

Controlo de um Sistema de Difusão de Calor

DIOGO ALEXANDRE MOREIRA DA COSTA
novembro de 2023

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Controlo de um Sistema de Difusão de Calor

Diogo Alexandre Moreira da Costa

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Automação e Sistemas



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Novembro, 2023

Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores, Área de Especialização em Automação e Sistemas.

Candidato: Diogo Alexandre Moreira da Costa, N.º 1210169,
1210169@isep.ipp.pt

Orientação Científica: Professora Isabel Maria de Sousa de Jesus,
isj@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Novembro, 2023

Agradecimentos

Esta dissertação é concluída graças à presença de várias pessoas que me apoiaram durante o seu período de desenvolvimento, onde apareceram inúmeros desafios pessoais, às quais quero deixar aqui os meus agradecimentos.

Agradeço à minha orientadora, Professora Isabel Jesus, pela sua orientação e dedicação, bem como a compreensão e apoio durante períodos mais desafiantes. Esteve sempre disponível para me ajudar e orientar, e as suas contribuições foram fundamentais para o desenvolvimento e conclusão deste trabalho.

Agradeço aos meus pais, irmã, avós, tios e restante família, que estiveram presentes durante a realização da dissertação, dando sempre o seu apoio.

Às minhas colegas de casa, que se encontram, tal como eu, a estudar engenharia, agradeço por toda a convivência e momentos de estudo em conjunto, que permitiram que tivesse motivação para continuar a escrever esta dissertação, mesmo quando a vontade desvanecia.

Por último, mas não menos importante, agradeço à minha namorada por estar perto de mim durante este período e não deixar que ficasse desmotivado, e a todos os meus amigos próximos que, de inúmeras formas, estiveram presentes e permitiram ter momentos de lazer para aproveitar os pequenos momentos disponíveis.

Resumo

O controlo de sistemas de difusão de calor pode ser usualmente controlado por PID, aplicando os diversos métodos existentes para este tipo de controlo. No entanto, existem outras opções de controlo possíveis que podem apresentar melhores resultados, como a lógica difusa.

A lógica difusa pode ser definida como a lógica precisa da imprecisão e aproximação. Esta permite que informação qualitativa e imprecisa, seja apresentada de forma quantitativa. Com esta capacidade, é possível aplicar uma resolução prática e rigorosa com a manipulação qualitativa de informação, inicialmente expressa de forma linguística.

Com o objetivo de verificar se a lógica difusa tem um melhor desempenho, no controlo de um sistema de difusão de calor, do que o controlo PID clássico, é efetuado um estudo de conceitos inerentes à lógica difusa e aos métodos de controlo convencionais.

Utilizando um sistema eletrónico de regulação térmica, inicialmente fazendo uma introdução da sua resposta à variação de temperatura, são aplicados vários métodos de controlo. Recorrendo ao MATLAB, são aplicados algoritmos para controlo PID e por lógica difusa, bem como a aquisição e a análise desses dados.

Com a apresentação dos resultados das experiências práticas realizadas, aplicando diversos métodos e parâmetros, é selecionado o método com melhor resposta temporal de cada opção de controlo, sendo apresentada a sua comparação e tecidas as correspondentes conclusões.

Palavras-Chave: Lógica Difusa, Controlo PID, Sistema Térmico, Resposta térmica ao controlo, MATLAB.

Abstract

The control of heat diffusion systems can usually be controlled by PID, applying the several methods existing for this type of control. However, there are other possible control options that may present better results, such as fuzzy logic.

Fuzzy logic can be defined as the precise logic of imprecision and approximation. It allows qualitative and imprecise information to be presented in a quantitative way. With this capability, it is possible to apply a practical and rigorous resolution with the manipulation of qualitative information, initially expressed in a linguistic way.

With the objective of verifying whether fuzzy logic has a better performance, in the control of a heat diffusion system, than PID control, a study of concepts inherent to fuzzy logic and conventional control methods is carried out.

Using an electronic thermal regulation system, initially making an introduction of its response to temperature variation, several control methods are applied. With MATLAB, algorithms for PID control and fuzzy logic control are applied, as well as the acquisition and analysis of these data.

With the presentation of the results of the practical experiments carried out, applying various methods and parameters, the method with the best temporal response of each control option is selected, and its comparison and conclusions are presented.

Keywords: Fuzzy Logic, PID Control, Thermal System, Thermal Response to Control, MATLAB.

Índice

Lista de Figuras	vii
Lista de Tabelas	xi
Listagens	xiii
Lista de Acrónimos	xv
1 Introdução	1
1.1 Definição do Problema	1
1.1.1 Objetivos	1
1.1.2 Resultados esperados	2
1.2 Plano de Trabalho	2
1.3 Organização da Dissertação	3
2 Lógica Difusa e Controlo de Sistemas	5
2.1 Lógica Difusa	5
2.1.1 História	6
2.1.2 Princípios da Lógica Difusa	7
2.1.3 Funções de Pertença e Variáveis Linguísticas	9
2.1.4 Regras Difusas	15
2.1.5 Sistema de Inferência Difusa	17
2.1.6 Tipos de Controlador	21
2.1.7 Vantagens e Aplicações	24
2.2 Toolbox do MATLAB	25
2.3 Sistemas de Controlo	31
2.3.1 Controladores PID	32
2.3.2 Sintonia por Ziegler-Nichols	36
2.3.3 Sintonia por Cohen-Coon	40
2.4 Índices de Desempenho	41
3 Sistema SERT e Controlo Convencional	43
3.1 Sistema de Controlo	43
3.1.1 Configuração do SERT	45

3.1.2	Teste inicial do SERT	46
3.2	Modelação dos Sistemas	49
3.3	Controlo PID	55
3.3.1	Sistema 1 - Ziegler-Nichols em malha fechada	58
3.3.2	Sistema 1 - Ziegler-Nichols em malha aberta	63
3.3.3	Sistema 1 - Cohen-Coon	65
3.3.4	Sistema 2 - Ziegler-Nichols em malha fechada	66
3.3.5	Sistema 2 - Ziegler-Nichols em malha aberta	69
3.3.6	Sistema 2 - Cohen-Coon	70
3.3.7	Comparação dos métodos	71
3.3.8	Aplicação de <i>Anti-windup</i>	72
3.3.9	Análise do melhor controlador PID	77
4	Lógica Difusa e Análise de Resultados	81
4.1	Controlo por Lógica Difusa	81
4.1.1	Controlo P	86
4.1.2	Controlo I	89
4.1.3	Controlo PD	92
4.1.4	Controlo PI	95
4.1.5	Controlo PD+I	99
4.1.6	Controlo PID por um FLC	102
4.1.7	Controlo PID por dois FLC	104
4.1.8	Controlo PID por <i>Gain Scheduling</i>	107
4.1.9	Comparação de Resultados Difusos	113
4.2	Controlo por PID e por Lógica Difusa	116
5	Conclusões	119
5.1	Desenvolvimentos Futuros	120
	Referências	121
	Anexo A <i>Datasheet</i>	125
A.1	Arduino UNO	125
A.2	ATmega328P	128
A.3	TMP36	131
	Anexo B Esquema SERT	135

Lista de Figuras

1.1	Diagrama de Gantt	2
2.1	Ilustração da Lógica Difusa [4]	6
2.2	Classificação de Lógica [12]	7
2.3	Distribuição de Idades em Lógica [2]	9
2.4	Classificação Difusa na Idade	9
2.5	Função de Pertença - Triangular	10
2.6	Função de Pertença - Trapezoidal	10
2.7	Função de Pertença - Gaussiana	11
2.8	Função de Pertença - Forma S	11
2.9	Função de Pertença - Forma S Linear	12
2.10	Função de Pertença - Forma Z	12
2.11	Função de Pertença - Forma Z Linear	13
2.12	Função de Pertença - Forma de Sino	13
2.13	Valores Linguísticos numa Função de Pertença [7]	14
2.14	Compressão de dados e granulação	15
2.15	Estrutura do SIF	17
2.16	Método de Defuzificação - Centro de Área/Gravidade	18
2.17	Método de Defuzificação - Centro de Maior Área	19
2.18	Método de Defuzificação - Menor/Maior do Máximo das Alturas	19
2.19	Método de Defuzificação - Meio do Máximo das Alturas	20
2.20	Método de Defuzificação - Média dos Máximos	20
2.21	Operações de um SIF [15]	21
2.22	Exemplo com o Controlador de Mamdani [13]	22
2.23	Exemplo com o Controlador de Sugeno [13]	23
2.24	Instalar a <i>toolbox</i>	25
2.25	Aceder à <i>app</i> da <i>toolbox</i>	25
2.26	<i>App</i> - Menu inicial	26
2.27	<i>App</i> - Vista geral do controlador de Mamdani	26
2.28	<i>App</i> - Propriedades do controlador	27
2.29	<i>App</i> - Gestão das entradas/saídas	27
2.30	<i>App</i> - Edição das funções de pertença	28
2.31	<i>App</i> - Gestão da base de regras	28

2.32	<i>App</i> - Edição de regras	29
2.33	<i>App</i> - Menu de simulação	29
2.34	<i>App</i> - Processo de inferência difusa	30
2.35	<i>App</i> - Valores de Controlo	30
2.36	<i>App</i> - Conversão de controladores	31
2.37	Sistema de Controlo Genérico	31
2.38	Controlo por Relé	32
2.39	Controlo por Histerese	32
2.40	Sistema de Controlo com componentes PID	34
2.41	Resposta-tipo de um controlador PID [24]	34
2.42	Período Crítico	37
2.43	Ganho e Período Crítico com Ziegler-Nichols [21]	38
2.44	Sistema em Malha Aberta	38
2.45	Teste do Sistema em Malha Aberta [27]	39
3.1	Sistema de Controlo - SERT	44
3.2	Sistema de Controlo - Identificação dos sistemas	44
3.3	Toolbox para comunicação com o SERT	45
3.4	IDE Arduino	45
3.5	Teste do SERT - Gráfico de Temperatura	48
3.6	Diagrama - Resposta em Degrau	50
3.7	Resposta em Degrau dos Sistemas 1 e 2	51
3.8	Constante de Atraso do Sistema 1	53
3.9	Experiência vs Modelo do Sistema 1	54
3.10	Experiência vs Modelo do Sistema 2	55
3.11	Diagrama - Controlo P	55
3.12	Diagrama - Controlo PI	56
3.13	Diagrama - Controlo PID	57
3.14	Janela de Controlo	59
3.15	Resposta à Variação do Ganho S1 - Experiência 1	60
3.16	Observação do Período S1 - Experiência 1	61
3.17	Resposta à Variação do Ganho S1 - Experiência 2	61
3.18	Ganho e Período Crítico S1	62
3.19	Controlo por Z-N em malha fechada do Sistema 1	63
3.20	Controlo por Z-N em malha aberta do Sistema 1	64
3.21	Controlo por C-C do Sistema 1	65
3.22	Resposta à Variação do Ganho S2 - Experiência 1	66
3.23	Resposta à Variação do Ganho S2 - Experiência 2	67
3.24	Ganho e Período Crítico S2	67
3.25	Controlo por Z-N em malha fechada do Sistema 2	68

3.26	Controlo por Z-N em malha aberta do Sistema 2	69
3.27	Controlo por C-C do Sistema 2	70
3.28	Diagrama - Controlo PI com <i>Anti-windup</i>	73
3.29	Diagrama - Controlo PID com <i>Anti-windup</i>	74
3.30	Gráfico para <i>Anti-windup</i> do Sistema 1	75
3.31	Gráfico para <i>Anti-windup</i> do Sistema 2	76
3.32	Gráfico da Melhor Resposta do Sistema 1	78
3.33	Gráfico da Melhor Resposta do Sistema 2	79
4.1	Classificação de Controlo Fuzzy [34]	82
4.2	Diagrama de um controlador difuso híbrido	83
4.3	Sistema do FLC	84
4.4	Configuração Geral do FLC	84
4.5	Função de Pertença de Entrada do FLC	85
4.6	Funções de Pertença de Saída do FLC	85
4.7	Esquema do FLC com Controlo P	86
4.8	Curva de Controlo P	87
4.9	Controlo P por FLC	88
4.10	Esquema do FLC com Controlo I	89
4.11	Curva do Controlo I	89
4.12	Controlo I por FLC	90
4.13	Esquema do FLC com Controlo I com <i>anti-windup</i>	91
4.14	Controlo I por FLC com <i>anti-windup</i>	92
4.15	Esquema do FLC com Controlo PD	93
4.16	Superfície de Controlo - PD	93
4.17	Controlo PD por FLC	94
4.18	Esquema do FLC com Controlo PI	95
4.19	Superfície de Controlo - PI	96
4.20	Controlo PI por FLC	97
4.21	Esquema do FLC com Controlo PI com <i>anti-windup</i>	97
4.22	Controlo PI por FLC com <i>anti-windup</i> do Sistema 2	98
4.23	Esquema do FLC com Controlo PD+I	99
4.24	Controlo PD+I por FLC	100
4.25	Esquema do FLC do Controlo PD+I com <i>anti-windup</i>	100
4.26	Controlo PD+I por FLC com <i>anti-windup</i>	101
4.27	Esquema de Controlo PID por um FLC	102
4.28	Controlo PID por um FLC	103
4.29	Esquema de Controlo PID por dois FLC	104
4.30	Controlo PID por dois FLC	106
4.31	Esquema simples do Controlo PID por <i>Gain Scheduling</i>	107

4.32	Sistema Difuso - <i>Gain Scheduling</i>	108
4.33	Superfície de Controlo do <i>Gain Scheduling</i> - Ganho P	109
4.34	Superfície de Controlo do <i>Gain Scheduling</i> - Ganho D	110
4.35	Superfície de Controlo do <i>Gain Scheduling</i> - α	110
4.36	Esquema de Controlo PID por <i>Gain Scheduling</i>	111
4.37	Controlo PID por Gain Scheduling	111
4.38	Esquema de Controlo PID por <i>Gain Scheduling</i> com <i>Anti-windup</i>	112
4.39	Controlo Fuzzy - <i>Gain Scheduling</i> com <i>Anti-windup</i>	112
4.40	Controlo Difuso do Sistema 1 - Comparação	114
4.41	Controlo Difuso do Sistema 2 - Comparação	115
4.42	Controlo do Sistema 1 - Comparação	116
4.43	Controlo do Sistema 2 - Comparação	117

Lista de Tabelas

2.1	Propriedades dos Métodos de Defuzificação	21
2.2	Vantagens do Controlador de Mamdani e Sugeno	23
2.3	Efeitos dos ganhos PID	35
2.4	Diferentes controladores PID e suas aplicações	36
2.5	Cálculo dos parâmetros de controladores através de Ziegler-Nichols em malha fechada	37
2.6	Cálculo dos parâmetros de controladores através de Ziegler-Nichols em malha aberta	39
2.7	Cálculo dos parâmetros de controladores através de Cohen-Coon	40
3.1	Teste do SERT - Resumo de Dados	49
3.2	Parâmetros do Sistema 1	52
3.3	Parâmetros do Sistema 2	54
3.4	Parâmetros para Z-N em malha fechada do Sistema 1	62
3.5	Parâmetros para Z-N em malha aberta do Sistema 1	64
3.6	Parâmetros para C-C do Sistema 1	65
3.7	Parâmetros para Z-N em malha fechada do Sistema 2	68
3.8	Parâmetros para Z-N em malha aberta do Sistema 2	69
3.9	Parâmetros para C-C do Sistema 2	70
3.10	Comparação de Erros do PID geral - Sistema 1	71
3.11	Comparação de Erros do PID geral - Sistema 2	72
3.12	Parâmetros de esquema com <i>Anti-windup</i> do Sistema 1	75
3.13	Parâmetros de esquema com <i>Anti-windup</i> do Sistema 2	76
3.14	Comparação de Erros PID melhorado - Sistema 1	77
3.15	Comparação de Erros PID melhorado - Sistema 2	78
4.1	Tipos de Controle Difuso por Ação Direta	82
4.2	Lista de Regras do Controle P	87
4.3	Lista de Regras do Controle I	89
4.4	Lista de Regras do Controle PD	93
4.5	Lista de Regras do Controle PI	95
4.6	Erros FLC de controle PI - Sistema 1	98
4.7	Erros FLC de controle PI - Sistema 2	98

4.8	Erros do controlo PID com um FLC - Sistema 1	104
4.9	Erros do controlo PID com um FLC - Sistema 2	104
4.10	Erros do controlo PID com dois FLC - Sistema 1	106
4.11	Erros do controlo PID com dois FLC - Sistema 2	107
4.12	Lista de Regras do Ganho P	109
4.13	Lista de Regras do Ganho D	109
4.14	Lista de Regras da constante α	110
4.15	Erros FLC de controlo PID por <i>Gain Scheduling</i>	113
4.16	Erros de Controlo Difuso - Sistema 1	114
4.17	Erros de Controlo Difuso - Sistema 2	115
4.18	Erros do Sistema 1	117
4.19	Erros do Sistema 2	117

Listagens

3.1	Teste do SERT - Comunicação com o SERT	46
3.2	Teste do SERT - Dados Iniciais	47
3.3	Teste do SERT - Ativação de Temperatura	47
3.4	Teste do SERT - Fim de Comunicação	48
3.5	Resposta em Degrau - Controlo Modular	50
3.6	Controlador P	56
3.7	Controlador PI - Inicialização	56
3.8	Controlador PI - Controlo	57
3.9	Controlador PID - Inicialização	57
3.10	Controlador PID - Controlo	58
3.11	Criação de Interface de Controlo	59
3.12	Edição de Valores pela Interface de Controlo	59
3.13	Controlo PI com <i>Anti-windup</i>	73
3.14	Controlo PID com <i>Anti-windup</i>	74
4.1	Utilização de Lógica Difusa	86

Lista de Acrónimos

COM	<i>Communication Port</i>
FLC	<i>Fuzzy Logic Controller</i>
FOM	<i>First of Maxima</i>
IAE	<i>Integral Absolute Error</i>
IDE	<i>Integrated Development Environment</i>
ISE	<i>Integral Squared Error</i>
ISEP	Instituto Superior de Engenharia do Porto
ITAE	<i>Integral Time Absolute Error</i>
ITSE	<i>Integral Time Squared Error</i>
LOM	<i>Last of Maxima</i>
MOM	<i>Middle of Maxima</i>
PID	<i>Proportional-Integral-Derivative</i>
PWM	<i>Pulse Width Modulation</i>
SERT	Sistema Eletrónico de Regulação Térmica
SIF	Sistema de Inferência Difusa
SSE	<i>Sum of Squared Errors</i>
UI	<i>User Interface</i>

Capítulo 1

Introdução

Neste capítulo é introduzido o contexto do tema abordado, especificando o problema analisado, o plano de desenvolvimento de resultados e a organização do documento.

1.1 Definição do Problema

É usual encontrar processos de difusão de calor que apresentam atrasos significativos na resposta do sistema. O atraso diminui a estabilidade do sistema de controlo o que, conseqüentemente, dificulta a sua gestão. Por esta razão, as clássicas metodologias de controlo, de entre as quais se destacam os métodos de Ziegler-Nichols e Cohen-Coon, tanto em malha aberta como em malha fechada, nem sempre permitem a obtenção de valores para os parâmetros do controlador que reflitam um bom desempenho dos sistemas.

1.1.1 Objetivos

Esta dissertação tem por objetivo o controlo de um sistema de difusão de calor, baseado em algoritmos de inteligência artificial. Pretende-se analisar vários algoritmos de controlo, os quais permitam colmatar os problemas inerentes ao controlo de sistemas com atrasos significativos nas suas respostas temporais.

1.1.2 Resultados esperados

Após uma análise de vários métodos e de alguns algoritmos, é esperado um melhor desempenho do sistema de temperatura com os algoritmos de inteligência artificial. Com esses dados pretende-se comprovar a eficiência destes métodos e poder indicar as diferenças temporais entre os mesmos.

1.2 Plano de Trabalho

O plano de trabalho para o desenvolvimento da presente dissertação, segue a planificação representada no diagrama de Gantt da Figura 1.1.

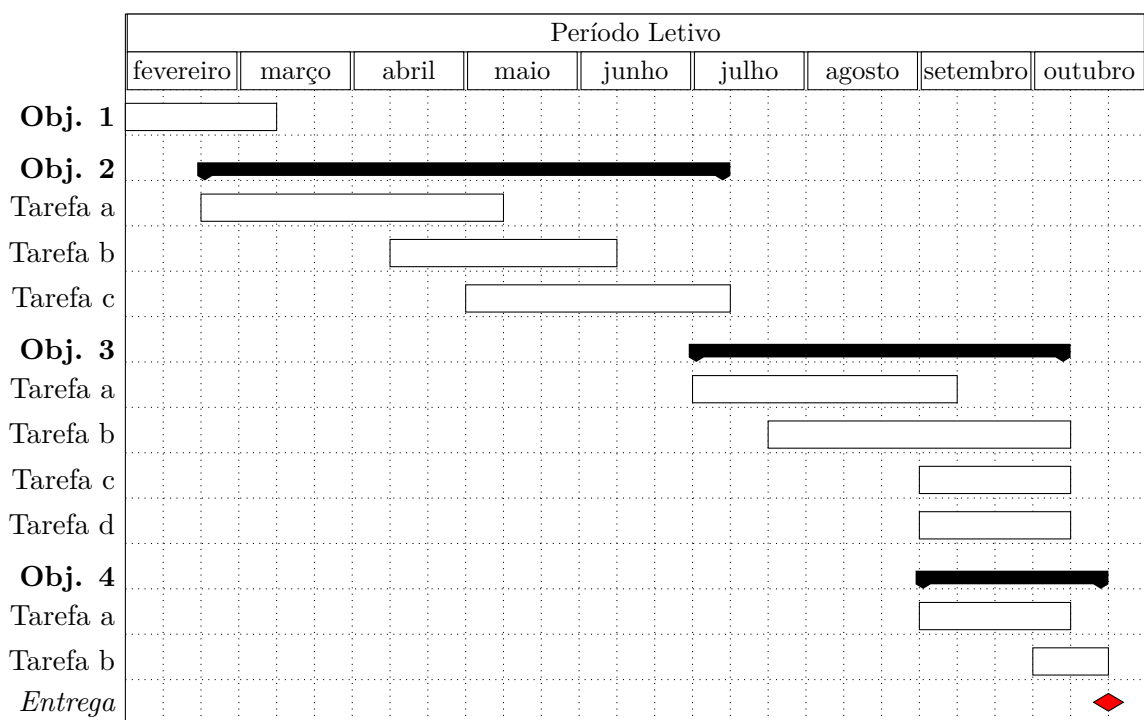


Figura 1.1: Diagrama de Gantt

O diagrama de Gantt é legendado pela seguinte lista de objetivos e tarefas:

1. Seleção do Tema
2. Estado da Arte
 - (a) Pesquisa Inicial
 - (b) Recolha e Análise de Documentos sobre o Tema
 - (c) Disposição e Organização de Dados

3. Desenvolvimento e Aplicação de Controlo

- (a) Análise de Métodos
- (b) Aplicação de Controlo
- (c) Testes de Algoritmos
- (d) Recolha de Dados

4. Dados e Conclusões

- (a) Análise e Tratamento de Dados
- (b) Conclusões

5. Entrega da Dissertação

1.3 Organização da Dissertação

O documento desta dissertação é composto por quatro capítulos.

O Capítulo 1, capítulo atual, é acompanhado pela contextualização do problema, os objetivos delineados e o planeamento de trabalho.

No Capítulo 2, é apresentado o estado da arte dos temas relevantes desta dissertação, a saber: *Fuzzy Logic*, a *toolbox* do MATLAB e aspetos de controlo de sistemas a ter em conta.

No Capítulo 3, é apresentado o sistema de difusão de calor, sendo destacadas as características do mesmo. São preparados e aplicados todos os métodos de controlo PID necessários para testar a reação do sistema, descrevendo os testes realizados e apresentam uma análise dos dados obtidos.

No Capítulo 4, são aplicados vários métodos de controlo por lógica difusa, apresentando os resultados dos mesmos. Além disso, é efetuada uma análise pormenorizada e comparativa dos métodos de controlo para um sistema de difusão de calor, demonstrado o melhor método de controlo obtido.

Por fim, é no Capítulo 5 que são apresentadas as conclusões do trabalho desenvolvido e as perspetivas de melhoria do controlo apresentado.

Capítulo 2

Lógica Difusa e Controlo de Sistemas

Neste capítulo é apresentada a informação e as definições sobre os temas envolventes à presente dissertação: "Lógica Difusa" e "Controlo de Sistemas".

2.1 Lógica Difusa

A lógica difusa é utilizada para a concretização de diversos objetivos, sendo um deles a criação de modelos de problemas ou situações onde pessoas tomam decisões num ambiente de elevada complexidade, havendo dificuldade em criar um modelo matemático [1].

Esta pode ser definida como a lógica precisa da imprecisão e aproximação [2]. Isto deve-se, ao contrário do modelo clássico de controlo binário, à utilização de valores relativos de verdade.

De um outro ponto de vista, esta lógica permite que informação qualitativa e imprecisa, seja apresentada de forma quantitativa. Com esta capacidade, é possível aplicar uma resolução prática e rigorosa com a manipulação qualitativa de informação, inicialmente expressa de forma linguística [3]. De forma a ilustrar esta representação linguística, na Figura 2.1 é possível verificar, num olhar cartoonista, a construção de uma máquina de café que percebe a imprecisão do mundo real (o gosto do café de cada pessoa) [4].

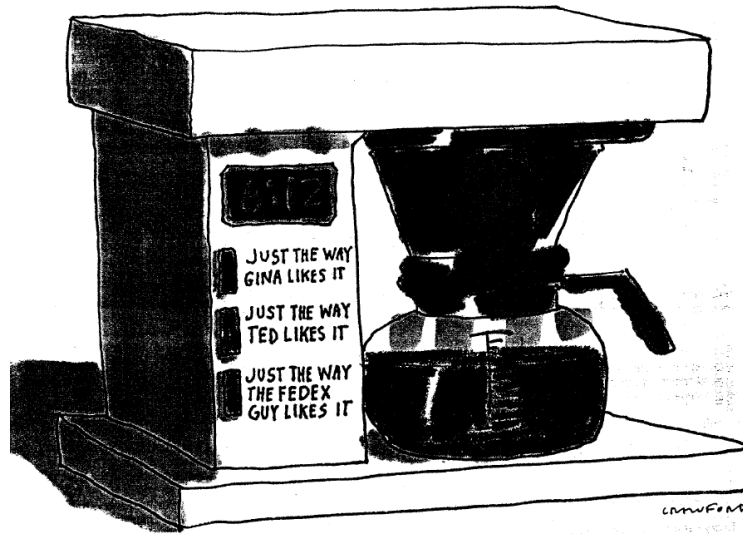


Figura 2.1: Ilustração da Lógica Difusa [4]

2.1.1 História

Apesar de, em 1920, ser proposto por Jan Lukasiewicz a *three-valued logic*, uma alternativa para a lógica clássica, a lógica difusa apareceu anos depois. Esta alternativa aceita a existência de três valores lógicos - verdade, falso e verdade indeterminada, onde o terceiro existe para valores de grau de pertença de, pelo menos, 0.5 [5].

A teoria inicial que veio a originar a conhecida lógica difusa, ao longo dos anos, tem sido desenvolvida e estudada por várias pessoas [5]. Alguns dos maiores contributos foram dados por:

- Bertrand Russel (1872-1970), filósofo e matemático britânico, tendo indicado que a lógica clássica leva a contradições, conduzindo a um estudo na imprecisão da linguagem, levando à conclusão que esta imprecisão é um grau importante;
- Max Black (1937), filósofo e físico quântico alemão, onde analisa o problema de modelar a imprecisão, dando os primeiros passos na teoria de *vague sets* ou, como hoje é denominado, teoria dos conjuntos difusos.

O professor Lofti A. Zadeh, matemático orientado a teorias de sistemas, inicialmente tinha a crença que para um sistema preciso, apenas seriam necessários dois valores lógicos (lógica clássica). No entanto, durante o desenvolvimento de um livro em 1961-1963, para sistemas altamente complexos, baseados em sistemas económicos e biológicos, chegou à conclusão que não é possível ser analisado com aproximações convencionais. Assim, o conceito de lógica difusa, apresentado em 1965 por Zadeh, contém os conceitos fundamentais para a sua compreensão, podendo usar valores de verdade entre zero e um, representado pelo grau de pertença [5, 6]. Passados três anos, em 1968, Lofti A. Zadeh publicou o artigo "Fuzzy Algorithms", com o propósito de introduzir o conceito básico que, através de lógica difusa em vez de usar

precisão, poderá resolver vários problemas relativos a processamento de informação, controlo, sistemas de identificação, inteligência artificial, entre outros [7, 8].

Já em 1974 é possível observar a primeira aplicação deste método de lógica em engenharia de controlo, por Mamdani e Assilan, na regulação de um motor a vapor. Nos anos seguintes, tendo esta lógica sido aprofundada e compreendida, iniciaram-se várias aplicações. A partir dos anos 80, o Japão envolveu-se em vários desenvolvimentos aplicando este método, sendo alguns deles o controlo de veículos (Laboratório de *Sugeno*, no Instituto de Tecnologia de Tóquio), a operação automática de comboios (Hitachi) e o controlo de estabilização (Laboratório de *Yamakawa*, na Universidade de *Kumamoto*) [6, 9, 10].

Com a crescente aplicação da lógica difusa nas diversas áreas, empresas de *softwares* de controlo, houve a necessidade da introdução de uma *toolbox* para a criação de sistemas difusos. Um dos exemplos é o MATLAB, que introduziu a "*The Fuzzy Logic Toolbox*" como seu componente em 1994 [6].

2.1.2 Princípios da Lógica Difusa

O início da lógica difusa é a possibilidade da verdade assumir qualquer valor do intervalo $[0,1]$. Com isto, esta pode ser vista como uma extensão da lógica multi-valor, que, como é possível observar na Figura 2.2, na lógica booleana, o grau de verdade apenas assume o valor de zero ou um (branco ou preto), enquanto que, na lógica multi-valor, existe uma variação incremental do grau de verdade [9, 11, 12].

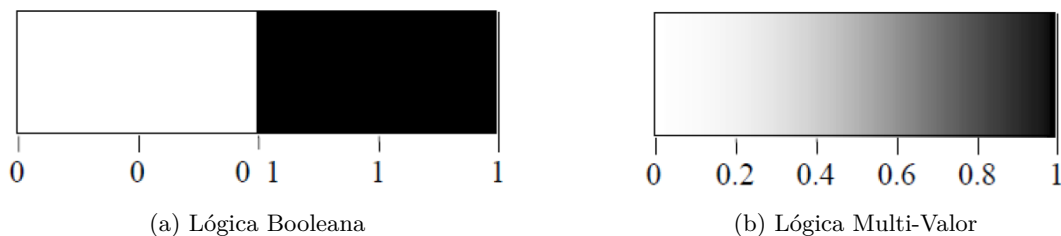


Figura 2.2: Classificação de Lógica [12]

No entanto, apesar de ser demonstrada a lógica multi-valor, a lógica difusa não é igual a esta, os seus usos e objetivos são diferentes. Nesta forma de lógica é utilizada a aproximação, ao invés da precisão, sendo que tudo, incluindo a verdade, é dada por um grau [9, 13].

Algumas das características que diferenciam esta lógica da tradicional são [9]:

- Num sistema tradicional booleano, uma proposição p pode ser falsa ou verdadeira. Assumindo um sistema multi-valor, p tem como possibilidades ser verdadeiro, falso ou ter um valor intermediário de verdade, sendo este um elemento finito ou infinito de verdade de um conjunto T . Enquanto isso, na lógica difusa, o valor de verdade poderá variar dentro dos subconjuntos difusos de T .

Neste caso, poderá haver uma caracterização imprecisa de um valor numérico de verdade, como "muito verdadeiro".

- Na lógica tradicional, tal como na multi-valor, permitem apenas dois quantificadores: "tudo" e "algum". Em contraste, a lógica difusa utiliza variados quantificadores: "muito", "vários", "pouco", "frequentemente", "ocasionalmente", entre outros. Com estes quantificadores é possível interpretar os valores difusos, oferecendo a avaliação imprecisa da caracterização do conjunto difuso avaliado.
- A lógica difusa permite dispor de métodos de representação para modificadores difusos e não-difusos, sendo alguns exemplos: "não", "mais ou menos", "extremamente", "um bocado", "ligeiramente", entre outros.
- Na lógica tradicional, a proposição p pode ser qualificada, principalmente, pela associação de valor de "verdade", podendo ser *verdadeira* ou *falsa*. De maneira complementar, pode-se introduzir os operadores modais, podendo estes ser de possibilidade, necessidade, crença ou conhecimento. A lógica difusa, considerando estes operadores, possui três principais modos de qualificação:

* *Qualificação de Verdade*, em que na constatação,

(A Daniela é nova) não é bem verdade,

a proposição qualificada é "A Daniela é nova" e o valor de verdade qualificativo é "não é bem verdade";

* *Qualificação de Probabilidade*, em que na constatação,

(A Daniela é nova) é improvável,

o qualificador difuso de probabilidade é "improvável";

* *Qualificação de Possibilidade*, em que na constatação,

(A Daniela é nova) é quase impossível,

o qualificador difuso de possibilidade é "quase impossível".

Num exemplo prático, a Figura 2.3 representa uma distribuição de idades, com a classificação de *novo*, *adulto* e *idoso*. Na distribuição clássica, sendo dado um conjunto convencional, é possível observar uma separação clara e imediata entre as três classificações de idade, não havendo sobreposição entre elas. Enquanto isso, assumindo que é aplicada à lógica difusa, o conjunto difuso poderá ter sobreposição de dois ou mais elementos do conjunto.

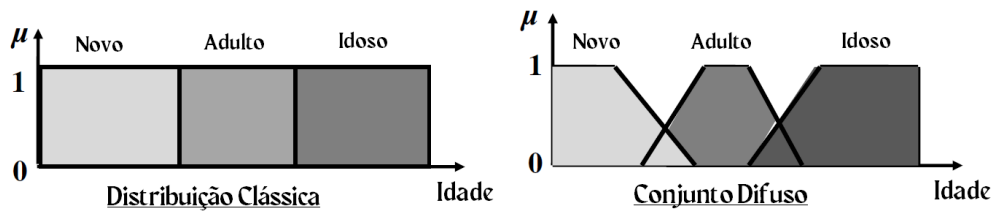


Figura 2.3: Distribuição de Idades em Lógica [2]

Esta sobreposição deve-se à imprecisão de classificação quando se atribui um rótulo relativamente à idade. Uma pessoa é considerada adulta a partir dos dezoito anos, no entanto, também pode ser considerada como nova. Seguindo a mesma lógica, uma pessoa adulta de sessenta anos, perante uma perspectiva de alguém muito novo, pode ser já considerado idoso. De forma a relativizar isso, cada classificação do conjunto das idades, terá atribuído um grau de pertença, μ . Assim, como observado na Figura 2.4, um exemplo usado meramente para explicação do conceito, será possível distinguir o grau de pertença de cada elemento ao seleccionar uma idade.

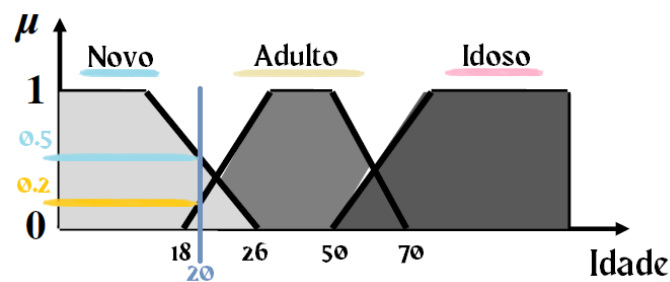


Figura 2.4: Classificação Difusa na Idade

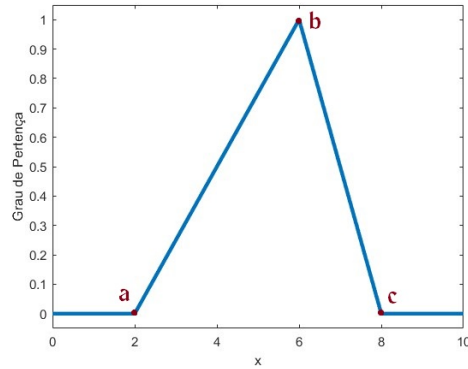
Neste caso, analisando uma pessoa que tenha vinte anos, com esta distribuição de classificação dos elementos difusos, terá $\mu_{NOVO} = 0.5$, $\mu_{ADULTO} = 0.2$ e $\mu_{IDOSO} = 0$. Esta pessoa poderia ser considerada, por esta classificação como "mais ou menos novo", "um pouco adulto" e até "nada idoso", podendo haver variações e outras hipóteses [2, 14].

2.1.3 Funções de Pertença e Variáveis Linguísticas

Ao avaliar conjuntos difusos, como o do exemplo anterior, podem ser consideradas várias funções de pertença, com diferentes formas, dependendo da perspectiva perante o objetivo final. Em cada cenário a avaliar, será necessário seleccionar o melhor método de representação, sendo as mais comuns de serem utilizadas as seguintes [13, 14, 15]:

- **Triangular**

A função de pertença triangular, representada na Figura 2.5, pode ser expressa pela equação (2.1).

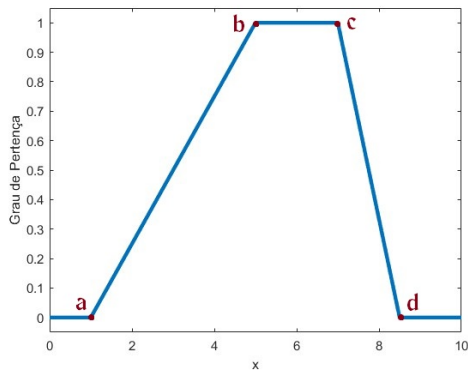


$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (2.1)$$

Figura 2.5: Função de Pertença - Triangular

- **Trapezoidal**

A função de pertença trapezoidal, representada na Figura 2.6, pode ser expressa pela equação (2.2).

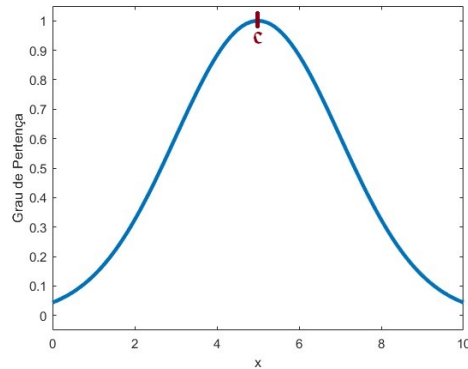


$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (2.2)$$

Figura 2.6: Função de Pertença - Trapezoidal

- **Gaussiana**

A função de pertença gaussiana, representada na Figura 2.7, pode ser expressa pela equação (2.3).

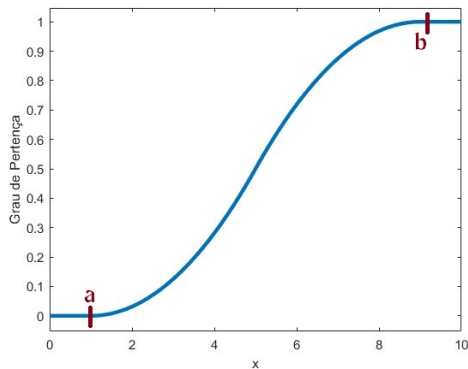


$$f(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (2.3)$$

Figura 2.7: Função de Pertença - Gaussiana

- **Forma S**

A função de pertença de forma S, representada na Figura 2.8, pode ser expressa pela equação (2.4).



$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases}$$

(2.4)

Figura 2.8: Função de Pertença - Forma S

- **Forma S Linear**

A função de pertença de forma S linear, representada na Figura 2.9, pode ser expressa pelas equações (2.5) e (2.6).

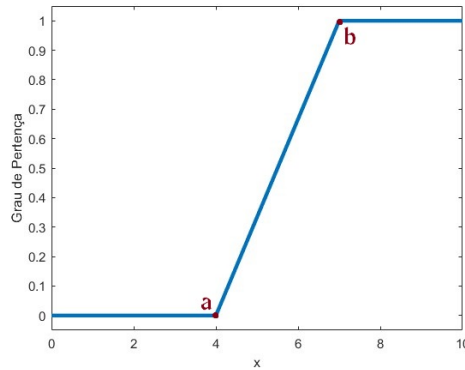


Figura 2.9: Função de Pertinência - Forma S Linear

– Se $a < b$:

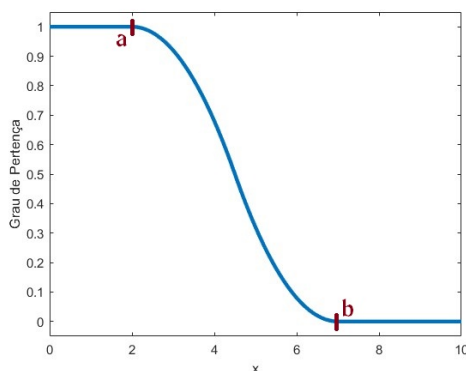
$$f(x; a, b) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases} \quad (2.5)$$

– Se $a = b$:

$$f(x; a, b) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases} \quad (2.6)$$

• Forma Z

A função de pertinência de forma Z, representada na Figura 2.10, pode ser expressa pela equação (2.7).



$$f(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2\left(\frac{x-b}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases} \quad (2.7)$$

Figura 2.10: Função de Pertinência - Forma Z

- **Forma Z Linear**

A função de pertinência de forma Z linear, representada na Figura 2.11, pode ser expressa pelas equações (2.8) e (2.9).

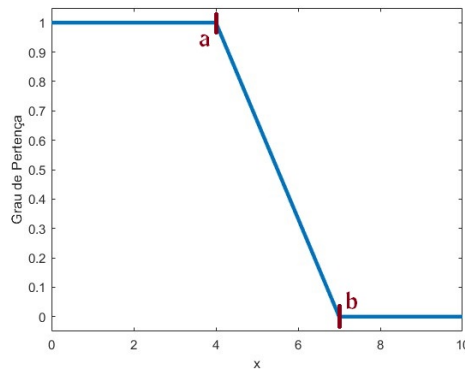


Figura 2.11: Função de Pertinência - Forma Z Linear

– Se $a < b$:

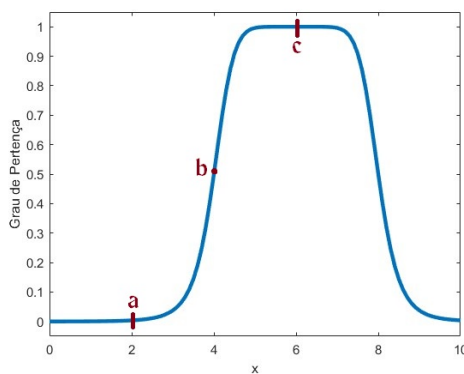
$$f(x; a, b) = \begin{cases} 1, & x < a \\ \frac{a-x}{a-b}, & a \leq x \leq b \\ 0, & x > b \end{cases} \quad (2.8)$$

– Se $a = b$:

$$f(x; a, b) = \begin{cases} 1, & x < a \\ 0, & x \geq a \end{cases} \quad (2.9)$$

- **Forma de Sino**

A função de pertinência de forma de Sino, representada na Figura 2.12, pode ser expressa pela equação (2.10).



