

Controlo de um Conversor de Energia das Ondas usando um sistema multi-agente e métodos de aprendizagem automática

Nuno Miguel Antunes Pereira

Dissertação para obtenção do Grau de Mestre em

Engenharia Mecânica

Orientador: Prof. Duarte Pedro Mata de Oliveira Valério

Júri

Presidente: Prof. Paulo Jorge Coelho Ramalho Oliveira

Orientador: Prof. Duarte Pedro Mata de Oliveira Valério

Vogal: Prof. Pedro Jorge Borges Fontes Negrão Beirão

Junho de 2018

Abstract

Modern societies are characterized by an intense use of energy in transport, industry and electricity production. In most countries, the economic model of development has been largely supported by non-renewable energy sources — particularly, fossil fuels — which present environmental problems, such as climate change, and economic constraints due to inherent scarcity. The transition to renewable energy sources presents a path to a sustainable development. In addition to the commonly known renewable energy sources, like solar and wind, wave energy from oceans and seas promises to be a valuable resource for electricity generation.

This work aims to provide ISWEC (inertial sea wave energy converter) — a device designed to extract power from sea waves using gyroscoping motion to generate electricity — the capability to adjust its PD controller gains according to current sea wave condition, so its performance is maximized. The forecasting process is carried out by a set of machine learning methods, namely neural networks and a SVM (support vector machine) classifier. The distributed arrangement of ISWEC devices on “wave farms” is handled by a multi-agent platform, based on Java, ensuring all devices receive due forecast data from the central agent. It is believed this strategy, as opposed to relying on a exogenous forecast set on an hourly basis, will result in improved performance, since sudden transitions of sea wave conditions may be taken into account thanks to a shorter time basis of gain setting operation and to an autonomous forecasting agent.

Keywords: Wave energy converter, PD controller, Support vector machines, Neural network inverse model, Multi-agent systems

Resumo

As sociedades modernas caracterizam-se por uma utilização intensa de energia nos transportes, indústria e produção de electricidade. Na maioria dos países, o modelo económico de desenvolvimento tem sido largamente suportado por uma utilização de fontes de energia não renováveis — com especial enfoque nas energias de origem fóssil — o que apresenta problemas ambientais, tais como as alterações climáticas, e limitações de ordem económica devido à escassez inerente. A transição para uma utilização de energias renováveis apresenta-se como o caminho para um desenvolvimento sustentável. Além das fontes de energia renovável amplamente conhecidas, por exemplo solar ou eólica, a energia das ondas marítimas apresenta-se como um recurso de elevado potencial para a produção de electricidade.

Este trabalho tem como objectivo dotar o dispositivo ISWEC (inertial sea wave energy converter) — um dispositivo desenvolvido de forma a converter a energia mecânica das ondas em electricidade, utilizando um sistema giroscópico — da capacidade de ajustar automaticamente os ganhos do seu controlador PD de acordo com o estado do mar actual, pretendendo-se maximizar o seu desempenho. O processo de previsão do estado do mar é realizado com recurso a um conjunto de métodos de aprendizagem automática, designadamente redes neuronais e um classificador SVM (máquina de vector suporte). A natureza distribuída dos dispositivos ISWEC numa “wave farm” é suportada por uma plataforma multi-agente, desenvolvida em Java, assegurando-se que todos os dispositivos recebem os respectivos dados de previsão extrapolados pelo agente central. Acredita-se que esta estratégia, em alternativa a uma previsão exógena à “wave farm” recebida numa base horária, resultará numa melhoria do desempenho do conversor dado que rápidas transições do estado do mar poderão ser detectadas e reflectidas no ajuste de ganhos, o qual se realiza segundo períodos mais curtos e com a contribuição de um agente de detecção endógeno à “wave farm”.

Palavras chave: Conversor de energia de ondas, controlador PD, máquinas de vectores suporte, rede neuronal de modelo inverso, sistemas multi-agente

Índice

1	Introdução.....	1
1.1	Objectivo.....	1
1.2	Estrutura da dissertação.....	1
1.3	Contexto histórico e situação energética actual.....	2
1.4	Energia das ondas.....	4
1.4.1	Tecnologias de extracção de energia.....	6
2	Estado da arte.....	12
2.1	Redes neuronais.....	12
2.1.1	Modelo teórico.....	12
2.1.2	Arquitecturas de redes neuronais.....	15
2.1.3	Fase de treino.....	17
2.1.4	Modelos dinâmicos.....	19
2.2	Classificadores SVM.....	21
2.2.1	Introdução.....	21
2.2.2	Formulação teórica.....	22
2.2.3	Funções de núcleo.....	25
2.2.4	Sistemas de multi-classificação.....	27
2.2.5	Ferramentas computacionais do algoritmo.....	29
2.3	Sistemas multi-agente.....	31
2.3.1	O Conceito de Agente.....	31
2.3.2	Comunicação entre agentes.....	32
2.3.3	Gestão dos agentes.....	33
2.3.4	Modelação do comportamento racional.....	34
2.3.5	Plataformas multi-agente.....	36
3	Descrição do conversor de energia ISWEC.....	39
3.1	Modelo matemático.....	40
3.1.1	Dinâmica do flutuador.....	40
3.1.2	Dinâmica do sistema giroscópico.....	42
3.1.3	Modelo do conversor de energia.....	42
3.2	Controlador PD.....	44
3.3	Modelo de ondas.....	45
4	Controlador.....	47

4.1	Pesquisa do modelo dinâmico inverso	47
4.1.1	Importação de dados	48
4.1.2	Análise estatística	49
4.1.3	Pré-processamento de dados	52
4.1.4	Algoritmo de pesquisa	53
4.1.5	Treino da rede para um sinal composto	58
4.2	Classificador de onda	62
4.2.1	Extracção de características.....	62
4.2.2	Algoritmo de classificação.....	64
4.3	Implementação do sistema multi-agente.....	68
4.3.1	Visão geral.....	68
4.3.2	Modelação	72
4.3.3	Comunicação.....	75
4.3.4	Sincronização de processos	76
4.3.5	Simulação dinâmica	77
5	Conclusão	80
6	Referências.....	81

