razoavelmente a dinâmica de um sistema não linear apenas numa estreita vizinhança do ponto de funcionamento (linearização local de um sistema não-linear) [Ast94].

O interesse pela identificação de modelos não lineares paramétricos surgiu no início da década de 80. Grande parte do interesse neste tipo de modelos foi motivado pelo desenvolvimento de ferramentas matemáticas e computacionais que simplificam a obtenção de tais modelos.

As abordagens desenvolvidas para a identificação de sistemas podem ser divididas em três grupos [Lju87]:

- Métodos paramétricos;
- Métodos não paramétricos;
- Métodos no domínio da frequência.

Os métodos paramétricos utilizam estruturas matemáticas parametrizadas para descrever o comportamento dinâmico do sistema no domínio do tempo. Os parâmetros destas estruturas são ajustados por algoritmos de estimação a partir de dados previamente coligidos dos sistemas a identificar [Lju87]. Os métodos ditos não paramétricos geram também modelos no domínio do tempo, diferindo de aqueles no facto de o comportamento

dinâmico do sistema ser determinado através de funções de correlações calculadas sobre os dados disponíveis. Por fim os métodos no domínio da frequência geram modelos representados neste domínio.

4.1 Processo de Identificação

O problema de identificação de sistemas pode ser dividido em cinco etapas principais [Lju87]:

- **Obtenção dos dados do sistema a modelar:** envolve a definição dos sinais a medir, a caracterização dos sinais de excitação e escolha do intervalo de amostragem;
- **Recolha de dados:** Obtenção das saídas para o conjunto de entradas de excitação, com um intervalo de amostragem previamente definido;
- **Escolha da estrutura de modelos:** selecção da topologia utilizada para representar o sistema dinâmico tendo em conta algum conhecimento sobre o sistema em causa;
- **Estimação dos parâmetros do modelo:** por meio de uma metodologia específica os parâmetros da estrutura pré-seleccionada são manipulados por forma que o modelo reproduza o comportamento dinâmico do sistema expresso pelos dados;
- **Validação do modelo:** através de testes avalia-se a qualidade do modelo obtido.

Na Figura 4.1 apresentam-se as etapas inerentes ao processo de identificação, utilizando para o efeito um fluxograma [BG07]. A partir deste fluxograma é possível constatar que o utilizador interage com o computador formando uma dupla malha de refinamento do modelo e selecção da estrutura adequada.

O núcleo do problema de identificação consiste em encontrar a estrutura adequada para a captura da respectiva dinâmica, de modo a obter um modelo que o represente adequadamente. Na fase de escolha da estrutura do modelo deve-se utilizar o conhecimento *a priori* que se tem sobre sistema, pois uma das regras básicas da identificação de sistemas é *Não estimar aquilo que já se sabe* [SZL⁺95]. Existem diversas formas e técnicas para se obter o modelo de um sistema, sendo estas classificadas de acordo com o nível de conhecimento utilizado na selecção do tipo de estrutura de modelos [Lju87]:

- **Modelos caixa-branca:** A utilização deste tipo de modelos nem sempre é viável, pois requer o conhecimento dos fenómenos que regem o comportamento do sistema em questão. Para além do conhecimento necessário para a modelação de sistemas segundo esta abordagem, outro factor que por vezes a inviabiliza é o correspondente tempo de síntese.

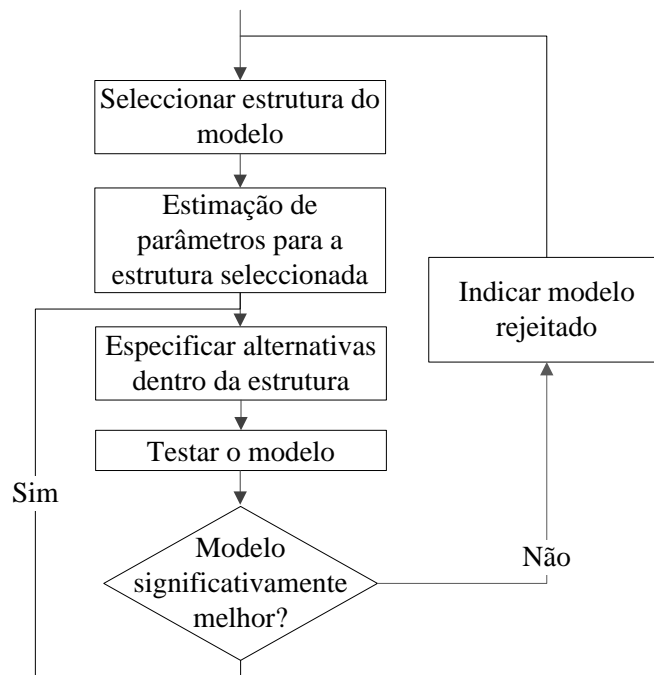


Figura 4.1: Fluxograma ilustrativo do procedimento de refinamento de estrutura.

- **Modelos caixa-cinzenta:** Nestes modelos é conhecido parte do comportamento físico do sistema, no entanto existem ainda alguns parâmetros que necessitam de ser obtidos. Consideraram-se dois sub-casos [SZL⁺95]:

I Modelação física - A estrutura do modelo pode ser obtida recorrendo aos fenómenos físicos, contendo um certo número de parâmetros estimados por informação retirada do sistema.

II Modelação semi-física - O conhecimento físico do processo é utilizado para selecção dos sinais a recolher. Estes sinais são então submetidos a uma modelação do tipo caixa-preta.

- **Modelos caixa-preta:** Nenhum conhecimento físico está disponível ou é utilizado na modelação do sistema.

4.2 Métodos paramétricos

Os métodos paramétricos são métodos usados para estimar os parâmetros ou vector de parâmetros θ de um certo modelo, sendo os dados utilizados para identificar os modelos dinâmicos denominados de dados de estimação. Estes são gerados por amostragem da resposta $y(k)$ do sistema, excitado com um sinal de entrada $u(k)$ pré-especificado. O espectro do sinal de entrada $u(k)$ deve ser suficientemente amplo, tanto em amplitude como em frequência, para capturar correctamente a dinâmica do sistema a modelar.

A estrutura geral dos modelos paramétricos (ver Figura 4.2) pode ser definida pela seguinte função de transferência discreta (4.1).

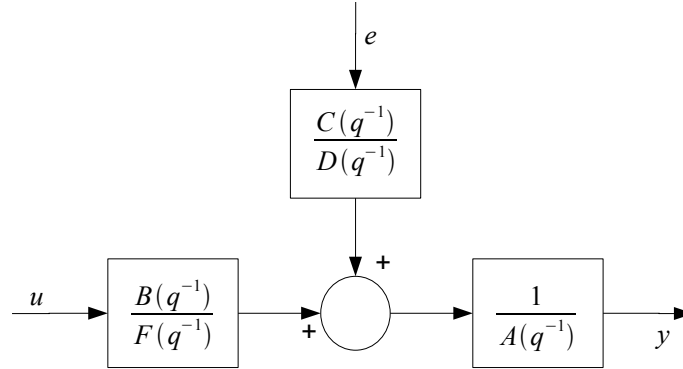


Figura 4.2: Modelo geral para identificação paramétrica.

$$A(q^{-1})y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k - nT) + \frac{C(q^{-1})}{D(q^{-1})}e(k) \quad (4.1)$$

em que q^{-1} representa o operador de atraso e

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_1q^{-1} + b_2q^{-2} + \dots + b_{nb}q^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \\ D(q^{-1}) &= 1 + d_1q^{-1} + \dots + d_{nd}q^{-nd} \\ F(q^{-1}) &= 1 + f_1q^{-1} + \dots + f_{nf}q^{-nf} \end{aligned} \quad (4.2)$$

onde n_a, n_b, n_c, n_d e n_f são as ordens dos respectivos polinómios.

É possível verificar que $A(q^{-1})$ corresponde aos pólos comuns à modelação da dinâmica do sistema e ao ruído. Já $F(q^{-1})$ e $B(q^{-1})$ representam os pólos e os zeros que afectam apenas a entrada e $D(q^{-1})$ e $C(q^{-1})$ os pólos e os zeros que afectam o ruído.

4.2.1 Modelo ARX

O modelo ARX (Auto Regressive with eXogenous input) é obtido a partir do modelo geral (4.1) com $C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$, dando origem à equação às diferenças (4.3) em que nT é o número de atrasos da entrada para a saída:

$$A(q^{-1})y(k) = B(q^{-1})u(k - nT) + e(k) \quad (4.3)$$

A estrutura ARX é uma das mais utilizadas para efeito de identificação de sistemas em virtude do respectivo predictor ser de simples interpretação, tendo no entanto a desvantagem do polinómio $A(q^{-1})$ afectar tanto a dinâmica do sistema como o ruído (ver Figura 4.3).

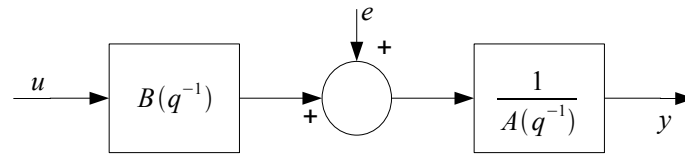


Figura 4.3: Modelo ARX.

4.2.2 Modelo ARMAX

O modelo ARMAX (Auto Regressive Moving Average with exogenous input) também obtido a partir de (4.1) é definido como:

$$A(q^{-1})y(k) = B(q^{-1})u(k - nT) + C(q^{-1})e(k) \quad (4.4)$$

Este modelo contém uma flexibilidade extra, no que se refere à modelação do ruído, em relação ao modelo ARX (ver Figura 4.4).

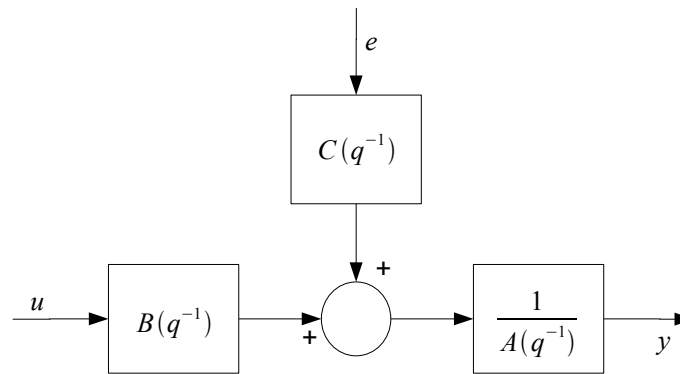


Figura 4.4: Modelo ARMAX.

4.2.3 Modelo erro de saída

O modelo OE (*Output Error*) é definido como:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k - nT) + e(k) \quad (4.5)$$

Este modelo tem a vantagem de a dinâmica do sistema ser modelada de forma separada, não tendo nenhum parâmetro que modele o ruído (ver Figura 4.5).

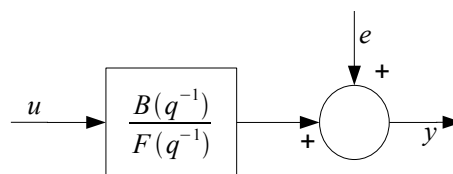


Figura 4.5: Modelo OE.

4.2.4 Modelo de Box Jenkins

O modelo Box Jenkins (BJ) pode ser igualmente obtido a partir de (4.1), apresentando a seguinte expressão:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k - nT) + \frac{C(q^{-1})}{D(q^{-1})}e(k) \quad (4.6)$$

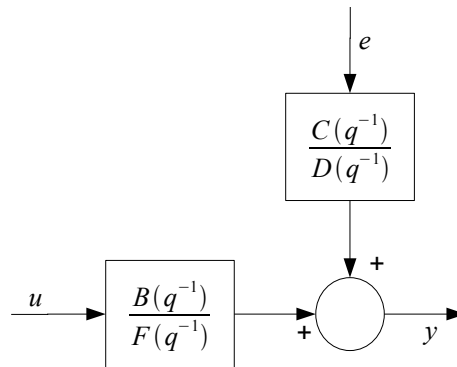


Figura 4.6: Modelo BJ.

O modelo BJ modela tanto o ruído como a dinâmica do sistema, embora separadamente (ver Figura 4.6).

4.2.5 Modelo no espaço de estado

Os modelos lineares no espaço de estado representam o mesmo tipo de relação linear entre a entrada e a saída do sistema expressa através de uma função de transferência discreta. As equações que representam o modelo de estado podem ser agrupadas sob a forma de vectores e matrizes:

$$x(k + 1) = Ax(k) + Bu(k) + Ke(k) \quad (4.7)$$

$$y(k) = Cx(k) + Du(k) + e(k) \quad (4.8)$$

onde $x(k)$ é o vector de estado. A matriz A representa a matriz da dinâmica e relaciona as variáveis de estado com o seu valor anterior, a matriz B é a matriz de entrada e relaciona as variáveis de estado com os sinais de entrada, a matriz C é a matriz de saída e relaciona as variáveis de estado com os sinais de saída e D a matriz de transmissão directa. A matriz K determina as propriedades do ruído. Para $K = 0$ o ruído afecta apenas a saída do sistema. Para $D = 0$ verifica-se uma influência instantânea do vector de estados no sinal de saída.

Para a obtenção da matriz de transferência a partir do modelo de estado, considerando o caso particular em que o erro é nulo, aplica-se o operador Transformada Z às equações

(4.8) e (4.7) obtendo-se:

$$zX(z) = AX(z) + BU(z) \quad (4.9)$$

$$Y(z) = CX(z) + DU(z) \quad (4.10)$$

A equação (4.9) pode tomar a seguinte forma:

$$X(z) = (zI - A)^{-1}BU(z) \quad (4.11)$$

em que I é a matriz identidade com dimensão igual a A. Substituindo (4.11) em (4.10) obtém-se a função de transferência a partir do modelo no espaço de estados:

$$\frac{Y(z)}{U(z)} = \{C(z^{-1}I - A)^{-1}B + D\} \quad (4.12)$$

4.3 Métodos dos mínimos quadráticos

Neste método o vector de parâmetros, θ , do modelo é escolhido de modo a minimizar a diferença entre a saída do modelo (predição) e a saída medida. Considere-se então o preditor definido pela seguinte regressão linear:

$$\hat{y}(k|\theta) = \varphi^T(k)\theta \quad (4.13)$$

com φ o vector de regressão constituído pelas entradas e saídas passadas. Para o caso do modelo ARX o seu regressor apresenta a seguinte forma:

$$\varphi(k) = [-y(k-1) \quad \dots \quad -y(k-n_a) \quad u(k-1) \quad \dots \quad u(k-n_b)]^T \quad (4.14)$$

onde pares $\{[y(k), \varphi(k)], k = 1, \dots, N\}$ são obtidos por meio de ensaios sobre o sistema a identificar.

O problema do método dos mínimos quadráticos consiste na determinação dos parâmetros de θ de modo a minimizar o seguinte critério:

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N [y(k) - \varphi^T(k)\theta]^2 \quad (4.15)$$

sendo o erro de modelação dado por:

$$\epsilon(k, \theta) = y(k) - \varphi^T(k)\theta \quad (4.16)$$

e

$$Z^N = \{y(k), u(k)\}, \quad k = 1, 2, \dots, N \quad (4.17)$$

Deste modo, o vector de parâmetros θ é escolhido de forma a minimizar (4.15), ou seja:

$$\hat{\theta}_N = \arg \min_{\theta} V_N(\theta, Z^N) \quad (4.18)$$

4.4 Validação de modelos

Validar um modelo pode ser visto como a avaliação da capacidade de um dado modelo representar de forma adequada o comportamento dinâmico do sistema a identificar. Dois métodos utilizados na validação de modelos são a validação cruzada e a análise dos resíduos.

4.4.1 Validação cruzada

Consiste em comparar, segundo um critério $V(\cdot)$, o desempenho do modelo, para um conjunto de dados diferente dos utilizados para efeito de estimação dos parâmetros do modelo. Para tal, simula-se o comportamento do sistema recorrendo ao conjunto de validação e utiliza-se, por exemplo, a seguinte métrica:

$$V(\theta, Z^N) = \frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k|\hat{\theta})]^2 \quad (4.19)$$

Se o somatório dos erros quadráticos for da ordem de grandeza do obtido com base no conjunto de estimação, então o modelo poderá ser considerado adequado. Caso tal não se verifique terá de se procurar outro modelo, alterando a ordem dos polinómios ou seleccionando outra estrutura (ver Figura 4.1).

4.4.2 Teste dos resíduos

Os resíduos entre as saídas do sistema e do modelo deverão apresentar-se como um ruído branco e serem independentes da entrada para que o modelo em causa possa descrever adequadamente o sistema.

Análise do erro de estimação

Seja a covariância definida por:

$$\hat{R}_{\epsilon}(\tau) = \frac{1}{N} \sum_{k=1}^{N-r} \epsilon(k)\epsilon(k+\tau) \quad (4.20)$$

e a covariância normalizada dada por:

$$\bar{R}_\epsilon(\tau) = \frac{\hat{R}_\epsilon(\tau)}{\hat{R}_\epsilon(0)} \quad (4.21)$$

A condição prática para que os resíduos possam ser considerados uma sucessão de variáveis aleatórias independentes com um nível de confiança de 95% é:

$$|\bar{R}_\epsilon(\tau)| \leq \frac{1.96}{\sqrt{N}}; \quad \tau \geq 1 \quad (4.22)$$

Independência entre resíduos e entradas passadas

Este teste permite verificar se existe informação contida na saída, originada pela entrada $u(k - \tau)$, que não seja reproduzida de forma adequada pelo modelo. Caso o teste falhe dever-se-á rever a ordem do polinómio $B(q^{-1})$ e o atraso puro do sistema. Para este teste a expressão da covariância toma a seguinte forma:

$$\hat{R}_{\epsilon u}(\tau) = \frac{1}{N} \sum_{k=r+1}^N \epsilon(k) \tilde{u}(k - \tau) \quad (4.23)$$

em que,

$$\tilde{u}(k) = u(k) - \bar{u}(k) \quad (4.24)$$

A covariância normalizada é dada por:

$$\bar{R}_{\epsilon u} = \frac{\hat{R}_{\epsilon u}(\tau)}{\sqrt{\hat{R}_\epsilon(0) \hat{R}_u(0)}} \quad (4.25)$$

Neste contexto, a condição prática de aceitação assumindo um nível de confiança de 95% vem dada por:

$$|\bar{R}_{\epsilon u}(\tau)| \leq \frac{1.96}{\sqrt{N}}; \quad \tau \geq 0 \quad (4.26)$$

No caso de existir uma eventual correlação entre os resíduos e entradas futuras, tal significa a existência de efeito de retroacção, pelo que a sua manifestação não implica a rejeição do modelo.

4.4.3 Actualização recursiva

Considere-se um sistema discreto SISO descrito por um modelo ARX (ver Figura 4.3) na forma:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) \quad (4.27)$$

em que d representa o atraso puro do sistema. O estimador $\hat{\theta}$ é actualizado recursivamente através do seguinte algoritmo:

$$P(k) = \frac{P(k-1)}{\lambda} \frac{\varphi(k)\varphi(k)^T P(k-1)}{\lambda + \varphi(k)^T P(k-1)\varphi(k)} \quad (4.28)$$

$$e(k) = y(k) - \hat{y}(k) \quad (4.29)$$

$$\hat{\theta} = \hat{\theta}(k-1) + P(k)\varphi(k)e(k) \quad (4.30)$$

onde $P(k)$ é a matriz de covariância e λ o factor de esquecimento.

