

Controlo de um Motor Diesel com Turbo-Compressor Utilizando Redes Neurais Artificiais

GONÇALO FILIPE DUARTE PINTO

Julho de 2016

CONTROLO DE UM MOTOR DIESEL COM TURBO-COMPRESSOR UTILIZANDO REDES NEURONAIS ARTIFICIAIS

Gonçalo Filipe Duarte Pinto



Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

2016

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese / Dissertação (TEDI), do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Gonçalo Filipe Duarte Pinto, Nº 1110782, 1110782@isep.ipp.pt

Orientação científica: Ramiro De Sousa Barbosa, rsb@isep.ipp.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

14 de Julho de 2016

Agradecimentos

Um agradecimento em especial à minha mãe. Quero também agradecer ao engenheiro Ramiro Barbosa pela ajuda e empenho ao longo deste projeto.

Resumo

Neste documento descreve-se o projeto desenvolvido na unidade curricular de Tese e Dissertação durante o 2^o ano do Mestrado de Engenharia Eletrotécnica e de Computadores no ramo de Automação e Sistemas, no Departamento de Engenharia Eletrotécnica (DEE) do Instituto Superior de Engenharia do Porto (ISEP).

Com a evolução dos sistemas automóveis, toda a gestão do funcionamento do motor passou a estar a cargo de um sistema eletrónico, originando a gestão eletrónica do motor. O seu objetivo é fazer com que o nível de emissões de gases poluentes esteja dentro dos padrões exigidos, sem comprometer o desempenho e conforto da condução.

O projeto escolhido teve como base o uso da tecnologia das redes neuronais artificiais (RNA) para controlo de emissões.

Relativamente à implementação, foram consideradas diferentes estruturas de controlo. A análise exposta neste trabalho trata de perceber em que medida é que a introdução das redes neuronais melhora o controlo de um processo.

Os testes de desempenho são aplicados no controlo de um motor diesel com VGT e EGR, sendo realizados através do software MATLAB. As simulações efetuadas têm diferentes configurações de modo a tirar conclusões o mais gerais possível.

Palavras-chave

RNA, PID, controlo, motor diesel, VGT, EGR, emissões.

Abstract

This dissertation reports the project developed in the Thesis/Dissertation course during the 2nd year of the Master of Electrical and Computer Engineering in the field of Automation and Systems, Department of Electrical Engineering (DEE) of the Instituto Superior de Engenharia do Porto (ISEP).

With the evolution of automotive systems, an electronic system became responsible for all the engine operations, yielding the electronic engine management system. Its goal is to comply with the required standards for the level of greenhouse gas emissions, without compromising the car performance and driving comfort.

The chosen project was based on the use of the technology of artificial neural networks for the control of emissions.

In what concerns the implementation, it was considered different control structures. The proposed idea in this work is the introduction of artificial neural networks (ANN) to improve the process control.

The performance tests are conducted through the use of MATLAB software and in the control of a diesel engine with VGT and EGR. The simulations have different settings in order to get the most general conclusions.

Keywords

ANN, PID, control, diesel engine, VGT, EGR, emissions.

Conteúdo

1	Introdução	1
1.1	Enquadramento e Motivação	2
1.2	Objetivos	3
1.3	Calendarização	3
1.4	Estrutura do Relatório	4
2	Redes Neurais	5
2.1	Desenvolvimento Histórico	6
2.1.1	William James (1890)	6
2.1.2	Neurónio de McCulloch e Pitts (1943)	7
2.1.3	<i>Hebbian Learning</i> (1949)	8
2.1.4	Neuro-computação (1946-1957)	10
2.1.5	Perceptrão (1958)	10
2.1.6	ADALINE	11
2.1.7	Limitações (1960)	11
2.1.8	Próxima Geração (anos 80)	12
2.2	Redes Neurais Biológicas	12
2.2.1	Comunicação entre Neurónios	13
2.2.2	Redes de Neurónios	14
2.2.3	Interação entre Arquiteturas Neurais	14
2.2.4	Aprendizagem das Redes Neurais Biológicas	15
2.3	Redes Neurais Artificiais	15
2.3.1	Neurónio Artificial	16

2.3.2	Ligações Artificiais	17
2.3.3	Funções de ativação dos neurónios artificiais	17
2.4	Análise das RNAs	22
2.4.1	Associação Entre os Dados de Entrada e Saída	22
2.4.2	Arquiteturas	22
2.4.3	Aprendizagem	23
2.4.4	Processos de Aprendizagem	26
2.4.5	Tipo de Procedimento de Ajuste das Conexões Sinápticas	35
2.5	Aplicações das RNAs	36
2.5.1	Previsão de Séries Temporais	36
2.5.2	Processamento de Voz	37
2.5.3	Caracterização Visual de Imagens Recuperadas (CVIR)	38
2.5.4	Processamento Biomédico de Imagens	38
3	Controlo e Casos de Estudo	39
3.1	Controlador com Estabilizador Fixo	39
3.2	Controlo Inverso Adaptativo	40
3.3	Modelo de Controlo de Referência ou Controlo Adaptativo Neuronal	41
3.4	Controlo Crítico Adaptativo	41
3.5	Casos de Estudo	42
3.5.1	Aplicação de RNAs na Previsão Específica de Combustível Consumido e Temperatura dos Gases de Escape para um Motor a Diesel	43
3.5.2	Previsão do Desempenho de um Motor que Utiliza Biocombustíveis	45
4	Software de Desenvolvimento	51
4.1	Emergent	51
4.2	Brain Simulator	52

4.3	MATLAB - <i>Neural Network Toolbox</i>	54
4.3.1	Interfaces NNTOOL e NNSTART	55
4.3.2	Comandos MATLAB	63
5	Emissões, VGT e EGR	71
5.1	Emissões	71
5.1.1	Relação Ar-Combustível	73
5.1.2	Legislação Europeia	74
5.2	Descrição do VGT e da EGR	75
5.2.1	O VGT	76
5.3	EGR	77
6	Análise ao Modelo de Johan Wahlström e Lars Eriksson	79
6.1	Sumário da Estrutura do Modelo	80
6.2	Modelação do Sistema	82
6.2.1	Coletores	82
6.2.2	Cilindros	84
6.2.3	Válvula-EGR	86
6.2.4	Turbo-Compressor	88
6.2.5	<i>Intercooler</i> e <i>EGR Cooler</i>	92
6.3	Exemplos de Controlo	93
7	Desenvolvimento	95
7.1	Processo a Controlar	96
7.1.1	Vantagens dos Parâmetros Escolhidos	97
7.2	Controlo PID	97
7.2.1	Estrutura dos Controladores PID	99
7.2.2	Controladores Primários	100
7.2.3	Controladores Adicionais	101
7.2.4	Método Automático de Sintonia	102
7.3	Controlador com Estabilizador Fixo	104

7.4	Índices de Desempenho	110
8	Testes e Resultados	113
8.1	Controlo PID	113
8.2	Configuração da RNA	117
8.3	Controlador com Estabilizador Fixo	118
8.3.1	Resultados Globais	119
8.3.2	Análise Gráfica	120
8.3.3	Índices de Desempenho IAE e ISE	126
8.3.4	Sumário dos Resultados	129
9	Conclusões	131
9.1	Conclusões Gerais	131
9.2	Sugestões para Trabalho Futuro	133
	Referências Documentais	141
	Anexo A. Esquemas do Controlo PID	143
	Anexo B. Esquemas do Controlo com Estabilizador Fixo	147

Lista de Figuras

Figura 2.1	Exemplo de um neurónio de McCulloch e Pitts que efetua cálculos lógicos [3].	9
Figura 2.2	Neurónio Biológico [5].	13
Figura 2.3	Neurónio Artificial.	16
Figura 2.4	Função de ativação linear.	19
Figura 2.5	Função de ativação em degrau.	19
Figura 2.6	Função de ativação em rampa saturada.	20
Figura 2.7	Função de ativação hiperbólica.	21
Figura 2.8	Função de ativação gaussiana.	21
Figura 2.9	Processo de aprendizagem supervisionado [2].	25
Figura 2.10	Processo de aprendizagem por reforço [2].	25
Figura 2.11	Processo de aprendizagem não-supervisionado [2].	26
Figura 2.12	Processo de aprendizagem com base na correção do erro.	28
Figura 2.13	Esquema genérico de uma rede neuronal multi-camada.	33
Figura 3.1	Controlador estabilizador [6].	40
Figura 3.2	Sistema de controlo inverso adaptativo [6].	41
Figura 3.3	Sistema com modelo de controlo de referência e controlo adaptativo neuronal [6].	42
Figura 3.4	Modelo de controlo crítico adaptativo [6].	42
Figura 3.5	Diagrama de blocos do motor [35].	43
Figura 3.6	Fotografia do motor de testes [35].	44

Figura 3.7	Previsão de consumos de combustível efetuado pela RNA na fase de testes [35].	45
Figura 3.8	Previsão de temperatura de escape efetuado pela RNA na fase de testes [35].	46
Figura 3.9	Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para a potência [36].	48
Figura 3.10	Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o binário [36].	48
Figura 3.11	Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o consumo de combustível [36].	49
Figura 3.12	Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o consumo de combustível específico [36].	49
Figura 4.1	Imagem do <i>layout</i> do <i>software</i> Emergent, com alguns exemplos de amostra [37].	52
Figura 4.2	Imagem do <i>layout</i> do <i>software</i> Brain Simulator [38].	53
Figura 4.3	Imagem referente ao <i>layout</i> da <i>nntool</i>	55
Figura 4.4	Aba de configuração da rede.	56
Figura 4.5	Aba de edição de dados da rede.	57
Figura 4.6	Imagem referente ao <i>layout</i> do <i>nntstart</i>	58
Figura 4.7	Página de introdução com esquema da RNA.	58
Figura 4.8	Página para introdução dos dados da experiência.	59
Figura 4.9	Página para definir os parâmetros de validação.	59
Figura 4.10	Página para definir a arquitetura da RNA.	60
Figura 4.11	Página para treino da RNA.	61
Figura 4.12	Página para análise do treino da RNA.	61

Figura 4.13	Página para avaliação da RNA.	62
Figura 4.14	Página onde é possível gerar a rede para implementação.	62
Figura 4.15	Imagem referente à página, para guardar resultados.	63
Figura 4.16	Gráfico com os resultados do treino.	69
Figura 5.1	Gráfico da composição do ar atmosférico.	72
Figura 5.2	Vista orientada de um VGT [39].	76
Figura 5.3	Válvulas do VGT.	77
Figura 5.4	Esquema do circuito de fluxo de ar e dos gases de escape [39].	78
Figura 6.1	Modelo das estruturas do motor Diesel [41].	80
Figura 6.2	Modelo geral do sistema para aplicação em MATLAB.	82
Figura 6.3	Comparação entre a dinâmica do atuador simulado e os dados obtidos nos diferentes testes [41].	89
Figura 7.1	Modelo em cascata da estrutura de controlo.	96
Figura 7.2	Esquema de controlo PID em realimentação.	98
Figura 7.3	Diagrama de blocos da estrutura de controlo do processo.	100
Figura 7.4	Diagrama de treino da RNA para obtenção do modelo inverso do processo.	106
Figura 7.5	Gráfico referente à forma de onda de treino da velocidade de rotação do motor.	107
Figura 7.6	Diagrama do modelo da RNA do controlo com estabilizador fixo.	108
Figura 7.7	Diagrama de treino da RNA para o controlo com estabilizador fixo (modelação inversa).	109
Figura 7.8	Diagrama de testes da RNA para o controlo com estabilizador fixo (modelação inversa).	110
Figura 7.9	Diagrama de testes da RNA para o controlo com estabilizador fixo (modelação inversa).	111

Figura 8.1	Resposta de saída do sinal λ_0 , para o controlo otimizado (PID) e não otimizado (PIDinic).	115
Figura 8.2	Resposta de saída do sinal x_{egr} , para o controlo otimizado (PIDo) e não otimizado (PIDinic).	116
Figura 8.3	Resposta de saída do sinal n_t , para o controlo otimizado (PIDo) e não otimizado (PIDinic).	116
Figura 8.4	Resposta de saída do sinal λ_0 , para a velocidade de 1200 rpm.	121
Figura 8.5	Resposta de saída do sinal λ_0 , para a velocidade de 1400 rpm.	121
Figura 8.6	Resposta de saída do sinal λ_0 , para a velocidade de 1500 rpm.	122
Figura 8.7	Resposta de saída do sinal λ_0 , para a velocidade 1700 rpm.	123
Figura 8.8	Resposta de saída do sinal x_{egr} , para a velocidade de 1200 rpm.	124
Figura 8.9	Resposta de saída do sinal x_{egr} , para a velocidade de 1400 rpm.	124
Figura 8.10	Resposta de saída do sinal x_{egr} , para a velocidade de 1500 rpm.	125
Figura 8.11	Resposta de saída do sinal x_{egr} , para a velocidade de 1700 rpm.	125
Figura 8.12	Resposta de saída do sinal n_t , para a velocidade de 1200 rpm.	126
Figura 8.13	Resposta de saída do sinal n_t , para a velocidade de 1400 rpm.	127
Figura 8.14	Resposta de saída do sinal n_t , para a velocidade de 1500 rpm.	127

Figura 8.15 Resposta de saída do sinal n_t , para a velocidade de
1700 rpm. 128

Lista de Tabelas

Tabela 1.1	Calendarização do projeto.	3
Tabela 3.1	Constituição dos combustíveis para testes e respetivas percentagens em termos de volume [36].	47
Tabela 6.1	Dados das medições dos três controladores analisados durante o ciclo de ETC.	93
Tabela 7.1	Valores dos ganhos obtidos em [53], através da aplicação do método de otimização.	104
Tabela 8.1	Resultados obtidos para o controlador PID sem ganhos ajustados (PIDinic) e com ganhos ajustados (PIDo).	114
Tabela 8.2	Ganhos obtidos para o controlador não otimizado (PIDinic) e otimizado (PIDo).	114
Tabela 8.3	Tabela com os resultados IAE e ISE de λ_0 , para o controlo PIDinic e PIDo.	117
Tabela 8.4	Tabela com os resultados IAE e ISE de x_{egr} , para o controlo PIDinic e PIDo.	117
Tabela 8.5	Resultados obtidos para o desempenho da RNA, com diferentes números de neurónios.	118
Tabela 8.6	Tabela referente aos resultados obtidos para o controlo estabilizador (NNPID) e o controlo PID que foi otimizado, tendo por medida $V(\theta)$	119

Tabela 8.7 Tabela com os resultados IAE e ISE de λ_0 , para diferentes velocidades de funcionamento do motor. 128

Tabela 8.8 Tabela com os resultados IAE e ISE de x_{egr} , para diferentes velocidades de funcionamento do motor. 128

Lista de Acrónimos

ANN *Artificial Neural Networks*

ARX Modelo Auto Regressivo com Entrada Externa (*Auto Regressive model with eXternal input*)

BSR Velocidade de Rotação das Pás (*Blade Speed Ratio*)

CVIR Caracterização Visual de Imagens Recuperadas

EGR Recirculação de Gases de Escape (*Exhaust Gas Recirculation*)

ETC *European Transient Cycle*

IAE Integral do Erro Absoluto (*Integral Absolute Error*)

IBM *International Business Machines*

ISEP Instituto Superior de Engenharia do Porto

ISE Integral do Erro Quadrático (*Integral Square Error*)

MEEC Mestrado em Engenharia Eletrotécnica e de Computadores

MIMO Múltiplas Entradas e Múltiplas Saídas (*Multiple-Input and Multiple-Output*)

MLP Perceptrão de múltiplas camadas (*Multi-Layer Perceptron*)

RBF Funções de Base Radial (*Radial Basis Function*)

RNA Rede Neuronal Artificial

SISO *Single Input Single Output*

SNARC *Stochastic Neural-Analog Reinforcement Computer*

TEDI Tese/Dissertação

UE União Europeia

VGT Turbo de Geometria Variável (*Variable Geometry Turbocharger*)

Capítulo 1

Introdução

O constante crescimento da frota de automóveis a nível mundial e o consequente aumento da poluição atmosférica devido à emissão de gases poluentes, levou a que fosse necessário tomar medidas no sentido de regulamentar esses níveis de emissões. As medidas restritivas têm vindo a ser cada mais rigorosas ao longo do tempo, exigindo níveis de emissões cada vez mais baixos e consumos de combustível progressivamente menores. Para cumprir com essas exigências e considerando os avanços tecnológicos que os sistemas eletrónicos sofreram, os construtores de automóveis passaram a adotar os sistemas de controlo eletrónico nos seus motores, designados de gestão eletrónica do motor.

Todos os automóveis produzidos atualmente são desenvolvidos com gestão eletrónica. Esta é toda ela controlada pela unidade que é o cérebro de todas as operações: a centralina ou unidade de controlo do motor, que recebe todas as informações necessárias para gerir o seu funcionamento. A unidade de controlo do motor ao receber uma grande quantidade de fluxo de informação ao mesmo tempo e com elevada capacidade de processar a informação, permite que o motor reaja de forma quase instantânea às solicitações do condutor, aliando as baixas emissões a um bom desempenho do motor. A unidade de controlo do motor recebe as informações das condições em

que o motor se encontra através do estado de uma vasta gama de sensores, comparando-os com uma série de valores pré-programados (mapas) alojados na sua memória, enviando depois sinais elétricos aos atuadores com as ações que devem efetuar.

A otimização, quando efetuada de forma correta, proporciona inúmeras vantagens, como uma aceleração mais rápida e aumento de binário. De tal forma proporciona-se uma resposta mais pronta e eficaz, reduzindo o uso da caixa de velocidades e aumentando o prazer da condução e até mesmo diminuição do consumo. Quando a fábrica produz um automóvel tem que satisfazer uma vasta gama de clientes e também tem um conjunto de restrições e fatores, devido a estratégias de mercado e vários compromissos, pelo que os automóveis não vêm explorados no seu potencial máximo.

Aliando o controlo presente nos veículos, a tecnologias de controlo inovadoras como é o caso das redes neuronais artificiais (RNAs), podem trazer inúmeras vantagens para o desenvolvimento de veículos. Investindo-se principalmente numa realidade, em que se pode associar cada vez mais baixos níveis de emissões poluentes a melhores desempenhos.

1.1 Enquadramento e Motivação

A realização do projeto, apresentado ao longo deste relatório, insere-se no âmbito da unidade curricular Tese/Dissertação (TEDI) do 2º ano do Mestrado em Engenharia Eletrotécnica e de Computadores (MEEC), no ramo de Automação e Sistemas, do Instituto Superior de Engenharia do Porto (ISEP). A motivação para este projeto deve-se à curiosidade de entender em que consiste e como se processa a gestão eletrónica de um motor Diesel com VGT e EGR, de forma a efetuar um controlo do sistema onde são utilizadas RNAs.

1.2 Objetivos

O desenvolvimento deste trabalho de dissertação, tem por objetivos a aplicação de RNAs em sistemas de controlo de motores diesel. A RNA deverá ser capaz de controlar um modelo de um motor Diesel generalista de 6 cilindros, para camiões de grande porte.

O estudo desenvolvido, tem por foco o controlo da razão estequiometria de oxigénio e combustível, bem como da razão de gases de escape recirculados. A implementação deste controlo, deve possibilitar a regulação das emissões de NOx e de gases de escape. Com a implementação das RNAs, deve ser perceptível a melhoria ou não do sistema diesel. Como forma de comparação propõe-se a implementação de um sistema com um controlador Proporcional Integrador e Derivativo (PID), com provas dadas, para o modelo do motor Diesel a utilizar.

1.3 Calendarização

A Tabela 1.1 contém a calendarização estipulada para o desenvolvimento do projeto, de forma a organizar o desenvolvimento deste.

Tabela 1.1: Calendarização do projeto.

Tarefas	Semanas	Dezembro				Janeiro				Fevereiro				Março				Abril				Maio				Junho			
		1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a	1 ^a	2 ^a	3 ^a	4 ^a
Definição do Modelo de Controlo		■																											
Estudo das Redes Neurais						■																							
Estudo e Desenvolvimento do Controlo PID										■																			
Desenvolvimento do Controlo Estabilizador																		■											
Testes e Resultados																						■							
Relatório						■																							

1.4 Estrutura do Relatório

No Capítulo 1 é apresentada uma breve contextualização ao trabalho proposto, assim como aos objetivos do trabalho, a sua calendarização e a organização.

No Capítulo 2 é apresentado o estado de arte das redes neuronais: a sua história, biologia associada, as estruturas das redes artificiais, os conceitos associados e as aplicações.

No Capítulo 3 é feita uma análise de alguns métodos de controlo que utilizam RNAs e dois casos de estudo associados.

No Capítulo 4 é feita uma análise de *software* de desenvolvimento para a aplicação e desenvolvimento de RNAs.

No Capítulo 5 analisa-se algumas generalidades das emissões associadas a veículos, bem como dos sistemas mecânicos de VGT e EGR.

No Capítulo 6 efetua-se uma análise ao modelo de um motor Diesel com VGT e EGR, desenvolvido por Johan Wahlström e Lars Eriksson.

No Capítulo 7 são descritos os sistemas de controlo utilizados. Onde são abordados todos os passos para a aplicação destes.

No Capítulo 8 são expostos os testes realizados e os resultados obtidos. Além disso é feita uma análise aos controlos aplicados, e respetivas comparações.

No Capítulo 9 são apresentadas as conclusões e sugestões para trabalho futuro.

Capítulo 2

Redes Neurais

O computador, é presentemente uma das ferramentas mais vulgarmente utilizadas para o desenvolvimento tecnológico e do próprio ser humano. Nos últimos anos, tem-se assistido a um desenvolvimento acelerado de tecnologias computacionais com a capacidade de executar procedimentos, com uma velocidade tal, que dificilmente se imaginaria há algumas décadas atrás. Já são várias as áreas, em que diversos programas são utilizados como ferramenta de apoio, abrangendo a medicina, economia, engenharia, etc.

O carácter específico dos métodos de programação, implica o aparecimento de uma grande diversidade de programas dedicados. Para contornar este problema, vários investigadores têm-se debruçado sobre técnicas de programação mais genéricas, para serem utilizadas num maior leque de tarefas. As RNAs parecem ser uma das técnicas mais adequadas a esse fim. O aumento de popularidade deste tipo de estruturas, tanto nos ambientes de desenvolvimento universitários como em ambientes comerciais, constata esse facto. Nos dias de hoje, é comum referenciar-se as RNAs a uma vasta área de aplicações, desde a ciência dos computadores, à física, à matemática, à economia, à engenharia e em muitos outros campos, onde novas aplicações estão continuamente a ser encontradas e investigadas.

As RNAs podem ser entendidas como dispositivos de processamento de

informação, caracterizados pela interconexão de unidades elementares de processamento (denominados neurónios artificiais), simples e similares entre si [1]. Consiste, na iniciativa de modelação matemática de algumas tarefas do sistema cognitivo, sendo atribuído ao padrão de conexões e aos valores dos pesos sinápticos o papel de moldar o comportamento de entrada-saída da rede, ou seja, a forma como a rede neuronal irá responder a determinados estímulos de entrada. Alguns dos incentivos que motivam a utilização das RNAs são descritos pelos seguintes pontos:

- A habilidade de aprender a partir de dados observados de um processo que se pretende modelar ou emular;
- Capacidade de aproximação universal de mapeamentos estáticos não-lineares e multi-dimensionais;
- Capacidade de aproximação de mapeamentos dinâmicos não-lineares e multi-dimensionais;
- Habilidade de implementação de memória associativa e para a solução de problemas combinatórios pela procura de pontos de equilíbrio;
- Capacidade de auto-organização e de treino não-supervisionado.

2.1 Desenvolvimento Histórico

A linha histórica apresentada ao longo desta secção, tem por fim, mostrar o desenvolvimento das redes neuronais ao longo dos anos, bem como os principais intervenientes na sua formulação e desenvolvimento. Toda a análise é baseada na referência [3].

2.1.1 William James (1890)

As fundações das redes neuronais precedem-se de há quase um século. Os primeiros passos, no desenvolvimento das redes neuronais artificiais foram

dados pelo psicólogo William James (1890). O trabalho de James, teve uma grande importância, pois foi o primeiro a discutir as funções de memória do cérebro como sendo algo compreensível, previsível e talvez uma estrutura fundamentalmente simples. No decorrer da aprendizagem de James, este não desenvolveu qualquer modelo matemático, que descrevesse as funções do cérebro. O contributo dado por este, foram modelos e princípios gerais, que carregam uma semelhança impressionante ao trabalho posterior de Donald Hebb (1949) e outros.

As questões fundamentais para o avanço do conhecimento sobre as RNAs, que foram efetuadas por psicólogos no final do século XIX e início do século XX, são baseadas nas seguintes questões: Como é que pensando em A, o cérebro, pensa imediatamente em B? Porque é que um determinado som ou cheiro ou visão invoca uma determinada memória ou pensamento? A resposta reside na memória associativa. James escreve que "(...) não há outra lei elementar causal de associação que não seja a lei do hábito neuronal".

A análise filosófica feita por James, levou a analisar os problemas neuronais do quotidiano (mais especificamente a associação de acontecimentos) e relacioná-los em termos de conexões neuronais (atualmente denominado de treino). O trabalho efetuado por James, possibilitou décadas depois o desenvolvimento da teoria de McCulloch e Pitts (1943), que é a seguir relatada.

2.1.2 Neurónio de McCulloch e Pitts (1943)

Chegado o ano de 1943, McCulloch e Pitts apresentaram um dos artigos mais importantes para o avanço das RNAs, *The logical calculus of ideas immanent in nervous activity*. Este documento, descreve pela primeira vez, como o comportamento do cérebro humano pode ser caracterizado computacionalmente, através de funções matemáticas.

O neurónio proposto por McCulloch e Pitts tem por conceitos cinco suposições:

- O neurónio é dispositivo binário. Os valores à entrada do neurónio só podem ser 0 ou 1.
- Cada neurónio tem um limite fixo. O limite, é o valor numérico da soma das entradas, que o neurónio deve exceder antes que este possa calcular uma saída. Esse limite é geralmente 1.
- O neurónio consegue receber entradas de pesos de conexões excitadas ($w = +1$). Também pode receber entradas com peso de conexão inibidos ($w = -1$), o que evita a ativação do neurónio.
- Há um *quantum*¹ de tempo para integrar as entradas sinápticas. Durante esse *quantum* de tempo, o neurónio responde à atividade cerebral consoante os seus pesos.
- Se nenhum parâmetro inibitório está ativo ($w = -1$), o neurónio soma as entradas e verifica se a soma excede o limite. Se isso se verificar o neurónio fica ativo.

A Figura 2.1, mostra um exemplo do neurónio de McCulloch e Pitts. Este apresenta uma unidade simples, com duas entradas que podem ser excitadas, A e B e com um limite de 1. Em $t = 0$, se A e B estão ambos inativos então em $t = 1$ a unidade continua inativa. Se em $t = 0$, A está ativo e B desativo então a unidade fica ativa. Esta unidade está a efetuar a operação **OU**. A unidade fica ativa apenas se A ou B ou ambos A e B forem ativos.

2.1.3 Hebbian Learning (1949)

A grande contribuição de Donald O. Hebb é conhecida como o *Hebbian Learning*. Em 1949 no seu livro *The Organization of Behavior*, este descreve a inter-relação de neurónios que acontece durante o processo de aprendizagem. "Se o axónio de um neurónio de entrada está perto de excitar um neurónio

¹Menor valor que determinada grandeza física pode apresentar.

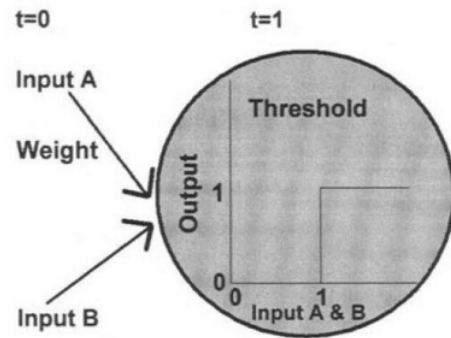


Figura 2.1: Exemplo de um neurônio de McCulloch e Pitts que efetua cálculos lógicos [3].

alvo, um processo de crescimento ocorre numa das células ou em ambas as células para aumentar a eficiência do neurônio de entrada". Hebb nunca definiu esta propriedade matematicamente, mas esta é aplicada nas RNAs como estrutura básica, dos pesos sinápticos, para definir a relação entre os elementos de processamento na rede.

A teoria de Hebb foi testada em 1956 por Rochester na empresa International Business Machines (IBM). O artigo apresentado marcou um grande avanço na investigação das RNAs, as teorias propostas podiam ser agora testadas em computador.

O aparecimento e o desenvolvimento do computador digital, influenciou diretamente o desenvolvimento das RNAs. Neumann surgiu o trabalho de McCulloch e Pitts e na publicação que fez reside a ideia de um programa armazenado em memória (1958- *The Computer and the Brain*). Neumann afirmou, que uma rede neuronal biológica não consegue ter uma precisão acima de três *bits*. Concluiu então, que era necessário ter cuidado com as analogias entre computadores e cérebros, visto que *chips* e neurónios não são a mesma coisa.

2.1.4 Neuro-computação (1946-1957)

Na década entre 1946 e 1957, surgiu o nascimento dos neuro-computadores e uma divisão entre redes neuronais e inteligência artificial. Marvin Minsky, construiu o primeiro neuro-computador com nome de *Stochastic Neuronal-Analog Reinforcement Computer* (SNARC), em 1951. O SNARC, é constituído por 40 "neurónios" conectados por ligações ajustáveis. A aprendizagem era feita ajustando a tensão e a polaridade nas ligações.

No Verão de 1956, John McCarthy e Minsky fundaram o primeiro laboratório de inteligência artificial no Massachusetts Institute of Technology (MIT). Nesta altura começaram a aparecer divergências nas teorias dos investigadores. Estas divisões criaram diferentes tipos de processos para o avanço do conhecimento na área das RNAs. Um grupo de investigadores, preferiu focar-se em encontrar um melhoramento na área digital. O outro grupo de investigadores, preferiu avançar de forma a compreender melhor o processo natural, para se possível chegar a uma melhor aproximação.

Após a separação dos investigadores, surge um novo avanço para o futuro das RNAs em 1958, em que um psicólogo Frank Rosenblatt e um engenheiro Charles Wightman desenvolveram o *Mark I Perceptron neurocomputer*.

2.1.5 Perceptrão (1958)

O investigador Frank Rosenblatt, utilizou um sistema visual para formular o vocabulário a utilizar no seu Perceptrão, sendo que este estava mais interessado em problemas de perceção. O sistema originário do Perceptrão, foi composto por três camadas: uma camada de entrada, definidas como *retinal units*; uma camada intermédia denominada de *association units*; e por fim uma camada de saída denominada de *response units*. Cada uma das camadas era interligada de forma aleatória, as ligações eram ajustadas consoante o treino efetuado à rede, que tinha por base um mecanismo de reforço.

Os padrões identificados pelo Perceptrão eram constituídos por vetores

binários. A rede desenvolvida foi limitada a duas camadas de processamento, com uma camada de pesos ajustáveis entre elas. Poderiam ser adicionadas mais camadas à RNA, mas estas não se adaptavam.

2.1.6 ADALINE

Bernard Widrow, professor de engenharia eletrotécnica e o seu aluno Ted Hoff (criador do microprocessador), avançaram com uma teoria diferente relativamente ao desenvolvimento das RNAs. Desta forma surgiu a criação de uma máquina parecida à de Rosenblatt, a ADALINE ou como foi chamada mais tarde a MADALINE. A ADALINE foi utilizada para redes adaptativas lineares, enquanto que a MADALINE são várias ADALINE. A ADALINE é familiar no que toca à sua aplicação nos dias de hoje, aos filtros para cancelar ecos durante conversas telefónicas.

A ADALINE, permitia entradas e saídas na rede, de valores positivos e negativos, de +1 ou -1, ao contrário do Perceptrão que só permitia entradas entre 1 e 0. O peso das conexões, seguia o mesmo princípio do Perceptrão.

Com a utilização da ADALINE, tornou-se possível resolver diversos problemas desde a tradução de linguagens, à previsão do tempo e a resolução de um problema mais concreto, esta permitia determinar a melhor rota para posicionar um camião, numa doca de carregamento.

Apesar do sucesso, tanto da ADALINE como do Perceptrão, ambos apresentaram o mesmo problema, não eram capazes de resolver problemas linearmente inseparáveis.

2.1.7 Limitações (1960)

Warren McCulloch em 1960, introduziu na área dois jovens investigadores Marvin Minsky e Seymour Papert, os quais já tinham artigos publicados na área das redes neuronais. Estes dois investigadores mostraram que a necessidade do processamento localizado e a utilização de unidades de limites

lineares, presentes no Perceptrão e na ADALINE, significavam que nenhum destes neuro-computadores era capaz de resolver problemas conectados (é o caso de desenhar uma figura sem levantar o lápis do papel).

2.1.8 Próxima Geração (anos 80)

A análise de Minsky e Papert levou à *destruição* das teorias entre 1970 e 1980, na área da pesquisa das redes neuronais.

O reaparecimento das pesquisas na área deveu-se ao aparecimento de uma publicação sobre algoritmos de redes não lineares, que ultrapassou muitas das limitações do Perceptrão e da ADALINE. O novo algoritmo foi chamado de *back-propagation* (em português - retro-propagação). Na realidade não foi algo novo, alguns investigadores já tinham chegado a algoritmos idênticos, Paul Werbos (1974) e David Parker (1985).

A partir deste surgimento deu-se novamente o *boom* das redes neuronais. Desde essa altura, é possível constatar as possibilidades que o desenvolvimento de um algoritmo traz, para a aplicação das RNAs em diversas áreas.

2.2 Redes Neurais Biológicas

O elemento básico do sistema nervoso é o neurónio. Os componentes principais do neurónio incluem um corpo central, as dendrites e o axónio. Num extremo do neurónio existe uma multiplicidade de filamentos curtos, denominados dendrites, agrupando-se à medida que se aproximam do corpo central ligando-se a este. No outro extremo nasce um filamento longo denominado axónio, cuja extremidade se ramifica unindo as diferentes ramificações de outros neurónios.

A Figura 2.2 representa esquematicamente o diagrama conceptual básico de um neurónio biológico. Existem vários tipos de neurónios com funções diferentes, por exemplo, alguns neurónios funcionam como motores trans-

mitindo informação, outros funcionam como sensores, captando informação. Para obter informação mais detalhada sobre as diferentes configurações e funções dos neurónios recorrer à referência [4].

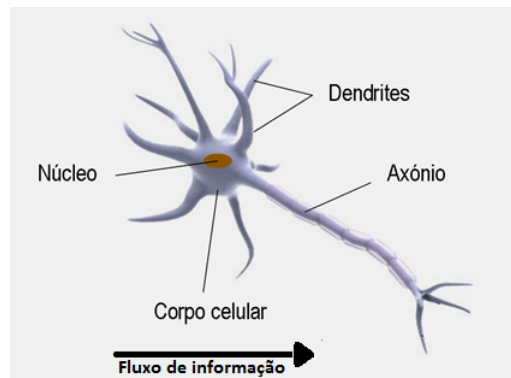


Figura 2.2: Neurónio Biológico [5].