Lista de figuras

Figura 1 - Evolução do preço de barril de petróleo (USD)	3
Figura 2 - Evolução do consumo de electricidade de origem renovável (milhões Tep)	4
Figura 3 - Evolução de artigos publicados (de Sciencedirect)	5
Figura 4 - Distribuição mundial do nível de potência das ondas kW/m	6
Figura 5 - Nível de potência de ondas em função da profundidade	6
Figura 6 - Diagrama de tecnologias de conversão de energia das ondas	7
Figura 7 - Conversor de energia de ondas "Limpet"	8
Figura 8 - Conversor AWS	9
Figura 9 - Pato de Salter	10
Figura 10 - Conversor Pelamis	10
Figura 11 - Planta de conversor de energia Tapchan	11
Figura 12 - Vista de corte de Wavedragon	11
Figura 13 - Modelo não linear de um neurónio	12
Figura 14 - Funções de activação	14
Figura 15 - Rede de camada única	15
Figura 16 - Rede neuronal de múltiplas camadas	16
Figura 17 - Rede neuronal recorrente	16
Figura 18 - Aprendizagem supervisionada	17
Figura 19 - Nó j de uma rede neuronal MLP	18
Figura 20 - Modelo NARX	19
Figura 21 - Exemplo de conjunto de duas classes separáveis.....	22
Figura 22 - Caracterização do conceito de margem em classificadores SVM	23
Figura 23 - Exemplo de transformação de dados linearmente não separáveis.	25
Figura 24 - Fronteira de função de núcleo polinomial.....	26
Figura 25 - Fronteira de função de núcleo gaussiana.....	27
Figura 26 - Exemplo de método "one-vs-the-rest" aplicado a três classes.	28
Figura 27 - Exemplo de método "one-vs-one" aplicado a três classes.....	28
Figura 28- Diagrama agente-ambiente	31
Figura 29 - Arquitectura genérica de sistema multi-agente	34
Figura 30 - Sistema PRS (Procedural reasoning system)	36
Figura 31 - Interface gráfica RMA	38
Figura 32 - Ilustração ISWEC	39
Figura 33 - Sistema mecânico ISWEC	39
Figura 34 - Modelo do conversor ISWEC	43
Figura 35 - Diagrama de controlador PD	44
Figura 36 - Distribuição estatística da altura das ondas	45
Figura 37 - Espectrogramas do conjunto de ondas	46
Figura 38 - Diagrama de processo.....	48
Figura 39 - Função de autocorrelação do binário induzido	50
Figura 40 - Correlação cruzada entre δ e T_w	50
Figura 41 - Correlação cruzada entre ε e T_w	51
Figura 42 - Correlação cruzada entre termo do binário do PTO e T_w	51

Figura 43 - Fluxograma do algoritmo de pesquisa.....	53
Figura 44 - Diagrama de espaço de resultados.....	54
Figura 45 - Resposta no tempo do modelo inverso treinado para uma única onda de referência.....	57
Figura 46 - Sinais de saída (gráficos 1 a 3) do modelo directo	59
Figura 47 - Configuração da rede neuronal	59
Figura 48 - Resposta da rede neuronal ao sinal composto por oito ondas	60
Figura 49 - Estimativa da onda 1 gerada por rede treinada por um sinal composto	62
Figura 50 - Método de janela deslizante.....	63
Figura 51 - Técnica de validação cruzada	65
Figura 52 - Matriz de confusão SVM	67
Figura 53 - Representação do sistema multi-agente real	69
Figura 54 - Diagrama UML da arquitectura do JADE	70
Figura 55 - Diagrama de comunicação do sistema multi-agente.....	73
Figura 56 - Diagrama de comunicação em rede	75
Figura 57 - Diagrama de mensagens do sistema multi-agente.....	77
Figura 58 - Modelo do dispositivo classificador.....	78
Figura 59 - Potência absorvida durante simulação de ajuste dinâmico	79
Figura 60 - Parâmetros do sistema ISWEC	85
Figura 61 - Conjunto de ganhos óptimos para cada tipo de onda	85
Figura 62 - Diagrama UML do sistema multi-agente	86
Figura 63 - Interfaces gráficas do sistema multi-agente.....	87
Figura 64 - Resposta a ondas 1 e 2.....	88
Figura 65 - Resposta a ondas 3 e 4.....	88
Figura 66 - Resposta a ondas 5 e 6.....	88
Figura 67 - Resposta a ondas 7 e 8.....	89

Lista de tabelas

Tabela 1 - Funções de núcleo típicas	26
Tabela 2 - Actos comunicativos FIPA	33
Tabela 3 - Exemplo de plataformas multi-agente	37
Tabela 4 - Parâmetros do modelo aproximado de forças de radiação	41
Tabela 6 - Características das ondas marítimas	46
Tabela 7 - Estatística de dados de treino	52
Tabela 8 - Parâmetros do algoritmo de pesquisa	54
Tabela 9 - Subconjunto de espaço de resultados	55
Tabela 10 - Configurações seleccionadas	56
Tabela 11 - VAF do modelo inverso	57
Tabela 12 - Qualidade da extrapolação das 8 ondas	61
Tabela 13 - Conjuntos de características propostos	66
Tabela 14 - Resultados de classificação	67
Tabela 15 - Desempenho do controlador sob configurações estática e dinâmica	78

Notação

Acrónimos

FIFO	First-in First-out
ISWEC	Inertial Sea Wave Energy Converter
MSE	Mean squared error
NARX	Nonlinear autoregressive network with exogenous inputs
NN	Neural network
PD	Proportional-derivative control
PTO	Power Take-off
SVM	Support vector machine
TCP-IP	Transmission control protocol - Internet protocol
VAF	Variance accounted for
WEC	Wave energy converter

1 Introdução

1.1 Objectivo

A consciência crescente da necessidade de fontes de energias renováveis tem-se revelado uma forte impulsionadora do aproveitamento de um recurso com elevado potencial energético: o oceano. A criação de parques de ondas (“*wave farms*”), constituídos por vários conversores de energia interligados (*grid*), poderá representar uma alternativa à utilização de combustíveis fósseis na produção de electricidade. O presente trabalho tem como objectivo estudar a aplicação de técnicas de aprendizagem automática (“*machine learning*”) em conversores de energia das ondas, do tipo ISWEC, suportados por uma arquitectura multi-agente. Pretende-se assim avaliar o contributo destas tecnologias na melhoria da adaptabilidade do controlador, esperando-se, assim, autonomizar a operação do dispositivo e, simultaneamente, melhorar o seu desempenho na produção de electricidade.

1.2 Estrutura da dissertação

O presente trabalho encontra-se dividido em cinco capítulos de acordo com a seguinte estrutura:

Capítulo 1: Introdução

Este capítulo é responsável pela introdução às tecnologias de conversão de energia das ondas e à problemática associada à utilização de fontes de energia não renováveis.