$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (2.10)$$

Figura 2.12: Função de Pertinência - Forma de Sino

Para diversas aplicações de controlo, assumindo uma função de pertinência triangular, serão aplicadas, usualmente, de três a sete valores linguísticos [7]. Estes, como observado na Figura 2.13, poderão ser caracterizados nos espectros negativos (*NA* - Negativo Alto; *NM* - Negativo Médio; *NB* - Negativo Baixo), zero (*Z*) e nos positivos (*PB* - Positivo Baixo; *PM* - Positivo Médio; *PA* - Positivo Alto).

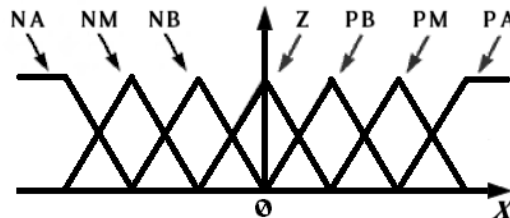


Figura 2.13: Valores Linguísticos numa Função de Pertinência [7]

Havendo esta filtragem de dados, usando o exemplo das idades da Figura 2.3 da subsecção anterior, é utilizada uma variável linguística a idade, tomando como valor: novo, adulto e idoso.

Uma variável linguística é, de uma forma simples, uma variável em linguagem corrente, cujo valor pode ser descrito de forma qualitativa, usando termos linguísticos, e quantitativamente, de forma a criar uma função de pertinência correspondente. Esta é importante devido a criar a interligação entre o termo linguístico, compreendido e utilizado na comunicação entre pessoas, e uma função de pertinência, útil para o processamento numérico de dados [15].

O conceito de variável linguística tem como base a sua utilidade, visto que utilizando este método é apenas necessário seleccionar um de três valores da variável idade para caracterizar uma pessoa, como por exemplo a classificação "novo", representada na Figura 2.14a. Enquanto isso, num método numérico, apresentado na Figura 2.14b, é mais simples e direto dizer que a pessoa tem 22 anos que usar a função anterior, pois tem que ser indicado um valor no meio de cerca de cem valores diferentes de idade. Este exemplo apresenta a compressão de dados que é possível fazer com este método, podendo ser denominado de granulação [7].

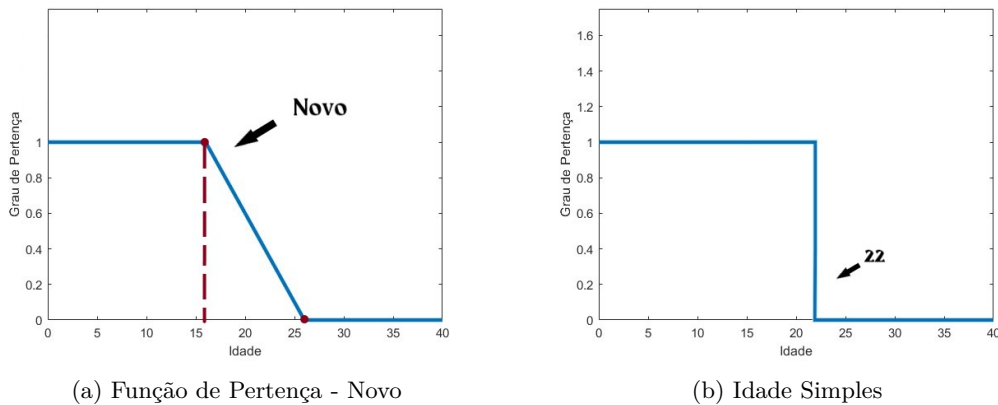


Figura 2.14: Compressão de dados e granulação

De modo a operar as funções de pertença adjudicadas aos vários valores das variáveis linguísticas, em similaridade às operações básicas de conjuntos convencionais - como a união, interseção e complemento -, existem operações básicas para conjuntos difusos, que permitem moldar as mesmas dependendo de cada situação [1, 16]. Assumindo dois conjuntos difusos, A e B , num universo X para um elemento x , temos:

- **União** ($A \cup B$) $\Rightarrow \mu_{A \cup B} = \max\{\mu_A(x), \mu_B(x)\}$
- **Interseção** ($A \cap B$) $\Rightarrow \mu_{A \cap B} = \min\{\mu_A(x), \mu_B(x)\}$
- **Complemento** (A') $\Rightarrow \mu_{A'}(x) = 1 - \mu_A(x)$
- **Igualdade** ($A = B$) $\Rightarrow \mu_A(x) = \mu_B(x)$
- **Inclusão** ($A \Rightarrow B$) $\Rightarrow \mu_A(x) \leq \mu_B(x)$
- **Produto** ($A.B$) $\Rightarrow \mu_{A.B} = \mu_A(x) \cdot \mu_B(x)$
- **Diferença** ($A - B = A \cap B'$) $\Rightarrow \mu_{A-B} = \min\{\mu_A(x), 1 - \mu_B(x)\}$

2.1.4 Regras Difusas

Um modelo é uma representação simplificada de aspetos de relevo de um sistema, permitindo uma melhor compreensão do seu comportamento. No caso da lógica difusa, para obter um modelo difuso é necessário criar um conjunto de regras difusas [17].

As regras difusas (muitas vezes denominadas de *fuzzy if-then rules*) são utilizadas para associar uma condição de uma variável linguística a uma conclusão. Sendo que servem para descrever, em termos linguísticos, a relação qualitativa entre duas ou mais variáveis, a aplicação destas regras permite a um sistema difuso captar conhecimento que envolve imprecisão [6, 15].

Cada uma destas regras assume a forma:

Se x é A , **Então** y é B .

Assumindo esta forma, "*Se...Então*" (*If...Then*), A e B são valores linguísticos definidos por conjuntos difusos do universo X e Y , respetivamente. Neste caso " x é A " é denominado como antecedente ou premissa, enquanto que " y é B " é dado como consequente ou conclusão [13].

Em alinhamento com a forma simples dada, as regras poderão ser moldadas perante a necessidade de classificação, variando o número de premissas e conclusões de cada uma.

De modo a exemplificar a formulação de regras, temos os seguintes cinco exemplos:

1. **Se** *pressão é alta*, **então** *volume é pequeno*.
2. **Se** *pressão é alta e temperatura é baixa*, **então** *volume é muito pequeno*.
3. **Se** *pressão é alta*, **então** *reduz volume ligeiramente*.
4. **Se** *pressão é alta*, **então** *volume é pequeno salvo se temperatura é alta*.
5. **Se** *pressão é alta*, **então** *usualmente volume é pequeno*.

Nestes casos, temos a *pressão*, *volume* e *temperatura* como variável linguística, enquanto que *pequeno* e *alta* são os seus valores linguísticos [7]. Apesar de seguirem a forma original, é necessário ter atenção a algumas variações das regras exemplificadas:

- Todas as regras representam dependências, exceto a (3), que apresenta uma atuação;
- Todas as regras são categóricas, exceto a (4) e (5), que são qualificativos (a (4) pela exceção e a (5) por generalização);
- A regra (2), ao contrário das restantes, apresenta dois antecedentes/premissas.

Uma das grandes vantagens da utilização deste método de regras é a capacidade de existir inferência apesar de uma condição ser apenas parcialmente satisfeita. O processamento de regras ou raciocínio difuso, oferecem o mecanismo necessário para a resposta de uma entrada de um sistema difuso ser computacional [6].

2.1.5 Sistema de Inferência Difusa

Os Sistema de Inferência Difusa (SIF), de forma a analisar problemas complexos onde o uso de lógica clássica é difícil ou impossível de aplicar, têm a sua estrutura orientada ao processamento de informação [14, 16].

Para executar estas operações, os SIF possuem uma estrutura como a representada na Figura 2.15 [14, 15]. Esta consiste em quatro principais componentes:

- **Módulo de Fuzificação;**
- **Base de Conhecimento** - Constituída pela Base de Dados e Base de Regras;
- **Mecanismo de Inferência** - Aplica o procedimento de inferência sobre as regras.
- **Módulo de Defuzificação.**

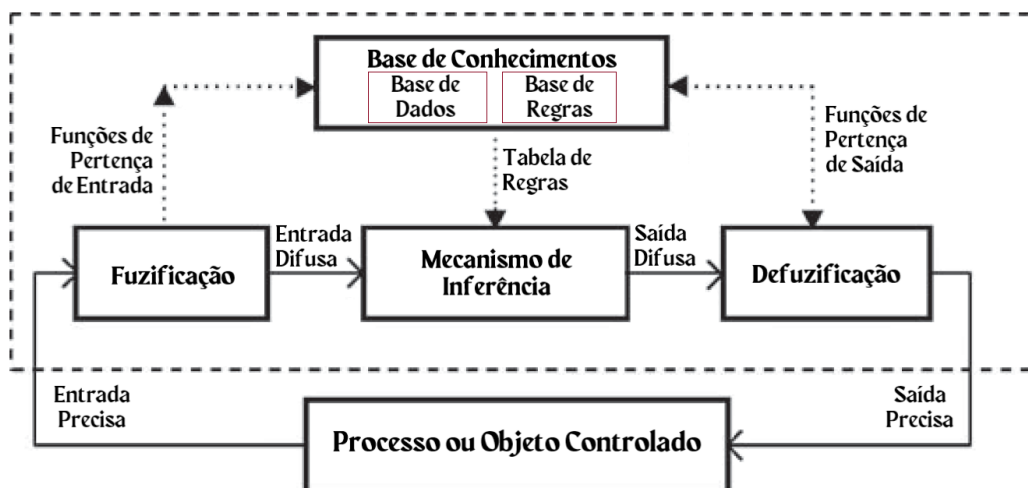


Figura 2.15: Estrutura do SIF

Para o funcionamento dos SIF, o mecanismo de inferência é considerado um componente essencial para o seu funcionamento. Assim, o método de operação deste mecanismo pode ser destacado pelos seguintes passos:

1. *Fuzificação dos valores de entrada*

Nesta operação associada ao módulo de fuzificação, pode-se dizer que as entradas de um SIF são, normalmente, um conjunto de valores precisos não preparados para mecanismos difusos. Assim, para cada entrada, tem que se determinar o grau de pertença perante o conjunto difuso apropriado, através das funções de pertença, $\mu(x)$.

2. *Aplicação operadores difusos nas premissas das regras difusas*

Após a fuzificação, esta operação é localizada no mecanismo de inferência com apoio da base de regras. Quando a premissa de uma determinada regra tem mais de um elemento, é necessário aplicar os operadores difusos apropriados - **e** , **ou** - , de modo que seja obtido um único valor a representar o resultado de toda a premissa.

3. *Avaliação das regras difusas*

As regras difusas são avaliadas por meio de implicação, através do mecanismo de inferência. Neste processo são aplicadas as entradas atribuídas nas premissas das regras, resultando num conjunto difuso para a saída.

4. *Agregação dos conjuntos difusos pelas várias regras*

Após a avaliação das regras, os conjuntos difusos individuais obtidos são combinados de forma a resultar num único conjunto difuso resultante. Este processo de agregação, com os dados obtidos, gera um conjunto difuso para cada variável de saída como resultado final.

5. *Defuzificação do resultado do conjunto difuso, de modo a obter o sinal de controlo*

O processo de defuzificação, envolvido pelo módulo de defuzificação, converte os conjuntos difusos resultantes num sinal de controlo, sendo o último passo do processo do SIF. Para efetuar este processo, existem vários **métodos de defuzificação** que podem ser aplicados:

(a) *Centro de Área/Gravidade*

Este método é um dos métodos de defuzificação mais conhecidos, em que determina o centro da área de funções de pertença combinadas, como exemplificado na Figura 2.16 [15].

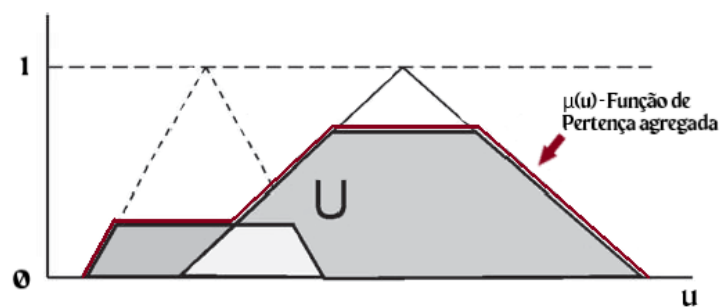


Figura 2.16: Método de Defuzificação - Centro de Área/Gravidade

De uma forma mais genérica, com os valores de sinal de controlo correspondentes aos graus de pertinência, $u_1/\mu(u_1), \dots, u_k/\mu(u_k)$, este método pode ser dado pela equação (2.11).

$$u^* = \frac{\sum_{i=1}^k u_i \times \mu(u_i)}{\sum_{i=1}^k \mu(u_i)} \quad (2.11)$$

(b) *Centro de Maior Área*

Este método é aplicado quando o centro U é não convexo, podendo acontecer quando um ou mais conjuntos difusos não se sobrepõem. Quando tal acontece, aplicando esta opção, é selecionado como ponto U o centro do conjunto de maior área, como observado na Figura 2.17 [15].

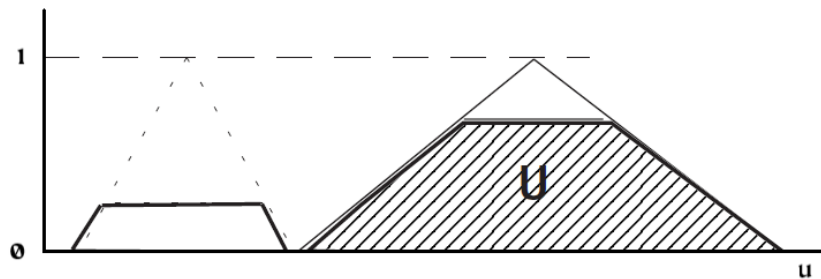


Figura 2.17: Método de Defuzificação - Centro de Maior Área

(c) *Menor/Meio/Maior do Máximo das Alturas*

Os métodos de Menor do Máximo das alturas (*First of Maxima* (FOM)), Meio do Máximo das Alturas (*Middle of Maxima* (MOM)) e Maior do Máximo das Alturas (*Last of Maxima* (LOM)), utilizam o domínio U , selecionando o valor menor, intermédio ou maior, respetivamente, do máximo grau de pertinência existente, como ilustrado nas Figuras 2.18 e 2.19 [15].

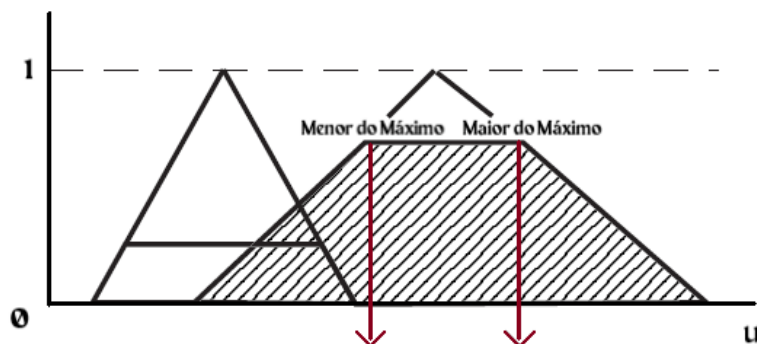


Figura 2.18: Método de Defuzificação - Menor/Maior do Máximo das Alturas

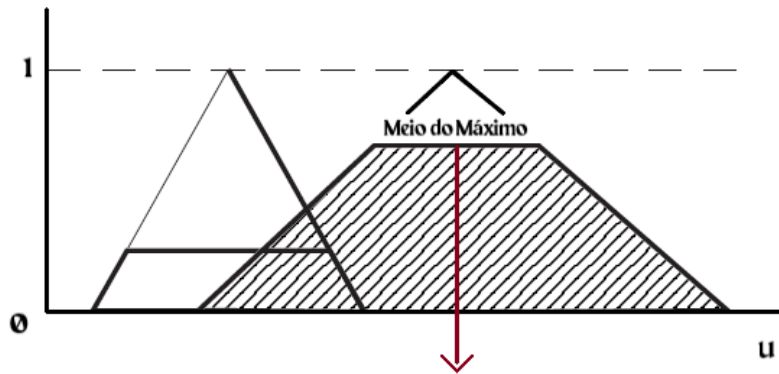


Figura 2.19: Método de Defuzzificação - Meio do Máximo das Alturas

(d) *Média dos Máximos*

Os métodos da Média dos Máximos calcula a média pesada dos máximos de cada subconjunto, sendo um dos mais eficientes, como ilustrado na Figura 2.20.

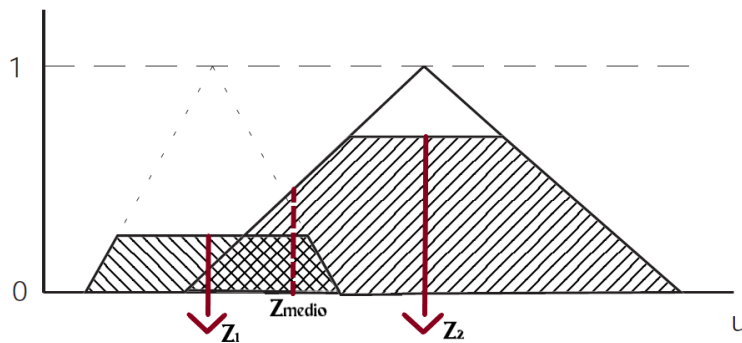


Figura 2.20: Método de Defuzzificação - Média dos Máximos

Todos estes métodos podem ser caracterizados por quatro propriedades, podendo ser utilizados para avaliar qual dos métodos é melhor mediante o objetivo:

- *Continuidade* - no processo de defuzzificação, esta propriedade indica que uma pequena alteração na entrada de um controlador difuso, não deve ter uma grande variação na saída do mesmo;
- *Desambiguidade* - a propriedade indica que o método selecionado, independentemente da entrada, tem uma resposta definida;
- *Plausibilidade* - indica que o valor que passa pela defuzzificação corresponde a um valor de pertença elevado do conjunto analisado;
- *Complexidade Computacional* - este critério é dos mais importantes para a aplicação prática de controladores difusos, de modo a ser selecionado o método mais compatível.

De modo a caracterizar cada um dos métodos, é apresentada na Tabela 2.1 uma comparação entre os mesmos, aplicando as propriedades referidas [15].

Tabela 2.1: Propriedades dos Métodos de Defuzificação

Métodos	a)	b)	c)	d)
Continuidade	Sim	Não	Não	Sim
Desambiguidade	Sim	Não	Sim	Sim
Plausibilidade	Sim	Sim	Não	Sim
Complexidade Computacional	Fraca	Fraca	Boa	Boa

Estes passos podem ser representados, de uma forma mais simplista, pela Figura 2.21, acompanhando o método de operação do mecanismo [1, 15, 16].

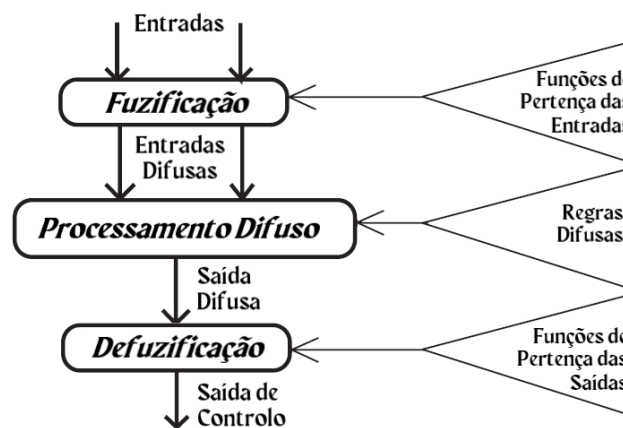


Figura 2.21: Operações de um SIF [15]

2.1.6 Tipos de Controlador

O sistema de inferência pode ser desenvolvido através de dois tipos de controladores diferentes:

- *Controlador de Mamdani*
- *Controlador de Sugeno*

O controlador de Mamdani segue a lógica do processo difuso exposta anteriormente. Num caso prático, observando a Figura 2.22, tem-se a percentagem de gorjeta a dar mediante o desempenho de um restaurante, assumindo a qualidade da comida e o serviço prestado, como entradas. Existe, inicialmente, a inserção das entradas difusas e a aplicação das operações presentes nas regras criadas. O resultado das regras, através de um método de implicação, origina um conjunto difuso para

cada uma delas. Com esses dados, existe uma agregação final para ser aplicado um método de defuzzificação, em que, neste caso, aplica-se o método de centro de área que origina o valor final de 16,7% [13, 16].

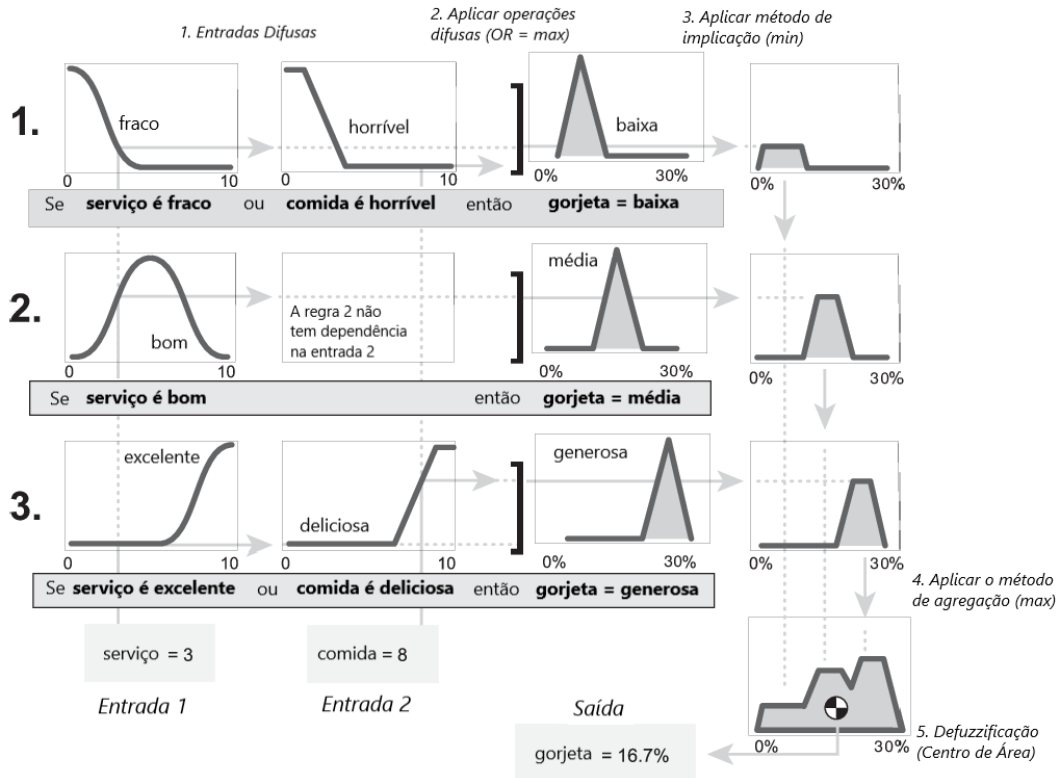


Figura 2.22: Exemplo com o Controlador de Mamdani [13]

Enquanto isso, o controlador de Sugeno, apesar de muito similar ao de Mamdani, tem uma diferença característica, que é a forma das regras e resultante valor de controlo. Neste controlador as regras são, genericamente, dados por:

$$\text{Se } x \text{ é } A \text{ e } y \text{ é } B, \text{ Então } z = f(x, y).$$

Ao contrário de Mamdani, o resultado de cada regra não origina um conjunto difuso, mas sim um valor em forma de equação linear, não necessitando do processo de defuzzificação. O valor final de controlo é dado pela soma de todos os resultados das regras e a sua média final. Através de um caso prático similar ao anterior, apresentado na Figura 2.23, é possível realçar as diferenças do processo. Inicialmente, tal como no controlador de Mamdani, existe a inserção das entradas difusas e a aplicação das operações presentes nas regras criadas, no entanto o resultado das regras são valores lineares. Com o resultado de cada regra é efetuada uma agregação dos valores, resultando no valor final de gorjeta, 16,3% [13, 14].

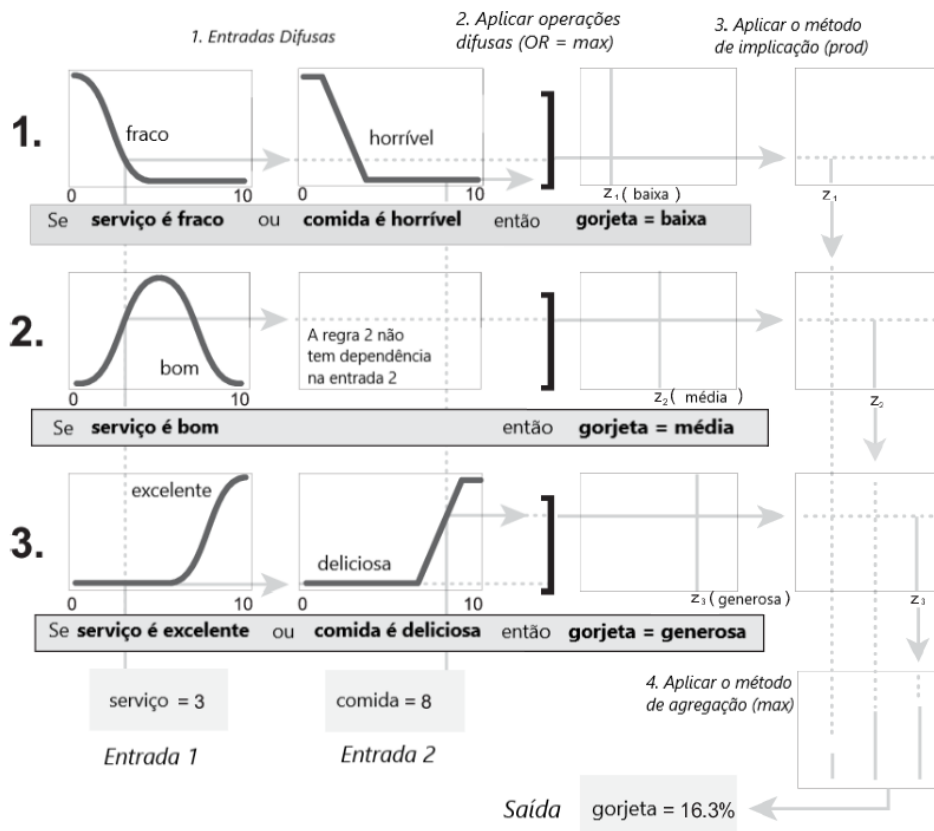


Figura 2.23: Exemplo com o Controlador de Sugeno [13]

Para se poder seleccionar o melhor método, é necessário averiguar o objetivo pretendido e qual dos dois tipos de controlador é mais adequado. De forma a ser mais fácil averiguar as vantagens de cada um, pode-se considerar a Tabela 2.2 [13].

Tabela 2.2: Vantagens do Controlador de Mamdani e Sugeno

Tipo de Controlador	Vantagens
<i>Mamdani</i>	<ul style="list-style-type: none"> • Intuitivo • Adequado para dados adquiridos por humanos • Base de regras mais compreensível • Grande aceitação pelos utilizadores
<i>Sugeno</i>	<ul style="list-style-type: none"> • Eficiente computacionalmente • Funciona bem em técnicas lineares, como controlo PID • Funciona com técnicas de otimização e adaptabilidade • Garante continuidade dos valores de controlo • Adequado para análises matemáticas

2.1.7 Vantagens e Aplicações

Em suma, algumas das vantagens e características de aplicar lógica difusa em sistemas controlo [10], são:

- Torna o sistema mais robusto por não necessitar de entradas precisas;
- Pode ser facilmente modificado, devido a ser baseado em regras;
- Possibilidade de controlo de sistemas não lineares sem a necessidade de modelos matemáticos;
- Em sistemas que usem micro controladores torna-se numa solução de simples aplicação;
- Em muitos casos, é uma solução mais rápida e barata de ser implementada.

As aplicações de lógica difusa em tecnologia são diversas, em que muitas delas estão presentes no nosso dia-a-dia, como:

- Ar Condicionado - antigamente eram definidos um valor mínimo e máximo de temperatura, em que ligava ou desligava consoante estava nos valores definidos, com a aplicação da lógica difusa é possível haver uma menor flutuação de temperatura e uma maior eficiência energética;
- Máquinas de Lavar - as máquinas, atualmente, aplicam lógica difusa de forma a regular a velocidade de rotação, duração dos ciclos e quantidade de água e detergente, mediante o peso total inserido nela. Além disso, algumas máquinas possuem sensores que detetam a coloração da água e permitem renová-la caso seja necessário.

Além destes dois exemplos, existem diferentes áreas que recorrem a este método de controlo:

- Aeroespacial
 - Controlo de altitude das aeronaves e satélites;
- Eletrónica
 - Controlo de humidade;
 - Micro-ondas;
- Medicina
 - Diagnósticos de suporte a médicos;
 - Controlo da pressão arterial durante anestésias;

- Diagnósticos de radiologia;
- Financeira
 - Previsão do mercado financeiro;
 - Gestão de fundos;
- Automóvel
 - Sistemas de autoestradas inteligentes;
 - Aumento da eficiência da transmissão automática;
 - Sistemas de controlo de velocidade.

É possível dizer que a lógica difusa está presente ao nosso redor [18, 19].

2.2 Toolbox do MATLAB

A aplicação da lógica difusa pode ser efetuada no *software* MATLAB, com recurso a uma *toolbox*. Recorrendo ao menu *Get More Apps*, a *toolbox* a procurar é denominada de *Fuzzy Logic Toolbox*, sendo apresentada na Figura 2.24.

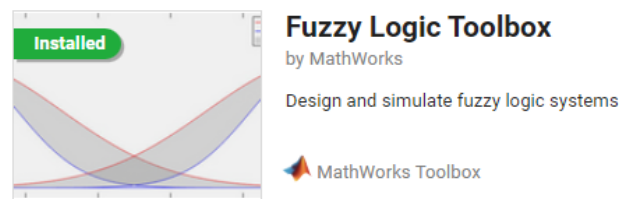


Figura 2.24: Instalar a *toolbox*

Ao ser instalada, a *app* representativa da *toolbox* é denominada de *Fuzzy Logic Designer*. Esta pode ser encontrada na zona de *Apps* do MATLAB, na secção de análise e *design* de sistemas de controlo, como demonstrado na Figura 2.25.

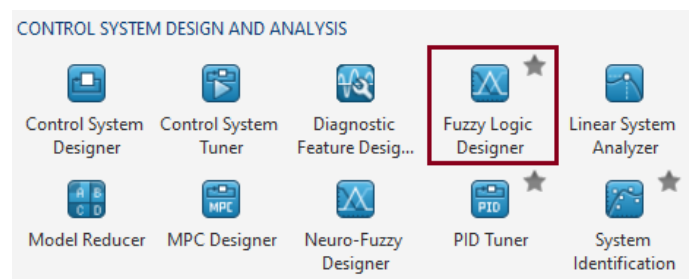


Figura 2.25: Aceder à *app* da *toolbox*

A *app*, ao ser iniciada, apresenta um menu (Figura 2.26) que permite aos utilizadores selecionarem um dos quatro *templates*: Mamdani tipo 1 ou tipo 2 e Sugeno tipo 1 ou tipo 2. Caso nenhum dos dois se adeque, é possível criar modelos customizáveis através de dados pré-existentes.

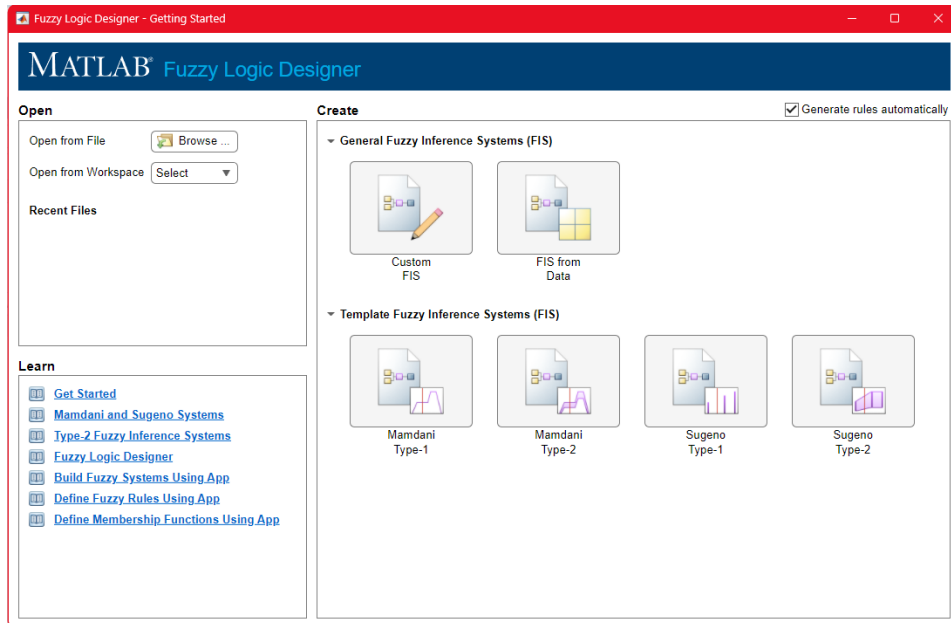


Figura 2.26: *App* - Menu inicial

Ao ser selecionado um controlador de Mamdani tipo 1 para exemplo, é dado o modelo genérico pré-criado da Figura 2.27, localizado na janela *Fuzzy Inference System (FIS) Plot*. Este apresenta nove regras na base de dados, duas entradas e uma saída, com três funções de pertinência em cada uma das entradas e saída.

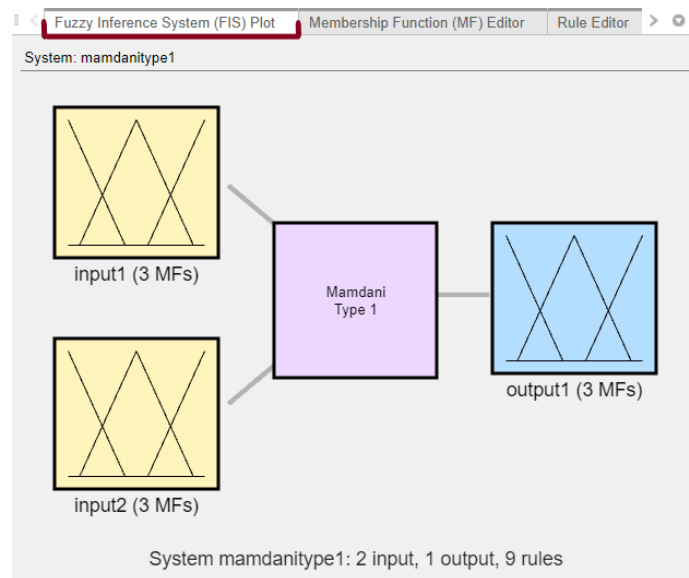


Figura 2.27: *App* - Vista geral do controlador de Mamdani

Caso seja pretendido configurar o SIF, a Figura 2.28 apresenta o menu que acompanha o controlador. É possível selecionar o modo de funcionamento dos vários métodos aplicados (*AND*, *OR*, implicação e agregação), bem como o método de defuzzificação pretendido.

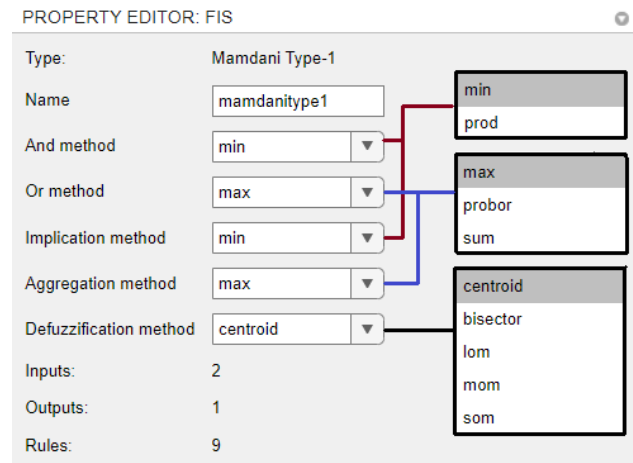


Figura 2.28: App - Propriedades do controlador

A janela *Membership Function (MF) Editor*, Figura 2.29, é utilizada para visualizar as funções de pertença atribuídas a cada uma das entradas e saídas do sistema. A distribuição e forma desta pode ser configurada na janela de propriedades que a acompanha, Figura 2.30, onde se destacam as opções do tipo de função de pertença disponibilizados e os parâmetros introduzidos.

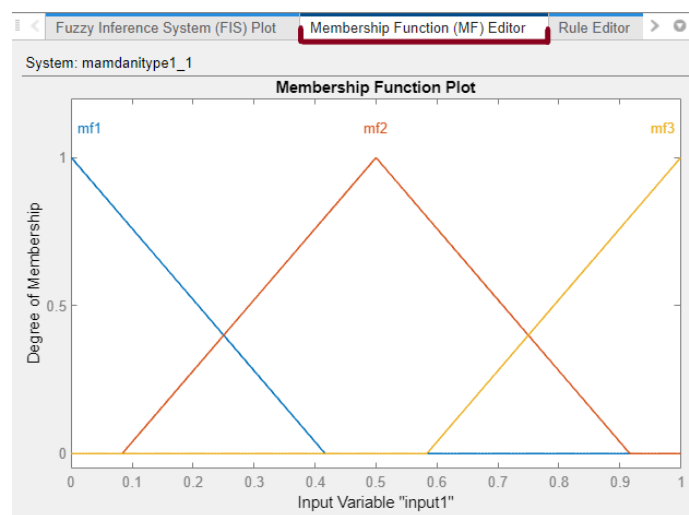


Figura 2.29: App - Gestão das entradas/saídas

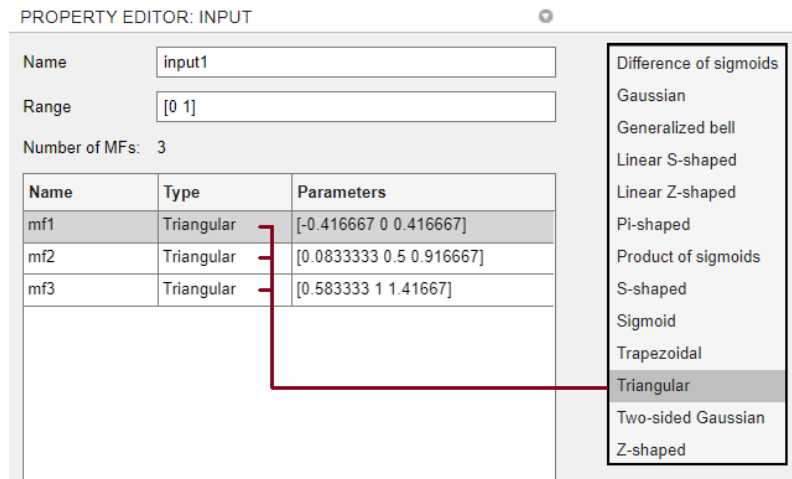


Figura 2.30: App - Edição das funções de pertença

Seguida destas, existe a janela *Rule Editor*, utilizada para gerir a base de regras do sistema. Na Figura 2.31 são apresentadas regras que interagem com todas as funções de pertença, com a indicação do seu respetivo peso. Na sua janela de configuração, Figura 2.32, ao seleccionar a regra pretendida, é possível ajustar a mesma. Na configuração é possível seleccionar o conector das regras (*And/Or*) e ajustar a função de pertença expectável para cada entrada do sistema, e a forma como este se deve comportar, atribuindo o resultado esperado na saída.

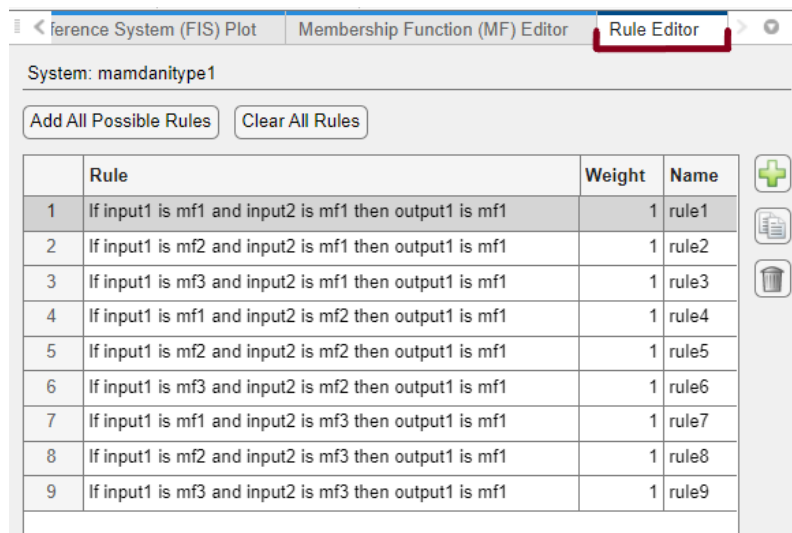


Figura 2.31: App - Gestão da base de regras

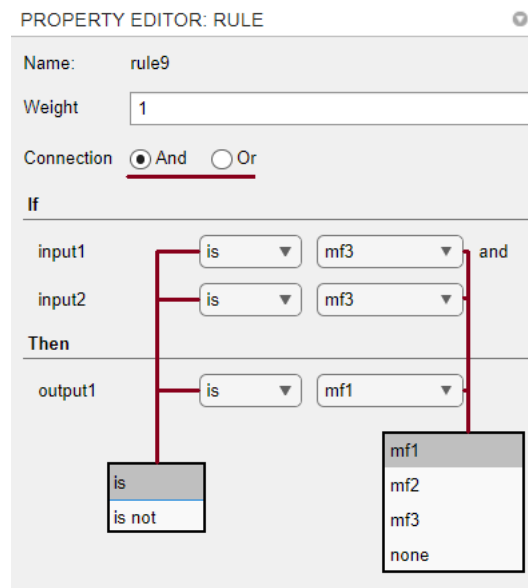


Figura 2.32: App - Edição de regras

Após a configuração do SIF, esta *app* possibilita a simulação da inferência de cada regra e uma representação do controle, no menu *Simulation* que se encontra no topo da aplicação, sendo representado na Figura 2.33.

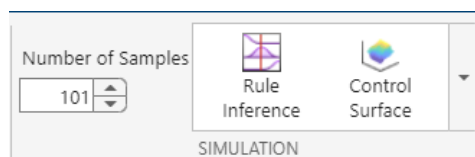


Figura 2.33: App - Menu de simulação

Perante a seleção da opção *Rule Inference* do menu de simulação, é apresentada a janela da Figura 2.34. Aqui são representadas todas as regras consoante um valor de entrada (entrada 1 e 2) e o seu valor de saída correspondente. Com a agregação dos valores de saída, é apresentado o conjunto de saída final.