2.2.1 Comunicação entre Neurónios

As dendrites e os axónios possuem propriedades eletrofisiológicas especiais que atribuem aos neurónios o papel de processadores de informação. A transmissão de informação entre neurónios é feita através de sinais elétricos. Os sinais elétricos fluem da esquerda para a direita, a partir das dendrites, através do corpo central, até ao axónio. No corpo central existe uma transformação/ativação do sinal mediante a informação recebida. O axónio distribui o sinal recebido do corpo central pelas dendrites de outros neurónios. O sinal proveniente de um neurónio é transmitido para outro neurónio através das ligações existentes entre o axónio do primeiro neurónio e uma dendrite do segundo. Estas ligações denominam-se por sinapses. As sinapses são as entidades responsáveis pela capacidade de aprendizagem de um sistema neuronal.

2.2.2 Redes de Neurónios

A interação, entre conjuntos de neurónios em grandes redes confere ao cérebro a capacidade para aprender, reconhecer e generalizar. Sabe-se que a atuação isolada de um neurónio, só por si, é muito limitada em termos de processamento de informação.

O cérebro humano é constituído por milhões de neurónios. Este é o maior e mais complexo sistema neuronal biológico sendo o que apresenta as capacidades mais imprevisíveis. Contudo, mesmo o diminuto cérebro de uma mosca, com um número muito menor de neurónios é capaz de a orientar na realização de uma diversidade de tarefas. Num sistema, com poucos neurónios a interligação e combinação destes, forma diferentes arquiteturas neuronais, permitindo que o sistema possa executar funções distintas.

2.2.3 Interação entre Arquiteturas Neuronais

O cérebro possui capacidades de processamento complexas devido à existência de arquiteturas neuronais sofisticadas. A informação chega ao cérebro através de entradas sensoriais consistindo em várias camadas de neurónios a partir das quais a informação é processada. Algumas destas camadas de neurónios podem realizar a filtragem preliminar da informação recolhida, enquanto que outras são responsáveis pela seleção, associação e distribuição de informação pelas diversas arquiteturas neuronais existentes. As diferentes arquiteturas neuronais disponíveis processam a informação recebida conforme as funções que desempenham. Depois da informação processada, esta é transmitida para outras camadas de neurónios, que são responsáveis pela execução de várias tarefas, como por exemplo, decisões, formas de procedimento e supervisão.

2.2.4 Aprendizagem das Redes Neurais Biológicas

A capacidade para aprender é uma das características mais importantes das redes neurais biológicas. Os humanos e os animais aprendem continuamente na sua interação com o meio envolvente. A informação momentânea disponível é automaticamente assimilada pelo cérebro afetando o comportamento futuro.

Muitos investigadores têm-se debruçado sobre o sistema de aprendizagem neuronal biológico de forma a descobrirem exatamente como e onde a informação momentânea disponível é armazenada no cérebro. Os resultados mais relevantes são descritos a seguir:

- A atividade sináptica facilita as comunicações neuronais. Um elevado grau de atividade entre dois neurónios ou grupos de neurónios num determinado instante pode facilitar a comunicação entre eles no momento e no futuro;
- A aprendizagem estimula o crescimento neuronal. Alguns dos investigadores, dedicados ao estudo das redes neurais, têm observado que a existência de uma aprendizagem contínua, causa um aumento do número de ligações entre neurónios, em determinadas regiões do cérebro. Apesar de existir uma grande quantidade de neurónios aquando da formação do cérebro, o número de neurónios aumenta significativamente com o decorrer do tempo.

2.3 Redes Neurais Artificiais

As redes neurais artificiais, são algoritmos computacionais baseados nos sistemas biológicos. Estes algoritmos, possuem capacidades computacionais que são quase unicamente atribuídas à inteligência humana.

Os elementos fundamentais dos modelos de redes neurais artificiais são o neurónio artificial, a arquitetura de rede (que descreve a forma como os

diversos neurónios estão dispostos e interligados) e a aprendizagem (ajustes dos pesos sinápticos). Nas subsecções seguintes, vão ser descritos os elementos básicos das estruturas neuronais artificiais e o seu modo de operação.

2.3.1 Neurónio Artificial

A estrutura básica de um neurónio artificial é a base constituinte de todas as estruturas neuronais artificiais. Com a análise da Figura 2.3 é possível fazer uma analogia entre o neurónio biológico e o artificial.

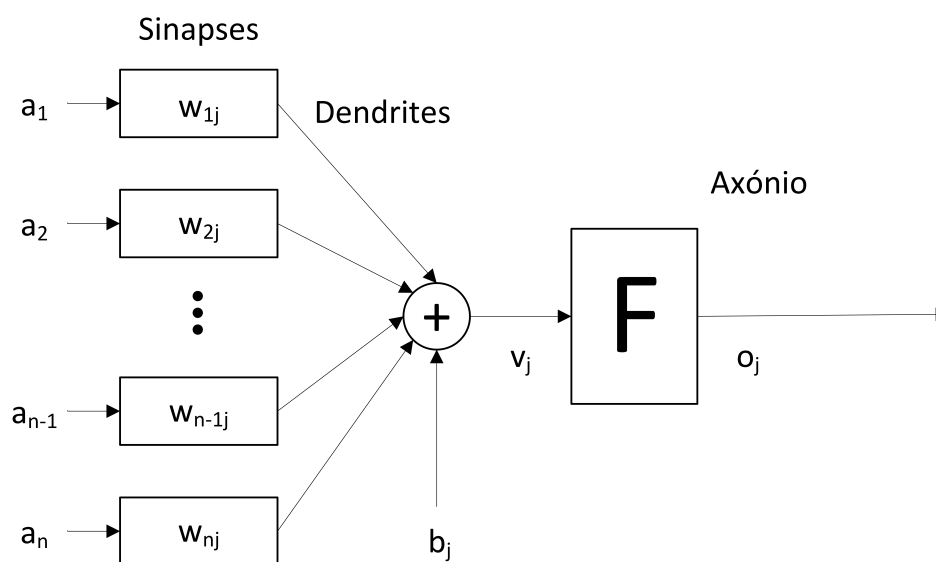


Figura 2.3: Neurónio Artificial.

Como se verifica nos neurónios biológicos, um neurónio artificial é um elemento simples de processamento que pode receber várias entradas de outros neurónios através das dendrites, produzindo um único sinal de saída. Este sinal pode ser propagado pelo axónio deste neurónio para outros neurónios.

Recorrendo aos dados presentes na Figura 2.3, é possível desenvolver para o neurónio uma expressão matemática simples, que permite obter os valores de saída em função das entradas. Passando a descrever, o valor da saída do neurónio artificial é igual ao valor transformado do somatório de

todas as entradas desse neurónio. De tal forma, as entradas de um neurónio são obtidas a partir da multiplicação das saídas pelos pesos das ligações de outros neurónios, para o qual existam ligações. São múltiplas as funções de ativação utilizadas nos neurónios artificiais. Sendo que algumas são mais usuais do que outras (estas são descritas a seguir, na subsecção 2.3.3).

2.3.2 Ligações Artificiais

Os comprimentos de ligação, ou também denominados de pesos, existentes entre os vários neurónios são parâmetros ajustáveis, os quais conferem às estruturas neuronais a capacidade de aprendizagem e de colocação em memória de informação. A uma determinada ligação entre dois neurónios está associado um peso, o qual é multiplicado pela saída do neurónio anterior, formando uma das entradas para o neurónio posterior. É o valor deste peso que determina o grau de interação entre os dois neurónios artificiais.

A entrada total de cada neurónio j , v_j é o somatório das saídas dos neurónios aos quais está interligado, a_j , multiplicando pelo respetivo peso, w_{ij} , mais uma entrada constante, b_j , definida por *bias*. A equação (2.1), representa o número de neurónios interligados com o neurónio j .

$$v_j = \sum_{i=1}^n w_{ij} a_i + b_j \quad (2.1)$$

A entrada constante, b_j , funciona como uma espécie de ajuste de zero para todas as entradas do neurónio. Diz-se que uma ligação funciona como excitação ou inibição para um neurónio se os pesos apresentarem valores positivos ou negativos, respetivamente.

2.3.3 Funções de ativação dos neurónios artificiais

Existem diversas funções de ativação associadas aos neurónios artificiais. As diferentes funções de ativação dos neurónios produzem diferentes saídas.

Os neurónios artificiais usam as funções de ativação para calcular o nível de transformação desse neurónio perante as suas entradas. As diferentes funções utilizadas, são na maior parte dos casos um fator de distinção das estruturas neuronais artificiais.

Em regra geral, é desejável que a função de ativação apresente um comportamento limitado por comparação com as respostas apresentadas pelos neurónios biológicos. As funções devem ser limitadas superior e inferiormente para que os valores calculados à saída dos neurónios não apresentem comportamentos instáveis.

As funções de ativação mais vulgarmente utilizadas na criação de modelos neuronais artificiais são:

- A **função de ativação linear** (Figura 2.4) é descrita por:

$$f(x) = m.x \quad (2.2)$$

O coeficiente m é um escalar positivo, que define o declive da respetiva reta, visto que a derivada da função de ativação é constante, $f'(x) = m$. Se for $m = 1$ todo o estado passa completamente pelos elementos de processamento para a saída, o que leva a que $f(x) = x$.

- A **função de ativação em degrau** (Figura 2.5), na sua conceção geral, pode originar apenas dois valores de saída β e $-\alpha$. Se o estado x exceder o limiar θ , a função de ativação origina o valor β na saída. Caso contrário a saída será $-\alpha$. Este funcionamento é descrito pela função (α , β e θ são escalares positivos):

$$f(x) = \begin{cases} \beta & , \text{ se } x \geq \theta \\ -\alpha & , \text{ se } x < \theta \end{cases} \quad (2.3)$$

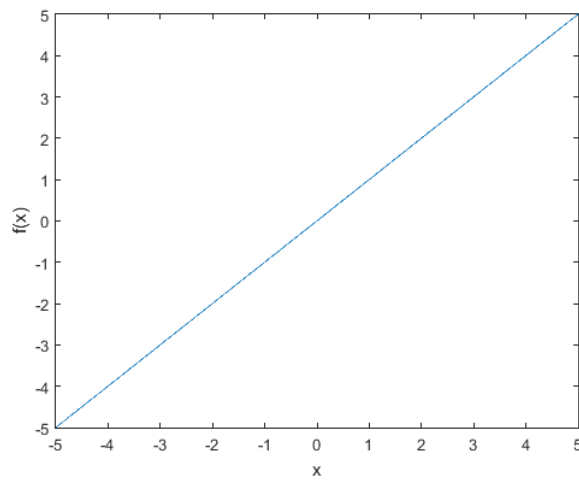


Figura 2.4: Função de ativação linear.

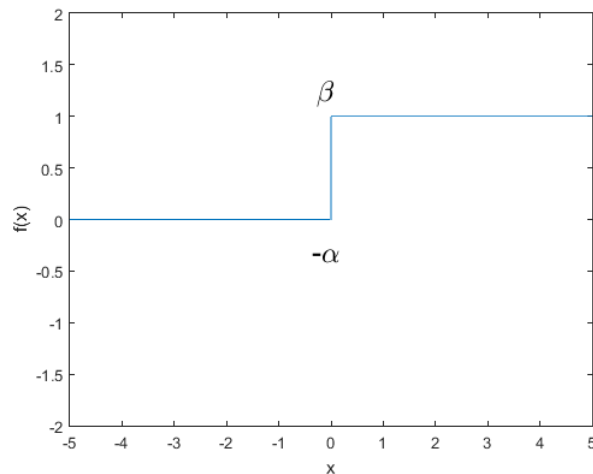


Figura 2.5: Função de ativação em degrau.

A partir desta equação pode-se chegar à função de ativação binária (função de Heaviside), se $\theta = 0$ e $\alpha = 0$ simultaneamente.

- A **função de ativação em rampa saturada** (Figura 2.6) equivale à combinação das funções lineares e degrau. Exibe uma saída com variação linear entre os limites superior γ e inferior $-\gamma$ impostos por uma saturação admitida simétrica e ideal, de onde se obtém:

$$f(x) = \begin{cases} \gamma & , \text{ se } x \geq \theta \\ a.x & , \text{ se } |x| < \theta \\ -\gamma & , \text{ se } x \leq -\theta \end{cases} \quad (2.4)$$

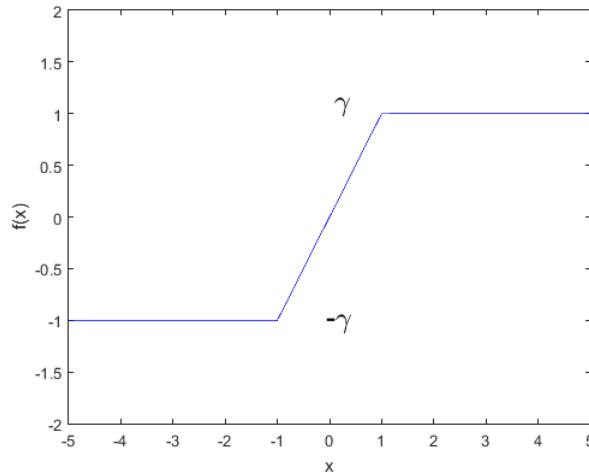


Figura 2.6: Função de ativação em rampa saturada.

- A **função tangente hiperbólica** (Figura 2.7) expressa por $f(x) = \tanh(x/2)$ é dada por:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.5)$$

variando entre -1 e 1 , desde $x = -\infty$ a $x = \infty$.

- A **função de ativação gaussiana** (Figura 2.8), simétrica em torno da origem, define a sua forma pela variância $v > 0$ de acordo com a expressão:

$$f(x) = e^{-\frac{x^2}{2}} \quad (2.6)$$

sendo

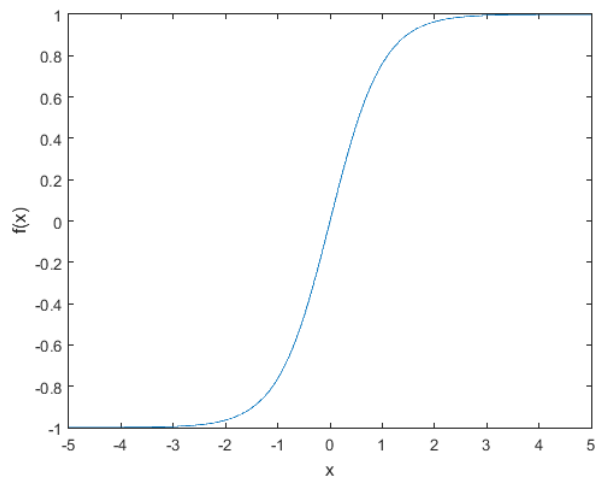


Figura 2.7: Função de ativação hiperbólica.

$$x = \frac{u - m}{\sigma} \tag{2.7}$$

com média m e o desvio-padrão σ para $v = \sigma^2$.

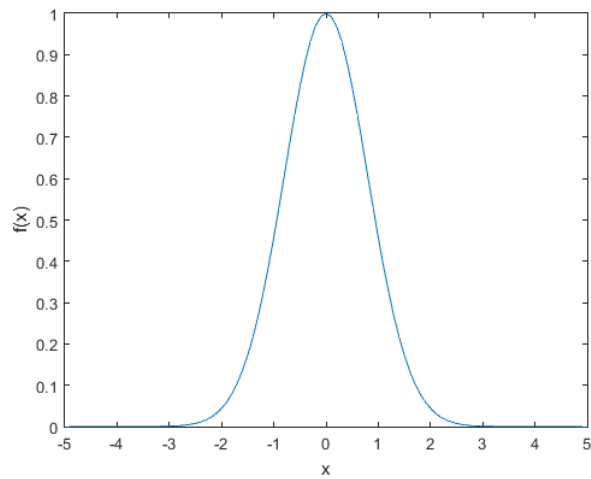


Figura 2.8: Função de ativação gaussiana.

2.4 Análise das RNAs

As redes neuronais artificiais podem ser classificadas conforme os critérios precedentes: tipo de associação relativamente às informações de entrada e saída, tipos de arquiteturas, aprendizagem, processos de aprendizagem e tipo de procedimento de ajuste dos pesos sinápticos.

2.4.1 Associação Entre os Dados de Entrada e Saída

A associação da informação entre a entrada e a saída são divididos em dois tipos, Auto-associativas e Hetero-associativas.

Auto-associativas A rede armazena determinados padrões de entrada, recebidos no processo de treino, por ajuste das sinapses. Quando aparece algum tipo de informação incompleta ou com ruído, ela realizará uma associação e responderá com o padrão mais parecido dentre os já armazenados. Exemplos deste tipo de redes: Rede de Hopfield (Hopfield, 1982) e a família de arquiteturas ART (Carpenter e Grossberg, 1988).

Hetero-associativas A rede armazena certas associações de entrada-saída recebidas no processo de treino, por ajuste das sinapses. Assim, quando se lhe apresenta um certo estímulo de entrada, esta deverá responder gerando a correspondente saída. Exemplos: rede MLP (do inglês *Multi-Layer Perceptron*) (Rumelhart e McClelland, 1986) e rede RBF (do inglês *Radial Basis Function*) (Broomhead e Lowe, 1988).

2.4.2 Arquiteturas

O tipo de arquitetura de uma rede neuronal pode ser classificado pelo número de camadas ou pelo tipo de conexões.

Pelo número de camadas Redes de uma camada, onde cada um dos neurónios recebe a entrada e produz a saída final. Este tipo de rede geralmente está associado a tarefas auto-associativas, por exemplo, reconstruindo padrões incompletos ou com ruído.

Redes de várias camadas, onde os neurónios estão dispostos em vários níveis ou camadas, que obedecem a certos padrões de conexão. Podem ser destacadas a camada de entrada, que recebe os padrões de entrada; uma ou várias camadas ocultas, que geralmente realizam o mapeamento de classes ou regressão de dados, e a camada de saída. Nos casos auto-associativos realiza um processamento associativo, mas nos casos hetero-associativos compõe a saída do neurónio, combinando a informação proveniente da última camada oculta.

Pelo tipo de conexões Redes *Feedforward*, onde os sentidos das conexões são à frente. Por exemplo: MLP, RBF.

Feedforward/feedback, onde existem conexões de realimentação, além das diretas. Por exemplo: redes recorrentes (Connor e Martin, 1994). Essas redes podem ser totalmente ou parcialmente recorrentes (Dos Santos e Von Zuben, 2000).

2.4.3 Aprendizagem

A aprendizagem é o processo no qual a RNA modifica os pesos das ligações (conexões sinápticas) e/ou outros parâmetros estruturais em função de:

- informação de entrada;
- informação da entrada associada a uma saída desejada.

Análogo ao processo biológico, onde os estímulos recebidos promovem modificações nas intensidades das sinapses, a maior parte dos mecanismos de aprendizagem em RNAs procuram ajustar as conexões ou pesos sinápticos em resposta aos estímulos recebidos.

O fundamental no processo de aprendizagem é definir a adaptação dos pesos quando se requer que a rede aprenda uma nova informação. Tal processo pode ser classificado como supervisionado ou não-supervisionado, e vai depender do problema a resolver. Existem também processos semi-supervisionados.

Supervisionado Este método de aprendizagem (Figura 2.9), tem por base a ideia de um supervisor que determina a resposta que a rede deverá ter para uma determinada entrada. Este mecanismo está fortemente vinculado às redes do tipo hetero-associativas. O supervisor verifica a saída da rede e, caso ela não coincida com a saída desejada, faz um ajuste nos pesos das conexões visando minimizar esta diferença. Por exemplo:

- Por correção do erro: onde o ajuste deve se efetuado com o intuito de minimizar o erro cometido, ou seja, a diferença entre a saída da rede e a saída desejada. Um dos algoritmos mais conhecidos relacionado a este método é o *back-propagation* (Werbos, 1974; Rumelhart e McClelland, 1986) para redes *feedforward*, e extensões deste algoritmo para redes *feedforward/feedback* ou redes recorrentes (Pineda, 1987; Almeida, 1987).
- Por reforço (Figura 2.10): opção com um grau menor de supervisão que a anterior, no qual não se dispõe de um exemplo completo do comportamento desejado ou saída desejada. Esta forma de treino é parecida a um crítico, que, em vez de minimizar o erro, fornece apenas um indicativo de nível de sucesso ou insucesso vinculado a uma sequência de ações da rede neuronal. Alguns exemplos deste tipo de algoritmo: *Linear Reward-Penalty* (Narendra e Thathacher, 1974) e *Associative RewardPenalty* (Bartp e Anandan, 1985).

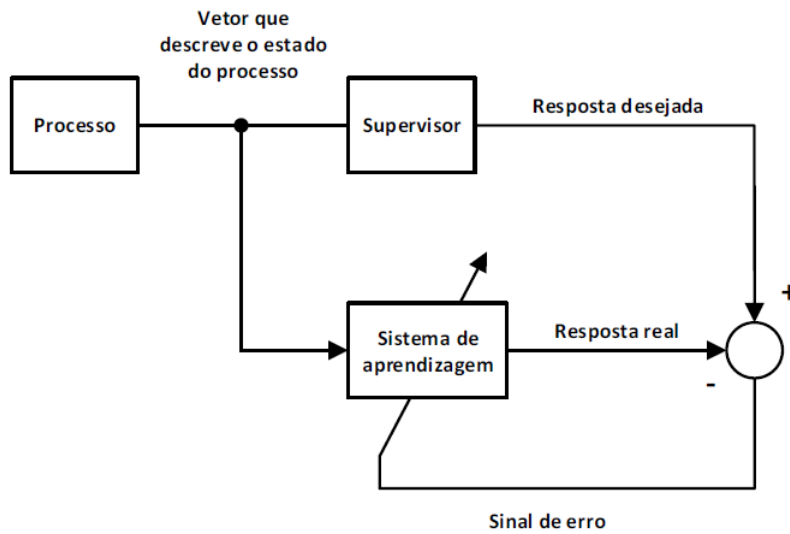


Figura 2.9: Processo de aprendizagem supervisionado [2].

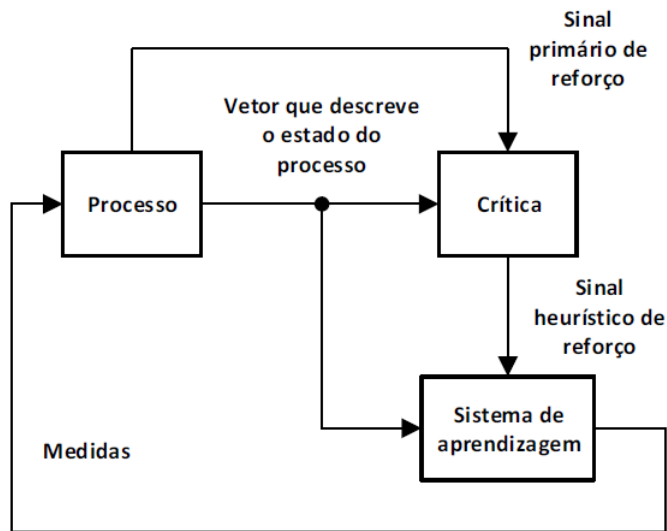


Figura 2.10: Processo de aprendizagem por reforço [2].

Sem-supervisão Para este tipo de aprendizagem (Figura 2.11), não existe supervisão nem indicativo do nível de sucesso ou insucesso. É conhecido também como aprendizagem auto-supervisionado, não requer indicativos de comportamento desejado para a rede neuronal. Com isso, interpreta-se o

processo de ajuste de conexões como resultado de um processo de auto-organização. Aplica-se em redes auto-associativas e pode ser de dois tipos:

- **Regra de Hebb** (Hebb, 1949): conexões associadas a neurónios que se encontram ativos simultaneamente tendem a ser fortalecidas, enquanto que conexões associadas a neurónios que sofrem ativações em instantes sem relação no tempo, tendem a ser enfraquecidas.
- **Competitivo e cooperativo**: neste conjunto de algoritmos, os neurónios concorrem para representar as amostras à entrada e o que vencer a competição ajusta os valores das suas conexões, de forma a incrementar, na direção da amostra à entrada. Os neurónios vizinhos ao neurónio vencedor, também são ajustados, embora sejam ajustes de menor intensidade.

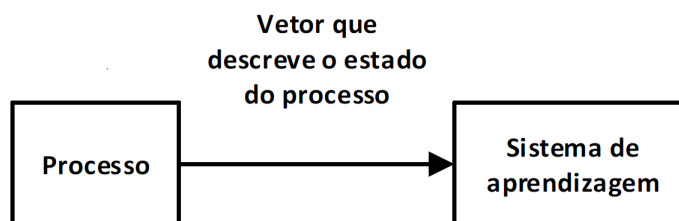


Figura 2.11: Processo de aprendizagem não-supervisionado [2].

2.4.4 Processos de Aprendizagem

Entre as várias propriedades interessantes de uma RNA é a capacidade de a rede aprender através do ambiente em que foi colocada e melhorar a sua capacidade de aprendizagem.

A definição do processo de aprendizagem é ditado pela seguinte sequência de acontecimentos:

1. A RNA é simulada num determinado ambiente;

2. A RNA sofre alterações devido aos resultados da simulação efetuada;
3. A RNA responde de uma nova forma ao ambiente em que está aplicada, isto porque aconteceram alterações na sua estrutura interna.

Sendo $w_{kj}(n)$ o valor do peso sináptico w_{kj} no instante n . No instante n um ajuste é aplicado $\Delta w_{kj}(n)$ ao peso sináptico $w_{kj}(n)$, o que gera um valor dado por:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (2.8)$$

Ao conjunto de regras prescritas para a solução de um problema de aprendizagem, é designado de *algoritmo de aprendizagem*. Como é expectável, não existe apenas um algoritmo de aprendizagem para o projeto de uma RNA. Em vez disso, existe um conjunto de ferramentas representadas por uma diversidade de algoritmos, em que cada uma tem a sua vantagem. Basicamente, os algoritmos de aprendizagem distinguem-se entre si, na forma em que o ajuste Δw_{kj} é aplicado aos pesos sinápticos w_{kj} .

2.4.4.1 Aprendizagem Tendo por Base a Correção do Erro/Regra Delta

Tendo por base da análise a Figura 2.12. O parâmetro $d_k(n)$ corresponde à resposta desejada ou o objetivo a alcançar pelo neurónio k no instante n . Em que o valor correspondente à resposta atual (saída) do neurónio é dada por $y_k(n)$. A resposta $y_k(n)$ é produzida por um estímulo, aplicado à entrada da rede na qual o neurónio k está inserido. O vetor de entrada e a resposta desejada $d_k(n)$, constituem uma amostra particular apresentada à rede no instante n .

Geralmente, a resposta atual ($y_k(n)$) do neurónio k é diferente da resposta desejada ($d_k(n)$). Assim, pode-se definir um sinal de erro dado por:

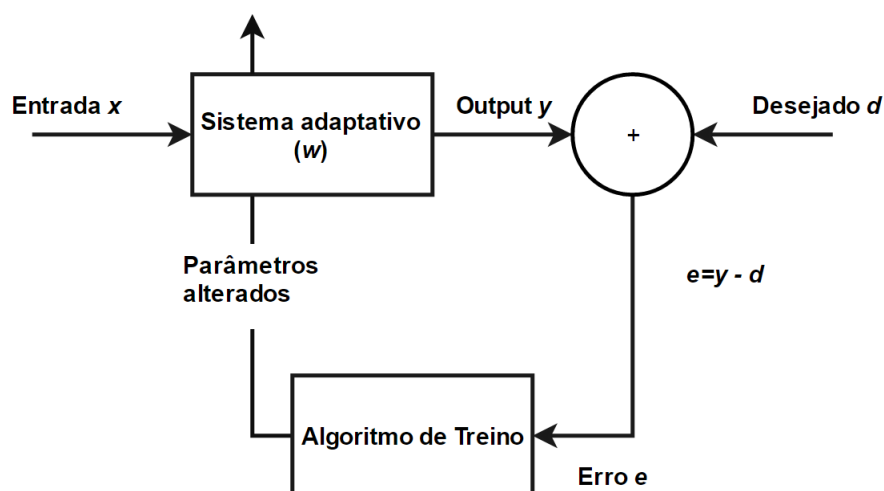


Figura 2.12: Processo de aprendizagem com base na correção do erro.

$$e_k(n) = y_k(n) - d_k(n) \quad (2.9)$$

O objetivo da aprendizagem com base no erro é a função de *custo* baseado no erro do sinal ($e_k(n)$).

O critério mais utilizado para a função de *custo* é o valor instantâneo do erro quadrático:

$$J(n) = \frac{1}{2} \sum_k e_k^2(n) \quad (2.10)$$

Por fim a RNA é então otimizada minimizando o parâmetro $J(n)$ com base nos pesos sinápticos da rede. De acordo com a regra de aprendizagem, *correção do erro ou regra delta*, os pesos sinápticos são ajustados seguindo a fórmula seguinte:

$$\Delta w_{kj}(n) = \eta e_k x_j(n) \quad (2.11)$$

2.4.4.2 Aprendizagem de *Hebb*

A formulação de regras de aprendizagem feita pelo pensador Hebb, é a mais antiga e mais famosa:

1. Se dois neurónios, posicionados em ambos os lados de uma sinapse forem ativos ao mesmo tempo, então a força dessa ligação é incrementada seletivamente.
2. Se dois neurónios, posicionados em ambos os lados de uma sinapse forem ativos assincronamente, então essa ligação é enfraquecida ou eliminada seletivamente.

Para formular o pensamento de Hebb matematicamente, considera-se o peso sináptico w_{kj} com atividade pré-sináptica e pós-sináptica, dadas por x_j e y_k , respetivamente. De acordo com a teoria de Hebb, o ajuste aplicado ao peso sináptico w_{kj} no instante n é:

$$\Delta w_{kj}(n) = F[y_k(n), x_j(n)] \quad (2.12)$$

Como caso especial pode ser utilizada a atividade dada pela regra do *produto*:

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \quad (2.13)$$

onde η é uma constante positiva que determina a *taxa de aprendizagem*. Esta regra enfatiza a relação com a natureza de uma sinapse "Hebbiana".

Para tal representação é possível constatar que a repetição da aplicação do sinal de entrada x_j gera um crescimento exponencial, conduzindo os pesos sinápticos w_{kj} a uma saturação, como descrevem as equações seguintes:

$$w_{kj}(n+1) = w_{kj}(n) + \eta y_k(n) x_j(n) = w_{kj}(n)(1 + \eta x_j^2(n)) \quad (2.14)$$

$$w_{kj}(n+2) = w_{kj}(n+1) + \eta y_k(n+1)x_{j+1}(n) = w_{kj}(n)(1 + \eta x_j^2(n))(1 + \eta x_j^2(n+1)) \quad (2.15)$$

Se x_j for constante então:

$$w_{kj}(n + N) = w_{kj}(n)[1 + \eta x_j^2]^N \quad (2.16)$$

Para evitar a expansão deste género de situações, é necessário impor um limite de crescimento, aos pesos sinápticos. Um dos processos é introduzir o *fator de esquecimento* não linear à formula de ajuste sináptico $\Delta w_{kj}(n)$, como uma regra *generalizada do produto da atividade*, dada por:

$$\Delta w_{kj}(n) = \alpha y_k(n)[cx_j(n) - w_{kj}(n)] \quad (2.17)$$

O parâmetro $c = \frac{\eta}{\alpha}$. No caso do peso $w_{kj}(n)$ crescer até ao ponto onde:

$$cx_j(n) - w_{kj}(n) = 0 \quad (2.18)$$

é alcançado um *ponto de equilíbrio*, o que leva a que a atualização dos pesos termine.

2.4.4.3 Retro-propagação

As redes de multi-camadas, têm sido aplicadas de uma forma bem sucedida para resolver problemas com alguma dificuldade, efetuando o treino de forma supervisionada utilizando a regra delta. A ideia essencial deste algoritmo reside na utilização de uma função derivável, tipicamente uma função sigmóide. Matematicamente, o algoritmo faz uso das regras de derivação composta. Este princípio permite propagar os sinais no sentido proativo, da entrada para a saída, como é usual, mas também no sentido retroativo, com a retro-propagação do erro da saída para as camadas anteriores.

O método mais usual, é calcular o erro como a soma quadrática dos erros de cada elemento processador de saída e fazer a sua retro-propagação na rede, de forma a modificar cada peso sináptico. O procedimento é repetido até serem atingidos os pares de entrada e saída desejados na aprendizagem supervisionada. Se o erro for inferior a uma tolerância determinada e admissível, afirma-se que ocorreu a convergência do sistema neuronal. Como é suposto, a aprendizagem consiste em minimizar o erro quadrático, por uma descida de gradiente. A dificuldade em efetuar esta descida nas redes de múltiplas camadas com uma função de ativação limitada, consiste no cálculo do erro quadrático em relação a um dado peso. Este problema é facilmente resolvido, com a aplicação de uma função de ativação derivável.

Para formalizar matematicamente este algoritmo de retro-propagação, partiu-se do erro $\delta(w_j)$ da rede neuronal em função dos pesos. Pretende-se variar os valores w_j de forma a que a função decresça rapidamente: segundo o princípio da *descida de gradiente*, a variação de cada peso deve ser proporcional à derivada parcial do erro em relação a esse peso:

$$\Delta w_{kj} = \eta \frac{\partial \delta_j}{\partial w_{kj}} \quad (2.19)$$

onde η é o fator de proporcionalidade.

Usualmente aplica-se a função de ativação sigmóide, visto esta ser derivável e permitir controlar o repetitivo ganho. O ganho à saída de um elemento de processamento é fornecido por:

$$y = \frac{1}{1 + e^{-x}} \quad (2.20)$$

da qual se obtém a derivada parcial $\partial y / \partial x = y(1 - y)$.

O erro quadrático utilizado no treino é expresso pela relação:

$$\delta_j = (y_j^d - y_j)^2 \quad (2.21)$$

através da qual possibilita definir a função energética de *Lyapunov*,

$$E_j = \frac{1}{2}(y_j^d - y_j)^2 \quad (2.22)$$

a qual se procura minimizar.

A saída atual da rede y_j depende dos pesos das conexões com base no avanço da convergência a ser considerada. De tal forma, o método de descida de gradiente, tendo por base as derivadas parciais do erro em relação aos pesos sinápticos, pode ser aplicado. Para obter o valor do erro, utiliza-se o cálculo dos erros parciais em cada um dos elementos do conjunto de treino. Efetua-se assim, a soma das alterações parciais que foram obtidas para cada entrada do conjunto de treino, tendo como finalidade a alteração final de cada peso.