Capítulo 2: Estado da arte

Este capítulo irá introduzir um conjunto de métodos no domínio da aprendizagem automática e dos sistemas multi-agente.

Capítulo 3: Descrição do conversor ISWEC

Este capítulo é dedicado à descrição das características do dispositivo ISWEC e do seu controlador.

Capítulo 4: Controlador

Este capítulo descreve o desenvolvimento de um controlador adaptativo, aplicável ao dispositivo ISWEC, com recurso a ferramentas de aprendizagem automática e de sistemas multi-agente.

Capítulo 5: Resultados

Este capítulo é dedicado à avaliação do desempenho do controlador desenvolvido.

1.3 Contexto histórico e situação energética actual

O modelo de desenvolvimento das sociedades ocidentais foi responsável por incrementos notáveis na produtividade e riqueza dos seus países, reflectindo-se na melhoria generalizada do padrão de vida das suas populações [1]. Este período de prosperidade, iniciado com a revolução industrial, é caracterizado, além da alteração do paradigma de produção, por uma intensa utilização de combustíveis fósseis como fonte energética em processos industriais, nos meios de transporte e na produção de electricidade. Contudo, a elevada dependência de combustíveis fósseis apresenta sérios riscos a longo prazo na medida em que estes recursos são finitos e a sua exploração apresenta externalidades negativas, nomeadamente nas alterações climáticas que induz.

Será a década de 70 a marcar a alteração da perspectiva das nações desenvolvidas em relação à sua dependência de combustíveis fósseis. A Figura 1 apresenta a evolução do preço de barril de petróleo, em dólares, a preços constantes de 2016. Os anos de 1973-1974 e 1979-1980 assinalam períodos em que a procura internacional de petróleo excedeu largamente a oferta, fenómeno designado por choque petrolífero, tendo resultado numa rápida subida de preços de, aproximadamente, 400% e 100% relativamente ao ano anterior.

O choque petrolífero de 1973 teve como causa a decisão da OPEP, num gesto de retaliação aos Estados Unidos da América pelo seu apoio ao Estado de Israel, em boicotar as exportações de petróleo para os E.U.A, Holanda, Portugal e África do Sul [2]. O segundo choque petrolífero, por sua vez, foi motivado pela revolução Iraniana, a qual conduziu a uma redução acentuada da extracção petrolífera. Sendo o petróleo um recurso tão importante nos países desenvolvidos, é evidente que uma elevada volatilidade de preços é um factor indesejável e, inclusive, perturbador do bom funcionamento das economias modernas.

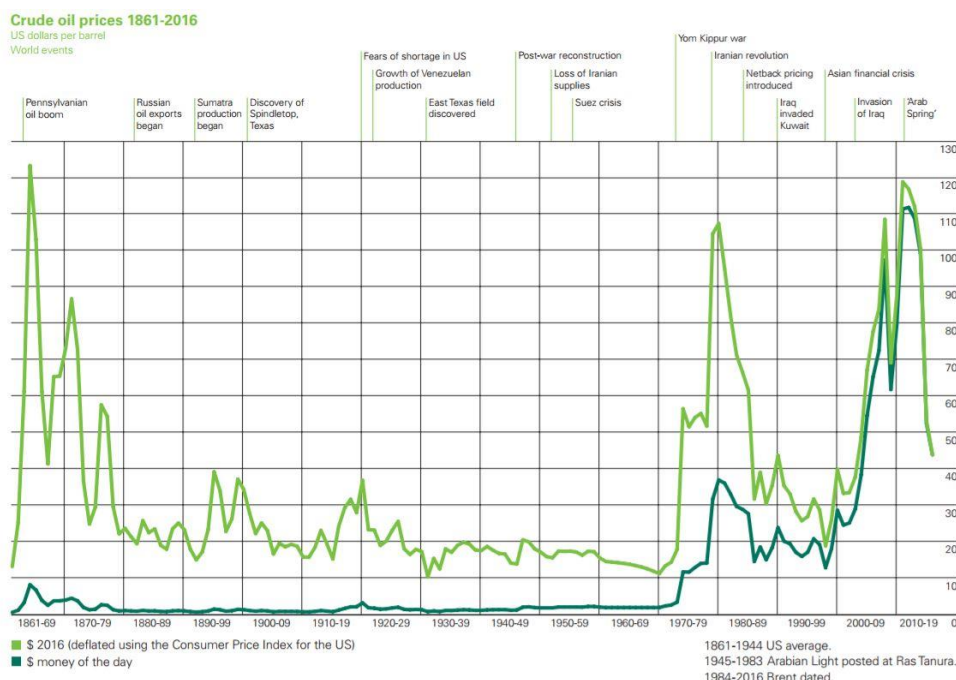


Figura 1 - Evolução do preço de barril de petróleo (USD) [3]

Deste modo é compreensível a razão pela qual a década de 70 propiciou a investigação de fontes de energia alternativas, procurando-se novas formas de produzir electricidade com recurso a energias renováveis. De facto, no início da crise petrolífera o Reino Unido implementou um programa, apoiado por fundos públicos, de investigação sobre o aproveitamento da energia das ondas do mar, procurando preparar-se para crises de abastecimento futuras. O programa foi terminado em 1982, tendo-se concluído que a tecnologia não seria viável no futuro próximo [4].

As circunstâncias actuais são distintas das verificadas na década de 70, vive-se um período de petróleo barato. Todavia, as questões associadas à problemática ambiental continuam a ser de enorme relevância, motivando as várias potências mundiais a assumirem compromissos de redução de emissões poluentes. O desafio é assinalável. Enquanto o mundo desenvolvido tem hoje a tecnologia e condições sociais para empreender um programa de redução de emissões a esta escala, o mesmo não ocorre nos países em desenvolvimento. Mais, os países em desenvolvimento apresentam, em média, taxas de crescimento económico superiores às verificadas nos países desenvolvidos, o que implicará uma crescente procura de energia, estimando-se que em 2040, mais de 60% do aumento no consumo de energia virá de países asiáticos não pertencentes à OCDE, incluindo a China e a Índia [5].

A aposta nas energias renováveis é hoje consensualmente aceite como uma chave para um desenvolvimento sustentável, seja ao nível ambiental como ao nível económico. A Figura 2 apresenta a

evolução, num período de 20 anos, do consumo mundial de electricidade de origem renovável, sendo evidente o seu crescimento exponencial na maioria das regiões do mundo.