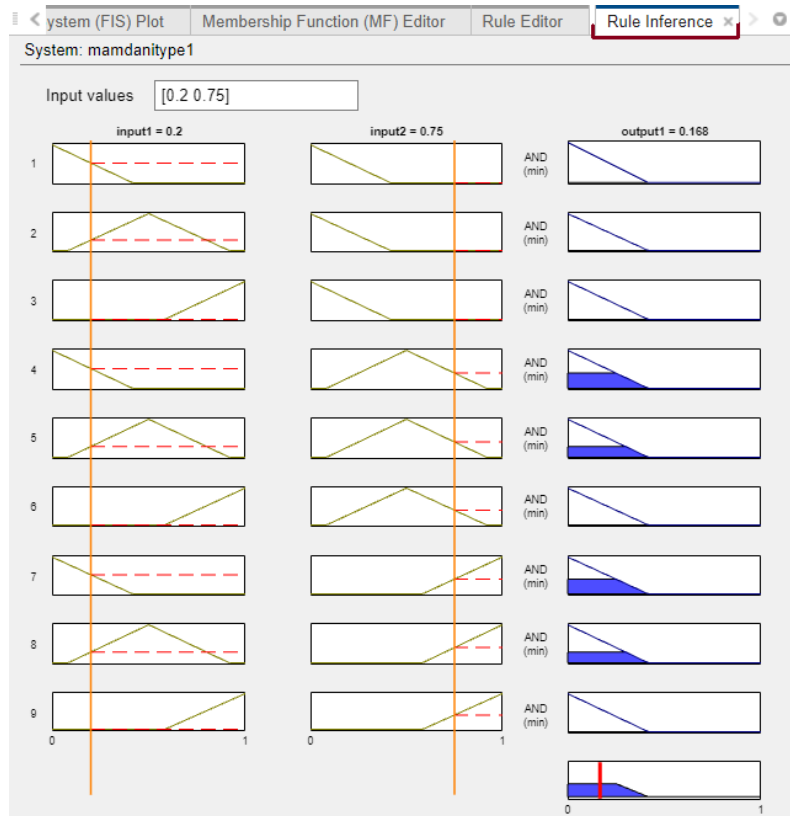


Figura 2.34: App - Processo de inferência difusa

A segunda opção do menu de simulação, *Control Surface*, permite visualizar num panorama tridimensional o resultado da variável de controlo, como apresentado na Figura 2.35.

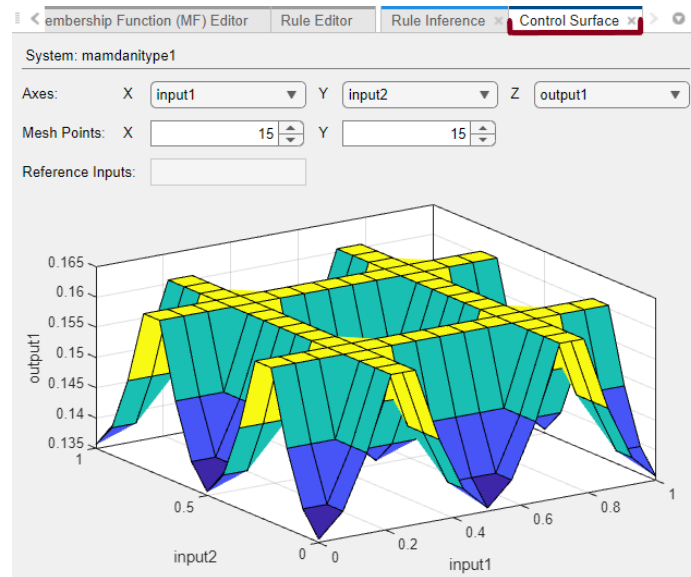


Figura 2.35: App - Valores de Controlo

A *app*, além do menu de simulação, apresenta a possibilidade de conversão de controladores do SIF, Figura 2.36. Neste menu, *Convert FIS*, é possível converter o controlador de Mamdani para Sugeno ou, uma aplicação não explorada nesta dissertação, a conversão de um sistema do tipo 1 para o tipo 2, e vice-versa.

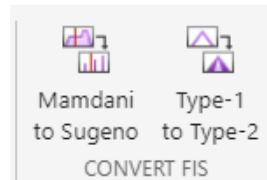


Figura 2.36: *App* - Conversão de controladores

Com esta ferramenta é possível criar o controlo de lógica difusa para diversos problemas, como o de controlo de sistemas de difusão de calor.

2.3 Sistemas de Controlo

Para efetuar o controlo de processos, ou seja, manter processos em condições e valores desejáveis perante a manipulação e ajuste de variáveis relativas ao mesmo, são necessários sistemas de controlo. Para tal ser possível, é necessário existir a comparação da variável a controlar com o valor de *setpoint*, onde a sua diferença dá-nos o erro:

$$Erro = Setpoint - (Valor da Variável Controlada)$$

Este valor aplica o conceito de informação do sistema, havendo comunicação do desvio do valor perante a referência, sendo denominado de controlo por *feedback*. Com este erro, o controlador fará os ajustes necessários à variável a controlar. Na Figura 2.37 é apresentado um sistema de controlo aplicado para processos genéricos, sendo, neste caso um sistema de controlo de malha fechada [20, 21].

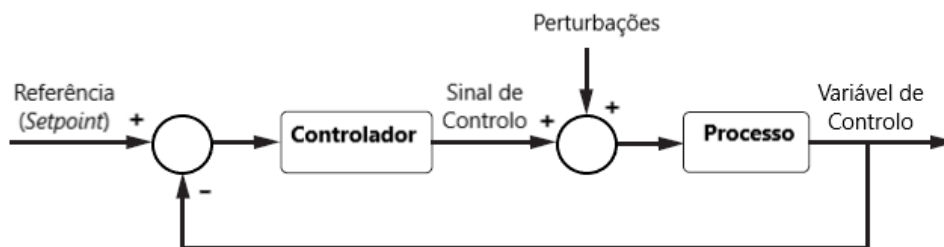


Figura 2.37: Sistema de Controlo Genérico

Para este tipo de controladores, como elemento final de controlo, podem ser aplicados métodos mais simples ou até complexos, dependendo da necessidade do utilizador. Os métodos mais simples são dados pelo controlo ON-OFF e controlo por

histerese. Estes dois métodos, representados nas Figuras 2.38 e 2.39, respetivamente, devido à sua simplicidade, têm uma implementação simples. No caso do método ON-OFF, este provoca rápidas alterações na saída do atuador, originando uma oscilação constante nas variáveis de saída. Para colmatar essa desvantagem, é possível aplicar o controlo por histerese que reduz a oscilação, mas não por completo [22].

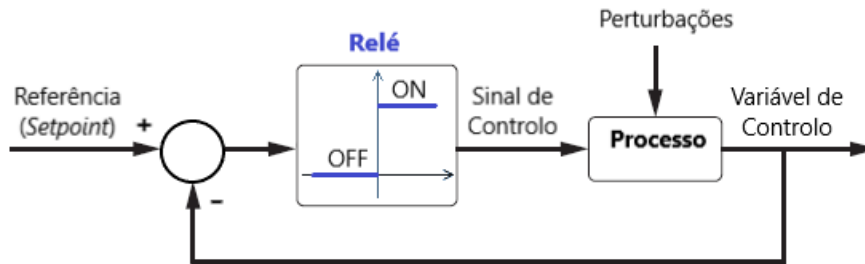


Figura 2.38: Controlo por Relé

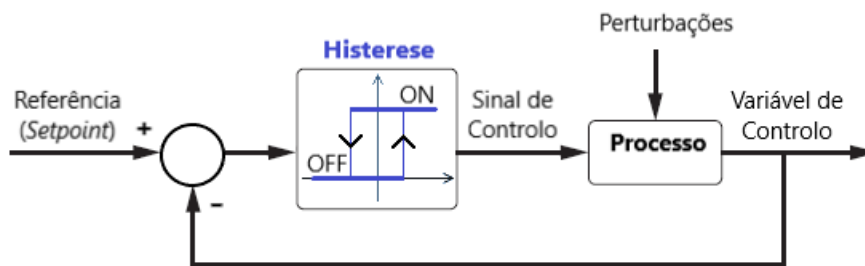


Figura 2.39: Controlo por Histerese

Para um controlo mais preciso e que permita a abordagem de processos mais complexos, normalmente a escolha mais eficiente são os controladores *Proportional-Integral-Derivative* (PID).

2.3.1 Controladores PID

Este tipo de controladores são os mais comuns na área da indústria, sendo a sua popularidade devida à sua robustez e variedade de funcionalidades. Este método de controlo, tal como o nome sugere, consiste em três parâmetros: P, proporcional; I, integral e D, derivativo. Através da atribuição, ajuste e otimização dos mesmos, é possível melhorar o controlo do processo desejado. Os controladores PID podem aparecer de diferentes formas, com controlos individuais (apenas proporcional (P), por exemplo), em pares (PI ou PI) ou os três em simultâneo [21, 23].

O controlo proporcional (P) permite obter uma resposta mais rápida que outros controladores. No entanto este poderá aumentar o valor de *overshoot* e, ao mesmo tempo, não ser suficiente para eliminar o erro em regime permanente. Para colmatar estas lacunas, por norma, são aplicadas outras formas de controlo, como o controlo

integral e controlo derivativo. Um controlador com apenas o controlo proporcional é dado pela equação (2.12).

$$u(t) = K_p e(t) \quad (2.12)$$

onde,

- $u(t)$: Saída do Controlador
- K_p : Ganho do Controlador Proporcional
- $e(t)$: Erro de Controlo

O controlador integral (I) é utilizado para remover qualquer desvio do valor de referência. Ao corrigir o *offset* do sistema, este destabiliza o sistema ao provocar *overshoot* na resposta. Este controlador relaciona a sua saída com o integral do erro, em ordem ao tempo, como demonstrado na equação (2.13).

$$u(t) = \frac{1}{T_i} \int e(t) dx \quad (2.13)$$

onde,

- $u(t)$: Saída do Controlador
- T_i : Constante de Tempo Integral
- $e(t)$: Erro de Controlo

O controlador derivativo (D), ao contrário dos dois anteriores, tenta "prever o futuro", antecipando as condições do processo pela análise da variação do erro. Com esta antecipação, o controlo derivativo permite diminuir a oscilação no sistema, normalmente aplicado para suavizar o impacto de controladores integrativos. Este controlador relaciona a sua saída com a derivada do erro, relativamente ao tempo, como apresentado na equação (2.14) [21, 22].

$$u(t) = T_d \frac{de(t)}{dt} \quad (2.14)$$

onde,

- $u(t)$: Saída do Controlador
- T_d : Constante de Tempo Derivativa
- $e(t)$: Erro de Controlo

De uma forma sumária, um controlador PID com estas três componentes, numa estrutura clássica, pode ser dado pela Figura 2.40.

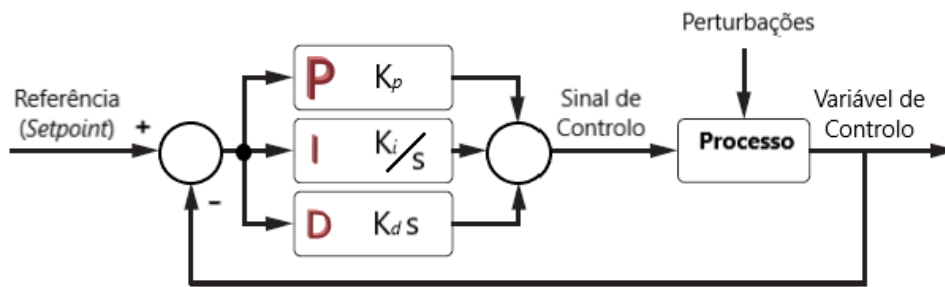


Figura 2.40: Sistema de Controlo com componentes PID

Sendo a expressão de controlo representada pela expressão:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (2.15)$$

Neste sistema, uma resposta tipo em malha fechada, pode ser dada pela Figura 2.41 [22, 23]. Perante um valor de referência (*setpoint*), teremos uma resposta do sistema (*process variable*) com quatro componentes importantes:

- **Tempo de Subida** (*Rise Time*): tempo que decorre até ao sistema se aproximar da sua referência pela primeira vez;
- (*Percent*) **Overshoot**: percentagem máxima de sinal que ultrapassa o valor de referência;
- **Tempo de Estabelecimento** (*Settling Time*): tempo que decorre até ao sistema estabilizar no seu valor final, com uma margem de variação de $\pm 1\%$ ou 5% do valor final;
- **Erro em Regime Permanente** (*Steady-State Error*): diferença entre o valor final em regime permanente e a sua referência.

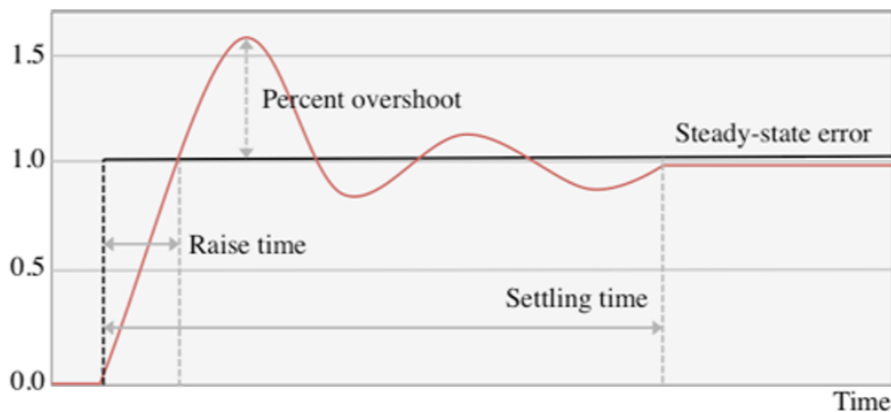


Figura 2.41: Resposta-tipo de um controlador PID [24]

Cada uma das componentes PID possui um ganho associado, representados na Figura 2.40, sendo estas:

- K_p - Ganho Proporcional;
- K_i - Ganho Integrativo;
- K_d - Ganho Derivativo.

Seguindo o sistema e expressão de controlo, dada pela equação (2.15), o ganho integrativo e derivativo pode ser calculado, com auxílio às suas constantes de tempo associadas, da seguinte forma:

$$\begin{aligned} K_i &= \frac{K_p}{T_i} \\ K_d &= K_p T_d \end{aligned} \quad (2.16)$$

O valor do ganho proporcional e das constantes de tempo podem ser obtidas por diferentes formas, dependendo do método de controlo seleccionado, sendo estes descritos posteriormente.

Com a variação e regulação dos ganhos de um controlador PID, consegue-se obter diversos tipos de resposta. Cada um dos ganhos (K_p , K_i e K_d) tem um efeito distinto nos quatro componentes referidos, havendo uma alteração na estabilidade do sistema. Para realçar as consequências destes, é apresentada a Tabela 2.3, indicando o efeito do aumento de cada um dos ganhos [21, 22].

Tabela 2.3: Efeitos dos ganhos PID

Ganho	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady-State Error</i>	Estabilidade do Controlo
$\uparrow K_p$	Diminui	Aumenta	Aumenta pouco	Diminui	Piora
$\uparrow K_i$	Diminui pouco	Aumenta	Aumenta	Anula	Piora
$\uparrow K_d$	Diminui pouco	Diminui	Diminui	Pouca variação	Melhora

Cada aplicação deste controlador implica uma observação do comportamento desejado, havendo o cuidado de saber em qual espaço de tempo será necessário prever e reagir, aplicando apenas as componentes necessárias. Na Tabela 2.4 é apresentado o espaço temporal estimado para cada tipo de controlador, quando é que este deve ser usado e exemplos de aplicações típicas [21].

Tabela 2.4: Diferentes controladores PID e suas aplicações

Controlador	Espaço Temporal	Quando Usar	Exemplos
P	Presente	Sistemas com resposta lenta e tolerância de <i>offset</i>	Termóstatos, Humidostato, Válvulas de Boia
I	Passado	Não são muito utilizados sozinhos, devido à sua resposta lenta	Sistemas com muito ruído
D	Futuro	Devido a não ter valor de referência e ser demasiado sensível ao ruído, não é utilizado sozinho	Nenhum
PI	Presente e Passado	Usado regularmente	Termóstatos, Controlo de Fluxo, Controlo de Pressão
PID	Todo o Espaço Temporal	Usado regularmente, sendo robusto mas um pouco sensível ao ruído	Aplicações onde o sistema possui inércia que pode sair do controlo

Para alcançar a sintonia do sistema é necessário ajustar os valores de ganho de cada componente presente no controlador utilizado [25, 26]. Para tal, podem ser aplicados diversos métodos, no entanto existem dois que se destacam:

- **Ziegler-Nichols**
- **Cohen-Coon**

Nesta dissertação, e tendo por base a documentação existente sobre este tema, aplicaram-se os métodos de sintonia do controlador Ziegler-Nichols em malha aberta e malha fechada e Cohen-Coon em malha aberta, por serem estes dos mais mencionados no controlo de sistemas.

2.3.2 Sintonia por Ziegler-Nichols

A sintonia pelas regras de Ziegler-Nichols é dos métodos mais conhecidos em sistemas de controlo, estando presente desde novembro de 1942.

Seguindo o conjunto de regras estipuladas, existe a possibilidade de aplicação de dois métodos distintos:

- Método em Malha Fechada
- Método em Malha Aberta

Dependendo do método selecionado, as regras sofrem algumas alterações.

A aplicação das regras de Ziegler-Nichols com o método de malha fechada envolve:

1. Aumentar o ganho proporcional do controlador até serem observadas oscilações periódicas, com uma amplitude constante, na saída do sistema, tornando-se um sistema criticamente estável;
2. O valor do ganho proporcional que permita alcançar a estabilidade corresponde ao ganho crítico, K_u ;
3. Com o sistema estabilizado, além do ganho crítico, é possível observar o período entre cada onda da oscilação presente, como demonstrado na Figura 2.42, sendo denominado de período crítico, P_u ;

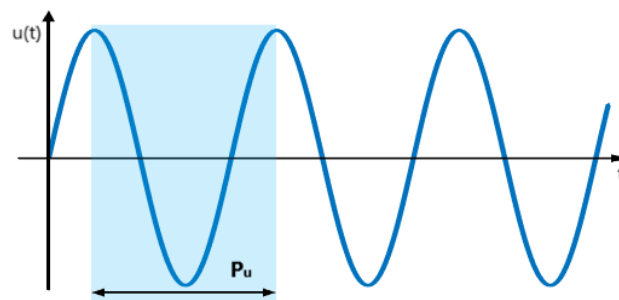


Figura 2.42: Período Crítico

4. Perante os valores obtidos, K_u e P_u , é necessário apenas verificar a Tabela 2.5 e, mediante o controlador desejado, calcular cada parâmetro.

Tabela 2.5: Cálculo dos parâmetros de controladores através de Ziegler-Nichols em malha fechada

Controlador	Ganho Proporcional (K_p)	Constante de Tempo Integrativa (T_i)	Constante de Tempo Derivativa (T_d)
P	$K_p = 0.5K_u$	—	—
PI	$K_p = 0.45K_u$	$T_i = P_u/1.2$	—
PID	$K_p = 0.6K_u$	$T_i = 0.5P_u$	$T_d = P_u/8$

Para ser melhor compreendido a regulação dos valores de ganho crítico, K_u , e período crítico, P_u , é apresentado um exemplo prático do processo na Figura 2.43 [21, 23, 25].

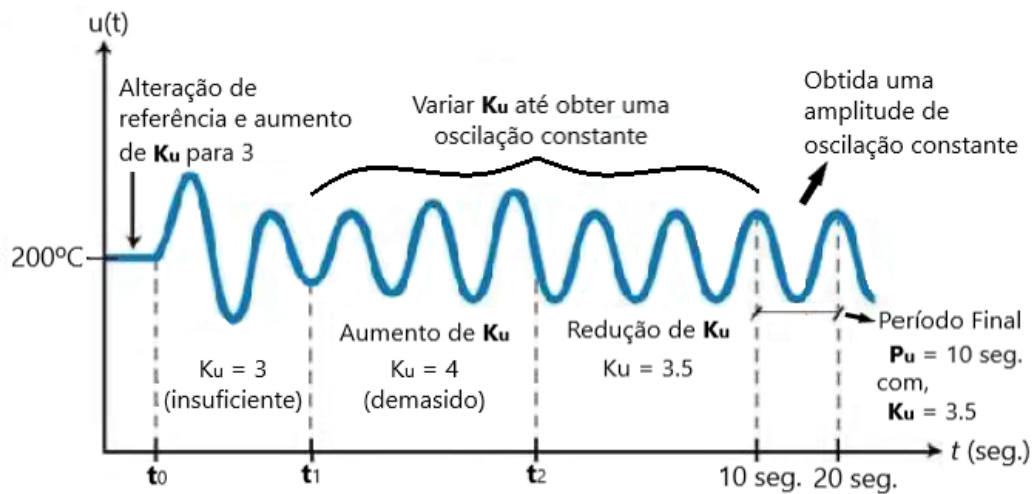


Figura 2.43: Ganho e Período Crítico com Ziegler-Nichols [21]

O método em malha fechada é vantajoso por apenas precisar de regular o parâmetro do controlador proporcional (P), tornando-se simples de testar, permitindo ter uma melhor percepção da resposta do sistema, podendo ter conhecimento do seu comportamento. No entanto, o teste de valores do parâmetro poder consumir algum tempo e levar o sistema a valores de saída não pressupostos, podendo danificá-lo [21].

Outra possível forma de obter os valores dos parâmetros de um sistema PID é pela aplicação do método de malha aberta, que envolve o teste da reação do sistema em, como o nome indica, malha aberta. Nesta situação é necessário existir a recolha da variável de saída, para posterior análise e aplicação das regras para obtenção dos parâmetros necessários ao controlador. Assim, o método em malha aberta envolve:

1. Efetuar um teste do sistema em malha aberta, como representado na Figura 2.44, aplicando um modelo de primeira ordem com atraso;

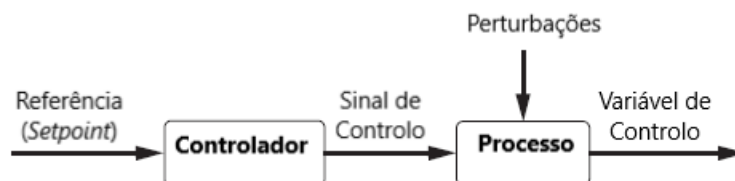


Figura 2.44: Sistema em Malha Aberta

2. Através da curva de resposta, pelo método da tangente apresentado na Figura 2.45, obter o ganho em regime permanente K . Além disso, utilizando o método de dois pontos, é obtida a constante de tempo do processo τ e o tempo de atraso τ_D . Estes valores são calculados pelas expressões,

$$K = \frac{\Delta y_{ss}}{\Delta u} \quad (2.17)$$

$$\tau = \frac{3}{2}(t_2 - t_1) \quad (2.18)$$

$$\tau_D = (t_2 - \tau) \quad (2.19)$$

onde Δy_{ss} é a variação da saída do processo em regime permanente e Δu é a variação do degrau na entrada do processo;

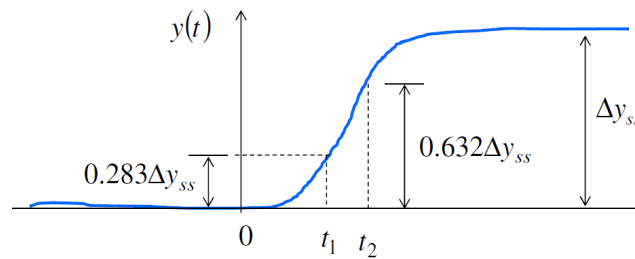


Figura 2.45: Teste do Sistema em Malha Aberta [27]

- Com os valores do ganho em regime permanente, K , do tempo de atraso do processo, τ_D , e da constante de tempo do processo, τ , é possível calcular os parâmetros do controlador desejado recorrendo à Tabela 2.6.

Tabela 2.6: Cálculo dos parâmetros de controladores através de Ziegler-Nichols em malha aberta

Controlador	Ganho Proporcional (K_p)	Constante de Tempo Integrativa (T_i)	Constante de Tempo Derivativa (T_d)
P	$K_p = \frac{\tau}{K\tau_D}$	—	—
PI	$K_p = \frac{0.9\tau}{K\tau_D}$	$T_i = \frac{\tau_D}{0.3}$	—
PID	$K_p = \frac{1.2\tau}{K\tau_D}$	$T_i = 2\tau_D$	$T_d = 0.5\tau_D$

O método em malha aberta tem como vantagens ser mais rápido e simples de aplicar que outros métodos, bem como ser robusto. Considerando as suas desvantagens, temos que os parâmetros I e D do controlador são meramente estimativas dadas através do parâmetro P, podendo levar a que as aproximações dos seus valores não sejam totalmente adequadas para todos os sistemas [21, 25, 27].

2.3.3 Sintonia por Cohen-Coon

As regras de sintonia de Cohen-Coon foram publicadas em 1953, sendo atualmente um dos processos mais conhecidos, logo a seguir a Ziegler-Nichols.

O método de Cohen-Coon permite corrigir a resposta lenta à estabilização do sistema, que se encontra presente com a aplicação do método de Ziegler-Nichols, no caso do tempo de atraso ser elevado. Uma forma simples de adjudicar o melhor método a aplicar é a comparação da constante de tempo e o tempo de atraso. As regras definidas por Ziegler-Nichols são adequadas para sistemas em que o tempo de atraso é inferior a metade da constante de tempo ($\tau_D < \tau/2$), enquanto que as regras de Cohen-Coon são adequadas para sistemas onde o tempo de atraso é inferior a duas vezes a constante de tempo ($\tau_D < 2\tau$).

Para se obter os parâmetros do controlador com as regras de Cohen-Coon, deve-se:

1. Efetuar um teste do sistema em malha aberta, aplicando um modelo de primeira ordem com atraso;
2. Através da curva de resposta, pelo método da tangente apresentado anteriormente, na Figura 2.45, obter o ganho em regime permanente K , a constante de tempo do processo τ e o tempo de atraso τ_D .
3. Com os valores obtidos, é possível calcular os parâmetros do controlador desejado recorrendo à Tabela 2.7.

Tabela 2.7: Cálculo dos parâmetros de controladores através de Cohen-Coon

Controlador	Ganho Proporcional (K_p)	Constante de Tempo Integrativa (T_i)	Constante de Tempo Derivativa (T_d)
P	$K_p = \frac{1.03}{K} \left(\frac{\tau}{\tau_D} + 0.34 \right)$	—	—
PI	$K_p = \frac{0.9}{K} \left(\frac{\tau}{\tau_D} + 0.092 \right)$	$T_i = 3.33\tau_D \frac{\tau + 0.092\tau_D}{\tau + 2.22\tau_D}$	—
PID	$K_p = \frac{1.35}{K} \left(\frac{\tau}{\tau_D} + 0.185 \right)$	$T_i = 2.5\tau_D \frac{\tau + 0.185\tau_D}{\tau + 0.611\tau_D}$	$T_d = 0.37\tau_D \frac{\tau}{\tau + 0.185\tau_D}$

Este método é adequado para sistemas com um tempo de atraso relevante, apresentando uma resposta rápida para sistemas em malha fechada, em contra-partida pode ser um pouco instável nesses sistemas [21, 25, 27].

2.4 Índices de Desempenho

O principal método de avaliação do desempenho do sistema, é o seguimento da soma dos quadrados dos erros - *Sum of Squared Errors* (SSE).

Este índice é dado por pela seguinte expressão,

$$SSE = \sum_{i=1}^N (y_i(\text{modelo}) - y_i(\text{processo}))^2 \quad (2.20)$$

onde N é o número de amostras da resposta do processo, $y_i(\text{modelo})$ é a amostra da resposta do modelo e $y_i(\text{processo})$ é a amostra da resposta do processo.

Quanto menor o valor do SSE, melhor vai ser a resposta do sistema [27].

Outros índices possíveis de aplicar são os do critério do integral do erro, que apresentam valores para minimizar o integral do erro em malha fechada. O seu valor pode variar mediante a entrada de referência e as perturbações do sistema, sendo necessário uma análise cuidadosa do sistema.

O critério do integral do erro apresenta quatro índices de desempenho:

- *Integral Absolute Error* (IAE)

$$I_{IAE} = \int_0^{\infty} |e(t)| dt \quad (2.21)$$

- *Integral Squared Error* (ISE)

$$I_{ISE} = \int_0^{\infty} e^2(t) dt \quad (2.22)$$

- *Integral Time Absolute Error* (ITAE)

$$I_{ITAE} = \int_0^{\infty} t |e(t)| dt \quad (2.23)$$

- *Integral Time Squared Error* (ITSE)

$$I_{ITSE} = \int_0^{\infty} t e^2(t) dt \quad (2.24)$$

Entre estes, um dos mais utilizados é o ITAE, pois beneficia parâmetros mais conservativos para o controlador, havendo um controlo mais moderado. Enquanto isso, o ISE apresenta melhores valores para parâmetros com uma resposta mais rápida ao erro existente. Para uma resposta entre estes dois índices, por norma, pode ser observado o IAE [25].

Para uma melhor análise do sistema parametrizado, deve-se fazer uma retrospectiva entre os diversos índices de desempenho e tirar conclusões.

Capítulo 3

Sistema SERT e Controlo Convencional

Neste capítulo é apresentado o sistema a controlar, referindo-se como o utilizar, sendo também descrita a aplicação de um controlo convencional.

3.1 Sistema de Controlo

Os sistemas de difusão de calor são comuns em várias aplicações, sendo escaláveis mediante a necessidade. Para ser possível analisar os métodos de controlo desejados, é utilizada uma plataforma laboratorial desenvolvida no ISEP, o Sistema Eletrónico de Regulação Térmica (SERT). Esta plataforma é dedicada à aplicação de métodos de modelação e controlo, sendo adequada para o tema desenvolvido.

O SERT, Figura 3.1, é constituído por um Arduino UNO (placa de desenvolvimento de microcontrolador, baseada no ATmega328P - Anexo A.1 [28]), acompanhado por um *shield* (placa que se encaixa ao Arduino para acrescentar funcionalidades de uma forma simples), que possui o processo térmico a controlar [29, 30].

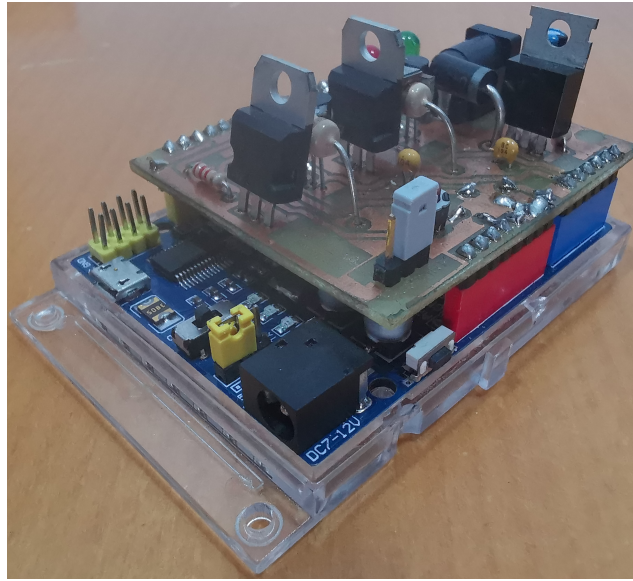


Figura 3.1: Sistema de Controlo - SERT

No SERT existem dois processos térmicos distintos, designando-os como sistema 1 e sistema 2, como ilustrado na Figura 3.2.

O sistema 1 tem uma resistência de aquecimento em contacto direto com um sensor de temperatura, encontrando-se apenas aplicada uma pasta térmica entre eles.

O sistema 2 tem uma resistência de aquecimento revestida por uma pequena placa de metal, causando uma maior dispersão da temperatura. O sensor de temperatura do sistema 2 encontra-se em contacto com a placa, mas devido à dispersão pelo objeto metálico, é expectável que seja necessário mais tempo para alcançar temperaturas elevadas e a perda de capacidade de atingir os valores de temperatura do sistema 1.

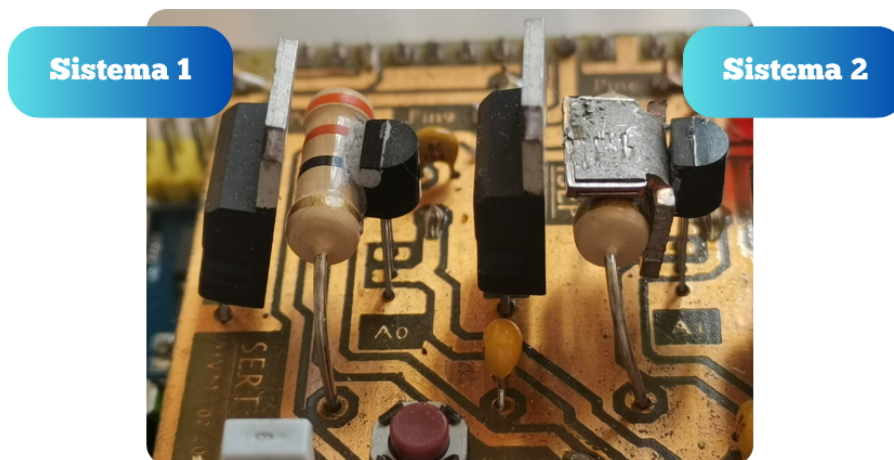


Figura 3.2: Sistema de Controlo - Identificação dos sistemas

3.1.1 Configuração do SERT

Para a configuração do SERT, inicialmente é necessário a instalação da *toolbox* "Legacy MATLAB and Simulink Support for Arduino", identificada na Figura 3.3, a qual permite que o MATLAB e o Simulink tenham funções de comunicação série.

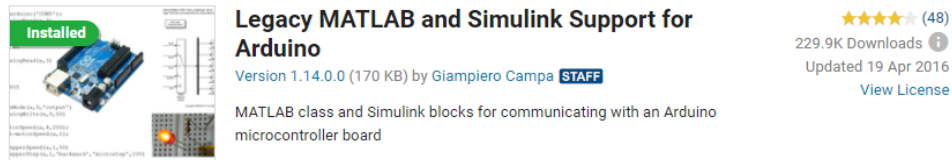


Figura 3.3: Toolbox para comunicação com o SERT

A *toolbox* instalada possui uma biblioteca própria para a comunicação, que deve ser aplicada no microcontrolador do SERT. Para tal, é necessário instalar a IDE do Arduino [28]. Com este *software* é possível fazer *upload* do programa que permitirá que o sistema se encontre permanentemente disponível, para receber comandos e retornar os resultados dos mesmos. Para o sistema utilizado, o SERT, é necessário utilizar o *adiones.pde*, que, como o nome indica, serve para aplicação de funções I/O analógicas e digitais ('*adio*'), ler *encoders* ('*e*') e ter interface com motores através de um *shield* ('*s*'). Na IDE, abrindo o programa *adiones.pde*, é necessário configurar a porta COM, onde o Arduino se encontra conectado, e pressionar o botão de *upload*, como destacado na Figura 3.4. Após este processo, o SERT encontra-se pronto para receber comandos e enviar dados para o MATLAB.

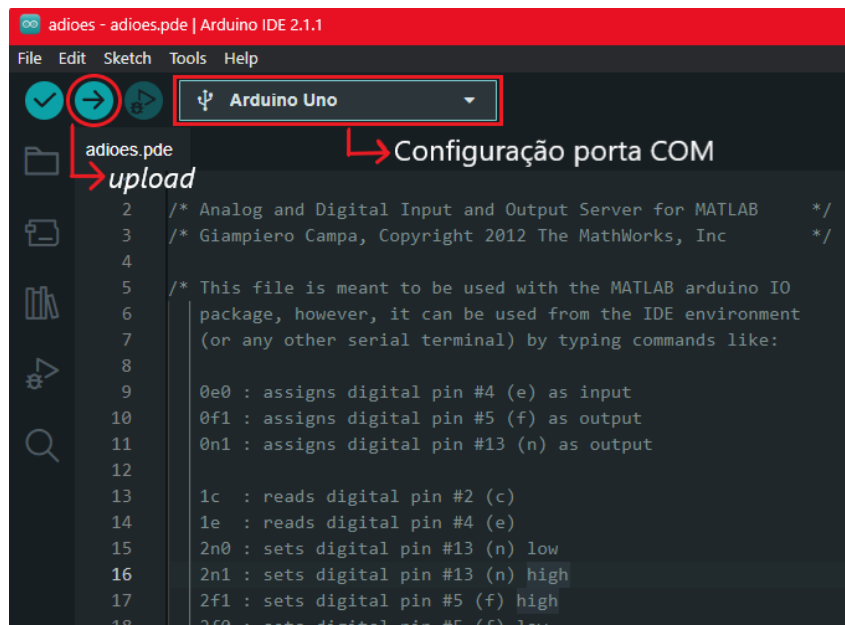


Figura 3.4: IDE Arduino

3.1.2 Teste inicial do SERT

Seguindo este processo, é feito um teste inicial de resposta ao sistema, havendo a necessidade de efetuar as conversões dos valores da entrada analógica do SERT para tensão e, de seguida, para temperatura mediante o sensor utilizado.

O Arduino possui um microcontrolador ATmega328P, o que permite conectar os sensores de temperatura em pinos analógicos e um conversor analógico puro digital, com uma resolução de 10 bits, como verificado no Anexo A.2 [31]. Assim, assumindo a tensão de referência de 5 V, é necessário calcular a tensão, em mV, presente no pino analógico:

$$U_{pino} \text{ (mV)} = \frac{ADC \times 5}{1023} \quad (3.1)$$

Para obter o valor em °C, é considerado o valor de tensão de referência do sensor TMP36, como indicado no Anexo A.3, o qual indica, para uma temperatura ambiente de 25 °C, a tensão de 750 mV [32]. Com o fator de escala de 10 mV/°C do sensor, sabe-se que para uma temperatura igual a 0 °C, o pino lê 250 mV. De forma a ser feita uma conversão direta, é subtraída à tensão de referência o valor correspondente aos 0 °C, podendo aplicar-se o fator de escala direto na equação de conversão:

$$T \text{ (}^\circ\text{C)} = \frac{U_{pino} - 500}{10} \quad (3.2)$$

Utilizando o MATLAB, é criado o código para comunicação entre o computador e o SERT. Com a Listagem 3.1, é identificada a porta COM em que se encontra conectado o SERT, bem como a definição dos pinos 10 e 9, como saídas, correspondente à conexão dos transístores do sistema 1 e do sistema 2, respetivamente, como indicado no Anexo B [33]. Estes permitem controlar a corrente transmitida à resistência de aquecimento do sistema.

```

1 % Conexao COM ao SERT - Verificar a porta COM utilizada
2 s = arduino('COM5');
3 % Pino de saida digital e analogico - pwm
4 s.pinMode(10, 'OUTPUT');    % PWM transistor 1
5 s.pinMode(9, 'OUTPUT');    % PWM transistor 2

```

Listagem 3.1: Teste do SERT - Comunicação com o SERT

Para ser efetuada a captação de todos os dados durante o teste do sistema, aplicando a Listagem 3.2, é definido: o *array* de dados para a temperatura, *Temp*; o tempo total de aquisição de dados, *t_end*, sendo 1200 segundos o valor definido

para o exemplo inicial; o período de amostragem, T , com 0.5 segundos; o número de amostras, N , que é dependente de t_{end} e T ; e a variável i que representa a amostra que se encontra presente em cada instante da experiência.

```

1 % Preparacao de Dados
2 Temp = [];           % guardar dados
3 t_end = 1200;       % tempo de aquisicao
4 T = 0.5;           % periodo de amostragem
5 N = round(t_end/T+1); % numero de amostras
6 i = 0;             % amostra atual

```

Listagem 3.2: Teste do SERT - Dados Iniciais

A ativação do transístor, que permite o aquecimento do sistema, tal como a leitura da temperatura, é dada pela Listagem 3.3. Inicialmente, é ativado o pino correspondente ao transístor, iniciando assim o aquecimento do sistema. Para existir o controlo do período de amostragem, é utilizado um ciclo *while*, sendo aplicadas no seu interior: as leituras de temperatura, as conversões anteriormente expostas e o registo destas no *array Temp*. De forma a ser visível a evolução da temperatura no sistema, é feito um *plot* num gráfico do tempo e da temperatura correspondente, sendo este processo repetido até alcançar o fim do tempo definido para a experiência.

```

1 % Ativacao de temperatura
2 digitalWrite(s,10,1); % ativa o transistor 1
3 % digitalWrite(s,9,1); % ativa o transistor 2
4 while i<N
5     tic
6     % Leitura
7     val = analogRead(s,0); % ler S1
8     % val = analogRead(s,1); % ler S2
9     U = val*(5000/1023); % tensao no pino de leitura
10    T_atual = (U - 500)*0.1; % temperatura atual
11    Temp = [Temp; i*T T_atual] % guardar valores
12    % Preparacao proxima leitura
13    i = i+1;
14    plot(Temp(:,1),Temp(:,2),'b') % grafico
15    hold on
16    drawnow
17    t = toc;
18    pause(max(0.01,T-t)); % pausa de 500 milisegundos
19 end

```

Listagem 3.3: Teste do SERT - Ativação de Temperatura

Concluída a experiência é necessário desativar o pino do transistor utilizado, garantindo assim que o sistema retorna ao estado inicial. Além disso, é essencial guardar os dados obtidos para análise posterior, caso seja necessário, e encerrar o porto de comunicação com o SERT, como apresentado na Listagem 3.4.

```

1 digitalWrite(s,10,0);    % coloca a OFF transistor 1
2 digitalWrite(s,9,0);    % coloca a OFF transistor 2
3 save dados_S1 Temp      % guardar dados S1 em ficheiro
4 % save dados_S2 Temp    % guardar dados S2 em ficheiro
5 delete(s)               % Conclusao de sessao

```

Listagem 3.4: Teste do SERT - Fim de Comunicação

A realização desta experiência, que coloca o SERT em esforço máximo durante o período de 1200 segundos, possibilita a observação dos seus limites. A Figura 3.5 retrata a resposta do sistema 1 (curva a azul), a resposta do sistema 2 (curva a cor-de-laranja) e um valor da temperatura de referência de 80 °C, a preto, utilizada para demonstrar a diferença no tempo de subida entre os dois sistemas.

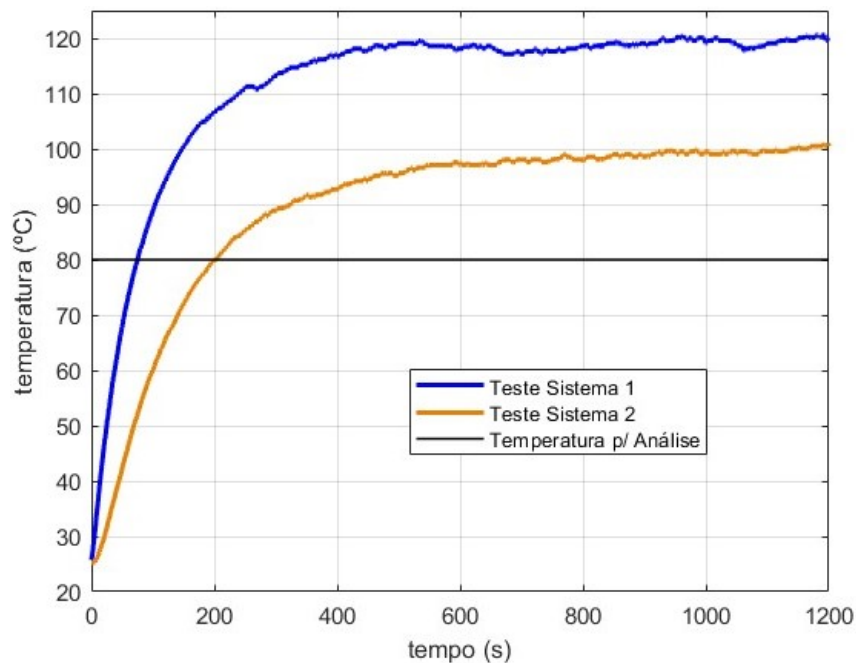


Figura 3.5: Teste do SERT - Gráfico de Temperatura

Além do gráfico ilustrado na figura anterior, são apresentados os valores de temperatura inicial, da temperatura de estabilização e o tempo que cada sistema demorou a alcançar a temperatura definida para esta análise, na Tabela 3.1.