Partindo do erro $E_j[y_j(w_{ij})]$ obtém-se (seguindo o esquema da Figura 2.13):

$$\frac{\partial E_j}{\partial w_{ij}} = \frac{\partial E_j}{\partial \tilde{y}_j} \cdot \frac{\partial y_j}{\partial w_{ij}} \quad (2.23)$$

e como $y_j = f(x_j)$, para o estado de ativação $x_j = \sum_i w_{ij} \tilde{y}_i$, pode-se deduzir:

$$\frac{\partial y_j}{\partial w_{ij}} = \frac{\partial f(x_j)}{\partial x_j} \cdot \frac{\partial x_j}{\partial w_{ij}} = \frac{\partial f(x_j)}{\partial x_j} y_i \quad (2.24)$$

O cálculo de $\partial E_j / \partial y_j$ faz com que, as sucessivas camadas intervenham. As alterações nas saídas dos elementos processadores da camada j propagam-se às saídas dos elementos da camada seguinte l , isto é, $E_l[y_l(y_j)]$ fornece:

$$\frac{\partial E_j}{\partial y_j} = \sum_l \frac{\partial E_l}{\partial \tilde{y}_l} \cdot \frac{\partial y_l}{\partial y_j} \quad (2.25)$$

sendo $y_l = f(x_l)$ e $x_l = \sum_l w_{jl} y_j$, permite obter:

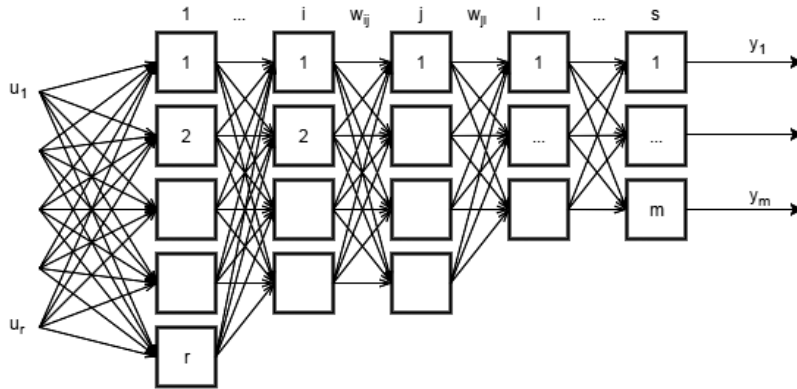


Figura 2.13: Esquema genérico de uma rede neural multi-camada.

$$\frac{\partial y_l}{\partial y_j} = \frac{\partial f(x_l)}{\partial x_l} \cdot \frac{\partial x_l}{\partial y_j} = \frac{\partial f(x_l)}{\partial x_l} w_{jl} \quad (2.26)$$

A aplicação da função logística define a expressão $\partial f(x)/\partial x = y(1 - y)$. Desta forma a derivada parcial do erro relativamente a uma saída da ultima camada, dada pela função de Lyapunov, chega-se à expressão:

$$\frac{\partial E_s}{\partial y_s} = \frac{1}{2} \frac{\partial f(y_s^d - y_s)^2}{\partial y_s} = y_s^d - y_s \quad (2.27)$$

Dado que, a variação dos pesos sinápticos depende do parâmetro referente à taxa de aprendizagem, pode-se estabelecer a proporcionalidade dada por:

$$\Delta w_{ij} = \eta \frac{\partial E_j}{\partial w_{ij}} \quad (2.28)$$

e tendo por base o cálculo da equação (2.23), obtém-se:

$$\Delta w_{ij} = \eta \frac{\partial E_j}{\partial y_j} y_j (1 - y_j) y_i \quad (2.29)$$

A equação (2.29), que se exprime em função do "benefício", definido por $\beta_j = \partial E_j / \partial y_j$, segundo a relação:

$$\Delta w_{ij} = \eta \beta_j y_j (1 - y_j) y_j \quad (2.30)$$

pode ser expressa de forma semelhante à regra delta,

$$\Delta w_{ij} = \eta \delta_j y_i \quad (2.31)$$

em que

$$\delta_j = \beta_j y_j (1 - y_j) \quad (2.32)$$

O "benefício", pode assumir as seguintes equações:

1. **Elementos processadores intermédios**, em que se utiliza $\beta_j = \partial E_j / \partial y_i$;

$$\beta_j = \sum_l \beta_l y_l (1 - y_l) w_{jl} \quad (2.33)$$

2. **Elementos processadores finais**, em que $\beta_s = \partial E_s / \partial y_s$, para $s = 1, \dots, m$.

Partindo de toda a análise feita e os resultados obtidos, é possível estabelecer o **algoritmo de retro-propagação**, tendo por base os passos seguintes:

1. Selecionar um par de treino do conjunto de treino e aplicar o vetor de entrada à rede neuronal.
2. Calcular a saída do sistema neuronal.
3. Calcular o benefício β_s para os elementos processadores terminais (da camada de saída do sistema neuronal), com $\beta_s = \partial E_s / \partial y_s$.

4. Calcular o benefício β_j para os elementos processadores intermédios (das camadas escondidas), com a equação (2.33).
5. Calcular a variação de cada peso, com a equação (2.30).
6. Ajustar cada um dos pesos $w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$, para a iteração k .
7. Repetir os passos anteriores para cada vetor do conjunto de treino, até se obter um erro satisfatório.

Para melhorar o algoritmo de retro-propagação, podem ser utilizadas as seguintes técnicas:

- Acrescentar uma polaridade a cada elemento processador, esta alteração leva a que a convergência seja mais rápida.
- Utilizar um fator α , designado de "momento", de maneira a adicionar ao ajuste dos pesos, uma parcela proporcional à variação de peso no passo anterior, através do momento adaptado, dado por:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} + \alpha \Delta w_{ij}(k), \quad 0 < \alpha < 1 \quad (2.34)$$

2.4.5 Tipo de Procedimento de Ajuste das Conexões Sinápticas

O treino da RNA pode ser efetuado de duas formas diferentes:

- Treino *Offline*: O ajuste dos pesos dá-se anteriormente à colocação em operação da RNA. Quando colocada em operação, os pesos sinápticos da rede são fixos.
- Treino *Online*: Neste modo, não há distinção entre a fase de treino e operação, isto porque os pesos variam de forma incremental sempre

que se apresenta um novo estímulo de entrada e o comportamento em cada instante da RNA depende dos valores atuais dos pesos sinápticos.

2.5 Aplicações das RNAs

Para mostrar algumas das capacidades das RNAs, ao longo desta secção são apresentados alguns exemplos de aplicações. O sucesso e a aplicabilidade das RNAs, nos diversos casos de aplicação, está dependente da natureza do problema a ser solucionado, bem como dos dados disponíveis à resolução do problema. Esta ideia deve estar sempre presente, de forma a destacar a complexidade dos problemas solucionados.

2.5.1 Previsão de Séries Temporais

As RNAs são uma ferramenta de modelação estatística, que tem uma ampla variedade de aplicações, incluindo a previsão de séries temporais. Exemplos que incluem a previsão de cargas de energia [13], risco médico [14, 15], previsão económica e financeira [16, 17, 18], e previsão de séries caóticas [19, 20]. Em grande parte dos casos, as RNAs apresentam um melhor desempenho do que outros tipos de aproximações. Por exemplo, Moody, Levin e Rehfuß [21] mostraram de forma convincente, a superioridade das RNAs na previsão de alguns indicadores da economia dos Estados Unidos da América; Outro caso, Hutchinson, Lo e Poggio [17] mostraram que a sua RNA supera o modelo de *Black-Scholes* em *delta-hedging*² dos preços diários na S&P; Giles, Lawrence e Toi [16], encontraram nas RNAs uma forma bastante assertiva, para previsão das taxas de câmbio internacionais.

²Estratégia utilizada para diminuir o risco associado às alterações de preços.

2.5.2 Processamento de Voz

O sinal de voz, é a comunicação de média distância fundamental para o ser humano. Este permite transmitir vários tipos de informação, como informação linguística, a identidade do locutor ou informação emocional. Para utilizar este meio eficazmente, tecnologias de processamento de voz têm vindo a ser estudadas e desenvolvidas.

Nos casos mais convencionais, as tecnologias de processamento de voz implicam áreas de estudo, como teoria da probabilidade, a teoria de decisão de Bayes [22], quantização de vetores [23] e análise de espetros [24].

Nos anos 80, com o reaparecimento dos conceitos associados às RNAs, deu-se o grande "boom" da utilização de RNAs no processamento de voz. Desde esses dias, e tendo por vantagens as capacidades não lineares das RNAs, a sua capacidade de discriminação e a capacidade adaptativa, estas foram extensamente exploradas no processamento de voz.

Os anos 80 foram muito produtivos, foi o crescimento das pesquisas na área de processamento de voz, utilizando RNAs, mas as pesquisas ainda se encontravam na sua infância. Alguns dos grandes desenvolvimentos dessa altura foram: A RNA de tempo de atraso [25]; o vetor de quantização de tolerância variável [26]; e a RNA de tempo de atraso para estados múltiplos [27].

Na verdade, o reconhecimento de voz, tem tido sempre um grande fluxo de aplicação com base nas RNAs. Recentemente, têm sido desenvolvidos tópicos, para o melhoramento do processo de reconhecimento de voz, como o método de descida probabilística generalizada [28], utilização de RNAs recorrentes [29], máquina para suporte de vetores [30] e tecnologias de separação de sinais [31].

2.5.3 Caracterização Visual de Imagens Recuperadas (CVIR)

Os dados multimédia, os quais incluem imagens digitais, vídeo, áudio, gráficos e texto, aparecem em diversas áreas de aplicação, como a indústria do entretenimento, educação, imagiologia médica e em sistemas de informação geográfica. Apesar das grandes diferenças entre as diferentes áreas, todas requerem acesso e manipulação de grandes quantidades de dados heterogêneos. Em particular, a CVIR recebeu uma grande atenção [32]. A pesquisa nesta área requer a caracterização dos conteúdos de imagem, utilizando características de baixo-nível. A junção das características de baixo-nível, permite gerar uma caracterização com melhor detalhe para pesquisa e recuperação de imagens [32].

As RNAs, fornecem uma excelente capacidade de aproximação não linear, o que as torna ideais para a aplicação na CVIR [33].

2.5.4 Processamento Biomédico de Imagens

Impulsionado pelo rápido desenvolvimento da imagiologia médica e o crescimento da disponibilidade de computadores, o processamento biomédico de imagens emergiu como uma das áreas de pesquisa mais pesquisadas na atualidade. Dada a qualidade da informação presentes, nas imagens, o processamento biomédico de imagens abriu novas fronteiras em novas áreas de investigação e conseguiu desenvolver novos desafios para os investigadores.

As RNAs, entre outras aproximações, demonstraram a sua importância para a área, o que levou ao crescimento da sua utilização. Estas são utilizadas para detetar e caracterizar padrões de doenças, análises (quantificação e segmentação), pressurização, modelação, estimação de movimentos, tomografia por emissão de positrões, ultra-som, radiografias e mamografias nas imagens [34].

Capítulo 3

Controlo e Casos de Estudo

As RNAs dada a sua versatilidade, mostraram-se uma mais valia na identificação e controlo de sistemas dinâmicos. As capacidades de aproximação generalizadas das redes de multi-camadas, tornou-se numa escolha popular para a modelação de sistemas não lineares e para a implementação de controladores não lineares. Ao longo deste capítulo é feita uma análise de alguns dos sistemas utilizados na área de controlo e dois casos de estudo em que são utilizadas RNAs.

3.1 Controlador com Estabilizador Fixo

O controlador com estabilizador fixo (Figura 3.1), foi proposto em [7],[8] e [9].

Este sistema foi aplicado para o controlo de trajetória de um braço robótico, onde um controlador de ganho proporcional foi utilizado como estabilizador de controlo da realimentação. Através da Figura 3.1 é possível constatar que o número total de entradas no processo é a soma do sinal de controlo da realimentação, o qual é calculado recorrendo a um modelo dinâmico inverso (a RNA). O modelo utiliza a trajetória desejada como entrada e o controlo realimentado como sinal de erro. Conforme o treino da RNA

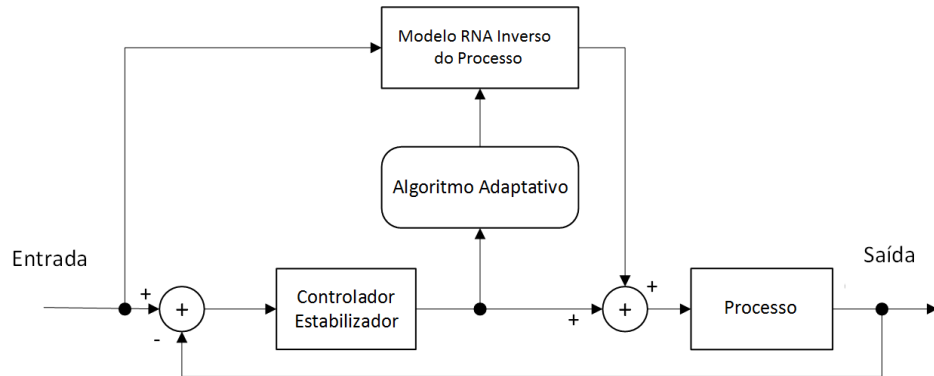


Figura 3.1: Controlador estabilizador [6].

avança, o erro da entrada converge para zero. A RNA irá aprender com o controlador realimentado.

A vantagem deste tipo de arquitetura, é a possibilidade de ter um arranque com um sistema estável, mesmo que a rede neuronal ainda não tenha sido treinada.

3.2 Controlo Inverso Adaptativo

A Figura 3.2 mostra o modelo de um controlo inverso adaptativo proposto em [10]. O algoritmo adaptativo recebe o erro entre a saída do processo e a referência de saída. Os parâmetros de controlo são ajustados para minimizar esse erro. A referência do modelo adaptativo, pode sofrer influências devido a ruídos provenientes do sensor ou distúrbios presentes no processo. Uma alternativa, que permite minimizar os ruídos e distúrbios inclui o modelo de uma RNA em paralelo com o processo. O modelo será treinado com as mesmas entradas do processo e produzir as mesmas saídas. Esse sinal irá entrar como um modelo inverso do processo para gerar um filtro de ruído e distúrbios que são subtraídos à entrada do processo. O objetivo é cancelar os distúrbios e os ruídos presentes no processo.

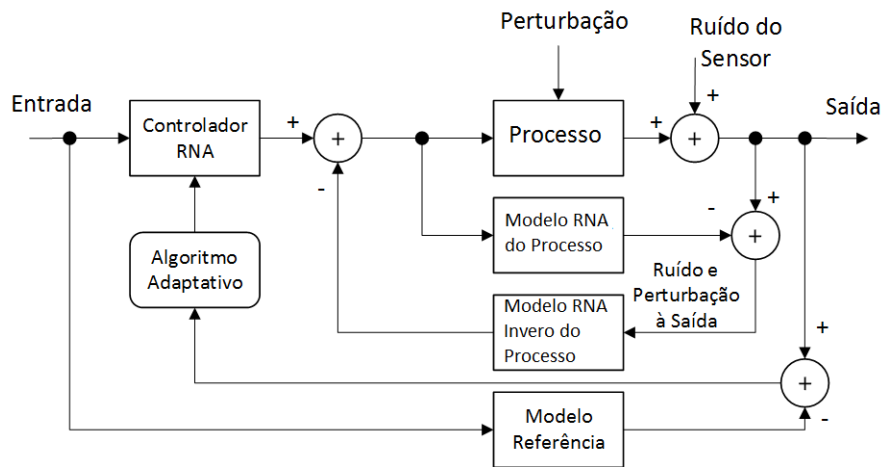


Figura 3.2: Sistema de controlo inverso adaptativo [6].

3.3 Modelo de Controlo de Referência ou Controlo Adaptativo Neuronal

Tal qual outras técnicas de controlo, o modelo de referência adaptativo (Figura 3.3) [11] usa duas RNAs: uma RNA de controlo e uma de modelo. A rede de modelo pode ser treinada *off-line* utilizando dados armazenados do processo. O controlador é treinado de forma adaptativa, com o objetivo de forçar a saída do processo para o valor desejado. O modelo em RNA é utilizado para prever os acontecimentos do controlo efetuado à saída do processo, isto permite atualizar os parâmetros de controlo.

3.4 Controlo Crítico Adaptativo

Como é possível constatar, na Figura 3.4, o controlador crítico adaptativo consiste em duas RNAs [12]. A primeira rede opera como um controlador inverso, o qual é designado por *rede de ação*. A segunda rede, é designada de *rede crítica*. A rede crítica é treinada para otimizar o desempenho futuro. O treino é efetuado utilizando a aprendizagem reforçada, a qual é uma

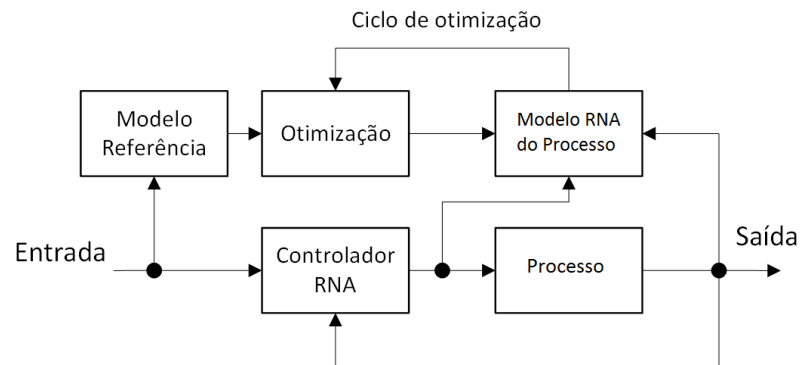


Figura 3.3: Sistema com modelo de controle de referência e controle adaptativo neural [6].

aproximação à programação dinâmica.

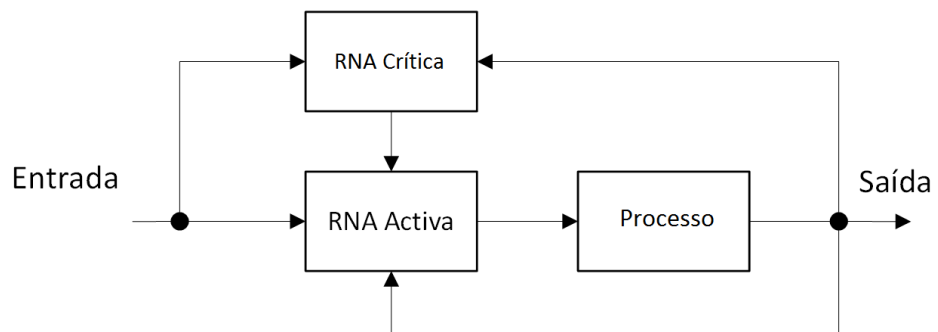


Figura 3.4: Modelo de controle crítico adaptativo [6].

3.5 Casos de Estudo

Ao longo desta secção são apresentados dois casos de estudo, que relacionam redes neurais, ao controlo de sistemas similares ao abordado neste projeto.

3.5.1 Aplicação de RNAs na Previsão Específica de Combustível Consumido e Temperatura dos Gases de Escape para um Motor a Diesel

O estudo feito em [35], mostra que a habilidade de uma rede neuronal, utilizando o mecanismo de aprendizagem de retro-propagação, é capaz de prever os consumos futuros de combustível e a temperatura dos gases de escape num motor Diesel. O modelo proposto é comparado com resultados experimentais.

Método de Recolha de Dados: No estudo efetuado, foi utilizado um motor Ricardo E6, com um cilindro, um *cooler* e uma câmara de pré combustão. O diagrama referente ao esquema experimental é representado pela Figura 3.5 e o motor real está apresentado na Figura 3.6. Os testes foram

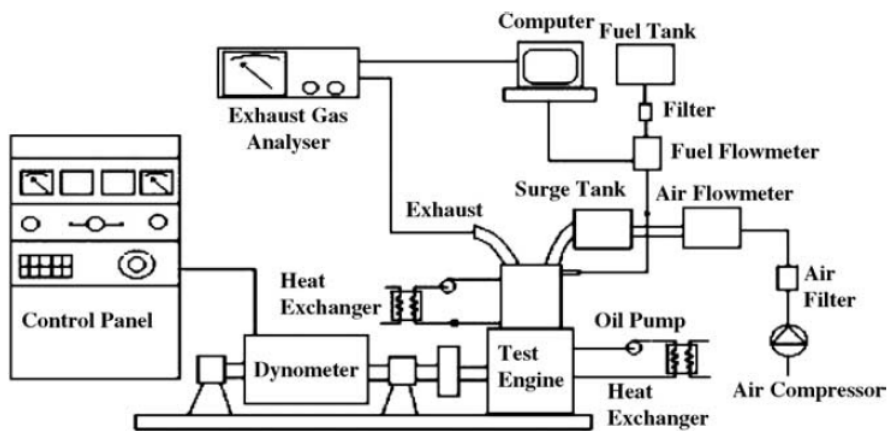


Figura 3.5: Diagrama de blocos do motor [35].

conduzidos com cargas variáveis e velocidades constantes, de 1000, 1400, 1800 e 2200 rotações por minuto. A injeção é constante, acontece no instante antes do ponto morto superior do pistão, com um ângulo de abertura de 38 graus. Em cada velocidade do motor, a injeção de combustível e carga do motor é alterada.

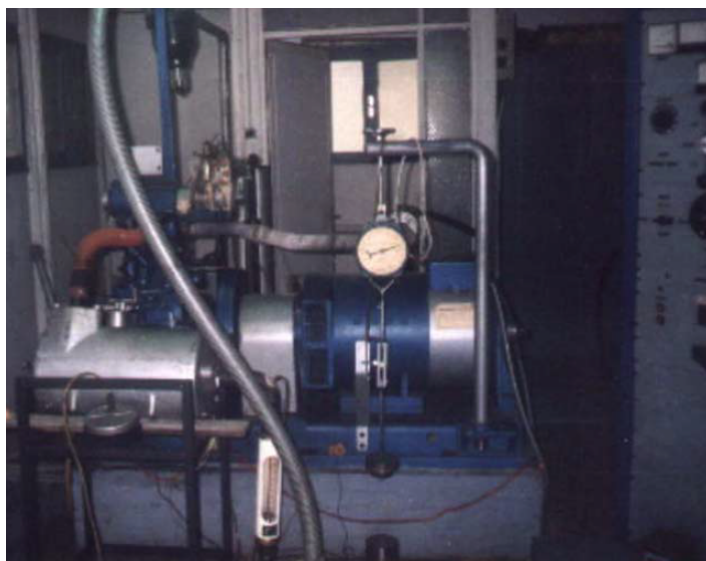


Figura 3.6: Fotografia do motor de testes [35].

RNA: Para resolver o problema foi utilizada a *Toolbox* do MATLAB 5.2 para criar as redes neurais. Foi utilizada a função de ativação sigmoide. As entradas do sistema determinam o número de neurónios a utilizar à saída da rede. Esta rede tem três neurónios à entrada e dois à saída. Na camada oculta foram utilizados sete neurónios, em que o alcance da saída de cada um deles podia variar entre 0,1 e 0,9.

O algoritmo de aprendizagem utilizado é o de retro-propagação, em que os parâmetros dos pesos e *bias* são atualizados de acordo com a otimização de *Levenberg-Marquardt*. O algoritmo de otimização utilizado, usou o erro quadrático para verificar o desempenho da rede. A formulação para o cálculo do erro quadrático médio é dado por:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - y_k)^2 \quad (3.1)$$

em que y_i é o valor previsto da amostra i , y_k é o valor de referência da amostra k e N o número de amostras.

Resultados e Conclusões: Foram utilizados 80 valores experimentais para a simulação. Em que 60 servem de referência e 20 para simulação da rede. Durante a validação alcançou-se um erro máximo relativo entre 1,93% e 2,36%. Depois de 900 ciclos de treino, o nível do erro era satisfatório e não existia uma redução significativa após esse número de amostras.

As Figuras 3.7 e 3.8 mostram os resultados experimentais da aplicação da RNA.

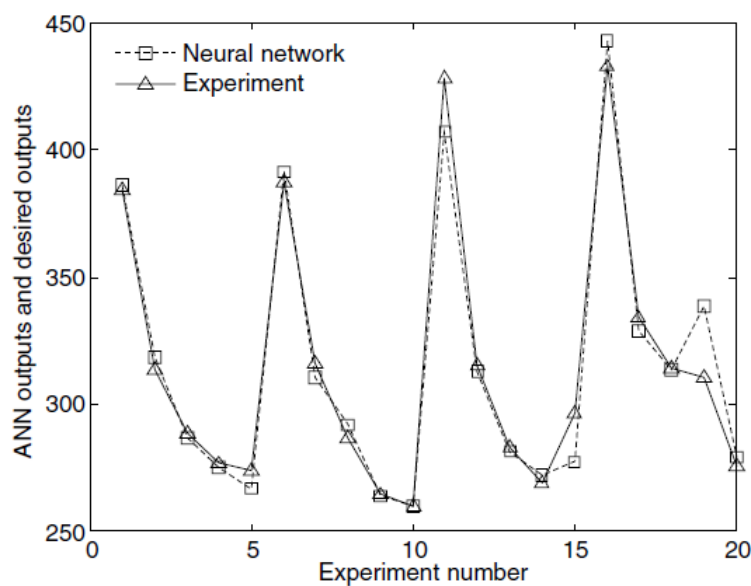


Figura 3.7: Previsão de consumos de combustível efetuado pela RNA na fase de testes [35].

Segundo o autor foi possível alcançar um resultado de acordo com o esperado, em que o erro era inferior a 2%. Esta aproximação pode ser utilizada como alternativa para avaliar os parâmetros de funcionamento de um motor.

3.5.2 Previsão do Desempenho de um Motor que Utiliza Biocombustíveis

O bio-diesel, o bio-etanol e bio-gás são alguns dos combustíveis alternativos mais importantes, produzidos de origem biológica. O objetivo deste artigo

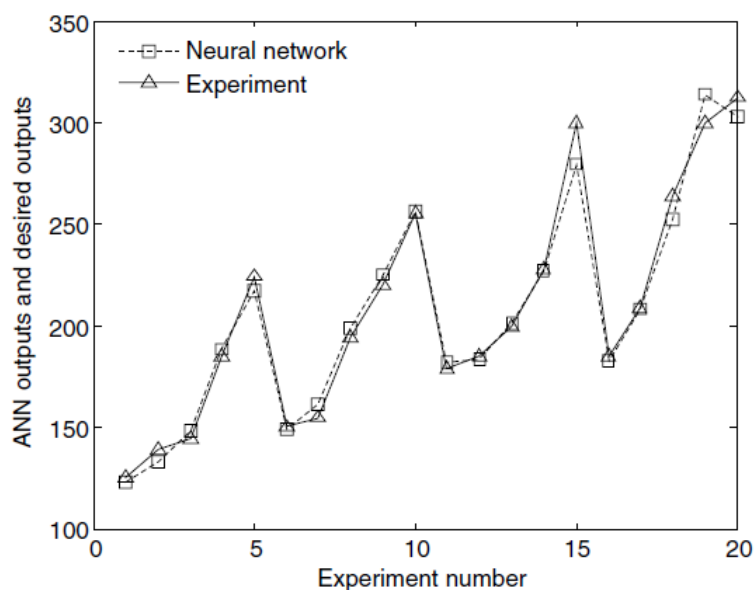


Figura 3.8: Previsão de temperatura de escape efetuado pela RNA na fase de testes [35].

tem por fim estudar o efeito do biocombustível no desempenho de um motor, tendo por ferramenta de análise as RNAs [36].

Neste estudo, as RNAs são utilizadas para mostrar a sua aplicabilidade no sector dos combustíveis (mais especificamente no ramo automóvel), bem como em diferentes áreas de tecnologia, com o objetivo de superar as dificuldades dos experimentos, minimizar o custo, o tempo e resíduos, aquando à produção destes.

Trabalho Experimental: Para iniciar o processo experimental várias amostras foram preparadas com volumes diferentes de constituintes. Foram produzidos três tipos de bio-combustíveis, com combustível diesel, metil-aspirado, éster e misturas de bio-etanol. Os combustíveis produzidos são os identificados pela Tabela 3.1, estes foram testados na unidade de testes da Turkish Motor Industry.

Tabela 3.1: Constituição dos combustíveis para testes e respetivas percentagens em termos de volume [36].

B100	B20	E5-B20	E10-B20	E15-B20
100% Bio-diesel	20% bio-diesel 80% diesel	5% etanol 20% bio-diesel 75% diesel	10% etanol 20% bio-diesel 70% diesel	15% etanol 20% bio-diesel 65% diesel

RNA: A RNA utilizada, tem por base de aprendizagem o algoritmo de retro-propagação, onde é utilizado o erro quadrático, equação (3.1), como parâmetro de desempenho da rede.

A RNA desenvolvida, é constituída por três camadas, uma camada de entrada (com dois neurónios: rpm e tipo de combustível), uma camada oculta e uma camada de saída (com quatro neurónios: combustível consumido (B), combustível consumido especificado (be), potência e binário). Foi utilizada a função de ativação sigmóide. As saídas dos neurónios foram limitadas entre os valores de 0 e 1. O melhor desempenho da rede foi obtido após 2000 ciclos de treino e a estrutura da rede que melhor se ajustou, apresentava 28 neurónios ocultos.

Resultados e Conclusões: Como resultado, a partir das análises estatísticas que foram realizadas, foi possível obter os resultados das Figuras 3.9, 3.10, 3.11 e 3.12. Com a utilização das redes neuronais foi possível alcançar uma estimativa muito próxima dos resultados reais para o desempenho dos combustíveis, com um desempenho a rondar os 95%.

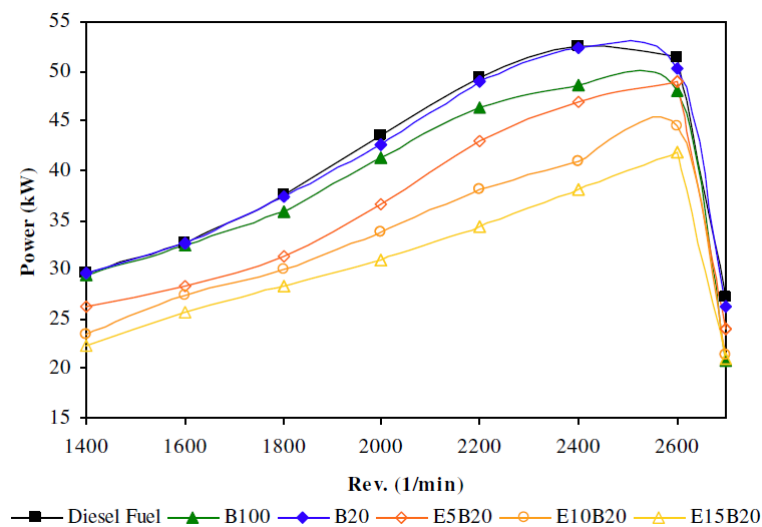


Figura 3.9: Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para a potência [36].

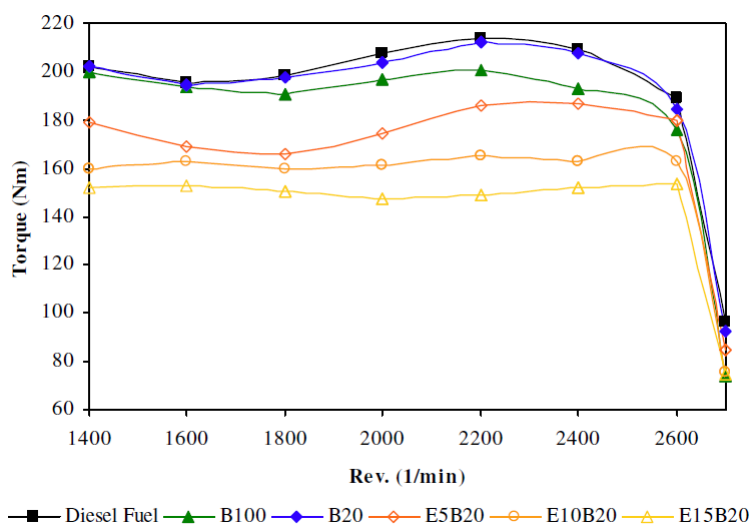


Figura 3.10: Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o binário [36].

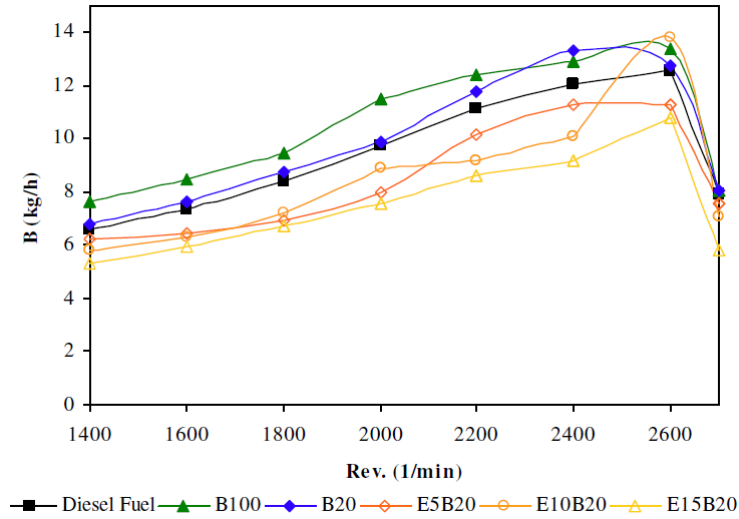


Figura 3.11: Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o consumo de combustível [36].

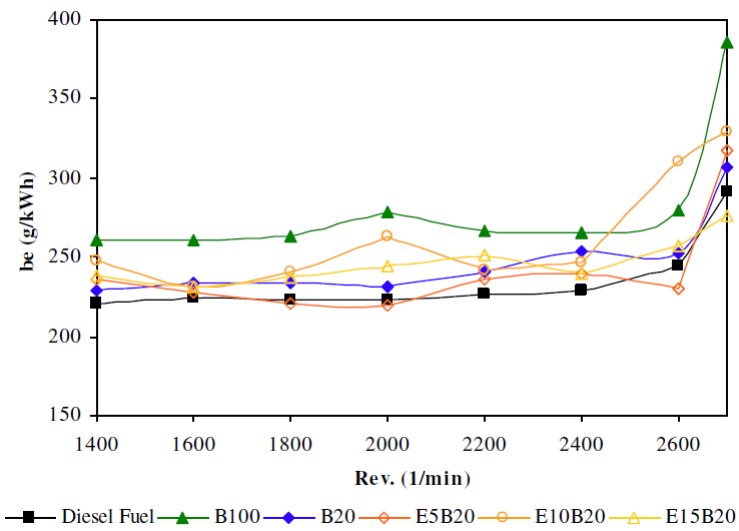


Figura 3.12: Comparação dos valores estimados e valores experimentais em termos de tempo obtidos através da RNA, para o consumo de combustível específico [36].

Capítulo 4

Software de Desenvolvimento

O crescimento das RNAs levou ao desenvolvimento de *software* específico para a criação e desenvolvimento das redes. Ao longo deste capítulo são apresentadas algumas opções de *software* existente para as RNAs, e é feita uma análise mais detalhada ao *software* de desenvolvimento MATLAB, mais especificamente à *Toolbox* de RNAs incorporada.

4.1 Emergent

O *software* Emergent Neural Network Simulation System (Figura 4.1), é um simulador para a criação de RNAs complexas, modelos sofisticados e aprendizagem cognitiva. Esta ferramenta inclui um ambiente totalmente gráfico para a criação das redes e amostras das entradas e saídas para simulação de redes mais generalistas ou apenas para testes. Inclui também um conjunto de ferramentas de análise que permitem efetuar uma análise mais pormenorizada do funcionamento da RNA criada. Desenvolvida em C++, esta é disponibilizada para os sistemas operativos Windows, Mac e Linux/Unix.

Os algoritmos suportados por esta ferramenta são diversos, enumerando-os: Retro-Propagação, algoritmos de auto-organização (aprendizagem de Hebb, Kohonen e aprendizagem competitiva), algoritmos de restrição (Bol-

man, Hopfield) e o algoritmo de Leabra, o qual integra elementos de todos os algoritmos descritos.