4.5 Identificação de sistemas não lineares

A identificação de sistemas baseada em métodos paramétricos foi inicialmente utilizada na obtenção de modelos lineares [BJ90]. Os sistemas físicos são, no entanto, geralmente não lineares, pelo que a utilização de modelos lineares tem as suas limitações naturais. Na essência, os modelos lineares de sistemas dinâmicos permitem descrever o comportamento dinâmico de sistemas não lineares na vizinhança estreita de um ponto de funcionamento.

Uma das questões centrais à implementação de metodologias de controlo baseadas em modelos matemáticos reside na sua susceptibilidade a eventuais enviesamentos do modelo, podendo o sistema em anel fechado, em situações extremas, vir a manifestar um comportamento instável. Uma das linhas de acção adoptadas para contornar o problema referido consiste na utilização de modelos não lineares no processo de identificação.

4.5.1 Modelos não lineares

Devido à presença de não linearidades na dinâmica dos sistemas a utilização de modelos lineares encontra-se restringida a uma estreita gama de operação em torno do ponto de funcionamento para o qual o modelo foi obtido. Fora desta gama verifica-se geralmente uma degradação do desempenho do modelo. Deste modo os modelos não lineares têm sido fortemente estudados, uma vez que apresentam um melhor desempenho num amplo conjunto de sistemas e regimes de operação.

No caso particular das redes neuronais como estruturas de modelos no contexto da modelação e controlo de sistemas dinâmicos não lineares [NP90a], numerosas aplicações têm surgido neste contexto devido à intrínseca capacidade de aproximação que as redes neuronais apresentam (ver [RC95a, RC95b, PSPY97]).

Um dos tipos de redes neuronais mais frequentemente utilizado no âmbito do controlo

de sistemas são as redes neuronais proativas multicamadas (NNARX) com função de activação sigmoideal.

4.5.2 Estruturas do tipo NNARX

As redes neuronais proativas multicamada não apresentam na sua forma convencional qualquer tipo de mecanismo de processamento de informação temporal. Um meio relativamente simples de ultrapassar esta limitação consiste na especificação de um vector de entrada constituído por entradas e saídas passadas, convertendo-se assim a informação temporal numa sucessão espacial. Esta abordagem foi sugerida independentemente por Chen *et al.* [CBG90] e por Narendra & Parthasarathy [NP90b] e alicerça-se no trabalho de Takens [Tak81] que demonstrou a possibilidade de reconstrução da dinâmica interna de um sistema com base num vector de dimensão finita de entradas e saídas passadas.

Considere-se então um sistema dinâmico não linear em tempo discreto, descrito pela seguinte equação às diferenças:

$$y(k) = f(y(k-1), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)) \quad (4.31)$$

onde $u(k)$ e $y(k)$ representam a entrada e a saída do sistema no instante k , n_a e n_b traduzem a ordem das saídas e entradas e f é uma função não linear a aproximar no decurso do processo de identificação.

Pelo facto de uma rede neuronal proativa multicamada apresentar uma ou mais camadas internas preenchidas por neurónios caracterizados por funções de activação sigmoideal é em teoria capaz de reproduzir o comportamento dinâmico de um sistema genérico descrito por (4.31). No entanto não está estabelecido um limite em relação ao número de neurónios a incluir nas várias camadas, de modo a obter um dado nível de aproximação, nem a ordem do vector de regressão.

Uma rede neuronal do tipo NNARX (*Neural Network Auto Regressive with eXogenous input*) apresenta como entrada uma pilha finita de entradas e saídas passadas. Para estas estruturas, a parametrização assume a seguinte forma:

$$\hat{y}(k) = g(\varphi(k), W) \quad (4.32)$$

onde $g(\cdot)$ representa a transformação não linear proporcionada pela rede neuronal, W as matrizes de pesos e $\varphi(k)$ o vector de regressão no instante k . A estrutura NNARX pode ser formalmente representada por um esquema de blocos, onde entradas e saídas passadas são obtidas através de operadores temporais de atraso, vd. Figura. 4.7.

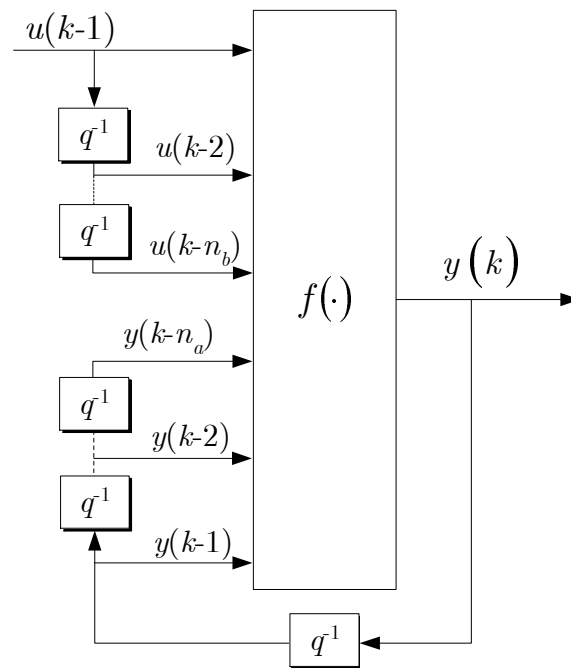


Figura 4.7: Esquema de blocos de uma estrutura NARX.

Selecionando um vector de regressão associado ao sistema a modelar,

$$\varphi(k) = [y(k-1) \quad \dots \quad y(k-n_a) \quad \dots \quad u(k-1) \quad \dots \quad u(k-n_b)]^T \quad (4.33)$$

a expressão geral correspondente à saída da rede neuronal $\hat{y}(k)$ toma a seguinte forma:

$$\hat{y}(k) = g(y(k-1) \quad \dots \quad y(k-n_a) \quad \dots \quad u(k-1) \quad \dots \quad u(k-n_b)) \quad (4.34)$$

onde $g(\cdot) \equiv \hat{f}(\cdot)$ representa a estimativa de $f(\cdot)$ e $y(k)$ a saída do sistema no instante k .

5

Optimização

Embora não exista consenso na literatura em relação à definição de optimização, esta pode ser entendida como a tentativa de obter a melhor solução possível para um problema, atendendo a determinados critérios e obedecendo a certas restrições [Sch95]. Geralmente o problema de optimização consiste na procura de um extremo, máximo ou um mínimo, de uma função, designada por função de custo, que represente o problema em questão.

A complexidade do problema, traduzida na função que o representa, está directamente relacionada com a dificuldade de busca pela melhor solução possível para esse problema. Quando se trata de um problema simples pode ser aplicada uma busca directa, ou exaustiva, onde todo o espaço pode ser explorado e o ponto máximo, ou mínimo, pode ser encontrado por inspecção directa. Este método é vulgarmente conhecido como "Busca Exaustiva" ou "Método da Força Bruta" [LC99] e é raramente aplicável.

Baseado no tipo de função de custo para o problema de optimização podem-se classificar os métodos de optimização em dois grandes grupos: os métodos lineares e os métodos não lineares. O primeiro método é aplicado a problemas que podem ser modelados utilizando combinações lineares das suas variáveis e restrições. Estes são problemas de solução mais simples, utilizando-se para a sua resolução métodos de programação linear, destacando-se o Método Simplex [Sid82]. Numa série de problemas de engenharia a função de custo apresenta características não lineares, recorrendo-se nestas circunstâncias a métodos de optimização não lineares.

5.1 Optimização analítica e optimização numérica

Quando a função matemática que descreve o fenómeno (sistema) a otimizar pode ser descrita através de uma equação contínua e diferenciável, a busca do ponto óptimo pode ser feita analiticamente, calculando-se o ponto no qual as condições de derivada igual a zero são satisfeitas. Neste caso diz-se que a optimização é analítica.

No entanto, na maioria dos casos práticos a função de custo não tem solução analítica, pois não é contínua nem diferenciável e muitas vezes nem pode ser descrita em forma de equações, como é o caso de diversos tipos de simulações em engenharia, onde a função a ser optimizada é resultado de diversos processos. Nestes casos só é possível determinar um ponto de máximo ou mínimo da função através de algoritmos numéricos. Este tipo de optimização é conhecida como optimização numérica. A característica geral dos métodos numéricos é o comportamento de aproximações sucessivas, partindo-se de um ou mais pontos iniciais.

A aplicação de métodos numéricos na resolução de problemas físicos de grande dimensão torna mais fácil e rápida a obtenção da solução, considerando à partida, que a solução não é exacta mas aproximada com um determinado erro máximo conhecido. Existem diversas técnicas de optimização que podem ser aplicadas na resolução destes problemas, dependendo do tipo de problema.

5.2 Componentes do problema de optimização

Independentemente do método de optimização utilizado, podem identificar-se alguns elementos comuns:

- **Função de custo:** Função ou sistema que descreve o problema de optimização. É também denominada de função de mérito, função objectivo ou critério de desempenho;
- **Variáveis livres ou independentes:** Variáveis que não dependem de outras variáveis, onde cada combinação de variáveis livres define um ponto do espaço solução;
- **Variáveis dependentes:** Variáveis que de algum modo dependem das variáveis livres, ficando os seus valores definidos a partir dos valores das variáveis independentes;
- **Restrições:** Limitações impostas ao problemas, definindo limites de viabilidade/admissibilidade para as soluções encontradas e funcionando como recortes do espaço solução;

- **Espaço solução:** Conjunto de todas as soluções possíveis, viáveis e inviáveis do problema, representando o domínio da função objectivo. Expressa-se através de limites inferior e superior para cada variável livre do problema;
- **Ponto óptimo:** É o ponto do espaço solução onde a função objectivo é máxima ou mínima.

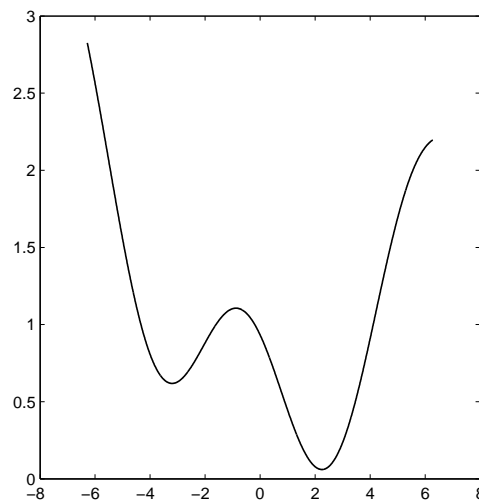
5.3 Classificação dos problemas de otimização

Os algoritmos ou técnicas de otimização dependem fundamentalmente do tipo de problema em questão. Existem diversas formas de classificar os problemas de otimização. A classificação a seguir apresentada foi proposta por Rao [Rao96].

- **Classificação baseada na existência de restrições:** Qualquer problema de otimização deve ser classificado de "com restrições" ou "sem restrições", dependendo da existência ou não de restrições;
- **Classificação baseada na natureza das variáveis de decisão (independentes):** Se as variáveis de decisão puderem ser tratadas como parâmetros, o problema é classificado como estático ou paramétrico. Se, por outro lado, as variáveis de decisão forem representadas por funções, o problema classifica-se como um problema de otimização de trajectória;
- **Classificação baseada na natureza das equações envolvidas:** De acordo com esta classificação, o problema de otimização pode ser classificado como linear, não linear, geométrico ou quadrático. Um problema diz-se geométrico se a função objectivo for expressa como uma função polinomial; um problema quadrático é um problema de programação não linear com uma função objectivo quadrática e restrições lineares;
- **Classificação baseada nos valores permitidos para as variáveis independentes:** Dependendo dos valores admissíveis para as variáveis de decisão, o problema de otimização pode ser classificado como inteiro, real ou misto-inteiro " *mixed integer* ";
- **Classificação baseada na natureza determinística das variáveis independentes:** Os problemas podem ser classificados como probabilísticos ou determinísticos;
- **Classificação baseada na separabilidade das funções:** O problema pode ser classificado como separável ou não separável, dependendo de a função objectivo ou de as funções contemplando o conjunto de restrições poderem ser descritas como somatório de n funções;
- **Classificação baseada no número de funções objectivo:** O problema pode ser classificado como simples ou multi-objectivo;

5.4 Ótimo local e ótimo global

Conforme referido anteriormente, o ponto ótimo ou extremo é o ponto do espaço solução onde a função objectivo é máxima ou mínima, dependendo da busca. No entanto muitas funções apresentam múltiplos máximos ou mínimos locais. Se um ponto for um extremo (máximo ou mínimo) de uma determinada região diz-se que este é um ótimo local. Se este for um extremo em todo o domínio da função objectivo este denomina-se de extremo global.



b

Figura 5.1: Exemplo de uma função difícil de otimizar.

Particularizando para a busca do ponto que minimize uma determinada função custo, o mínimo local e o mínimo global podem ser definidos como:

Mínimo local

x^* é um ótimo local se existe $\varepsilon > 0$, tal que $g(x^*) \leq g(x)$ para todo o $x \in R^n$ tal que $\|x - x^*\| \leq \varepsilon$.

Mínimo global

x^* é um ótimo global se existe $g(x^*) \leq g(x)$ para todo $x \in R^n$.

A Figura 5.1 representa uma função com múltiplos máximos e mínimos locais. Para este tipo de funções é complicada a obtenção do máximo ou mínimo global, pois a generalidade dos algoritmos tendem a ficar "presos" em extremos locais. Esta questão está ainda em aberto e diversas metodologias têm sido desenvolvidas para diminuir o risco da otimização convergir para um ótimo local.

5.5 Métodos de otimização

Como referido anteriormente na secção 5.4, um dos problemas mais frequentes dos métodos de otimização é o problema dos múltiplos extremos locais. Por este motivo o método utilizado, assim como os valores iniciais atribuídos às variáveis independentes influenciam o resultado final do processo iterativo. Outro problema comum neste tipo de processos é que nem sempre a função objectivo é suave e contínua, inviabilizando assim a utilização de algoritmos baseados no método do gradiente. Este tipo de problemas sugere a utilização dos algoritmos do tipo evolucionários ou genéticos [Cam04].

Os métodos de otimização podem ser divididos em três famílias:

- Métodos baseados no gradiente;
- Algoritmos evolucionários;
- Redes neuronais.

Ao longo desta secção serão abordados apenas os métodos baseados no gradiente da função de custo. Este tipo de métodos é conhecido por ser excessivamente dependente dos parâmetros atribuídos inicialmente, em problemas não convexos. Tal deve-se à elevada probabilidade de ser obtida uma solução não global [Pol97].

Seguidamente apresentam-se algumas definições importantes para o desenvolvimento do trabalho.

Ponto crítico

Vector \bar{x} onde o gradiente de f se anula, ou seja, onde $\nabla f(\bar{x}) = 0$.

Gradiente

Quando uma função $f : R^n \rightarrow R$ é diferenciável, o gradiente é o vector das derivadas parciais de f num ponto x .

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix} \quad (5.1)$$

Hessiana

Quando a mesma função f é duas vezes diferenciável obtém-se a matriz Hessiana de f num ponto x que tem como elementos as derivadas dos elementos do vector gradiente:

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix} \quad (5.2)$$

5.5.1 Método de Newton para otimização sem restrições

Tendo em conta o conceito de mínimo local (ver secção 5.4), para encontrar um zero da função $g : R^n \rightarrow R^n$ utilizando o método de Newton recorre-se ao seguinte algoritmo iterativo.

1. inicialização de x^1
 2. enquanto $\|x^{k+1} - x^k\| > \varepsilon$

$$x^{k+1} = x^k - \nabla g(x^k)^{-1} g(x^k);$$

$$k = k + 1;$$
- fim

Considerando $g(x) = \nabla f(x)$ o algoritmo de Newton para a minimização de funções pode ser reescrito da seguinte forma:

1. inicialização de x^1
 2. enquanto $\|x^{k+1} - x^k\| > \varepsilon$

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k);$$

$$k = k + 1;$$
- fim

Teorema 5.1: Convergência local do método de Newton [DW79]

Seja $f : R^n \rightarrow R$ uma função duas vezes diferenciável numa vizinhança do ponto \bar{x} , onde a segunda derivada é igualmente contínua. Supondo que \bar{x} é um ponto crítico de f e $\nabla^2 f(x)$ é não singular, então para todo o ponto inicial x^1 suficientemente próximo de \bar{x} o método de Newton gera uma sequência que converge para \bar{x} .

O teorema acima referido não impõe que o ponto \bar{x} seja um mínimo, bastando apenas que este seja um ponto crítico para que o método de Newton, com um ponto inicial suficientemente próximo, convirja para este. Para contornar a situação de não convergência global do método de Newton, surgiram ainda outros métodos, como o método das regiões de confiança e os métodos quasi-Newton.

5.5.2 Métodos Quasi-Newton para otimização sem restrições

Os métodos quasi-Newton são uma classe de métodos de otimização inspiradas no método de Newton para minimização de uma função.

Direcção de busca

Seja $f : R^n \rightarrow R$ uma função diferenciável no ponto $x \in R^n$, então:

- Para toda a direcção de busca descendente d de f no ponto x decorre que $\nabla f(x)^T d \leq 0$.
- Se d satisfaz $\nabla f(x)^T d < 0$, então d é uma direcção de busca descendente de f em x .

O algoritmo correspondente aos métodos Quasi-Newton pode ser formalmente descrito da seguinte forma:

1. Inicialização de x^1
 2. Enquanto $\|x^{k+1} - x^k\| > \varepsilon$,
 - Escolher uma matriz $n \times n$ simétrica positiva definida Q_k para a iteração corrente
 - e adoptar como direcção de busca $d^k = -Q_k \nabla f(x^k)$;

 - Calcular o passo α_k segundo uma regra de busca linear,
 - $x^{k+1} = x^k + \alpha_k d^k$
 - $k = k + 1$
- fim

Para escolha da matriz Q_k é necessário que esta seja positiva definida para que se garanta que a direcção busca d^k se apresente descendente. Caso tal se verifique vem:

$$\nabla f(x^k)^T d^k = \nabla f(x^k)^T Q_k \nabla f(x^k) < 0 \quad (5.3)$$

pelo que $-Q_k \nabla f(x^k)$ será uma direcção de descendente.

O método de Newton utiliza como Q_k a inversa da matriz Hessiana, o que corresponde a um cálculo de difícil computação. A dificuldade do cálculo da matriz Hessiana pode ser aliviada através de um abordagem aproximada, recorrendo às derivadas de primeira ordem estimadas por diferenças finitas.

Teorema 5.2: Teorema de Dennis-Moré [DM73]

Seja $f : R^n \rightarrow R$ uma função duas vezes diferenciável numa vizinhança do ponto \bar{x} , onde a segunda derivada é também continua. Assuma-se que \bar{x} é um ponto crítico de f , $\nabla^2 f(\bar{x})$ é não singular e $\{\bar{x}\}$ uma sucessão iterativa convergindo para \bar{x} , gerada pelo método de quasi-Newton com métrica variável, busca linear de Armijo com passo igual a um e factor de diminuição do passo $\theta \in [0; 0.5]$. Então as seguintes proposições são equivalentes:

- A taxa de convergência de $\{x^k\}$ para \bar{x} é superlinear, $\alpha_k = 1$, para todo o k suficientemente grande e $|Q_k \nabla f(x^k)| < M |\nabla f(x^k)|, \forall k = 1, 2, \dots$, onde $M > 0$ é uma constante que não depende k ;

- Verifica-se que $(Q_k - (\nabla^2 f(x^k))^{-1})\nabla f(x^k) = o(|\nabla f(x^k)|)$, onde o descreve o comportamento limite de uma função.