Tabela 3.1: Teste do SERT - Resumo de Dados

<i>Dados</i>	Sistema 1	Sistema 2
Temperatura Inicial	25.758 °C	25.269 °C
Temperatura de Estabilização	119.488 °C	99.643 °C
Tempo de Alcançe a 80 °C	74.5 s	199.0 s

Com estas duas fontes de informação, pode ser observado o início do aquecimento com uma temperatura ambiente similar, aproximadamente 25.5 °C. Enquanto que o sistema 1 possui uma temperatura de estabilização de 119.5 °C, o sistema 2 apenas alcança os 99.6 °C, sendo visível pelo gráfico que o tempo de resposta do primeiro sistema é muito superior ao do segundo. Esta diferença pode ser realçada ao apresentar o tempo necessário para os sistemas alcançarem os 80 °C, onde o sistema 1 necessita apenas de 74.5 segundos e o sistema 2 de 199.0 segundos.

A razão pela qual se visualiza esta discrepância, apesar do controlo aplicado ser exatamente o mesmo, é facilmente observado pela Figura 3.2, apresentada anteriormente. Como expectável, devido à maior perda de calor no sistema 2, a sua resposta é mais lenta e com um menor alcance térmico que o sistema 1.

Assim, é possível confirmar o funcionamento do SERT, definir o método de comunicação e transmissão de dados, bem como as capacidades máximas de aquecimento do sistema. Um aspeto positivo das diferenças entre o sistema 1 e 2, é a capacidade de serem estudados dois comportamentos diferentes do controlo, devido à sua diferença de rapidez na atuação do sistema.

3.2 Modelação dos Sistemas

Para o controlo PID, como referido no subsecção 2.3, pode ser aplicado um dos dois principais métodos: Sintonia por Ziegler-Nichols ou Sintonia por Cohen-Coon. De modo a analisar qual destes seria a melhor opção, é aplicada uma resposta em degrau aos sistemas do SERT, validando o seu comportamento e parâmetros. O controlo do sistema aplicando uma resposta em degrau é obtido pelo diagrama apresentado na Figura 3.6, correspondendo à programação apresentada anteriormente na Listagem 3.3.

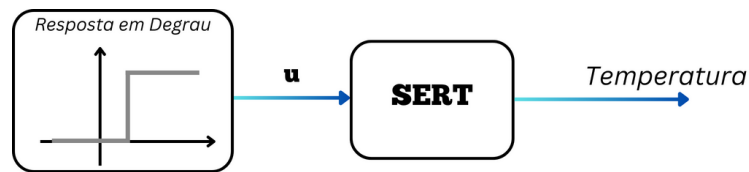


Figura 3.6: Diagrama - Resposta em Degrau

A melhor maneira de regular o controle do sistema é de forma modular, aplicando PWM nos transístores. Assim é criada a função $T1(duty)$ e $T2(duty)$, que permitem, ao inserir um valor de 0 a 100, regular o *duty cycle* do controle, como observado na Listagem 3.5. Nesta aplicação, é utilizado o valor de 50% do sinal de controle, não havendo necessidade de exigir que o sistema alcance a sua temperatura máxima, mantendo assim uma resposta sem esforço.

```

1 % Controle PWM por Duty Cycle dos Transistores
2 T1 = @(duty) analogWrite(s,10,round(max(0,min(100,duty))
   *255/100));
3 T2 = @(duty) analogWrite(s,9,round(max(0,min(100,duty))
   *255/100));

```

Listagem 3.5: Resposta em Degrau - Controle Modular

Com a adição do controle do sinal de forma modular, é iniciada a experiência da resposta em degrau para um período de 1500 segundos (25 minutos), de forma a assegurar que o sistema tem tempo de estabilizar. Além deste aumento do período de experiência, o tempo de amostragem é reduzido para 0.1 segundos para haver uma maior resolução na captação de valores durante a experiência. O resultado obtido da resposta em degrau dos dois sistemas é representado na Figura 3.7. É possível observar uma diferença de comportamento similar ao obtido na fase de testes do funcionamento do SERT.

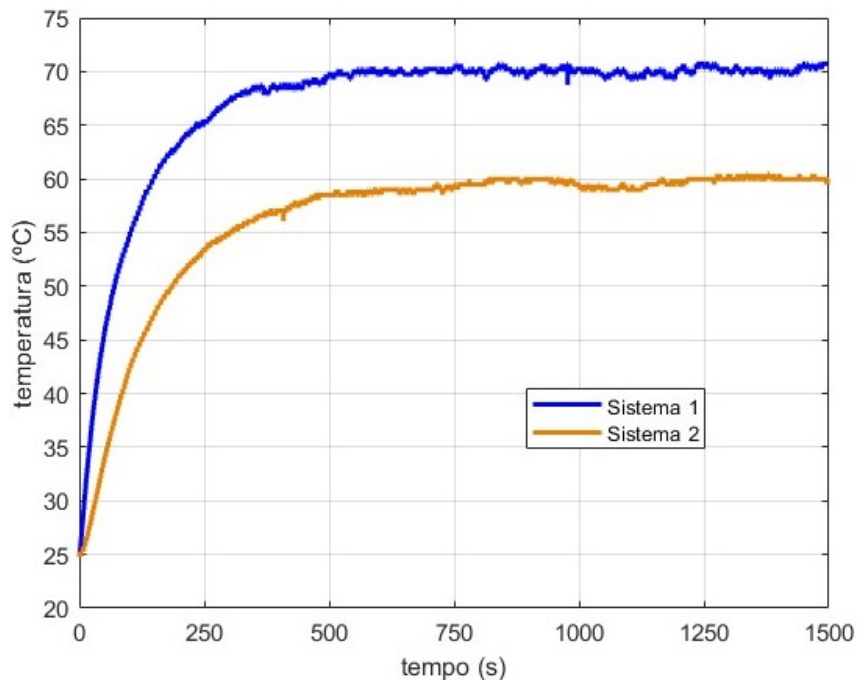


Figura 3.7: Resposta em Degrau dos Sistemas 1 e 2

Com os dados obtidos nesta experiência é possível fazer um tratamento dos mesmos, obtendo-se os parâmetros de cada sistema, para posterior aplicação na sintonia do controlo PID. A análise efetuada engloba os seguintes dados:

- **Temperatura Inicial** - T_0 ;
- **Temperatura de Estabilização** - T_{ss} - calculada tendo em consideração os últimos 220 segundos da experiência, de forma a validar a temperatura média do sistema;
- **Variação de Temperatura** - ΔT ;

$$\Delta T = T_{ss} - T_0 \quad (3.3)$$

- **Amplitude de Entrada** - ΔU - valor do controlo aplicado ao sistema;
- **Ganho DC** - K ;

$$K = \frac{\Delta T}{\Delta U} \quad (3.4)$$

- **Constante de Tempo** - τ - aplicando o método dos dois pontos, determinando t_1 (tempo para alcançar a temperatura correspondente a $0.283 \times \Delta T$) e t_2 (tempo para alcançar a temperatura correspondente a $0.632 \times \Delta T$) é possível determinar este valor;

$$\tau = \frac{3}{2} (t_2 - t_1) \quad (3.5)$$

- **Tempo de Atraso** - τ_D - obtido pela diferença entre t_2 e τ ;

$$\tau_D = t_2 - \tau \quad (3.6)$$

- **Resposta do Modelo** - y_m ,

– Para resposta do sistema usando um modelo de aproximação de 1^a ordem:

$$y_m(t) = K \times \Delta U (1 - e^{-t/\tau}) \quad (3.7)$$

– Para resposta do sistema usando um modelo de aproximação de 1^a ordem com atraso:

* Para $\tau > \tau_D$:

$$y_m(t) = K \times \Delta U (1 - e^{-(t-\tau_D)/\tau}) \quad (3.8)$$

* Para $\tau \leq \tau_D$:

$$y_m(t) = 0 \quad (3.9)$$

- **Erro em Regime Permanente** - SSE;

$$SSE = \sum_{i=1}^N (y_{modelo_i} - y_{experiencia_i})^2 \quad (3.10)$$

Na Tabela 3.2 são apresentados os parâmetros obtidos para o sistema 1. Entre os valores, é importante destacar o valor da constante de tempo, τ_D , onde o seu valor foi extraído por observação, pela Figura 3.8, devido ao método de interpolação de dois pontos atribuir um valor negativo, ou seja, classificando-o como sem atraso. Havendo uma resolução de captação de dados de 0.1 segundos, é possível comprovar a presença de um atraso de 0.6 segundos.

Tabela 3.2: Parâmetros do Sistema 1

Temperatura Inicial	T_0	24.780 °C
Temperatura de Estabilização	T_{ss}	70.217 °C
Variação de Temperatura	ΔT	45.437 °C
Amplitude de Entrada	ΔU	50
Ganho DC	K	0.9087
Constante de Tempo	τ	96.900 s
Tempo de Atraso	τ_D	0.600 s
Erro	SSE	830.921

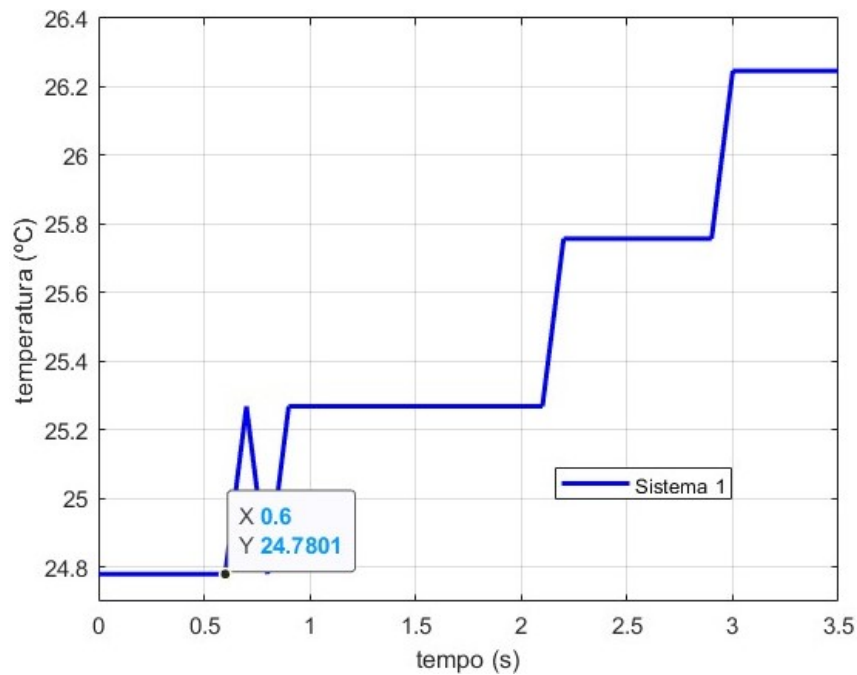


Figura 3.8: Constante de Atraso do Sistema 1

Conhecendo o comportamento do sistema e os seus parâmetros, é possível retratá-lo como um sistema de 1ª ordem com atraso. Criando o seu modelo de 1ª ordem (a vermelho na Figura 3.9) e fazendo a comparação com a experiência anteriormente efetuada (a azul), com o suporte da Figura 3.9, é possível verificar as semelhanças dos dois, apesar do erro em regime permanente ser relativamente elevado. Este valor pode ser justificado pelo longo período de experiência, onde pequenas variações causam um crescente aumento do erro, mediante a passagem do tempo, e a precisão do sensor do SERT.

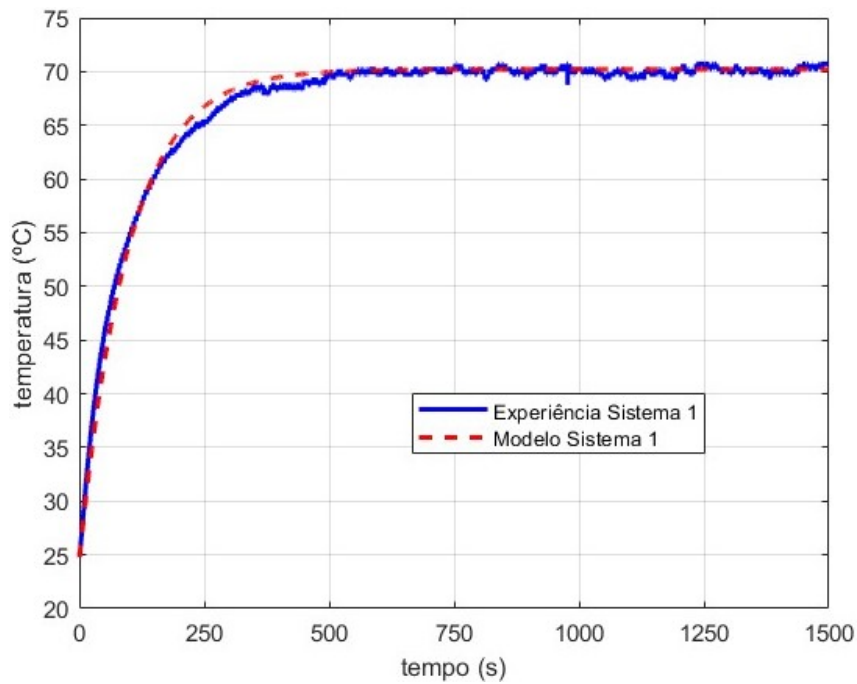


Figura 3.9: Experiência vs Modelo do Sistema 1

Na Tabela 3.3 são apresentados os parâmetros calculados para o sistema 2. Com a análise dos mesmos, pode-se dizer que nos primeiros 9.3 segundos de atuação do controle, o sistema apresenta poucos ou nulos sinais de aquecimento.

Tabela 3.3: Parâmetros do Sistema 2

Temperatura Inicial	T_0	24.780 °C
Temperatura de Estabilização	T_{ss}	59.972 °C
Variação de Temperatura	ΔT	35.193 °C
Amplitude de Entrada	ΔU	50
Ganho DC	K	0.7038
Constante de Tempo	τ	136.500 s
Tempo de Atraso	τ_D	9.300 s
Erro	SSE	4463.245

Tal como no sistema 1, este sistema pode ser considerado como um sistema de 1ª ordem com atraso. As similaridades do modelo (a vermelho) criado com a experiência (a laranja), são comparados e apresentados na Figura 3.10, onde se verifica, com menor impacto do que no sistema 1, pequenas variações, apresentado um erro em regime permanente menor.

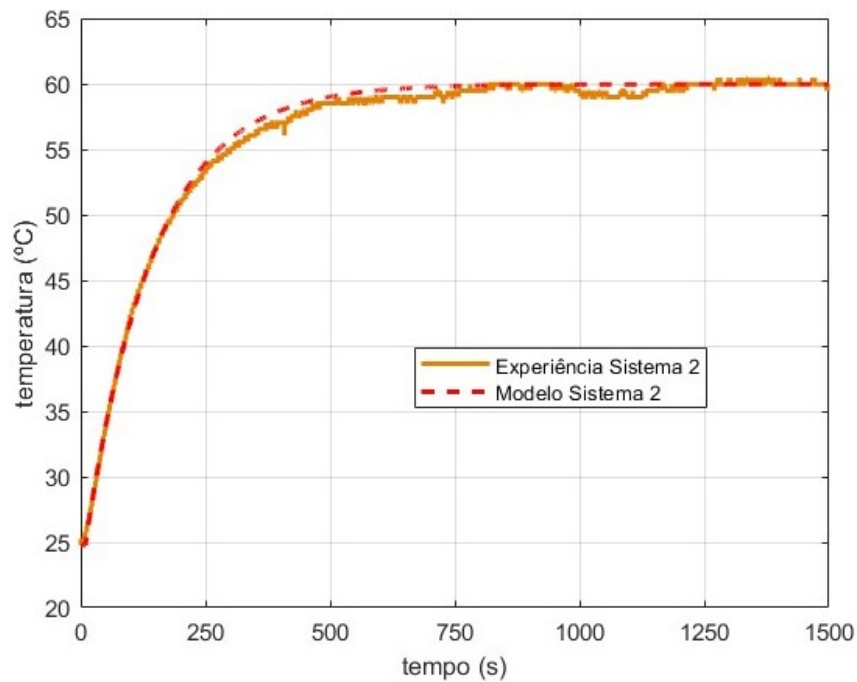


Figura 3.10: Experiência vs Modelo do Sistema 2

3.3 Controlo PID

Para a aplicação de um controlo PID, com base nos dados obtidos na resposta em degrau, os sistemas 1 e 2, sendo considerados de 1^a ordem com atraso, permitem um estudo diversificado, aplicando o método de sintonia do controlador por Ziegler-Nichols em malha fechada e malha aberta, bem como o método de sintonia por Cohen-Coon em malha aberta. Com esta abertura, é possível verificar qual dos três métodos apresenta melhores resultados. Estes três métodos de sintonia são selecionados por serem os mais referenciados na área de controlo.

Na aplicação dos vários métodos, de forma a verificar a melhor resposta possível, são testados três tipos de controlador:

- **Controlador P**

A aplicação do controlador proporcional é representada pelo diagrama da Figura 3.11.

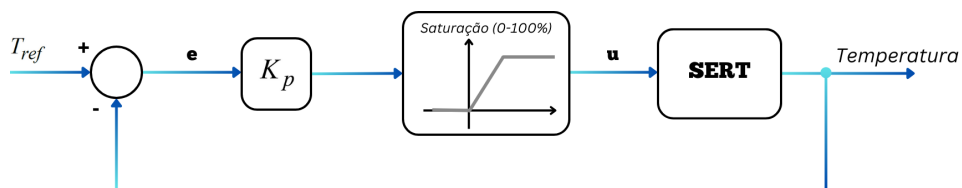


Figura 3.11: Diagrama - Controlo P

Para ser aplicado, é necessário ter em atenção o erro presente e a aplicação do parâmetro K_p , como se encontra ilustrado na Listagem 3.6.

```

1   e = Tref - T_atual; % erro
2   u = P_kp * e;      % controlo P
3
4   % Saturacao
5   if u>100
6       u = 100;
7   elseif u<0
8       u = 0;
9   end
10  T1(u); % ON

```

Listagem 3.6: Controlador P

• Controlador PI

A aplicação do controlador proporcional e integrativo é representada pelo diagrama da Figura 3.12.

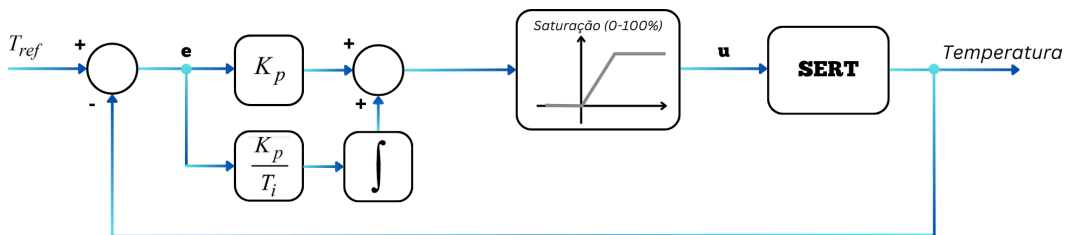


Figura 3.12: Diagrama - Controlo PI

Para ser aplicado, inicialmente, é calculado o ganho integrativo, K_i , utilizando os parâmetros obtidos em cada método, como se encontra presente na Listagem 3.7.

```

1   Ki = PI_kp * T / PI_ti; % ganho integrativo
2   I=0;                    % componente I

```

Listagem 3.7: Controlador PI - Inicialização

Durante o ciclo de controlo é adicionada a componente integrativa, no fim de cada ciclo e atualizada essa componente como consta na Listagem 3.8.

```

1   e = Tref - T_atual; % erro
2   P = PI_kp * e;     % controlo P
3   u = P + I;         % controlo PI
4
5   % Saturacao
6   if u>100
7       u = 100;
8   elseif u<0
9       u = 0;
10  end
11  T1(u); % ON
12
13  I = I + Ki * e;    % Atualizar o Controlo I

```

Listagem 3.8: Controlador PI - Controlo

• Controlador PID

A aplicação do controlador proporcional, integrativo e derivativo é representada pelo diagrama da Figura 3.13.

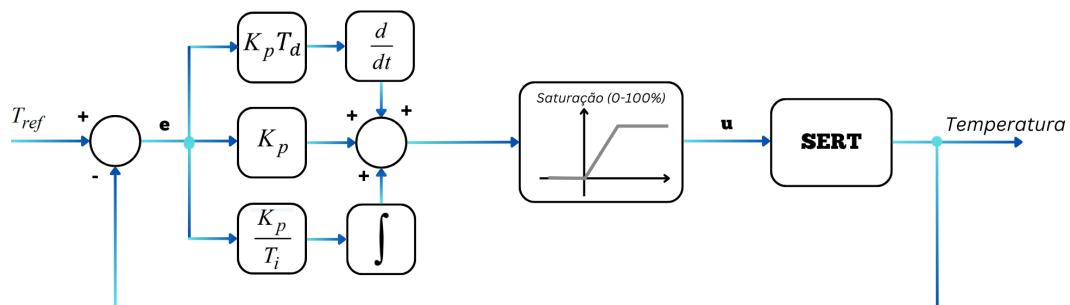


Figura 3.13: Diagrama - Controlo PID

Para ser aplicado, inicialmente, são calculados os ganhos integrativo e derivativo, K_i e K_d , aplicando os parâmetros obtidos em cada método, como se encontra presente na Listagem 3.9.

```

1   Ki = PID_kp * T / PID_ti; % ganho integrativo
2   I=0;                       % componente I
3   Kd = PID_kp * PID_td / T; % ganho derivativo
4   D = 0;                     % componente D
5   e0 = 0;                    % erro anterior

```

Listagem 3.9: Controlador PID - Inicialização

Durante o ciclo de controle são aplicadas as três componentes de controle, fazendo uma gestão do erro anterior para a componente derivativa, como ilustrado na Listagem 3.10.

```

1     e = Tref - T_atual; % erro
2     P = PID_kp * e;     % controle P
3     I = I + Ki * e;     % controle I
4     D = Kd * (e - e0); % controle D
5
6     u = P + I + D; % controle PID
7
8     % Saturacao
9     if u>100
10        u = 100;
11    elseif u<0
12        u = 0;
13    end
14    T1(u); % ON
15
16    e0 = e; % Atualizar erro anterior

```

Listagem 3.10: Controlador PID - Controle

Nos três tipos de controlador, como é possível observar nos diagramas e listagens, é aplicado um bloco de saturação, o qual limita a variável de controle de 0 a 100%. Sendo aplicado um controle por *duty cycle*, pelo que qualquer valor fora dos limites estabelecidos não será válido.

3.3.1 Sistema 1 - Ziegler-Nichols em malha fechada

O método de sintonia por Ziegler-Nichols em malha fechada, tem como requisito inicial obter o valor do ganho crítico, onde o sistema 1 entra em oscilação constante. Aplicando um diagrama em malha fechada com um ganho K_p , representado anteriormente na Figura 3.11, é necessário deixar o sistema estabilizar na temperatura de referência, não sendo aplicado qualquer ganho no sistema. Quando estabilizar, como indicado no subsecção 2.3.2, é necessário aplicar e variar o ganho proporcional, fazendo uma observação contínua à resposta do sistema, para alcançar o ganho crítico, K_u .

Neste caso são utilizadas ferramentas de controle em tempo real, usufruindo do *User Interface (UI) control* do MATLAB para controle do sistema. Ao criar uma interface de controle, esta possibilita a análise da variação do valor de temperatura de referência e do ganho. A criação da interface e dos seus controles é dada pela Listagem 3.11.

```

1 % Cria uma figura e os controlo para mudancas interativas
2 fig = figure('Name', 'Controlo de Temperatura');
3 editTref = uicontrol('Style', 'edit', 'String', num2str(
    Tref), 'Position', [60, 10, 50, 20]);
4 editK = uicontrol('Style', 'edit', 'String', num2str(K), '
    Position', [160, 10, 50, 20]);

```

Listagem 3.11: Criação de Interface de Controlo

A atualização dos valores é comandada dentro do *loop* do ciclo de controlo, atribuindo novos valores a T_{ref} e a K , como observado na Listagem 3.12, garantindo assim a variação dos mesmos em tempo real.

```

1 % Atualiza Tref e K pelos controlos implementados
2 Tref = str2double(get(editTref, 'String'));
3 K = str2double(get(editK, 'String'));

```

Listagem 3.12: Edição de Valores pela Interface de Controlo

Durante os testes efetuados, a janela de controlo apresenta um gráfico como ilustrado na Figura 3.14.

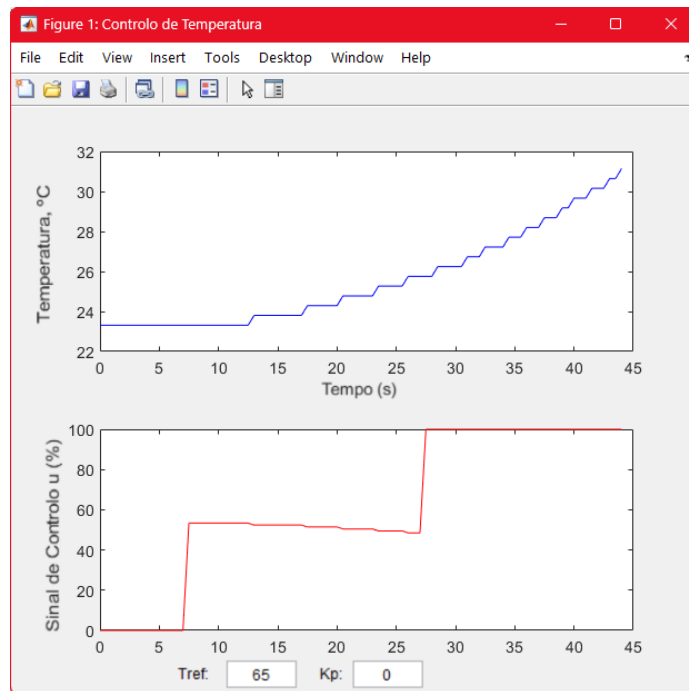


Figura 3.14: Janela de Controlo

A validação do valor de ganho crítico é alcançada quando a oscilação da temperatura tiver uma amplitude e período constante. Este estado não foi possível alcançar na experiência inicial, havendo uma continuação da experiência variando o valor de

ganho até alcançar o estado pretendido. Cada experiência tem uma duração de 1800 segundos (30 minutos), tornando esta observação num processo moroso.

A primeira experiência é representada pela Figura 3.15, encontrando-se identificados os vários valores do ganho aplicados.

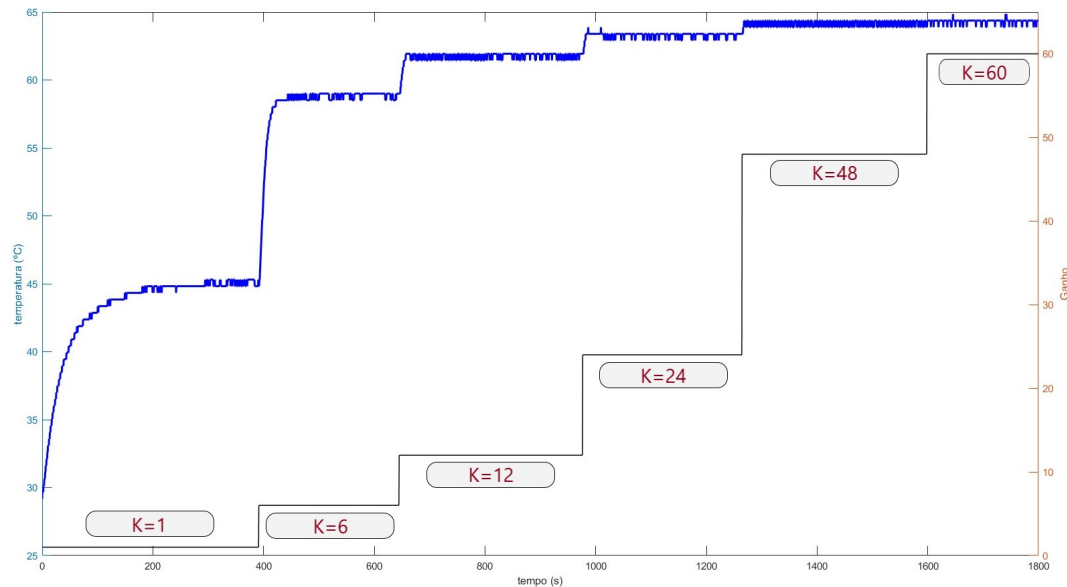


Figura 3.15: Resposta à Variação do Ganho S1 - Experiência 1

Na primeira experiência é aplicada uma variação de ganho gradual e elevada, no entanto, apesar do sistema estar com uma amplitude de onda relativamente constante, numa inspeção mais próxima à resposta obtida, Figura 3.16, é verificado que não é possível destacar, de forma conclusiva, um período de oscilação constante. É de referir, todavia, que é verificado que os valores de ganho $K = 48$ e $K = 60$, aproximam-se do período esperado.

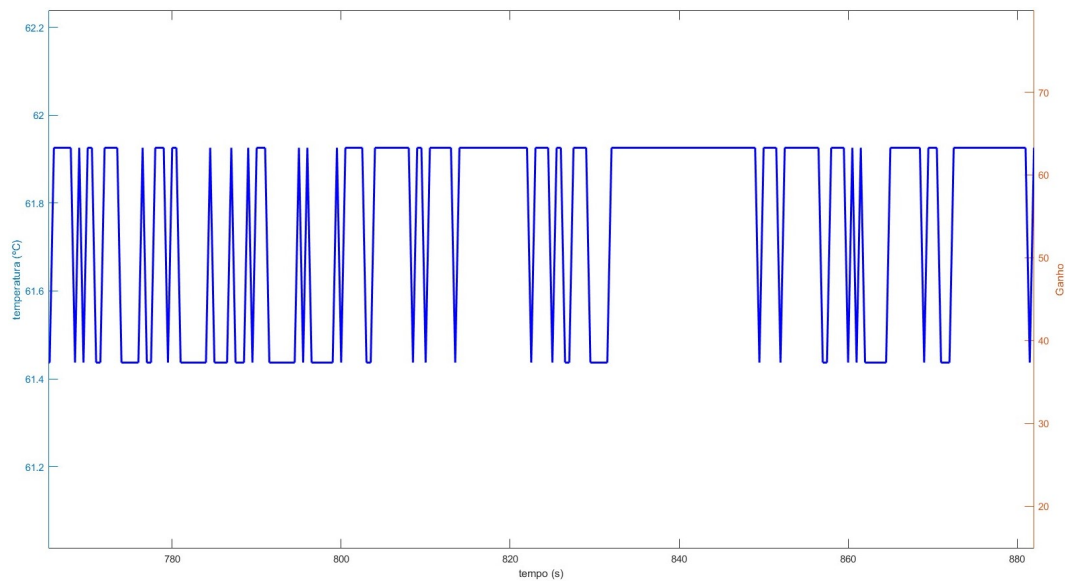


Figura 3.16: Observação do Período S1 - Experiência 1

Na segunda experiência, Figura 3.17, de forma mais precisa, são aplicados valores de ganho K com variações mais reduzidas. Com este ajuste mais pormenorizado, é possível alcançar a oscilação constante desejada.

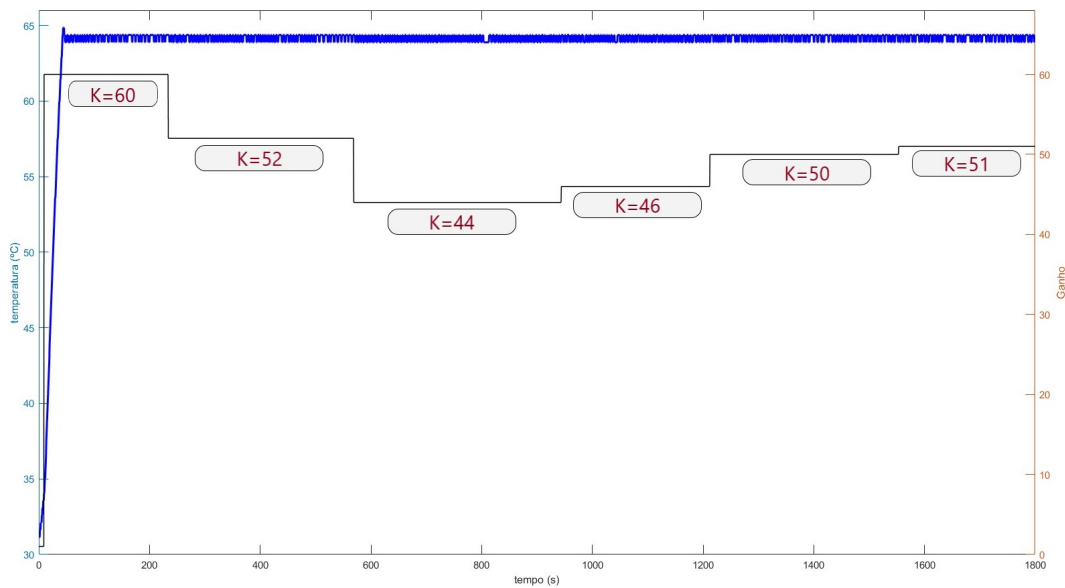


Figura 3.17: Resposta à Variação do Ganho S1 - Experiência 2

Nesta experiência, ao ser efetuada uma ampliação dos dados, é possível observar em determinados períodos, como o da Figura 3.18, uma oscilação constante que

permite a validação do ganho crítico e do período crítico, assumindo os valores:

$$K_u = 52$$

$$P_u = 12.5 \text{ s}$$

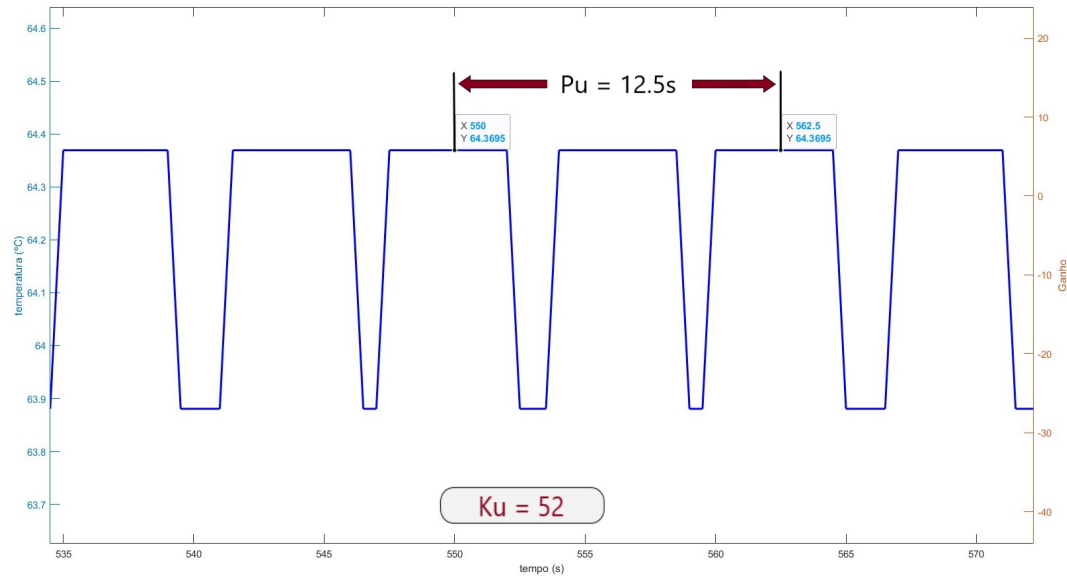


Figura 3.18: Ganho e Período Crítico S1

Com a obtenção destes valores, é possível aplicar as fórmulas tabeladas na subsecção 2.3.2 e calcular os parâmetros de controlo P, PI e PID. Os parâmetros a aplicar em cada tipo de controlador é indicado na Tabela 3.4.

Tabela 3.4: Parâmetros para Z-N em malha fechada do Sistema 1

Controlador	K_p	T_i	T_d
P	26.000	—	—
PI	23.400	10.417	—
PID	31.200	6.250	1.562

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obtém-se o gráfico da Figura 3.19.

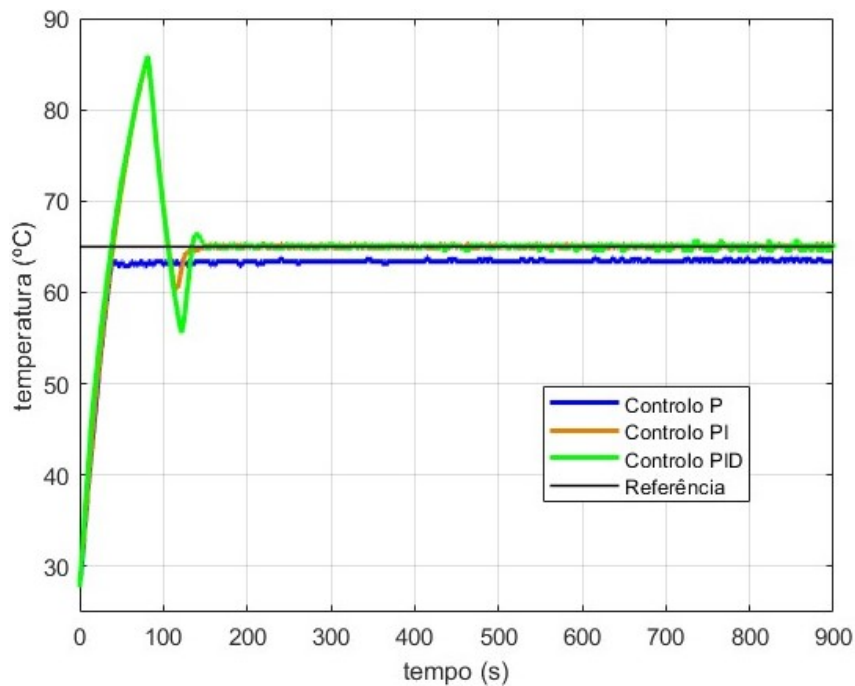


Figura 3.19: Controlo por Z-N em malha fechada do Sistema 1

Assim, pode ser observada uma resposta do controlador P rápida mas com um erro em regime permanente relevante, perante a temperatura de referência. Os controladores PI e PID possuem ambos um elevado *overshoot*, no entanto o controlador PI consegue estabilizar ligeiramente antes do controlador PID.

É importante referir que, ao contrário da resposta em degrau, ao aplicar estes controladores, o tempo de processamento era elevado, havendo necessidade de aumentar o tempo de amostragem. Passando este tempo de 0.1 segundos para 0.5 segundos, permitiu que o SERT tivesse tempo para processar e comunicar todos os dados.

3.3.2 Sistema 1 - Ziegler-Nichols em malha aberta

No método de sintonia por Ziegler-Nichols em malha aberta, para o cálculo dos parâmetros PID, utilizam-se os dados obtidos da resposta ao degrau do sistema 1, dos quais:

$$\text{Ganho DC: } K = 0.9087$$

$$\text{Constante de Tempo: } \tau = 96.90 \text{ s}$$

$$\text{Tempo de Atraso: } \tau_D = 0.60 \text{ s}$$

Com estes valores, seguindo o tabelado na subsecção 2.3.2, obteve-se os parâmetros da Tabela 3.5.

Tabela 3.5: Parâmetros para Z-N em malha aberta do Sistema 1

Controlador	K_p	T_i	T_d
P	177.726	—	—
PI	159.954	2.000	—
PID	213.272	1.200	0.300

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obteve-se o gráfico da Figura 3.20.

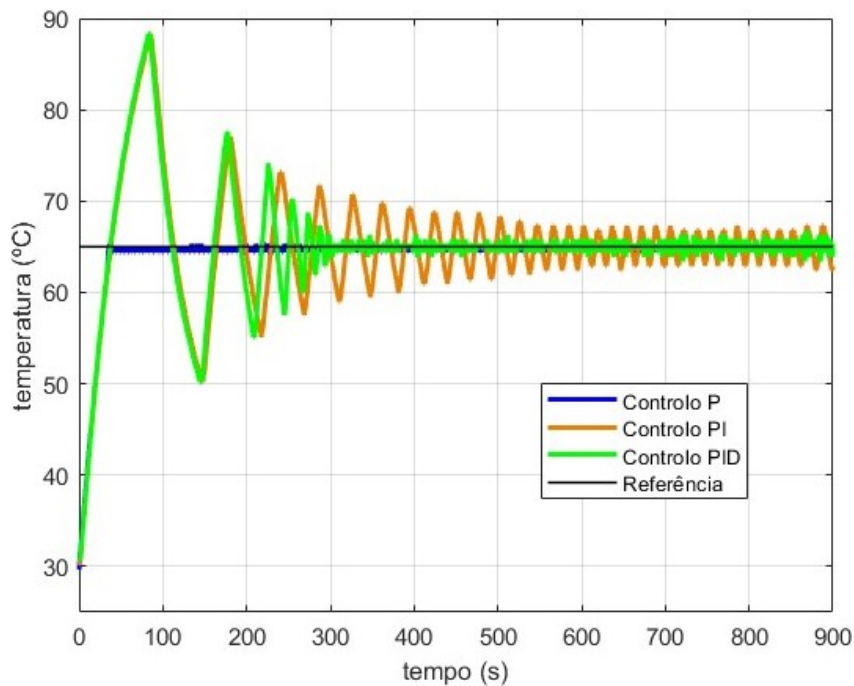


Figura 3.20: Controle por Z-N em malha aberta do Sistema 1

Com a aplicação do controlador P no sistema 1, a resposta é rápida e com um erro em regime permanente muito reduzido perante a temperatura de referência, como é possível observar. Os controladores PI e PID possuem ambos um elevado *overshoot*, seguido de uma elevada oscilação. O controlador PI, apesar de reduzir a amplitude, mantém um valor sensivelmente constante, enquanto que o controlador PID, apesar de uma amplitude reduzida inicial, é verificado um aumento da sua amplitude no fim da experiência, revelando alguma instabilidade.

3.3.3 Sistema 1 - Cohen-Coon

No método de sintonia por Cohen-Coon, para o cálculo dos parâmetros PID, tal como no método anterior, utilizam-se os dados obtidos da resposta do sistema à entrada em degrau. Assim, seguindo o tabelado na subsecção 2.3.3, é possível alcançar os parâmetros da Tabela 3.6.

Tabela 3.6: Parâmetros para C-C do Sistema 1

<i>Controlador</i>	K_p	T_i	T_d
P	183.444	—	—
PI	160.045	1.972	—
PID	240.206	1.496	0.222

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obtiveram-se as respostas ilustradas na Figura 3.21.