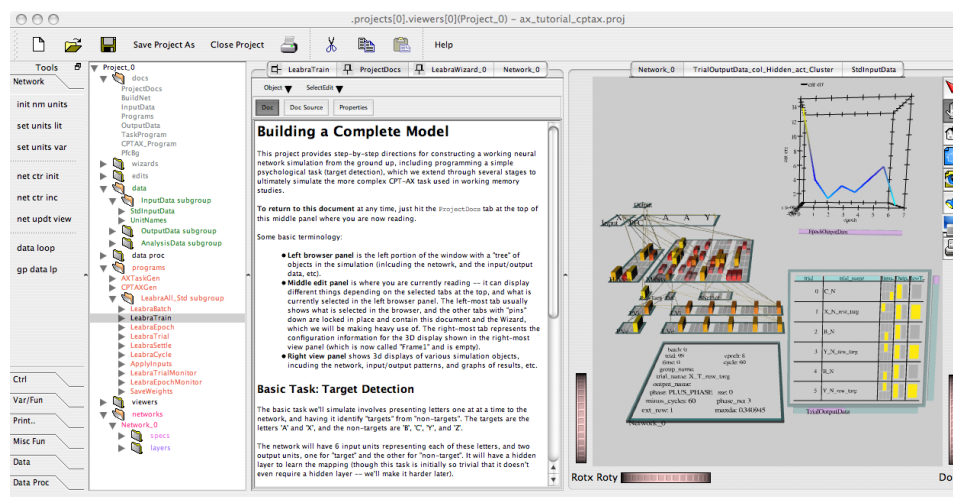


Figura 4.1: Imagem do *layout* do *software* Emergent, com alguns exemplos de amostra [37].

As principais características do Emergent são:

- *Software open source*;
- Fornece muitas capacidades para visualização e criação de estruturas para as RNAs, diminuindo o esforço na escrita de código;
- Interface apelativa e estruturada, o que torna acessível a qualquer criador editar redes de outros autores e adaptá-las;
- Suporta processamento paralelo e distribuição de memória;
- Inúmeras bibliotecas já desenvolvidas, dedicadas a diferentes áreas de investigação.

4.2 Brain Simulator

O *software* Brain Simulator (Figura 4.2) desenvolvido pela GoodAI, foi projetado para simplificar a colaboração, teste e implementação de novas teorias,

e visualizar facilmente experiências e dados. Não existe a necessidade de saber programar ou ter conhecimentos matemáticos para usar o Brain Simulator. A GoodAI, ainda se encontra a melhorar continuamente a plataforma, estando atualmente a investigar e desenvolver a aplicação com base no *feedback* dos utilizadores e também tenta expandir para diferentes sistemas operativos (atualmente disponível para o sistema operativo Windows).

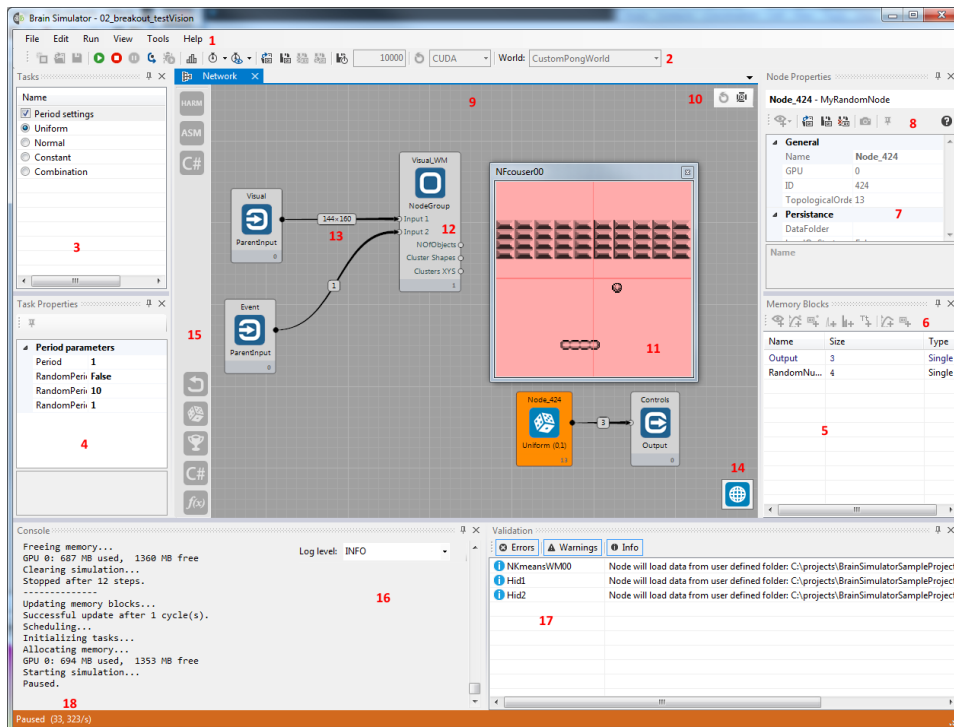


Figura 4.2: Imagem do *layout* do *software* Brain Simulator [38].

Os algoritmos de aprendizagem, até agora desenvolvidos para este *software*, têm por base o algoritmo de aprendizagem recursivo e a técnica de retro-propagação.

As principais características do Brain Simulator são:

- A capacidade de projetar rapidamente arquiteturas e protótipo de RNAs;
- Instantaneamente testar diferentes hipóteses;

- Apresenta modelos de teste para simulação e testes;
- Ambiente atrativo e de simples manuseamento.

4.3 MATLAB - *Neural Network Toolbox*

O MATLAB é um ambiente de computação e desenvolvimento de aplicações, totalmente integrado e orientado para levar a cabo projetos que implicam cálculos matemáticos complexos e a sua análise. Esta ferramenta possibilita inúmeras funcionalidades, como análise numérica, cálculo matricial, processamento de sinais, visualização gráfica, entre outras. Com esta ferramenta é possível, expressar problemas e suas soluções de modo semelhante àquele em que se escreviam tradicionalmente, sem necessidade de fazer uso de técnicas de programação tradicionais.

A criação do nome MATLAB resultou da contração das palavras "MA-Trix LABoratory" e foi inicialmente concebido para proporcionar fácil acesso às bibliotecas LINPACK e EISPACK, que representavam duas das mais importantes linguagens em computação e cálculo matricial.

O MATLAB dispõe também de um amplo conjunto de programas de apoio especializados, denominados *Toolboxes* e de interfaces gráficas, que estendem significativamente o número de funções incorporadas no programa principal. Estas interfaces gráficas cobrem praticamente todas as áreas principais no mundo da engenharia, destacando entre elas a interface gráfica de processamento de imagens, sinais, controlo robusto, estatística, análise financeira, cálculo matemático simbólico, redes neuronais, lógica difusa, identificação de sistemas, simulação de sistemas dinâmicos e algoritmos genéticos.

Nesta secção é feita uma análise às interfaces gráficas de RNAs, *nstart* e *nntool* e à implementação utilizando comandos MATLAB.

4.3.1 Interfaces NNTOOL e NNSTART

De forma a facilitar o uso da *Toolbox*, estão à disposição do utilizador as interfaces *nntool* e *nntstart*. Estas podem ser executadas na linha de comandos do MATLAB, e fornecem interfaces gráficas para facilitar a construção e simulação de redes neuronais, tornando o processo mais atrativo e intuitivo, para o utilizador.

4.3.1.1 NNTOOL

Iniciando a análise ao comando, *nntool*, este apresenta como interface a Figura 4.3. Nesta janela, destacam-se as zonas onde aparecem os dados de entrada, dados desejados e as RNAs criadas, estes correspondem aos campos *Input Data*, *Target Data* e *Networks*, respetivamente. Nesta janela é possível criar uma nova rede, simular e importar ou exportar os dados para o espaço de trabalho do MATLAB.

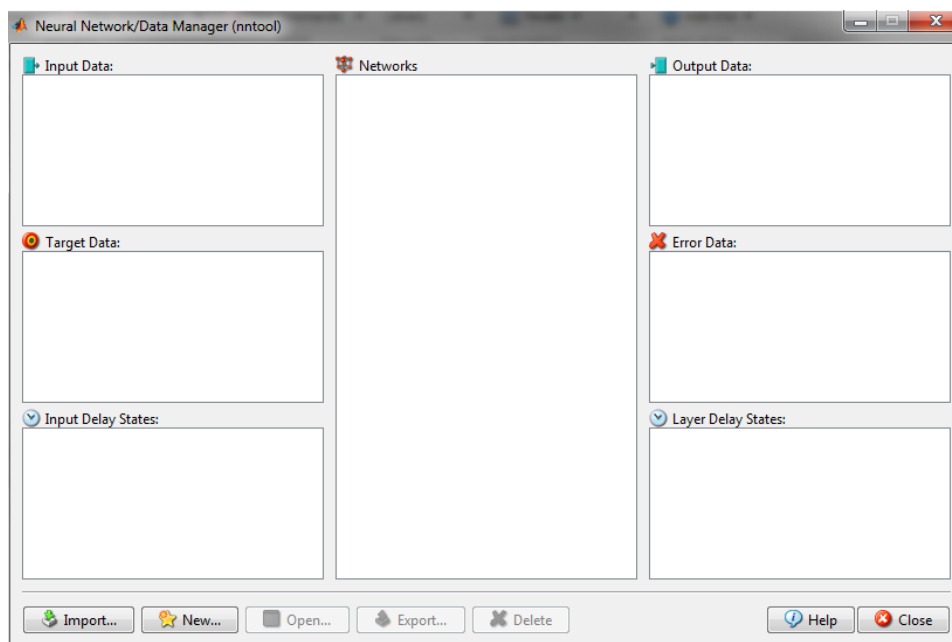


Figura 4.3: Imagem referente ao *layout* da *nntool*.

Para criar uma nova rede, é necessário pressionar o botão *New*. Depois dessa ação, é aberta uma nova janela, com duas abas. A primeira aba, *Network* (Figura 4.4), contém todas as configurações para a criação da RNA pretendida. Aparecem como opções de seleção e edição: dados de entrada, dados desejados, função de treino, função de aprendizagem, função de desempenho, número de camadas, respetivo número de neurónios e funções de ativação. Ao pressionar a aba *Data* surge a interface representada pela Figura 4.5. A nível estrutural, este separador é dividido em três áreas distintas. Cada uma dessas áreas representa um aspeto do conjunto: nome, valores e tipo. O parâmetro tipo permite classificar os valores do conjunto como sendo: de entrada, valores desejados, estados de atraso de entrada, estado de atraso de camada, saída ou erros.

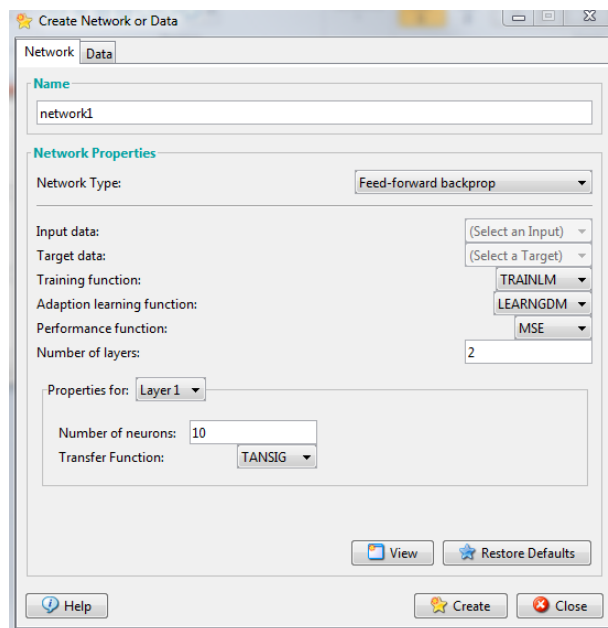


Figura 4.4: Aba de configuração da rede.

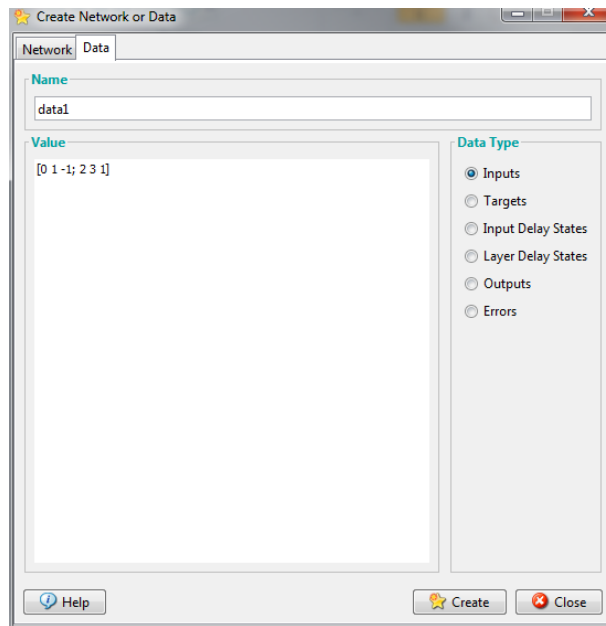


Figura 4.5: Aba de edição de dados da rede.

4.3.1.2 NNSTART

O NNSTART é a interface de RNAs introduzida mais recentemente. Ao executar o comando *nnstart*, é aberta a janela representada pela Figura 4.6. A interface, disponibiliza quatro tipos de aplicações:

- *Fitting app* - Ajuste à entrada-saída e curva pretendida;
- *Pattern Recognition app* - Reconhecimento e classificação de padrões;
- *Clustering app* - Associação e extração de características;
- *Time Series app* - Identificação de sistemas e modelação.

Escolhendo uma das aplicações surge uma página com a estrutura da rede escolhida, onde é dada uma explicação sobre o modelo de rede selecionado. A Figura 4.7 representa a escolha da *Fitting app*.

Pressionando *Next* surge uma nova página, Figura 4.8. Nesta é possível selecionar os dados a utilizar, ou utilizar dados do modelo já preparados para

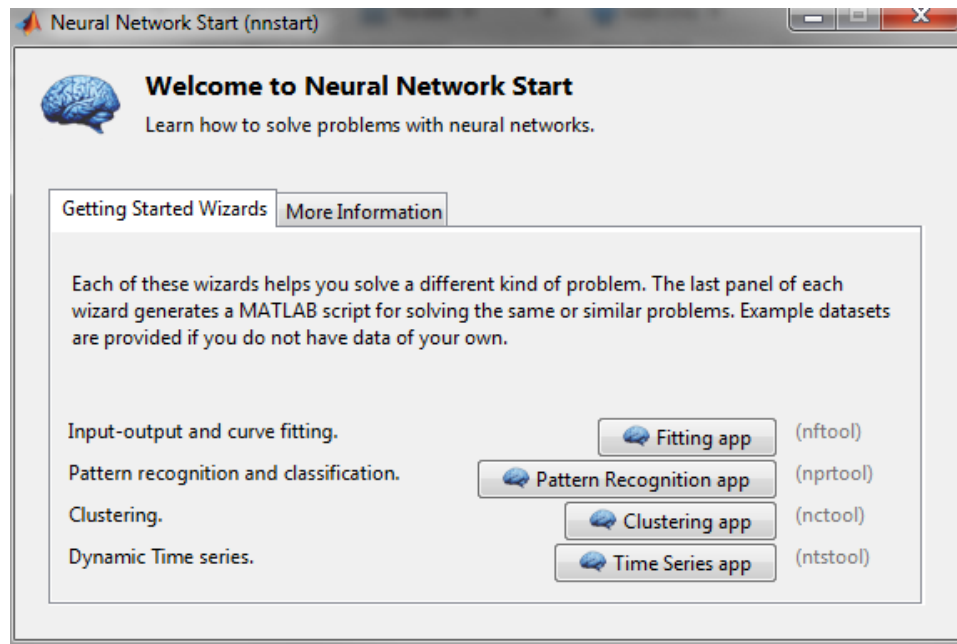
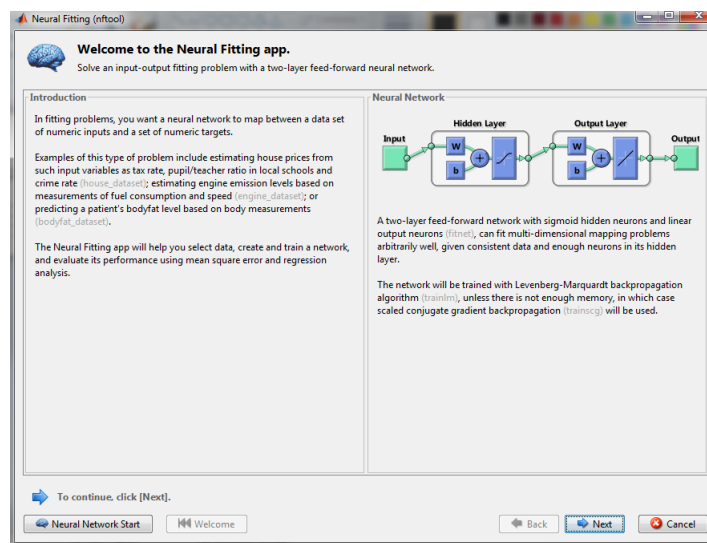
Figura 4.6: Imagem referente ao *layout* do *nnstart*.

Figura 4.7: Página de introdução com esquema da RNA.

simular um exemplo (o caso dos passos seguintes).

Estando já os parâmetros escolhidos é possível avançar para a próxima página, Figura 4.9. Nesta página surge os limites para validação da rede,

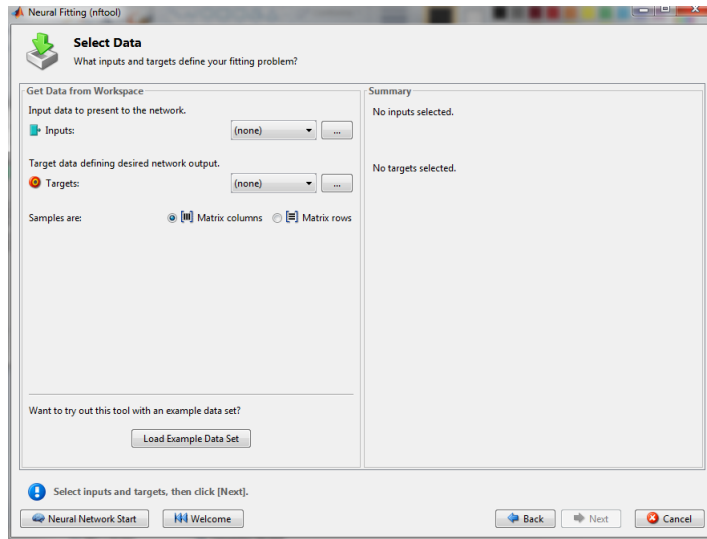


Figura 4.8: Página para introdução dos dados da experiência.

Training, *Validation* e *Testing*. Os dois primeiros parâmetros são utilizados para estabelecer o critério de paragem para o treino da rede. O terceiro e último parâmetro, fornece a informação sobre o desempenho da rede, durante e após o treino.

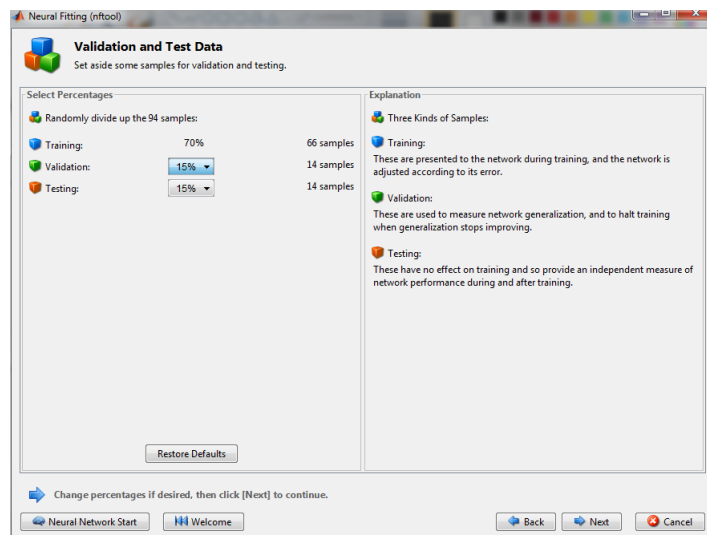


Figura 4.9: Página para definir os parâmetros de validação.

Finalizado o ponto anterior, avança-se para a próxima janela, Figura 4.10. Nesta janela é possível definir a função de treino e o número de neurónios da camada oculta.

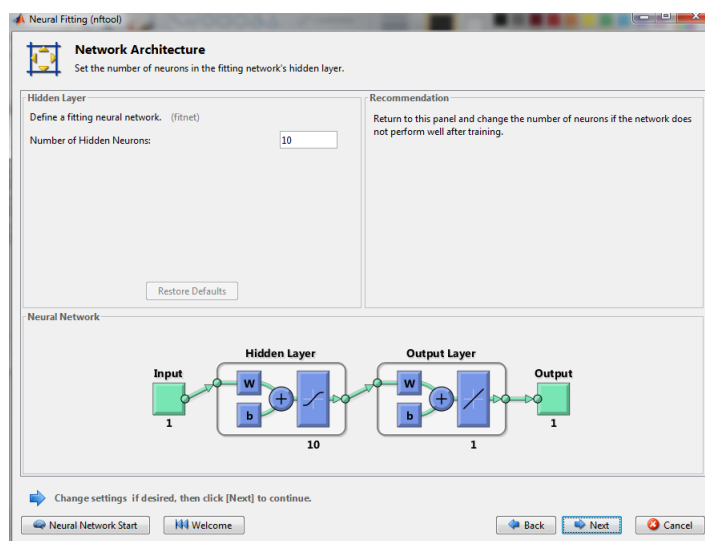


Figura 4.10: Página para definir a arquitetura da RNA.

Após estes passos a RNA está definida, avança-se para o treino. Na página de treino, Figura 4.11, é possível definir o algoritmo de treino a utilizar. De seguida é possível treinar a rede, acionando o botão *Train*. Logo de seguida, surge uma janela para análise do desempenho da rede, durante e após o treino, Figura 4.12.

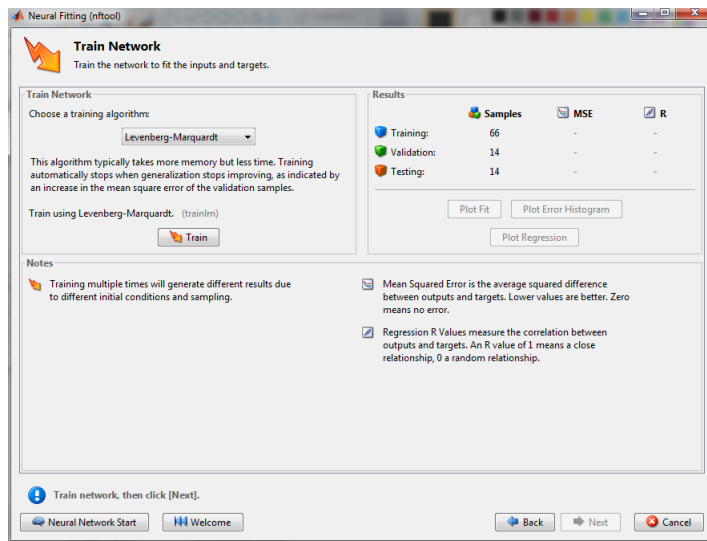


Figura 4.11: Página para treino da RNA.

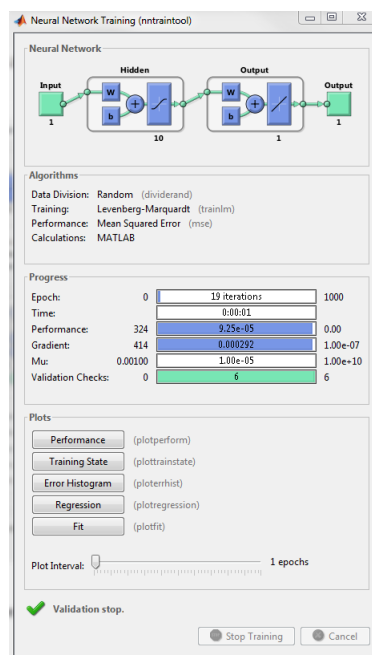


Figura 4.12: Página para análise do treino da RNA.

Terminado o treino, surge um painel (Figura 4.13) que permite treinar novamente, ajustar o tamanho da rede e importar mais dados para treino

(se disponíveis).

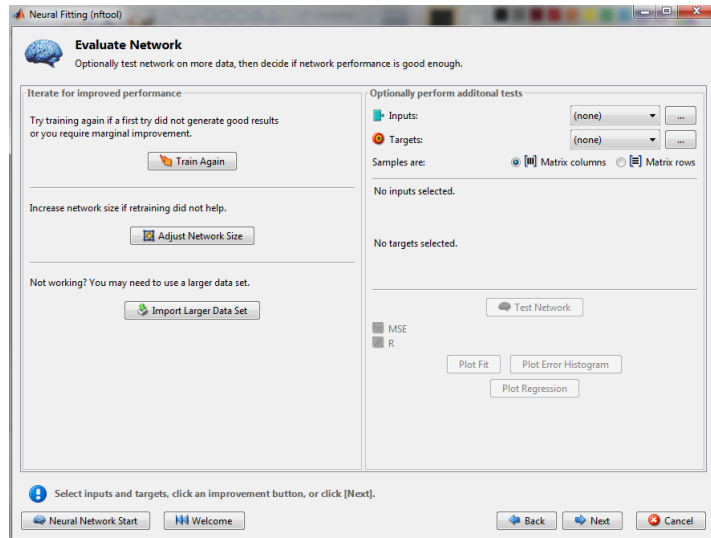


Figura 4.13: Página para avaliação da RNA.

De forma a simplificar a implementação da rede, é possível avançar para uma nova página, Figura 4.14. Nesta página, é possível criar uma função que permite aplicar a RNA, ou gerar um bloco para utilizar em *Simulink*.

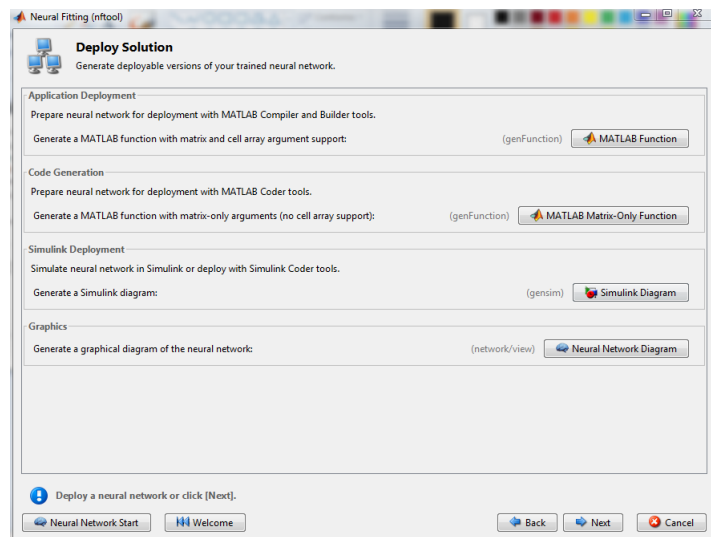


Figura 4.14: Página onde é possível gerar a rede para implementação.

A última página, Figura 4.15, permite guardar *scripts* na área de trabalho do MATLAB. Pode-se optar, por um primeiro *script*, menos detalhado e mais fácil de editar, ou um segundo *script* com todos os detalhes da rede, em que requer um conhecimento mais alargado sobre a *Toolbox*.

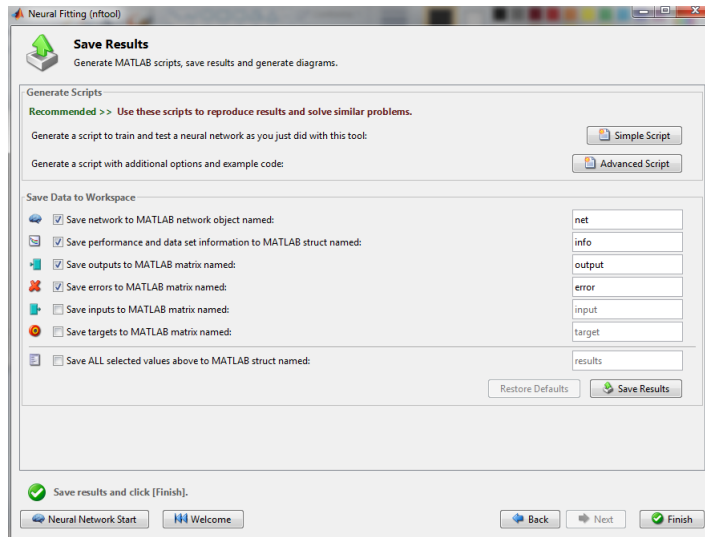


Figura 4.15: Imagem referente à página, para guardar resultados.

4.3.2 Comandos MATLAB

A par das interfaces que simplificam a implementação das RNAs, o MATLAB disponibiliza a opção de desenvolvimento da rede através de comandos. Esta opção, apresenta como grande vantagem, a possibilidade de controlar uma maior quantidade de parâmetros, e diversificar os tipos de estruturas da rede.

4.3.2.1 Configuração das Camadas da Rede

Para poder criar uma rede é necessário gerar um novo objeto, neste caso vai ser designado de 'net'. Para gerar este objeto, é utilizado o excerto de código, descrito por:

```
1 net = network;
```

Como passo seguinte define-se o número de camadas, e número de neurónios por camada. Para definir estes parâmetros, utiliza-se a sequência de código seguinte:

```
1 net.numLayers = 2;  
2 net.layers{1}.size = 3;  
3 net.layers{2}.size = 1;
```

No próximo passo definem-se as ligações entre as camadas. Para definir a que camadas estão ligadas as camadas de entrada, utiliza-se o comando *inputConnect(i)*, em que *i* é o número da camada a conectar. As ligações entre as restantes camadas é efetuado através do comando, *layerConnect(i,j)*. Para este caso, se o comando for igualado a 1, então as saídas de *i* são ligadas às entradas de *j*. De seguida, são definidas as camadas de saída através do comando *outputConnect(i)*, em que *i* define a camada atribuída, e iguala-se a 1 para validar a escolha. Para terminar este passo, no caso de ser utilizado um algoritmo de aprendizagem supervisionado, define-se qual das camadas está ligada aos valores desejados, através do comando *targetConnect(i)* (este comando pode ser omitido, visto que o comando *outputConnect(i)* já define os valores de saída, como os valores desejados). A descrição dada, é aplicada através dos comandos seguintes:

```
1 net.inputConnect(1) = 1;  
2 net.layerConnect(2, 1) = 1;  
3 net.outputConnect(2) = 1;  
4 net.targetConnect(2) = 1;
```

Como passo seguinte, deve-se definir a função de ativação a utilizar. Para cada camada pode ser definida uma função diferente, como é definido no código seguinte:

```

1     net.layers{1}.transferFcn = 'logsig';
2     net.layers{2}.transferFcn = 'purelin';

```

4.3.2.2 Pesos e *Bias*

Estando a configuração das camadas da rede efetuadas, passa-se para a configuração dos **pesos** e **bias**. Através do comando *biasConnect(i)* configura-se cada uma das camadas, em que *i* indica a qual das camadas se atribui o valor de polarização. De forma a atribuir o mesmo valor a todas as camadas, utiliza-se o excerto de código a seguir apresentado:

```

1     net.biasConnect = [1 , 1];

```

Finalizado este passo, atribui-se os valores dos pesos das ligações. É recomendado, reiniciar todos os pesos da rede com o comando:

```

1     net = init(net);

```

Para definir os valores dos pesos é necessário utilizar o comando *initFcn*. No caso de o utilizador não ter a sua própria rotina de inicialização, iguala-se ao parâmetro *initlay*. Com isto, cada camada tem a sua rotina de inicialização. Para definir uma rotina a cada camada da rede, utiliza-se o comando *layeri.initFcn*.

4.3.2.3 Funções e Parâmetros de Treino

As etapas de *treino* e *adaptação* da rede, são dissociadas. Contudo ambas as funções são utilizadas para treinar a RNA, sendo possível utilizar as duas numa mesma RNA.

A distinção que mais se nota, entre os dois métodos, é referente ao treino incremental (corresponde à atualização dos pesos, depois de apresentada

cada uma das amostras do treino) *versus* o treino por lote (corresponde à atualização dos pesos depois de apresentado o conjunto de dados).

Quando se aplica a *adaptação* da rede, podem ser aplicados os dois tipos de treino. Exclui-se os casos em que, o conjunto de dados é constituído por duas matrizes de vetores de entrada, neste caso é utilizado o treino por lote e o caso em que o conjunto de treino é dado em forma de células, para esta situação utiliza-se o treino incremental.

De uma forma geral, grande parte dos casos utiliza o método de *treino*. Para tal é utilizada a função *trainFcn*, onde é definido o tipo de algoritmo. Através do comando *trainParam*, são definidos os parâmetros de treino. O excerto de código a seguir mostra um método de treino em gradiente descendente com *momentum*:

```
1 net.trainFcn = 'traingdm';
2 net.trainParam.lr = 0.1;
3 net.trainParam.mc = 0.9;
```

O parâmetro *lr* define a taxa de aprendizagem do algoritmo, enquanto que *mc* é o termo de *momentum*.

Através dos comandos *trainParam.epochs*, pode-se definir o número de iterações que o conjunto de dados pode ser treinado. O comando *trainParam.show*, representa o tempo entre relatórios do estado da função de treino. Como é descrito pelo código seguinte:

```
1 net.trainParam.epochs = 10000;
2 net.trainParam.show = 100;
```

Como passo seguinte, efetua-se o treino da rede, terminando com a simulação da mesma. Para tal é necessário existir uma matriz de entrada que contém os padrões de entrada e uma matriz de saída que contém os padrões de saída. A estrutura da matriz deve obedecer à estrutura da rede criada,

as dimensões dos dados de entrada/saída devem ter a mesma dimensão das entradas/saídas da rede. Seguindo-se esta regra, apresenta-se as matrizes P de entrada e a T de valores desejados, representadas no extrato de código abaixo:

```

1     P = {[0.5;0] [0.3;0.6] [0.2;0.4] [0.5;0.4] [0.2;0.6] [1;0] ...
          [0.1;0.2] [0.6;0.1] [0.9;0.1] [0.2;0.2]}
2     T = {0.5 0.9 0.6 0.9 0.8 1 0.3 0.7 1 0.4}

```

A relação entre estas duas matrizes é dada pela expressão $T = P_A + P_B$, em que o valor mínimo é 0 e o máximo é 1. Esta estratégia, permite verificar como a rede se comporta para valores aleatórios que possam ser colocados à entrada.

Para dar início ao treino é utilizado o comando *train*. Este comando apresenta como argumentos, o objeto criado ('net'), a matriz que contém os padrões de entrada e a matriz com os valores de saída desejados, como é descrito a seguir:

```

1     net = train(net,P,T);

```

Após a rede treinada, é necessário verificar a sua validade, isto é, se o seu desempenho é o esperado. Os dois métodos de verificação são o erro médio absoluto (*mae*) e o erro médio quadrático (*mse*). O primeiro método é mais utilizado em redes de classificação, enquanto que o segundo é mais utilizado em aproximação de funções. De tal forma a função de desempenho é definida com o comando *performFcn*, como no excerto de código seguinte:

```

1     net.performFcn = 'mse';

```

Para finalizar e simular a rede é utilizado o comando *sim*, que tem por argumentos o objeto que identifica a rede, e a matriz de entrada. Na variável

Y é guardado a resposta da saída da rede. Exemplificado pelo código:

```
1 Y = sim(net,P);
```

Como forma de analisar a rede é possível utilizar o comando `view(net)`, que apresenta uma ilustração da estrutura da rede.