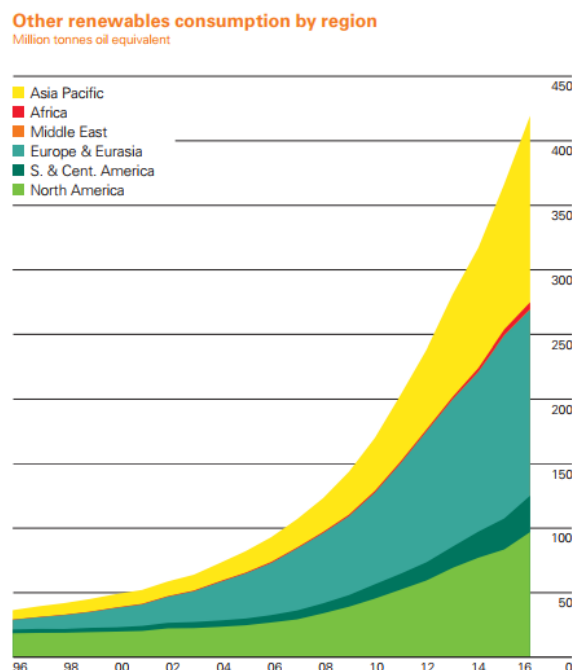


Figura 2 - Evolução do consumo de electricidade de origem renovável (milhões Tep) [3]

1.4 Energia das ondas

Os oceanos, cuja área corresponde a aproximadamente 70% da superfície do planeta, são uma potencial fonte de energia renovável, designadamente pelas energias potencial e cinética das suas ondas. De acordo com a agência internacional de energia (IEA), o potencial energético global nas ondas do mar é estimado em 2 TW, antevendo-se que possa atingir uma quota de 10% no fornecimento mundial de electricidade [6, p. 20].

O interesse por tecnologias de conversão de energia de ondas marítimas (WEC), não sendo um fenómeno recente, tem hoje um papel relevante na diversificação de fontes energéticas das nações desenvolvidas. Neste âmbito, importa salientar a valiosa contribuição de Yoshio Masuda (1925-2009) no desenvolvimento do tema desde a década de 40, tendo sido responsável pelo desenvolvimento de um conversor de energia (WEC) cujo funcionamento se baseava no princípio de coluna de água oscilante (OWC) [7, p. 3]. Num período mais recente, é igualmente relevante a investigação conduzida por Stephen Salter no desenvolvimento de conversores de energia de ondas, resultando na publicação de um artigo em 1974

que viria a apresentar à comunidade científica o “pato de Salter”, um dispositivo que ainda hoje é uma referência [7]. A Figura 3, cujos dados foram extraídos da plataforma *Sciencedirect*, apresenta a evolução, nos últimos oito anos, de artigos publicados sobre o tema de conversão de energia das ondas, sendo notória a tendência no crescimento das publicações.

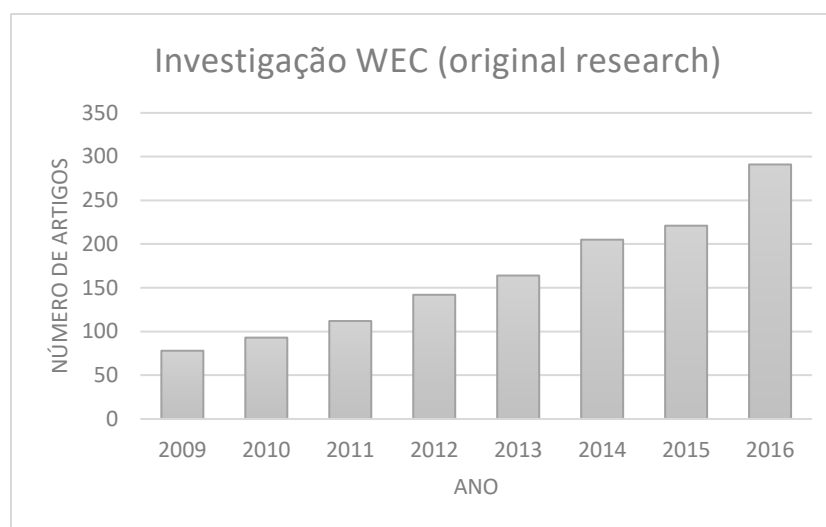


Figura 3 - Evolução de artigos publicados (de Sciencedirect) [8]

A Figura 4 apresenta a distribuição mundial do nível de potência das ondas em valores médios anuais, relativo a zonas de águas profundas. É possível concluir que países como Portugal, Reino Unido e Noruega possuem um elevado potencial de exploração. O nível de potência (kW/m), medido por frente de onda, é dependente da profundidade do mar, atingindo-se os maiores valores de potência a profundidades mais elevadas. A Figura 5 indica, a título ilustrativo, os níveis de potência estimados num arquipélago ao largo da costa oeste da Escócia, em função da profundidade de exploração.

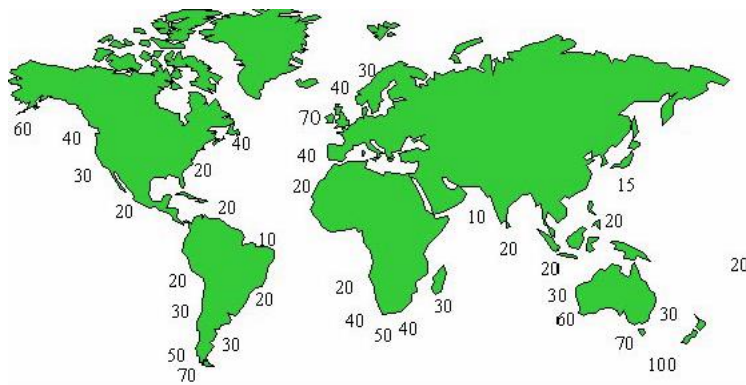


Figura 4 - Distribuição mundial do nível de potência das ondas kW/m [9]

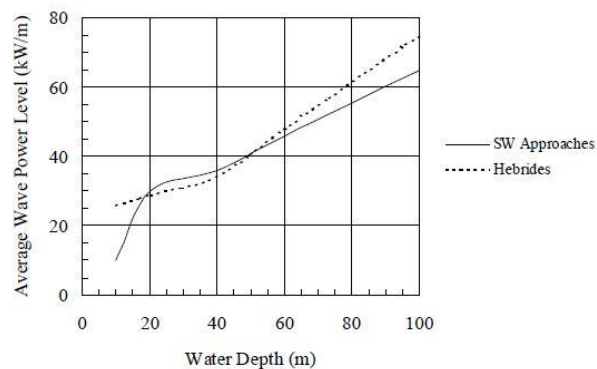


Figura 5 - Nível de potência de ondas em função da profundidade [6, p. 21]

1.4.1 Tecnologias de extração de energia

O desenvolvimento de conversores de energia das ondas é geralmente um processo moroso, resultado da complexidade dos fenómenos hidrodinâmicos e da fase de ensaio experimental. Em termos genéricos, um conversor de energia das ondas é um dispositivo que aproveita o movimento relativo da superfície livre de água e rectifica esse movimento oscilatório de forma a accionar uma máquina eléctrica (PTO) e assim produzir electricidade [4].