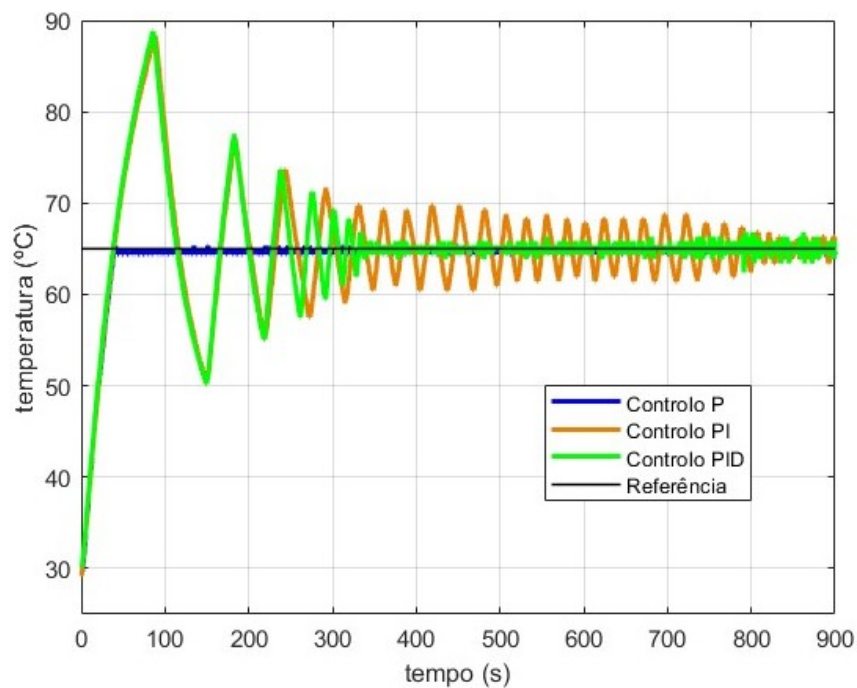


Figura 3.21: Controlo por C-C do Sistema 1

Pelo método de Cohen-Coon, a resposta dos três tipos de controladores aplicados é similar à do método Ziegler-Nichols em malha aberta, no entanto a oscilação do controlador PI tem uma amplitude ao longo do tempo superior ao método apresentado anteriormente.

3.3.4 Sistema 2 - Ziegler-Nichols em malha fechada

Para o sistema 2, no método de sintonia por Ziegler-Nichols em malha fechada, tal como no sistema 1, é necessária a obtenção do valor do ganho crítico onde o sistema entra em oscilação constante.

O método de controle em tempo real segue o representado na subsecção 3.3.1. Tal como aplicado no sistema 1, para garantir a validação do valor de ganho crítico, cada experiência tem uma duração de 1800 segundos (30 minutos).

No sistema 2, durante a experiência inicial, Figura 3.22, após aplicar alguns valores de ganho K , é possível começar a visualizar uma oscilação com um período mais constante, mas não suficiente.

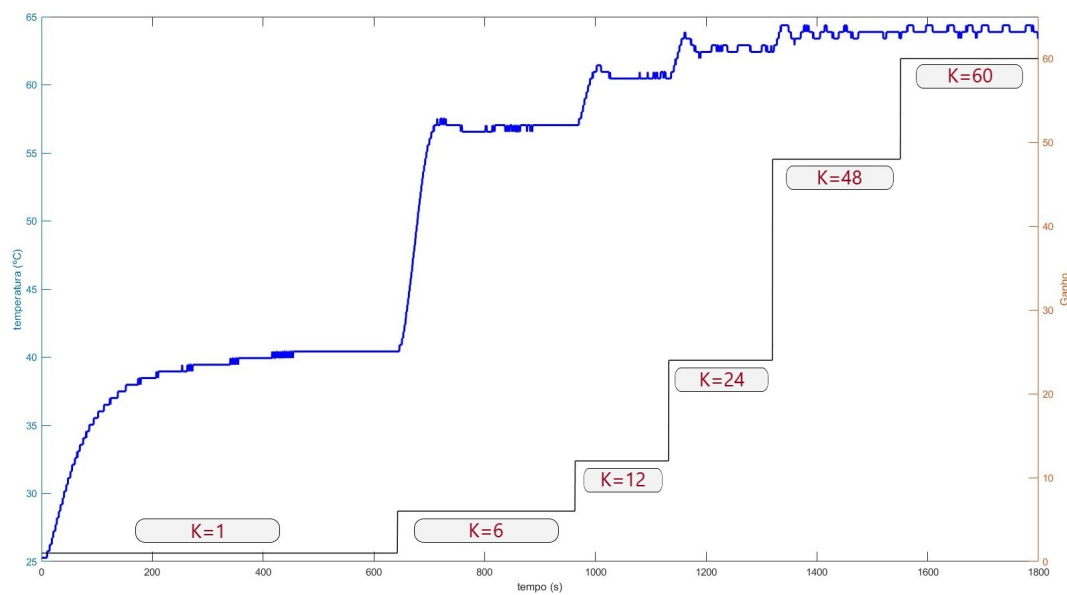


Figura 3.22: Resposta à Variação do Ganho S2 - Experiência 1

Assim, para verificar o valor correto de ganho, é realizada uma segunda experiência, apresentada na Figura 3.23. Nesta experiência é possível observar nitidamente o valor correto de ganho crítico.

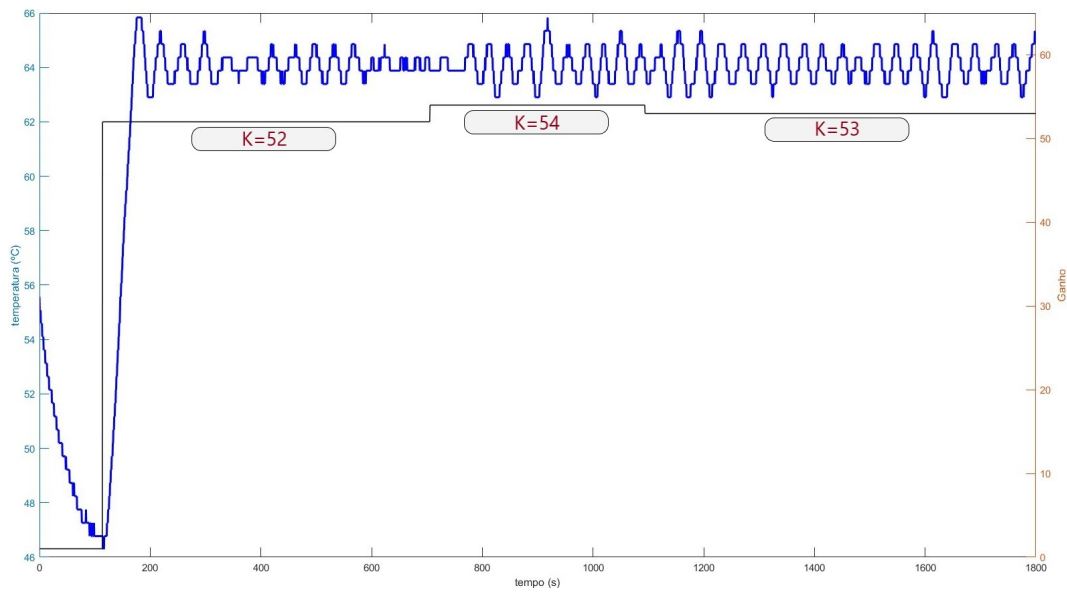


Figura 3.23: Resposta à Variação do Ganho S2 - Experiência 2

Realizando uma ampliação dos dados, Figura 3.24, é possível verificar uma oscilação constante que permite a validação do ganho crítico e do período crítico. Ao observar a imagem é possível destacar algumas variações não normais, no entanto, devido a ser utilizado um sistema físico real, podem haver interferências externas em algumas situações. Os valores de ganho crítico e do período crítico correspondem a:

$$K_u = 53$$

$$P_u = 35 \text{ s}$$

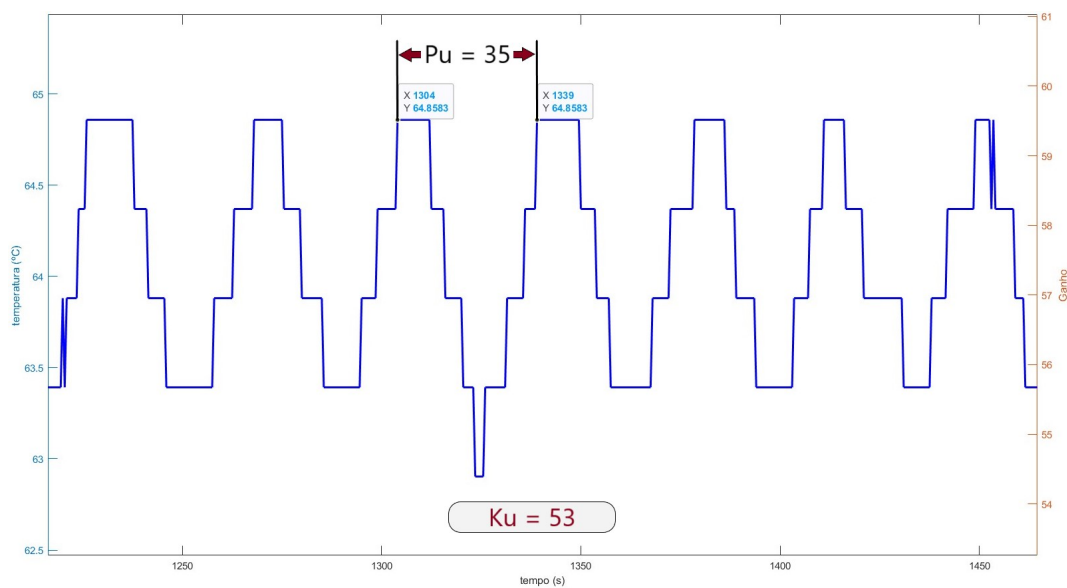


Figura 3.24: Ganho e Período Crítico S2

Com a obtenção destes valores, é possível aplicar as fórmulas tabeladas na subsecção 2.3.2 e calcular os parâmetros de controle P, PI e PID. Os parâmetros a aplicar em cada tipo de controlador são os indicados na Tabela 3.7.

Tabela 3.7: Parâmetros para Z-N em malha fechada do Sistema 2

Controlador	K_p	T_i	T_d
P	26.5	—	—
PI	23.85	29.167	—
PID	31.8	17.5	4.375

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obtém-se o gráfico da Figura 3.25.

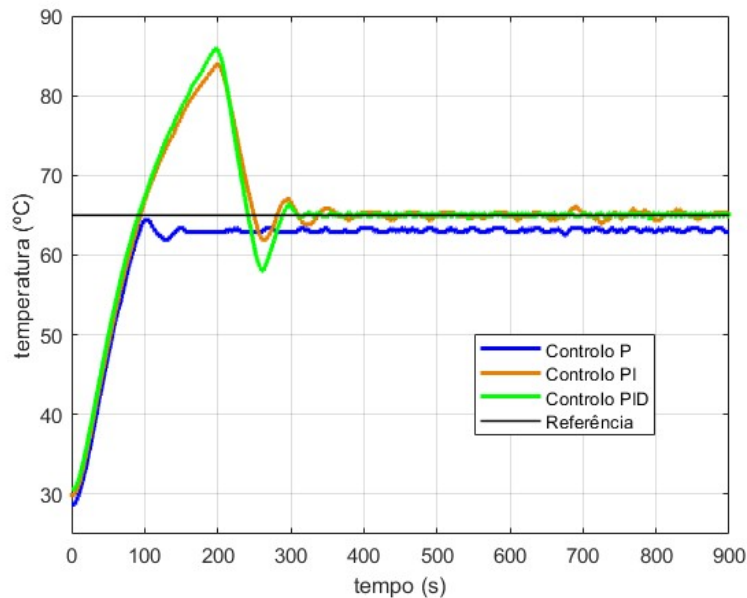


Figura 3.25: Controle por Z-N em malha fechada do Sistema 2

Com as respostas apresentadas, pode ser observada uma resposta do controlador P rápida, em contrapartida apresenta algum erro em regime permanente, relevante perante a temperatura de referência. Os restantes possuem ambos um elevado *overshoot*, existindo uma ligeira diferença no controlador PI, que apresenta um menor *overshoot*, mas em contrapartida atrasa um pouco a estabilização da temperatura do sistema. Apesar de reduzida, após os controladores PI e PID estabilizarem, existe uma pequena oscilação na temperatura, a qual se deve ao próprio sistema físico.

3.3.5 Sistema 2 - Ziegler-Nichols em malha aberta

No método de sintonia por Ziegler-Nichols em malha aberta, para cálculo dos parâmetros PID, utilizam-se os dados obtidos na execução da resposta em degrau do sistema 2, dos quais:

$$\text{Ganho DC: } K = 0.7038$$

$$\text{Constante de Tempo: } \tau = 136.50 \text{ s}$$

$$\text{Tempo de Atraso: } \tau_D = 9.30 \text{ s}$$

Com estes valores, seguindo o tabelado na subsecção 2.3.2, é possível alcançar os parâmetros da Tabela 3.8.

Tabela 3.8: Parâmetros para Z-N em malha aberta do Sistema 2

<i>Controlador</i>	K_p	T_i	T_d
P	20.855	—	—
PI	18.769	31.000	—
PID	25.025	18.600	4.650

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obteve-se o gráfico da Figura 3.26.

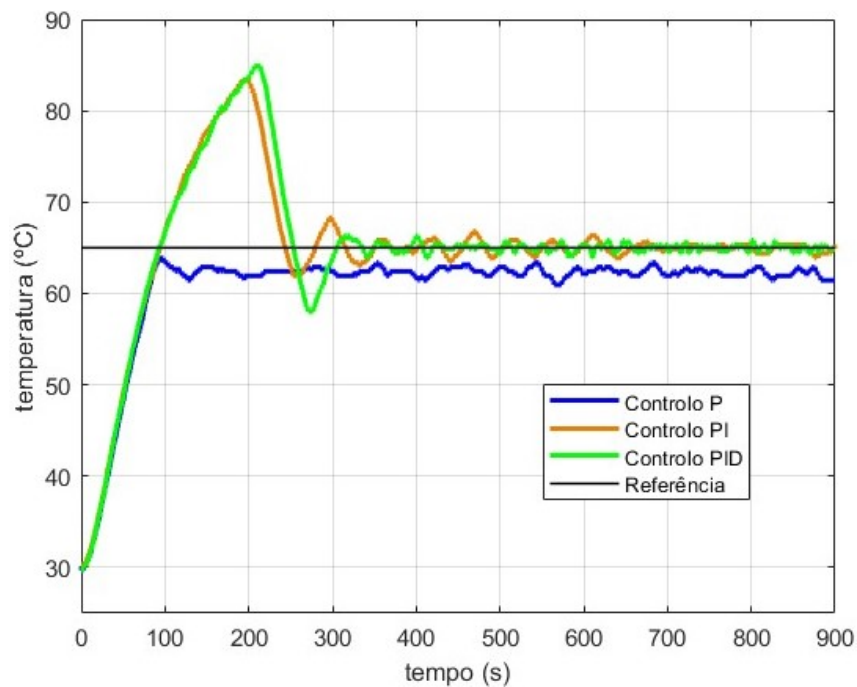


Figura 3.26: Controlo por Z-N em malha aberta do Sistema 2

A resposta dada para este método possui alguma variação quando alcança a sua temperatura de estabilização. Tal como em análises anteriores, o controlador P possui um erro em regime permanente relevante e as respostas obtidas pelos controladores PI e PID alcançam o valor de referência. Estes, no entanto, demoram um longo período a estabilizar.

3.3.6 Sistema 2 - Cohen-Coon

No método de sintonia por Cohen-Coon, para cálculo dos parâmetros PID, tal como no método anterior, utilizam-se se dados obtidos na execução da resposta em degrau do sistema correspondente. Assim, seguindo o tabelado na subsecção 2.3.3, é possível alcançar os parâmetros da Tabela 3.9.

Tabela 3.9: Parâmetros para C-C do Sistema 2

Controlador	K_p	T_i	T_d
P	21.978	—	—
PI	18.887	27.069	—
PID	28.508	22.602	3.398

Aplicando os parâmetros e observando a resposta do sistema aos três tipos de controlador, obteve-se o gráfico da Figura 3.27.

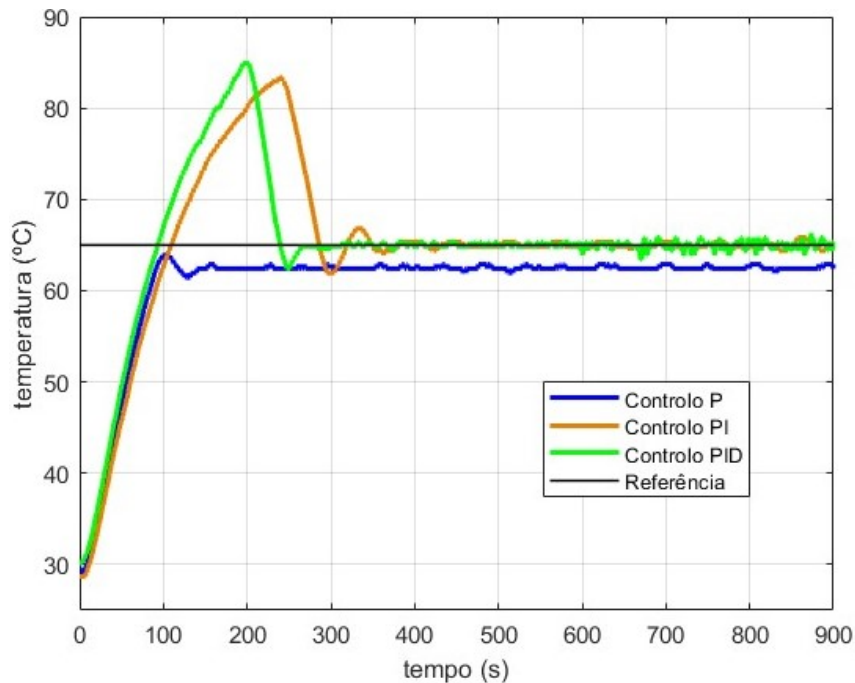


Figura 3.27: Controlo por C-C do Sistema 2

Pelas respostas do sistema apresentadas, pode ser observada uma resposta com um tempo de subida similar entre os três controladores, onde o controlador P é mais rápido a estabilizar, mas com um erro em regime permanente relevante perante a temperatura de referência. Os controladores PI e PID possuem ambos um elevado *overshoot*, no entanto o controlador PID consegue estabilizar primeiro que o controlador PI.

3.3.7 Comparação dos métodos

Após a análise individual de cada método e os seus controladores, de forma a se verificar quais destes possuem um melhor desempenho, são dispostos os diversos valores de erro de cada critério, em cada sistema, nas Tabelas 3.10 e 3.11. Para cada um dos critérios de erro apresentados (*Integral Absolute Error* (IAE), *Integral Squared Error* (ISE), *Integral Time Absolute Error* (ITAE), *Integral Time Squared Error* (ITSE)), são destacadas as três melhores respostas, identificadas da melhor para a pior, com as cores verde, amarelo e cor-de-laranja, respetivamente.

Tabela 3.10: Comparação de Erros do PID geral - Sistema 1

<i>Sistema 1</i>				
Critério	Método	P	PI	PID
<i>IAE</i>	Z-N F	4168	3237	3376
	Z-N A	1642	7343	5570
	C-C	1804	7744	5980
<i>ISE</i>	Z-N F	38.3×10^3	53.7×10^3	52.1×10^3
	Z-N A	27.7×10^3	82.4×10^3	75.2×10^3
	C-C	32.2×10^3	87.0×10^3	79.9×10^3
<i>ITAE</i>	Z-N F	129.5×10^4	30.0×10^4	35.1×10^4
	Z-N A	21.4×10^4	169.4×10^4	83.0×10^4
	C-C	22.9×10^4	187.9×10^4	93.9×10^4
<i>ITSE</i>	Z-N F	234.8×10^4	196.7×10^4	215.8×10^4
	Z-N A	31.7×10^4	834.4×10^4	559.0×10^4
	C-C	38.1×10^4	941.4×10^4	632.0×10^4

Tabela 3.11: Comparação de Erros do PID geral - Sistema 2

<i>Sistema 2</i>				
Critério	Método	P	PI	PID
<i>IAE</i>	Z-N F	6742	7382	7393
	Z-N A	7718	7475	7839
	C-C	7613	8347	7052
<i>ISE</i>	Z-N F	100.3×10^3	131.6×10^3	134.2×10^3
	Z-N A	98.0×10^3	127.5×10^3	139.1×10^3
	C-C	101.4×10^3	154.9×10^3	128.9×10^3
<i>ITAE</i>	Z-N F	163.5×10^4	100.7×10^4	96.5×10^4
	Z-N A	225.7×10^4	109.5×10^4	112.3×10^4
	C-C	210.7×10^4	119.2×10^4	93.6×10^4
<i>ITSE</i>	Z-N F	51.5×10^5	106.0×10^5	118.3×10^5
	Z-N A	79.5×10^5	101.8×10^5	125.4×10^5
	C-C	71.5×10^5	140.9×10^5	105.1×10^5

No sistema 1, ao contrário do sistema 2, é apresentado o controlador P como a melhor resposta em todos os critérios. Isto é devido ao elevadíssimo valor de ganho proporcional que leva a um reduzido erro em regime permanente e ao facto dos outros dois controladores (PI e PID) apresentarem *overshoots* muito grandes.

No sistema 2, apesar de nos critérios IAE e ISE o controlador P ter vantagem, acrescentando a componente *tempo* com os critérios ITAE e ITSE, os controladores PI e PID passam a ser uma melhor solução.

Para ser possível realçar a vantagem dos controladores PI e PID, será aplicado um esquema *anti-windup* em ambos os sistemas, pelos métodos que tiveram melhores resultados.

3.3.8 Aplicação de *Anti-windup*

Para ser possível melhorar a resposta de controladores com as componentes PI e PID, pode ser aplicado um esquema *anti-windup*.

Uma das razões para um elevado valor de erro presente nestes controladores (tabelas da subsecção 3.3.7), é a ação integrativa continuar a elevar o seu valor, mesmo ocorrendo a saturação do controlador. Este comportamento origina o elevado *overshoot* presente nas respostas, verificando-se que, apesar de se ter alcançado a temperatura de referência, o SERT continua a elevar a sua temperatura.

Com um modelo *anti-windup*, assim que o atuador satura, este atua de forma a que a entrada no integrador se mantenha num valor baixo [22].

A aplicação deste modelo num controlador PI resulta no diagrama representado na Figura 3.28. Em adição ao diagrama do controlador habitual, é adicionada a

diferença entre o valor de controlo limitado com o valor não limitado ($u - v$), em que, quando dividido pela constante de tempo adicionada, T_i , é somado esse valor à entrada na componente integrativa. No caso do controlo PI, a constante de tempo adicionada pode ser dada por:

$$T_i = 0.5 \times T_i \quad (3.11)$$

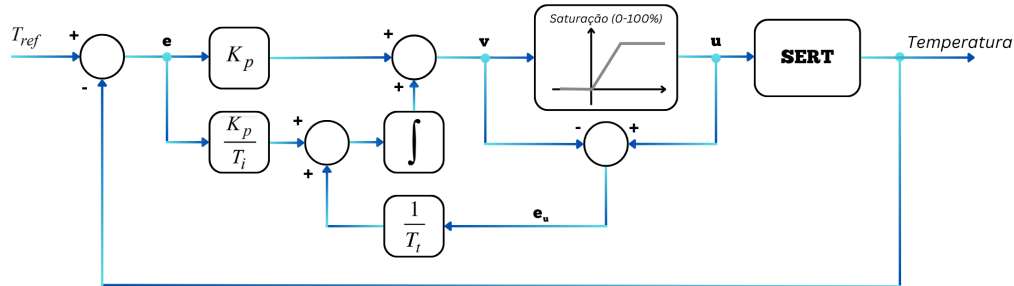


Figura 3.28: Diagrama - Controlo PI com *Anti-windup*

A Listagem 3.13 apresenta a alteração efetuada ao método de controlo considerado. Aqui é introduzido o controlo não limitado, v , e a diferença entre este e o controlo limitado, e_u .

```

1      % erro
2      e = Tref - T_atual;
3      % controlo P
4      P = PI_kp * e;
5      % controlo PI
6      v = P+I;
7
8      % Saturacao
9      if v>100
10         u = 100;
11     elseif v<0
12         u = 0;
13     else
14         u = v;
15     end
16     T1(u); % ON
17
18     e_u = u - v;
19     I = I + Ki * e + e_u/PI_tt; % Atualizar o Controlo I

```

Listagem 3.13: Controlo PI com *Anti-windup*

No caso do controlo PID, acrescentando apenas a componente derivativa ao sistema, Figura 3.29, a constante de tempo T_t passa a ser dada por:

$$T_t = \sqrt{T_i T_d} \quad (3.12)$$

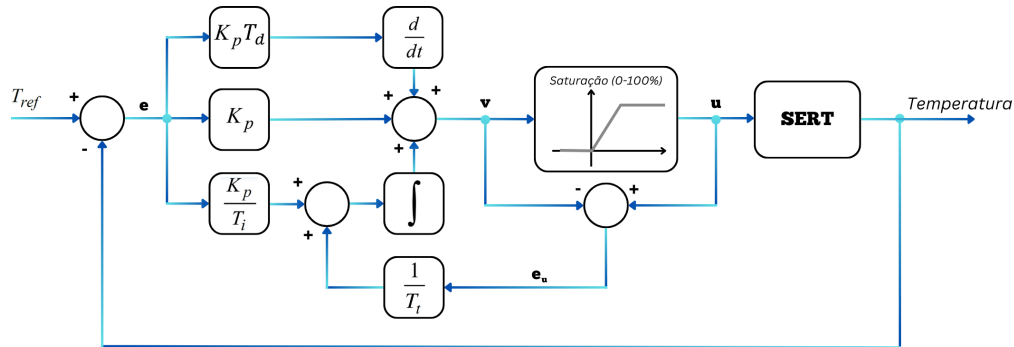


Figura 3.29: Diagrama - Controlo PID com *Anti-windup*

De forma similar ao controlador PI, é necessário incorporar as novas variáveis no controlo do sistema, estando estas representadas pela Listagem 3.14.

```

1      % erro
2      e = Tref - T_atual;
3
4      P = PID_kp * e;           % controlo P
5      I = I + Ki * e + e_u/PID_tt; % controlo I
6      D = Kd * (e - e0);       % controlo D
7
8      v = P + I + D; % Controlo PID
9
10     % Saturacao
11     if v>100
12         u = 100;
13     elseif v<0
14         u = 0;
15     else
16         u = v;
17     end
18     T1(u); % ON
19
20     % Atualizar erro anterior
21     e0 = e;
22     e_u = u - v;

```

Listagem 3.14: Controlo PID com *Anti-windup*

Com os diagramas apresentados e as devidas reestruturações de código, é possível aplicar o controlo com *anti-windup* no:

- **Sistema 1**

As respostas de controlo com maior proveito, em que é possível aplicar a estrutura de *anti-windup*, são os do controlador PI e PID do método de Ziegler-Nichols em malha fechada, devido ao seu valor nos critérios de erro apresentados. Com os seus parâmetros, e aplicando o cálculo para obter o T_t , obtém-se os valores de controlo a aplicar, indicados na Tabela 3.12.

Tabela 3.12: Parâmetros de esquema com *Anti-windup* do Sistema 1

Controlador	K_p	T_i	T_d	T_t
Z-N F: PI	23.400	10.417	—	5.208
Z-N F: PID	31.200	6.250	1.562	3.125

Mantendo a temperatura de referência e o período de experiência, é alcançado o resultado apresentado na Figura 3.30.

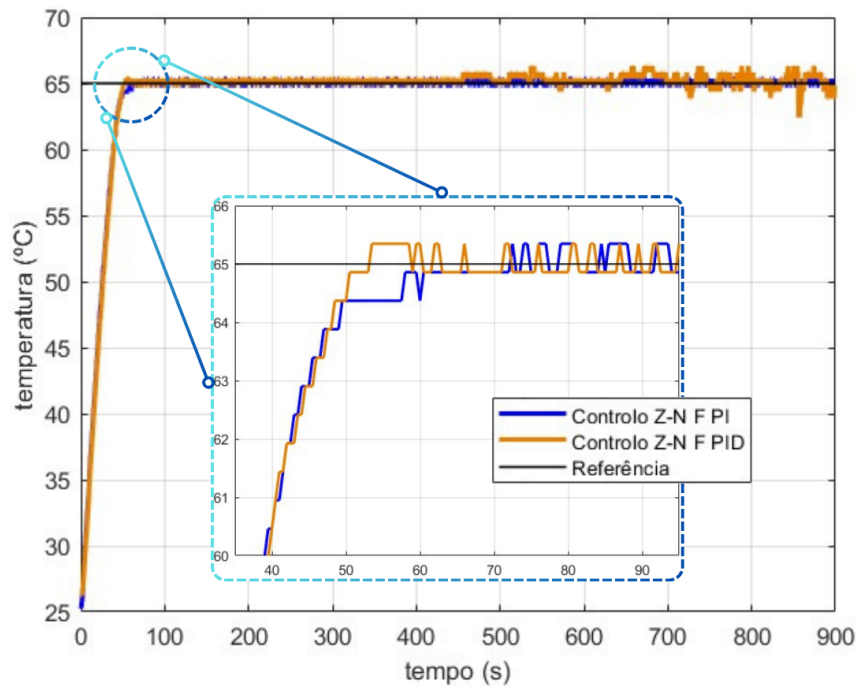


Figura 3.30: Gráfico para *Anti-windup* do Sistema 1

A resposta passou a ter um tempo de subida muito menor e uma eliminação completa do *overshoot*. Com este esquema de controlo PI, o sistema apresenta uma resposta mais estável que com o controlo PID.

- Sistema 2

As respostas de controlo tiveram um proveito similar, sendo analisadas as do controlador PI e PID do método de Ziegler-Nichols em malha fechada, do controlador PI do método de Ziegler-Nichols em malha aberta e do controlador PID do método de Cohen-Coon, devido ao seu valor nos critérios de erro apresentados. Com os seus parâmetros, e aplicando o cálculo para obter T_t , obtém-se os valores de controlo a aplicar, indicados na Tabela 3.13.

Tabela 3.13: Parâmetros de esquema com *Anti-windup* do Sistema 2

Controlador	K_p	T_i	T_d	T_t
Z-N A: PI	18.769	31.000	—	15.500
Z-N F: PI	23.850	29.167	—	14.583
Z-N F: PID	31.800	17.500	4.375	8.750
C-C: PID	28.508	22.602	3.398	8.764

Mantendo a temperatura de referência e o período de experiência, é alcançado o resultado apresentado na Figura 3.31.

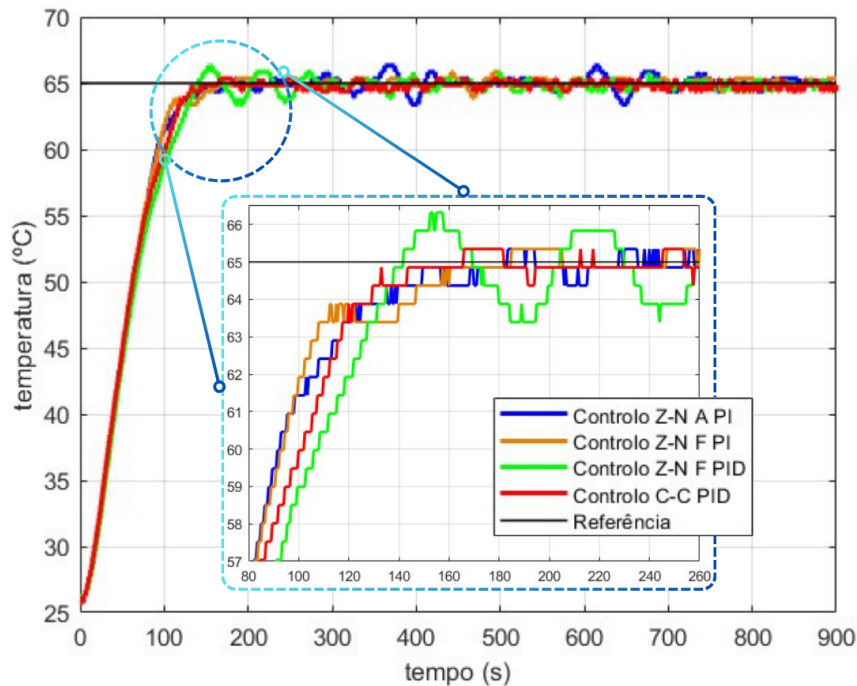


Figura 3.31: Gráfico para *Anti-windup* do Sistema 2

A resposta passou a ter um tempo de subida muito menor, havendo ligeiras diferenças entre os diferentes métodos, e uma redução relevante do *overshoot*. O método Ziegler-Nichols em malha aberta com controlo PI apresenta algumas oscilações durante a experiência. Do mesmo modo, o método Ziegler-Nichols

em malha fechada com controlo PID continua a ter um pequeno *overshoot* e uma ligeira oscilação até estabilizar. É de destacar que o método Ziegler-Nichols em malha fechada com o controlo PI e controlo PID, não apresentam *overshoot* e estabilizam de imediato quando alcançam a temperatura de referência.

3.3.9 Análise do melhor controlador PID

Para seleccionar o melhor método de controlo para cada sistema, é feita uma última análise entre as melhores respostas obtidas na subsecção 3.3.7 em que, quando possível, é aplicado o esquema de *anti-windup*.

Para o sistema 1, é observada a resposta do controlador P para Ziegler-Nichols em malha aberta, controlador PI e controlador PID para Ziegler-Nichols em malha fechada, com aplicação de *anti-windup*. O erro da resposta obtida para estas três experiências é dado na Tabela 3.14, apresentando os quatro critérios de erro.

Tabela 3.14: Comparação de Erros PID melhorado - Sistema 1

Critério	Z-N A: P	Z-N F: PI	Z-N F: PID
<i>IAE</i>	1642	2077	2383
<i>ISE</i>	27.7×10^3	44.7×10^3	46.0×10^3
<i>ITAE</i>	21.4×10^4	18.9×10^4	36.9×10^4
<i>ITSE</i>	31.7×10^4	51.0×10^4	71.6×10^4

Apesar do controlo P pelo método de Ziegler-Nichols em malha aberta conseguir ter melhores valores para os critérios IAE e ISE, o controlo PI por Ziegler-Nichols em malha fechada consegue ser superior na prespetiva temporal dos critérios ITAE e ITSE. Assim, o controlador PI sintonizado por Ziegler-Nichols em malha aberta, é considerado o melhor para o sistema 1, sendo ilustrado individualmente na Figura 3.32. É possível observar um tempo de subida baixo e onde não é verificado nenhum *overshoot*, nem oscilação.

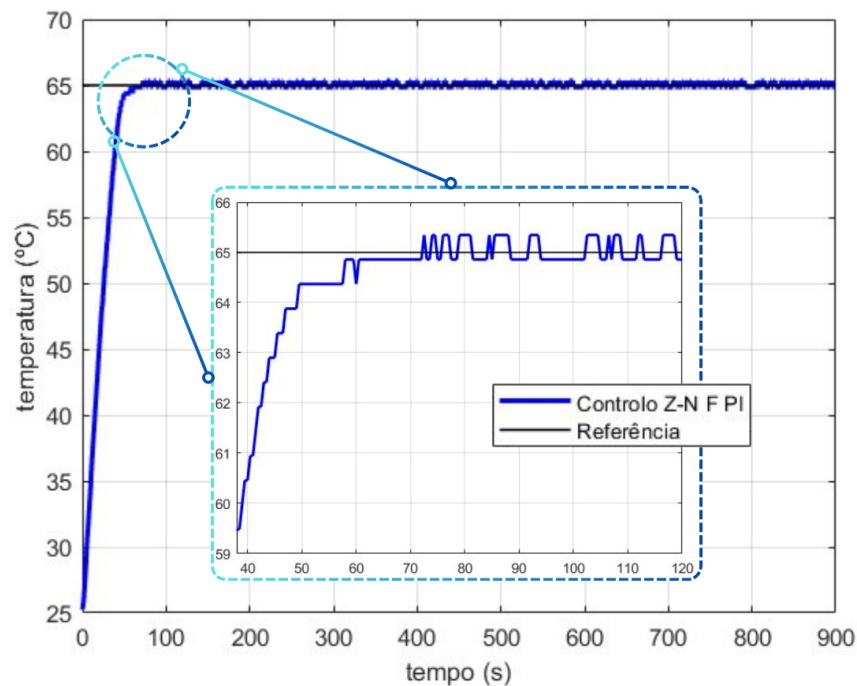


Figura 3.32: Gráfico da Melhor Resposta do Sistema 1

Para o sistema 2, é observada a resposta do controlador P, controlador PI e controlador PID para Ziegler-Nichols em malha fechada, o controlador PI para Ziegler-Nichols em malha aberta e do controlador PID para Cohen-Coon. Todos os controladores PI e PID apresentam um esquema com *anti-windup*. O erro da resposta obtida para estas cinco experiências é dado na Tabela 3.15.

Tabela 3.15: Comparação de Erros PID melhorado - Sistema 2

Critério	Z-N F: P	Z-N F: PI	Z-N F: PID	Z-N A: PI	C-C: PID
<i>IAE</i>	6742	4767	5255	5003	4838
<i>ISE</i>	100.3×10^3	116.6×10^3	122.8×10^3	117.2×10^3	116.4×10^3
<i>ITAE</i>	163.5×10^4	36.2×10^4	44.2×10^4	48.0×10^4	38.5×10^4
<i>ITSE</i>	51.5×10^5	30.5×10^5	35.1×10^5	32.1×10^5	31.1×10^5

Neste sistema, o controle P pelo método de Ziegler-Nichols em malha fechada apenas consegue ter um melhor valor de erro no critério ISE, sendo ultrapassado nos restantes critérios pelos restantes métodos. O controle PI por Ziegler-Nichols em malha fechada destaca-se por apresentar os melhores resultados em três critérios: IAE, ITAE e ITSE. Assim este controle, apresentado individualmente na Figura 3.33, possui um tempo de subida relativamente rápido, com uma paragem repentina perto da temperatura de referência, é considerado o melhor para o sistema 2.

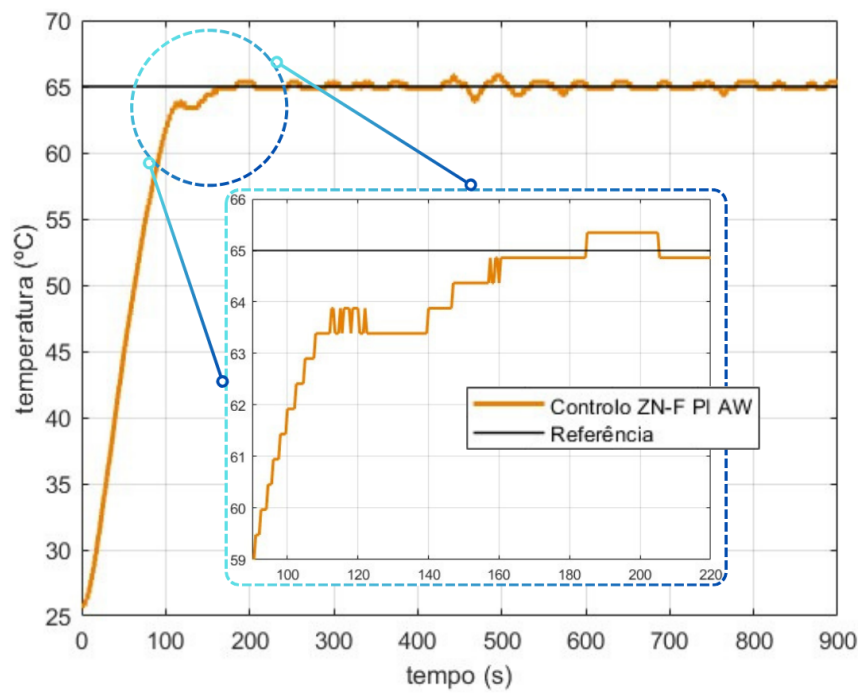


Figura 3.33: Gráfico da Melhor Resposta do Sistema 2

Capítulo 4

Lógica Difusa e Análise de Resultados

Neste capítulo é efetuada a aplicação de métodos de controlo com lógica difusa e uma comparação de resultados com o controlo convencional.

4.1 Controlo por Lógica Difusa

Para o controlo do SERT com lógica difusa, seguindo as orientações da secção 2.2, é necessário criar um controlador de lógica difusa - *Fuzzy Logic Controller* (FLC).

A criação deste FLC deve ser regida perante o tipo de esquema de controlo a aplicar, bem como o tipo de resposta pretendida. Os controladores difusos podem ser divididos em dois grupos distintos, como ilustrado na Figura 4.1, controladores *Fuzzy PID* e controladores *Fuzzy non-PID* [34].

No caso do estudo presente, de um sistema de controlo de difusão de calor, tendo por base o SERT, é mais indicado seguir uma filosofia de PID. A aplicação de algoritmos de lógica difusa num controlador necessita de várias experiências e modelos, recorrendo à aplicação de parâmetros que sejam requeridos pelo sistema a analisar. A implementação prática de um esquema que recorra a diversos parâmetros variáveis leva à dificuldade de aplicação do mesmo, excluindo assim como opção de utilização de um controlador *Fuzzy non-PID*. Como observado na secção 2.3 e 3.3, os algoritmos PID são uma solução validada e com um conjunto de parâmetros já definidos, possibilitando assim uma forma mais simples e direta de aplicação de

lógica difusa. Aproximando a construção do FLC aos algoritmos PID, é alcançado um ajuste mais simples dos parâmetros a aplicar e uma melhor resposta do sistema [35].

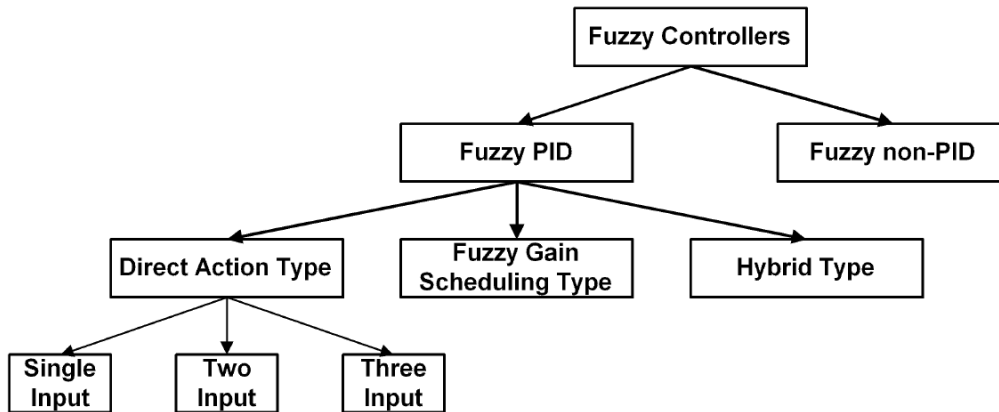


Figura 4.1: Classificação de Controle Fuzzy [34]

A variante *Direct Action* dos controladores *Fuzzy PID* é utilizada quando o FLC atua diretamente no processo. Esta pode ser segmentada pelo número de entradas aceite pelo FLC, em que dependendo da variável de entrada, pode constituir tipos diferentes de controle. Na Tabela 4.1, são apresentados alguns dos diferentes tipos de controle por ação direta, bem como se o algoritmo aplicado origina um controle de valor exato (ext.) ou um controle incremental (inc.) [36, 37]. No controle de valor exato, a saída do FLC é considerada como valor de controle do SERT, enquanto que num controle incremental, a saída do FLC é somada ao valor de controle anterior.

Tabela 4.1: Tipos de Controle Difuso por Ação Direta

Tipo	Variáveis de Entrada	Saída
P	e	ext.
I	e	inc.
PD	$e, \Delta e$	ext.
PI	$e, \Delta e$	inc.
PI	e, ie	ext.
PID	$e, \Delta e$	ext.
PID	$e, \int e dt, de$	inc.
PID	$e, \Delta e, \Delta^2 e$	ext.

Legenda: *ext.* - controle exato, *inc.* - controle incremental, e - erro, Δe - variação do erro, ie - integral do erro e $\Delta^2 e$ - taxa de variação do erro.