Para simular a totalidade da rede pode ser utilizado o código completo, do exemplo:

```
1 %%Codigo para Desenvolver uma RNA
2 net=network; %Criacao do objeto da rede
3 net.numInputs = 1; %No de camadas de entrada
4 net.inputs{1}.size = 2; %No entradas na primeira camada
5 net.numLayers = 2; % No de camadas de neuronios
6 net.layers{1}.size = 3; % No de neuronios na camada 1
7 net.layers{2}.size = 1; % No de neuronios na camada 2
8 net.inputConnect(1) = 1; % Ligacao entre as entradas e ...
   camada 1
9 net.layerConnect(2,1) = 1; % Ligacao entre a camada 1 e 2
10 net.outputConnect(2) = 1; % Definicao que a camada de ...
   saida e a camada 2
11 net.layers{1}.transferFcn = 'logsig'; % Definir a funcao ...
   sigmoid para a camada 1
12 net.layers{2}.transferFcn = 'purelin'; % Definir a funcao ...
   linear para a camada 2
13 net.biasConnect = [1;1]; % Valor de polarizacao ligado as ...
   duas camadas
14 net.initFcn = 'initlay';
15 net.layers{1}.initFcn = 'initnw';
16 net.layers{2}.initFcn = 'initnw';
17 net = init(net)
18 view(net)
19 net.trainFcn = 'traingdm';
20 net.trainParam.lr = 0.1; % Taxa de aprendizagem
```

```

21 net.trainParam.mc = 0.9; % Momentum
22 net.trainParam.epochs = 10000; % Maximo de iteracoes
23 net.trainParam.show = 100; % Intervalo de iteracoes que ...
    sao exibidas
24
25 % Matrizes de entrada e saida -> valores desejados sao a ...
    soma dos valores de entrada
26 P = {[0.5;0] [0.3;0.6] [0.2;0.4] [0.5;0.4] [0.2;0.6] [1;0] ...
    [0.1;0.2] [0.6;0.1] [0.9;0.1] [0.2;0.2]}
27 T = {0.5 0.9 0.6 0.9 0.8 1 0.3 0.7 1 0.4}
28
29 % Comando de treino
30 net = train(net,P,T)
31 % Simulacao da rede
32 Y = sim(net,P)
33 % Graficos (Comparacao entre valores obtidos e desejados)
34 plot(1:10,cell2mat(T),'x') hold on plot(1:10,cell2mat(Y),'-o')
35 % TESTE
36 Resultado = net([a;b])

```

Como resultado da rede treinada e validada deve ser possível obter o gráfico apresentado na Figura 4.16.

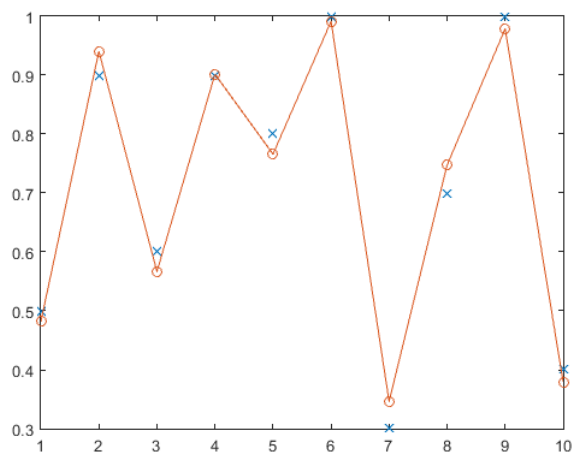


Figura 4.16: Gráfico com os resultados do treino.

Capítulo 5

Emissões, VGT e EGR

Ao longo deste capítulo, é feita uma contextualização entre emissões e os principais componentes mecânicos que intervêm no controlo destas, nos motores diesel (o VGT e a EGR).

5.1 Emissões

O ar que todos respiramos é o nome dado à mistura de gases que compõem a atmosfera da Terra. Estes gases (Figura 5.1) dividem-se em 20,95% de oxigénio (O_2), 0,03% de dióxido de carbono (CO_2), 78,09% de azoto (N_2) e 0,93% de partículas e outros gases. A poluição do ar consiste em vários elementos e gases que combinados com o oxigénio e nitrogénio conduzem à destruição do ar atmosférico.

Existem diversas fontes de poluição do ar [45], incluindo a produção de energia, a indústria e as atividades de transporte. Os motores de combustão interna são assim os principais responsáveis pela poluição do ar nos centros urbanos. São conhecidos três modos pelos quais estes motores emitem elementos considerados poluentes para a atmosfera:

- Por evaporação do combustível no depósito ou nos elementos do circuito de fornecimento de combustível;

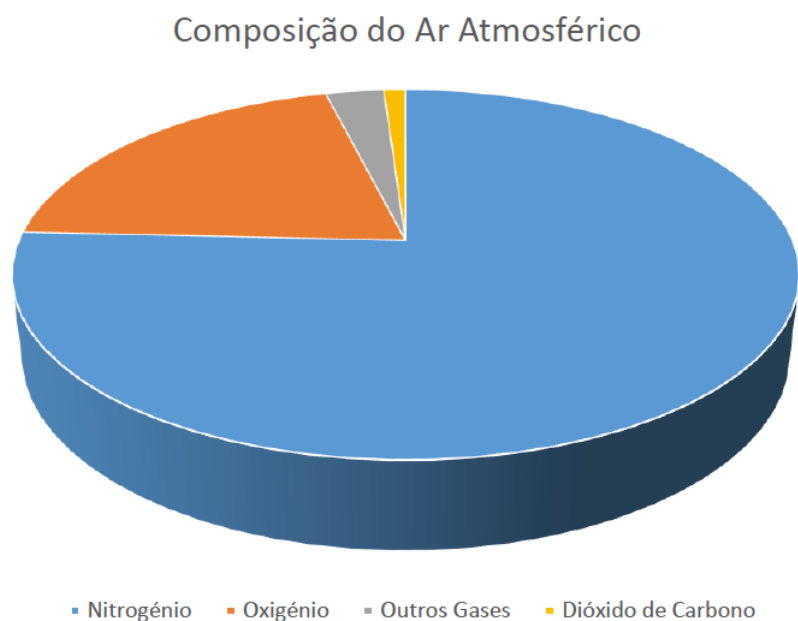


Figura 5.1: Gráfico da composição do ar atmosférico.

- Por emissão para a atmosfera dos gases contidos no cárter;
- Pelos gases de escape.

Tanto o gasóleo como a gasolina são primariamente compostos por hidrocarbonetos, que por sua vez são produtos resultantes do refinamento do petróleo que são misturados com aditivos químicos para o uso nos motores de combustão interna. O carbono no combustível produz os subprodutos CO (monóxido de carbono), HC (hidrocarbonetos), partículas, NO_x (óxidos de azoto) e óxidos sulfúricos. Os dois tipos de combustíveis referidos são assim misturas de hidrocarbonetos contendo átomos de hidrogénio e carbono. Num motor perfeito, o oxigénio converte o hidrogénio e o carbono do combustível em, respetivamente, água e dióxido de carbono.

As quantidades dos compostos poluentes que são emitidos pelos escapes dos veículos são dependentes do desenho do motor e das condições de utilização.

5.1.1 Relação Ar-Combustível

A emissão dos vários poluentes varia grandemente com a riqueza da mistura [45]. Para misturas pobres há pequena formação de CO. A sua formação é primeiramente dependente da riqueza da mistura, aumentando com o seu enriquecimento. Em misturas ricas é evidente que, não havendo suficiente oxigênio para oxidar todos os átomos de carbono, dar-se-á a produção de CO. Quanto aos HC, a sua emissão também aumenta com a riqueza da mistura, mas não tão intensamente como no caso do CO.

O nível de produção de NOx diminui para misturas pobres e ricas, sendo o máximo de emissão coincidente com misturas estequiométricas e levemente pobres. Desta forma, considerando um motor a trabalhar à temperatura normal de funcionamento (estando este quente), quanto mais pobre for a mistura, menor quantidade de poluentes emitirá, com a exceção dos HC, que aumenta a partir de um certo valor de pobreza da mistura, em virtude da ocorrência de ciclos sem combustão.

Se a temperatura do motor for mais baixa, o espectro de emissão de poluentes muda, havendo uma menor produção de NOx (temperatura mais baixa) e uma muito maior produção de HC. A elevação da produção dos hidrocarbonetos não queimados deve-se ao facto de as paredes dos cilindros e câmaras de combustão estarem a temperaturas baixas, o que origina que a extinção de chama se dê mais facilmente, originando que a mistura por queimar seja descarregada pelo escape.

No caso específico de motores a gasóleo, o processo de formação dos NOx é semelhante aos motores a gasolina mas, como as temperaturas de combustão são mais baixas, a produção desses compostos são menores. Já no caso dos HC, devido há presença de menos combustível em contato com as paredes do cilindro nos motores a gasóleo, a produção desses compostos é menor que nos motores a gasolina. A principal fonte de HC é o combustível que permanece no bico do injetor durante as elevadas pressões de combustão e

é daí desalojado durante a súbita expansão aquando da abertura da válvula de escape. As partículas sólidas, características destes motores, são resultantes de combustões incompletas dos hidrocarbonetos de cadeias longas, onde os átomos de carbono não têm tempo para reagirem com o oxigénio do ar. Felizmente parte destas partículas ardem nos gases quentes e oxidantes do escape.

5.1.2 Legislação Europeia

Entre 1990 e 2009 [44], as emissões de gases de efeito de estufa provenientes do tráfico automóvel na União Europeia (UE) aumentaram em 27%, passando de 12 para 17% das emissões totais na UE. Já as emissões dos outros gases poluentes têm diminuído desde 1990, mas muitas cidades continuam a exceder os limites impostos pela UE. Os limites para os níveis de emissões de poluentes para veículos ligeiros existem na UE desde o início dos anos 70, enquanto para veículos pesados surgiram no fim dos anos 80. Hoje em dia, os níveis de emissões são controladas sobre dois enquadramentos básicos: as *Euro standards* (ou normas Euro) e regulação para as emissões de dióxido de carbono. Os testes são efetuados tendo por padrões o teste denominado por European Transient Cycle (ETC).

As *Euro standards* regulam as emissões de óxidos de azoto (NOx), hidrocarbonetos (HC), monóxido de carbono (CO) e partículas (PM). Existem limitações separadas para veículos ligeiros (abaixo de 3500 kg) e veículos pesados, sendo designadas de *Euro* seguida por um número.

Em dezembro de 2006 foi aprovado o regulamento que hoje se encontra em vigor: norma Euro 5. O principal efeito da Euro 5 foi a redução dos limites das partículas (PM) nos automóveis Diesel em 80%, reforçando também os limites para o NOx. Quanto à Euro 6, norma que entrou em vigor em setembro de 2014, a principal mudança contempla uma maior redução das emissões de NOx dos motores Diesel para um nível mais próximo ao

que é exigido atualmente nos motores a gasolina. Repare-se que as normas Euro 5 e Euro 6 limitam o mesmo valor de emissões de NOx nos motores a gasolina, mas nos motores a gasóleo o limite cai de 180 mg/km para 80 mg/km, traduzindo-se numa diferença de apenas 20 mg/km de emissões daquele poluente entre automóveis diesel e gasolina.

Quanto às emissões de CO₂, a primeira regulamentação para a limitação do seu nível foi acordado em 2009. Esta regulamentação estipulou para 2015 um limite de 130 g/km para uma média de automóveis produzidos combinando todos os construtores. Para os construtores individuais foi permitido um limite superior de emissões de CO₂, dependendo do peso médio do automóvel das suas frotas. Quanto maior o peso médio dos automóveis vendidos pelo construtor, menos agressivo era o limite imposto. Uma nova norma similar para os novos automóveis ligeiros de mercadorias foi introduzida em 2011, definindo um limite de 175 g/km para 2017.

Por sua vez, no fim de 2013 o Parlamento Europeu e o Conselho da União Europeia chegaram a um acordo quanto a duas propostas a serem implementadas em 2020. O limite de emissões de CO₂ a partir desse ano estipulou-se em 95 g/km para os automóveis ligeiros de passageiros e de 147 g/km para os automóveis ligeiros de mercadorias.

A regulamentação de 2015 já levou a bons resultados: os níveis médios de emissões dos novos automóveis caiu de 160 g/km em 2006 para 132 g/km em 2012, uma redução de 17%. A taxa de redução anual é cerca do dobro do que era antes da introdução daquelas metas.

5.2 Descrição do VGT e da EGR

Nesta secção é efetuada uma breve descrição teórica dos mecanismos de VGT e da EGR, de forma a compreender o funcionamento destes, para aplicação no controlo de emissões.

5.2.1 O VGT

A presença de um turbo-compressor num motor, tem como função principal gerar mais binário, aumentando a quantidade de ar que entra no motor. Este método, permite que um motor mais pequeno em conjunto com um turbo-compressor, pode ser aplicado numa sistema no qual seria necessário um motor de maior dimensão sem turbo-compressor. A constituição do turbo-compressor consiste na junção de uma turbina e de um compressor (Figura 5.2).

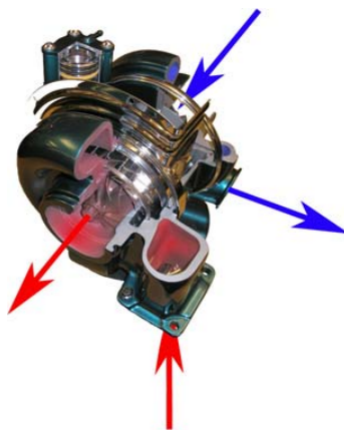


Figura 5.2: Vista orientada de um VGT [39].

A turbina, com o seu rotor produz binário, através do fluxo de massa que passa. O movimento gerado é utilizado pelo compressor para aumentar a pressão no motor. A utilização de um VGT em relação ao turbo mecânico usual, possibilita controlar o fluxo de massa na turbina, dando a possibilidade de controlar o binário produzido pelo motor. Este controlo é feito através da abertura das válvulas que controlam o fluxo. Utilizando este modelo de turbo-compressor, a turbina pode ser controlada de forma mais eficiente e eficaz, o que permite acompanhar as necessidades do motor.

As válvulas do VGT (Figura 5.3), nunca podem ser completamente fechadas, esta regra é imposta pelos gases de escape que têm de ser recirculados

(para serem libertados do circuito) e a rotação da turbina deve ser mantida (isto para não ocorrer um arranque bruto que a possa danificar, num estado quase parada).

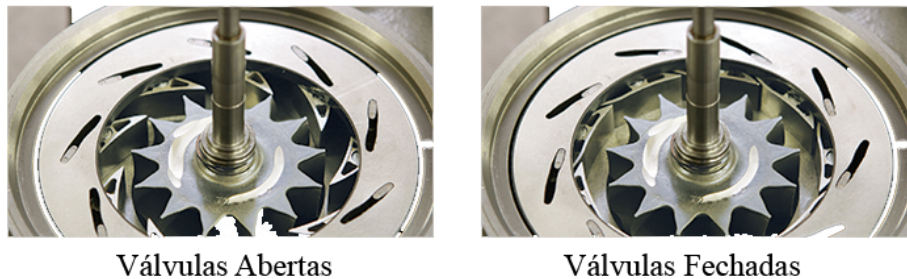


Figura 5.3: Válvulas do VGT.

O comportamento de um VGT, é extremamente não linear, este acontecimento é devido à constituição das válvulas que controlam o fluxo de ar na turbina. As válvulas não têm uma forma uniforme, o que leva a que o movimento do ar mude de forma não linear. Uma alteração na posição das válvulas do VGT quando fechadas tem uma maior influência do que estando totalmente abertas.

5.3 EGR

O circuito de EGR é o método de recircular os gases de escape, de volta para os cilindros do motor. O fundamento para a utilização deste sistema é a redução das emissões de Óxidos de Azoto (NOx) do motor. As partículas de NOx são criadas a altas temperaturas quando existe oxigénio disponível. Os gases de escape contêm muito pouco oxigénio, comparado com o ar ambiente. A substituição do ar ambiente pelos gases de escape, significa menor quantidade de oxigénio presente nos cilindros do motor, o que se traduz numa menor produção de NOx.

A quantidade de gases de escape que são recirculados são controlados por uma válvula (Figura 5.4). A válvula tem um comportamento não linear, mas não tanto como as presentes no VGT. Esta não linearidade é devida às não idealidades na produção da válvula. Uma pequena abertura da válvula, implica que os gases apenas podem circular por uma área mais pequena do tubo que conduz os gases. Abrindo apenas mais um pouco, relativamente a uma posição mais fechada, traduz-se num grande aumento de fluxo de ar. A forma da válvula, é feita de maneira a que pequenas variações, relativamente a uma posição fechada, se traduza numa grande alteração dos fluxos de massa. Uma pequena abertura da válvula, num estado já aberto, por exemplo de 60% para 70% da área, não causa uma grande alteração no sistema. Este último fenómeno deve-se ao fluxo de massas já estar saturado. A utilização da válvula, contribui para a criação de um turbilhão no fluxo de gases, o que introduz uma não linearidade no sistema. Estes turbilhão, é gerado quando os gases passam a secção da válvula e entram numa região de maior secção.

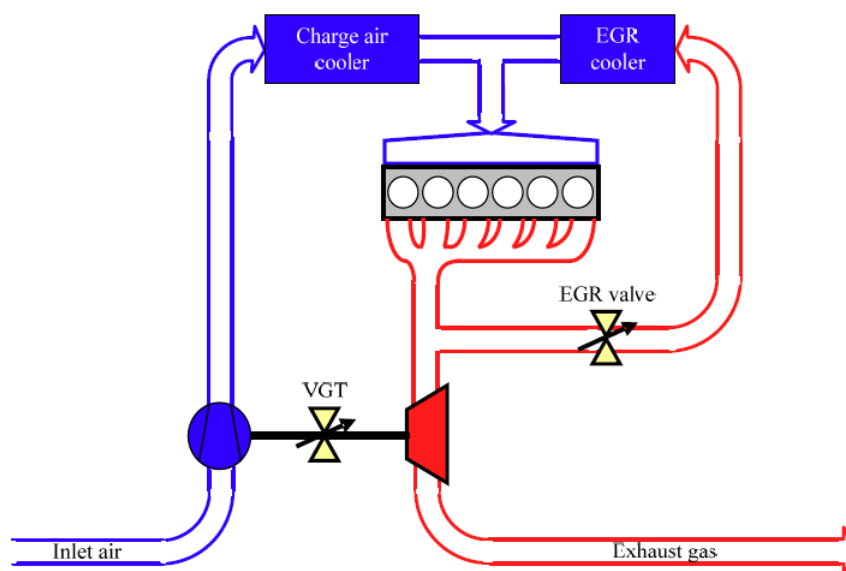


Figura 5.4: Esquema do circuito de fluxo de ar e dos gases de escape [39].

Capítulo 6

Análise ao Modelo de Johan Wahlström e Lars Eriksson

Johan Wahlström e Lars Eriksson [41], apresentam uma solução de um modelo para a simulação de um motor. Este modelo, consiste num sistema que incorpora um motor a diesel, um sistema de turbo-compressor VGT e um sistema de recirculação dos gases de escape EGR.

O desenvolvimento deste modelo, tem por ponto base o cumprimento da legislação que limita as emissões de gases dos camiões pesados. Estas limitações têm se tornado mais rígidas com o avançar dos anos. De forma a respeitar os regulamentos são utilizadas técnicas e tecnologias que permitem fazer cumprir a legislação, sendo utilizados os sistemas VGT e EGR. O principal mecanismo de redução de emissões a ser incorporado (nesta abordagem) é a redução dos óxidos de nitrogénio (NOx). A diminuição deste tipo de gases pode ser efetuada aumentando a fração de recirculação de gases de escape (x_{egr}) e diminuindo o fumo emitido, que tem por princípio o aumento do rácio de oxigénio-combustível (λ_O). O controlo das componentes x_{egr} e de λ_O , depende de uma forma complexa das diferentes válvulas do sistema de EGR e VGT. Consequentemente o controlo das válvulas dos dois sistemas tem de ser coordenado, com o fim de obedecer às emissões legisladas. Estas

propriedades do sistema são descritas em [42] e [43] onde se mostra que o sistema não apresenta fase mínima, possui *overshoots* e ocorre inversão de sinal, sendo uma boa aproximação do comportamento de um sistema real.

6.1 Sumário da Estrutura do Modelo

O modelo tem por base a Figura 6.1, este utiliza oito variáveis de estado: a pressão do coletor da admissão (p_{im}), a pressão do coletor do escape (p_{em}), a fração de massa-oxigênio no coletor de admissão (X_{Oim}), a fração de massa-oxigênio no coletor de escape (X_{Oem}), a velocidade de rotação do turbo (w_t) e três estados que descrevem a dinâmica dos sinais de controlo para os três atuadores (\tilde{u}_{egr1} , \tilde{u}_{egr2} e \tilde{u}_{vgt}).

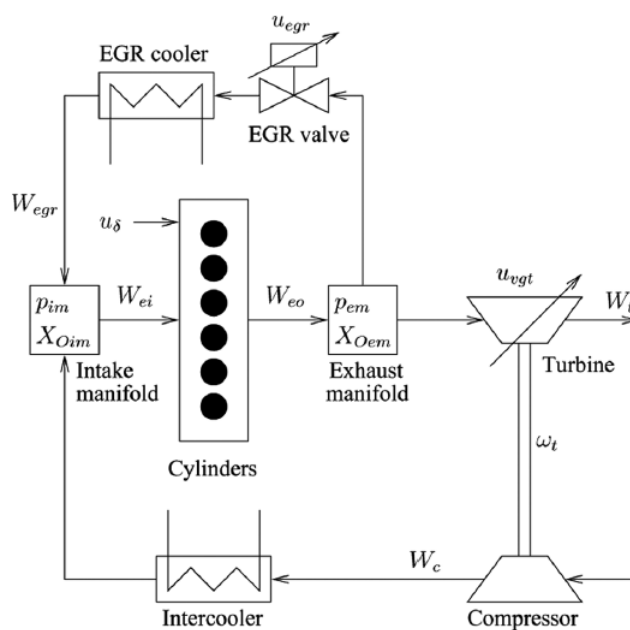


Figura 6.1: Modelo das estruturas do motor Diesel [41].

Os estados são agregados num vetor de estados, representado na equação (6.1), em que p_{im} , p_{em} e w_t descrevem a dinâmica principal do sistema. Estes elementos introduzem as propriedades já referidas: o comportamento não ser

de fase mínima, a existência de *overshoots* e a inversão de sinal.

Para modelar a dinâmica do rácio oxigénio-combustível (λ_O), são utilizados os estados X_{Oim} e X_{Oem} .

Os estados \tilde{u}_{egr1} , \tilde{u}_{egr2} e \tilde{u}_{vgt} descrevem a dinâmica dos atuadores, neste caso o modelo que atua a válvula do EGR tem dois estados, para ser possível introduzir *overshoot* no modelo do atuador da válvula.

$$\mathbf{x} = (p_{im} \quad p_{em} \quad X_{Oim} \quad X_{Oem} \quad w_t \quad \tilde{u}_{egr1} \quad \tilde{u}_{egr2} \quad \tilde{u}_{vgt})^T \quad (6.1)$$

O sistema foi desenvolvido e pensado de forma a poder ser utilizado em diferentes veículos para testar motores e calibrar os mesmos. Nestes casos o funcionamento do motor é definido pela velocidade de rotação (n_e). Para o sistema utilizar como parâmetro n_e é necessária uma nova expressão para a descrição no espaço de estados (expressão (6.2)). A equação (6.2), apresenta os parâmetros referentes aos oito estados que definem o sistema (\mathbf{x}), \mathbf{u} é o vetor de controlo descrito na equação (6.3) e n_e (velocidade de rotação) é a alteração medida.

$$\dot{\mathbf{x}} = (x, \quad u \quad n_e) \quad (6.2)$$

$$\mathbf{u} = (u_\delta \quad u_{egr} \quad u_{vgt})^T \quad (6.3)$$

Os valores presentes na equação (6.3) referem-se ao combustível injetado (u_δ), à posição da válvula EGR (definido por u_{egr} , em que $u_{egr} = 0\%$ fechado e $u_{egr} = 100\%$ aberto) e à posição do atuador do VGT (definido por u_{vgt} , em que $u_{vgt} = 0\%$ fechado e $u_{vgt} = 100\%$ aberto).

6.2 Modelação do Sistema

O modelo do sistema diesel, para ser mais simples de analisar e ser aplicado foi repartido em submodelos (Figura 6.2). Cada um dos submodelos contém as equações referentes à dinâmica do sistema. Estes submodelos encontram-se divididos em coletores (de escape e admissão), cilindros, válvula de EGR, turbo-compressor, *intercooler* e EGR-cooler.

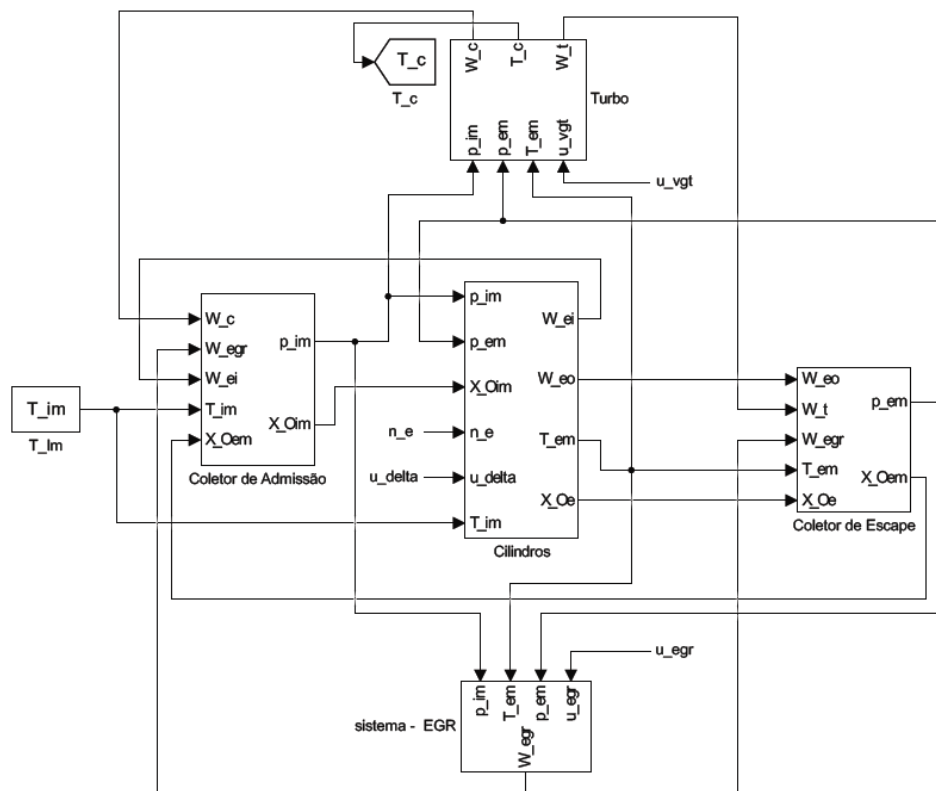


Figura 6.2: Modelo geral do sistema para aplicação em MATLAB.

6.2.1 Coletores

Inicia-se a análise pelos coletores de escape e admissão. Estes, são modelados utilizando as equações diferenciais (6.4) e (6.5). A sua formulação é efetuada através de dois conjuntos de propriedades termodinâmicas: a primeira em

que o ar apresenta uma constante de gás ideal R_a e o rácio de aumento de temperatura γ_a ; e a segunda em que os gases de escape apresentam uma constante de gás ideal R_e e também o rácio de aumento de temperatura associado γ_e . A temperatura do coletor de admissão T_{im} , é assumida como uma constante, sendo igual à temperatura no *intercooler*. As variáveis V_{im} e V_{em} são os volumes dos coletores. A temperatura no coletor de escape T_{em} é analisada na secção 6.2.2.2. Os fluxos de massa W_c , W_{egr} , W_{ei} , W_{eo} e W_t são abordados mais detalhadamente nas secções 6.2.2 a 6.2.4.

$$\frac{d}{dt}p_{im} = \frac{R_a T_{im}}{V_{im}}(W_c + W_{egr} - W_{ei}) \quad (6.4)$$

$$\frac{d}{dt}p_{em} = \frac{R_e T_{em}}{V_{em}}(W_{eo} - W_t - W_{egr}) \quad (6.5)$$

A fração de gases no coletor de admissão é dada pela equação (6.6).

É de ter em atenção, que os gases provenientes da EGR também contém oxigénio, o que causa uma influência no rácio de oxigénio-combustível no cilindro. Este efeito é modelado através das concentrações de oxigénio X_{Oim} (equação (6.7)) e X_{Oem} (equação (6.8)) presente nos volumes de controlo.

$$x_{egr} = \frac{W_{egr}}{W_c + W_{egr}} \quad (6.6)$$

$$\frac{d}{dt}X_{Oim} = \frac{R_a T_{im}}{V_{im} p_{im}}((X_{Oem} - X_{Oim})W_{egr} + (X_{Oc} - X_{Oim})W_c) \quad (6.7)$$

$$\frac{d}{dt}X_{Oem} = \frac{R_e T_{em}}{V_{em} p_{em}}(X_{Oe} - X_{Oem})W_{eo} \quad (6.8)$$

A variável X_{Oc} corresponde à concentração de oxigénio que passa pelo turbo-compressor e X_{Oe} corresponde à concentração de oxigénio nos gases de escape provenientes dos cilindros do motor.

6.2.2 Cilindros

O modelo, referente aos cilindros divide-se em três submodelos: um primeiro modelo dos fluxos de massa, um segundo referente à temperatura dos gases de escape do coletor e por fim um modelo que descreve o binário do motor.

6.2.2.1 Fluxo de massa

O modelo referente aos fluxos tem por submodelos: os fluxos de gases e combustível que entram e saem do cilindro, a razão de oxigénio e combustível e a concentração de oxigénio à saída do cilindro.

O parâmetro referente ao fluxo de massa W_{ei} , proveniente do coletor de admissão para os cilindros é modelado pela equação (6.9). Nesta, são utilizados os parâmetros de eficiência volumétrica η_{vol} [46], a pressão no coletor de admissão p_{im} , a temperatura no coletor de admissão T_{im} , a velocidade de rotação n_e , o volume deslocado V_d e o raio da entrada de ar R_a .

$$W_{ei} = \frac{\eta_{vol} p_{im} n_e V_d}{120 R_a T_{im}} \quad (6.9)$$

Para dimensionar o fluxo de massa de combustível W_f é definida através da equação (6.10). A quantidade de combustível introduzida nos cilindros, é controlada pelo parâmetro μ_δ (quantidade de combustível em miligramas por ciclo). O último parâmetro da equação corresponde ao número de cilindros n_{cyl} do motor modelado.

$$W_f = \frac{10^{-6}}{120} \mu_\delta \cdot n_e \cdot n_{cyl} \quad (6.10)$$

Para obter o fluxo de massa W_{eo} , à saída do cilindro é aplicada a equação (6.11), que descreve o balanço do fluxo de massas.

$$W_{eo} = W_f + W_{ei} \quad (6.11)$$

No cilindro o parâmetro referente à razão de oxigénio e combustível λ_O é definido em (6.12). O parâmetro (O/F) corresponde à relação estequiométrica entre o oxigénio e a massa de combustível. Durante a combustão o oxigénio é queimado na presença do combustível. Dado esse facto, nos motores Diesel para evitar a presença de fumos $\lambda_O > 1$.

$$\lambda_O = \frac{W_{ei} X_{Oim}}{W_f (O/F)_s} \quad (6.12)$$

6.2.2.2 Temperatura dos Cilindros e do Coletor de Escape

A temperatura dos cilindros T_e é modelada com base na Tese de Skogtjärn [47]. Esta aproximação é baseada no cálculo circular de gás ideal de Seliger, o qual permite obter a temperatura à saída do cilindro. As equações utilizadas para o dimensionamento são não-lineares e estão relacionadas entre si. De tal forma, foi necessário proceder ao cálculo dos parâmetros, utilizando um ponto fixo para cada iteração.

A temperatura do coletor de escape T_{em} não é totalmente descrita pela temperatura nos cilindros, isto porque existem perdas de calor. Desta forma a análise é efetuada por Eriksson na referência [48], onde descreve um modelo contento essa propriedade.

As temperaturas referidas foram dimensionadas como constantes. Aconselha-se a análise da referência [41], para uma melhor descrição destes dois parâmetros.

6.2.2.3 Binário dos Cilindros

O binário produzido pelo motor M_e (equação (6.13)) é dividido em três diferentes modelos:

- M_{ig} - Binário bruto;
- M_p - Binário de bombeamento;

- M_{fric} - Binário de fricção.

$$M_e = M_{ig} - M_p - M_{fric} \quad (6.13)$$

O parâmetro M_p , é modelado utilizando as pressões provenientes dos coletores de escape e de admissão, dado por:

$$M_p = \frac{V_d}{4\pi}(p_{em} - p_{im}) \quad (6.14)$$

O binário bruto é definido pela equação (6.15). O parâmetro η_{ig} indica a eficiência máxima do binário bruto e q_{HV} a constante de aquecimento do combustível.

$$M_{ig} = \frac{u_\delta \cdot 10^{-6} \cdot n_{cyl} \cdot q_{HV} \cdot \eta_{ig}}{4\pi} \quad (6.15)$$

O parâmetro M_{fric} segue uma função polinomial (equação (6.16)) em que o rácio do número de rotações do motor n_{eratio} é dado pela equação (6.17). Os parâmetros c_{fric1} , c_{fric2} e c_{fric3} são os coeficientes da função polinomial de fricção e são obtidos experimentalmente.

$$M_{fric} = \frac{V_d}{4\pi} 10^5 (c_{fric1} n_{eratio}^2 + c_{fric2} n_{eratio}^2 + c_{fric3}) \quad (6.16)$$

$$n_{eratio} = \frac{n_e}{1000} \quad (6.17)$$

6.2.3 Válvula-EGR

O modelo da válvula-EGR tem por constituição os submodelos do fluxo de massa da válvula-EGR e o atuador da válvula-EGR.

6.2.3.1 Válvula-EGR Fluxo de Massa

O fluxo de massas pela válvula de EGR é modelado, utilizando uma simplificação de uma restrição de fluxo compressível e de área variável. Assumiu-se também que não existe um fluxo reverso, em que $p_{em} < p_{im}$. Com a aplicação desta restrição é possível construir um modelo mais simples. Num modelo real, o fluxo reverso é sempre evitado durante um caso de operação normal do motor, fechando a válvula de EGR.

O fluxo de massa W_{egr} é então descrito da seguinte forma:

$$W_{egr} = \frac{A_{egr} p_{em} \Psi_{egr}}{\sqrt{T_{em} R_e}} \quad (6.18)$$

no qual

$$\Psi_{egr} = 1 - \left(\frac{1 - \Pi_{egr}}{1 - \Pi_{egropt}} - 1 \right)^2 \quad (6.19)$$

A razão da pressão Π_{egr} sobre a válvula é definida por:

$$\Pi_{egr} = \begin{cases} \Pi_{egropt} & \text{se } p_{im}/p_{em} < \Pi_{egropt} \\ p_{im}/p_{em} & \text{se } \Pi_{egropt} \leq p_{im}/p_{em} \leq 1 \\ 1 & \text{se } 1 < p_{im}/p_{em} \end{cases} \quad (6.20)$$

O parâmetro Π_{egropt} corresponde ao valor ótimo de Π_{egr} para o valor máximo da equação (6.20).