Tendo por base estudos científicos, concluiu-se que um conversor de energia de ondas, em benefício da eficiência, deverá apresentar uma frequência natural semelhante à frequência da onda incidente, operando em condições próximas da ressonância, o que poderá ser realizado actuando directamente no PTO, uma estratégia designada por controlo de fase [7, p. 7]. Actualmente, existem várias tecnologias de conversão

de energia das ondas, dividindo-se, quanto à localização de exploração, em dispositivos costeiros (*onshore*), dispositivos perto da costa (*nearshore*) e dispositivos distantes da costa (*offshore*).

O diagrama ilustrado na Figura 6 apresenta uma síntese das tecnologias utilizadas na conversão de energia de ondas, incluindo a designação, não exaustiva, de projectos desenvolvidos. A próxima secção irá abordar os princípios de funcionamento dos dispositivos de Coluna de água oscilante (OWC), corpos oscilantes e de galgamento.

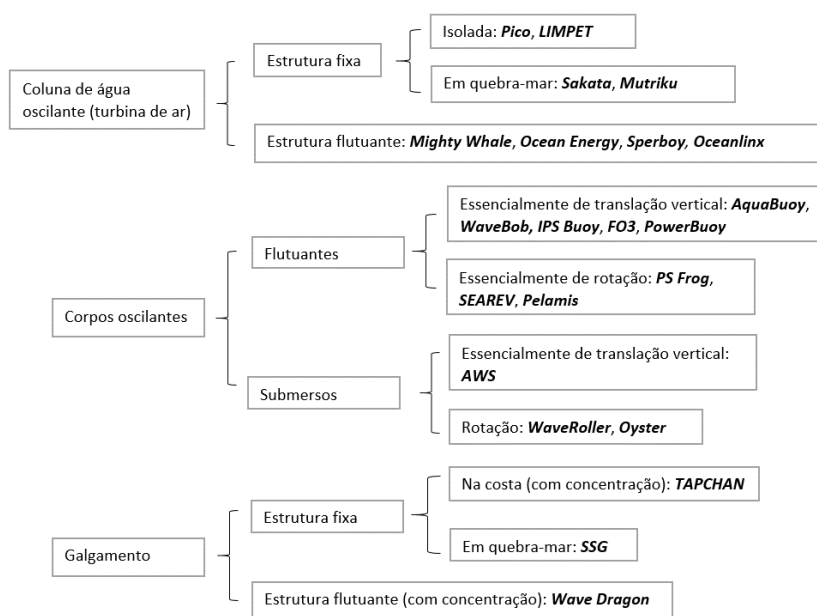


Figura 6 - Diagrama de tecnologias de conversão de energia das ondas, adaptado de [7]

Relativamente às tecnologias de conversão de energia das ondas, importa ressaltar algumas características que se terá de ter em consideração relativamente à segurança e manutenção. O aspecto da segurança é fundamental, dado que um aparelho conversor de energia de ondas será sujeito a forças de elevada magnitude, podendo as mesmas conduzir à destruição do dispositivo. A manutenção do dispositivo conversor de energia exigirá um cuidado planeamento pelo ambiente corrosivo e pela localização, sendo evidente que localizações distantes da costa (*“offshore”*) oferecerão dificuldades logísticas acrescidas. Relativamente à produção de electricidade, importará notar que a energia das ondas, à semelhança de outras energias renováveis, apresenta períodos de intermitência, sendo necessária uma gestão criteriosa da rede de modo a satisfazer a procura de energia.

1.4.1.1 Tipologia de coluna de água oscilante

O princípio de funcionamento de uma coluna de água oscilante (OWC) na extracção de energia das ondas baseia-se no movimento relativo da superfície livre de água no interior de uma estrutura oca, assemelhando-se ao movimento de um êmbolo no interior de um cilindro, provocando ciclos alternados de compressão e descompressão de uma coluna de ar. A existência de um canal de comunicação da coluna de ar com a atmosfera permite que, com recurso a um conjunto turbina e PTO, seja possível extrair energia da onda marítima, convertendo-a em electricidade. A selecção de uma turbina de tipo *Wells* permite que o sentido da rotação do gerador não se altere, independentemente do sentido do fluxo de ar no canal, rectificando assim o comportamento oscilatório da coluna de ar de forma bastante elegante e, acima de tudo, simples. A tipologia é aplicada em estruturas flutuantes e fixas.

A Figura 7 ilustra a vista de corte de um conversor de energia de coluna de água oscilante, de designação “Limpet”, realizado numa estrutura fixa de betão. Dados relativos à ilha de Islay (Escócia), indicam que este equipamento deverá atingir uma potência eléctrica média de 206 kW, produzindo anualmente aproximadamente 1800 MWh [6, p. 32]. No panorama nacional, importa destacar o contributo da central de energia das ondas do Cachorro — localizada na ilha do Pico, Açores — partilhando o mesmo princípio de funcionamento (OWC). A central, concluída em 1999, cessou a sua operação a 17 de Abril de 2018 devido à existência de defeitos estruturais que culminaram no desmoronamento parcial da central [10].

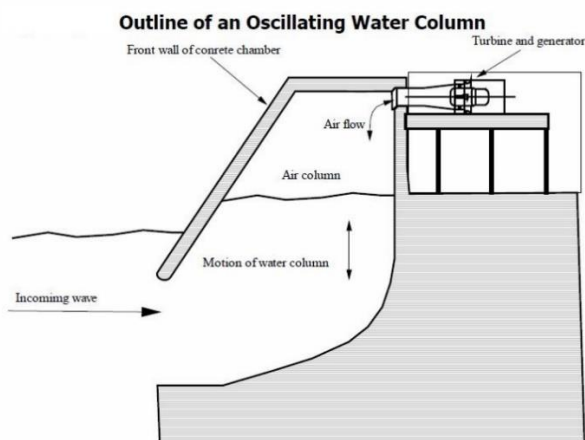


Figura 7 - Conversor de energia de ondas "Limpet" [6]

1.4.1.2 Tipologia de sistemas de corpos oscilantes

A tipologia de corpos oscilantes designa um tipo de conversor de energia de ondas especialmente adequado para aplicação em alto-mar (*offshore*), aproveitando o elevado potencial energético verificado em águas profundas [7, p. 16]. Na configuração do sistema de corpos oscilantes, a extracção de energia ocorre na sequência da reacção do conversor em relação ao fundo do mar, pontão ou a outro corpo oscilante [7, p. 16]. O mesmo autor conclui que os conversores de energia desta tipologia não usufruíram de um desenvolvimento especialmente lesto, face à necessidade de longos cabos submarinos para distribuição de electricidade e de complexos sistemas de amarração.