Pode-se concluir a partir deste teorema que uma condição suficiente para que se obtenha convergência superlinear é que a matriz Q_k se aproxime da inversa da matriz Hessiana, isto é, $(\nabla^2 f(x^k))^{-1}$. O método de Newton anteriormente referido satisfaz a condição, pois utiliza a própria inversa da matriz Hessiana.

Historicamente, o primeiro método do tipo quasi-Newton conhecido é o método quasi-Newton DFP (Davidon, Fletcher & Powell). Para este método o cálculo da matriz Q_k é feito iterativamente do seguinte modo:

1. Inicialização da matriz Q_1 como uma matriz positiva definida arbitrária, por exemplo a matriz identidade.
2. $x^{k+1} = x^k - \alpha_k Q_k \nabla f(x^k)$, sendo α dado por um método de busca linear.
3. $s^k = x^{k+1} - x^k$ e $r^k = \nabla f(x^{k+1}) - \nabla f(x^k)$

4.

$$Q_{k+1} = Q_k + \frac{r^k (r^k)^T}{(r^k)^T s^k} + \frac{(Q_k s^k)(Q_k s^k)^T}{(s^k)^T Q_k s^k}$$

Outro método importante, e que é actualmente considerado o mais eficiente dos métodos quasi-Newton, é o método BFGS (Broyden, Fletcher, Goldfarb & Shanno) [IM05]. Este método é semelhante ao DFP, diferindo na actualização da matriz Q_k que é feita iterativamente de acordo com a seguinte expressão:

$$Q_{k+1} = Q_k + \frac{(r^k - Q_k s^k)(r^k)^T}{(r^k)^T s^k} + \frac{(r^k - Q_k s^k)^T s^k r^k (r^k)^T}{((r^k)^T s^k)^2} \quad (5.4)$$

5.6 Optimização sujeita a restrições

Em muitos problemas de optimização existem regiões do domínio da função de custo que não são admissíveis, ou seja, pontos que geram soluções que violam uma ou mais condições de exequibilidade funcional. Neste contexto, estas regiões deverão ser formalmente excluídas do processo de optimização, através da imposição de restrições efectivas representadas por equações de igualdade ou desigualdade.

As técnicas de optimização desenvolvidas para a resolução de problemas de optimização não linear com restrições são divididas em duas categorias: métodos indirectos e métodos directos. Os métodos indirectos reformulam os problemas com restrições, transformando-os em problemas sem restrições e resolvem-nos através de uma sequência de minimização de problemas sem restrições, enquanto que nos métodos directos as restrições são

tratadas de forma explícita.

5.6.1 Extensão do método de quasi-nwton para optimização com restrições

Considere-se o seguinte problema de optimização, em que h e g são funções que representam respectivamente as restrições de desigualdade e igualdade:

$$\min_x f(x) \text{ sujeito a } g(x) \leq 0, \quad h(x) = 0 \quad (5.5)$$

As condições de KKT (Kharash-Kuhn-Tucker) permitem verificar se uma solução obtida é um mínimo local, ou seja, se verifica a condição de optimalidade. Estas condições são genericamente semelhantes à condição de o gradiente de uma função ser nulo num extremo de uma função, mas modificadas de modo a ter em conta as restrições impostas. As condições de KKT recorrem ao seguinte Lagrangiano:

$$L(x, \lambda) = f(x) + \sum \lambda_{g,i} g_i(x) + \sum \lambda_{h,i} h_i(x) \quad (5.6)$$

onde o vector λ , representa a concatenação de λ_g e λ_h o vector de multiplicadores de Lagrange. As condições de KKT são dadas por:

$$\lambda_x L(x, \lambda) = 0 \quad (5.7)$$

$$\lambda_{g,i} g_i(x) = 0 \quad \forall i \quad (5.8)$$

$$\begin{aligned} g(x) &\leq 0, \\ h(x) &= 0, \\ \lambda_{g,i} &\geq 0. \end{aligned} \quad (5.9)$$

A ideia de utilizar um método quasi-Newton melhorado para obter a Hessiana do Lagrangiano foi inicialmente proposta por Murray [Mur69]. A abordagem mais utilizada dentro dos métodos quasi-Newton melhorados é denominado de aproximação directa e modela a Hessiana do lagrangiano $\nabla_{xx}^2 L(x, \lambda)$ através de uma matriz Q_k que combina as estimativas dos multiplicadores de Lagrange e as estimativas da segunda derivada da função objectivo e restrições.

A adaptação do método quasi-Newton sem restrições a este problema é efectuada definindo a diferença do gradiente como:

$$r^k = \nabla_x L(x^{k+1}, \lambda) - \nabla_x L(x^k, \lambda) \quad (5.10)$$

Como tal, o método quasi-Newton BFGS pode ser adaptado ao problema vigente substituindo a nova definição de r^k em (5.4) e utilizando a mesma abordagem iterativa do

método quasi-Newton BFGS tendo em vista o cálculo da matriz Q_k .

6

Redes de sensores sem fios

As Redes de Sensores Sem Fios (RSSF) são compostas por pequenos sensores com capacidade de processamento e de computação limitada, apresentando baixo custo, quando comparados com os sensores tradicionais. Estes pequenos dispositivos permitem a "percepção" do ambiente que os rodeia e, com base em algum processo de decisão local, podem transmitir a informação recolhida [YMG08].

Estas redes diferem das redes de computadores tradicionais em diversos aspectos: geralmente possuem um elevado número de nós distribuídos espacialmente, têm restrições de energia e englobam imperativamente mecanismos para auto configuração e adaptação, de modo a acomodar falhas de comunicação e supressão de nós.

6.1 Sensores sem fios

Os sensores sem fios são constituídos por cinco unidades básicas: fonte de alimentação, memória, unidade de processamento, módulo de aquisição e rádio [ASSC02]. Seguidamente apresentam-se algumas características inerentes aos sensores sem fios.

- **Eficiência energética**- Tal como foi referido anteriormente, este é um dos grandes desafios relativos à implementação de redes de sensores sem fios. Os sensores sem fios possuem uma autonomia energética limitada, o que determina o seu tempo de vida útil. Dado que as redes sem fios são compostas por inúmeros sensores, recarregá-los é uma tarefa praticamente inviável, pelo que cada sensor deverá ser o mais eficiente possível.

- **Baixo custo-** O custo individual de cada sensor afecta o custo total da rede, pelo que o seu custo deverá ser relativamente baixo.
- **Comunicação sem fios-** Em diversas aplicações a instalação de cabos poderá ser difícil, ou até mesmo impossível. Portanto os sensores deverão ter um meio de comunicação sem fios.
- **Fácil de programar-** A programação deverá ser fácil devido à necessidade de re-programação e desenvolvimento de novas aplicações e protocolos de comunicação.

Os sensores sem fios, ou *motes*, utilizados neste trabalho são do tipo TelosB (ver Figura 6.1), alimentados por baterias AA. Construtivamente, têm a capacidade de monitorizar vários tipos de variáveis, tais como, humidade relativa, temperatura ambiente ou luminosidade. Contudo os dispositivos considerados no presente trabalho não integravam quaisquer transdutores, sendo as variáveis disponibilizadas por transdutores externos adquiridas através dos conversores analógico-digital integrados no próprio *mote*.

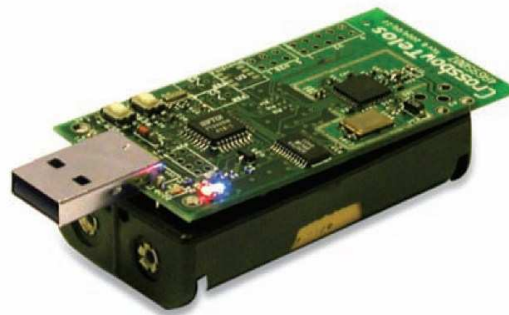


Figura 6.1: Mote TelosB.

Fisicamente são compostos por um microprocessador Texas Instruments MSP430 de 16 bits com 10 Kb de RAM, placa I/O, conversores analógico-digital e digital-analógico de 12 bits, rádio integrado, temporizadores e conexão USB para programação e/ou para alimentação eléctrica. O rádio compatível com a norma 802.15.4 opera entre a banda dos 2.4 a 2.4835 GHz e disponibiliza uma taxa de transferência de 250 Kbps. O alcance está compreendido entre 75 a 100 metro no exterior e entre 20 a 30 metro no interior de edifícios, embora alguns autores indiquem que o alcance poderá em determinadas condições chegar até 125 metro [BPC⁺07].

Sendo a capacidade de armazenamento um ponto fulcral, os *motes* incorporam, de origem, uma memória flash de 1MB, que para além de permitir uma maior capacidade de armazenamento possibilita um incremento da capacidade computacional do microprocessador. De referir que a capacidade computacional do microprocessador é talvez uma das suas maiores limitações, patente em aplicações computacionalmente exigentes.

6.2 Redes de comunicação

A questão básica no que se refere às redes de comunicação é a transmissão de mensagens satisfazendo determinadas medidas de desempenho, tais como a Qualidade e Quantidade de Serviço (*Quantity of Service and Quality of Service, QoS*). Tais métricas podem ser especificadas em termos de atraso de mensagens, tempos de latência, perda de pacotes, potência de transmissão, economia de transmissão, entre outras, as quais dependem em grande medida do ambiente de instalação [Lew04].

Uma rede de comunicação é composta por nós, cada um possuindo capacidade de transmissão e de recepção de mensagens através de canais de comunicação, *wireless* ou cablados. A escolha de como os nós serão ligados numa rede poderá ser um assunto crítico, já que uma escolha inadequada da topologia poderá conduzir a custos globais injustificados, assim como a um mau aproveitamento dos recursos da rede. As topologias tipicamente adoptadas para as redes de sensores sem fios são em árvore, estrela ou *mesh*.

As redes sem fios são uma área em franca expansão. Incluídas neste paradigma encontram-se as redes de sensores sem fios que são compostas por diversos dispositivos individuais designados por nós sensores colaborando entre si, tendo em vista recolher e encaminhar informação de um determinado ambiente ou sistema até uma estação base.

Este tipo de redes possui diversas vantagens, relativamente às redes de sensores com fios, destacando-se os custos de implementação, tamanho dos dispositivos, flexibilidade, inteligência atribuída e redução de erros. Também em termos de monitorização de ambientes de difícil acesso, estas redes apresentam as vantagens relativamente a outras, pois têm a possibilidade de serem instaladas em qualquer lugar. De uma forma resumida, pode-se dizer que as RSSFs são extremamente apelativas porque combinam as vantagens de uma comunicação *wireless* com capacidades de computação e detecção de determinadas grandezas físicas (temperatura, luz, humidade, CO₂, vibração, pressão, som, radiação, etc.) [LMPLdB05].

6.2.1 Topologia de rede em árvore

Esta topologia é baseada numa estrutura hierárquica de várias redes e sub-redes. Existem um ou mais agregados de nós que se ligam a outros agregados de forma hierárquica recorrendo a concentradores. Esta topologia (Figura 6.2) facilita a manutenção da rede e permite, em caso de avaria, detectar a origem com maior facilidade.

6.2.2 Topologia de rede em estrela

Na topologia em estrela (Figura 6.3) todos os nós da rede estão ligados a um único nó *hub*. O nó *hub* requer uma elevada capacidade de retenção de mensagens, encaminhamento e

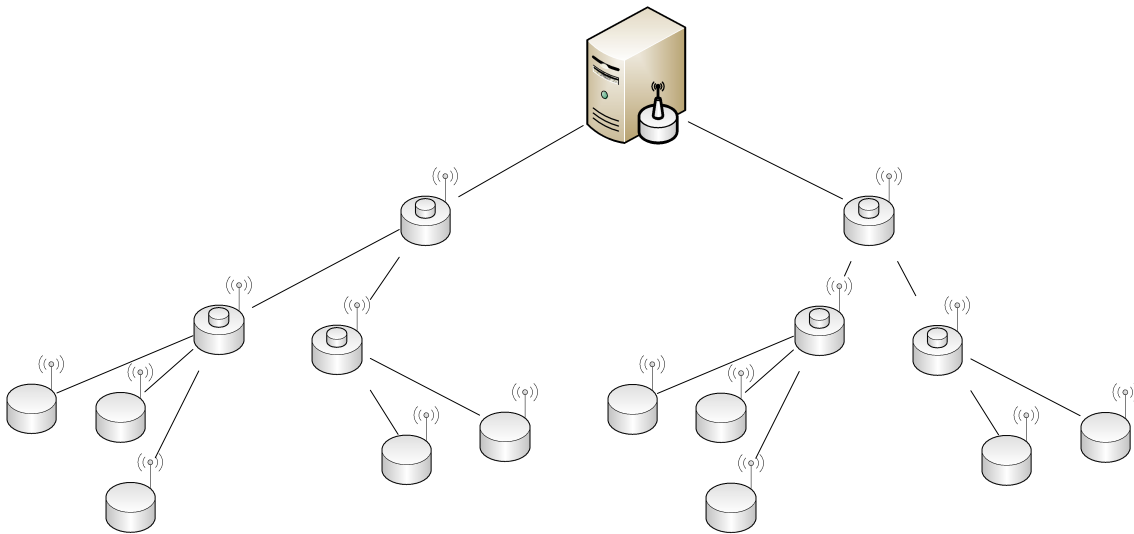


Figura 6.2: Topologia de rede em árvore.

tomada de decisão quando comparado com restantes nós. Se uma ligação for quebrada, apenas afecta um nó. No entanto, se o *hub* ficar incapacitado, toda a rede ficará comprometida [Lew04].

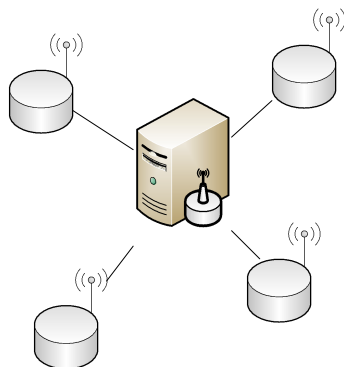
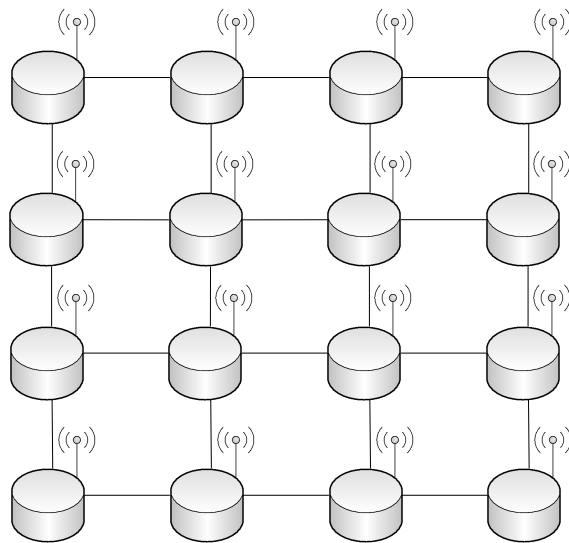


Figura 6.3: Topologia de rede em estrela.

6.2.3 Topologia de rede *mesh*

As topologias de rede *mesh* (Figura 6.4) são essencialmente redes distribuídas e normalmente apenas é efectuada a ligação aos nós vizinhos mais próximos, sendo os respectivos nós geralmente idênticos. Dado que existem múltiplos caminhos de encaminhamento de mensagens, estas redes são mais robustas a falhas de nós ou de ligações entre nós [Lew04].

Numa RRSF define-se *source* como uma entidade, tipicamente um sensor, pertencente à

Figura 6.4: Topologia de rede *mesh*.

rede, que pode fornecer informação, enquanto que um *sink* é uma entidade onde a informação é requerida. O *sink* pode ser mais um sensor pertencente à rede, ou uma entidade externa, como por exemplo, um PDA ou um gateway, que liga a rede de sensores sem fios, a uma outra rede, como por exemplo a Internet, em que os pedidos para a rede estão associados a sensores que se encontram fisicamente distantes [KW05].

6.3 Áreas de aplicação das redes de sensores sem fios

Devido às potencialidades inerentes aos sensores sem fios, a utilização de redes baseadas nesta tecnologia tem vindo a aumentar consideravelmente. A sua utilização não se tem limitado a um tipo de indústria em particular, encontrando-se um vastíssimo conjunto de áreas de aplicação [ASSC02]. Na Figura 6.5 [YMG08] apresenta-se esquematicamente as potencialidades aplicacionais deste paradigma.

6.4 Factores de influência no projecto de RSSFs

Embora existam diversas vantagens subjacentes à utilização de RSSFs subsistem ainda vários problemas técnicos a superar. Um desafio de Projecto e Desenvolvimento consiste em desenvolver mecanismos de comunicação de baixo custo energético e de processamento, assim como protocolos e topologias auto-organizáveis ou dinâmicas. O baixo consumo de energia consiste num factor importante para garantir longos períodos de operação em sistemas alimentados por baterias. Esta questão relacionada com a eficiência energética subjacente é geralmente abordada de três formas:

- Operação em baixo ciclo de processamento;

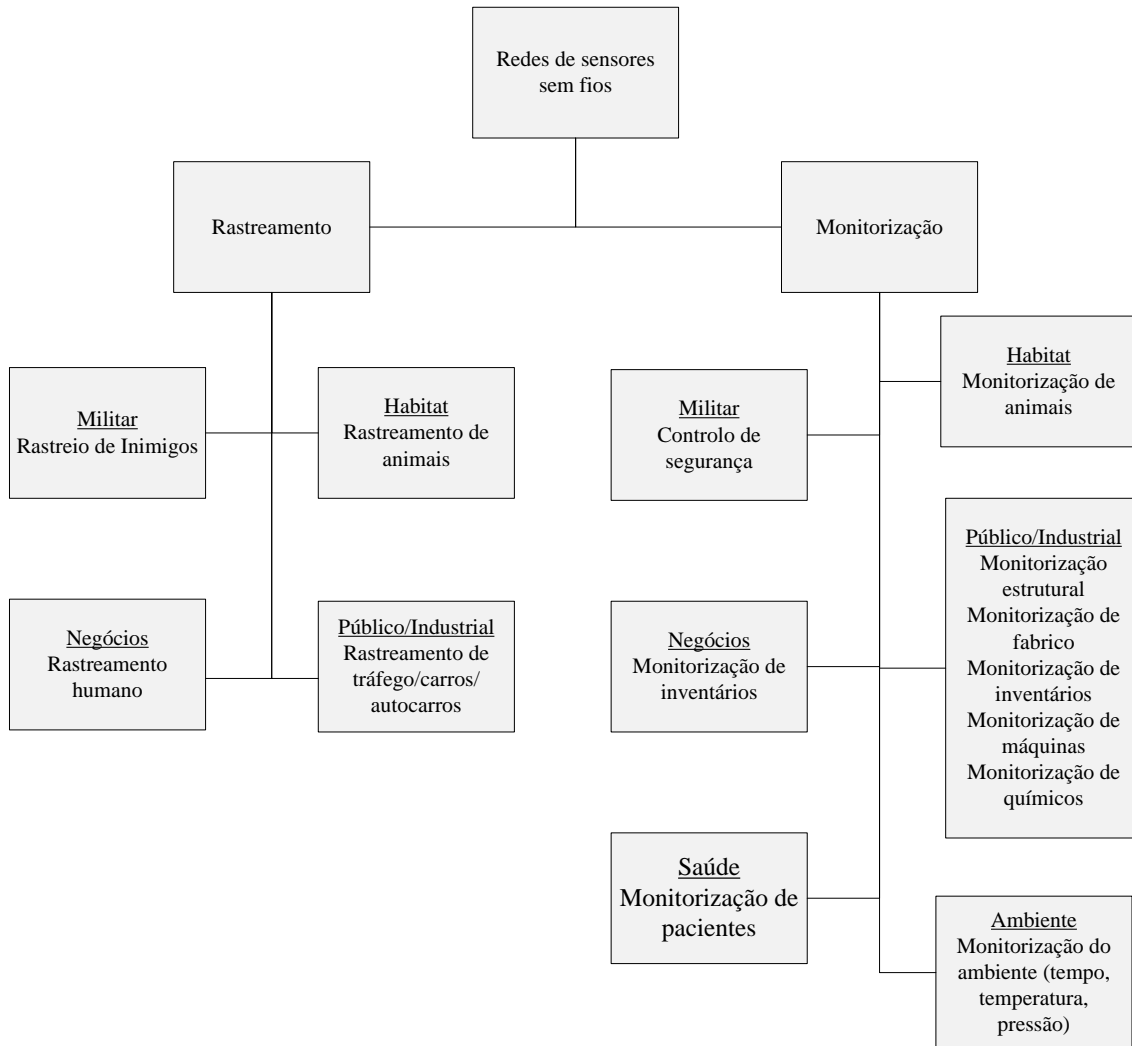


Figura 6.5: Aplicações das redes de sensores sem fios.