A área efetiva¹, é definida pela equação (6.21), esta é modelada utilizando uma função polinomial, em função da posição da válvula de EGR \tilde{u}_{egr} (a função polinomial é definida em (6.22)).

$$A_{egr} = A_{egrmax} f_{egr}(\tilde{u}_{egr}) \quad (6.21)$$

¹Corresponde ao valor líquido da área de ar que pode passar pela abertura da válvula de EGR.

$$f_{egr}(\tilde{u}_{egr}) = \begin{cases} c_{egr1}\tilde{u}_{egr}^2 + c_{egr2}\tilde{u}_{egr} + c_{egr3} & \text{se } \tilde{u}_{egr} \leq -c_{egr2}/2c_{egr1} \\ c_{egr3} - c_{egr2}^2/4c_{egr1} & \text{se } \tilde{u}_{egr} > -c_{egr2}/2c_{egr1} \end{cases} \quad (6.22)$$

Os parâmetros c_{egr1} , c_{egr2} e c_{egr3} são os coeficientes da função polinomial (6.22). O parâmetro \tilde{u}_{egr} descreve a dinâmica da válvula (o seu dimensionamento é explicado na subsecção seguinte 6.2.3.2).

6.2.3.2 Atuador da Válvula de EGR

Para modelar a dinâmica do atuador da válvula, é utilizado um sistema de segunda ordem com um *overshoot* e um atraso. Esta modelação é efetuada, tendo por base os dados representados na Figura 6.3, a qual representa a resposta do sistema real e dos dados modelados, para uma entrada em degrau no sistema. Este modelo consiste na subtração entre dois sistemas de primeira ordem, com ganhos e constantes de tempo diferentes, de acordo com as equações (6.23), (6.24) e (6.25). As constantes de tempo $\tau_{egr1,2}$, o atraso τ_{degr} e o ganho K_{egr} que afeta o *overshoot* são parametrizados através de dados experimentais.

$$\tilde{u}_{egr} = K_{egr}\tilde{u}_{egr1} - (K_{egr} - 1)\tilde{u}_{egr2} \quad (6.23)$$

$$\frac{d}{dt}\tilde{u}_{egr1} = \frac{1}{\tau_{egr1}}[u_{egr}(t - \tau_{degr}) - \tilde{u}_{egr1}] \quad (6.24)$$

$$\frac{d}{dt}\tilde{u}_{egr2} = \frac{1}{\tau_{egr2}}[u_{egr}(t - \tau_{degr}) - \tilde{u}_{egr2}] \quad (6.25)$$

6.2.4 Turbo-Compressor

O turbo-compressor é composto por um modelo da inércia associada ao turbo, um modelo da turbina, o modelo do atuador VGT e um modelo

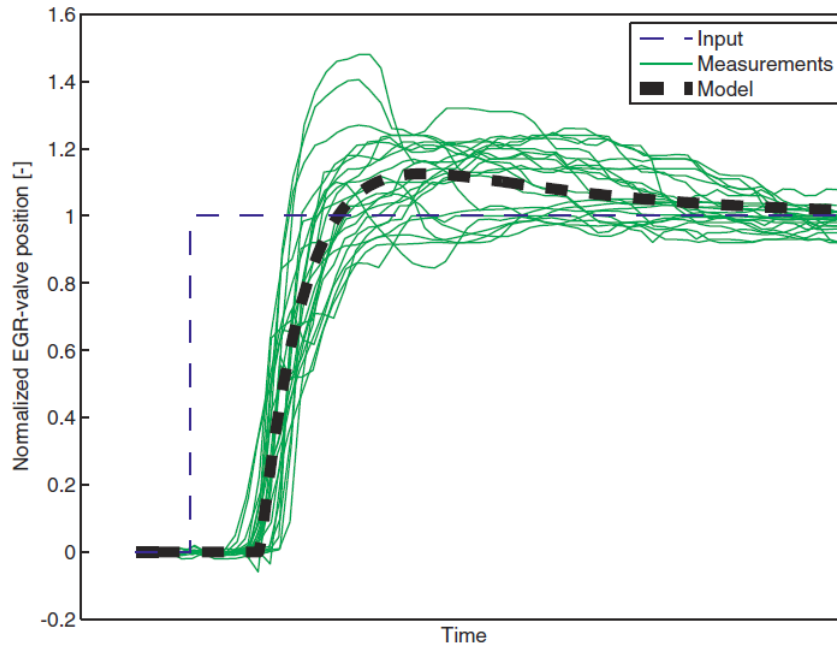


Figura 6.3: Comparação entre a dinâmica do atuador simulado e os dados obtidos nos diferentes testes [41].

do compressor.

6.2.4.1 Inércia do Turbo

A velocidade de rotação do turbo W_t , é dada pela *segunda lei de Newton* (equação (6.26)). Na qual o parâmetro J_t corresponde à inércia, P_t é a potência fornecida pela turbina, P_c é a potência necessária para acionar o compressor e η_m é a eficiência mecânica do turbo-compressor.

$$\frac{d}{dt}w_t = \frac{P_t\eta_m - P_c}{J_t w_t} \quad (6.26)$$

6.2.4.2 Turbina

A turbina é constituída pelos submodelos da eficiência, do fluxo de massas e do atuador VGT.

Eficiência da turbina: Um método de modelar a potência P_t é utilizando a eficiência da turbina η_t . Esta é definida com base na equação (6.27), onde T_t é a temperatura à saída da turbina e Π_t é a razão entre a pressão ambiente p_{amb} e a pressão do coletor de escape p_{em} (equação (6.28)). O parâmetro $P_{t,s}$ é a potência do processo isentrópico².

$$\eta_t = \frac{P_t}{P_{t,s}} = \frac{T_{em} - T_t}{T_{em}(1 - \Pi_t^{1-1/\gamma_e})} \quad (6.27)$$

$$\Pi = \frac{p_{amb}}{p_{em}} \quad (6.28)$$

Segundo o autor, é muito complexo efetuar a modelação do sistema recorrendo às equações anteriores [41], desta forma o parâmetro de eficiência utilizado n_{tm} passa a ser uma aproximação do anterior (equação (6.29), assumindo que $P_t \approx P_c$, P_c - potência no compressor). Este pode ser definido tendo em conta a velocidade de rotação das pás (BSR) e o parâmetro parabólico c_m [41].

$$\eta_{tm} = \eta_{tm,max} - c_m(BSR - BSR_{opt})^2 \quad (6.29)$$

Fluxo de Massa da Turbina: O fluxo de massa W_t é dado pela expressão:

$$W_t = \frac{A_{vgtmax} p_{em} f_{\Pi t}(\Pi_t) f_{vgt}(\tilde{u}_{vgt})}{\sqrt{T_{em} R_e}} \quad (6.30)$$

em que A_{vgtmax} é a área máxima da turbina por onde os gases podem passar. A análise feita pelo autor [41], mostra que o fluxo de massa, depende da razão da pressão Π_t e o sinal utilizado no atuador VGT \tilde{u}_{vgt} . Quando o rácio de pressão (Π_t) diminui, o fluxo de massa de ajuste deve ser incrementando até atingir uma velocidade elevada, levando ao estrangulamento do fluxo de massa. Este processo é descrito pela função:

²É um processo em que a entropia do sistema permanece constante [49].

$$f_{\Pi_t} = \sqrt{1 - \Pi_t^{K_t}} \quad (6.31)$$

A função $f_{vgt}(\tilde{u}_{vgt})$ que descreve a área efetiva do fluxo de massas é dada por:

$$f_{vgt}(\tilde{u}_{vgt}) = c_{f2} + c_{f1} \sqrt{\max(0, 1 - (\frac{\tilde{u}_{vgt} - c_{vgt2}}{c_{vgt1}})^2)} \quad (6.32)$$

O parâmetro K_t presente na exponencial, e os coeficientes c_{f1} , c_{f2} , c_{vf1} e c_{vf2} são obtidos experimentalmente.

Atuador VGT A dinâmica, associada ao atuador VGT é modelada através de um sistema de primeira ordem com um atraso de acordo com a seguinte equação:

$$\frac{d}{dt} \tilde{u}_{vgt} = \frac{1}{\tau_{vgt}} [u_{vgt}(t - \tau_{dvgt}) - \tilde{u}_{vgt}] \quad (6.33)$$

A constante de tempo τ_{vgt} e o atraso τ_{dvgt} são obtidos experimentalmente.

6.2.4.3 Compressor

O compressor é dividido em dois submodelos:

- Eficiência do compressor;
- Fluxo de massas no compressor.

Eficiência do Compressor: A potência do compressor P_c é definida utilizando o parâmetro de eficiência η_c , definido como:

$$\eta_c = \frac{P_{c,s}}{P_c} = \frac{T_{amb}(\Pi_c^{1-1/\gamma_a} - 1)}{T_c - T_{amb}} \quad (6.34)$$

O parâmetro T_c é a temperatura ambiente à saída do compressor, T_{amb} é a temperatura ambiente, γ_a é a capacidade de aquecimento do ar e Π_c é a razão entre pressões dada por:

$$\Pi_c = \frac{p_{im}}{p_{amb}} \quad (6.35)$$

O parâmetro $P_{c,s}$ é a potência do processo isentrópico dado por:

$$P_{c,s} = W_c c_{pa} T_{amb} (\Pi_c^{1-1/\gamma_a} - 1) \quad (6.36)$$

em que W_c é o fluxo de massa no compressor e c_{pa} é uma constante de aquecimento.

A potência do compressor, é obtida resolvendo a equação (6.34) em ordem a P_c e utilizando a equação (6.36). A equação (6.37) exemplifica o processo.

$$P_c = \frac{P_{c,s}}{\eta_c} = \frac{W_c c_{pa} T_{amb}}{\eta_c} (\Pi_c^{1-1/\gamma_a} - 1) \quad (6.37)$$

Fluxo de Massa do Compressor: O fluxo de massa no compressor W_c (equação (6.38)) é obtido utilizando duas variáveis adimensionais, o coeficiente de transferência de energia Ψ_c e o coeficiente volumétrico do fluxo Φ_c [50].

$$W_c = \frac{P_{amb} \pi R_c^3 \omega_t}{R_a T_{amb}} \Phi_c \quad (6.38)$$

6.2.5 Intercooler e EGR Cooler

De forma a simplificar a aplicação do modelo tanto o *intercooler* como o *EGR cooler* foram assumidos como ideais. Isto significa que não existem perdas de pressão, acumulação de massas e a eficiência é perfeita.

6.3 Exemplos de Controle

O modelo do motor Diesel descrito foi utilizado nas referências [51] e [52] para aplicações de diferentes técnicas de controle. A este sistema foram aplicados diferentes tipos de controladores, com o objetivo de diminuir as emissões, e manter a eficiência do motor. Os controladores foram desenvolvidos com diferentes parâmetros de controle. A Tabela 6.1 contém todos os resultados obtidos para os diferentes tipos de controladores, durante o teste European Transient Cycle (ETC). Tipos de controle efetuados:

1. Modelo de Controle Preditivo (MPC), $\text{MPC}(x_{egr}, pmep)$: este controlador usa como parâmetros de saída a fração de gases circulados (x_{egr}) e as perdas de pressão ($pmep = p_{em} - p_{im}$).
2. $\text{MPC}(W_{comp}, p_{im})$: este controle utiliza como parâmetros de saída o fluxo de massa no turbo (W_{comp}) e a pressão no coletor de admissão (p_{im}).
3. Proporcional-Integral-Derivativo (PID): este controle, utiliza como principais parâmetros de saída o rácio de oxigénio-combustível (λ_0) e a fração de gases recirculados (x_{egr}).

Tabela 6.1: Dados das medições dos três controladores analisados durante o ciclo de ETC.

Controlador	E_{λ_0}	$E_{x_{egr}}$	PMEP
$\text{MPC}(W_{comp}, p_{im})$	1.00	1.00	1.00
$\text{MPC}(x_{egr}, pmep)$	0.91	0.20	0.88
PID	0.86	0.32	1.00

Capítulo 7

Desenvolvimento

Ao longo deste capítulo são apresentados algumas propostas de controlo. Como foco, espera-se alcançar um sistema para controlo de emissões em motores Diesel. O processo a controlar, tem por base um motor Diesel com VGT e EGR, para veículos pesados.

Como proposta inicial, é explorado o controlo PID com realimentação. A aplicação deste método, tem como intuito o estudo de um tipo de controlo que já foi utilizado no modelo do motor.

O segundo método de controlo a aplicar, tem por fim a aplicação de redes neuronais no controlo do mesmo modelo, o motor Diesel. A RNA deve ser treinada para emular a dinâmica não linear do processo, mediante a apresentação de padrões de entrada/saída gerados pelo processo.

Este estudo envolve também, uma comparação da diferença de resultados obtidos entre os diferentes sistemas de controlo, tendo por análise o controlo PID e o controlo com RNAs. Desta forma, pretende-se analisar em que medida a inclusão das RNAs pode melhorar o sistema do motor a Diesel.

Por fim, serão apresentados alguns métodos, que ajudam a avaliar o desempenho de cada um dos controlos de controlo desenvolvidos.

7.1 Processo a Controlar

Como processo de controlo, vai ser utilizado o modelo do motor Diesel desenvolvido por Johan Wahlström e Lars Eriksson, descrito no capítulo 6. Os parâmetros de controlo a utilizar, são os analisados nas referências [53, 54]. O sistema geral, segue o diagrama da Figura 7.1: um bloco onde são mantidos os dados de gestão do motor e performance, em que este estabelece os padrões de controlo a seguir. Estes valores, são aplicados ao motor. As respostas são comparadas às referências. Caso as respostas dos parâmetros de saída (λ_0 e x_{egr}) não sejam iguais aos valores de referência (λ_0^s e x_{egr}^s), ajustam-se as posições das válvulas de EGR (u_{egr}) e do VGT (u_{vgt}) para alcançar essa igualdade.

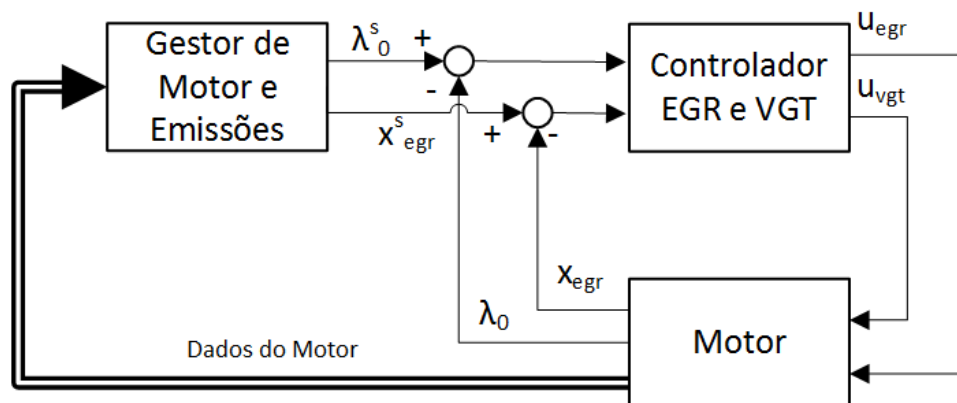


Figura 7.1: Modelo em cascata da estrutura de controlo.

Perante o diagrama da Figura 7.1, o sistema de controlo deve ser capaz de cumprir os seguintes objetivos:

1. O parâmetro λ_0 , deve ser capaz de seguir o *setpoint* λ_0^s (este permite controlar a relação de emissões, consumo de combustível e tempo de resposta);
2. O parâmetro x_{egr} , deve seguir o seu *setpoint* x_{egr}^s (caso o valor de x_{egr} seja muito baixo, será emitido mais NOx, caso o x_{egr} seja muito

elevado, será emitido mais fumo);

3. A velocidade de rotação do turbo-compressor n_t , não deve exceder um limite máximo, para este não ser danificado.

7.1.1 Vantagens dos Parâmetros Escolhidos

Existem três vantagens para a escolha dos parâmetros x_{egr} (fração de gases de escape recirculados) e λ_0 (razão de oxigénio/combustível), para serem utilizados como variáveis de desempenho do sistema.

A primeira vantagem, é que estas variáveis fornecem informação direta sobre a possibilidade de minimizar o bombeamento de combustível, em comparação com outras, por exemplo as variáveis p_{em} (pressão no coletor de escape), p_{im} (pressão no coletor de admissão) e W_c (fluxo de massa no compressor).

A segunda vantagem, está na forte ligação dos dois parâmetros às emissões. Estes parâmetros conseguem impor uma separação natural do sistema de gestão do motor.

A terceira vantagem, tem por fundamento a segunda vantagem. No que toca à produção dos sistemas de controlo do motor a nível industrial, numa fase inicial o motor pode ser configurado para atingir a melhor performance possível e posteriormente ser ajustado o controlo de emissões.

7.2 Controlo PID

O controlo PID (proporcional, integral e derivativo) é a forma mais comum de controlo por realimentação [2]. Devido à sua simplicidade e excelente desempenho em muitas aplicações, estes controladores são usados em 95% dos processos industriais de malha-fechada. Tal utilização deve-se ao facto de ser facilmente implementável, de baixo custo, versátil e capaz de alterar os comportamentos transitórios e de regime permanente dos processos sob

controle. Atualmente, a maioria dos processos automatizados que utilizam PLCs (Programmable Logic Controller), possuem malhas de controle de algoritmos PID, sendo responsabilidade dos engenheiros/técnicos a tarefa de sintonia dos parâmetros dos controladores. A sua adaptação às novas mudanças na tecnologia permitiu o aumento de recursos adicionais tais como o ajuste automático, a programação dos ganhos e a adaptação contínua.

O método de controle PID representado na Figura 7.2, efetua o cálculo do erro entre o valor à saída do processo e o valor de referência. O controlador tenta minimizar o erro ajustando as entradas que controlam o processo, de forma a minimizar o erro da relação entrada e saída. O cálculo da ação do

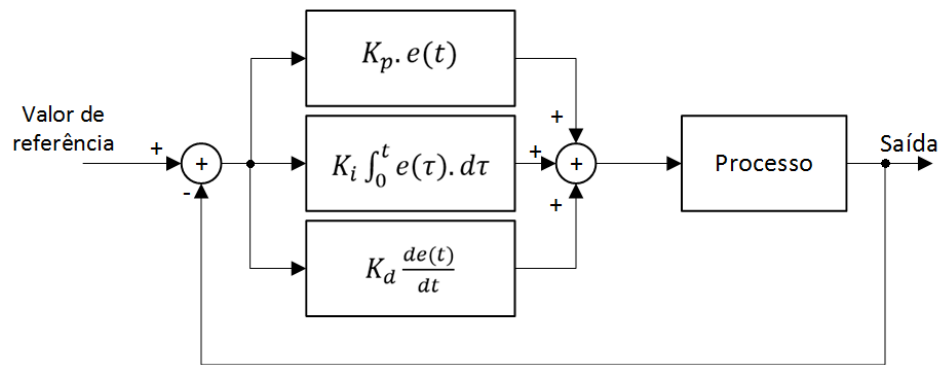


Figura 7.2: Esquema de controle PID em realimentação.

controlador, envolve três tipos diferentes de ações: proporcional (P), integral (I) e derivativo (D). A junção destas três componentes permite calcular a ação de controle a ser tomada pelo PID. A saída de controle é dada pela equação seguinte:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) \cdot d\tau + K_d \frac{de(t)}{dt} \quad (7.1)$$

em que os ganhos integral (K_i) e derivativo (K_d) são dados pelas seguintes equações:

$$K_i = K_p \frac{1}{T_i} \quad (7.2)$$

$$K_d = K_p \cdot T_d \quad (7.3)$$

Cada um dos ganhos, tem uma importância diferente na resposta do processo a controlar. Dependendo do valor de cada um, a sua utilização tanto pode melhorar como piorar a resposta do sistema. Tendo em conta essa realidade, é necessário utilizar o método mais apropriado para fazer a sintonia dos parâmetros. Para um processo mais simples, é possível utilizar um controlador sintonizado através de *tentativa-erro*. Para tipos de processo mais complexos e com necessidade de uma melhor resposta, aplicam-se métodos de sintonia, como exemplos utilizam-se os métodos de Ziegler-Nichols, Coon-Cohen e Zhuang-Atherton [1].

7.2.1 Estrutura dos Controladores PID

O esquema de controlo atua os parâmetros u_{egr} e u_{vgt} de forma a atingir os objetivos de controlo do processo (referidos na secção 7.1). O processo de controlo, aqui descrito, é baseado no sistema desenvolvido na referência [53]. Os dados para formulação do controlo são fornecidos na forma de equações matemáticas e são desenvolvidas neste trabalho, com recurso ao *software* MATLAB/Simulink.

A estrutura de controlo, é desenvolvida tendo por base seletores *min/max* e controladores SISO (*Single Input Single Output*) para o VGT e a EGR. Sendo que, os controladores SISO são do tipo PID.

Na Figura 7.3, encontra-se representado o esquema para o controlo das emissões do sistema (o respetivo modelo aplicado em MATLAB, encontra-se descrito no Anexo A.1. Modelo Simulink do esquema geral de controlo PID).

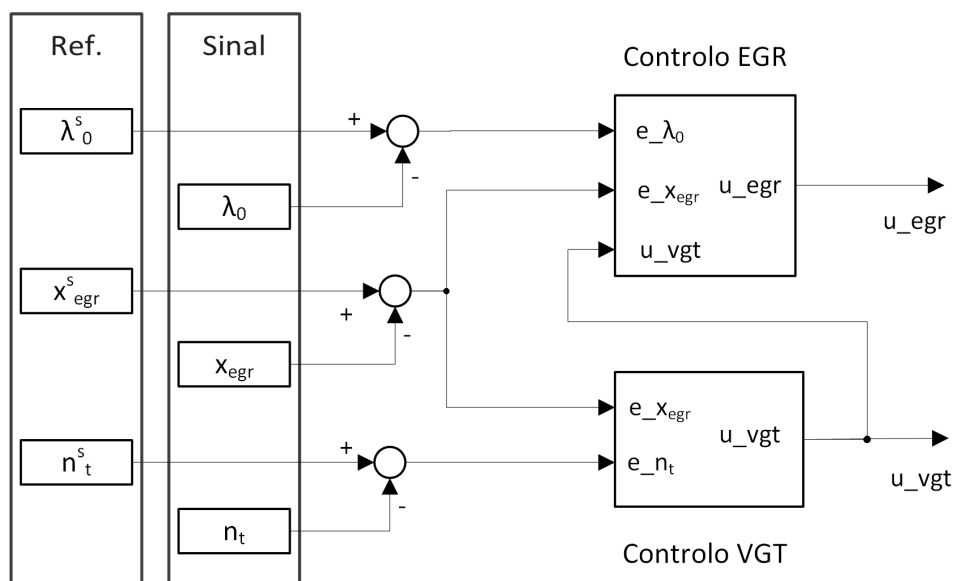


Figura 7.3: Diagrama de blocos da estrutura de controlo do processo.

7.2.2 Controladores Primários

Os controladores com maior influência no controlo do processo, são apresentados nas equações seguintes (modelo em Simulink de u_{egr} e u_{vgt} Anexo A.1.1. Modelo Simulink do bloco controladores PID):

$$u_{egr} = -PID_1(e_{\lambda_0}) \quad (7.4)$$

$$u_{vgt} = -PID_2(e_{x_{egr}}) \quad (7.5)$$

onde $e_{\lambda_0} = \lambda_0^s - \lambda_0$ e $e_{x_{egr}} = x_{egr}^s - x_{egr}$. Estes dois controladores, são responsáveis pelo controlo dos objetivos 1 e 2, referidos na secção 7.1.

A escolha destes tipos de controlo é justificada pelos seguintes fatos:

- O canal de ligação entre o u_{vgt} ao λ_0 , apresenta uma inversão de sinal e um comportamento de fase não mínima. Minimiza-se este ponto através da utilização do u_{egr} para o controlo do parâmetro λ_0 ;

- A segunda razão, para a escolha destes dois controlos, reside na presença de suboscilações no parâmetro λ_0 , caso os parâmetros nos controladores sejam invertidos em relação aos apresentados (o u_{vgt} a controlar o λ_0 e o u_{egr} a controlar o x_{egr}).

7.2.3 Controladores Adicionais

De forma, a atingir os objetivos de controlo 2 e 3, há a necessidade de adicionar mais dois controladores. Estes controladores são aplicados através das seguintes expressões:

$$u_{egr}(t_i) = \begin{cases} \min(-PID_1(e_{\lambda_0}), PID_2(e_{x_{egr}})) & , se(u_{vgt}(t_i - 1) = 100) \\ -PID_1(e_{\lambda_0}) & , senão \end{cases} \quad (7.6)$$

$$u_{vgt}(t_i) = \begin{cases} 100 & , se((u_{vgt}(t_i - 1) = 100) \& \\ & (e_{x_{egr}} < 0.01)) \\ \max(-PID_3(e_{x_{egr}}), -PID_4(e_{n_t})) & , senão \end{cases} \quad (7.7)$$

sendo que $e_{nt} = n_t^s - n_t$.

A adição destes dois modos de controlo, são motivados pelos pontos seguintes:

- **Para alcançar o objetivo 2** - há a necessidade de diminuir o parâmetro x_{egr} , utilizando para tal o controlador $PID_2(e_{x_{egr}})$, em (7.6);
- **Para alcançar o objetivo 3** - como o VGT é influenciado pela rotação da turbina n_t , para esta não sofrer danos, o controlo em (7.7) limita a velocidade máxima de rotação.

O controlador PID_1 , PID_2 e PID_3 , só utilizam as componentes proporcional e integral (PI). O controlador PID_4 é o único que apresenta controlo

proporcional, integral e derivativo (PID) (modelo Simulink: Anexo A.1.1.1. Modelo Simulink do bloco Uegrsub e A.1.1.2. Modelo Simulink do bloco Uvgtsub).

7.2.4 Método Automático de Sintonia

Na estrutura de controlo descrita, estão presentes quatro controladores do tipo PID (3 PI e 1 PID), que necessitam ser sintonizados. O método de controlo é baseado na referência [53], em que os objetivos são formulados através de um método de minimização dos mínimos quadrados, dado por:

$$\min[V(\theta)], \theta > 0 \quad (7.8)$$

em que θ é o parâmetro que contém os ganhos dos controladores, descrito por:

$$\theta = [K_1, K_{i1}, K_2, K_{i2}, K_3, K_{i3}, K_4, K_{i4}, K_{d4}]^T \quad (7.9)$$

Os valores dos ganhos presentes em θ , são ajustados pela minimização da função $V(\theta)$, dada por:

$$\begin{aligned} V(\theta) = \sum_{i=1}^N \gamma_{\lambda_0} \left(\frac{e_{\lambda_0}(t_i, \theta)}{\lambda_{0Norma}} \right)^2 + \gamma_{egr} \left(\frac{e_{xegr}(t_i, \theta)}{x_{egrNorma}} \right)^2 \\ + \left(\frac{u_{egr}(t_i, \theta) - u_{egr}(t_i - 1, \theta)}{u_{egrNorma}} \right)^2 \\ + \left(\frac{u_{vgt}(t_i, \theta) - u_{vgt}(t_i - 1, \theta)}{u_{vgtNorma}} \right)^2 \\ + \gamma_{nt} \left(\frac{\max(n_t(t_i) - n_t^{max}, 0)}{n_tNorma} \right)^2 \end{aligned} \quad (7.10)$$

em que o valor t_i é o tempo da amostra i . Os diferentes termos, utilizados na função, justificam-se nos seguintes pontos:

- **1º Termo** - O primeiro termo, é utilizado para minimizar o erro entre o λ_0^s e o λ_0 , em que $e_{\lambda_0} = \lambda_0^s - \lambda_0$;

- **2º Termo** - O segundo termo, é utilizado para minimizar o erro entre o x_{egr}^s e o x_{egr} , em que $e_{xegr} = x_{egr}^s - x_{egr}$;
- **3º e 4º Termo** - Estes termos, servem para evitar oscilações nos sinais de controlo u_{vgt} e u_{egr} ;
- **5º Termo** - Evita uma velocidade excessiva do turbo, para que este não seja danificado.

Otimização

A otimização proposta em [53], consiste em três fases: inicialização do método, aplicação de uma heurística global e obtenção de um mínimo local. Os resultados foram obtidos utilizando o método de Aström-Hägglund [55]. A este método, foi aplicada uma heurística para determinar os melhores valores numa região mais abrangente. Desta forma, evita-se a obtenção de um mínimo local indesejável e diminui-se o tempo de computação.

1ª Fase: Encontrar uma estimativa inicial.

1. Inicialização: método de Aström-Hägglund.

2ª Fase: Encontrar uma solução próxima de um bom mínimo local.

1. Para todos $n = 1$ até 9:
 - Multiplicar $\theta(n)$ por 3, calcular $V(\theta)$ e guardar o valor deste e do respetivo θ ;
 - Dividir $\theta(n)$ por 3, calcular $V(\theta)$ e guardar o valor deste e do respetivo θ .
2. Escolher o conjunto de parâmetros θ que correspondem ao menor valor de $V(\theta)$ calculado no passo anterior.
3. Voltar a 1., até encontrar um valor menor para $V(\theta)$.

3ª Fase: Encontrar uma solução para o mínimo local.

1. Suposição inicial: Solução da 2ª fase.
2. Utilizar um método *standard* para minimizar os mínimos quadrados.

O método de sintonia descrito (para os controladores) foi aplicado, utilizando o ETC como referência. Visto que, o ciclo de testes (ETC) é muito demorado (cerca de 30 horas), este foi reduzido e dividido em diferentes fases transitórias: meio urbano, rural e em vias rápidas (sendo que, em cada uma delas foram selecionadas as zonas mais irregulares). Com esta alteração, o processo de obtenção de resultados passou de 30 horas para 3 horas. Os resultados que foram obtidos para este método são os apresentados na Tabela 7.1 (dados na referência [53]).

Como forma de melhoria, aos parâmetros dos controladores aplica-se em MATLAB a função *fminsearch* [56]. Esta tem como pontos iniciais de sintonia, os valores da Tabela 7.1. Desta forma efetua-se uma sintonia mais fina para o controlo a aplicar.

Tabela 7.1: Valores dos ganhos obtidos em [53], através da aplicação do método de otimização.

	$\gamma_{\lambda_0} = 3/2$ e $\gamma_{xegr} = 1/2$
K1	100
Ki1	180
K2	450
Ki2	900
K3	100
Ki3	420
K4	316×10^{-6}
Ki4	310×10^{-6}
Kd4	140×10^{-6}

7.3 Controlador com Estabilizador Fixo

Como método de controlo baseado em RNAs, propõe-se a aplicação do controlador com estabilizador fixo. Na secção 3.1, é feita uma análise a este tipo

de controlo. A justificação para a aplicação deste método de controlo, reside em dois motivos:

1. A RNA, pode não ser capaz de controlar o sistema devido a um treino desadequado ou falta deste, nesta situação, o controlador estabilizador, tenta manter estabilidade da resposta do processo;
2. A utilização de dois controladores, dá a possibilidade de dividir tarefas de controlo, melhorando o desempenho do processo.

O primeiro controlador é o responsável pelo controlo da saída do processo, para garantir que seja seguida a entrada de referência, sendo utilizada a RNA, para alcançar este objetivo.

O segundo controlador (o estabilizador), tem como finalidade manter a estabilidade do processo. Este tem como objetivo, reagir aos distúrbios que deterioreem os sinais do sistema e que podem posteriormente influenciar a saída do processo. Para esta situação é utilizado o controlo PID, para garantir o cumprimento da tarefa.

Para aplicar o sistema de controlo, vai ser utilizado o controlo PID (desenvolvido na secção 7.2) como estabilizador. Fica a faltar o controlo neuronal, para tal é necessário desenvolver uma RNA, para recriar o mapeamento inverso da saída-entrada do processo. Desta forma é gerado o modelo inverso do motor modelado. O processo de treino da rede neuronal para implementar o controlador com estabilizador fixo, deve seguir o diagrama da Figura 7.4.

Como passo inicial, é definido o tempo de simulação para obter as amostras para o treino da rede. Dada a instabilidade do processo a controlar e a existência de inúmeros pontos de funcionamento, a modelação do processo é limitada. Estes limites, impõem a captura de diferentes tipos de transições da abertura das válvulas do VGT e EGR (u_{vgt} e u_{egr} , respetivamente), em diferentes pontos de rotação do motor. As velocidades, para a obtenção des-

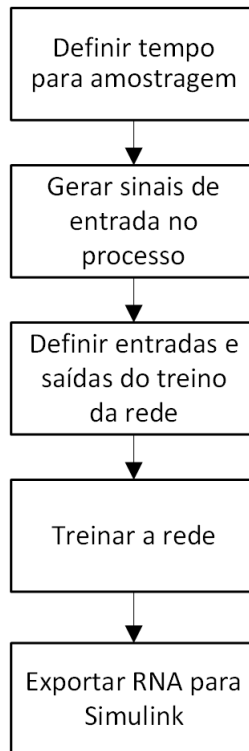


Figura 7.4: Diagrama de treino da RNA para obtenção do modelo inverso do processo.

tes valores restringe-se entre 1000 rpm e 1700 rpm. A escolha deste intervalo de velocidades de rotação, tem como justificação o limite mínimo para uma zona de rotação do veículo em movimento (1000 rpm) e o limite máximo de rotação do motor modelado (1700 rpm). O combustível requerido em cada ponto de velocidade é obtido pela equação seguinte:

$$u_{delta} = \frac{X_{Oim} \cdot W_{ei} \cdot 120}{\lambda_0 \cdot 10^{-6} \cdot n_e \cdot n_{cyl}} \quad (7.11)$$

em que, n_{cyl} é o número de cilindros do motor Diesel (seis), X_{Oim} a concentração de oxigénio e W_{ei} o fluxo de massa proveniente do coletor de admissão.

Os sinais atribuídos aos parâmetros u_{vgt} e u_{egr} , para obtenção dos pontos de funcionamento, são ondas triangulares com um período de 10 segundos. As amplitudes máximas e mínimas, são respetivamente [20;100] e [0;100]

(percentagem da abertura das válvulas). Os valores não são aleatórios, para ser possível obter as respostas das saídas, em diferentes intervalos de velocidade constantes. De tal forma, a velocidade de rotação do motor n_e , é definida no intervalo de 1000 rpm a 1700 rpm, com degraus de 100 rpm positivos de 10 em 10 segundos. Alcançado a velocidade máxima, aplica-se o mesmo processo na forma decrescente. Diminuindo 100 rpm de 10 em 10 segundos. A simulação requer 150 segundos, para obter todas as respostas necessárias para o treino da rede (Figura 7.5).