Um primeiro exemplo de conversor de corpo oscilante é o AWS (*Archimedes Wave Swing, AWS Ocean energy*), cuja ilustração se encontra na Figura 8. Este conversor encontra-se totalmente submerso, sendo constituído por duas partes: flutuador e base. Baseando-se no princípio de Arquimedes, o flutuador irá descrever um movimento oscilatório vertical resultante da variação da pressão hidrostática. A base, que se encontra ancorada no fundo do mar, inclui um gerador de electricidade do tipo linear acoplado ao flutuador. Este é o primeiro conversor de energia das ondas a optar por um gerador linear [7, p. 21].

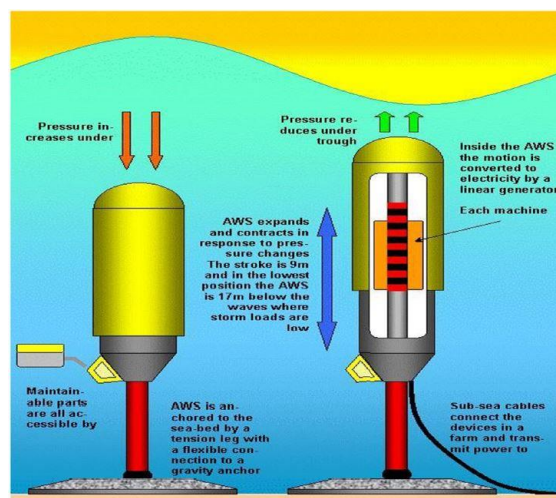


Figura 8 - Conversor AWS [11]

Os conversores de corpos oscilantes não estão restritos a um único tipo de movimento relativo, existindo também exemplos de conversores cuja extracção de energia ocorre no movimento relativo de rotação, sendo um exemplo o já mencionado “pato de Salter” (ver Figura 9), um conversor de eixo excêntrico desenvolvido por Stephen Salter nas décadas de 70 e 80, sendo provavelmente o conversor com melhor desempenho criado à sua época, ainda que nunca tenha alcançado a fase de exploração plena em ambiente real [7, p. 23].

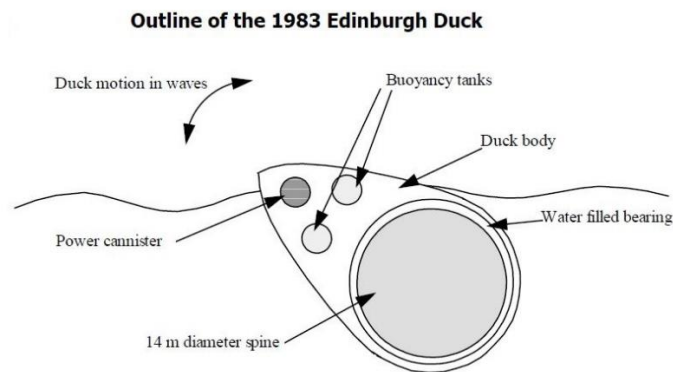


Figura 9 - Pato de Salter [6, p. 72]

O conversor de energia *Pelamis* (ver Figura 10) é constituído por quatro elementos cilíndricos acoplados por juntas flexíveis, cujo movimento relativo acciona um circuito hidráulico que irá por sua vez transmitir a energia mecânica a um gerador eléctrico, existente em cada elemento. O projecto, desenvolvido no Reino Unido, foi objecto de uma intensa investigação, incluindo o ensaio de protótipos em tamanho real com 750 kW de potência e a implementação do primeiro parque de conversores de energia de ondas (*wave farm*) conectado à rede de distribuição nacional [7, p. 24].



Figura 10 - Conversor Pelamis [7, p. 24]

1.4.1.3 Tipologia de conversores por galgamento

O princípio de funcionamento de um conversor de energia de ondas por galgamento (*overtopping*) consiste na captura de água do mar e na sua introdução num reservatório de grande dimensão. A água no reservatório, armazenada a um nível superior ao do mar envolvente, será escoada através de um canal, transferindo a sua energia a uma turbina, accionando por fim um gerador eléctrico [7, p. 32]. O princípio de funcionamento é similar ao verificado numa barragem hidroeléctrica, existindo naturalmente diferenças ao nível da cota máxima do reservatório, que no conversor é menor. A Figura 11 apresenta a planta do conversor Tapchan, de estrutura fixa, desenvolvido na Noruega na década de 80. Nesta figura é possível observar, entre outros elementos, uma rampa¹ (*converter*) de dimensão significativa face à área do reservatório. Esta rampa tem a forma de um canal que se estreita gradualmente, amplificando assim a altura da onda até que o reservatório seja preenchido [7, p. 32].

Os conversores de energia de ondas por galgamento podem apresentar uma estrutura flutuante, sendo exemplo o projecto Wave Dragon (ver Figura 12), promovendo a flexibilidade na localização do conversor.