- Processamento local de modo a reduzir o volume de dados e o tempo de transmissão;
- Redes em múltiplos saltos de modo a reduzir a necessidade de longas distâncias de transmissão (cada nó pode agir como repetidor).

Os sensores sem fios podem ser dispersos na área geográfica de interesse de duas formas distintas: um a um, ou dispersos em massa (atirados a partir de um avião, por exemplo). Normalmente as considerações a ter por parte de quem projecta uma rede de sensores sem fios são as seguintes [ASSC02]:

- Tolerância a falhas;
- Escalabilidade;
- Custos de produção;

- Ambiente de operação;
- Topologia da rede de sensores sem fios;
- Restrições de *hardware*;
- Meio de transmissão;
- Consumo de energia.

6.4.1 Tolerância a falhas

Neste paradigma de comunicação os nós sensores podem falhar ou ficar bloqueados devido a falhas energéticas, danos físicos ou interferências ambientais, devendo a falha de alguns sensores não afectar a tarefa global da rede, ou seja, a rede deverá apresentar características de robustez face a um conjunto de falhas. Neste contexto, a tolerância a falhas representa a capacidade de manter interruptamente as funcionalidades da rede sem fios, mesmo perante a ocorrência de falhas em alguns nós[ASSC02].

6.4.2 Escalabilidade

A ordem de grandeza do número de nós de uma RSSF pode variar entre as dezenas e centenas, podendo mesmo atingir os milhares em algumas aplicações específicas. As topologias adoptadas devem ser suficientemente flexíveis para serem capazes de suportar o efeito de escalabilidade. Esta flexibilidade está directamente associada à tolerância a falhas. A grande quantidade de nós defeituosos de uma aplicação poderá provocar a mudança das características de escalabilidade de uma região das RSSFs. Esta mudança deverá ser automaticamente acomodada pela rede [ASSC02].

6.4.3 Custos de produção

Dado que uma rede de sensores sem fios é tipicamente composta por um elevado número de nós, o custo individual de cada sensor é bastante importante para justificar o custo total da rede. Se o custo for mais elevado comparativamente ao desenvolvimento de uma rede tradicional, então a utilização da rede de sensores sem fios não é deste ponto de vista apelativa [ASSC02].

6.5 Tipos de Comunicação

Existem dois tipos de comunicação sem fios possíveis de ser implementados em RSSFs, a comunicação óptica que opera na banda dos infra vermelhos e a Rádio Frequência (RF) que funciona na banda de microondas. [RIG99] No âmbito desta dissertação apenas será aprofundado o tipo de comunicação recorrendo a RF.

6.5.1 Rádio Frequência

Este tipo de comunicação baseia-se na transmissão de ondas electromagnéticas na banda de frequências de 2.4 e 5 GHz. Na maioria das transmissões de dados por RF é utilizada modulação de amplitude, frequência ou fase. A modulação em amplitude torna mais simples a codificação e a decodificação, embora seja mais susceptível a interferências, devido ao facto dos dados serem codificados na potência do sinal e do ruído externos poder ser adicionado ao sinal. A modulação em frequência é menos sujeita a ruídos, já que todos os dados são transmitidos no mesmo nível de potência.

Devido às limitações inerentes às comunicações rádio, nomeadamente no que se refere à potência necessária para transmitir um sinal para um receptor a longa distância, a comunicação directa entre o emissor e o receptor nem sempre é exequível. Para ultrapassar esta limitação a utilização de outros sensores tornou-se necessária para encaminhamento das mensagens até ao destino. Esta arquitectura *multi-hop* torna assim possível, através do reencaminhamento de pacotes, um menor consumo de energia, assim como uma maior capacidade de contornar obstáculos naturais que impeçam a comunicação directa entre o transmissor e o receptor.

Como referido em [KW05], a energia requerida para uma comunicação directa a uma distancia d é cd^α (em que c é uma constante e α o coeficiente de perda de energia) enquanto que utilizando um sensor como reencaminhador, à distância $d/2$ reduz essa energia para $2c(d/2)^\alpha$.

A optimização dos caminhos mais próximos para um dado *sink* ou para uma *source* pode ser calculado mediante vários critérios de optimização [DP10]:

- Caminho com menor número de saltos - o caminho escolhido é o que apresenta um menor número de saltos, i.e., o que necessita de um menor número de reencaminhamentos para chegar ao destino;
- Energia;
 - Menor energia consumida por transmissão de pacote;
 - Tempo máximo de partição da rede;
 - Minimização da variância dos níveis de potência associados aos nós;
 - Máxima (em média) capacidade energética dos nós;
 - Máximo do mínimo da capacidade energética - Neste caso, para todos os caminhos, é calculado o nó com menor capacidade energética, deste conjunto, sendo escolhido o caminho que contém o nó correspondente ao majorante da capacidade energética. Este esquema é utilizado para prevenir que o caminho escolhido se torne impossível, devido ao facto de um nó ficar sem energia.

- Qualidade de Serviço;
- Robustez - qualidade e estabilidade do *link*.

6.6 Sistemas Operativos

Um dos primeiros sistemas operativos disponíveis para o este tipo de sensores sem fios foi o sistema operativo TinyOS [HHKK04]. Este é um sistema operativo livre, baseado no conceito de eventos e de "código aberto" que foi desenvolvido a partir de uma colaboração entre a Universidade da Califórnia e o centro de pesquisa da Intel. A linguagem de programação utilizada denomina-se NesC, representando uma extensão da linguagem de programação C baseada em eventos, de modo a estruturar os conceitos e o modelo de execução do TinyOS.

As aplicações construídas neste sistema operativo são interligadas estaticamente usando o conceito de componentes. Estes componentes contêm as semânticas relacionadas com cada funcionalidade disponível, sendo organizados hierarquicamente, desde componentes de baixo nível, próximos do *hardware*, a componentes de alto nível.

Quanto ao sistema operativo Contiki [DGV04], este foi desenvolvido por Adam Dunkels no Instituto de Ciências Computacionais da Suécia. Este sistema operativo é, à semelhança do TinyOS, um sistema livre e de código aberto baseado em eventos. A linguagem utilizada para programar neste sistema é a linguagem C.

Uma das suas principais características é a utilização das chamadas *protothreads* [DSVA06]. Estas apresentam-se como um tipo particular de *threads* mais simples, sem pilha, desenhadas para sistemas com severas restrições de memória.

Em sistemas em que o problema de alocação de memória é tão importante, o Contiki possui outra excelente característica, consubstanciada no facto de poder carregar e descarregar código dinamicamente, ao contrário do TinyOS, que se apresenta estaticamente interligado. Isto permite que sejam efectuadas remotamente alterações no código base de cada sensor. Tal, constitui uma mais valia, particularmente em redes que possuam um grande número de sensores.

Neste trabalho foi utilizado o sistema operativo Contiki, não só pelas características descritas, mas pelo facto desta dissertação se inserir no âmbito do projecto Europeu, o projecto GINSENG¹, onde a escolha do sistema operativo recaiu sobre o Contiki.

¹O projecto GINSENG terminou em Fevereiro de 2012, concluindo um programa de investigação bem sucedido no que trata o controlo e monitorização de redes de sensores sem fios. <http://www.ict-ginseng.eu/>

7

Arquitecturas Propostas

Apesar da grande potencialidade de aplicação dos controladores PID difusos em diversos contextos, obter um bom compromisso entre variáveis linguísticas, funções de pertinência, regras e factores de escala, representa ainda um grande desafio, dada a ausência de uma abordagem sistemática. A dificuldade de projecto do controlador difuso é acrescida quando o sistema a controlar possui múltiplas entradas e saídas (MIMO), em virtude de efeitos de acoplamento.

O presente capítulo tem por objectivo apresentar duas arquitecturas para sintonização automática dos ganhos de um controlador PID difuso, recorrendo à optimização de uma função de custo explicitada pelo projectista. Estas duas arquitecturas, para além de possibilitarem uma sintonização simples e automática dos factores de escala, ou ganhos, evitando metodologias de tentativa e erro baseadas em heurísticas mais ou menos quantitativas, contribuem para uma utilização mais eficiente e racional dos recursos disponíveis, nomeadamente, energia e uma maior longevidade no que se refere aos sistemas de controlo.

Para efeito de sintonização dos controladores foi utilizado o método de optimização quasi-Newton melhorado de modo a lidar com as restrições relativas à acção de controlo e aos valores dos ganhos do controlador.

7.1 PID difuso com otimização em diferido dos ganhos

Na arquitectura proposta nesta secção, a otimização dos ganhos do controlador PID difuso ocorrerá em diferido, utilizando para o efeito um modelo não linear de modo a simular o comportamento do sistema para os ganhos a testar. A escolha de um modelo não linear para descrição da dinâmica do sistema deve-se essencialmente ao facto deste tipo de modelos, em geral, permitir descrever convenientemente o comportamento dinâmico do sistema numa gama alargada de regimes de funcionamento. Outra característica que determinou a escolha deste tipo de modelos foi, geralmente, estes permitirem um bom ajuste com um número menor de parâmetros que os modelos lineares quando utilizados na modelação de sistemas não lineares

O mecanismo de otimização dos ganhos (ver Figura 7.1) consiste em, ciclicamente, testar o desempenho dos mesmos segundo um determinado critério de desempenho até que seja obtida uma solução óptima, ou a solução mais perto possível de acordo com o número máximo de iterações especificado.

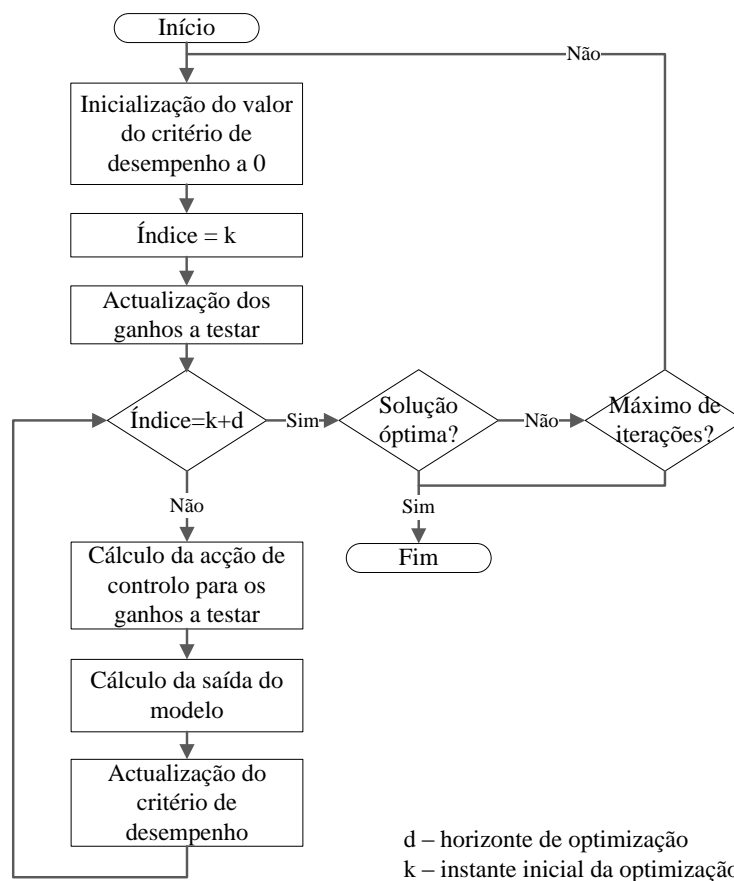


Figura 7.1: Algoritmo de optimização dos ganhos do controlador.

Ao utilizar esta arquitectura pretende-se optimizar os ganhos do controlador para que

seja obtida uma resposta óptima por parte do sistema para uma dada referência e critério de desempenho escolhidos pelo utilizador. Por esta razão, o instante inicial de optimização será o instante inicial de controlo (zero segundo) e o horizonte de optimização será equivalente ao horizonte de controlo. Dado que a optimização dos ganhos é feita previamente ao início do ciclo de controlo, o tempo de optimização não necessita de ser limitado, pelo que não é definido um número máximo de iterações. Na Tabela 7.1 encontram-se sumariadas as características intrínsecas à optimização em diferido dos ganhos do controlador.

Tabela 7.1: Características do processo de optimização em diferido.

Parâmetro	Valor
Tipo de modelo do sistema	Não linear
Instante inicial da optimização	0 s
Horizonte de optimização	Horizonte do ciclo de controlo
Número de iterações do ciclo de optimização	Até que seja encontrada a solução óptima

7.2 PID difuso com optimização em linha dos ganhos

Geralmente, para ganhos fixos do controlador PID difuso, não é possível ter simultaneamente em conta o regime transitório e estacionário, para além do facto de no caso do sistema ser variante a sua dinâmica altera-se ao longo do tempo. De modo a ter em atenção estas situações é apresentada uma arquitectura em que os ganhos do controlador são optimizados em-linha. Tal permitirá que o sistema de controlo se apresente mais robusto a variações de dinâmica, dado o carácter adaptativo do controlador.

7.2.1 Computação paralela para optimização dos ganhos

A optimização pode ser um processo demorado, sendo por vezes superior ao tempo de amostragem vigente. Este facto determinou que para resolução do problema de optimização não linear com restrições fosse adoptada uma abordagem alicerçada na computação distribuída. Nesta abordagem o ciclo de controlo não fica bloqueado aguardando a recepção dos ganhos provenientes do ciclo de optimização, caso estes não estejam disponíveis, mantendo-se os ganhos resultantes da optimização anterior.

A arquitectura SISO (*Single-Input-Single-Output*) do controlador com optimização em linha dos ganhos é composta por duas rotinas, a rotina principal (ver Figura 7.3), onde é feita a comunicação com o processo industrial e a rotina de optimização (ver Figura 7.4), responsável pelo cálculo dos ganhos óptimos do controlador. A arquitectura geral do controlador proposto pode ser representada pela Figura 7.2, em que foi atribuído um computador para cada uma das rotinas implementadas.

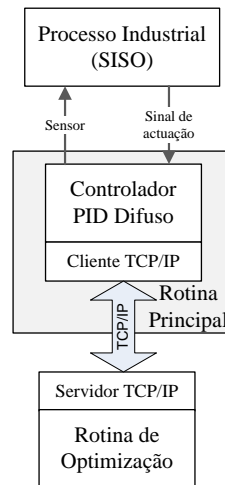


Figura 7.2: Arquitectura geral do controlador com optimização em linha dos ganhos.

A comunicação entre as duas rotinas é feita utilizando o protocolo TCP/IP (*Transmission Control Protocol/ Internet Protocol*), estando o servidor associado à rotina de optimização. Se a comunicação entre as duas rotinas (rotina principal e rotina de optimização) não se puder estabelecer os ganhos do controlador utilizados no ciclo de controlo serão os resultantes da optimização em diferido, não ocorrendo a sua actualização ao longo do ciclo de controlo. Caso a ligação entre as rotinas for perdida, os ganhos utilizados pela rotina principal serão os correspondentes à ultima optimização.

Características relativas à escolha do protocolo TPC/IP

A principal vantagem da comunicação TPC/IP que determinou a sua utilização na arquitectura proposta foi a garantia de que a informação enviada chega íntegra ao outro lado. Contrariamente, o protocolo UDP é uma escolha mais adequada para a transmissão de fluxos de dados em tempo real, especialmente aqueles que permitem perda ou corrompimento de parte do seu conteúdo, como é o exemplo de ficheiros de vídeo ou voz. Resumidamente, a utilização de UDP é adequada em aplicações sensíveis a atrasos na rede, mas pouco sensíveis a perda de pacotes.

Outras características que motivaram a escolha do protocolo TCP/IP foram a entrega ordenada dos pacotes enviados, a possibilidade de *full duplex*, ou seja, a transferência simultânea em ambas as direcções da ligação e, por fim, o facto de ser orientado à conexão, pois permite uma fácil detecção de quando a ligação foi perdida.

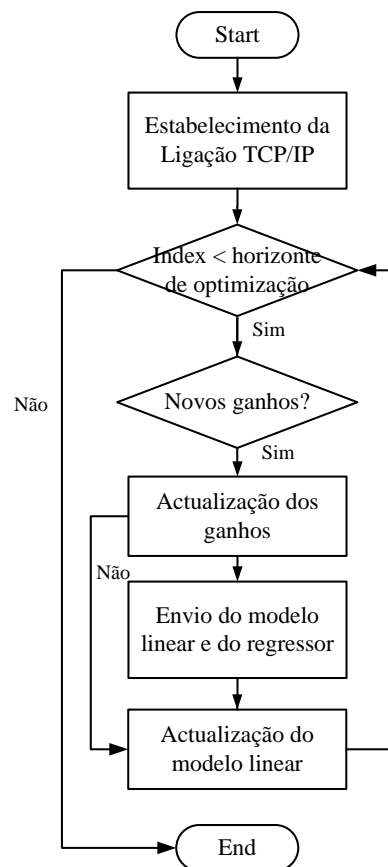


Figura 7.3: Rotina principal.

Rotina principal

Ao iniciar, a rotina principal obtém em diferido os ganhos do controlador PID difuso óptimo, tal como é descrito na secção anterior. Após este cálculo é criado um cliente TCP/IP que se tenta ligar à rotina de optimização.

Caso a comunicação entre as duas rotinas (rotina principal e rotina de optimização) não se puder realizar o controlador funcionará como descrito na secção anterior. Para cada ciclo de controlo é verificado se o modo de funcionamento é o de optimização em linha, se a ligação entre as rotinas ainda é possível e se estão disponíveis novos ganhos para o controlador. Se todos estes requisitos forem cumpridos, os ganhos do controlador são actualizados e é enviado para a rotina de optimização o regressor e o modelo linear actualizado do processo. Independentemente de as condições anteriores serem cumpridas, a cada ciclo de controlo é adquirida a informação dos sensores do processo industrial, calculada e enviada a acção de controlo e actualizado o modelo linear do sistema.