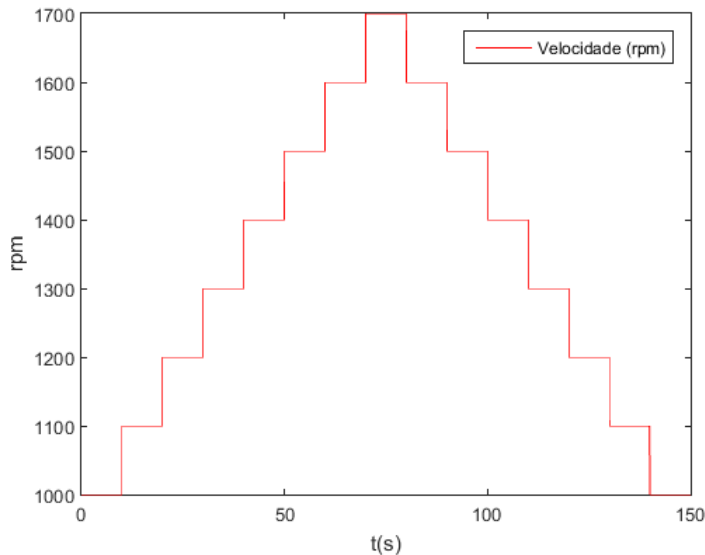


Figura 7.5: Gráfico referente à forma de onda de treino da velocidade de rotação do motor.

Depois de se ter os sinais para modelação do sistema, é necessário definir as entradas e saídas da rede para efetuar o treino da RNA. As entradas são compostas pelos valores de saída do processo λ_0 , x_{egr} e n_t , bem como o estado de rotação do motor (n_e) e o combustível (u_{delta}). Como saídas da RNA, aplicam-se os dois sinais de controlo do sistema: o u_{vgt} e o u_{egr} . Esta configuração permite obter a dinâmica inversa do processo.

Com as entradas e saídas configuradas, define-se a estrutura da RNA. O tipo de rede escolhido, tem por conceito o ajuste das entradas em relação às

saídas, através do conjunto de valores obtidos do processo. Este tipo de RNA, é usualmente aplicada na estimação de emissões. É uma RNA *feed-forward* (Figura 7.6), que tem na sua estrutura base duas camadas: uma camada oculta e uma camada de saída. Na camada oculta, a função de ativação usada é do tipo sigmóide, esta camada é constituída por 25 neurónios. Foram testados diferentes valores relativamente ao números de neurónios, sendo o valor referido, a apresentar o melhor desempenho no controlo do processo. A camada de saída, é constituída por dois neurónios. A função de ativação aplicada à camada de saída é do tipo linear.

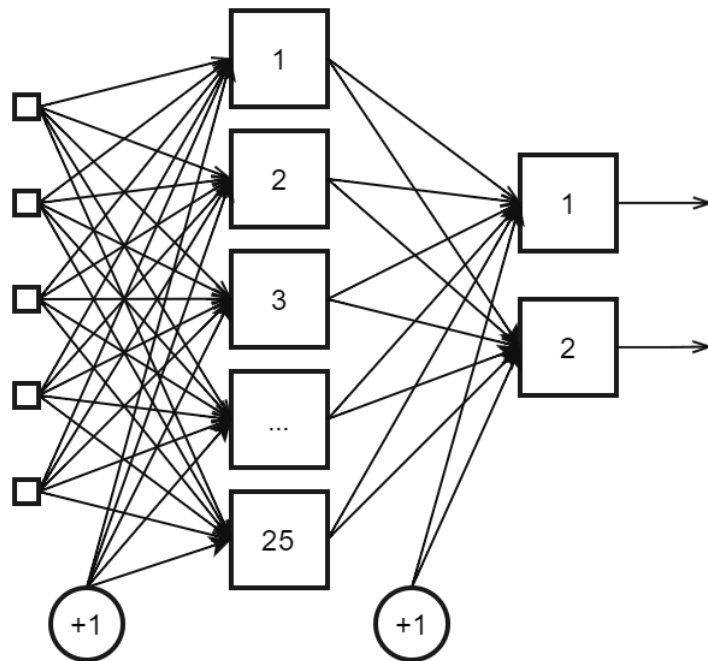


Figura 7.6: Diagrama do modelo da RNA do controlo com estabilizador fixo.

O passo seguinte, é configurar os parâmetros de treino da rede e inicialização da mesma. É utilizado como algoritmo de treino (definido com o comando `fitnet(neurónios ocultos, 'função de treino')`), o método de retropropagação, com regularização Bayesiana (`trainbr`). O recurso a este tipo de algoritmo, é vantajoso dada a sua capacidade de obter uma melhor generalização, para conjuntos de dados complexos, pequenos e com muito ruído.

Como desvantagem, destaca-se o tempo de treino ser maior relativamente a outros métodos, por exemplo o de Levenberg-Marquardt. O fim do treino é atingido pela minimização da adaptação dos pesos (a regularização). A função de desempenho utilizada pela RNA é a minimização do erro quadrático médio (MSE).

Aplicados os passos anteriores, estão reunidas todas as condições para efetuar o treino da RNA. Para iniciar o treino desta, utiliza-se o comando *train*. A Figura 7.7, representa a estrutura de treino utilizada para a obtenção da RNA. Os blocos de normalização são utilizados para poder atingir a melhor performance da rede (que se dá para valores de entrada e saída entre -1 e 1).

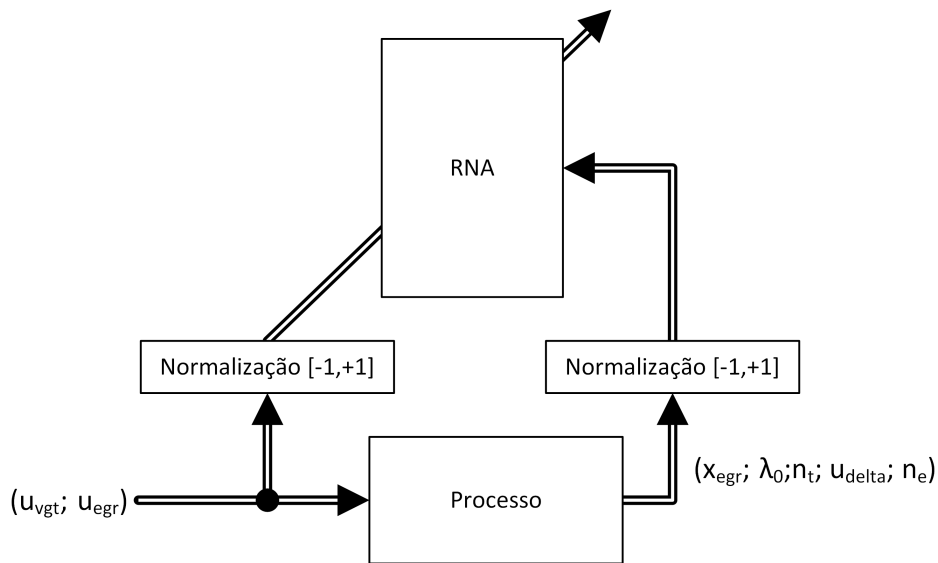


Figura 7.7: Diagrama de treino da RNA para o controlo com estabilizador fixo (modelação inversa).

Após o treino da RNA terminado, há a necessidade de ser testada para validar a sua aplicação. Para tal, a RNA tem de ser comparada com o processo, para demonstrar se esta é capaz de reproduzir a dinâmica inversa do sistema. O método utilizado para o teste, encontra-se representado no diagrama da Figura 7.8. À RNA, aplica-se as entradas de referência $(\lambda_0^s,$

x_{egr}^s , u_{delta}^s e n_e^s) e comparam-se com os as saídas referentes, à saída do processo (λ_0 , x_{egr} , u_{delta} e n_e). Caso a RNA esteja bem dimensionada, os sinais à saída do processo devem seguir os sinais de referência.

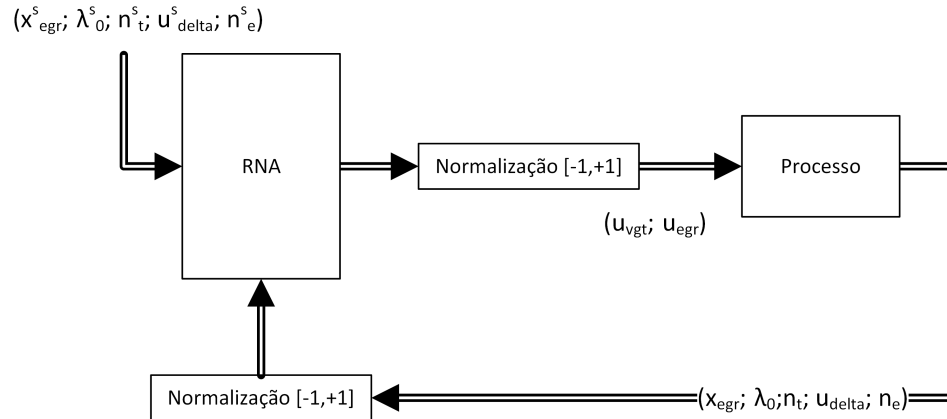


Figura 7.8: Diagrama de testes da RNA para o controlo com estabilizador fixo (modelação inversa).

Criada e testada a rede, esta pode ser adicionada ao controlador com estabilizador fixo. Tendo por princípio a arquitetura analisada na secção 3.1. Através das adaptações necessárias, à configuração do controlo para o projeto criado, obtém-se o diagrama de blocos presente na Figura 7.9 (Anexo B.1. Modelo Simulink do Controlador Estabilizador).

O bloco PID, presente no controlo estabilizador fixo, é o desenvolvido na secção 7.2. Neste controlo, são aplicados quatro PIDs para o controlo do processo, os quais foram sintonizados através da minimização da função (7.10).

7.4 Índices de Desempenho

Para ter uma base de comparação entre o desempenho dos sistemas de controlo testados, é necessário comparar os resultados obtidos. Numa primeira

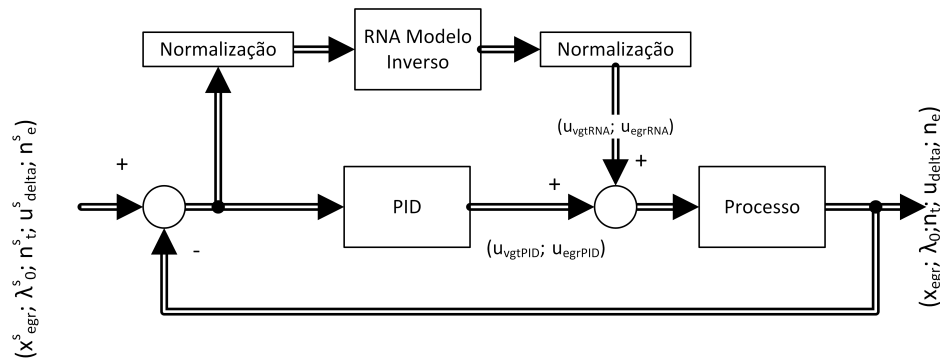


Figura 7.9: Diagrama de testes da RNA para o controle com estabilizador fixo (modelação inversa).

análise, o método gráfico pode ser o suficiente quando se pretende visualizar diferenças mais acentuadas.

Contudo, as respostas obtidas podem ter alterações mínimas não sendo perceptíveis visualmente. Por tal motivo é requerido um método de análise mais cuidado. Este deve garantir que mesmo as mais pequenas alterações são avaliadas e contabilizadas.

Para analisar de uma forma mais fina as diferenças entre as respostas aplicam-se índices de desempenho: integral do erro quadrático (ISE) e integral do erro absoluto (IAE).

Os índices de medida escolhidos são utilizados frequentemente em estudos de simulações. Neste tipo de simulações requer a existência de uma experiência a ser realizada. Esta deve ter por base um sinal de referência a ser seguido. Ao longo do período de simulação é obtido o integral do erro. Caso o erro não seja capaz de se aproximar de zero, a componente integral vai levar a que o resultado do critério tenha valores muito elevados.

O primeiro índice a ser analisado é o ISE. Este índice penaliza em grande parte os maiores erros em detrimento dos menores. O ISE, pode ser calculado aplicando a seguinte expressão:

$$ISE = \int_0^t (r_i - y_i)^2 dt \quad (7.12)$$

O segundo método de medida é o IAE. Este método efetua o integral do erro absoluto. Ao contrário do ISE, este não acrescenta peso aos erros do sistema. A expressão, que define o IAE, é a seguinte:

$$IAE = \int_0^t |r_i - y_i| dt \quad (7.13)$$

Capítulo 8

Testes e Resultados

Neste capítulo são apresentados os testes e resultados, para cada um dos métodos de controlo apresentados no capítulo 7.

Para obter diferentes tipos de resultados são utilizados: métodos gráficos, a função (7.10) ($V(\theta)$) e os índices de desempenho (IAE e ISE). Todos estes métodos são obtidos através de simulações em ambiente MATLAB/Simulink. Os intervalos das formas de onda, são ajustados para permitir uma melhor análise dos gráficos apresentados.

As formas de onda de referência utilizadas, são baseadas nos sinais do artigo [54].

Os testes efetuados, serão utilizados para estudar os efeitos introduzidos pela utilização de RNAs em sistemas de controlo. Retirando assim conclusões, sobre a sua aplicabilidade num caso específico, que envolve o controlo de emissões, em motores Diesel.

8.1 Controlo PID

O controlo PID aqui analisado, destina-se a servir como método de comparação e como controlo estabilizador para o controlo com estabilizador fixo. Este tipo de controlo, é constituído por quatro controladores PID (três PI e

um PID) com realimentação (descrito na secção 7.2). O sistema de controlo desenvolvido, tem como ganhos iniciais os resultados obtidos da referência [53]. Aos valores dos ganhos, foi aplicada uma sintonia fina através da função *fminsearch*. Os sinais PIDinc e PIDo (nesta secção), são respetivamente o controlo PID sem ganhos otimizados e com ganhos otimizados. Os resultados obtidos pela minimização da função $V(\theta)$, são os expostos na Tabela 8.1. Verifica-se, que a diferença entre os erros alcançados pela minimização da função $V(\theta)$ é mínima. Os ganhos obtidos, são apresentados na Tabela 8.2.

Tabela 8.1: Resultados obtidos para o controlador PID sem ganhos ajustados (PIDinc) e com ganhos ajustados (PIDo).

	$V(\theta)$
PIDinc	2,0009
PIDo	2,0004

Tabela 8.2: Ganhos obtidos para o controlador não otimizado (PIDinc) e otimizado (PIDo).

$\gamma_{\lambda_0} = 3/2$ e $\gamma_{x_{egr}} = 1/2$	PIDinc	PIDo
K1	100	100,0003
Ki1	180	183,9373
K2	450	450,0389
Ki2	900	750,4855
K3	100	31,58438
Ki3	420	566,0602
K4	316×10^{-6}	$100,0406 \times 10^{-6}$
Ki4	310×10^{-6}	$204,0819 \times 10^{-6}$
Kd4	140×10^{-6}	$449,9972 \times 10^{-7}$

Na Figura 8.1, está presente o gráfico do sinal de saída λ_0 (para o controlo PID otimizado (PIDo) e não otimizado (PIDinc)). Pela análise visual do gráfico constata-se que, ambos os sinais seguem de forma muito próxima a referência definida. Entre os resultados, destaca-se uma ligeira melhoria para o controlo PIDo. Os *overshoots* do PID otimizado (PIDo) são ligeiramente menores, contudo não são alterações muito expressivas.

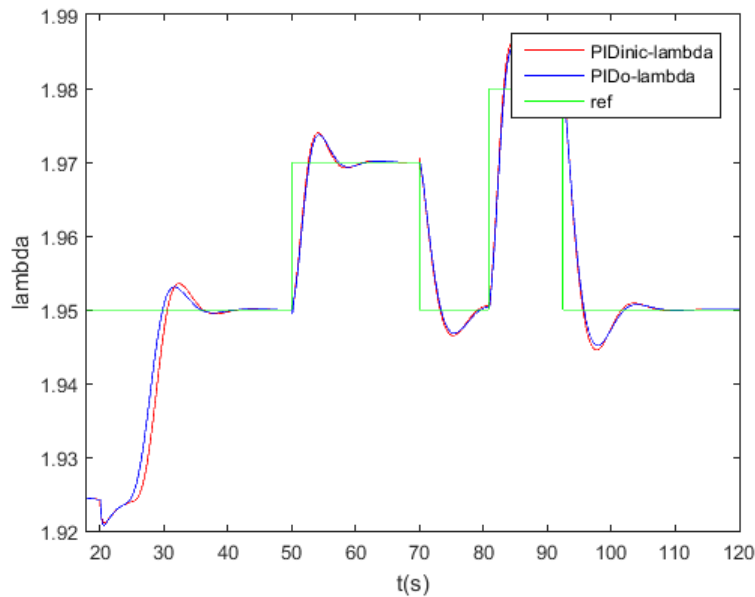


Figura 8.1: Resposta de saída do sinal λ_0 , para o controle otimizado (PID) e não otimizado (PIDinic).

O sinal de saída x_{egr} , é representado na Figura 8.2. Através da análise visual feita ao gráfico, é possível perceber a existência de uma melhoria mais expressiva entre as respostas (em relação ao que foi analisado para λ_0). Observa-se que o controle não otimizado (PIDinic) apresenta um grau de oscilação superior, em diversas zonas, relativamente ao controlador otimizado (PIDo). Apesar dessas irregularidades ambos conseguem seguir a referência, com resultados muito próximos.

O último parâmetro a analisar é a velocidade de rotação do turbo. Este sinal de saída deve manter-se abaixo de um limite de rotação máximo de 10×10^4 rpm. O que se verifica, na Figura 8.3, tanto para o controle otimizado (PIDo), como para o controle não otimizado (PIDinic).

Utilizando os índices IAE e ISE, para os parâmetros λ_0 e x_{egr} , chega-se à conclusão obtida através das restantes análises. Nas Tabelas 8.3 e 8.4, verifica-se que os valores para o controle PIDo são melhores que os obtidos para o PIDinic.

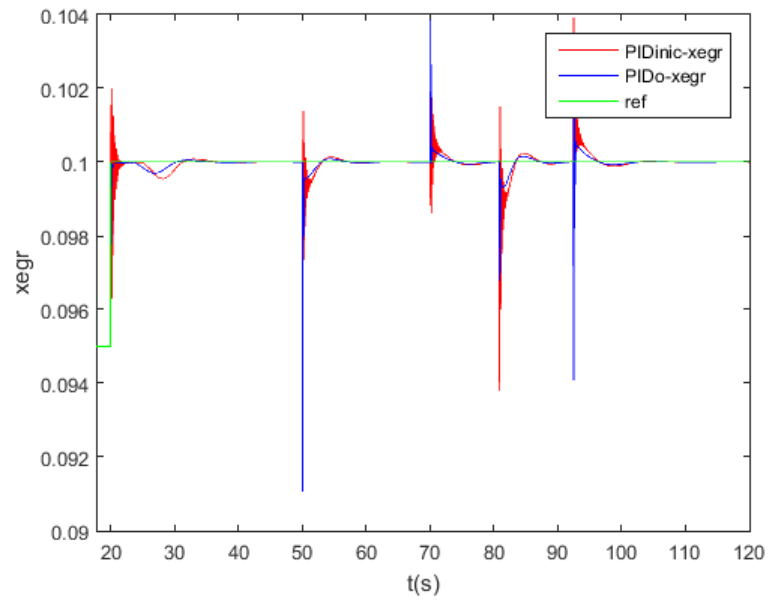


Figura 8.2: Resposta de saída do sinal x_{egr} , para o controlo otimizado (PIDo) e não otimizado (PIDinic).

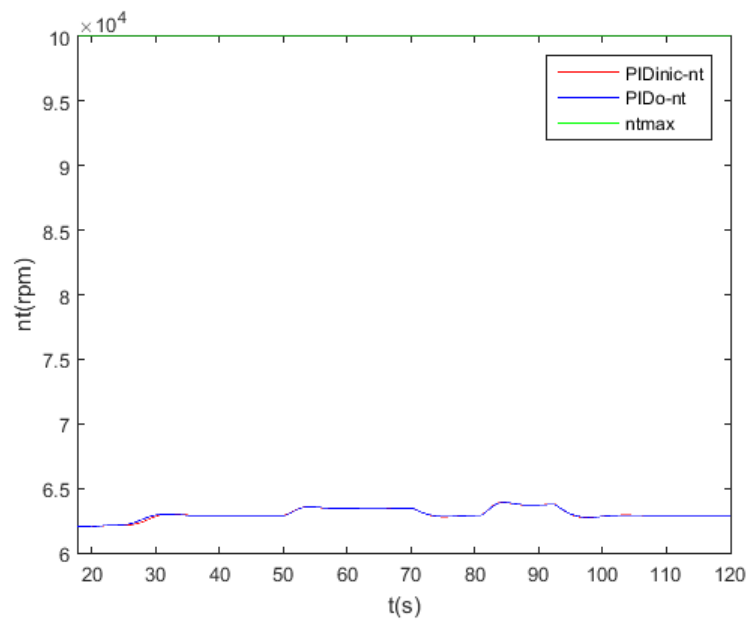


Figura 8.3: Resposta de saída do sinal n_t , para o controlo otimizado (PIDo) e não otimizado (PIDinic).

Tabela 8.3: Tabela com os resultados IAE e ISE de λ_0 , para o controlo PIDinic e PIDo.

Controladores	PIDinic(lambda)		PIDo(lambda)	
Índices	IAE	ISE	IAE	ISE
Resultados	0,4462	0,0084	0,4284	0,0080

Tabela 8.4: Tabela com os resultados IAE e ISE de x_{egr} , para o controlo PIDinic e PIDo.

Controladores	PIDinic(xegr)		PIDo(xegr)	
Índices	IAE	ISE	IAE	ISE
Resultados	0,0120	$8,3938 \times 10^{-6}$	0,0075	$3,8795 \times 10^{-6}$

Os valores de erro obtidos ($V(\theta)$), os resultados gráficos e os índices IAE e ISE, permitem apurar a existência de ligeiras melhorias, alcançadas pela sintonia fina aplicada. Por este motivo, o controlo PID otimizado é utilizado como estabilizador e como ponto de comparação entre os métodos de controlo.

8.2 Configuração da RNA

Para criação da RNA, foram avaliados diferentes números de neurónios na camada oculta, mantendo-se sempre o número de camadas da rede (uma oculta e uma de saída [1]). A qualidade da aproximação atingida por cada teste, foi medida utilizando o erro quadrático (MSE) e a regressão linear (R) (duas das formas de medida disponibilizadas na *Toolbox* de RNAs do MATLAB).

O primeiro método de medida, MSE mede a média da diferença quadrática entre as saídas e as entradas, sendo que zero significa que não existe erro. O segundo método, mede a relação entre as saídas e as entradas, em que valores próximos de 1 significa uma relação muito próxima e valores próximos de 0, uma relação mais divergente. A Tabela 8.5, contém os valores dos erros, relações e iterações para cada uma das RNAs testadas.

O melhor resultado obtido foi para 50 neurónios (camada oculta), onde se constata um menor erro (MSE) e uma melhor relação (R), nos valores ob-

tidos. Contudo, foi referido no capítulo 7 que a rede oculta utilizada foi de 25 neurónios. Justifica-se esta escolha pelo número de iterações requeridas e no *overfit* da rede com 50 neurónios na camada oculta. O número de iterações cresce com a quantidade de neurónios presentes na rede, aumentando o tempo de treino. Isto deve-se ao número de ligações que são acrescentadas, que se traduz em mais cálculos para o mesmo *hardware*. A segunda questão é o *overfit* da RNA. O valor da relação chega a um resultado muito próximo de 1 (0,999). Quando as RNAs são comparadas nos limites de funcionamento máximo do processo, em que o valor de rotação do motor é 1700 rpm, a rede de 50 neurónios não apresenta resultados tão bons como a rede de 25 neurónios.

Tabela 8.5: Resultados obtidos para o desempenho da RNA, com diferentes números de neurónios.

Número de neurónios	MSE	R	Iterações
5	$2,05 \times 10^{-2}$	$9,73 \times 10^{-1}$	475
15	$4,91 \times 10^{-3}$	$9,93 \times 10^{-1}$	953
25	$1,97 \times 10^{-3}$	$9,97 \times 10^{-1}$	1202
50	$1,28 \times 10^{-4}$	$9,99 \times 10^{-1}$	1931

8.3 Controlador com Estabilizador Fixo

O controlador com estabilizador fixo, incorpora os dois tipos de controlo analisados no capítulo 7. O controlo PID, como controlador estabilizador e a RNA como controlo de referência. Os resultados a seguir apresentados, são referentes às formas de onda de saída do processo, para o controlo com estabilizador fixo (NNPID) e o controlo PID já otimizado (PID).

Como parâmetros de análise são utilizadas as respostas das saídas λ_0 , x_{egr} e n_t , as quais são definidas como objetivos de controlo na secção 7.1.

Por forma a ter diferentes pontos de análise numéricos, são utilizados os índices de desempenho IAE e ISE (apresentados na secção 7.4) e a função

$V(\theta)$. Para existir uma perspetiva mais geral das alterações causadas por cada método de controlo, são apresentados os gráficos de cada uma das respostas (λ_0 , x_{egr} e n_t), em diferentes pontos de funcionamento do motor.

8.3.1 Resultados Globais

A primeira medição, a ter em conta para análise e comparação dos resultados, é a função $V(\theta)$. Com esta é possível ter em consideração os erros de todos os parâmetros de saída a controlar.

Na Tabela 8.6, são apresentados os resultados dos controladores para diferentes pontos de rotação do motor. Partindo da análise desta, constata-se que em todos os pontos de funcionamento do motor, o controlador com estabilizador fixo (referido como NNPID) tem melhores resultados.

Para ambos os métodos de controlo, o erro aumenta com a velocidade. À velocidade de 1200 rpm, o erro entre os controlos é próximo de zero. Para a velocidade de 1400 rpm, o controlo com estabilizador fixo já mostra um melhor resultado, em que a diferença entre os dois valores se torna mais perceptível. O ponto mais relevante na diferença entre os controlos, é quando se atinge uma velocidade de 1500 rpm. Neste caso a diferença entre os erros, é mais acentuada. Com o atingir da velocidade máxima, o aumento da diferença entre os erros também é máximo. O controlador NNPID atinge uma melhoria no valor do erro com uma diferença de 1 décima.

Tabela 8.6: Tabela referente aos resultados obtidos para o controlo estabilizador (NNPID) e o controlo PID que foi otimizado, tendo por medida $V(\theta)$.

Controladores	NNPID	PID
Velocidades	$V(\theta)$	$V(\theta)$
1200	2,0002	2,0004
1400	2,0004	2,0093
1500	2,0077	2,0679
1700	2,5125	2,6728

A análise efetuada permite afirmar que, baseado nos resultados obtidos pelo calculo de $V(\theta)$, o controlador estabilizador fixo conseguiu alcançar

melhores resultados em comparação com o controlo PID otimizado.

8.3.2 Análise Gráfica

Os gráficos a seguir apresentados, representam as formas de onda para diferentes velocidades de rotação do motor, das saídas λ_0 , x_{egr} e n_t . Os controlos representados, são: o PID (controlo PID otimizado) e o NNPID (controlo com estabilizador fixo).

Resultados Gráficos - λ_0

As Figuras 8.4, 8.5, 8.6 e 8.7, correspondem aos valores da saída λ_0 (rácio de oxigénio-combustível) para as velocidades 1200, 1400, 1500 e 1700 (rpm), respetivamente.

Com a análise visual da Figura 8.4, é perceptível que o controlo PID consegue obter melhores resultados para a saída λ_0 (com o motor a uma velocidade de 1200 rpm). Constata-se, que na fase inicial o controlo PID consegue estabilizar mais rápido, em comparação com controlo com estabilizador fixo. A partir do instante $t = 50$ segundos, ambos os controlos seguem de forma equivalente o sinal referência.

A Figura 8.5, contém a resposta do sinal λ_0 , para uma velocidade de 1400 rpm. É perceptível, que o NNPID estabiliza mais rápido, em relação ao controlo PID. Comparando com o caso anterior (velocidade de 1200 rpm), os dois controladores conseguem chegar a uma zona estável mais rapidamente. É possível visualizar que ambos os sinais, a partir do instante $t = 30$ segundos, seguem de forma igual a referência.

Com a velocidade estabelecida para 1500 rpm, ambos os controlos mostram alguma dificuldade em reproduzir o primeiro impulso, da fase inicial do sinal de referência (Figura 8.6). Contudo o sinal para o controlo com estabilizador fixo, consegue no geral uma melhor resposta em comparação com o PID.

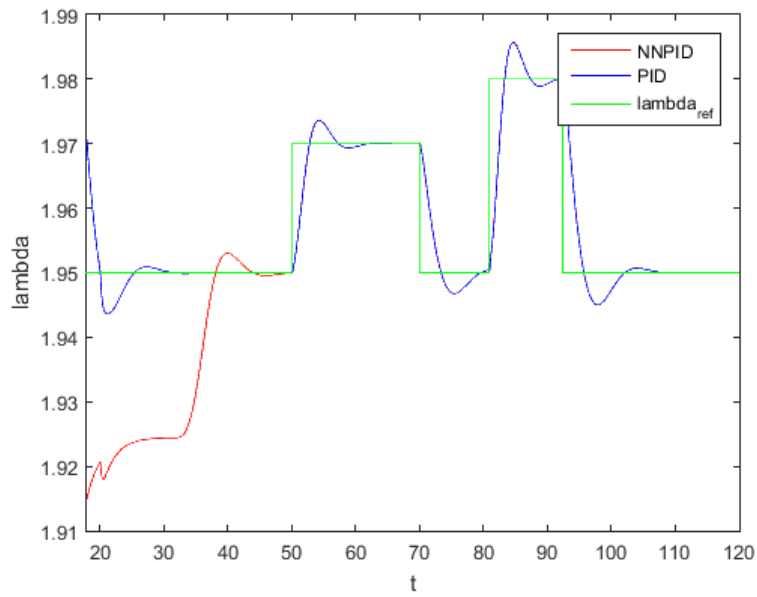


Figura 8.4: Resposta de saída do sinal λ_0 , para a velocidade de 1200 rpm.

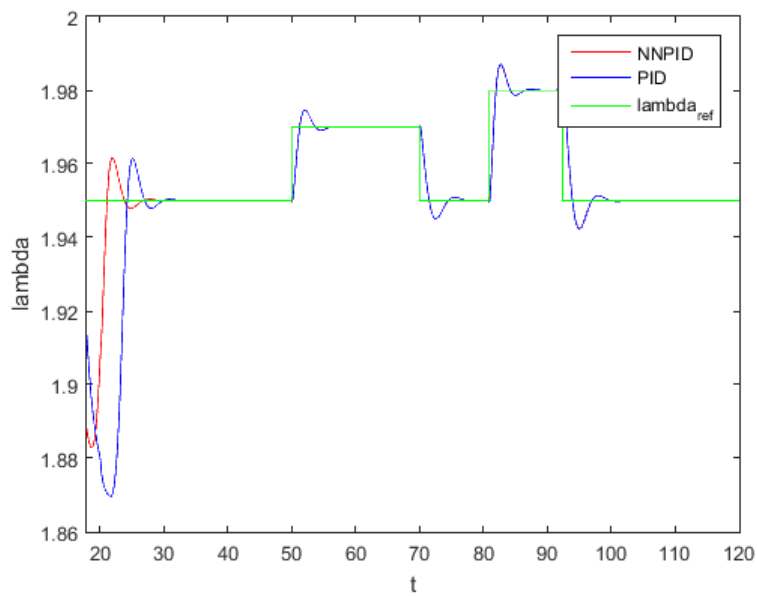


Figura 8.5: Resposta de saída do sinal λ_0 , para a velocidade de 1400 rpm.

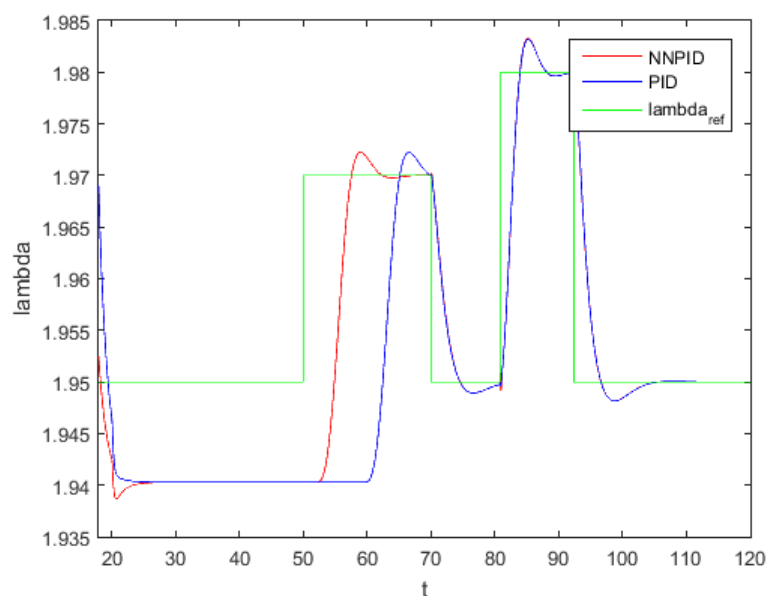


Figura 8.6: Resposta de saída do sinal λ_0 , para a velocidade de 1500 rpm.

Na Figura 8.7, é apresentado o gráfico da saída λ_0 , para a velocidade de 1700 rpm. Com a visualização do gráfico, observa-se uma melhor resposta na fase inicial do controle com estabilizador fixo, em que este estabiliza mais rapidamente que o controle PID.

Tal como para as velocidades anteriores, os controles seguem a referência de forma muito próxima depois de estabilizados.

Comparando os resultados obtidos graficamente e os obtidos na Tabela 8.6, percebe-se que a forma de onda para uma velocidade de 1200 rpm, não transmite a ideia de melhoria associada a $V(\theta)$. Para os restantes pontos de funcionamento do sistema, os resultados obtidos são melhores para o controle com estabilizador fixo, nos dois diferentes métodos de análise.

Ao analisar os dois pulsos presentes em todas as formas de onda de λ_0 , verificam-se algumas diferenças. O *overshoot* na segunda onda é sempre superior ao da primeira, tanto para o controle PID como para o controle com estabilizador fixo. Esta constatação permite mostrar como o processo a

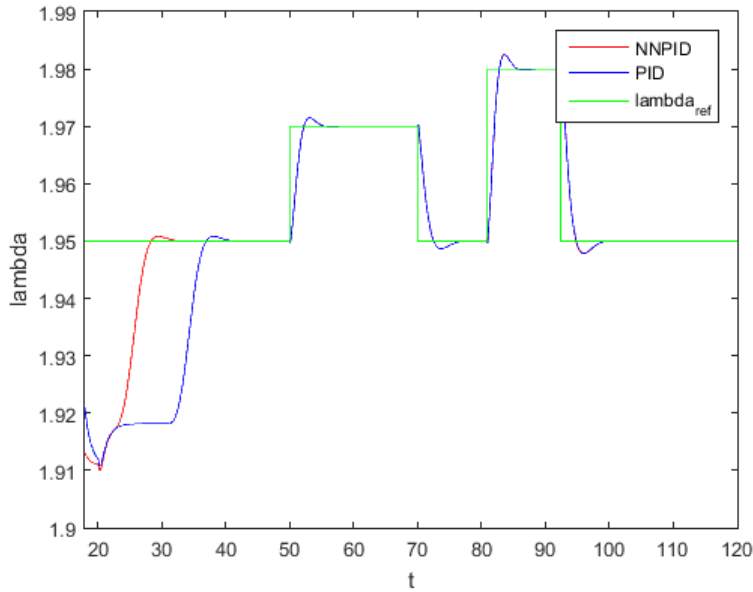


Figura 8.7: Resposta de saída do sinal λ_0 , para a velocidade 1700 rpm.

controlar é um sistema não linear.