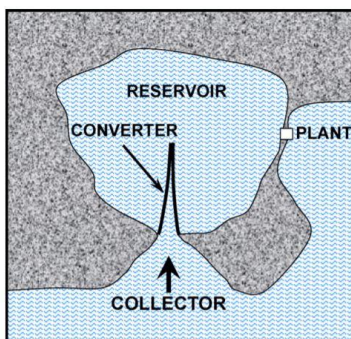


Figura 11 - Planta de conversor de energia Tapchan [12, p. 13]

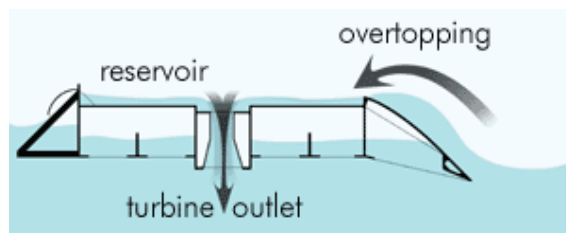


Figura 12 - Vista de corte de Wavedragon [13]

¹ Optou-se por traduzir o termo inglês original "*converter*" para "rampa", evitando a confusão com a designação de aparelho conversor de energia de ondas (WEC)

2 Estado da arte

2.1 Redes neurais

A inteligência artificial é um tema que tem suscitado um elevado interesse da comunidade científica, nomeadamente, na forma como se poderá dotar um sistema computacional de uma capacidade de raciocínio e deliberação análogas à de um ser humano. A área da aprendizagem automática (“*machine learning*”) compreende o estudo de sistemas computacionais cujo comportamento seja suportado por uma capacidade de aprendizagem, traduzindo-se, assim, numa forma de inteligência. Na segunda metade do século XX observou-se o desenvolvimento de vários sistemas computacionais inteligentes, tais como: Redes neurais, controladores de lógica “*Fuzzy*” e algoritmos genéticos.

2.1.1 Modelo teórico

A rede neuronal é um sistema que procura mimetizar o comportamento de um cérebro humano, seja na sua inteligência natural, seja na sua plasticidade, isto é, na capacidade de se adaptar a novas condições com base na sua experiência. A organização do cérebro humano em várias unidades de processamento — os neurónios — que comunicam entre si segundo uma arquitectura largamente distribuída e paralela, permite-lhe apresentar uma elevada adaptabilidade e rapidez na realização de tarefas complexas tais como a detecção de padrões, inferência lógica e controlo motor [14].

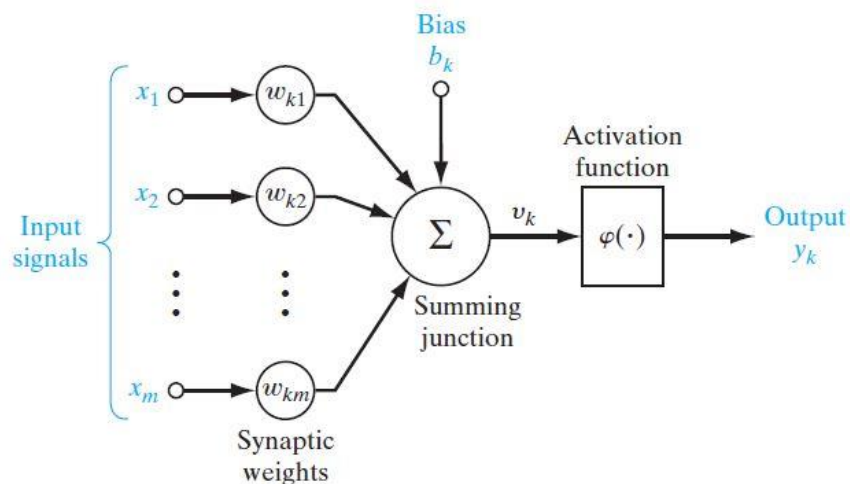


Figura 13 - Modelo não linear de um neurónio [14, p. 11]

A rede neuronal, inspirada no exemplo biológico, designa um sistema computacional constituído por vários processadores em paralelo, cujas conexões são reforçadas — mecanismo em que se impõe uma preferência através de pesos sinápticos — de acordo com a sua importância na resolução do problema, sendo esta tarefa realizada com recurso a um algoritmo de aprendizagem [14]. No desenvolvimento de redes neuronais, importa destacar os contributos pioneiros de: McCulloch e Pitts, Hebb e Rosenblatt [14, p. 47]. A Rosenblatt (1958) é atribuído o contributo do desenvolvimento do primeiro modelo de aprendizagem supervisionada — o Perceptrão —, constituindo-se a primeira rede neuronal descrita de forma algorítmica [14, pp. 47,48]. Ainda segundo [14], considera-se que a configuração mais simples de rede neuronal apta à resolução de problemas de classificação linearmente separáveis é o Perceptrão.

A Figura 13 apresenta um modelo não linear de um neurónio k . Neste modelo é possível identificar um conjunto de portas de entrada de sinais, um somador e uma função de activação. Em cada porta j existe um ramo de transmissão — designado por sinapse — de um sinal x_j em que lhe é atribuído um determinado peso numérico w_{kj} . O somador tem como função retornar a soma dos valores de entrada multiplicados pelos respectivos pesos. A função de activação tem como função garantir que o sinal proveniente do somador se encontra limitado a um intervalo. O modelo possui também um parâmetro b_k exterior ao somador, tendo como função o ajuste de tendência (“*bias*”) do sinal de entrada da função de activação [14, p. 11].

As expressões (2.1) a (2.5) resumizam a descrição matemática do modelo de neurónio em [14]. A equação (2.1) traduz a soma dos sinais x_j multiplicados pelos pesos de cada sinapse. A equação (2.2) define a existência de um parâmetro de “*bias*”. A equação (2.3) define o sinal de entrada da função de activação.

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (2.1)$$

$$y_k = \varphi(u_k + b_k) \quad (2.2)$$

$$v_k = u_k + b_k \quad (2.3)$$

A expressão (2.4) evidencia o efeito da função de activação $\varphi(v)$ no sinal de saída y_k da rede neuronal. Relativamente aos tipos de funções de activação tipicamente utilizadas em redes neuronais, destacam-se os tipos: Função limiar (“*threshold*”) ou de *Heaviside*; função sigmóide [14]. A Figura 14 apresenta os dois tipos de funções de activação.

$$y_k = \varphi(v_k) \quad (2.4)$$

A função de “*Threshold*”, descrita pela expressão (2.5), limita a saída da rede neuronal a uma resposta binária, tal como indicado em (2.6). A expressão (2.7) representa o campo localizado induzido do neurónio, incluindo o efeito do parâmetro de “*bias*” [14]. Este comportamento de um modelo de neurónio com uma resposta binária é encontrado no modelo *McCulloch-Pitts* (1943), constituindo-se o primeiro modelo de rede neuronal artificial [14, p. 14].