A saída do FLC por um controlo exato pode ser dada pela equação (4.1), enquanto que a saída por um controlo incremental é dada pela equação (4.2).

$$u(k) = u_{FLC}(k) \quad (4.1)$$

$$u(k) = u(k-1) + u_{FLC}(k) \quad (4.2)$$

A variante *Fuzzy Gain Scheduling* é aplicada de forma a ser efetuado o ajuste automático dos parâmetros PID durante o controlo de um sistema. Aplicando o controlo tradicional, os seus parâmetros vão sendo modificados para dar a melhor resposta ao erro atual, mediante o FLC criado.

Por último, a variante *Hybrid* (Híbrida), está presente quando as aplicações convencionais de PID são combinadas com um, ou mais, FLC de ação direta. De uma forma minimalista, este controlo pode ser representado pela Figura 4.2 [34, 38].

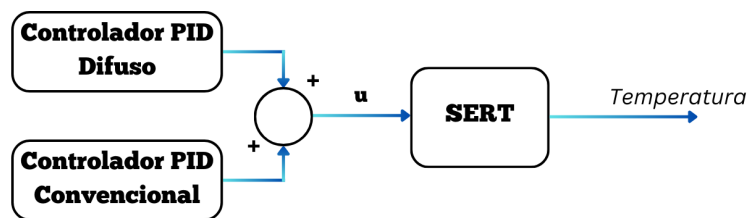
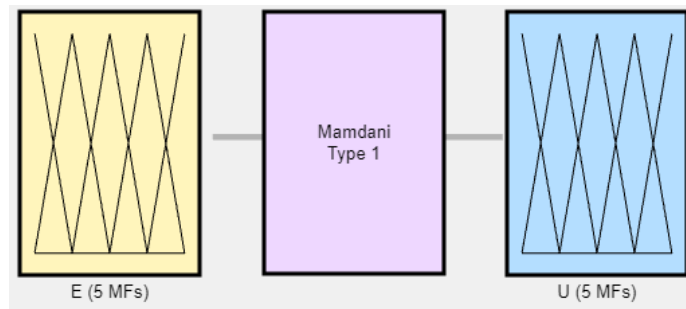


Figura 4.2: Diagrama de um controlador difuso híbrido

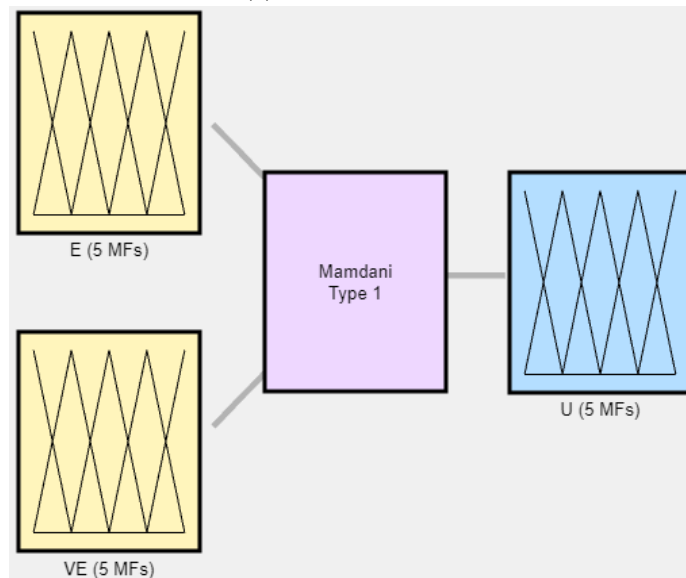
Perante os métodos apresentados, os selecionados para análise são:

- **Controlo P** - *Direct Action*
- **Controlo I** - *Direct Action*
- **Controlo PD** - *Direct Action*
- **Controlo PI** - *Direct Action*
- **Controlo PD+I** - *Hybrid*
- **Controlo PID por 1 FLC** - *Direct Action*
- **Controlo PID por 2 FLC** - *Direct Action*
- **Controlo PID** - *Gain Scheduling*

Os métodos selecionados, de uma forma geral, são constituídos por uma ou duas entradas e uma saída. Assim, recorrendo à *toolbox Fuzzy Logic Designer*, os sistemas dos FLC podem assumir um dos esquemas da Figura 4.3.



(a) FLC - 1 Entrada



(b) FLC - 2 Entradas

Figura 4.3: Sistema do FLC

A configuração dos FLC é dada de forma padrão pela *toolbox*, assumindo os parâmetros representados na Figura 4.4.

Type:	Mamdani Type-1
Name	fuzzy_i
And method	min
Or method	max
Implication method	min
Aggregation method	max
Defuzzification method	centroid

Figura 4.4: Configuração Geral do FLC

As funções de pertinência são similares nos FLC aplicados, sendo selecionado o tipo triangular como o pré-definido para todas. As entradas do FLC são submetidas ao mesmo processo de análise, Figura 4.5, podendo assumir valores entre -1 e 1 , sendo

os seus valores devidamente escalados para não infringir estes limites. Na saída do FLC, apesar do número de funções ser igual, os intervalos de valores admitidos são diferentes, como ilustrado na Figura 4.6. No controlo exato, é previsto que o controlador utilize o valor direto da saída, aplicando assim um intervalo entre 0 e 1, Figura 4.6a, onde é convertido, em código, para o intervalo de 0% a 100%. Ligeiramente diferente, o controlo incremental, de modo a permitir a variação do valor de controlo, permite um intervalo de -1 a $+1$, Figura 4.6b, sendo que o valor de escala será o parâmetro a aplicar no esquema implementado. Na Figura 4.5 e 4.6b, as funções de pertinência são identificadas por Negativo Grande (**NG**), Negativo Pequeno (**NP**), Zero (**Z**), Positivo Pequeno (**PP**) e Positivo Grande (**PG**). Enquanto isso, na Figura 4.6a, para um controlo exato, as identificações são dados por Zero (**Z**), Pequeno (**P**), Médio (**M**), Grande (**G**) e Muito Grande (**MG**).

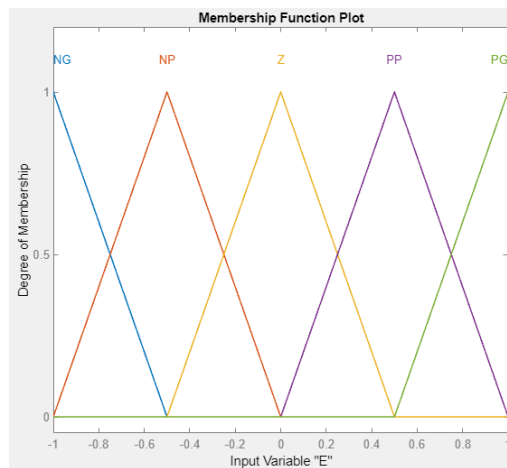


Figura 4.5: Função de Pertinência de Entrada do FLC

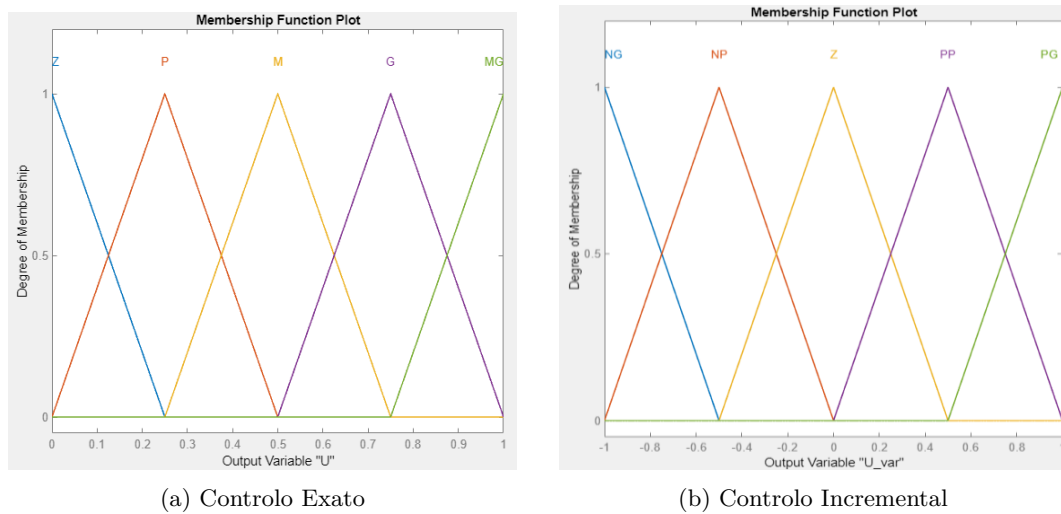


Figura 4.6: Funções de Pertinência de Saída do FLC

Consoante o FLC utilizado, o número de regras varia, variando a superfície de controlo. Em cada controlo, havendo modificação das mesmas, será apresentada a tabela de regras utilizada e, com base nela, a superfície de controlo gerada.

A aplicação do FLC nas experiências é obtido pela leitura do modelo criado e posterior obtenção do valor da saída ao aplicar o valor das entradas definidas, como apresentado na Listagem 4.1.

```

1 % Leitura do modelo de Logica Difusa criado
2 fuzzy = readfis('modeloFuzzy.fis');
3
4 % Uso do controlo com uma entrada
5 U = evalfis(fuzzy,e);
6
7 % Uso do controlo com duas entradas
8 U = evalfis(fuzzy,[e, de]);

```

Listagem 4.1: Utilização de Lógica Difusa

As experiências dos diversos controlos, em face das experiências efetuadas na secção 3.2, regem-se nos mesmos moldes de código, com uma temperatura de referência de 65 °C e um tempo de experiência de 900 segundos.

4.1.1 Controlo P

Num controlador P, apenas é necessário na entrada do FLC o valor do erro, aplicando um ganho G_U na saída U , para otimizar a resposta.

Como é possível observar na Figura 4.7, o erro (e) é multiplicado por G_e , originando o valor de erro escalado (E) [36, 37]. O ganho G_e é aplicado para escalar o valor do erro, tornando-o uniforme para os limites do FLC. Este ganho é uniforme a todos os controlos e a ambos os sistemas (1 e 2), assumindo o valor:

$$G_e = \frac{1}{80} \quad (4.3)$$

O valor aplicado é obtido pela consideração da menor temperatura máxima entre os sistemas 1 e 2, 100 °C, e pela temperatura ambiente mínima possível durante as experiências, 20 °C, onde a diferença entre estes é de 80 °C.

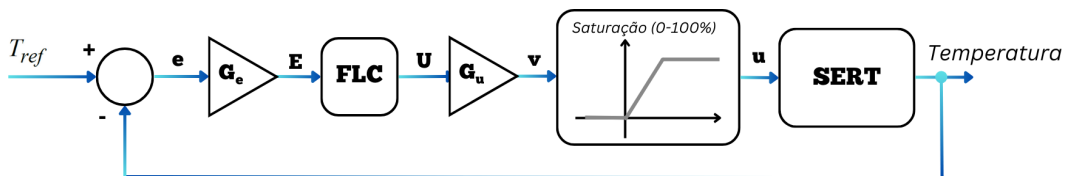


Figura 4.7: Esquema do FLC com Controlo P

Para este tipo de esquema, aplicando apenas um controle P, as regras aplicadas são simples, havendo apenas a interação das cinco funções de pertinência do erro (entrada) com as cinco funções de pertinência do valor de controle (saída), sendo esta atribuição representada na Tabela 4.2.

Tabela 4.2: Lista de Regras do Controle P

E	U
NG	Z
NP	P
Z	M
PP	G
PG	MG

A representação deste controle, devido à sua simplicidade, é dada por uma curva, Figura 4.8, ao invés de uma superfície.

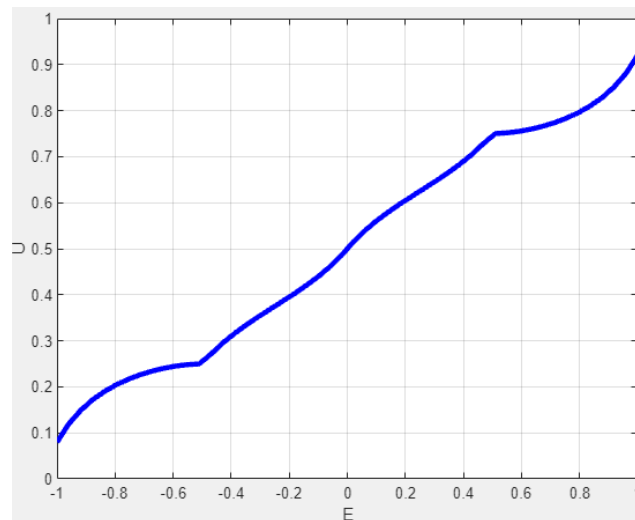
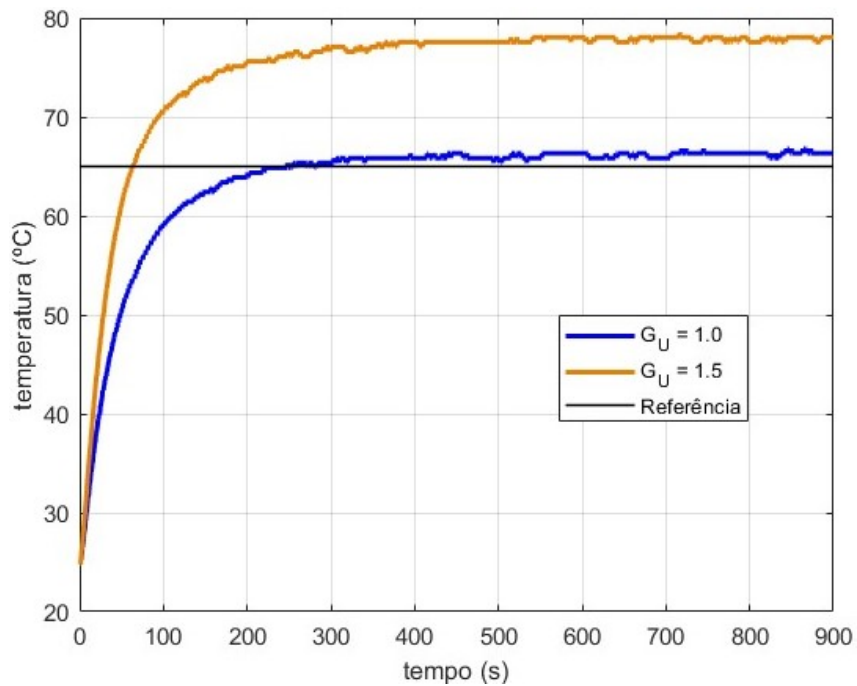


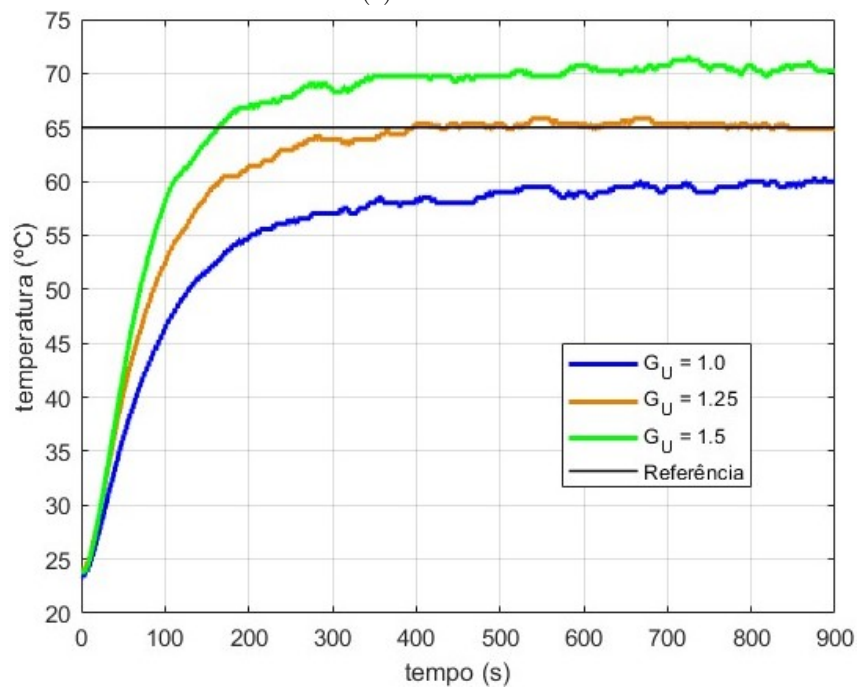
Figura 4.8: Curva de Controle P

Com a estrutura e curva apresentadas, as experiências efetuadas envolveram no sistema 1 os valores ganho $G_U = \{1.0, 1.5\}$ e no sistema 2 os valores de ganho $G_U = \{1.0, 1.25, 1.5\}$.

Os resultados das experiências do sistema 1 e sistema 2 são apresentados na Figura 4.9.



(a) Sistema 1



(b) Sistema 2

Figura 4.9: Controle P por FLC

Assim, com a aplicação do controle P, é possível constatar que este método não é o indicado para o controle de temperatura. Apesar de no sistema 1, Figura 4.9a, com um ganho de $G_U = 1.5$, e de no sistema 2, Figura 4.9b, com um ganho de $G_U = 1.25$, a resposta se encontrar próxima da temperatura de referência, é

possível perceber que este modelo é altamente dependente do ajuste direto de G_U para a correção do erro em regime permanente.

4.1.2 Controlo I

Num controlador I, em similaridade com o controlo anterior, é apenas necessário na entrada do FLC o valor do erro, aplicando um ganho G_U na saída ΔU do FLC, para otimizar a resposta [36, 37].

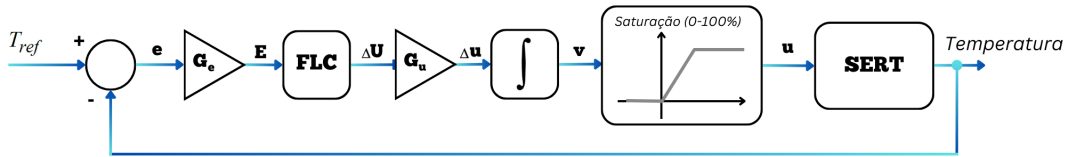


Figura 4.10: Esquema do FLC com Controlo I

Neste esquema, tal como no controlo P, as regras aplicadas são simples, sendo a atribuição de regras representada na Tabela 4.3.

Tabela 4.3: Lista de Regras do Controlo I

E	U
NG	<i>NG</i>
NP	<i>NP</i>
Z	<i>Z</i>
PP	<i>PP</i>
PG	<i>PG</i>

Este controlo, tal como o anterior, também é representado por uma curva, Figura 4.11.

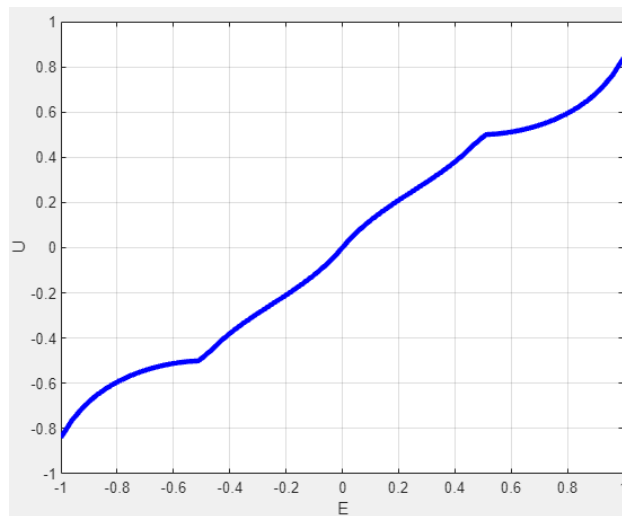
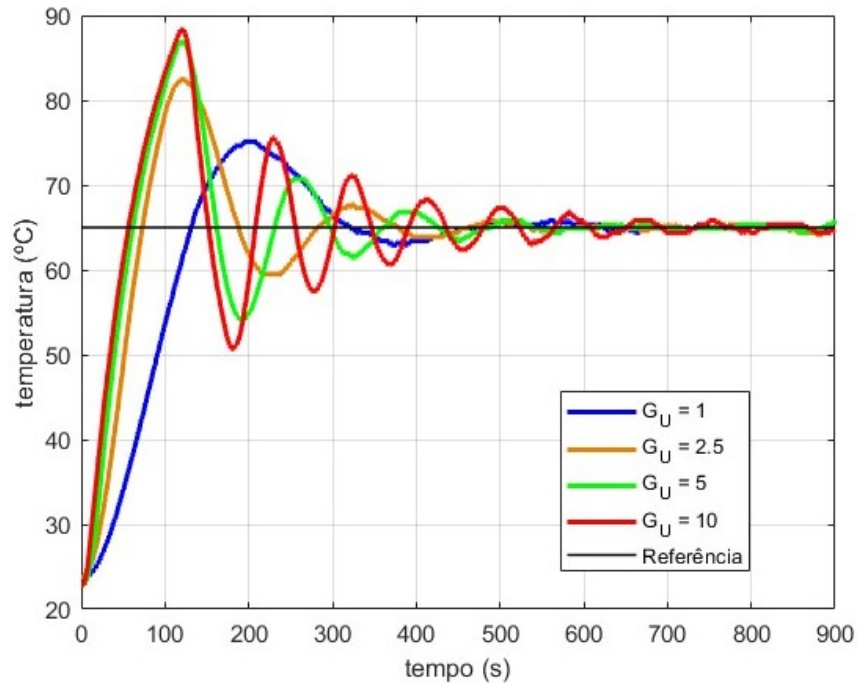
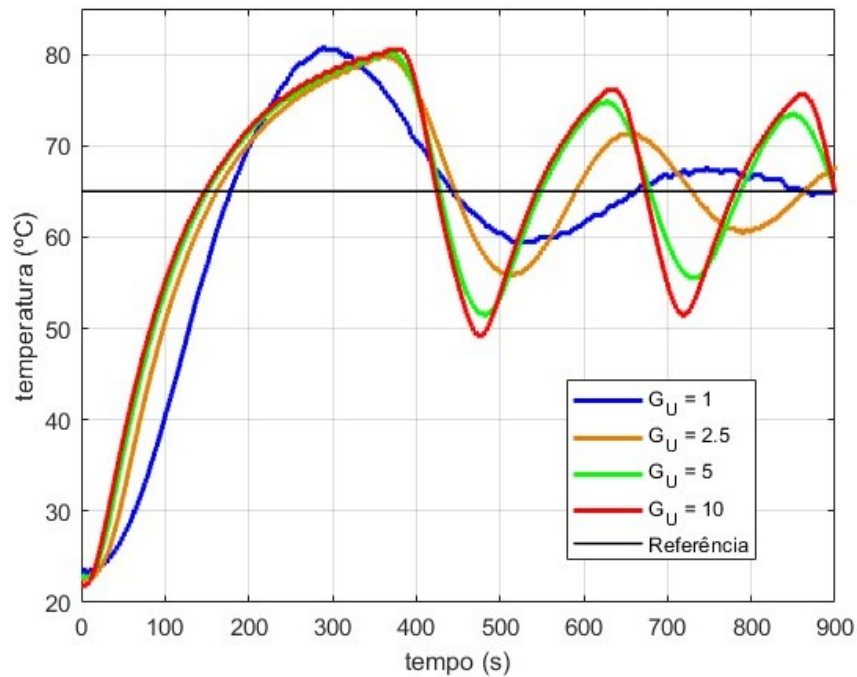


Figura 4.11: Curva do Controlo I

Com a estrutura e curva apresentadas, as experiências efetuadas do sistema 1 e sistema 2 envolveram os ganhos $G_U = \{1.0, 2.5, 5.0, 10.0\}$. Os resultados destas experiências são apresentados na Figura 4.12, verificando-se um valor de *overshoot* e oscilação elevados. Constatou-se ainda que, para o sistema 2, não se obteve uma estabilização do sistema durante o tempo total de simulação.



(a) Sistema 1



(b) Sistema 2

Figura 4.12: Controle I por FLC

De modo a melhorar a resposta, pode ser aplicado o esquema *anti-windup* apresentado na Figura 4.13, aplicando o parâmetro T_t utilizado na subsecção 3.3.8.

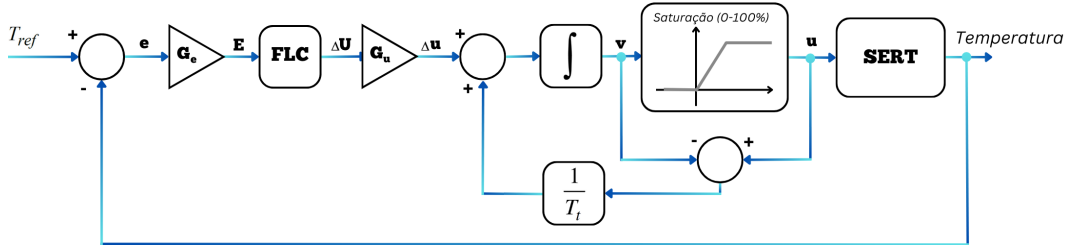
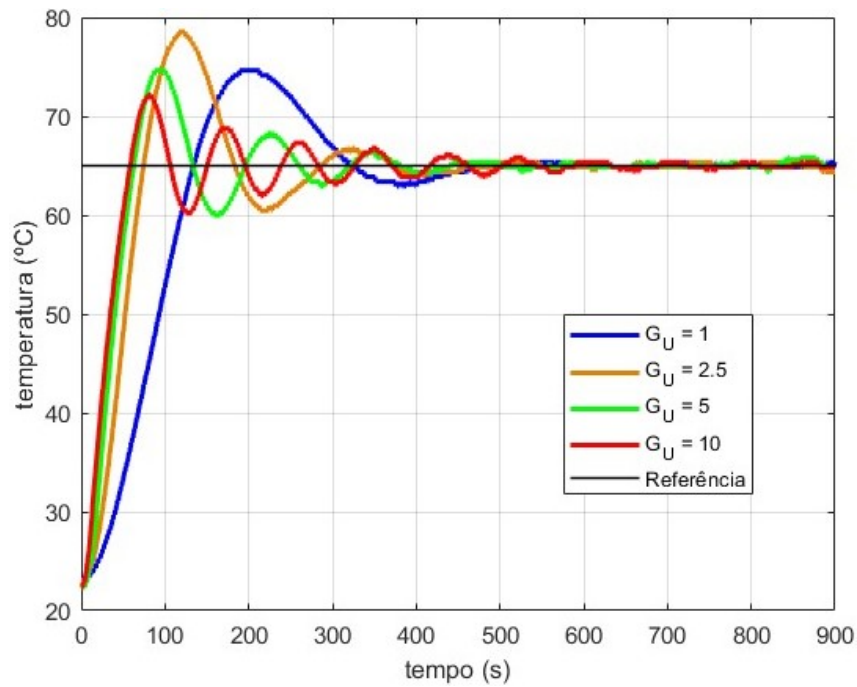
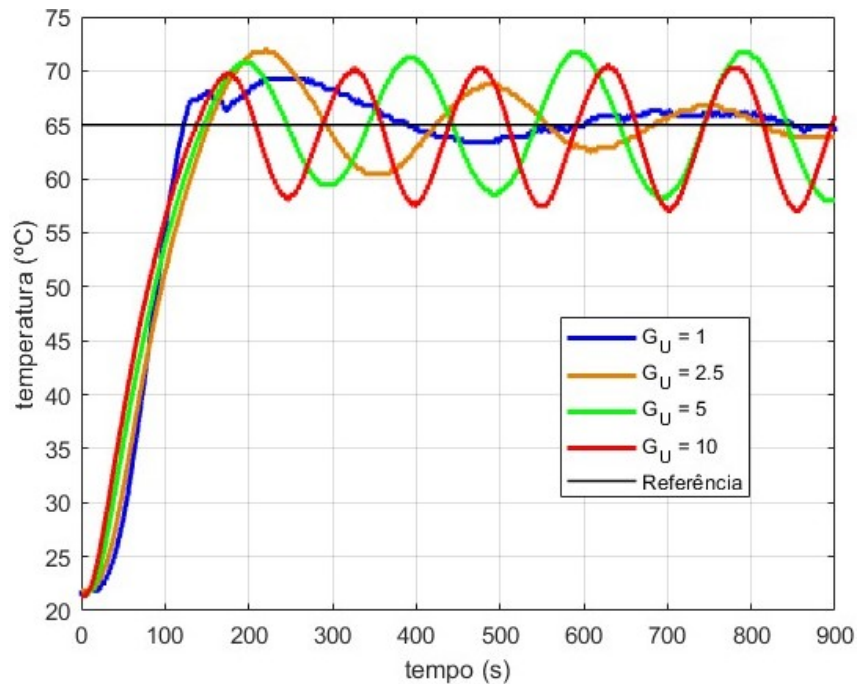


Figura 4.13: Esquema do FLC com Controlo I com *anti-windup*

Aplicando o esquema modificado, os resultados dos dois sistemas são representados pelos gráficos da Figura 4.14.



(a) Sistema 1



(b) Sistema 2

Figura 4.14: Controle I por FLC com *anti-windup*

A aplicação do controle I com *anti-windup* no sistema 1, Figura 4.14a, permitiu alcançar a estabilidade na temperatura de referência após alguma oscilação e, perante a resposta do esquema inicial, Figura 4.12a, uma redução geral do *overshoot* para os vários valores de ganho G_U . O valor inicial do ganho, $G_U = 1$, comparativamente aos restantes ganhos, possui um maior tempo de subida, tornando a resposta mais lenta. Para $G_U = \{2.5, 5, 10\}$, apesar da otimização da resposta pela aplicação do esquema com *anti-windup*, o método contém um *overshoot* e um tempo de estabilização elevados, não sendo qualificado como método eficiente para este sistema.

No sistema 2, Figura 4.14b, a aplicação do esquema *anti-windup* permite reduzir o *overshoot* e aproximar a resposta um pouco à temperatura de referência. No entanto, devido à sua resposta mais lenta à variação de controle, o modelo do FLC com controle I não consegue alcançar a estabilização para nenhum dos valores de ganho, no intervalo de tempo atribuído às experiências.

4.1.3 Controle PD

Num controlador PD, é necessário na entrada do FLC considerar o valor do erro e da derivada do erro, que representa a sua variação. Na saída do FLC, é aplicado um ganho G_U sobre o valor U , de modo a otimizar a resposta. Este controlador é ilustrado no esquema da Figura 4.15 [29, 36, 37, 38].

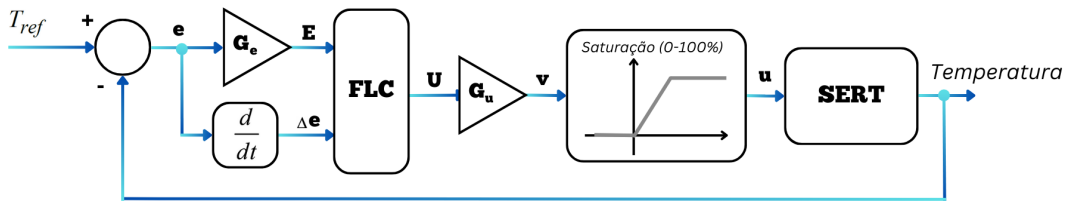


Figura 4.15: Esquema do FLC com Controle PD

Para este tipo de esquema, aplicando apenas o controle PD, são produzidas vinte e cinco regras, havendo a interação entre as cinco funções de pertinência das duas entradas. O valor assumido pelas funções de pertinência do valor de controle (saída), é apresentado na Tabela 4.4.

Tabela 4.4: Lista de Regras do Controle PD

$\Delta e \backslash e$	NG	NP	Z	PP	PG
NG	Z	Z	Z	P	M
NP	Z	Z	Z	P	M
Z	Z	Z	Z	P	G
PP	Z	Z	P	M	G
PG	Z	Z	P	M	MG

A representação deste controle, sendo dependente de duas variáveis de entrada (variação do erro - Δe , erro - e), é dada por uma superfície, Figura 4.16.

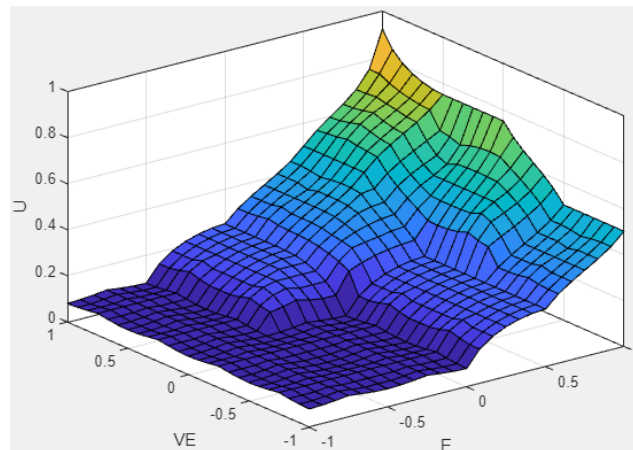
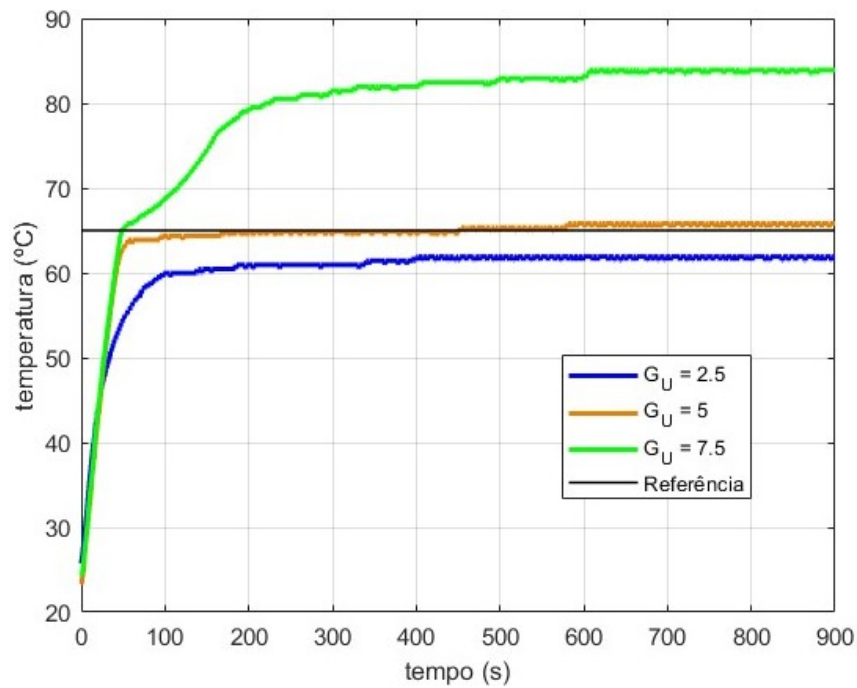
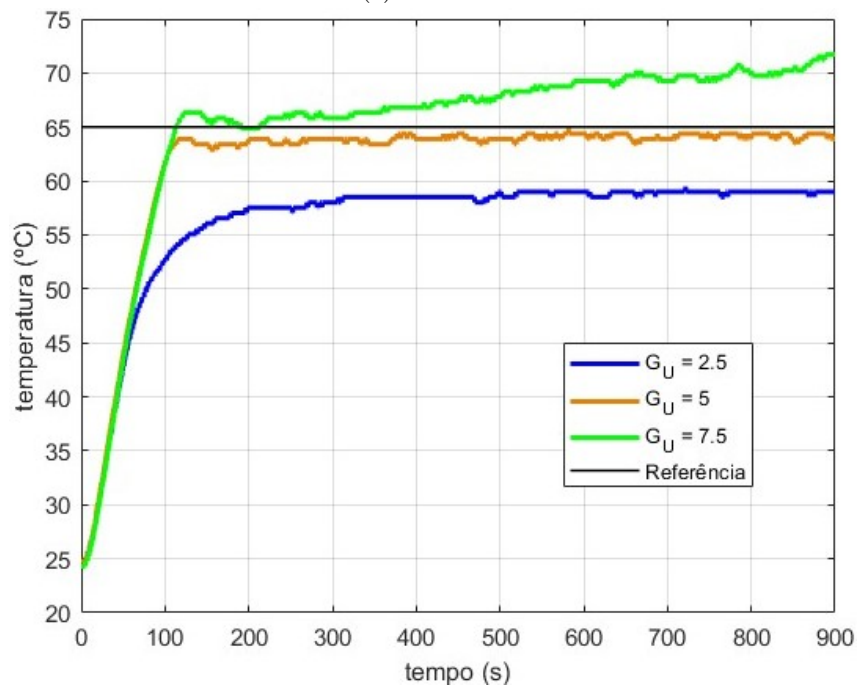


Figura 4.16: Superfície de Controle - PD

Com a estrutura e superfície de controle apresentadas, as experiências efetuadas do sistema 1 e sistema 2 envolveram os ganhos $G_U = \{2.5, 5.0, 7.5\}$. Os resultados destas experiências são apresentados na Figura 4.17.



(a) Sistema 1



(b) Sistema 2

Figura 4.17: Controle PD por FLC

Assim, tal como na aplicação do controle P, é possível indicar que o controle PD não é o indicado para o controle de temperatura. Apesar de que no sistema 1, Figura 4.9a, e de no sistema 2, Figura 4.9b, ao aplicar um ganho de $G_U = 5.0$, a resposta corresponder a um valor exato ou muito próxima da temperatura de referência, é

possível verificar que este modelo é altamente dependente do ajuste direto de G_U para a correção do erro em regime permanente.

4.1.4 Controlo PI

Num controlador PI, é necessário indicar, na entrada do FLC, o valor do erro e a derivada do erro, tal como no controlador PD. Na saída do FLC, é aplicado um ganho G_U sobre o valor U , de modo a otimizar a resposta. Este controlador é representado pelo esquema da Figura 4.18 [34, 36, 38].

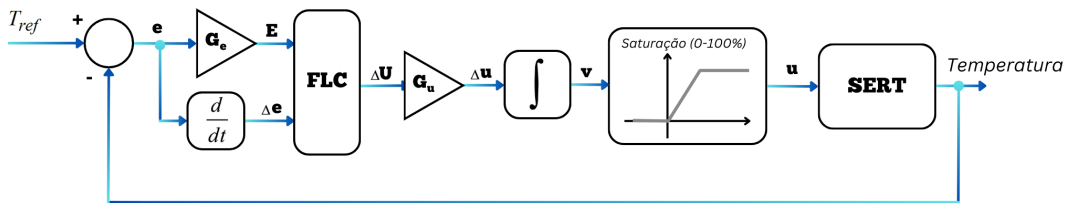


Figura 4.18: Esquema do FLC com Controlo PI

Para este tipo de esquema, aplicando apenas o controlo PI, são produzidas vinte e cinco regras, havendo a interação entre as cinco funções de pertinência das duas entradas. O valor assumido pelas funções de pertinência do valor de controlo (saída), é apresentado na Tabela 4.5.

Tabela 4.5: Lista de Regras do Controlo PI

$\Delta e \backslash e$	NG	NP	Z	PP	PG
NG	NG	NG	NG	NP	Z
NP	NG	NG	NP	Z	PP
Z	NG	NP	Z	PP	PG
PP	NP	Z	PP	PG	PG
PG	Z	PP	PG	PG	PG

A representação deste controlo, sendo dependente de duas variáveis de entrada (variação do erro - Δe , erro - e), é dada por uma superfície, Figura 4.19.

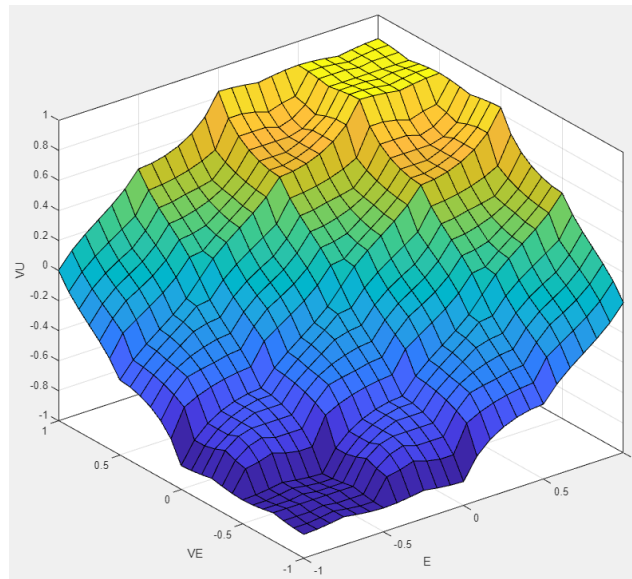
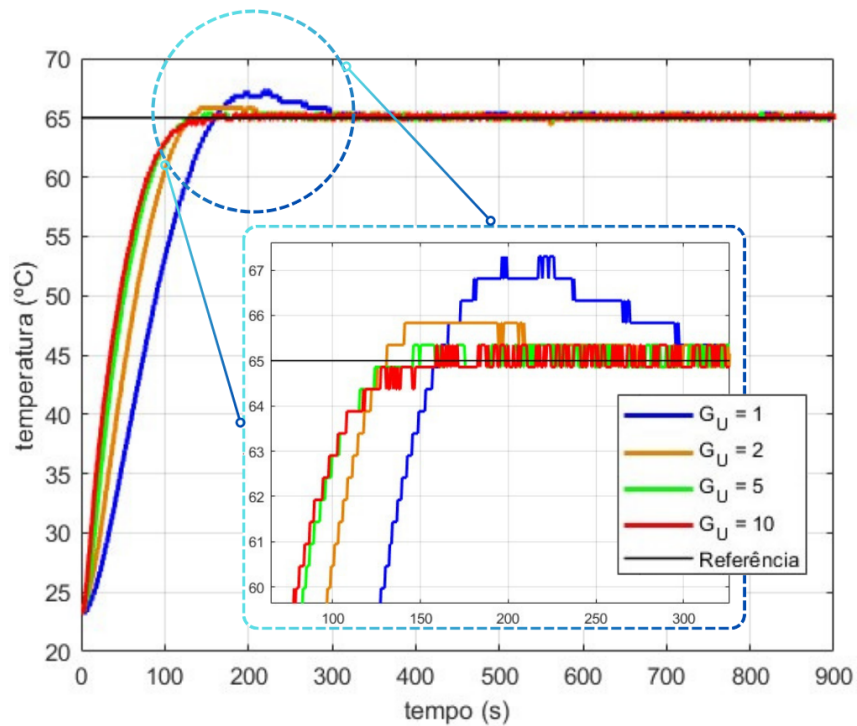


Figura 4.19: Superfície de Controllo - PI

Com a estrutura e superfície de controllo apresentadas, as experiências efetuadas do sistema 1 e sistema 2 envolveram os ganhos $G_U = \{1, 2, 5, 10\}$. Os resultados destas experiências são apresentados na Figura 4.20.