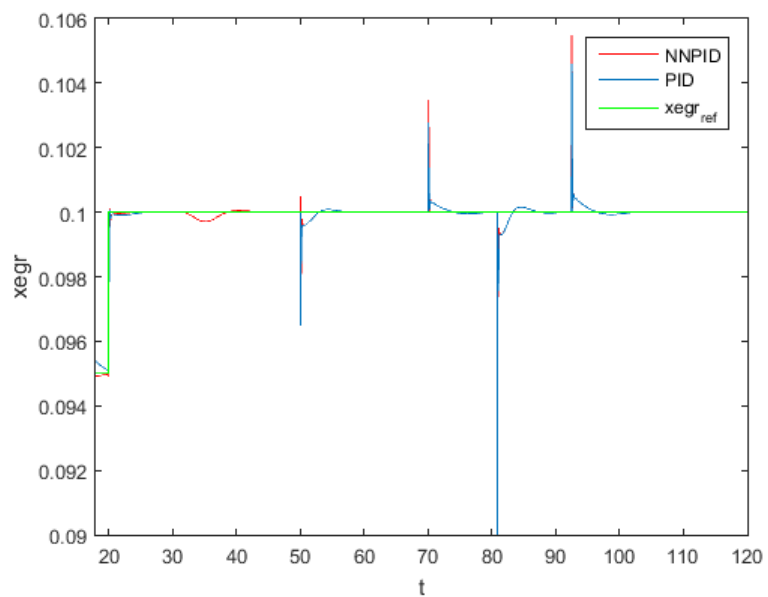
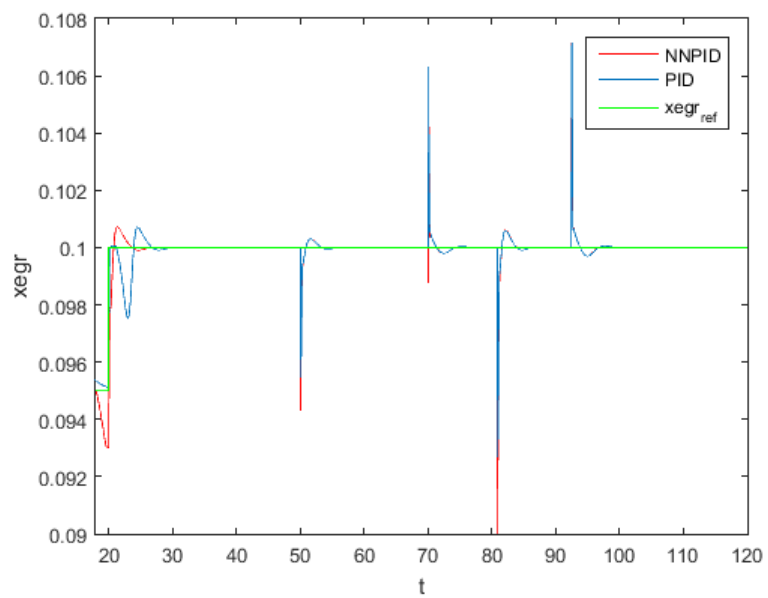
Resultados Gráficos - x_{egr}

Os resultados gráficos obtidos para o sinal de saída da fração de gases recirculados (x_{egr}), são apresentados nas Figuras 8.8, 8.9, 8.10 e 8.11. Os resultados, são obtidos para os mesmos pontos de velocidade aplicados a λ_0 .

No caso das respostas obtidas, não é perceptível graficamente se existem melhorias, entre as respostas do controle com estabilizador fixo e do controle PID. Pode-se afirmar, que ambos os controladores conseguem seguir o sinal de referência, para todas as velocidades. Contudo, existem zonas onde os controles apresentam alguns picos, voltando a estabilizar rapidamente.

Resultados Gráficos - n_t

A resposta de saída n_t , referente à velocidade de rotação do turbo, deve ser sempre inferior a um limite de rotação máximo (10×10^4), para os pontos

Figura 8.8: Resposta de saída do sinal x_{egr} , para a velocidade de 1200 rpm.Figura 8.9: Resposta de saída do sinal x_{egr} , para a velocidade de 1400 rpm.

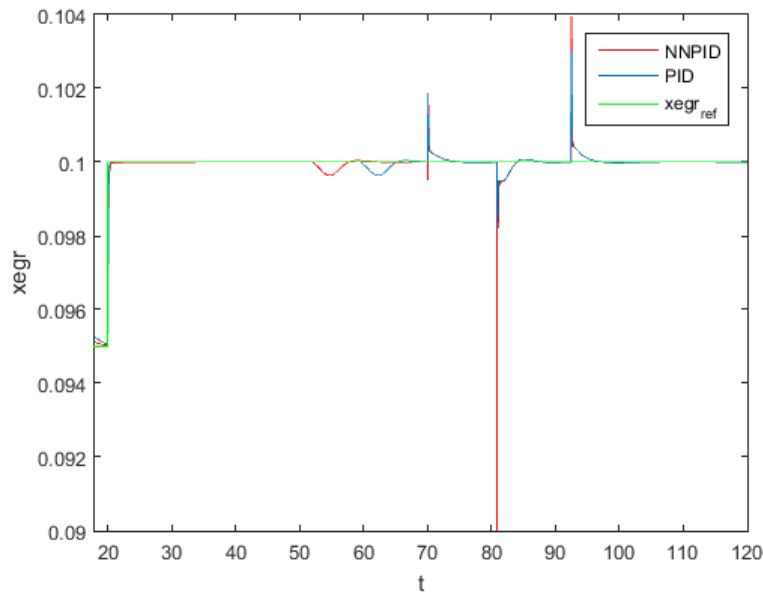


Figura 8.10: Resposta de saída do sinal x_{egr} , para a velocidade de 1500 rpm.

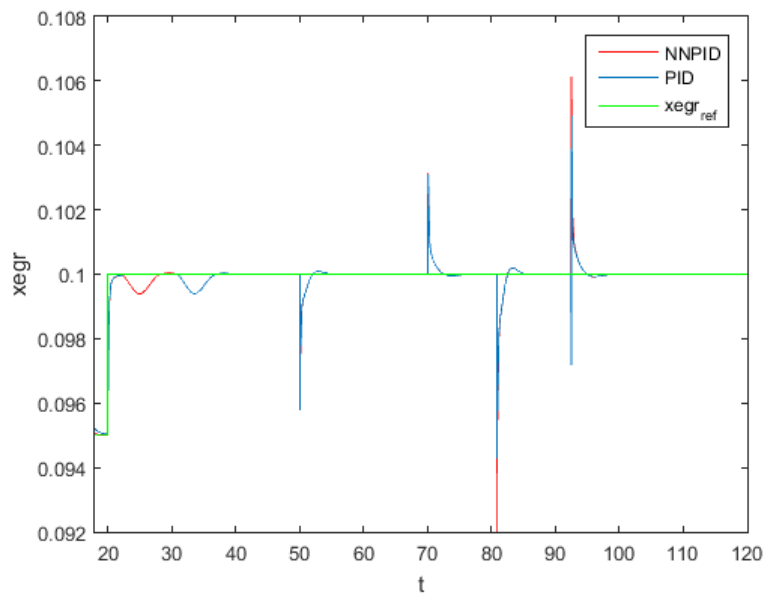


Figura 8.11: Resposta de saída do sinal x_{egr} , para a velocidade de 1700 rpm.

de velocidade já abordados. Esta saída, ao contrário das anteriores, não necessita de seguir qualquer referência. A sua importância, está em limitar a velocidade de rotação do turbo, para este não ser danificado. Visualizando as Figuras 8.12, 8.13, 8.14 e 8.15, é perceptível que ambos os métodos de controlo conseguem impor a restrição colocada.

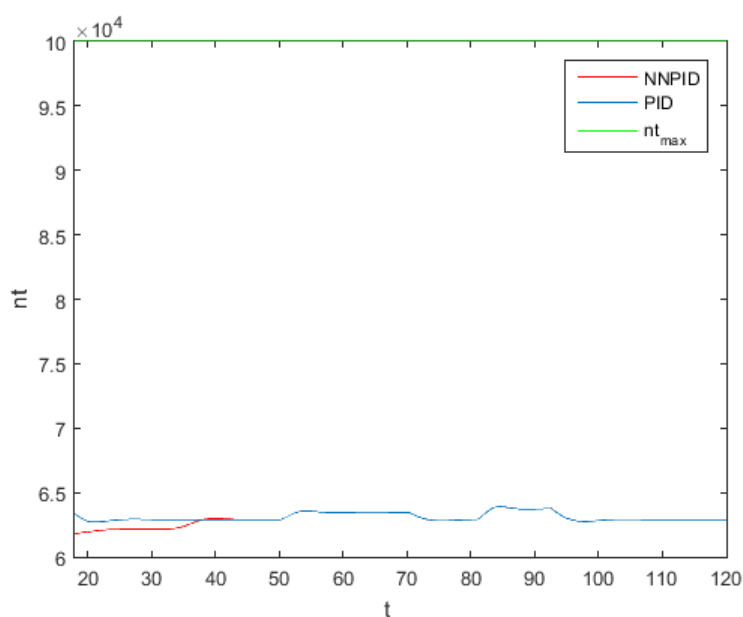


Figura 8.12: Resposta de saída do sinal n_t , para a velocidade de 1200 rpm.

8.3.3 Índices de Desempenho IAE e ISE

A utilização dos índices IAE e ISE, permitem analisar o erro entre os sinais de saída e a referência dos parâmetros λ_0 e x_{egr} , para poder comparar com os resultados obtidos através de $V(\theta)$ e graficamente. Para tal, são apresentadas as Tabelas 8.7 e 8.8, para o controlo com estabilizador fixo (referenciado em tabela como NNPID) e o controlo PID otimizado (referenciado em tabela como PID).

Iniciando a análise pelo parâmetro λ_0 (Tabela 8.7), para a velocidade de rotação de 1200 rpm, ambos os índices mostram melhores resultados, para

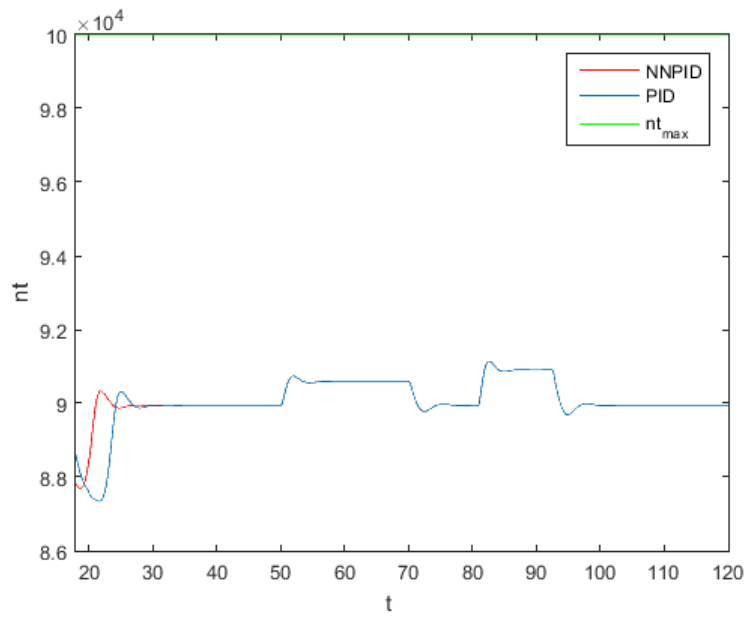


Figura 8.13: Resposta de saída do sinal n_t , para a velocidade de 1400 rpm.

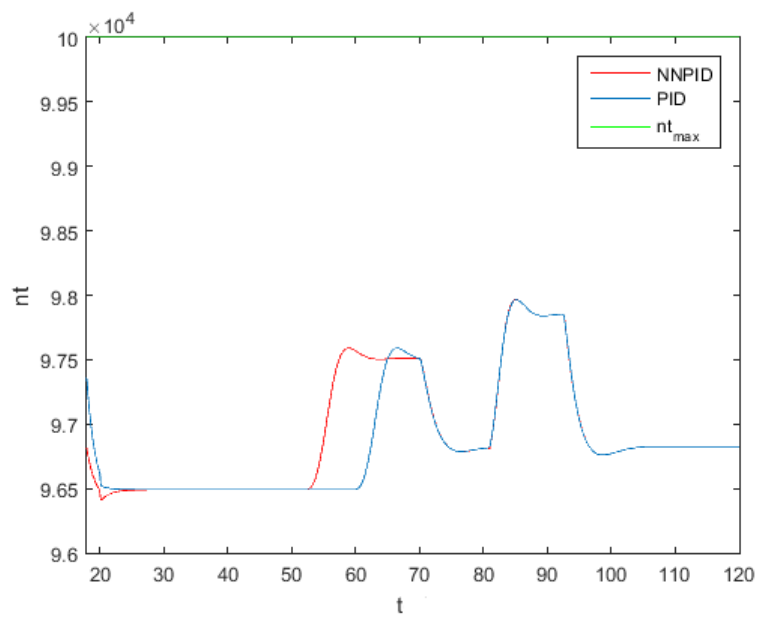
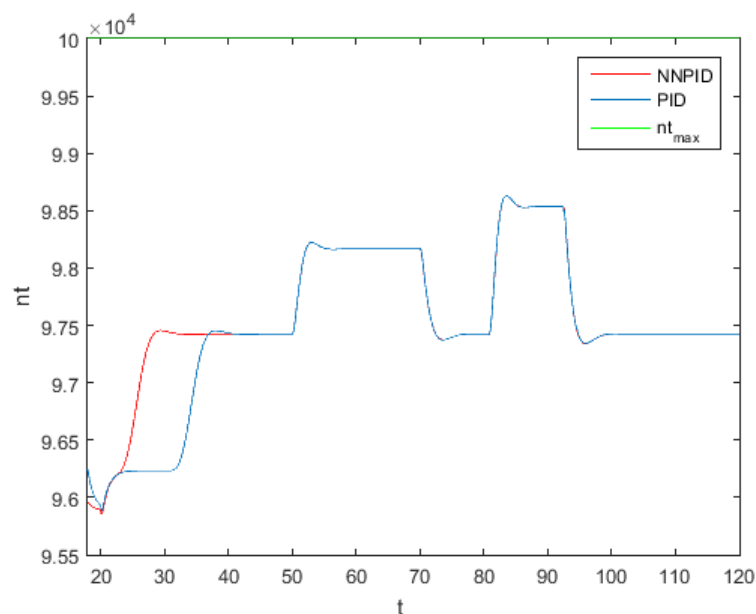


Figura 8.14: Resposta de saída do sinal n_t , para a velocidade de 1500 rpm.

Figura 8.15: Resposta de saída do sinal n_t , para a velocidade de 1700 rpm.Tabela 8.7: Tabela com os resultados IAE e ISE de λ_0 , para diferentes velocidades de funcionamento do motor.

Controladores	NNPID(lambda)		PID(lambda)	
	IAE	ISE	IAE	ISE
1200	0,6618	0,0142	0,2352	0,0031
1400	0,1723	0,0027	0,4118	0,0208
1500	0,6137	0,0095	0,8409	0,0163
1700	0,3241	0,0081	0,5924	0,0166

Tabela 8.8: Tabela com os resultados IAE e ISE de x_{egr} , para diferentes velocidades de funcionamento do motor.

Controladores	NNPID(xegr)		PID(xegr)	
	IAE	ISE	IAE	ISE
1200	0,0075	$6,8875 \times 10^{-6}$	0,0065	$6,6047 \times 10^{-6}$
1400	0,0101	$2,5021 \times 10^{-5}$	0,0125	$2,7002 \times 10^{-6}$
1500	0,0051	$4,5463 \times 10^{-6}$	0,0054	$4,5995 \times 10^{-6}$
1700	0,0103	$1,5865 \times 10^{-6}$	0,0102	$1,5724 \times 10^{-6}$

o controlo PID. A partir da velocidade de rotação 1400 rpm, invertem-se os resultados. O controlador com estabilizador fixo, passa a apresentar os melhores resultados, para os índices IAE e ISE.

O maior erro encontra-se para a velocidade de 1500 rpm. A maior diferença entre resultados é obtida para a velocidade de 1700 rpm, em que o controlo estabilizador apresenta um melhor resultado.

Os resultados do parâmetro x_{egr} , como foi possível constatar graficamente, apresentam formas de onda muito próximas, para os dois controlos. Com a utilização dos índices, é possível discriminar alguns casos: para as velocidades 1200 e 1700 (rpm), os valores dos índices são ligeiramente melhores para o controlo PID; para as velocidades 1400 e 1700 (rpm), os índices são ligeiramente melhores para o controlo com estabilizador fixo. A maior diferença entre os erros encontra-se para a velocidade de 1400 rpm.

8.3.4 Sumário dos Resultados

Os resultados obtidos, não são coerentes em todos os pontos de funcionamento analisados. Constata-se que para a velocidade de 1200 rpm os parâmetros λ_0 e x_{egr} , apresentam melhores resultados para o controlo PID. O que leva a afirmar, que o controlador com estabilizador fixo, não conseguiu alcançar qualquer melhoria. Acontece o inverso para as velocidades de 1400 e 1500 rpm. Nestas duas situações, todos os métodos de análise ($V(\theta)$, análise gráfica e índices (IAE e ISE), mostram que o controlador estabilizador conseguiu melhorar as respostas.

Para a velocidade de 1700 rpm, o controlador com estabilizador fixo mostrou os melhores resultados, no que toca ao controlo do parâmetro λ_0 (para todos os tipos de análises efetuadas). Para o caso do parâmetro x_{egr} , os índices IAE e ISE, mostram que ambos os controlos conseguiram ter uma resposta muito próxima, sendo que o controlo PID apresenta um erro ligeiramente menor.

Com as análises efetuadas, é possível concluir que o controlo com estabilizador fixo consegue melhorar as respostas do sistema, para a maioria dos pontos de funcionamento testados. Mostra-se desta forma, a sua validade no

controlo do sistema de emissões do motor Diesel.

Capítulo 9

Conclusões

Neste capítulo retratam-se as conclusões retiradas dos estudos efetuados ao longo do documento. Este surge como um complemento, às análises efetuadas em cada um dos capítulos apresentados.

9.1 Conclusões Gerais

O comportamento dinâmico de um processo com base num motor Diesel com VGT e EGR foi analisado, recorrendo a metodologias clássicas e a metodologias baseadas em redes neuronais artificiais. A interligação destes modelos com sistemas de controlo e a sua posterior integração em diferentes configurações para controlo de emissões do processo, constitui os objetivos principais deste trabalho.

Numa primeira fase, analisou-se o funcionamento das RNAs, onde se abordou o desenvolvimento histórico das redes, a relação entre RNAs biológicas e artificiais, o processo de criação e as aplicações associadas. Com o estudo efetuado nesta primeira fase, tornou possível compreender os conceitos básicos para utilização das RNAs no controlo do processo pretendido.

Como passo seguinte, efetuou-se uma análise a várias ferramentas de controlo, que permitem simplificar a utilização das RNAs. Tendo-se optado

pelo *software* MATLAB/Simulink, dadas as suas capacidades, de integração de sistemas e desenvolvimento de RNAs.

Com o conceito de controlo a aplicar analisado e as ferramentas a utilizar definidas, efetuou-se um estudo baseado no processo. Com este, foi possível retirar algumas conclusões acerca dos conceitos associados às emissões e de que forma se poderia controlar um motor Diesel, para o controlo das mesmas.

Como quarto passo, foi analisado o modelo proposto para controlo. Este passo tem como fundamento a análise mais detalhada do modelo de um motor Diesel VGT e EGT para veículos pesados, desenvolvido por Johan Wahlström e Lars Eriksson. Este foi o modelo escolhido para o controlo, por ser validado em comparação com um motor real, ser um modelo de livre acesso e conter toda a documentação requerida para a modelação do mesmo.

Após os estudos e análises atrás mencionados, foi proposto a aplicação de dois controlos diferentes. Um primeiro controlo PID com realimentação, o qual foi dimensionado, com base num controlo já aplicado ao processo. A este foi ainda aplicado uma sintonia fina, para ajustar aos sinais de saída pretendidos. Os sinais de saída utilizados para o controlo, foram o λ_0 , o x_{egr} e o n_t .

Estando na base do trabalho a aplicação das RNAs para o controlo do sistema, foi proposto como segundo método um controlo com estabilizador fixo. Este é constituído por dois tipos de controladores, PID e RNAs. As variáveis a controlar foram as mesmas utilizadas no controlo PID, sendo este utilizado como método de comparação com o controlo com estabilizador fixo.

Na última fase deste trabalho, foram apresentados testes e resultados dos controlos desenvolvidos. Estes mostraram que no geral o controlador com estabilizador teve melhor desempenho em quase todos os funcionamentos testados. Contudo no ponto de funcionamento do motor Diesel, para a velocidade de 1200 rpm, o controlo PID mostrou melhor desempenho no controlo dos parâmetros λ_0 e x_{egr} .

9.2 Sugestões para Trabalho Futuro

Os temas focados nesta tese podem ser objetos de estudo para trabalhos posteriores. As áreas aqui descritas são diversificadas e o conhecimento existente ainda se encontra em desenvolvimento.

As RNAs podem seguir várias vertentes de aplicação e desenvolvimento, não estando restritas às aqui abordadas. O desenvolvimento de diferentes métodos para o controlo do processo seriam benéficos para a comparação e melhoramento. O que poderia trazer vantagens em relação aos métodos que são aqui aplicados.

Relativamente ao processo, este pode seguir diferentes modos de controlo, sendo possível escolher outros parâmetros de referência a seguir.

Dado que todo o trabalho foi desenvolvido como um modelo, seria interessante validar todo o sistema desenvolvido num processo real. E não obstante disto, analisar a sua extrapolação para outros modelos de motor Diesel com VGT e EGR.

Referências Documentais

- [1] Haykin. "Neural Networks and Learning Machines.", Prentice-Hall, 3rd Edition, 2008.
- [2] Luís Miguel Mota Ferreira. "Projeto de Controladores Usando Redes Neurais", ISEP-DEE, Tese de Mestrado, 2015.
- [3] Poulton, Mary M.. "Computational neural networks for geophysical data processing", Elsevier, 2001.
- [4] House, E.L. Pansky, B.. "A Functional Approach to Neuro-anatomy", McGraw-Hill, New York, 1967.
- [5] Dicionário da Saúde. "Neurónio (célula nervosa)", 13 de Setembro de 2015. Acedido em: 5 Março 2016. Disponível em: <http://dicionariosaude.com/neuronio-celula-nervosa/>.
- [6] Hagan, Martin T., and Howard B. Demuth. "Neural networks for control." American Control Conference, 1999. Proceedings of the 1999. Vol. 3. IEEE, 1999.
- [7] L.G. Kraft e Howard B. Demuth. "A Comparison between CMAC Neural Network Control and Two Traditional Control Systems", IEEE Control Systems Magazine, vol. 10, no. 2, pg. 36-43, 1990.
- [8] M. Kawato. "Computational Schemes and Neural Network Models for Formation and Control of Multijoint Arm Trajectory", Neural Networks

- for Control, W.T. Miller, R.S. Sutton, and P.J. Werbos, Eds., Boston: MIT Press, pg. 197-228, 1990.
- [9] W.T. Miller. "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm", *IEEE Journal of Robotics and Automation*, vol. 3, no. 2, pg. 157-165, 1987.
- [10] B. Widrow and E. Walach. "Adaptive Inverse Control", New Jersey: Prentice Hall, 1996.
- [11] K.S. Narendra and K. Parthasarathy. "Identification and Control of Dynamical Systems Using Neural Networks" *IEEE Transactions on Neural Networks*, vol. 1, pp. 4-27, 1990.
- [12] R.S. Sutton, and A.G. Barto. "Introduction to Reinforcement Learning", Cambridge, Mass.: MIT Press, 1998.
- [13] D.J.C. MacKay. "Bayesian non-linear modelling for the 1993 energy prediction competition", In G. Heidbreder, Editor, *Maximum Entropy and Bayesian Methods*, Santa Barbara 1993, pages 221–234. Kluwer, Dordrecht, 1996.
- [14] D.R. Lovell, M.J.J. Scott, M. Niranjana, R.W. Prager, K.J. Dalton and R. Derom. "On the use of expected attainable discrimination for feature selection in large scale medical risk prediction problems", Technical Report CUED/F-INFENG/TR299, Cambridge University Engineering Department, August 1997.
- [15] V. Tresp, J. Moody and W. DeLong. "Prediction and control of the glucose metabolism of a diabetic", *NIPS Post-Conference Workshop: Neural Network Applications in Medicine*, Vail, CO, 1994.
- [16] C. Lee Giles, S. Lawrence and A.C. Tsoi. "Rule inference for financial prediction using recurrent neural networks", In *Proceedings of IEEE/I-*

- AFE Conference on Computational Intelligence for Financial Engineering (CIFER), pages 253–259. IEEE, Piscataway, NJ, 1997.
- [17] J. Hutchinson, A. Lo, and T. Poggio. "A nonparametric approach to pricing and hedging derivative securities via learning networks", Technical Report A.I. No. 1471, MIT, AI Lab., April 1994.
- [18] J. Moody. "Economic forecasting: challenges and neural network solutions", In Proceedings of the International Symposium on Artificial Neural Networks, Hsinchu, Taiwan, 1995.
- [19] J. Moody. "Fast learning in multi-resolution hierarchies", In D.S. Touretzky, Editor, Advances in Neural Information Processing Systems 2, Morgan Kaufmann Publishers, San Mateo, CA, 1989.
- [20] R. Rosipal, M. Koska, and I. Farkas. "Prediction of chaotic time-series with a resource-allocating RBF network", Neural Processing Letters, 7:1–13, 1998.
- [21] J. Moody, U. Levin, and S. Rehfuss. "Predicting the U.S. index of industrial production", In Proceedings of the 1993 Parallel Applications in Statistics and Economics Conference, Zeist. The Netherlands. Special issue of Neural Network World, 3(6):791–794, 1993.
- [22] R. Duda and P. Hart. "Pattern Classification and Scene Analysis", New York: John Wiley & Sons, 1973.
- [23] R. Gray. "Vector quantization", IEEE ASSP Magazine, pp. 4–29, 1984.
- [24] L. Rabiner and B.H. Juang. "Fundamentals of Speech Recognition", Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [25] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. "Phoneme recognition using time-delay neural networks", IEEE Trans. ASSP, vol. 37, pp. 328–339 1989.

- [26] E. McDermott and S. Katagiri. "LVQ-based shift-tolerant phoneme recognition", *IEEE Trans. SP*, vol. 39, pp. 1398–1411, 1991.
- [27] P. Haffner. "A new probabilistic framework for connectionist time alignment", *Proc. ICSLP94*, pp. 1559–1562, 1994.
- [28] S. Katagiri, B.H. Juang, and C.H. Lee. "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descentmethod", *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, 1998.
- [29] A.J. Robinson. "An application of recurrent nets to phone probability estimation", *IEEE Trans. NN*, vol. 5, no. 2, pp. 298–305 1994.
- [30] V.N. Vapnik. "The Nature of Statistical Learning Theory", New York: Springer-Verlag, 1995.
- [31] C. Jutten and J. Herault. "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture", *Signal Processing*, vol. 24, pp. 1–10, 1991.
- [32] W.Y. Ma and B.S. Manjunath. "Edge flow: a framework of boundary detection and image segmentation", in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 744–749, 1997.
- [33] S. Haykin. "Neural Networks: A Comprehensive Foundation", New York, MacMillan, 1994.
- [34] Applications of Neural Networks in Biomedical Signal Processing special session. "Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP)", Amelia Island Plantation, FL, Sept. 1997.
- [35] Adnan Parlak, Yasar Islamoglu, Halit Yasar e Aysun Egrisogut. "Application of artificial neural network to predict specific fuelconsumption

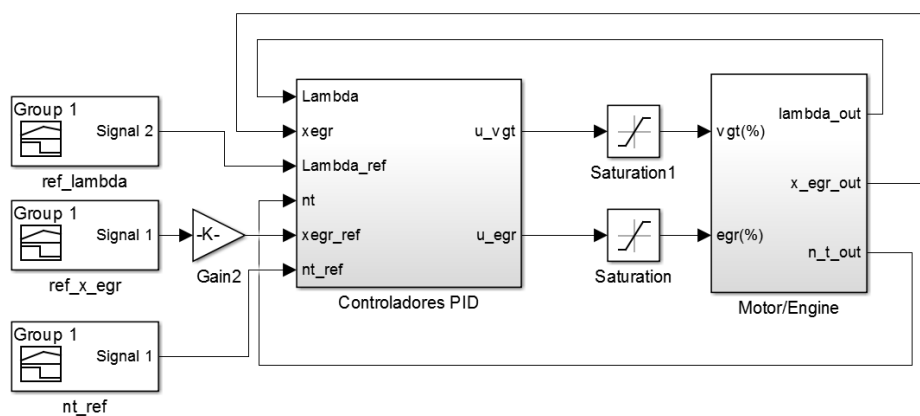
- and exhaust temperature for a Diesel engine", Sakarya University, Technical Education Faculty, Mechanical Education Department, Esentepe, Sakarya, Turkey, 18 November 2005.
- [36] Hidayet Oguz, Ismail Saritas, Hakan Emre Baydan. "Prediction of diesel engine performance using biofuels with artificial neural network", Selcuk University, Turkey, *Expert Systems with Applications* 37 6579–6586, Elsevier Ltd, 2010.
- [37] Brad Aisa, Brian Mingus e Randall C. O'Reilly. "The Emergent Neural Network Modeling System", 2007. Acedido em: 15 Abril 2016. Disponível em: <https://grey.colorado.edu/mediawiki/sites/emergent/images/a/ab/AisaMingusOReilly07.pdf>.
- [38] GoodAI. "Brain Simulator", 2016. Acedido em: 15 Abril 2016. Disponível em: <http://www.goodai.com/#!/brain-simulator/c81c>.
- [39] Helena Andersson, Martin Hedvall. "Model Based Control of Air and EGR into a Diesel Engine", Chalmers University of Technology, Master of Science Thesis , 2008.
- [40] Deng, Jiamei, Stobart, Richard and Maass, Bastian. "The Applications of Artificial Neural Networks to Engines, Artificial Neural Networks - Industrial and Control Engineering Applications", Prof. Kenji Suzuki (Ed.), InTech, 2011.
- [41] Johan Wahlström and Lars Eriksson. "Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics", *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, Volume 225, Issue 7, July 2011.

- [42] Kolmanovsky, Stefanopoulou, Moraal and van Nieuwstadt. "Issues in modeling and control of intake flow in variable geometry turbocharged engines". In Proceedings of 18th IFIP Conference on System modeling and optimization, Detroit, Michigan, USA, July 1997, 1998, pp. 436–445 (Addison-Wesley Longman, Harlow, Essex).
- [43] Vigild, C.. "The internal combustion engine: modelling, estimation and control issues". PhD Thesis, Technical University of Denmark, Lyngby, Denmark, 2001.
- [44] Mock, Peter. "EU CO2 Emission Standards for passenger cars and light-commercial vehicles."International Council on Clean Transportation, 2014.
- [45] Martins, Jorge. "Motores de Combustão Interna", 3^a ed. Porto: Publindústria, Edições Técnicas, 2011.
- [46] Heywood, J. B.. "Internal combustion engine fundamentals", McGraw-Hill, New York, 1988.
- [47] Skogtjärn. "P. Modelling of the exhaust gas temperature for diesel engines", Master's Thesis, Department of Electrical Engineering, Linköping University, Sweden 2002.
- [48] Eriksson. "Mean value models for exhaust system temperatures", SAE paper 2002-01-0374, 2002.
- [49] Van Wylen, G. J., Sonntag, R. E., Borgnakke, C.. "Fundamentals of Thermodynamics", 6a ed., Edgard Blücher Ltda, 2003.
- [50] Chevalier, A., Mueller, M., and Hendricks, E.. "On the validity of mean value engine models during transient operation", SAE paper 2000-01-1261, 2000.

- [51] Johan Wahlström, Lars Eriksson, and Lars Nielsen. "EGR-VGT Control and Tuning for Pumping Work Minimization and Emission Control", IEEE Transactions on Control Systems Technology, vol. 18, No. 4, 2010.
- [52] Johan Wahlström and Lars Eriksson. Output Selection and Its Implications for MPC of EGR and VGT in Diesel Engines, IEEE Transactions on Control Systems Technology, vol. 21, NO. 3, 2013.
- [53] J. Wahlström and L. Eriksson. "Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing nonlinear system dynamics", in Proc. Inst. Mech. Eng. D, J. Automob. Eng., 2011, pp. 1–27.
- [54] J. Wahlström and L. Eriksson. "Nonlinear EGR and VGT control with integral action for diesel engines", in Proc. IFAC Workshop Eng. Powertrain Control Simul. Model., Paris, France, 2009, pp. 1–23.
- [55] K. J. Aström and T. Hägglund. "PID Controllers: Theory, Design and Tuning", Research Triangle Park, Instrument Society of America, 2nd edition, 1995.
- [56] MATLAB. "fminsearch". Acedido em: 2 Maio 2016. Disponível em: <http://www.mathworks.com/help/matlab/ref/fminsearch.html>.

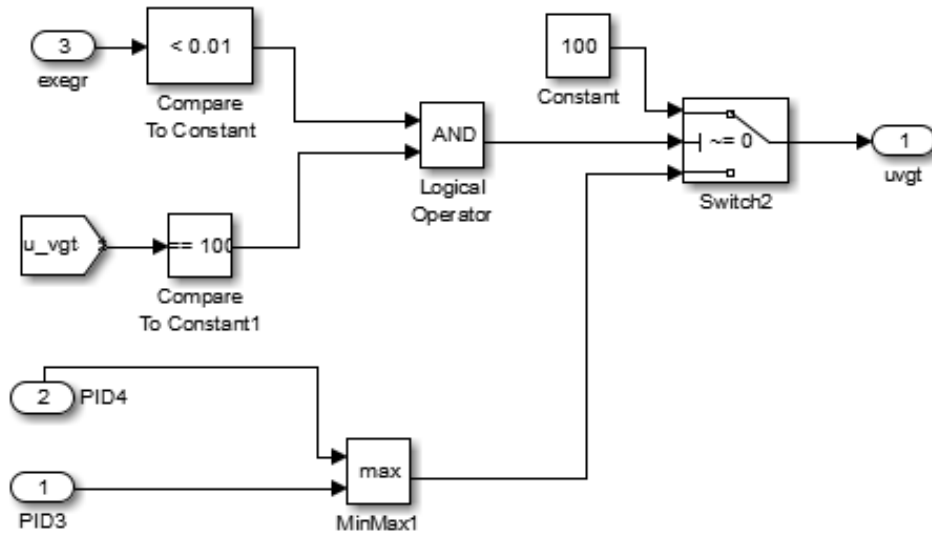
Anexo A. Esquemas do Controlo PID

A.1. Modelo Simulink do esquema geral de controlo PID



Anexo A.

A.1.1.2. Modelo Simulink do bloco Uvgtsub



Anexo B. Esquemas do Controlo com Estabilizador Fixo

B.1. Modelo Simulink do Controlador Estabilizador