O carácter adaptativo do controlador é conseguido a partir da actualização em linha do modelo linear, sendo este utilizado como simulador do comportamento do sistema no processo de optimização dos factores de escala.

Rotina de otimização

Na rotina de otimização é onde são calculados os ganhos óptimos do controlador. Ao iniciar é criado um servidor TCP/IP que se mantém à espera que um cliente TCP/IP se ligue. Após ser estabelecida a ligação são recebidos o sinal de referência e os ganhos resultantes da otimização em diferido provenientes da rotina principal.

Por cada ciclo de otimização é recebido o regressor e o modelo ARX actualizado do sistema que serão utilizados na rotina de otimização em linha.

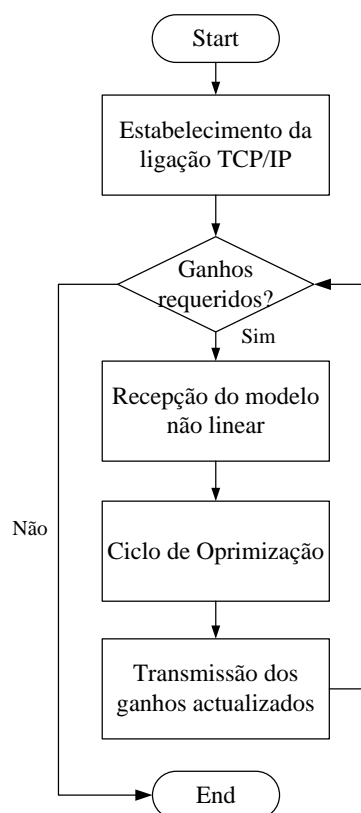


Figura 7.4: Rotina de otimização dos ganhos.

A otimização é feita de acordo com a metodologia presente no fluxograma da Figura 7.1. No caso da otimização em linha, ao contrário da otimização em diferido, foi definido um número máximo de iterações para encontrar a solução óptima, sendo o instante inicial do ciclo de optimização o instante de controlo e o horizonte de optimização limitado (ver Figura 7.5). Na Tabela 7.2 encontram-se sumariadas as características intrínsecas à optimização em-linha dos ganhos do controlador.

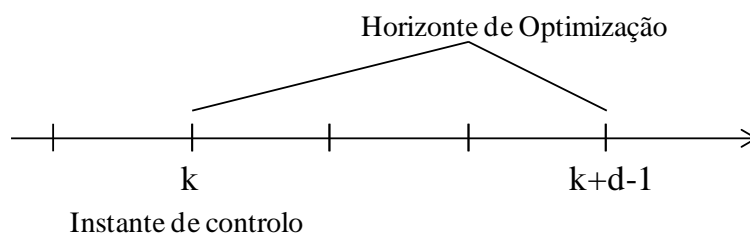


Figura 7.5: Horizonte de otimização.

Tabela 7.2: Características do processo de otimização em em-linha.

Parâmetro	Valor
Tipo de modelo do sistema	Linear
Instante inicial da otimização	Instante de controlo
Horizonte de otimização	Limitado
Número de iterações do ciclo de otimização	Limitado

7.2.2 Generalização do controlador a um sistema MIMO

Os processos centralizados caracterizam-se por ter a informação sobre todo o sistema disponível e os cálculos efectuados sobre essa informação serem operados centralmente, ou seja, têm lugar em apenas uma localização [SVAS78]. Na abordagem adoptada em sistemas MIMO prevê-se, em vez de obter um modelo que representasse o sistema global, encontrar n modelos SISO de acordo com o número de saídas a controlar, e projectar n controladores semelhantes ao descrito anteriormente. Como tal, em vez de se utilizar um modelo não linear para caracterizar o processo global, recorreu-se a modelos lineares para descrever a dinâmica destes subsistemas SISO.

Para superar o facto de o processo industrial ser não linear e de não ser modelada a interacção entre as dinâmicas modeladas, cada um dos n modelos lineares será actualizado ao longo do horizonte de controlo. Com uma abordagem descentralizada a computação associada ao processo de otimização é reduzida, visto que o problema de otimização pode ser partido em n problemas de otimização de menor dimensão. Algumas das principais características associadas à utilização de abordagens centralizadas e descentralizado encontram-se sumariadas na Tabela 7.3 [AB00].

Para um sistema MIMO com n saídas a arquitectura a utilizar será a da Figura 7.6. Esta arquitectura é composta por tantas rotinas de otimização quanto o número de saídas a controlar, tendo cada uma das rotinas a função de otimizar os ganhos do controlador PID cliente.

As rotinas de otimização serão iguais às descritas para o caso do sistema SISO. Quanto à

Tabela 7.3: Características dos paradigmas centralizado e descentralizado.

	Centralizados	Descentralizados
Unidades de Processamento (UP) são normalmente	Caras	Baratas
A informação das UP é	Global	Local
A aquisição de informação é normalmente	Cara	Barata
Os algoritmos de processamento das UP são normalmente	Complexos	Simple
Geralmente as UP conseguem processar a informação	Devagar	Rapidamente
Os "sistemas" são robustos a falhas das UP	Não	Sim
Os sistemas podem ser apanhados em situações patológicas	Não	Sim
Os sistemas conseguem encontrar um máximo global	Sim	Não
As UP normalmente operam	Base programada	Em tempo real

rotina principal, esta terá tantos clientes TCP/IP quanto o número de controladores PID difusos para os quais se pretende a optimização dos ganhos.

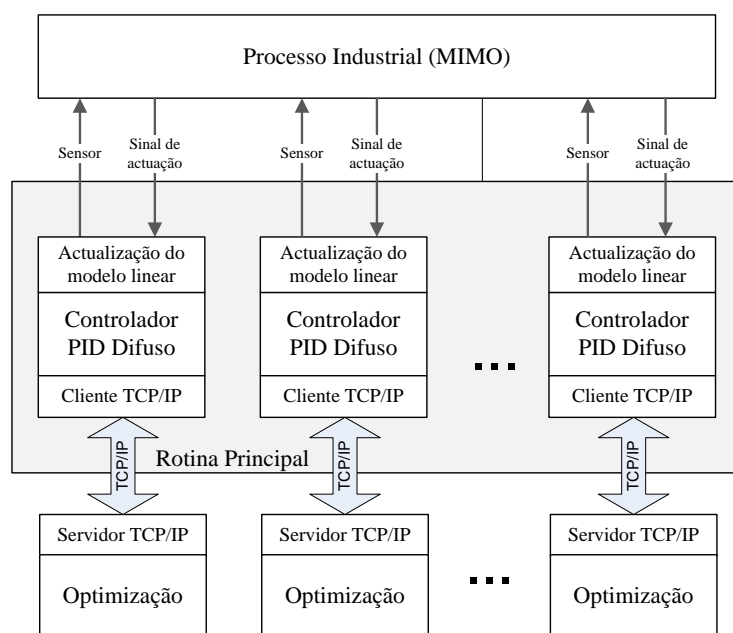


Figura 7.6: Esquema simplificado da rotina principal para um sistema MIMO.

Recorrendo à arquitectura generalizada para o controlo de sistemas MIMO, é possível acomodar os efeitos de acoplamento físico entre dinâmicas recorrendo à actualização recursiva dos modelos lineares que descrevem cada um dos sistemas a controlar.

8

Casos de Estudo

Nesta secção apresentam-se os resultados da aplicação do algoritmo de optimização dos factores de escala do controlador PID difuso a um processo laboratorial MIMO representado pelo sistema de três tanques, AMIRA[®] DTS 200. Os resultados serão obtidos, em primeiro lugar, considerando o servidor directamente acoplado à placa de aquisição com o processo, ou seja, sem recurso a comunicação sem fios. Por último, apresentam-se os resultados obtidos com recurso a uma rede de sensores sem fios, de modo a analisar a robustez perante tempos de latência. Para que exista uma base de referência, o desempenho das arquitecturas dimensionadas será comparado com uma metodologia de sintonização baseada na equivalência dos controladores PID convencionais com os controladores difusos.

8.1 Descrição do sistema

Como se referiu, o sistema de teste MIMO utilizado para a análise de desempenho dos controladores implementados consiste no sistema de três tanques presente na Figura 8.1. Este sistema compreende três vasos de plexiglas ligados entre si através de tubos de secção circular, munidos de válvulas de seccionamento esféricas. Os tanques T_1 e T_2 são alimentados directamente com água destilada, que se encontra armazenada no reservatório inferior, por meio de duas bombas que lhes estão associadas.

No tanque T_2 encontra-se localizada a saída principal de líquido do sistema, conjuntamente com três outras saídas complementares previstas na base de cada um dos vasos e destinadas à introdução de eventuais perturbações, sob a forma de um caudal de fuga.



Figura 8.1: Sistema de três tanques.

Considerando o sistema MIMO apresentado, pretende-se efectuar o controlo dos níveis de água dos tanques T_1 e T_2 . Este controlo será realizado pela leitura dos sensores representados no esquemático da Figura 8.2 e actuando respectivamente nas bombas $u_1(t)$ e $u_2(t)$.

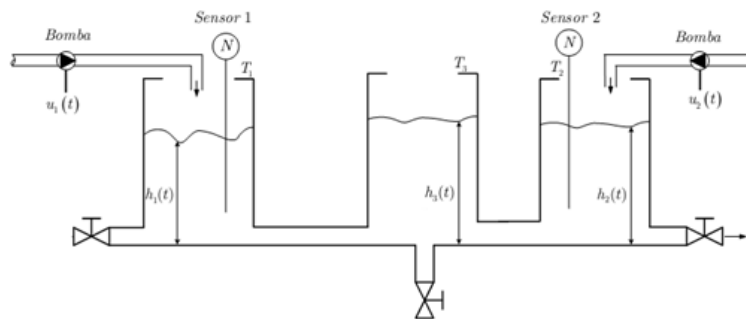


Figura 8.2: Esquemático do processo industrial AMIRA DTS200.

Em termos de ligação entre o servidor (*Personal Computer, PC*) e o processo foi utilizada uma placa de aquisição de dados *National Instruments USB-6009* (Figura 8.3). Esta é caracterizada por ter ligação USB, 8 entradas analógicas de 14 bits, 2 saídas analógicas de 10 bits, 12 ligações digitais I/O e um contador de eventos de 32 bits.

8.2 Topologia de redes sem fios adoptada

Na presente dissertação optou-se por utilizar um sensor sem fios ligado ao servidor (PC), onde se encontra a rotina principal descrita na secção anterior, e outros dois sensores sem fios acoplados ao processo. Este últimos nós são responsáveis pela leitura dos níveis de



Figura 8.3: *National Instrument USB-6009.*

líquido nos tanques T_1 e T_2 e envio das acções de controlo para as electrobombas b_1 e b_2 (Figura 8.4). Entre os sensores sem fios acoplados ao processo (nós folha) e o próprio processo foi necessária a incorporação de *hardware* adicional para condicionamento das gamas de sinais envolvidos. Os sensores sem fios funcionam na gama $[0; +2,5]$ V e o processo opera na gama dos $[-1; 1]$ V. O condicionamento foi feito recorrendo a conversores tensão-tensão.

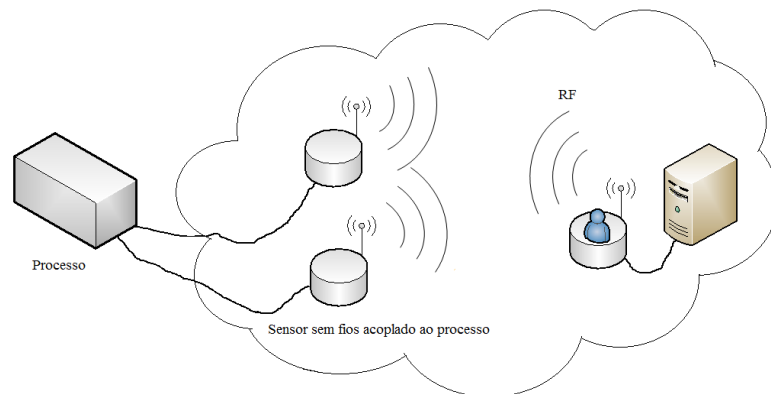


Figura 8.4: Arquitectura utilizada na rede de sensores sem fios (*single-hop*).

Em termos de implementação do algoritmo de controlo, as acções de controlo são calculadas recorrendo a código implementado em MATLAB e enviadas por TCP/IP para uma plataforma JAVA (implementada no âmbito do projecto GINSENG) que comunica com o nó *sink* através do *middleware*. O *sink* é responsável pelo envio das acções de controlo e recepção dos valores enviados pelos nós folha.

8.3 Projecto do controlador PID difuso

O projecto do controlador difuso foi feito de acordo com a arquitectura híbrida apresentada na Figura 3.7. Os universos de discurso das variáveis erro, e_k , e variação do erro, Δe_k foram definidos como $[-1.5, 1.5]$ e considerando sete conjuntos difusos, nomeadamente $\{NG, NM, NP, ZO, PP, PM, PG\}$.

Para a saída dos controladores difusos pertencentes à arquitectura híbrida utilizada, $\Delta u_{\tilde{P}I}$ e $u_{\tilde{P}D}$, o respectivo universo de discurso foi definido como $[-1.0, 1.0]$, e assumindo uma partição em sete conjuntos difusos semelhante ao erro e à variação do erro.

O tipo de mecanismo de inferência utilizado foi o de Mamdani (8.1), enquanto que a saída crespa do controlador é gerada pelo método de desfuzificação do centróide:

$$\mu_{e/\Delta e}(e, \Delta e) = \min(\mu_A(e), \mu_B(\Delta e)) \quad (8.1)$$

sendo μ o valor da função de pertença.

O sistema de regras correspondente a cada um dos controladores difusos (PI e PID) pertencentes à arquitectura utilizada compreende 49 regras. O formato das regras utilizado é o formato típico [LG96].

As funções de pertença do erro, \tilde{e} , e variação do erro, $\Delta\tilde{e}$, encontram-se representadas na Figura 8.5. Relativamente às funções de pertença correspondentes à acção de controlo proporcionada pelo controlador PD difuso, \tilde{u}_{PD} , e à variação da acção de controlo produzida pelo controlador PI difuso, \tilde{u}_{PI} , estas encontram-se representadas na Figura 8.6.

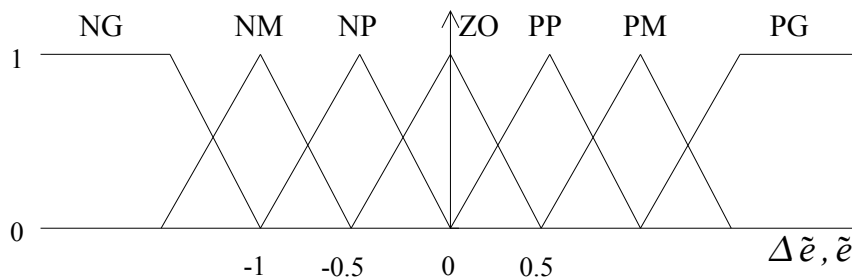


Figura 8.5: Funções de pertença de \tilde{e} e $\Delta\tilde{e}$.

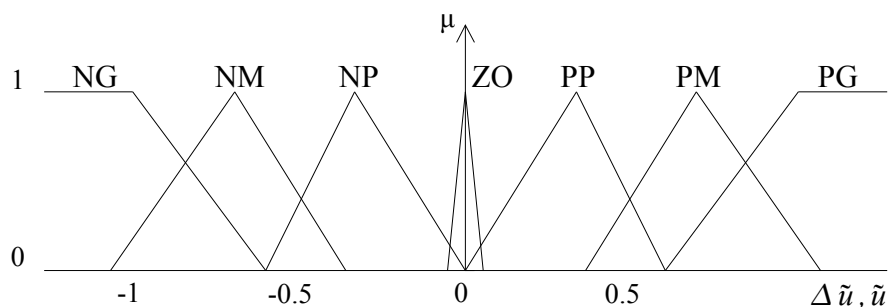


Figura 8.6: Funções de pertença de $\Delta\tilde{u}_{PI}$ e \tilde{u}_{PD} .

Em forma de resumo, o controlador PID difuso híbrido utilizado apresenta as seguintes características, no que trata ao controlador PI e PD que o constituem:

- Número de termos linguísticos: 7;
- Funções de pertinência: Triangulares;
- Bases de regras: Genéricas (ver tabelas 3.4 e 3.3);
- Método de inferência: Mamdani;
- Método de defusificação: Centróide;

Para o projecto do controlador difuso implementado nesta dissertação foi utilizada a ferramenta do MATLAB *Fuzzy Logic Toolbox* [Doc12c].

8.4 PID difuso com sintonização de ganhos baseada em controladores convencionais

Os primeiros resultados experimentais coligidos no sistema de três tanques foram obtidos num contexto onde a sintonização dos ganhos foi feita baseando-se numa pseudo-equivalência entre controladores convencionais e controladores difusos [Piv02]. O controlo do sistema MIMO será realizado de forma distribuída, atribuindo-se um controlador SISO a cada um dos níveis de água a controlar.

8.4.1 PID difuso com sintonização de ganhos baseada em controladores convencionais sobre placas de aquisição

Neste conjunto de experiências utilizou-se uma placa de aquisição de dados USB, para a ligação entre o servidor (PC) e o sistema a controlar. As Figuras 8.7 e 8.8 mostram a resposta em anel fechado para esta abordagem de sintonização dos ganhos.

Como pode ser observado, a resposta em anel fechado, no que se refere ao erro, é bastante satisfatória. No entanto é preciso realçar que a acção de controlo é relativamente agressiva, oscilando entre o valor máximo e mínimo da placa de aquisição em quase todo o horizonte de controlo.

8.4.2 PID difuso com sintonização de ganhos baseada em controladores convencionais sobre redes de sensores sem fios

No conjunto das experiências efectuadas nesta sub-secção pretende-se aferir o desempenho do controlador projectado sobre redes de sensores e actuadores sem fios, tendo em vista avaliar o impacto da variabilidade dos tempos de latência sobre a robustez global do sistema de controlo. As Figuras 8.9 e 8.10 mostram a resposta em anel fechado para a

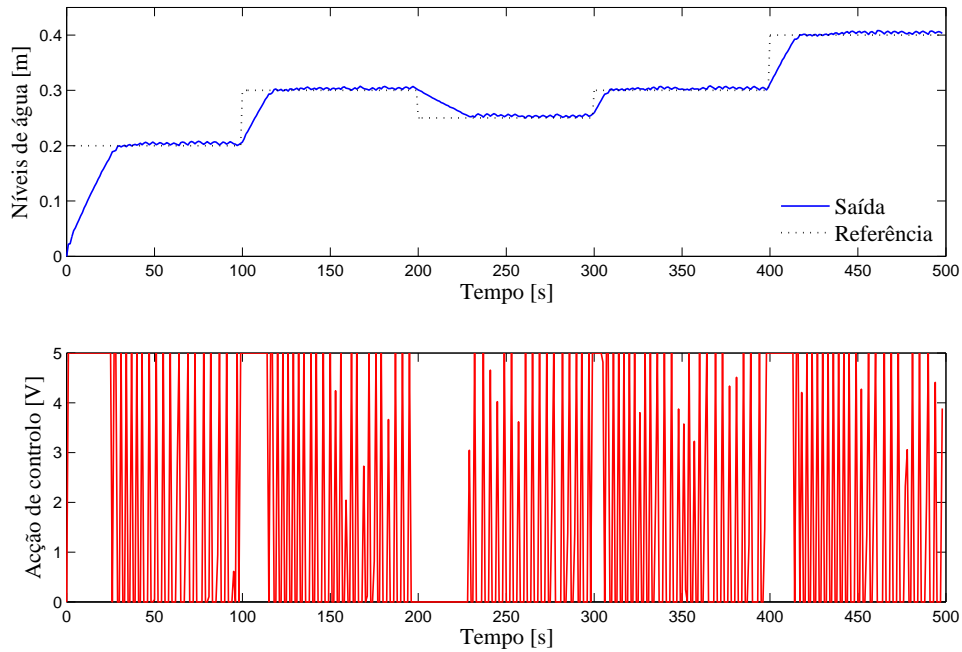


Figura 8.7: Tanque 1 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre placas de aquisição.