(a) Sistema 1

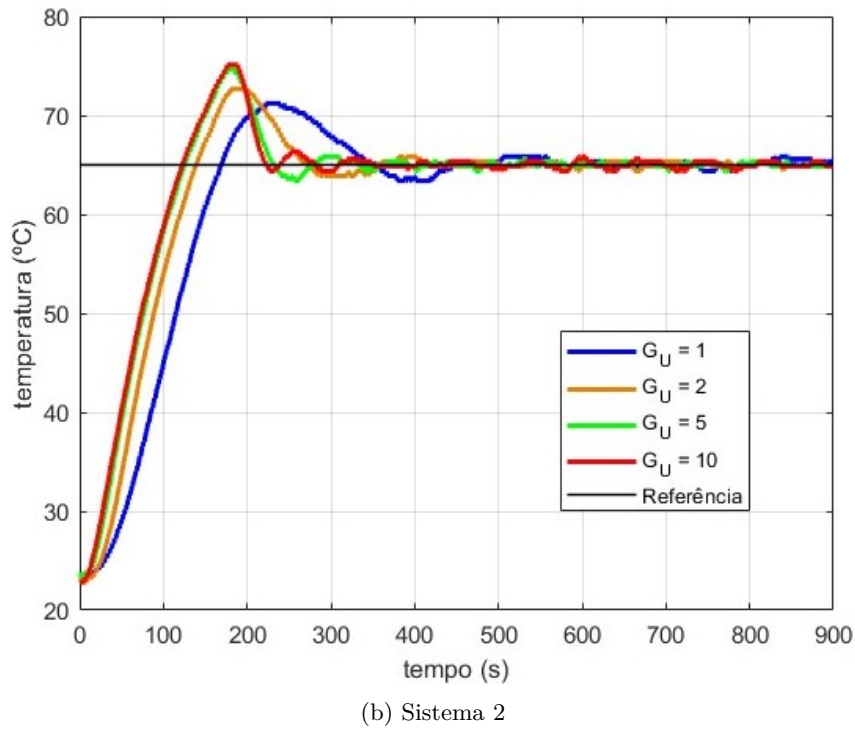
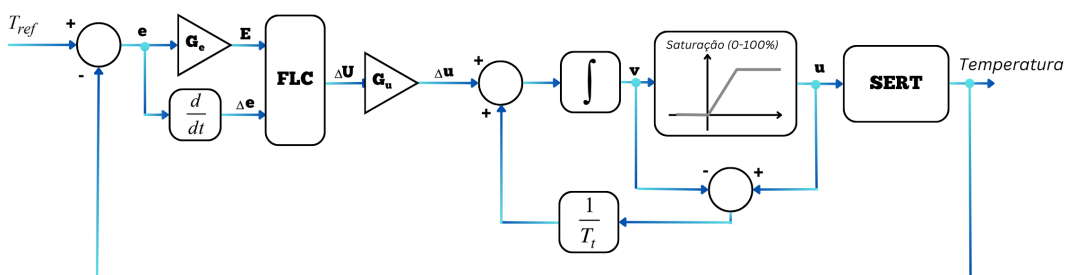


Figura 4.20: Controlo PI por FLC

Observando os gráficos, no sistema 1, para os ganhos $G_U = \{5, 10\}$, verifica-se uma resposta sem *overshoot* e estável ao alcançar a temperatura de referência. Enquanto isso, o sistema 2, apesar do tempo de subida reduzido, todos os ganhos apresentam *overshoot*. Para melhorar a resposta deste sistema, face à resposta do sistema 1, é aplicado um esquema *anti-windup*, como apresentado na Figura 3.28, aplicando o parâmetro T_t utilizado na subsecção 3.3.8.

Figura 4.21: Esquema do FLC com Controlo PI com *anti-windup*

Aplicando os valores de ganho $G_U = \{1, 5, 10, 20\}$ e o valor de T_t obtido na subsecção 4.1.8 para o controlador PI, é obtido o gráfico da Figura 4.22. Com a aplicação deste esquema, para os ganhos $G_U = \{10, 20\}$ é verificada a eliminação completa do *overshoot*.

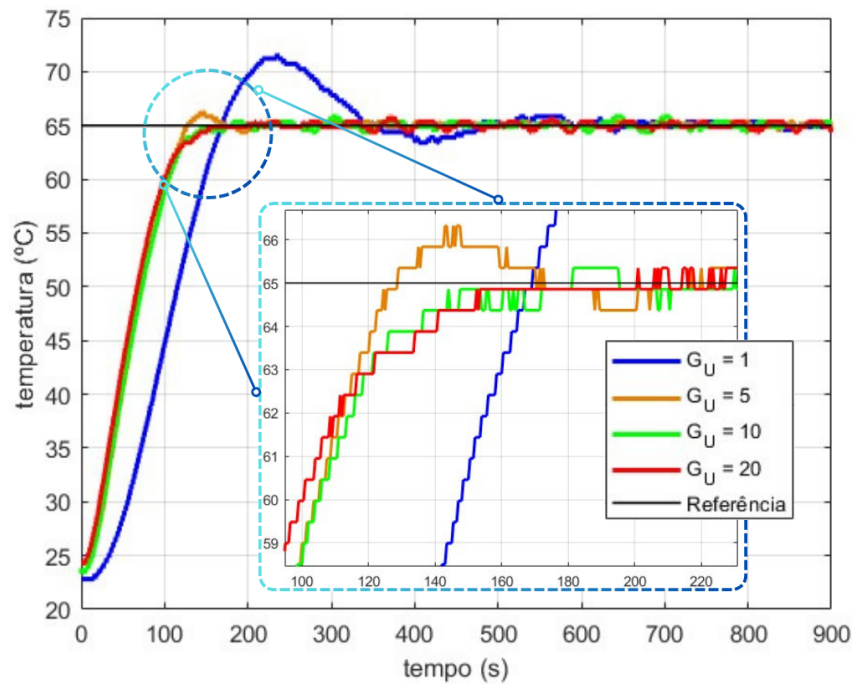


Figura 4.22: Controle PI por FLC com *anti-windup* do Sistema 2

Sendo o controle PI o primeiro método a ter respostas de tempo de subida reduzido, sem *overshoot* e estável, é calculado o valor do erro de cada critério. Destacando o valor de ganho que obtém o melhor valor, são apresentados na Tabela 4.6 os valores dos critérios de erro para o sistema 1, e na Tabela 4.7 para o sistema 2.

Tabela 4.6: Erros FLC de controle PI - Sistema 1

Critério	$G_U = 1$	$G_U = 2$	$G_U = 5$	$G_U = 10$
<i>IAE</i>	6955	5085	4077	3742
<i>ISE</i>	185.6×10^3	131.7×10^3	99.4×10^3	85.2×10^3
<i>ITAE</i>	55.1×10^4	36.4×10^4	29.2×10^4	27.9×10^4
<i>ITSE</i>	663.4×10^4	347.5×10^4	210.0×10^4	168.9×10^4

Tabela 4.7: Erros FLC de controle PI - Sistema 2

Critério	$G_U = 1$	$G_U = 5$	$G_U = 10$	$G_U = 20$
<i>IAE</i>	10047	5480	5589	5172
<i>ISE</i>	273.5×10^3	149.0×10^3	149.5×10^3	132.6×10^3
<i>ITAE</i>	106.1×10^4	37.9×10^4	42.5×10^4	39.4×10^4
<i>ITSE</i>	1361×10^4	430.1×10^4	433.7×10^4	362.3×10^4

4.1.5 Controle PD+I

O controlador PD+I, que aplica um de controle difuso PD com uma atuação integrativa (I), pode ser considerado como um método híbrido. Na saída do FLC, neste caso, não é aplicado o ganho G_U , no entanto seria possível ser adicionado para uma maior dinâmica do sistema. A atuação integrativa passa por um ganho G_I , de modo a ajustar a sua influência na resposta, como apresentado no esquema da Figura 4.23 [29, 34].

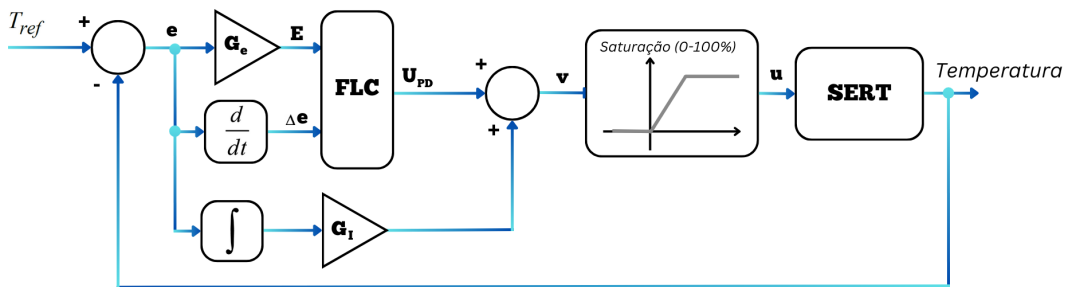
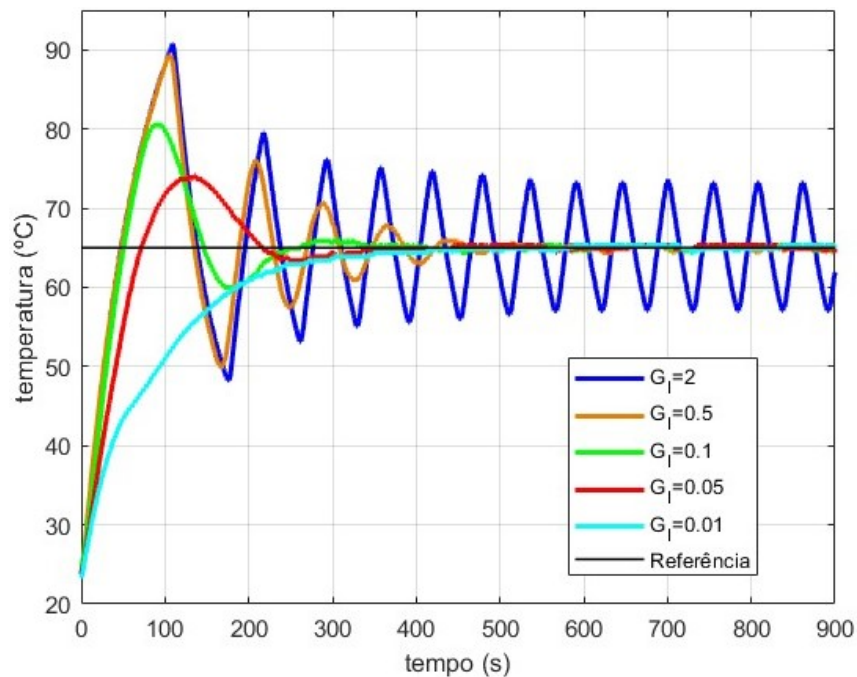


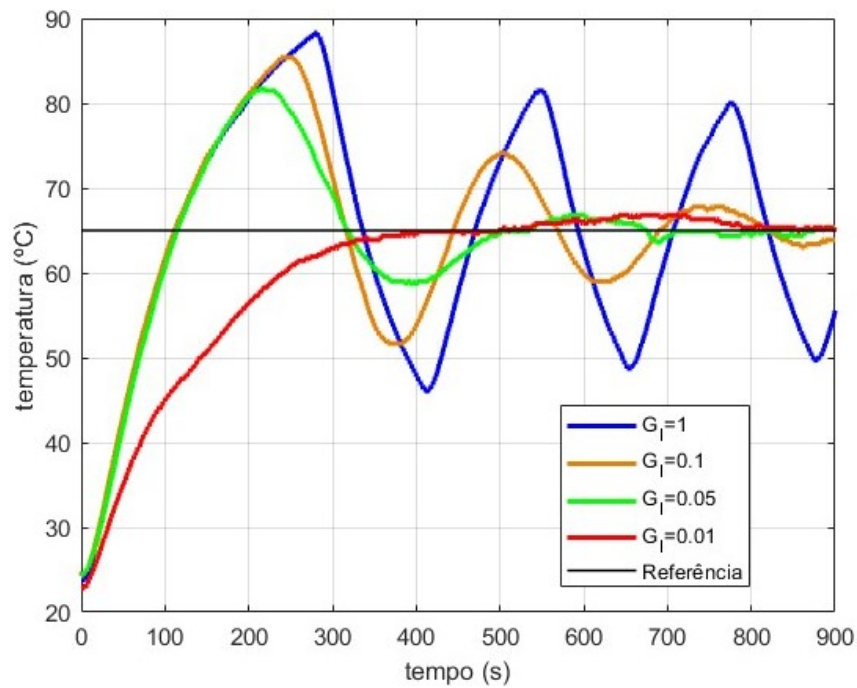
Figura 4.23: Esquema do FLC com Controle PD+I

A lista de regras e superfície de controle deste FLC, sendo aplicado um controle PD, são iguais ao da subsecção 4.1.3.

Aplicando o devido controle, e variando o valor do ganho G_I , perante cada experiência, são obtidos os resultados ilustrados nos gráficos da Figura 4.24.



(a) Sistema 1



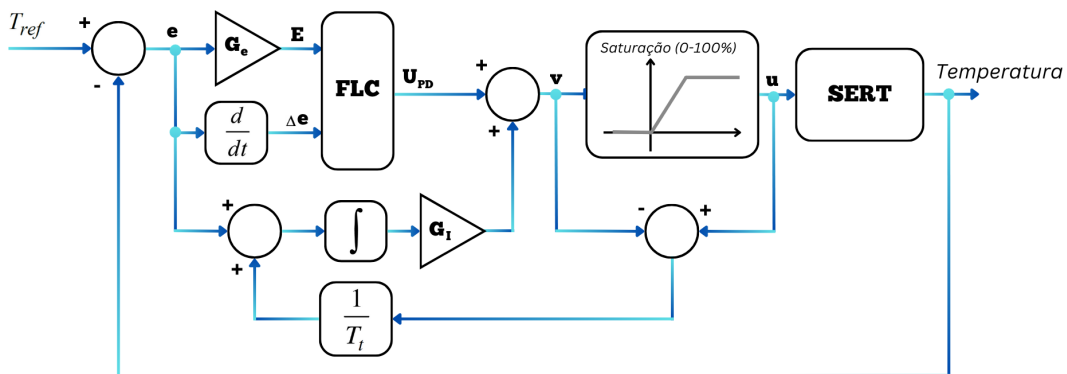
(b) Sistema 2

Figura 4.24: Controle PD+I por FLC

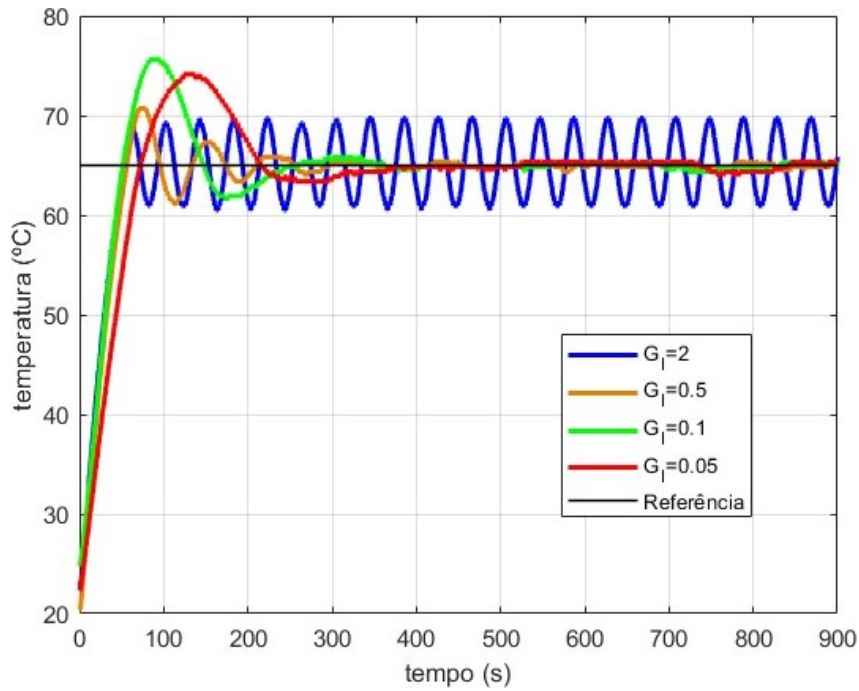
No sistema 1, aplicando os valores de ganho $G_I = \{0.01, 0.05, 0.1, 0.5, 2\}$, é possível observar que o valor com o melhor desempenho, face ao seu *overshoot* e tempo de subida, é $G_I = 0.05$. No entanto, face a análises anteriores, não é um resultado com valores aceitáveis para uma resposta ótima do sistema.

No sistema 2, aplicando os valores de ganho $G_I = \{0.01, 0.1, 0.5, 1\}$, é possível concluir que face ao atraso presente neste sistema, este método não é o mais indicado.

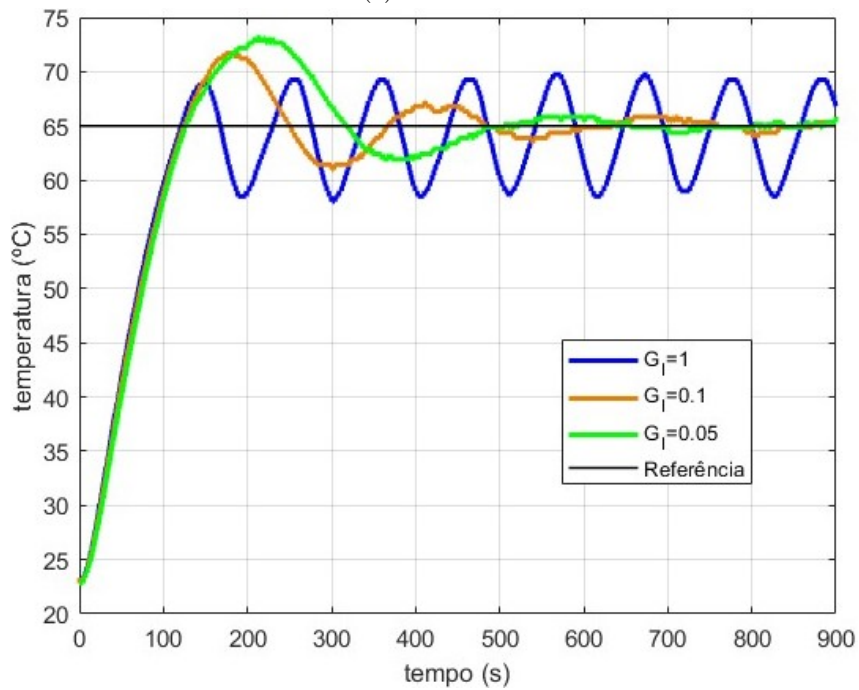
Nestes sistemas, para tentar combater o seu *overshoot*, é aplicado um esquema *anti-windup*, como apresentado na Figura 4.25, aplicando o parâmetro T_t utilizado na subsecção 3.3.8.

Figura 4.25: Esquema do FLC do Controle PD+I com *anti-windup*

Aplicando o devido controlo, com os melhores valores do ganho G_I observados, são obtidos os resultados apresentados nos gráficos da Figura 4.26.



(a) Sistema 1



(b) Sistema 2

Figura 4.26: Controlo PD+I por FLC com *anti-windup*

Com esta modificação do esquema, apesar de não conseguir eliminar o *overshoot*, nem a oscilação presente nas respostas, conseguiu-se melhorar consideravelmente a

amplitude da oscilação. Perante estas observações é possível indicar que este método não é o indicado para o controlo de temperatura.

4.1.6 Controlo PID por um FLC

O controlador PID por um FLC é constituído por um FLC de controlo PD, como no esquema da Figura 4.27. O valor de controlo obtido, é multiplicado pelo parâmetro K_{PD} para obter a componente proporcional e derivativa, e um integral seguido do parâmetro K_{PI} para obter a componente integrativa, de forma a obter o valor de controlo PID [37].

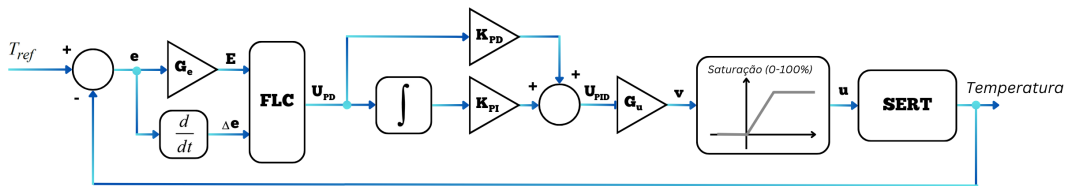
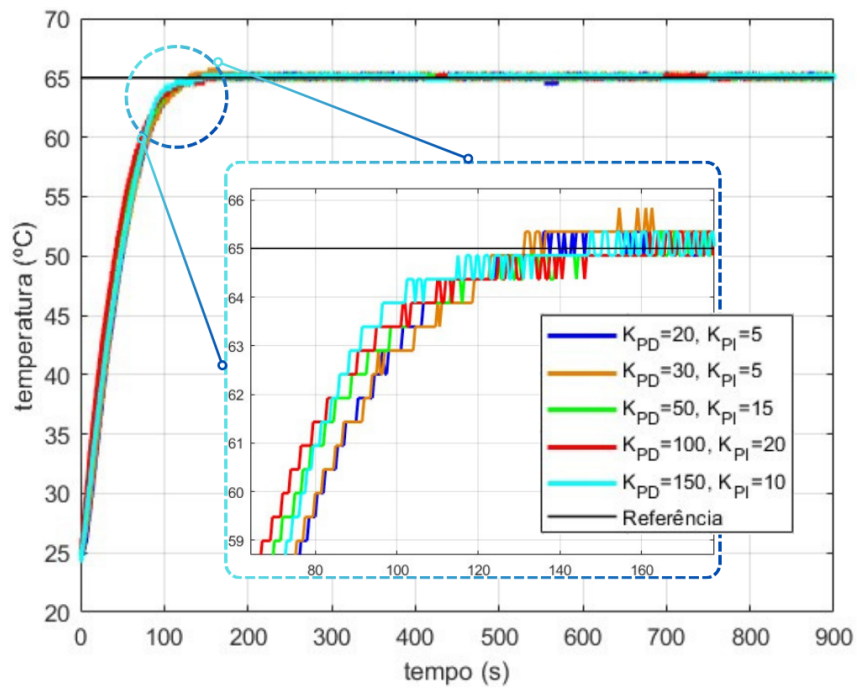


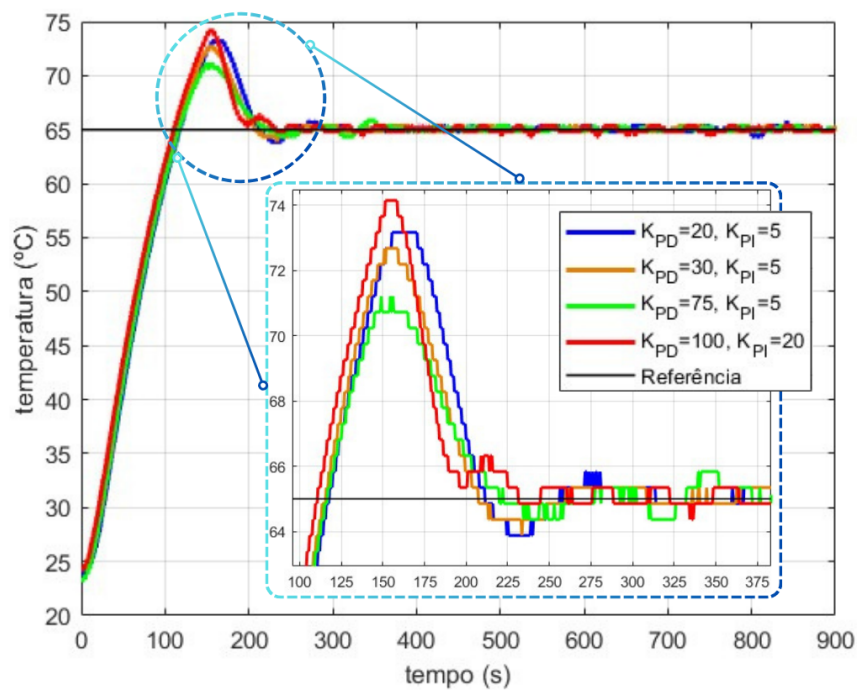
Figura 4.27: Esquema de Controlo PID por um FLC

A lista de regras e superfície de controlo deste FLC, sendo aplicado um controlo PD, são iguais ao da subsecção 4.1.3.

Aplicando o devido controlo e variando o valor dos parâmetros K_{PD} e K_{PI} perante cada experiência, os resultados são apresentados nos gráficos da Figura 4.28. No sistema 1 são aplicados cinco conjuntos de parâmetros, $\{[K_{PD} = 20, K_{PI} = 5], [K_{PD} = 30, K_{PI} = 5], [K_{PD} = 50, K_{PI} = 15], [K_{PD} = 100, K_{PI} = 20], [K_{PD} = 150, K_{PI} = 10]\}$, enquanto que no sistema 2 são aplicados quatro conjuntos, $\{[K_{PD} = 20, K_{PI} = 5], [K_{PD} = 30, K_{PI} = 5], [K_{PD} = 75, K_{PI} = 5], [K_{PD} = 100, K_{PI} = 20]\}$.



(a) Sistema 1



(b) Sistema 2

Figura 4.28: Controlo PID por um FLC

O sistema 1 apresenta uma resposta muito similar para qualquer um dos conjuntos de parâmetros aplicados apresentando uma boa resposta, isenta de *overshoot*

e oscilação.

No sistema 2, todas as respostas apresentam *overshoot*, estabilizando ao alcançar novamente a temperatura de referência. Entre estas, é possível observar um menor valor de *overshoot* para o conjunto de parâmetros [$K_{PD} = 75, K_{PI} = 5$].

Os quatro critérios de erro do sistema 1 e sistema 2, são apresentados nas Tabelas 4.8 e 4.9, respetivamente. A melhor resposta presente será utilizada para comparação com os restantes métodos.

Tabela 4.8: Erros do controlo PID com um FLC - Sistema 1

Critério	$K_{PD} = 20,$ $K_{PI} = 5$	$K_{PD} = 30,$ $K_{PI} = 5$	$K_{PD} = 50,$ $K_{PI} = 15$	$K_{PD} = 100,$ $K_{PI} = 20$	$K_{PD} = 150,$ $K_{PI} = 10$
<i>IAE</i>	3923	3894	3418	3308	3663
<i>ISE</i>	91.9×10^3	89.1×10^3	72.2×10^3	68.4×10^3	82.5×10^3
<i>ITAE</i>	28.8×10^4	29.1×10^4	26.7×10^4	26.3×10^4	27.8×10^4
<i>ITSE</i>	193.8×10^4	188.5×10^4	138.3×10^4	128.9×10^4	170.0×10^4

Tabela 4.9: Erros do controlo PID com um FLC - Sistema 2

Critério	$K_{PD} = 20,$ $K_{PI} = 5$	$K_{PD} = 30,$ $K_{PI} = 5$	$K_{PD} = 75,$ $K_{PI} = 5$	$K_{PD} = 100,$ $K_{PI} = 20$
<i>IAE</i>	6186	5886	5810	5640
<i>ISE</i>	150.5×10^3	142.9×10^3	143.4×10^3	131.6×10^3
<i>ITAE</i>	51.9×10^4	47.1×10^4	46.0×10^4	46.2×10^4
<i>ITSE</i>	497.3×10^4	450.0×10^4	430.0×10^4	413.6×10^4

4.1.7 Controlo PID por dois FLC

O controlador PID por dois FLC é constituído por um FLC de controlo PD e um FLC de controlo PI, como no esquema da Figura 4.29. Com a soma dos valores de controlo, devidas às suas ações individuais, será obtido o valor de controlo PID [37].

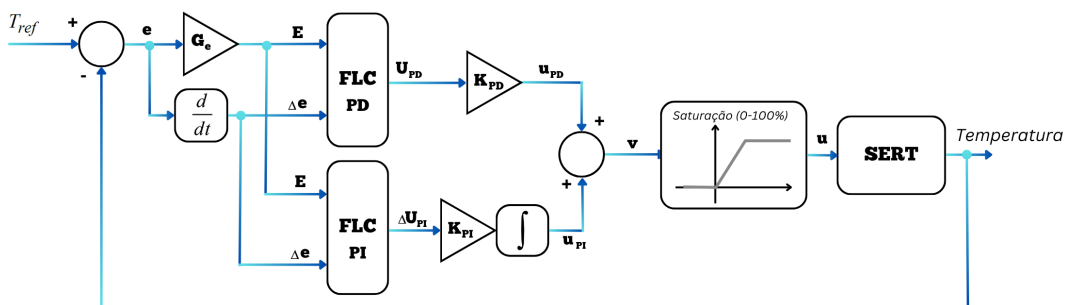
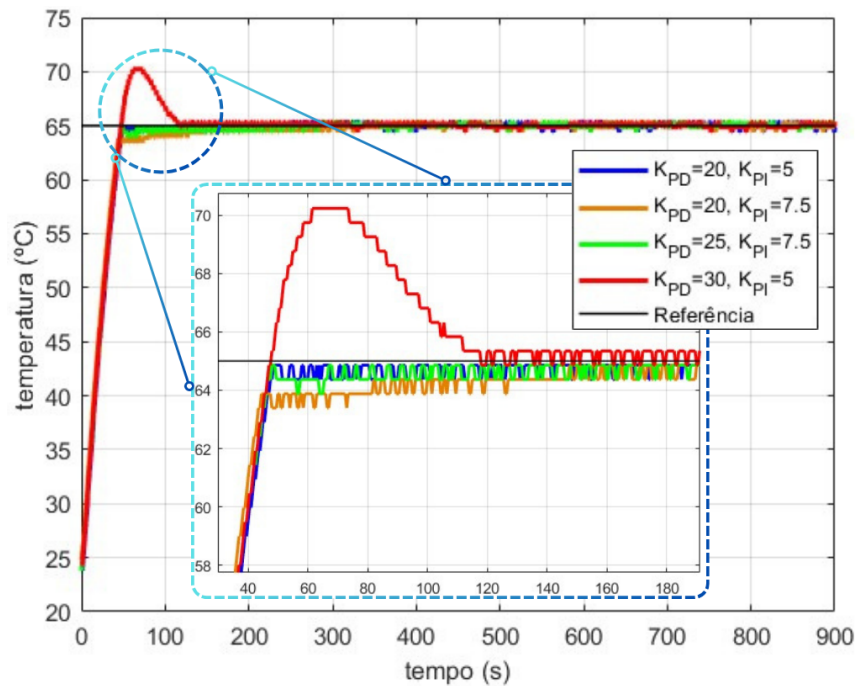


Figura 4.29: Esquema de Controlo PID por dois FLC

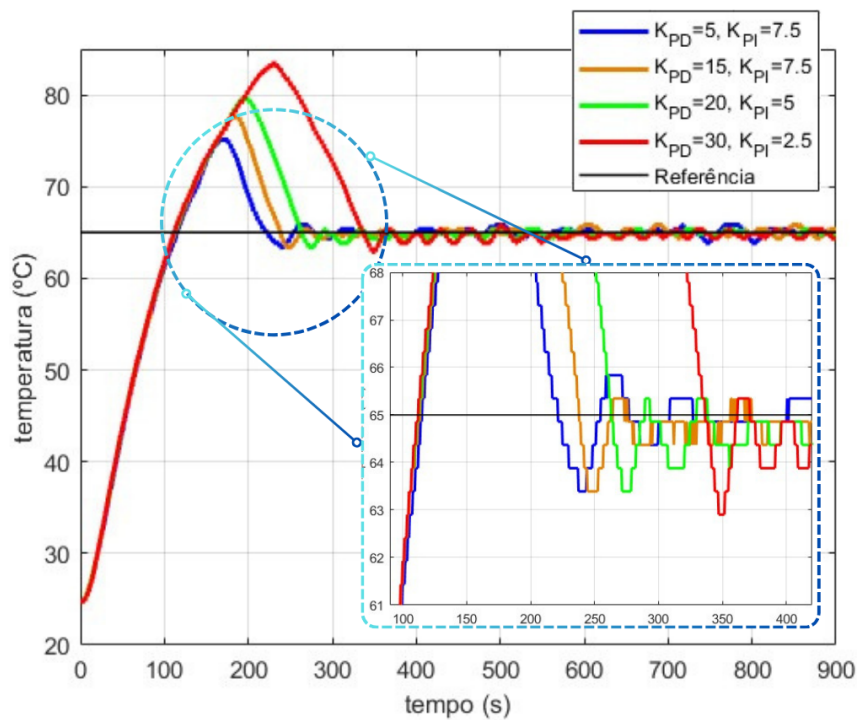
As listas de regras e superfícies de controlo destes FLC, sendo aplicado um controlo PD e PI, são iguais ao da subsecção 4.1.3 e 4.1.4, respetivamente.

Aplicando o devido controlo e variando o valor dos parâmetros K_{PD} e K_{PI} perante cada experiência, os resultados são apresentados nos gráficos da Figura 4.30. No sistema 1 e sistema 2 são aplicados quatro conjuntos de parâmetros, sendo estes:

- Para o sistema 1 - $\{[K_{PD} = 20, K_{PI} = 5], [K_{PD} = 20, K_{PI} = 7.5], [K_{PD} = 25, K_{PI} = 7.5], [K_{PD} = 30, K_{PI} = 5]\}$.
- Para o sistema 2 - $\{[K_{PD} = 5, K_{PI} = 7.5], [K_{PD} = 15, K_{PI} = 7.5], [K_{PD} = 20, K_{PI} = 5], [K_{PD} = 30, K_{PI} = 2.5]\}$.



(a) Sistema 1



(b) Sistema 2

Figura 4.30: Controle PID por dois FLC

O sistema 1 apresenta uma resposta muito similar, exceto para o conjunto $[K_{PD} = 30, K_{PI} = 5]$, que apresenta *overshoot*. Para qualquer um dos restantes conjuntos de parâmetros aplicados, apresentam uma boa resposta, isenta de *overshoot* e oscilação.

No sistema 2, todas as respostas apresentam *overshoot*, mantendo uma ligeira oscilação ao alcançar novamente a temperatura de referência. Entre estas, é possível observar um menor valor de *overshoot* para o conjunto de parâmetros $[K_{PD} = 5, K_{PI} = 7.5]$.

Para os quatro critérios de erro do sistema 1 e sistema 2, são utilizadas as Tabelas 4.10 e 4.11, respetivamente, para a apresentação dos seus valores. A melhor resposta presente será utilizada para comparação com os restantes métodos.

Tabela 4.10: Erros do controlo PID com dois FLC - Sistema 1

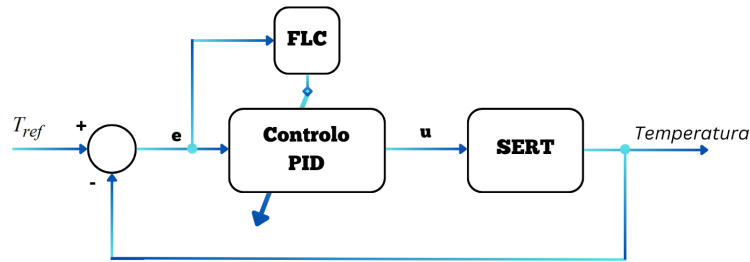
Critério	$K_{PD} = 20,$ $K_{PI} = 5$	$K_{PD} = 20,$ $K_{PI} = 7.5$	$K_{PD} = 25,$ $K_{PI} = 7.5$	$K_{PD} = 30,$ $K_{PI} = 5$
<i>IAE</i>	2249	2247	2225	2594
<i>ISE</i>	50.4×10^3	45.5×10^3	48.7×10^3	50.0×10^3
<i>ITAE</i>	21.1×10^4	21.6×10^4	20.6×10^4	24.0×10^4
<i>ITSE</i>	60.2×10^4	53.7×10^4	57.5×10^4	69.1×10^4

Tabela 4.11: Erros do controlo PID com dois FLC - Sistema 2

Critério	$K_{PD} = 20,$ $K_{PI} = 5$	$K_{PD} = 20,$ $K_{PI} = 7.5$	$K_{PD} = 25,$ $K_{PI} = 7.5$	$K_{PD} = 30,$ $K_{PI} = 5$
IAE	6163	6655	7324	9686
ISE	135.4×10^3	141.1×10^3	151.2×10^3	188.0×10^3
ITAE	61.5×10^4	71.3×10^4	82.7×10^4	151.0×10^4
ITSE	494.3×10^4	625.5×10^4	834.80×10^4	1756×10^4

4.1.8 Controlo PID por *Gain Scheduling*

O método de controlo PID por *Gain Scheduling* utiliza um controlador PID convencional. Neste caso, o FLC é utilizado numa dinâmica diferente, controlando diretamente os parâmetros do controlador PID: K_P , K_I e K_D . De uma forma simplificada, na Figura 4.31, é apresentado o esquema de controlo deste método, variando os parâmetros para encontrar a melhor resposta possível [34, 39, 40].

Figura 4.31: Esquema simples do Controlo PID por *Gain Scheduling*

No sistema deste FLC, Figura 4.32, de forma a ajustar-se os parâmetros do controlo PID, é utilizado o valor do erro e a sua variação como entradas. Indiretamente, este controla os parâmetros K_P , K_I e K_D , no entanto a saída do FLC dará os valores normalizados K'_P e K'_D , bem como a constante α .

De forma a existir o controlo de K_P e K_D , é necessário impor os limites $[K_{P_{\min}}, K_{P_{\max}}]$ e $[K_{D_{\min}}, K_{D_{\max}}]$, sendo estes calculados através do ganho crítico (K_u) e do período crítico (P_u) [39], aplicando:

$$K_{P_{\min}} = 0.32K_u, \quad K_{P_{\max}} = 0.6K_u \quad (4.4)$$

$$K_{D_{\min}} = 0.08K_uP_u, \quad K_{D_{\max}} = 0.15K_uP_u \quad (4.5)$$

Para o controlo do sistema, aplicaram-se os valores de K_u e P_u calculados na subsecção 4.1.1 e 4.1.4.

Enquanto isso, os valores K'_P e K'_D normalizados entre 0 e 1, para facilidade de processamento, são definidos por:

$$K'_P = \frac{K_P - K_{P_{\min}}}{K_{P_{\max}} - K_{P_{\min}}} \quad (4.6)$$

$$K'_D = \frac{K_D - K_{D_{\min}}}{K_{D_{\max}} - K_{D_{\min}}} \quad (4.7)$$

O sistema difuso do método de *gain scheduling*, nas entradas analisa sete funções de pertinência de forma triangular em cada entrada. Os valores de ganho, K'_P e K'_D , são obtidos por duas funções gaussianas, e a constante α por um conjunto de valores unitários (2, 3, 4, 5).

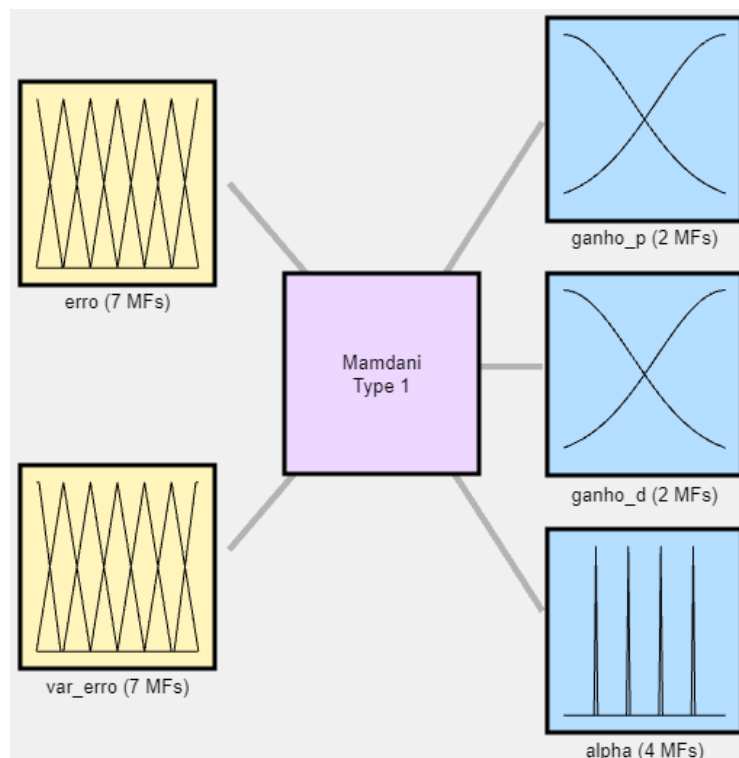


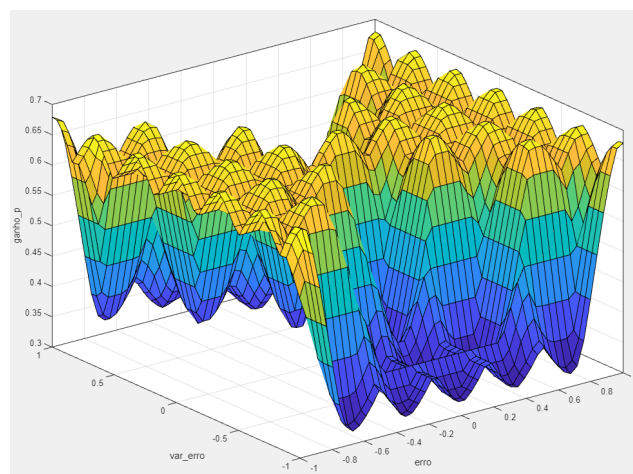
Figura 4.32: Sistema Difuso - *Gain Scheduling*

Para a obtenção dos parâmetros, é utilizado um conjunto de regras que formam três tabelas 7×7 , ou seja, com 49 regras.

Na Tabela 4.12, são apresentados os valores obtidos para o parâmetro K'_P , que origina a superfície de controle apresentada na Figura 4.33.

Tabela 4.12: Lista de Regras do Ganho P

$\Delta e \backslash e$	NG	NM	NP	Z	PP	PM	PG
NG	G	P	P	P	P	P	G
NM	G	G	P	P	P	G	G
NP	G	G	G	P	G	G	G
Z	G	G	G	G	G	G	G
PP	G	G	G	P	G	G	G
NM	G	G	P	P	P	G	G
PG	G	P	P	P	P	P	G

Figura 4.33: Superfície de Controle do *Gain Scheduling* - Ganho P

Na Tabela 4.13, são apresentados os valores obtidos para o parâmetro K'_D , que origina a superfície de controle apresentada na Figura 4.34.

Tabela 4.13: Lista de Regras do Ganho D

$\Delta e \backslash e$	NG	NM	NP	Z	PP	PM	PG
NG	P	G	G	G	G	G	P
NG	P	G	G	G	G	G	P
NG	P	P	G	G	G	P	P
NG	P	P	P	G	P	P	P
NG	P	P	G	G	G	P	P
NG	P	G	G	G	G	G	P
NG	P	G	G	G	G	G	P

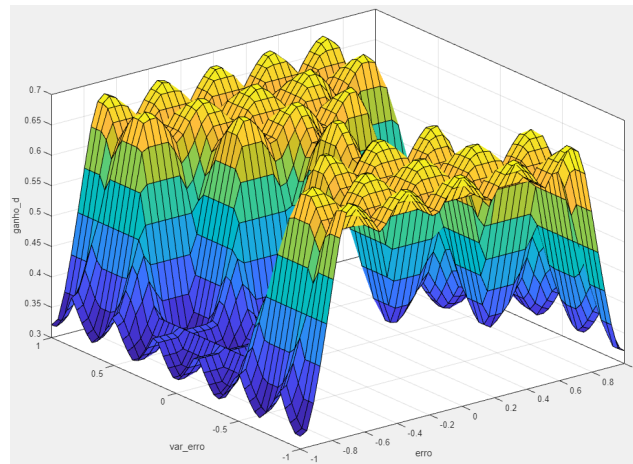


Figura 4.34: Superfície de Controlo do *Gain Scheduling* - Ganho D

Na Tabela 4.14, são apresentados os valores obtidos para a constante α , que origina a superfície de controlo apresentada na Figura 4.35.

Tabela 4.14: Lista de Regras da constante α

$\Delta e \setminus e$	NG	NM	NP	Z	PP	PM	PG
NG	2	3	4	5	4	3	2
NG	2	3	3	4	3	3	2
NG	2	2	3	3	3	2	2
NG	2	2	2	3	2	2	2
NG	2	2	3	3	3	2	2
NG	2	3	3	4	3	3	2
NG	2	3	4	5	4	3	2

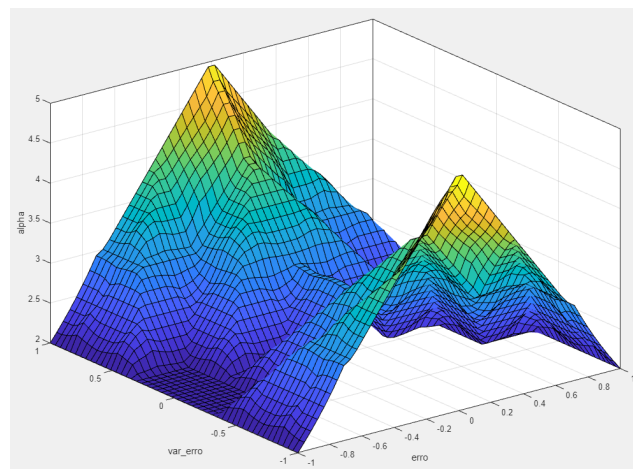


Figura 4.35: Superfície de Controlo do *Gain Scheduling* - α

Com o sistema apresentado, o esquema necessário para controlar o SERT, seguido o conceito de um controlo PID convencional, é destacado pela Figura 4.36 [39].

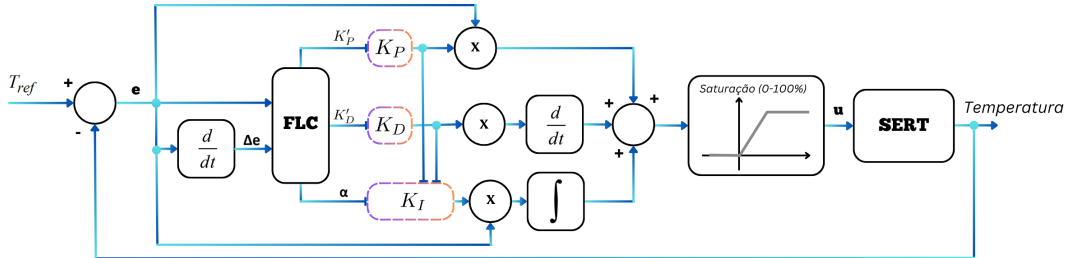


Figura 4.36: Esquema de Controlo PID por *Gain Scheduling*

Para o cálculo dos parâmetros do controlo PID (K_P , K_I e K_D), tem-se que [39]:

Como $T_I = \alpha T_D$, vem:

$$\begin{aligned} K_P &= (K_{P_{m\acute{a}x}} - K_{P_{m\acute{i}n}})K'_P + K_{P_{m\acute{i}n}} \\ K_D &= (K_{D_{m\acute{a}x}} - K_{D_{m\acute{i}n}})K'_D + K_{D_{m\acute{i}n}} \\ K_I &= \frac{K_P}{T_I} = \frac{K_P}{\alpha T_D} = \frac{K_P^2}{\alpha K_D} \end{aligned} \quad (4.8)$$

Aplicando o controlo convencional PID com os parâmetros obtidos, o sistema 1 e o sistema 2 do SERT, apresentam o comportamento ilustrado na Figura 4.37.

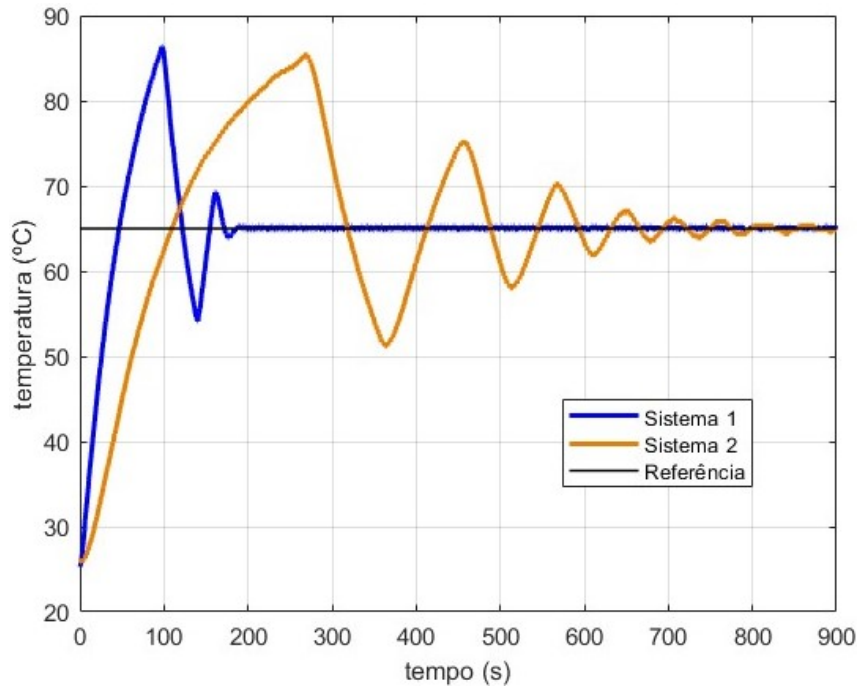


Figura 4.37: Controlo PID por *Gain Scheduling*

A resposta, para ambos os sistemas, apresenta um elevado *overshoot* e, no caso do sistema 2, apenas consegue estabilizar perto do fim do período de experiência. Assim, tal como nos controladores convencionais, para melhorar a resposta é aplicado um esquema *anti-windup*, como apresentado na Figura 4.38, aplicando o parâmetro T_t utilizado na subsecção 3.3.8.

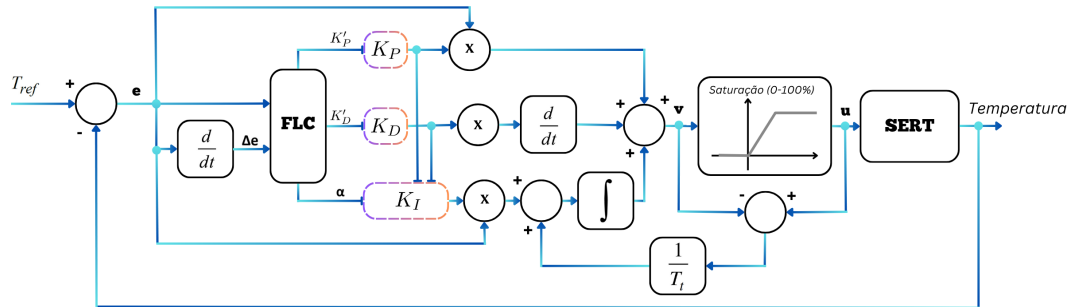


Figura 4.38: Esquema de Controlo PID por *Gain Scheduling* com *Anti-windup*

Como seria expectável, observando a Figura 4.39, a resposta dos sistemas é melhorada consideravelmente. No caso do sistema 1 elimina completamente o *overshoot* e quando alcança a temperatura de referência mantém o seu valor estável. No entanto, no sistema 2, apesar de ser uma resposta melhorada, este permanece em oscilação com uma amplitude reduzida de, aproximadamente, 3 °C.

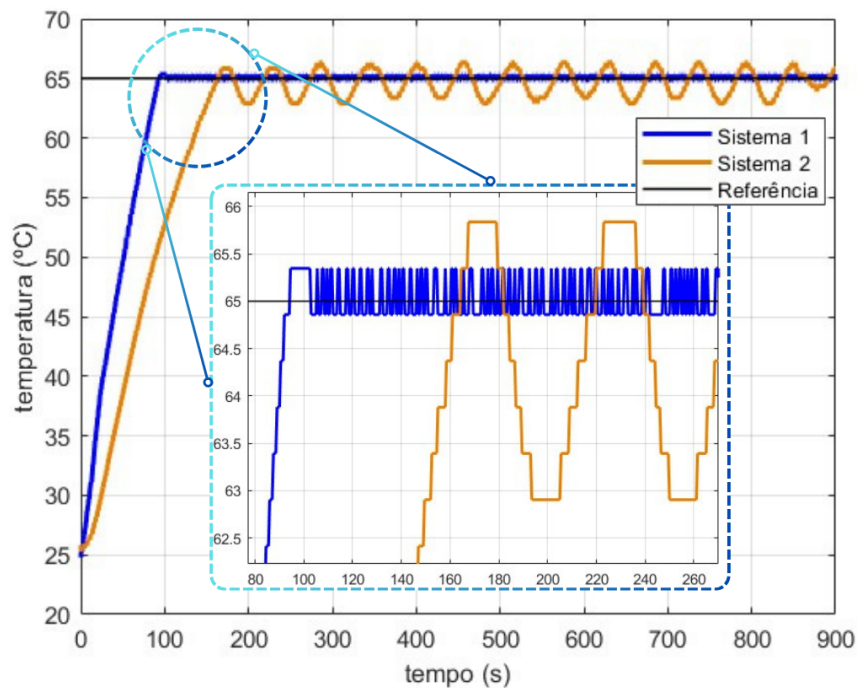


Figura 4.39: Controlo Fuzzy - *Gain Scheduling* com *Anti-windup*

Para posterior análise dos critérios de erro presentes neste método, é apresentado na Tabela 4.15 o valor dos mesmos.

Tabela 4.15: Erros FLC de controlo PID por *Gain Scheduling*

Critério	Sistema 1	Sistema 2
<i>IAE</i>	3678	7531
<i>ISE</i>	84.6×10^3	164.4×10^3
<i>ITAE</i>	27.1×10^4	107.9×10^4
<i>ITSE</i>	180.0×10^4	687.0×10^4

4.1.9 Comparação de Resultados Difusos

Após analisar os diversos métodos baseados em lógica difusa, os que se destacaram com as melhores respostas são:

- Controlo PI
 - Sistema 1 com $G_U = 10$
 - Sistema 2 com $G_U = 20$ com *anti-windup*
- Controlo PID com um FLC
 - Sistema 1 e 2 com $K_{PD} = 100$, $K_{PI} = 20$
- Controlo PID com dois FLC
 - Sistema 1 com $K_{PD} = 20$, $K_{PI} = 7.5$ (**A**) e $K_{PD} = 25$, $K_{PI} = 7.5$ (**B**)
 - Sistema 2 com $K_{PD} = 20$, $K_{PI} = 5$
- Controlo PID por *Gain Scheduling* com *anti-windup* (**G.S.**)

No sistema 1, as várias respostas são apresentadas em forma de gráfico na Figura 4.40, onde os critérios de erro respetivos encontram-se na Tabela 4.16.

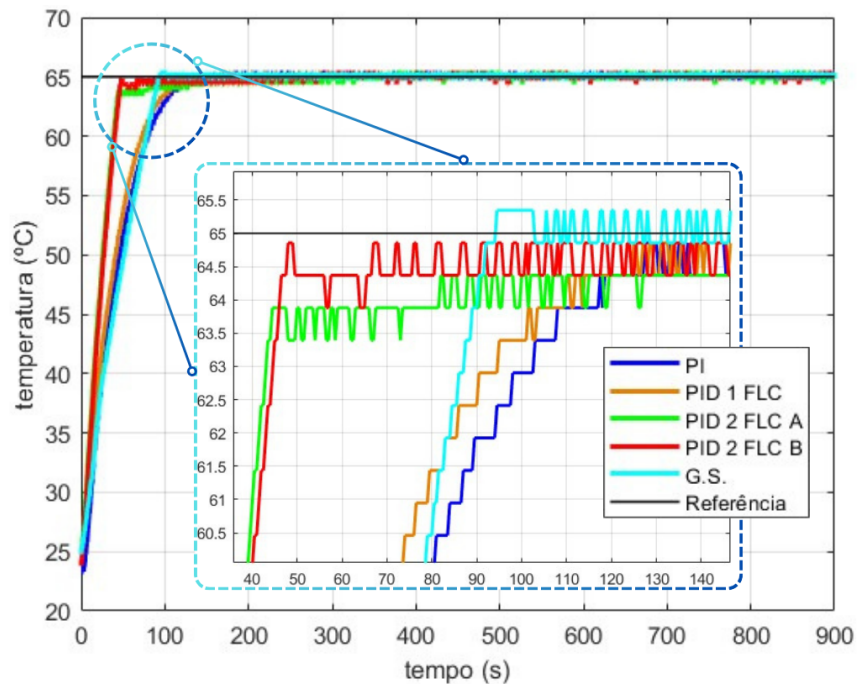


Figura 4.40: Controle Difuso do Sistema 1 - Comparação

Tabela 4.16: Erros de Controle Difuso - Sistema 1

Critério	PI	PID 1 FLC	PID 2 FLC A	PID 2 FLC B	G.S.
<i>IAE</i>	3742	3308	2247	2225	3678
<i>ISE</i>	85.2×10^3	68.4×10^3	45.5×10^3	48.7×10^3	84.6×10^3
<i>ITAE</i>	27.9×10^4	26.3×10^4	21.6×10^4	20.6×10^4	27.1×10^4
<i>ITSE</i>	168.9×10^4	128.9×10^4	53.7×10^4	57.5×10^4	180.0×10^4

Perante os dados observados, para o sistema 1, o melhor método por lógica difusa é o controle PID por dois FLC B, com os parâmetros $K_{PD} = 25$, $K_{PI} = 7.5$. Este possui um tempo de subida de, aproximadamente 50 segundos, estabilizando a sua resposta ao alcançar a temperatura de referência.

No sistema 2, as várias respostas são apresentadas em forma de gráfico na Figura 4.41, onde os critérios de erro respetivos encontram-se na Tabela 4.17.

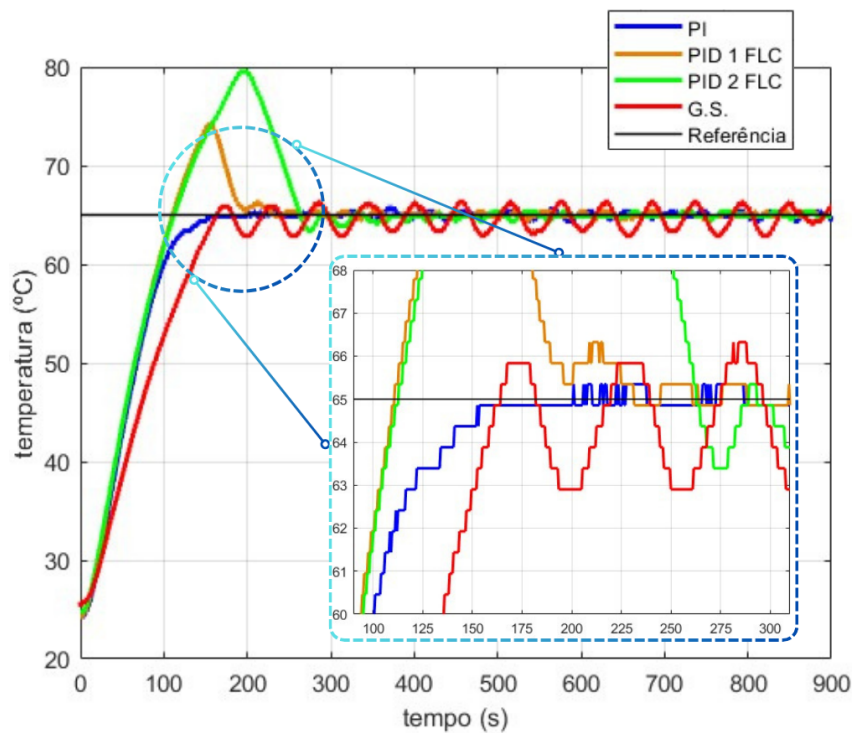


Figura 4.41: Controlo Difuso do Sistema 2 - Comparação

Tabela 4.17: Erros de Controlo Difuso - Sistema 2

Critério	PI AW	PID 1 FLC	PID 2 FLC	G.S.
<i>IAE</i>	5172	5640	7324	7531
<i>ISE</i>	132.6×10^3	131.6×10^3	151.2×10^3	164.4×10^3
<i>ITAE</i>	39.4×10^4	46.2×10^4	82.7×10^4	107.9×10^4
<i>ITSE</i>	362.3×10^4	413.6×10^4	834.8×10^4	687.0×10^4

Perante os dados observados, para o sistema 2, o melhor método por lógica difusa é o controlo PI com o parâmetro $G_U = 20$ e *anti-windup*. Este possui um tempo de subida de, aproximadamente 150 segundos, uma resposta mais lenta que o sistema 1, mas de encontro com as características do sistema. Após alcançar a temperatura de referência é estabilizada a sua temperatura.

4.2 Controlo por PID e por Lógica Difusa

Com a análise efetuada nas secções 3.3 e 4.1, é possível fazer a comparação entre o melhor método, para cada sistema, entre o controlo convencional PID e por aplicações de lógica difusa.

O melhor método de cada tipo de controlo pode ser considerado:

- **Sistema 1**

- Controlo PI por Ziegler-Nichols em malha fechada com *anti-windup*
- Controlo PID por dois FLC B - com $K_{PD} = 25$, $K_{PI} = 7.5$

- **Sistema 2**

- Controlo PI por Ziegler-Nichols em malha fechada com *anti-windup*
- Controlo PI - com $G_U = 20$ e *anti-windup*

No sistema 1, entre o controlo convencional e a lógica difusa, os dois métodos que obtiveram melhores resultados encontram-se representados na Figura 4.42, seguido do valor do erro de cada método na Tabela 4.18.

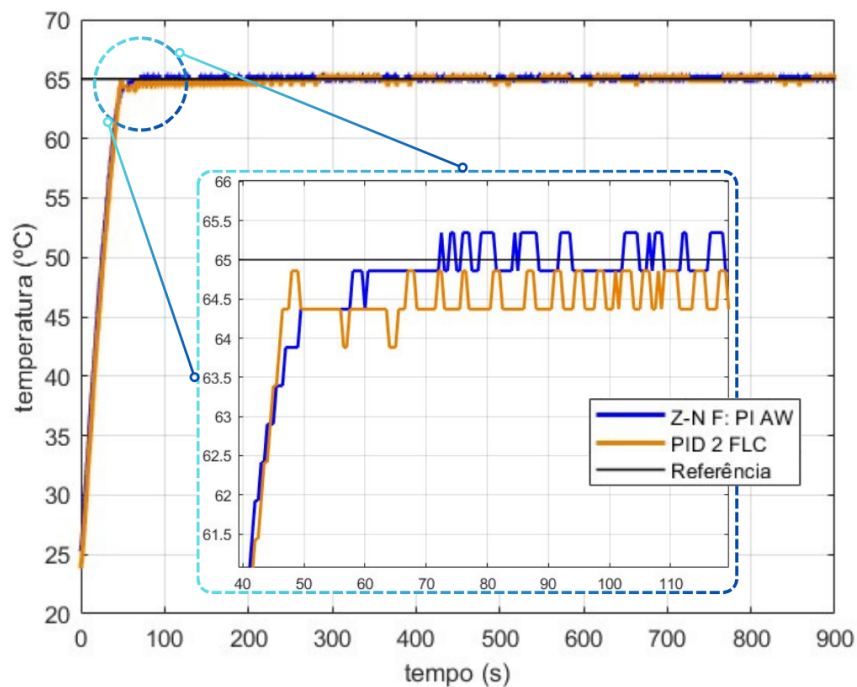


Figura 4.42: Controlo do Sistema 1 - Comparação

Tabela 4.18: Erros do Sistema 1

Critério	Z-N F:PI AW	PID 2 FLC B
<i>IAE</i>	2077	2225
<i>ISE</i>	44.7×10^3	48.7×10^3
<i>ITAE</i>	18.9×10^4	20.6×10^4
<i>ITSE</i>	51.0×10^4	57.5×10^4

Observando o gráfico apresentado, é possível dizer que ambas as respostas são similares, havendo pouca diferença entre elas. No entanto, perante a tabela de erros do sistema 1, pode-se verificar que a resposta do controlo convencional é ligeiramente melhor.

No sistema 2, tal como no sistema 1, os dois métodos que obtiveram melhores resultados encontram-se representados na Figura 4.43, seguido do valor do erro de cada método na Tabela 4.19.

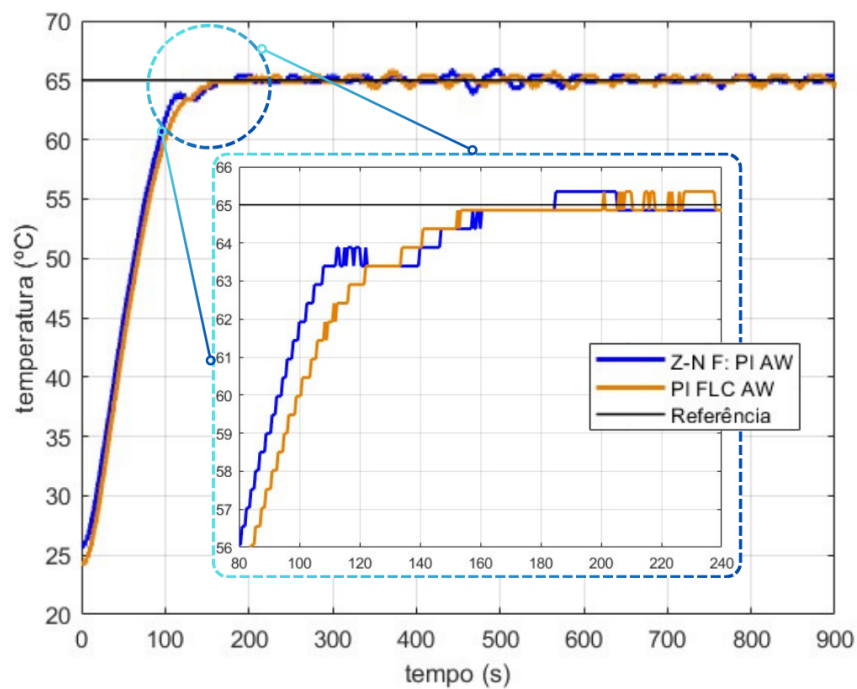


Figura 4.43: Controlo do Sistema 2 - Comparação

Tabela 4.19: Erros do Sistema 2

Critério	Z-N F:PI AW	PI FLC AW
<i>IAE</i>	4767	5172
<i>ISE</i>	116.6×10^3	132.6×10^3
<i>ITAE</i>	36.2×10^4	39.4×10^4
<i>ITSE</i>	305.6×10^4	362.3×10^4

Observando o gráfico apresentado, é possível dizer que o controlo por lógica difusa tem um tempo de subida ligeiramente superior. Ambos apresentam uma resposta similar, sendo os seus valores de erro muito próximos, no entanto é verificado que o controlo convencional, tal como no sistema 1, consegue ser ligeiramente superior ao controlo por lógica difusa no sistema 2.

Capítulo 5

Conclusões

Neste trabalho foi possível analisar o funcionamento real de um sistema de difusão de calor, representado pelo SERT, sendo verificadas algumas irregularidades e limitações. Ao longo do desenvolvimento das experiências efetuadas, é dada especial atenção à temperatura ambiente e aos longos períodos de espera para arrefecimento dos sistemas. Tentando manter o SERT protegido, o quanto possível, de correntes de ar e de mudanças de localização, é impossível evitar pequenas variações nos resultados causadas por estas interferências externas.

Como é descrito na secção 4.1, no caso do sistema 1, o controlo convencional (sendo o controlo PI pelo método de sintonia de Ziegler-Nichols em malha fechada com *anti-windup*, o melhor observado entre os métodos convencionais para este sistema) consegue alcançar um desempenho ligeiramente melhor que o controlo difuso (onde o melhor controlo difuso para este sistema é dado pelo PID por dois FLC com $K_{PD} = 25$, $K_{PI} = 7.5$). De igual forma, no sistema 2, é observado que o controlo difuso (dado pelo controlo PI com $G_U = 20$ e *anti-windup*), obtém um desempenho ligeiramente pior que o controlo convencional (controlo PI pelo método de sintonia de Ziegler-Nichols em malha fechada com *anti-windup*).

Ao longo do extenso estudo, demonstrado pelas várias experiências apresentadas, verificando diversos métodos de controlo e as suas respostas, o resultado entre ambos é similar, não havendo discrepâncias relevantes entre o método convencional e o método por lógica difusa. Algo a referir, é a diferença de complexidade de aplicação entre os dois, do ponto de vista de interface com o utilizador. Um controlo

convencional por PID envolve cálculos e expressões matemáticas que podem não ser compreendidas por todos os utilizadores. Aplicando lógica difusa, apesar de exigir um maior poder de processamento computacional por ser mais complexa, devido à *toolbox* disponibilizada, é facilitada a sua utilização. Uma exceção desta descrição é o método de *gain scheduling*, em que é necessário, além da definição de funções de pertinência e as regras que as interligam, a utilização do controlo PID.

Concluindo assim que, apesar dos resultados entre ambas ser similar, a lógica difusa é considerada uma alternativa mais intuitiva para o controlo de um sistema de difusão de calor.

5.1 Desenvolvimentos Futuros

Para um melhor e mais completo desenvolvimento da análise e comparação apresentada, de um ponto de vista académico, seria importante comparar as respostas obtidas pelo sistema real com respostas adquiridas por um sistema simulado, aplicando a função de transferência dada pelo SERT. Neste sentido, poderia ser verificado o impacto que as interferências externas causam na aplicação de cada método.

O trabalho realizado nesta dissertação apresenta vários métodos e esquemas de controlo difuso, no entanto, de forma a melhorar as respostas apresentadas, poderá ser explorado uma maior otimização das funções de pertinência e regras difusas utilizadas em cada método. Havendo uma demonstração da variação das várias respostas perante a modificação dos FLC, permite haver uma maior noção do impacto que estas têm nos resultados da temperatura do sistema.

Uma abordagem diferente para melhorar a resposta do sistema seria a aplicação de outros métodos, ou esquemas, não explorados nesta dissertação. Algumas destas opções seriam: esquemas com *smith-predictor* [41]; aplicação da metodologia "*Genetic Design of a Three-Rule Fuzzy PI Controller*", onde é aplicada uma otimização, com algoritmos genéticos, dos parâmetros necessários [42]; controladores PID com um FLC de três entradas [34]; controladores PID com três FLC [37]; entre outros.

Referências

- [1] T. Rehman, “An introduction to fuzzy logic controller and its applications,” *International Journal of Scientific Research in Science and Technology*, vol. 7, pp. 540–545, 2017. [Citado nas páginas 5, 15 e 21]
- [2] L. A. Zadeh, “Is there a need for fuzzy logic?,” *Information Sciences*, vol. 178, pp. 2751–2779, 7 2008. [Citado nas páginas vii, 5 e 9]
- [3] S. Ghwanmeh, K. Jones, and D. Williams, “Robustness study of an on-line application of a self-learning fuzzy logic controller,” *IFAC Proceedings Volumes*, vol. 29, pp. 5363–5368, 6 1996. [Citado na página 5]
- [4] B. Kosko and S. Isaka, “Fuzzy logic,” *Scientific American*, pp. 76–81, 7 1993. [Citado nas páginas vii, 5 e 6]
- [5] A. Garrido, “A brief history of fuzzy logic,” 2012. [Citado na página 6]
- [6] J. Yen and S. Member, “Fuzzy logic - a modern perspective,” 1998. [Citado nas páginas 6, 7, 15 e 16]
- [7] L. A. Zadeh, “The role of fuzzy logic in modeling, identification and control,” 1994. [Citado nas páginas vii, 7, 14 e 16]
- [8] L. A. Zadeh, “Fuzzy algorithms,” *Information and Control*, vol. 12, pp. 94–102, 1968. [Citado na página 7]
- [9] L. A. Zadeh, “Fuzzy logic,” 4 1988. [Citado na página 7]
- [10] L. C. D. Junges, “Introdução a lógica fuzzy,” 2006. [Citado nas páginas 7 e 24]
- [11] J. J. Buckley and E. Eslami, *An Introduction to Fuzzy Logic and Fuzzy Sets*. Physica-Verlag, 2002. [Citado na página 7]
- [12] O. Abul-Haggag, “Application of fuzzy logic for the determination of safety integrity in light of international journal of emerging technology and advanced engineering application of fuzzy logic for the determination of safety integrity in light of iec 61508 & 61511 standards,” *Certified Journal*, vol. 9001, 2008. [Citado nas páginas vii e 7]

- [13] MathWorks, *Fuzzy Logic Toolbox™ User's Guide*. 2023. [Citado nas páginas vii, 7, 9, 16, 22 e 23]
- [14] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Pearson College Div, 1997. [Citado nas páginas 9, 17 e 22]
- [15] L. Reznik, “Fuzzy controllers,” 1997. [Citado nas páginas vii, 9, 14, 15, 17, 18, 19 e 21]
- [16] F. Almeida, “Predictive long-term asset maintenance strategy: Development of a fuzzy logic condition-based control system,” 2022. [Citado nas páginas 15, 17, 21 e 22]
- [17] C. Moraga, “Introduction to fuzzy logic,” *ELEC. ENERG*, vol. 18, pp. 319–328, 2005. [Citado na página 15]
- [18] G. DeRose, “Fuzzy logic in real life.” Disponível em www.1-tron.com/fuzzy-logic-in-real-life/ (acedido a: 11-07-2023), 1 2017. [Citado na página 25]
- [19] Tutorialspoint, “Fuzzy logic - applications.” Disponível em www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_applications.htm (acedido a: 11-07-2023), 2017. [Citado na página 25]
- [20] S. E. LeBlanc and D. R. Coughanowr, *Process systems analysis and control*. McGraw-Hill Higher Education, 2009. [Citado na página 31]
- [21] P. Woolf, *CHEMICAL PROCESS DYNAMICS AND CONTROLS*. 5 2023. [Citado nas páginas viii, 31, 32, 33, 35, 37, 38, 39 e 40]
- [22] R. S. Barbosa, “Controladores pid,” 2022. [Citado nas páginas 32, 33, 34, 35 e 72]
- [23] N. Instruments, “The pid controller & theory explained.” Disponível em www.ni.com/pt-pt/shop/labview/pid-theory-explained.html (acedido a: 27-06-2023), 3 2023. [Citado nas páginas 32, 34 e 37]
- [24] R. da Silva, R. Filgueira, E. Pairo-Castineira, I. Overton, and M. Atkinson, “Using simple pid controllers to prevent and mitigate faults in scientific workflows,” 9 2016. [Citado nas páginas viii e 34]
- [25] R. S. Barbosa, “Sintonia controladores pid,” 2022. [Citado nas páginas 36, 37, 39, 40 e 41]
- [26] J. F. Smuts, “Control notes | reflections of a process control practitioner.” Disponível em blog.opticontrols.com/site-map (acedido a: 07-07-2023). [Citado na página 36]
- [27] R. S. Barbosa, “Modelos do processo,” 2022. [Citado nas páginas viii, 39, 40 e 41]

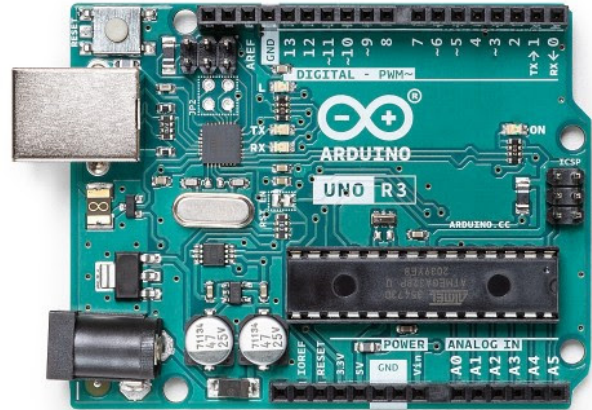
- [28] Arduino, “Arduino ide.” Disponível em www.arduino.cc/en/software (acesso a: 19-08-2023). [Citado nas páginas 43 e 45]
- [29] R. S. Barbosa, “Educational platform for modeling and control,” *XIV Technologies Applied to Electronics Teaching Conference (TAEE)*, 2020. [Citado nas páginas 43, 92 e 99]
- [30] R. S. Barbosa, “Teaching neural control with an arduino based control kit,” vol. 695 LNEE, pp. 415–424, Springer Science and Business Media Deutschland GmbH, 2021. [Citado na página 43]
- [31] M. Technologies, “Atmega328p.” Disponível em www.microchip.com/en-us/product/atmega328p (acesso a: 27-10-2023). [Citado na página 46]
- [32] A. DEVICES, “Tnp36 | voltage output temperature sensors.” Disponível em www.analog.com/en/products/tmp36.html#product-overview (acesso a: 27-10-2023), 2002. [Citado na página 46]
- [33] R. Barbosa, “Sistema eletrônico de regulação térmica para arduino,” [Citado na página 46]
- [34] E. Yesil, M. Güzelkaya, and I. Eksin, “Fuzzy pid controllers: An overview,” 2003. [Citado nas páginas ix, 81, 82, 83, 95, 99, 107 e 120]
- [35] S. Jaszczak and J. Kolodziejczyk, “A method of fast application of the fuzzy pid algorithm using industrial control device,” 2012. [Citado na página 82]
- [36] I. Sekaj, “Fuzzy logic controllers and their replacement by a non-fuzzy approximation algorithm,” *IFAC Proceedings Volumes*, vol. 30, pp. 363–368, 9 1997. [Citado nas páginas 82, 86, 89, 92 e 95]
- [37] G. K. Mann, B. G. Hu, and R. G. Gosine, “Analysis of direct action fuzzy pid controller structures,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, pp. 371–388, 1999. [Citado nas páginas 82, 86, 89, 92, 102, 104 e 120]
- [38] B. G. Hu, G. K. Mann, and R. G. Gosine, “A systematic study of fuzzy pid controllers - function-based evaluation approach,” *IEEE Transactions on Fuzzy Systems*, vol. 9, pp. 699–712, 10 2001. [Citado nas páginas 83, 92 e 95]
- [39] Z. Y. Zhao, M. Tomizuka, and S. Isaka, “Fuzzy gain scheduling of pid controllers,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, pp. 1392–1398, 1993. [Citado nas páginas 107 e 111]
- [40] K. R. Larson, “Fuzzy logic tuning of a proportional-integral-derivative controller,” 2016. [Citado na página 107]

- [41] MathWorks, “Control of processes with long dead time: The smith predictor.” Disponível em www.mathworks.com/help/control/ug/control-of-processes-with-long-dead-time-the-smith-predictor.html (acedido a: 27-10-2023). [Citado na página 120]
- [42] B. G. Hu, G. K. I. Mann, and R. G. Gosine, “Theoretic and genetic design of a three-rule fuzzy pi controller,” vol. 1, pp. 489–496, IEEE, 1997. [Citado na página 120]

Anexo A

Datasheet

A.1 Arduino UNO



Description

The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile development board is equipped with the well-known ATmega328P and the ATmega 16U2 Processor. This board will give you a great first experience within the world of Arduino.

Target areas:

Maker, introduction, industries



Features

- **ATMega328P Processor**
 - **Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - **Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - **Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- **Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- **Power**
 - 2.7-5.5 volts

A.2 ATmega328P

8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash

DATASHEET

Features

- High performance, low power AVR® 8-bit microcontroller
- Advanced RISC architecture
 - 131 powerful instructions – most single clock cycle execution
 - 32 × 8 general purpose working registers
 - Fully static operation
 - Up to 16MIPS throughput at 16MHz
 - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
 - 32K bytes of in-system self-programmable flash program memory
 - 1Kbytes EEPROM
 - 2Kbytes internal SRAM
 - Write/erase cycles: 10,000 flash/100,000 EEPROM
 - Optional boot code section with independent lock bits
 - In-system programming by on-chip boot program
 - True read-while-write operation
 - Programming lock for software security
- Peripheral features
 - Two 8-bit Timer/Counters with separate prescaler and compare mode
 - One 16-bit Timer/Counter with separate prescaler, compare mode, and capture mode
 - Real time counter with separate oscillator
 - Six PWM channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature measurement
 - Programmable serial USART
 - Master/slave SPI serial interface
 - Byte-oriented 2-wire serial interface (Phillips I²C compatible)
 - Programmable watchdog timer with separate on-chip oscillator
 - On-chip analog comparator
 - Interrupt and wake-up on pin change
- Special microcontroller features
 - Power-on reset and programmable brown-out detection
 - Internal calibrated oscillator
 - External and internal interrupt sources
 - Six sleep modes: Idle, ADC noise reduction, power-save, power-down, standby, and extended standby

- I/O and packages
 - 23 programmable I/O lines
 - 32-lead TQFP, and 32-pad QFN/MLF
- Operating voltage:
 - 2.7V to 5.5V for ATmega328P
- Temperature range:
 - Automotive temperature range: -40°C to $+125^{\circ}\text{C}$
- Speed grade:
 - 0 to 8MHz at 2.7 to 5.5V (automotive temperature range: -40°C to $+125^{\circ}\text{C}$)
 - 0 to 16MHz at 4.5 to 5.5V (automotive temperature range: -40°C to $+125^{\circ}\text{C}$)
- Low power consumption
 - Active mode: 1.5mA at 3V - 4MHz
 - Power-down mode: 1 μA at 3V

A.3 TMP36

FEATURES

- Low voltage operation (2.7 V to 5.5 V)
- Calibrated directly in °C
- 10 mV/°C scale factor (20 mV/°C on **TMP37**)
- ±2°C accuracy over temperature (typ)
- ±0.5°C linearity (typ)
- Stable with large capacitive loads
- Specified -40°C to +125°C, operation to +150°C
- Less than 50 µA quiescent current
- Shutdown current 0.5 µA max
- Low self-heating
- Qualified for automotive applications

APPLICATIONS

- Environmental control systems
- Thermal protection
- Industrial process control
- Fire alarms
- Power system monitors
- CPU thermal management

GENERAL DESCRIPTION

The **TMP35/TMP36/TMP37** are low voltage, precision centi-grade temperature sensors. They provide a voltage output that is linearly proportional to the Celsius (centigrade) temperature. The **TMP35/TMP36/TMP37** do not require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the -40°C to +125°C temperature range.

The low output impedance of the **TMP35/TMP36/TMP37** and its linear output and precise calibration simplify interfacing to temperature control circuitry and ADCs. All three devices are intended for single-supply operation from 2.7 V to 5.5 V maximum. The supply current runs well below 50 µA, providing very low self-heating—less than 0.1°C in still air. In addition, a shutdown function is provided to cut the supply current to less than 0.5 µA.

The **TMP35** is functionally compatible with the LM35/LM45 and provides a 250 mV output at 25°C. The **TMP35** reads temperatures from 10°C to 125°C. The **TMP36** is specified from -40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply. The **TMP36** is functionally compatible with the LM50. Both the **TMP35** and **TMP36** have an output scale factor of 10 mV/°C.

The **TMP37** is intended for applications over the range of 5°C to 100°C and provides an output scale factor of 20 mV/°C. The **TMP37** provides a 500 mV output at 25°C. Operation extends to 150°C with reduced accuracy for all devices when operating from a 5 V supply.

The **TMP35/TMP36/TMP37** are available in low cost 3-lead TO-92, 8-lead SOIC_N, and 5-lead SOT-23 surface-mount packages.

FUNCTIONAL BLOCK DIAGRAM

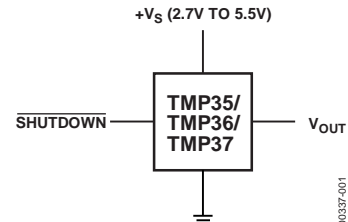


Figure 1.

00337-001

PIN CONFIGURATIONS

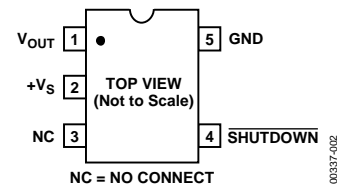


Figure 2. RJ-5 (SOT-23)

00337-002

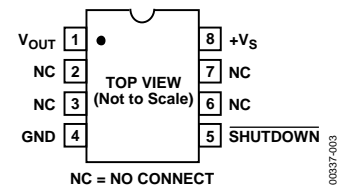


Figure 3. R-8 (SOIC_N)

00337-003

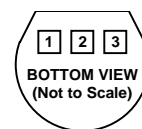


Figure 4. T-3 (TO-92)

00337-004

SPECIFICATIONS

$V_S = 2.7\text{ V to }5.5\text{ V}$, $-40^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$, unless otherwise noted.

Table 1.

Parameter ¹	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
ACCURACY						
TMP35/TMP36/TMP37 (F Grade)		$T_A = 25^\circ\text{C}$		± 1	± 2	$^\circ\text{C}$
TMP35/TMP36/TMP37 (G Grade)		$T_A = 25^\circ\text{C}$		± 1	± 3	$^\circ\text{C}$
TMP35/TMP36/TMP37 (F Grade)		Over rated temperature		± 2	± 3	$^\circ\text{C}$
TMP35/TMP36/TMP37 (G Grade)		Over rated temperature		± 2	± 4	$^\circ\text{C}$
Scale Factor, TMP35		$10^\circ\text{C} \leq T_A \leq 125^\circ\text{C}$		10		$\text{mV}/^\circ\text{C}$
Scale Factor, TMP36		$-40^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$		10		$\text{mV}/^\circ\text{C}$
Scale Factor, TMP37		$5^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$		20		$\text{mV}/^\circ\text{C}$
		$5^\circ\text{C} \leq T_A \leq 100^\circ\text{C}$		20		$\text{mV}/^\circ\text{C}$
Load Regulation		$3.0\text{ V} \leq V_S \leq 5.5\text{ V}$ $0\ \mu\text{A} \leq I_L \leq 50\ \mu\text{A}$ $-40^\circ\text{C} \leq T_A \leq +105^\circ\text{C}$		6	20	$\text{m}^\circ\text{C}/\mu\text{A}$
Power Supply Rejection Ratio	PSRR	$-105^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ $T_A = 25^\circ\text{C}$ $3.0\text{ V} \leq V_S \leq 5.5\text{ V}$		25 30 50	60 100	$\text{m}^\circ\text{C}/\text{V}$ $\text{m}^\circ\text{C}/\text{V}$ $\text{m}^\circ\text{C}/\text{V}$
Linearity				0.5		$^\circ\text{C}$
Long-Term Stability		$T_A = 150^\circ\text{C}$ for 1000 hours		0.4		$^\circ\text{C}$
SHUTDOWN						
Logic High Input Voltage	V_{IH}	$V_S = 2.7\text{ V}$	1.8			V
Logic Low Input Voltage	V_{IL}	$V_S = 5.5\text{ V}$			400	mV
OUTPUT						
TMP35 Output Voltage		$T_A = 25^\circ\text{C}$		250		mV
TMP36 Output Voltage		$T_A = 25^\circ\text{C}$		750		mV
TMP37 Output Voltage		$T_A = 25^\circ\text{C}$		500		mV
Output Voltage Range			100		2000	mV
Output Load Current	I_L		0		50	μA
Short-Circuit Current	I_{SC}	Note 2			250	μA
Capacitive Load Driving	C_L	No oscillations ²	1000	10000		pF
Device Turn-On Time		Output within $\pm 1^\circ\text{C}$, $100\text{ k}\Omega 100\text{ pF}$ load ²		0.5	1	ms
POWER SUPPLY						
Supply Range	V_S		2.7		5.5	V
Supply Current	I_{SY} (ON)	Unloaded			50	μA
Supply Current (Shutdown)	I_{SY} (OFF)	Unloaded		0.01	0.5	μA

¹ Does not consider errors caused by self-heating.

² Guaranteed but not tested.

Anexo B

Esquema SERT