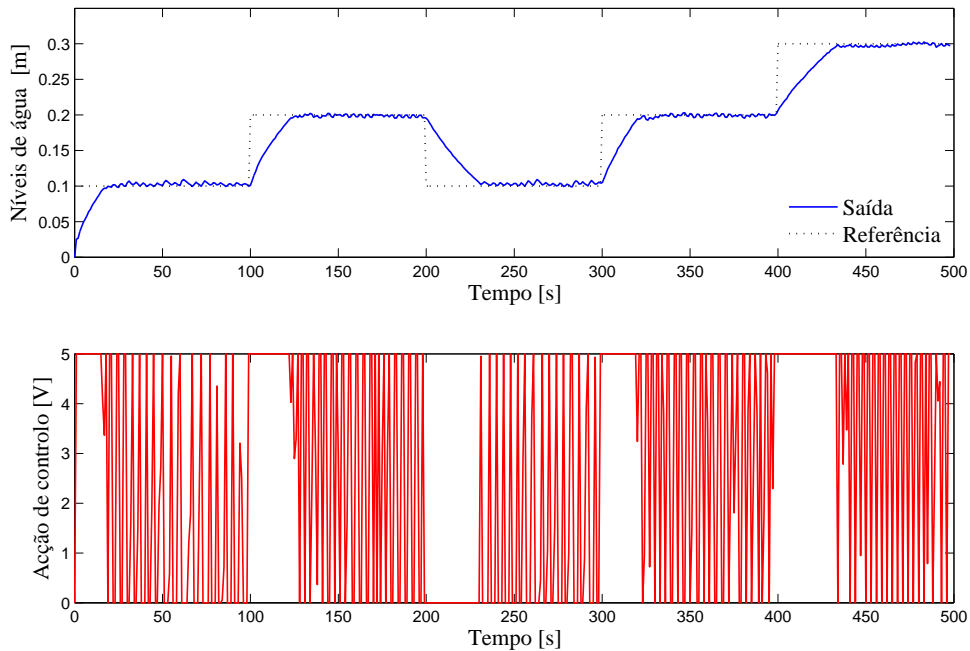


Figura 8.8: Tanque 2 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre placas de aquisição.

abordagem descrita sobre redes de sensores e actuadores sem fios.

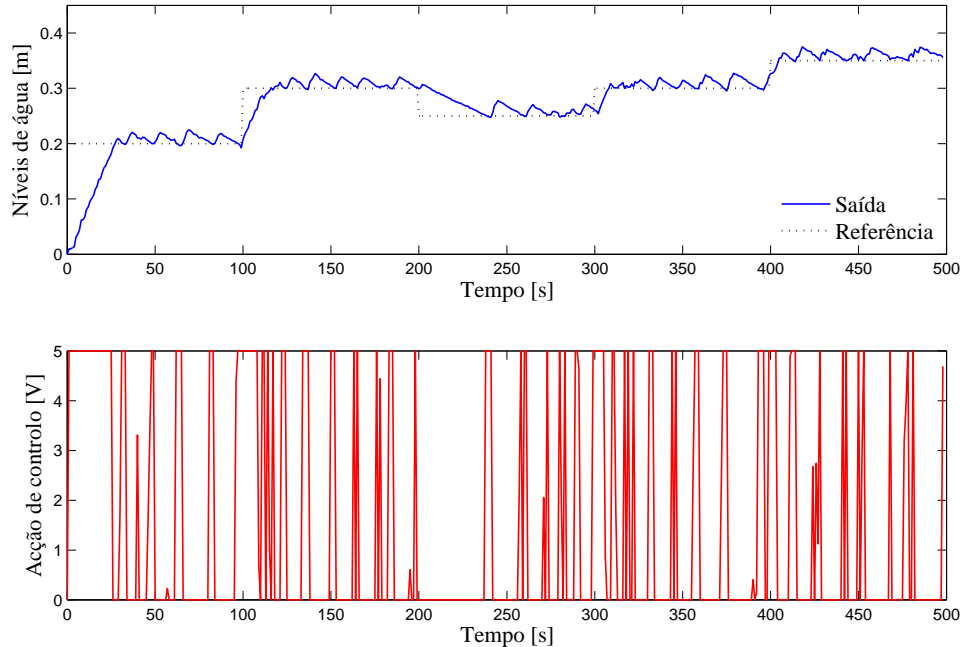


Figura 8.9: Tanque 1 do sistema MIMO controlado por um controlador PID com sintonizaç o baseada em controladores convencionais sobre redes de sensores sem fios.

Relativamente ao controlo efectuado sobre RSSFs, curiosamente, a acç o de controlo   mais suave que no caso do controlo sobre placas de aquisiç o. Contudo, em rela o ao seguimento da refer ncia,   vis vel que este   deteriorado comparativamente   utiliza o directa de placas de aquisiç o.

8.5 Optimiza o do ganhos

A optimiza o dos ganhos do controlador foi feita utilizando a funç o do MATLAB *fmincon* da ferramenta *Optimization Toolbox*. Esta funç o do MATLAB possibilita implementar o m todo de quasi-Newton (descrito na sec o 5.6) na procura pelo m nimo de uma funç o.[Doc12b].

Como funç o de m rito, ou de custo, foi escolhida uma funç o quadr tica que envolve as penaliza es do erro e da varia o da acç o de controlo (ver (8.2)). Utilizando esta funç o de custo pretende-se uma resposta que, para al m do erro ao longo do todo o horizonte de controlo tenha ainda em considera o o desgaste associado ao efeito de actua o.

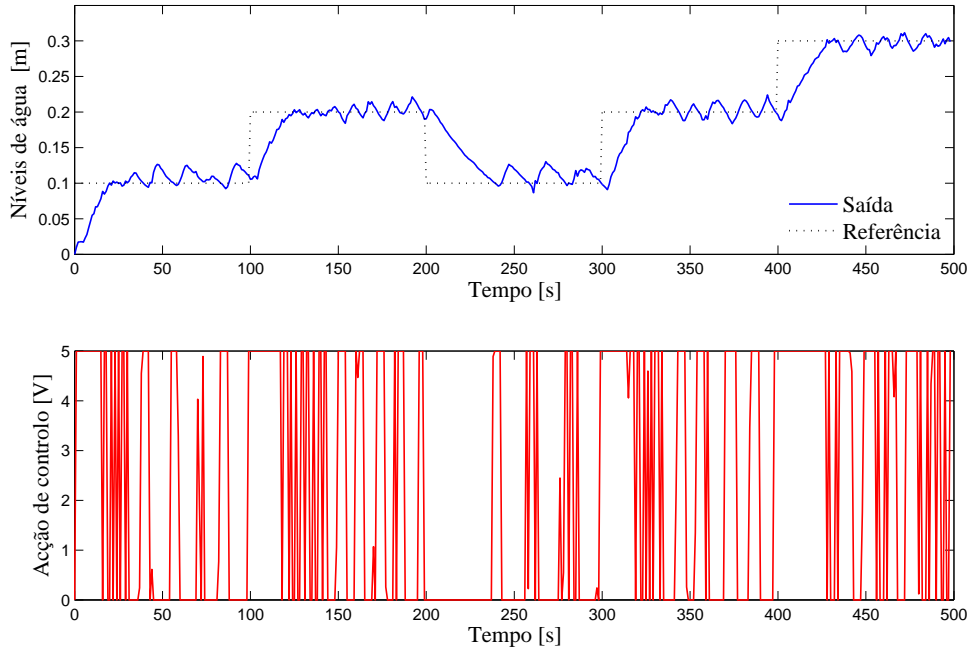


Figura 8.10: Tanque 2 do sistema MIMO controlado por um controlador PID com sintonização baseada em controladores convencionais sobre redes de sensores sem fios.

$$\min_K \left\{ \begin{array}{l} \sum_{k=0}^{N_P} 20 \left(\left[h_1(k) - r_1(k) \right]^2 + \left[h_2(k) - r_2(k) \right]^2 \right) \\ + 0.1 \left(\left[u_2(k) - u_2(k-1) \right]^2 + \left[u_1(k) - u_1(k-1) \right]^2 \right) \end{array} \right\} \quad (8.2)$$

onde $K = [K_{e_1}, K_{\Delta e_1}, K_{\Delta u_1}, K_{u_1}, K_{e_2}, K_{\Delta e_2}, K_{\Delta u_2}, K_{u_2}]$ e N_P a dimensão do horizonte de optimização sujeito à dinâmica do sistema e às restrições:

$$\begin{aligned} 0 &\leq y(k) \leq 1 & k &= 1, \dots, N_P \\ 0 &\leq u(k) \leq 5 & k &= 0, \dots, N_P - 1 \\ K_{e_1}, K_{e_2} &\geq 0 \\ K_{\Delta e_1}, K_{\Delta e_2} &\geq 0 \\ K_{\Delta u_1}, K_{\Delta u_2} &\geq 0 \\ K_{u_1}, K_{u_2} &\geq 0 \end{aligned} \quad (8.3)$$

8.6 PID difuso com otimização em diferido

Os seguintes resultados experimentais coligidos do sistema dizem respeito a um contexto onde a optimização dos ganhos é efectuada em diferido, segundo a arquitectura presente na secção 7.1. Tal como foi referido nesta secção, para a optimização dos ganhos

considerou-se um modelo não linear do sistema, a dimensão do horizonte de otimização será equivalente ao horizonte de controlo e o número de iterações ilimitado.

O modelo não linear do sistema é descrito por uma rede neuronal de três camadas com funções de activação do tipo sigmoidal na camada interna e função de activação linear para as restantes camadas 8.4. De modo a capturar a dinâmica do sistema em anel aberto a rede neuronal foi treinada em diferido utilizando o algoritmo Levenberg-Marquardt. Para este fim foi utilizada a ferramenta de MATLAB *Neural Network Toolbox* [Doc12a].

$$y(k) = \begin{bmatrix} 0.327 & 0.540 \\ 0.059 & 0.036 \\ -0.625 & -1.466 \\ 0.132 & 0.217 \\ 0.001 & -0.002 \\ 0.004 & 0.008 \\ -0.002 & -0.002 \\ -0.001 & -0.005 \end{bmatrix}^T \times \tanh \left(\begin{bmatrix} 6.24 & -2.45 \\ 2.87 & -1.90 \end{bmatrix} \right) \quad (8.4)$$

$$\times \begin{pmatrix} \begin{bmatrix} y_1(k-1) \\ y_1(k-2) \\ u_1(k-1) \\ y_2(k-1) \\ y_2(k-2) \\ u_2(k-1) \end{bmatrix} + \begin{bmatrix} -0.511 \\ -0.435 \end{bmatrix} \times 10^{-3} \end{pmatrix}$$

8.6.1 PID difuso com otimização em diferido dos ganhos sobre placas de aquisição

As Figuras 8.12 e 8.11 mostram a resposta em anel fechado num contexto onde os ganhos são otimizados em diferido recorrendo a placas de aquisição.

Como pode ser observado, a resposta em anel fechado é relativamente satisfatória, exibindo um pequeno tempo de estabelecimento e baixas sobrelevações. Para além disso, em relação à acção de controlo, esta apresenta um comportamento bastante suave. É importante realçar que a sobrelevação não se mantém constante para os diferentes degraus, isso deve-se ao comportamento não linear do processo.

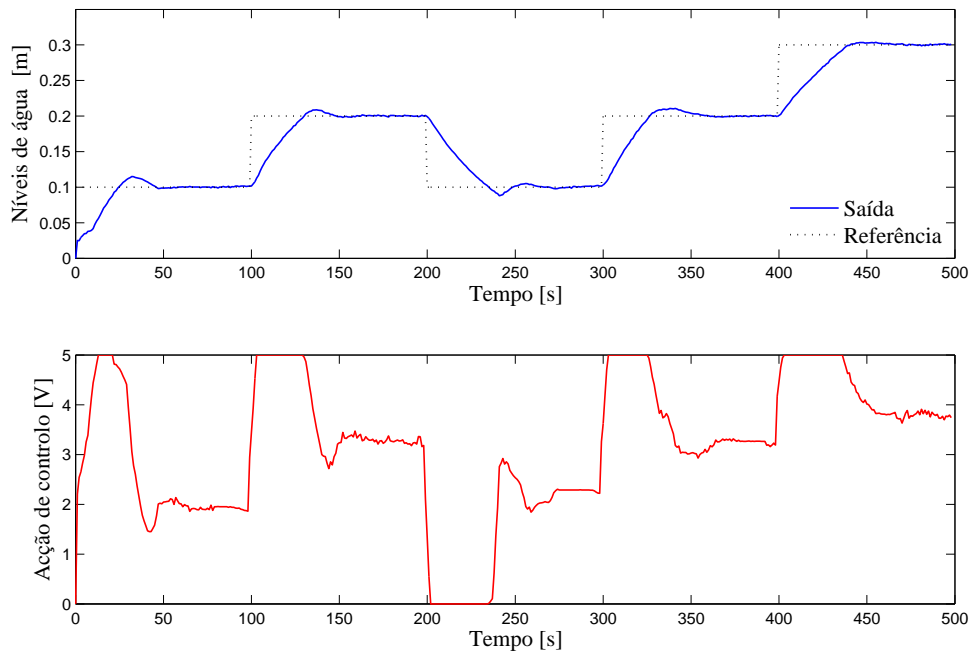


Figura 8.11: Tanque 2 do sistema MIMO controlado por um controlador PID com otimização em diferido dos ganhos sobre placas de aquisição.

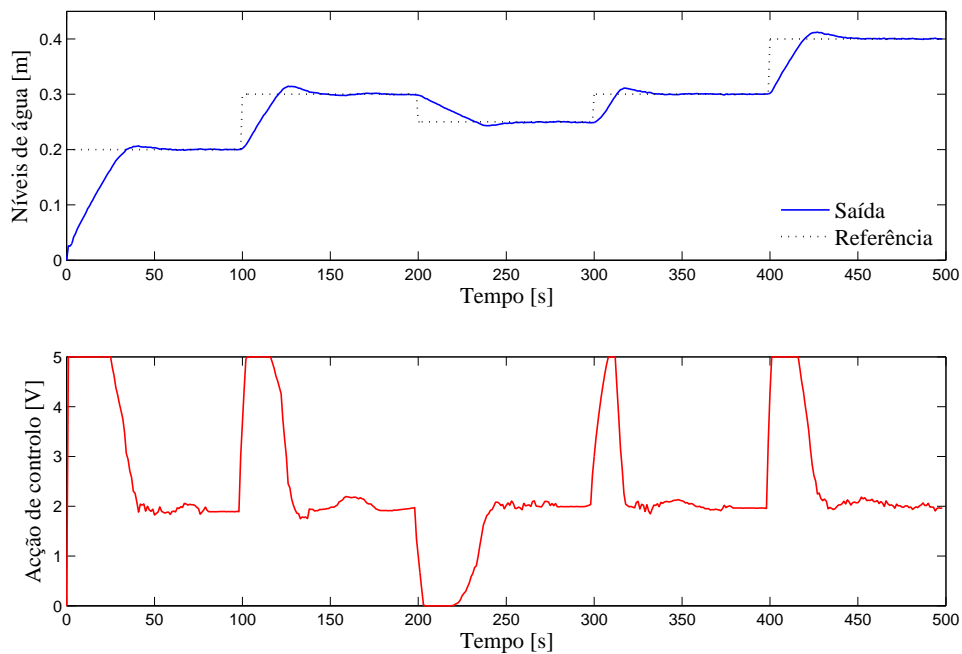


Figura 8.12: Tanque 1 do sistema MIMO controlado por um controlador PID com otimização em diferido dos ganhos sobre placas de aquisição.

8.6.2 PID difuso com otimização em diferido dos ganhos sobre redes de sensores sem fios

As Figuras 8.13 e 8.14 mostram a resposta em anel fechado num contexto onde os ganhos são otimizados em diferido sobre RSSFs.

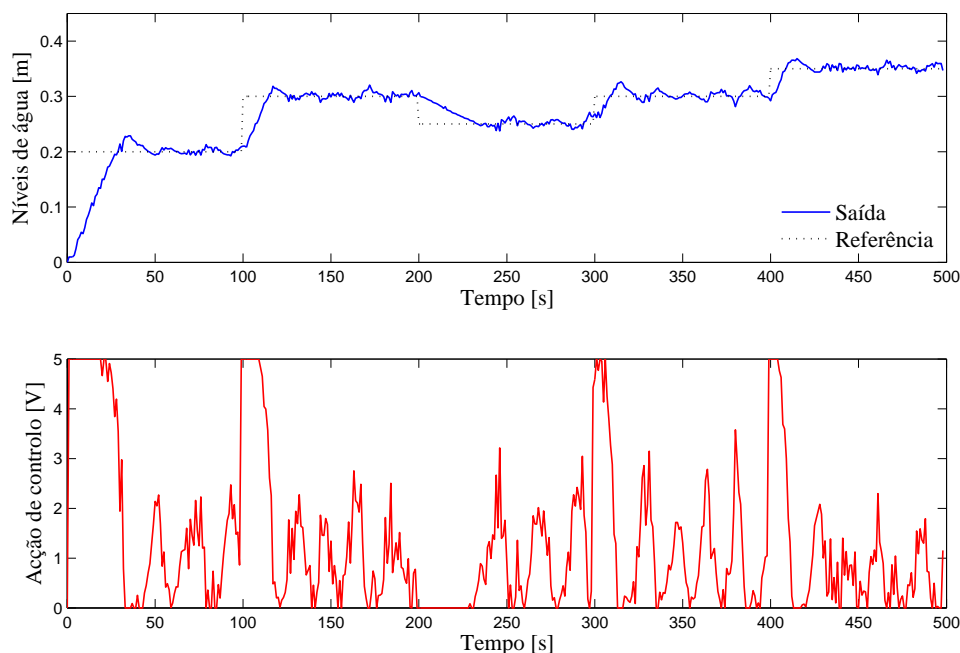


Figura 8.13: Tanque 1 do sistema MIMO controlado por um controlador PID com otimização diferido dos ganhos sobre RSSFs.

Analisando a resposta em anel fechado recorrendo a redes de sensores e actuadores sem fios é possível notar que o seguimento da referência foi ligeiramente deteriorado comparativamente ao controlo efectuado sobre placas de aquisição. Relativamente à acção de controlo, esta mostra-se mais agressiva, provavelmente consequente do ruído de leitura (*outliers*) por parte dos sensores sem fios.

8.7 PID difuso com otimização em linha dos ganhos

Previamente à escolha do número de amostras a adoptar para o horizonte de optimização foi feito um estudo em diferido do seu efeito em relação ao valor do critério de desempenho utilizado na optimização (ver Figura 8.15).

Ao analisar a Figura 8.15 é possível notar que o valor do critério de desempenho tende a estabilizar com o aumento do número de amostras do horizonte de optimização. Tendo

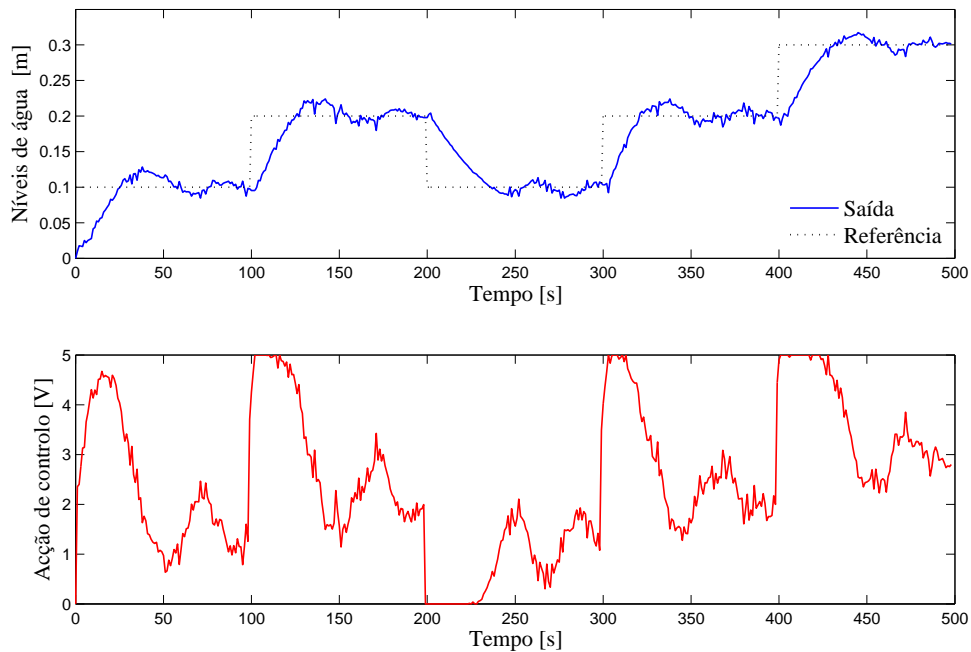


Figura 8.14: Tanque 2 do sistema MIMO controlado por um controlador PID com otimização diferido dos ganhos sobre RSSFs.

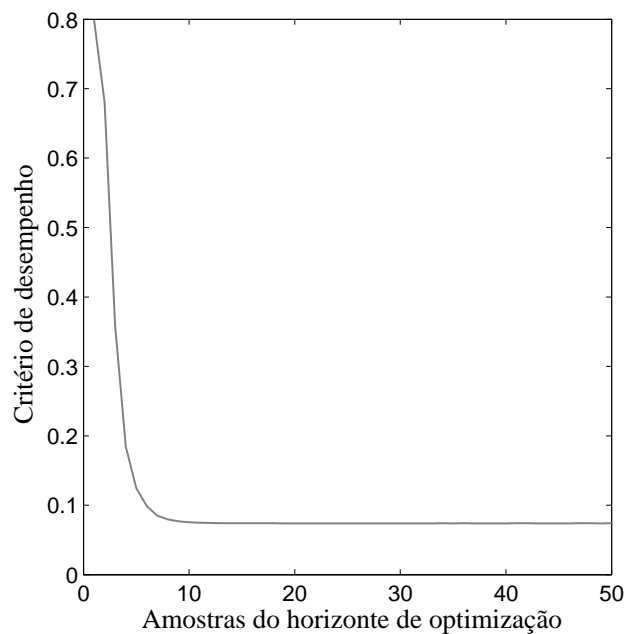


Figura 8.15: Evolução do valor do critério de desempenho em ordem ao número de amostras do horizonte de otimização.

em conta o aspecto referido anteriormente, o número de amostras escolhido para o horizonte de otimização foi vinte, pois é onde ocorre pela primeira vez o mínimo do valor

do critério de desempenho utilizado. A escolha deste valor deve-se ao facto de o tempo consumido no processo de optimização aumentar à medida que o número de amostras do horizonte de optimização aumenta, o que implicará uma redução do número de ganhos optimizados e poderá comprometer o funcionamento do controlador.

De acordo com o estudo efectuado anteriormente a configuração do processo e do ambiente de controlo utilizado encontra-se apresentados na Tabela 8.1.

Tabela 8.1: Configuração do processo e ambiente de controlo.

Parâmetro	Valor
Referência	Conjunto de degraus
Intervalo de amostragem	1 s
Horizonte de optimização	20
Número de iterações do ciclo de optimização	7

A arquitectura utilizada para a optimização em-linha dos ganhos do controlador corresponde à arquitectura apresentada na secção 7.2. Para a comunicação entre as rotinas de optimização, necessárias à optimização dos ganhos referentes aos controladores dos tanques T_1 e T_2 , foi utilizado o protocolo TCP/IP. A implementação das funções necessárias à comunicação TCP/IP foi utilizada a ferramenta do MATLAB *TCP/UDP/IP Toolbox 2.0.6* [Ryd12].

Para a obtenção em diferido dos modelos ARX foi utilizada a ferramenta do MATLAB *System Identification Toolbox* [Doc12c], enquanto que a sua actualização recursiva foi utilizado o algoritmo apresentado na subsecção 4.4.3.

8.7.1 PID difuso óptimo sobre placa de aquisição

As Figuras 8.16 e 8.17 mostram a resposta em anel fechado num contexto onde os ganhos do controlador são optimizados em-linha e as leituras, bem como, actuações realizadas sobre uma placa de aquisição USB.

A evolução dos ganhos resultante do processo de optimização em-linha encontra-se nas Figuras 8.18 e 8.19, respectivamente para os tanques T_1 e T_2 .

Como pode ser observado, à semelhança dos resultados obtidos com a optimização em diferido dos ganhos, a resposta é relativamente satisfatória mostrando pequenos tempos de estabelecimento e baixa sobrelevação. Para os resultados da optimização em-linha é possível notar que a resposta aos diferentes degraus é mais homogénea, consequência da característica de adaptabilidade dos controladores.

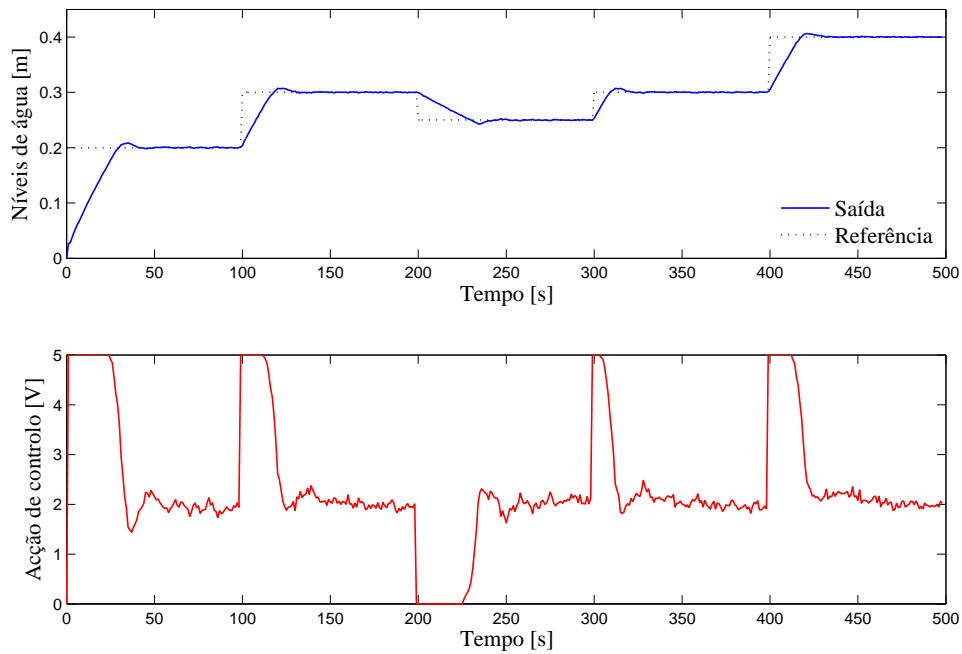


Figura 8.16: Tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.

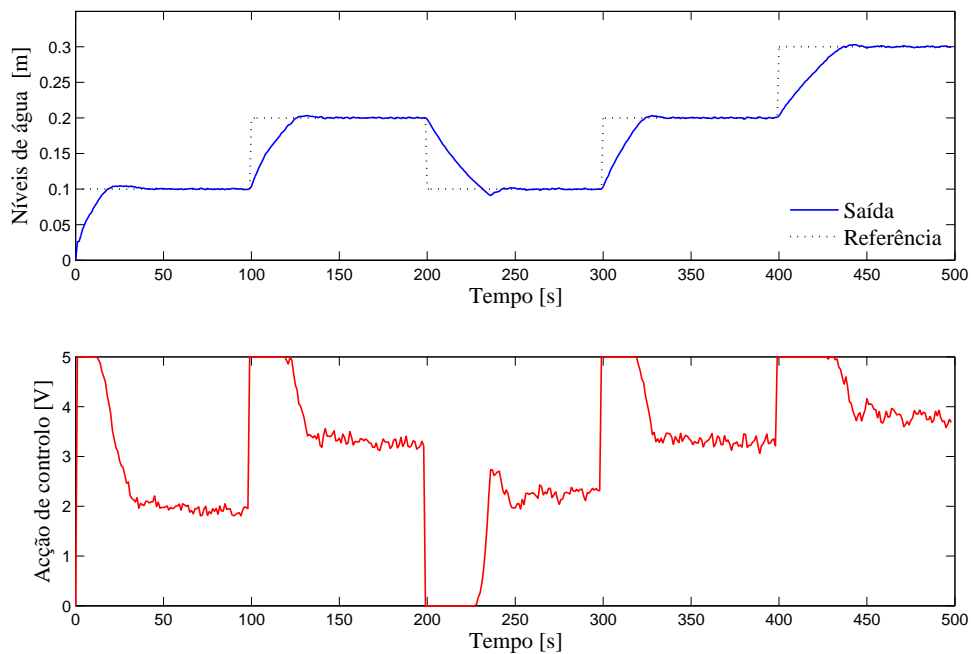


Figura 8.17: Tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.

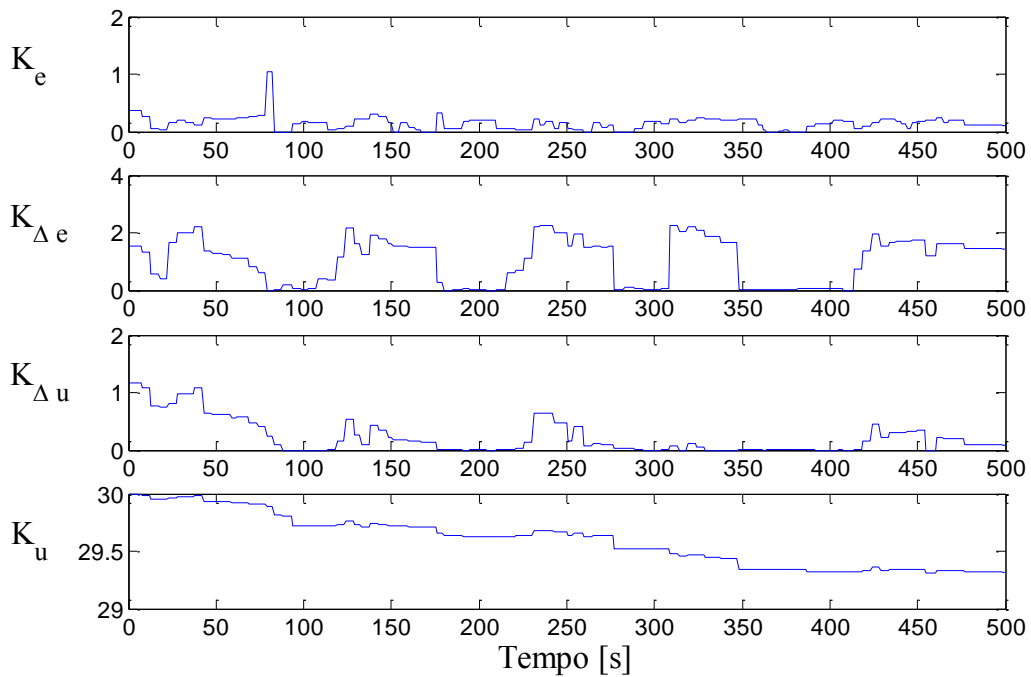


Figura 8.18: Evolução dos ganhos para o tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.

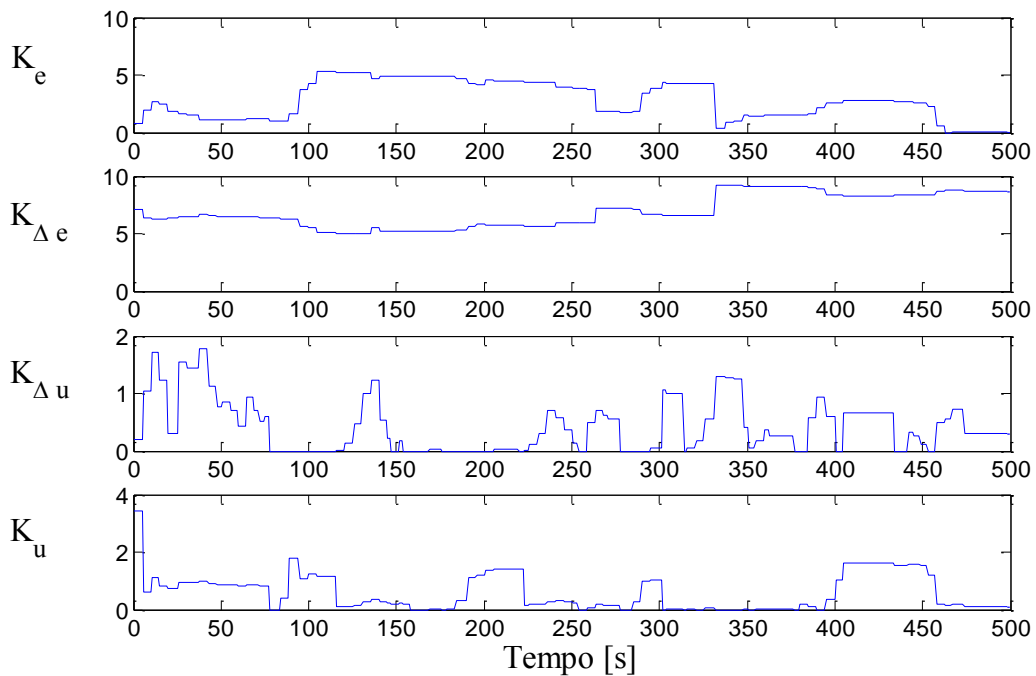


Figura 8.19: Evolução dos ganhos para o tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre placas de aquisição.

8.7.2 PID difuso com otimização em linha dos ganhos sobre redes de sensores sem fios

As Figuras 8.20 e 8.21 mostram a resposta em anel fechado num contexto onde os ganhos do controlador são otimizados em-linha e as leituras e actuações realizadas sobre uma rede de sensores sem fios.

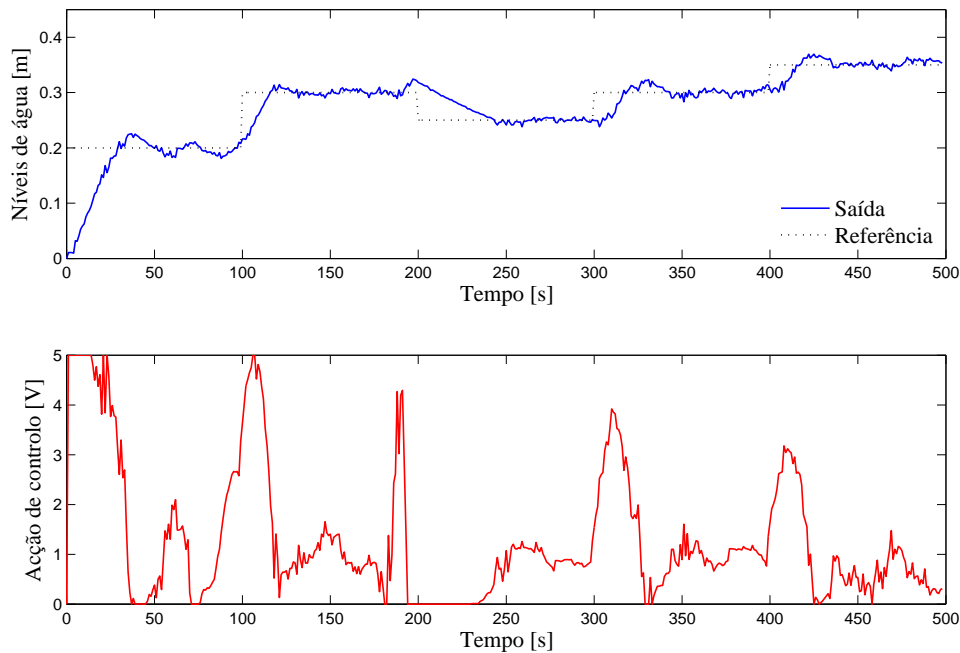


Figura 8.20: Tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre RSSFs.

A evolução dos ganhos resultante do processo de otimização em-linha encontra-se nas Figuras 8.22 e 8.23 respectivamente para os tanques T_1 e T_2 . Conforme se pode constatar, os resultados sobre redes de sensores sem fios, denotam o efeito positivo da adaptação dos ganhos em linha.

8.8 Resultados comparativos

A avaliação dos resultados obtidos será feita segundo métricas que avaliem não só o desempenho do controlador no que refere ao erro, mas também relativamente à variação da acção de controlo. As métricas seleccionadas para avaliar o desempenho do controlador foram a média da raiz dos quadrados do erro (8.5) (RMSR *Root Mean Squared Root*) e a média da raiz dos quadrados da variação da acção de controlo (8.6) (RMSI *Root Mean Squared of action Increment*).

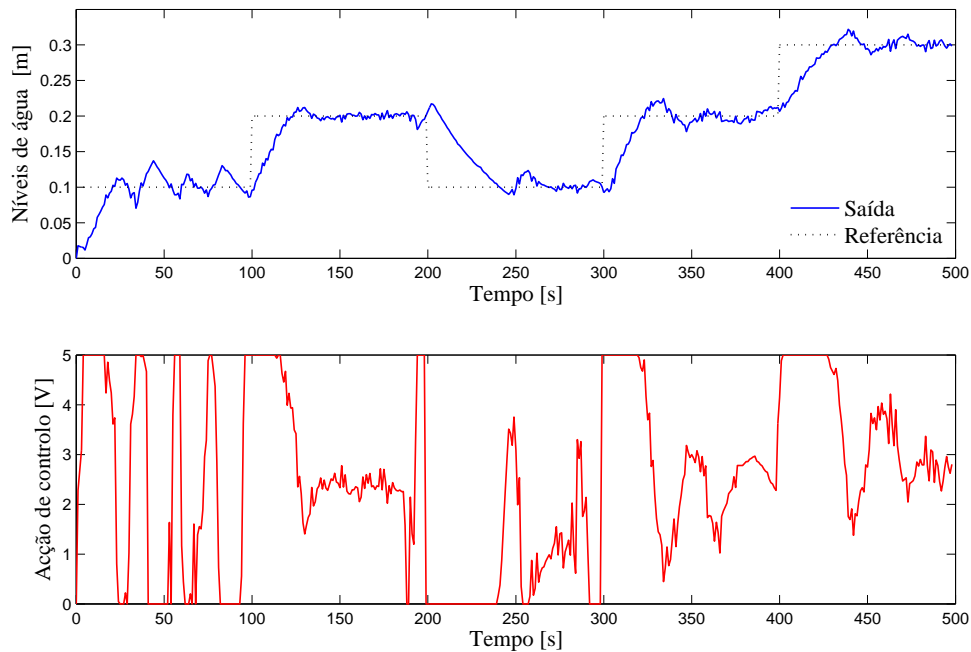


Figura 8.21: Tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre RSSFs.

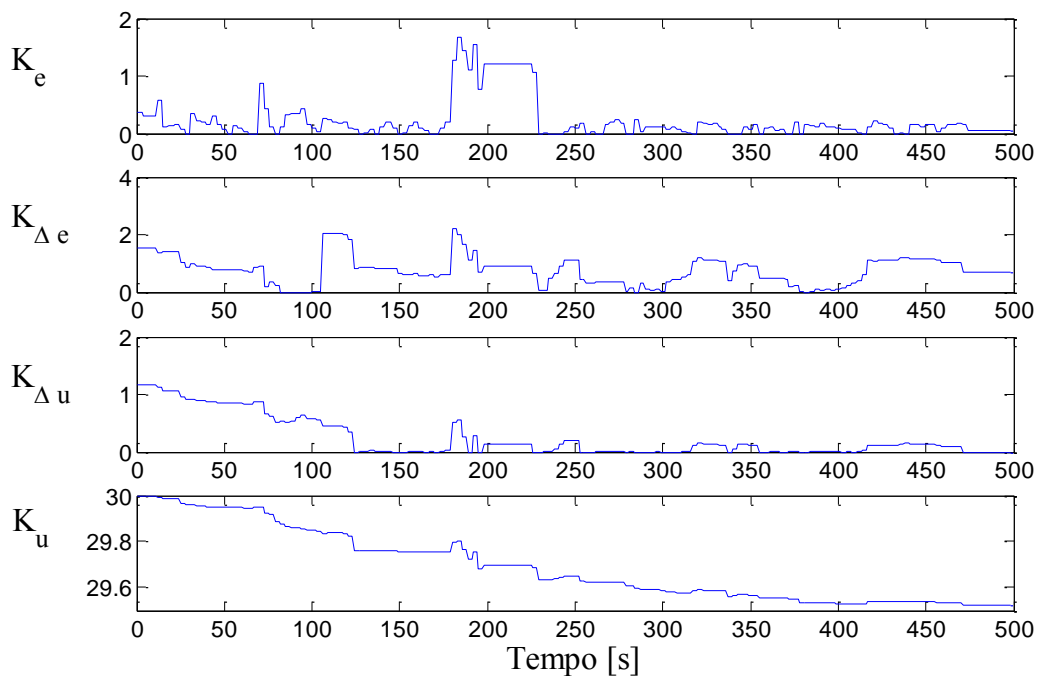


Figura 8.22: Evolução dos ganhos para o tanque 1 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre uma rede de sensores sem fios.

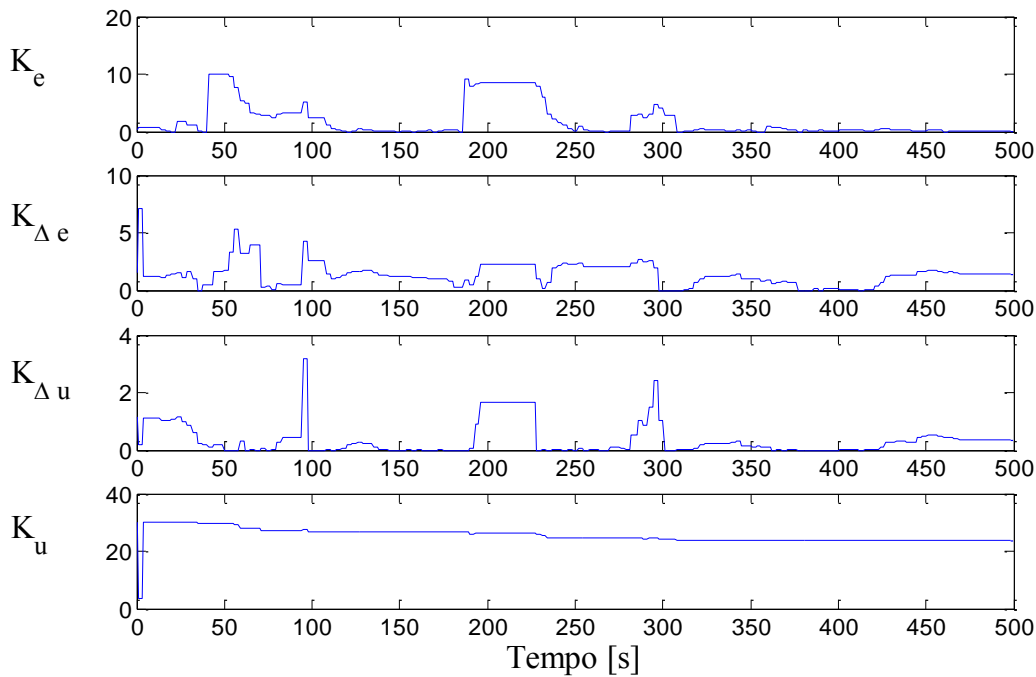


Figura 8.23: Evolução dos ganhos para o tanque 2 do sistema MIMO controlado por um controlador PID com otimização em linha dos ganhos sobre uma rede de sensores sem fios.

Os resultados obtidos segundo as métricas referidas para a experiência efectuada utilizando uma placa de aquisição e sobre redes de sensores sem fios encontram-se respectivamente nas Tabelas 8.2 e 8.3.

$$RMSE = \sqrt{\frac{\sum_{k=1}^N [y(k) - r(k)]^T [y(k) - r(k)]}{\sum_{k=1}^N r(k)^T r(k)}} \quad (8.5)$$

$$RMSI = \frac{\sqrt{[u(k) - u(k-1)]^T [u(k) - u(k-1)]}}{N} \quad (8.6)$$

Tabela 8.2: Medidas de desempenho dos controladores sobre placas de aquisição.

Controlador	Tanque	RMSE[%]	RMSI
PID difuso baseado em PID convencional	T_1	0.1016	0.1431
	T_2	0.1354	0.1347
PID com ganhos otimizados em diferido	T_1	0.1112	0.0123
	T_2	0.1468	0.0086
PID com ganhos otimizados em linha	T_1	0.1028	0.0161
	T_2	0.1372	0.0157

Tabela 8.3: Medidas de desempenho dos controladores sobre RSSFs.

Controlador	Tanque	RMSE[%]	RMSI
PID difuso baseado em PID convencional	T_1	0.1173	0.0944
	T_2	0.1720	0.1082
PID com ganhos otimizados em diferido	T_1	0.1158	0.0296
	T_2	0.1704	0.0145
PID com ganhos otimizados em linha	T_1	0.1233	0.0177
	T_2	0.1727	0.0308

Como pode ser inferido a partir das tabelas 8.2 e 8.3, os três controladores apresentam respostas aproximadamente semelhantes no que se refere ao RMSE. No entanto, em relação ao RMSI os resultados evidenciam um melhor desempenho por parte dos controladores sintonizados recorrendo a um processo de otimização.

9

Conclusões e perspectivas futuras

Nesta secção apresentam-se as conclusões gerais relativas à presente dissertação, assim como algumas ideias que poderão ser desenvolvidas num trabalho futuro.

9.1 Conclusões gerais

Esta dissertação abordou o problema da ausência de uma metodologia sistemática para efeito da sintonização dos ganhos de controladores PID difusos. As metodologias de sintonização propostas são formuladas em termos de um problema de optimização não linear sujeito a restrições, onde o desempenho do sistema em anel fechado é explicitamente tido em consideração.

A primeira metodologia proposta baseia-se na optimização dos ganhos em diferido, utilizando um modelo não linear para descrever o comportamento dinâmico do sistema. Na segunda arquitectura proposta os factores de escala são optimizados em linha recorrendo igualmente à resolução de um problema de optimização com restrições, sendo neste caso a dinâmica do sistema aproximada com base em modelos lineares do tipo ARX. De modo a tornar independente a resolução do problema de optimização do ciclo de controlo e a agilizar o processo de obtenção dos ganhos optou-se por abordagem alicerçada na computação distribuída.

O desempenho dos controladores foi comparado utilizando um processo didáctico dos Três-tanques da AMIRA - DTS 200, e tomando como referência um controlador para o

qual se adoptou como metodologia de sintonização a pseudo-equivalência com um controlador PID convencional. A partir dos resultados obtidos constata-se que o desempenho deste último é inferior no que diz respeito à variação da acção de controlo e evidenciando que as metodologias propostas, para além de serem de prática utilização, consistem também numa abordagem que tem em atenção não só o seguimento da referência, mas também o tempo de vida útil do sistema e um uso mais equilibrado de energia.

Por fim, as arquitecturas implementadas foram aplicadas a uma rede de sensores e actuadores sem fios, observando-se um bom desempenho em ambos os controladores, particularmente no caso do controlador com optimização em linha dos ganhos que apresentou uma maior resiliência relativamente à acomodação do efeito da variação dos tempos de latência.

No âmbito da Dissertação foram desenvolvidos e publicados os artigos de conferência [LCPG12b, LCPG12a, LCPG12c].

9.2 Perspectivas futuras

No decorrer do desenvolvimento da dissertação surgiram algumas ideias, que poderão, eventualmente suscitar algum interesse e enriquecer o trabalho efectuado. Estas ideias poderão servir como base para uma futura investigação, destacando-se:

- Utilização de algoritmos de optimização mais avançados, como por exemplo algoritmos genéticos, no processo de optimização em diferido;
- Aplicação dos métodos de sintonização desenvolvidos a um controlador PID convencional de modo a concluir sobre as reais vantagens da utilização de controladores difusos em sistemas não lineares;
- Análise de desempenho do sistema perante acomodação de *outliers* na rede de sensores sem fios;
- Acoplamento de um módulo que permita efectuar a detecção e a identificação de falhas de comunicação (na rede de sensores sem fios e entre a rotina principal e a de optimização);

Bibliografia

- [AB00] Carl Anderson e John J. Bartholdi. Complexity and complex systems in industry , proceedings , university of centralized versus decentralized control in manufacturing : lessons from social insects. (September):92–105, 2000.
- [ASSC02] Ian F. Akyildiz, Weiiian Su, Yogesh Sankarasubramaniam, e Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102 – 114, 2002. Survey.
- [Ast94] Karl J. Aström. The future of control. *Modeling, Identification and Control*, 15(3):127–134, 1994.
- [Ber88] J. A. Bernard. Use of a rule-based system for process control. 8(5):3–13, 1988.
- [BG07] T Bohlin e S F Graebe. Issues in nonlinear stochastic grey box identification. *International Journal of Adaptive Control and Signal Processing*, 9(6):465, 2007.
- [BJ90] George Edward Pelham Box e Gwilym Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.
- [BPC⁺07] P Baronti, P Pillai, V Chook, S Chessa, A Gotta, e Y Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655–1695, 2007.
- [Cam04] A. Campos. *Modelação e Análise numérica do Comportamento Mecânico e Térmico de Ligas de Alumínio*. Phd, Universidade de Aveiro Aveiro. Wiley-Interscience, 2004.
- [CBG90] S. Chen, S. A. Billings, e P. M. Grant. Non-linear system identification using neural networks. In *International Journal of Control*, pág. 1191–1214, 1990.
- [CZ72] Sheldon S. L. Chang e Lofti A. Zadeh. On fuzzy mapping and control. (1):30–34, 1972.

- [DGV04] Adam Dunkels, Björn Grönvall, e Thiemo Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, Novembro 2004.
- [DHR93] D. Driankov, H. Hellendoorn, e M. Reinfrank. *An introduction to fuzzy control*. Springer-Verlag New York, Inc., New York, NY, USA, 1993.
- [DM73] John E. Dennis, Jr. e Jorge J. More. *A Characterization of Superlinear Convergence and its Application to Quasi-Newton Methods*. Cornell University, Ithaca, NY, USA, 1973.
- [Doc12a] R2011b Documentation. Neural network toolbox. Website, 2012. <http://www.mathworks.com/help/toolbox/nnet/ref/trainlm.html>.
- [Doc12b] R2011b Documentation. Optimization toolbox. Website, 2012. <http://www.mathworks.com/help/toolbox/optim/ug/fmincon.html>.
- [Doc12c] R2011b Documentation. System identification toolbox. Website, 2012. <http://www.mathworks.com/help/toolbox/ident/ug/bq54wup.html>.
- [DP10] W. Dargie e C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wireless Communications and Mobile Computing. John Wiley & Sons, 2010.
- [DSVA06] Adam Dunkels, Oliver Schmidt, Thiemo Voigt, e Muneeb Ali. Protothreads: Simplifying event-driven programming of memory-constrained embedded systems. In *Proceedings of the Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Boulder, Colorado, USA, Novembro 2006.
- [DW79] J.E. Dennis e H.F. Walker. *Local convergence theorems for quasi-Newton methods*. Technical report (Cornell University. Dept. of Computer Science). Dept. of Computer Science, Cornell University, 1979.
- [Fen06] Gang Feng. A survey on analysis and design of model-based fuzzy control systems. 14(5):676–697, 2006.
- [HHKK04] Jason Hill, Mike Horton, Ralph Kling, e Lakshman Krishnamurthy. The platforms enabling wireless sensor networks. *Commun. ACM*, 47(6):41–46, Junho 2004.
- [HL09] F. Herrera e M. Lozano. Fuzzy evolutionary algorithms and genetic fuzzy systems: A positive collaboration between evolutionary algorithms and fuzzy systems. 1:83–130, 2009.

- [IM05] A. Izmailov e Solodov M. *Otimização*, volume 2. IMPA, 2005.
- [KF87] George J. Klir e Tina A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.
- [Kos92] B. Kosco. Fuzzy systems as universal approximators. *Proc. IEEE Int Fuzzy Systems Conf*, pág. 1153–1162, 1992.
- [KW05] Holger Karl e Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.
- [KY95] G. J. Klir e B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall PTR, NJ, USA, 1st edition, Maio 1995.
- [LC99] E.G.M. Lacerda e A. P. L. F. Carvalho. *Introdução aos Algoritmos Genéticos*, volume 2. 1999.
- [LCPG12a] C. Lucena, A. Cardoso, L. Palma, e P. Gil. Adaptive quasi-optimal gains tuning of pi-fuzzy controllers. *IEEE 16th International Conference on Intelligent Engineering Systems 2012*, 2012.
- [LCPG12b] C. Lucena, A. Cardoso, L. Palma, e P. Gil. Gains tuning of fuzzy controllers based on a cost function optimization. *Controlo 2012*, 2012.
- [LCPG12c] C. Lucena, A. Cardoso, L. Palma, e P. Gil. Optimal gains tuning of pi-fuzzy controllers. *20th Mediterranean Conference on Control and Automation*, 2012.
- [Lee90] C. C. Lee. Fuzzy logic in control systems: fuzzy logic controller. i. *IEEE J SMC*, 20(2):404–418, 1990.
- [Lew04] FL Lewis. Wireless sensor networks. *Smart Environments*, 2004. Query date: 2012-03-31.
- [LG96] Han-Xiong Li e H. B. Gatland. Conventional fuzzy control and its enhancement. *IEEE J SMCB*, 26(5):791–797, 1996.
- [Lju87] Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 1987.
- [LMPLdB05] L. M. Rodriguez Peralta L. M. Pestana Leão de Brito. *Basic Concepts of Wireless Sensor Networks*. Encyclopedia of Networked and Virtual Organizations, 2005.
- [Mam74] E. H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12):1585–1588, 1974.

- [MMC96] Dave Misir, Heidar A. Malki, e Guanrong Chen. A heuristic approach to determine the gains of a fuzzy pid controller. In *Proceedings of the 1996 ACM symposium on Applied Computing, SAC '96*, pág. 609–613, New York, NY, USA, 1996. ACM.
- [MS85] Michio e Sugeno. An introductory survey of fuzzy control. *Information Sciences*, 36(2):59 – 83, 1985.
- [Mur69] Walter Murray. An algorithm for constrained minimization. pág. 247–258, London and New York, 1969. Academic Press.
- [NP90a] K. S. Narendra e K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, Março 1990.
- [NP90b] K S Narendra e K Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.
- [Oli96] J. V Oliveira. *Formalismos Difusos em Controlo Adaptativo Preditivo. Tese de Doutoramento, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal.* 1996.
- [PH11] Radu-Emil Precup e Hans Hellendoorn. A survey on industrial applications of fuzzy control. *Computers in Industry*, 62(3):213 – 226, 2011.
- [Piv02] P. Pivonka. Comparative analysis of fuzzy pi/pd/pid controller based on classical pid controller approach. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, volume 1, pág. 541–546, 2002.
- [Pol97] Elijah Polak. *Optimization. Algorithms and Consistent Approximations.* Applied Mathematical Sciences. Springer, 1997.
- [PSPY97] A. S. Poznyak, E. N. Sanchez, J. P. Perez, e W. Yu. Robust adaptive nonlinear system identification and trajectory tracking by dynamic neural networks. In *Proc. American Control Conf the 1997*, volume 1, pág. 242–246, 1997.
- [R.95] Jagger R. *Logic in Control.* Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 1995.
- [Rao96] Singiresu S. Rao. *Engineering Optimization: Theory and Practice, 3rd Edition.* Wiley-Interscience, 1996.
- [RC95a] G. A. Rovithakis e M. A. Christodoulou. Direct adaptive regulation of unknown nonlinear dynamical systems via dynamic neural networks. *IEEE J SMC*, 25(12):1578–1594, 1995.

- [RC95b] George A. Rovithakis e Manolis A. Christodoulou. Regulation of unknown nonlinear dynamical systems via dynamical neural networks. *Journal of Intelligent and Robotic Systems*, 12(3):259–275, 1995.
- [Rez97] Leonid Reznik. Fuzzy controllers. *Automatica*, pág. 8, 1997.
- [RIG99] R Ramirez-Iniguez e R. J. Green. Indoor optical wireless communications. *IEE Colloquium on Optical Wireless Communications*, 1999.
- [Ryd12] Peter Rydesäter. Tcp/udp/ip toolbox 2.0.6. Website, 2012. <http://www.mathworks.com/matlabcentral/fileexchange/345>.
- [Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [Sid82] James N. Siddall. *Optimal Engineering Design: Principles and Applications*. Marcel Dekker, Inc., New York, NY, USA, 1982.
- [SVAS78] Jr. Sandell, N., P. Varaiya, M. Athans, e M. Safonov. Survey of decentralized control methods for large scale systems. *Automatic Control, IEEE Transactions on*, 23(2):108 – 128, apr 1978.
- [SZL⁺95] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Deylon, Pierre yves Glorennec, Hakan Hjalmarsson, e Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, 1995.
- [Tak81] Floris Takens. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence*, 898:366–381, 1981.
- [WTG96] H. O. Wang, K. Tanaka, e M. F. Griffin. An approach to fuzzy control of nonlinear systems: stability and design issues. 4(1):14–23, 1996.
- [YMG08] Jennifer Yick, Biswanath Mukherjee, e Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52:2292–2330, August 2008.
- [Zad65] L.A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
- [Zad73] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE J SMC*, (1):28–44, 1973.