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (2.5)$$

$$y_k = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0 \end{cases} \quad (2.6)$$

$$v_k = \sum_{j=1}^m w_{kj} \cdot x_j + b_k \quad (2.7)$$

A função de activação sigmóide, indicada em (2.8), é uma função não linear largamente utilizada em redes neuronais, exibindo, ao contrário da função “*Threshold*”, uma condição de diferenciabilidade e transições suaves entre regiões lineares e não lineares [14, p. 14]. O parâmetro a permite ajustar o declive das curvas da função sigmóide, aproximando-se da função “*Threshold*” para valores elevados do parâmetro a .

$$\varphi(v) = \frac{1}{1 + \exp(-a \cdot v)} \quad (2.8)$$

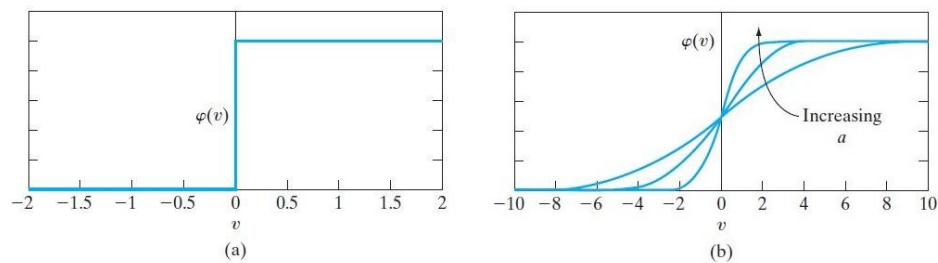


Figura 14 - Funções de activação: a) função limiar; b) função sigmóide [14, p. 13]

2.1.2 Arquitecturas de redes neuronais

A configuração de uma rede neuronal, isto é, o modo como os neurónios artificiais comunicam entre si, é um factor especialmente relevante na capacidade de aprendizagem da rede [14, p. 21]. As configurações de redes neuronais dividem-se fundamentalmente em:

- Redes de camada e sentido únicos (*"single-layer feedforward networks"*)
- Redes de múltiplas camadas de sentido único (*"multilayer feedforward networks"*)
- Redes recorrentes (*"recurrent networks"*)

A configuração em camada única, ilustrada na Figura 15, designa uma rede em que existe apenas uma camada de computação, a qual é constituída por neurónios alimentados directamente pela camada de entrada. É possível observar que nesta configuração, as duas camadas possuem conexões de apenas um sentido — desde a camada de entrada até à camada de saída, isto é, da esquerda para a direita — razão pela qual se denominam por redes *"feedforward"*. Existem, contudo, redes em que se verifica uma realimentação (*"feedback"*) da saída dos neurónios, seja para a própria entrada do neurónio, seja para a entrada de outro neurónio. Esta é uma característica de redes neuronais recorrentes.

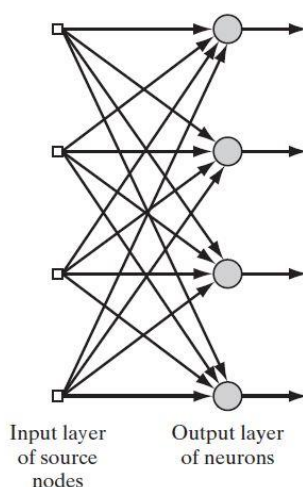


Figura 15 - Rede de camada única [14, p. 21]

A configuração em múltiplas camadas de sentido único designa uma rede neuronal, sem realimentação, em que existe, pelo menos, uma camada de neurónios oculta. A camada de neurónios diz-se oculta quando se localiza entre as camadas de entrada e de saída. A Figura 16 ilustra uma rede com uma camada oculta em que todos os neurónios de cada camada se encontram conectados aos nós das camadas adjacentes, tomando a designação de rede totalmente conectada.

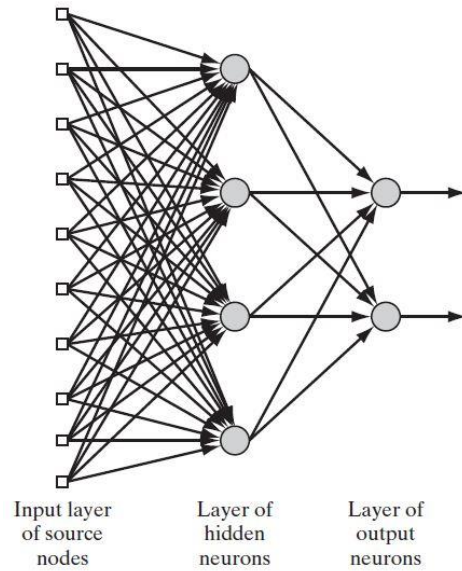


Figura 16 - Rede neuronal de múltiplas camadas [14, p. 22]

Por último, a rede neuronal recorrente designa uma configuração em que os neurónios comunicam com camadas anteriores através de ligações de realimentação. Esta configuração, ilustrada na Figura 17, apresenta um impacto significativo na capacidade de aprendizagem e desempenho da rede [14, p. 23]. Tal como indicado na figura, os ramos de realimentação possuem elementos de atraso unitários, resultando num comportamento que será especialmente útil na resolução de problemas dinâmicos dependentes de estados passados.

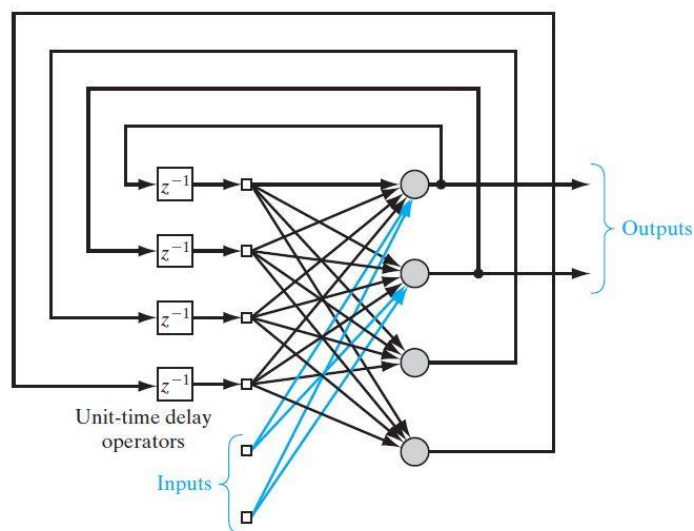


Figura 17 - Rede neuronal recorrente [14, p. 23]

2.1.3 Fase de treino

A aprendizagem de uma rede neuronal pode ser classificada como supervisionada ou não supervisionada. A aprendizagem supervisionada, utilizada ao longo deste estudo, constitui um método em que são fornecidas à rede as observações que se pretendem obter, permitindo que a rede minimize o erro de extrapolação no mapeamento entre os sinais de entrada e saída. A Figura 18 ilustra o método de aprendizagem supervisionada, evidenciando o carácter adaptativo da rede neuronal face ao desvio entre a resposta da rede e a resposta desejada fornecida pelo agente tutor.

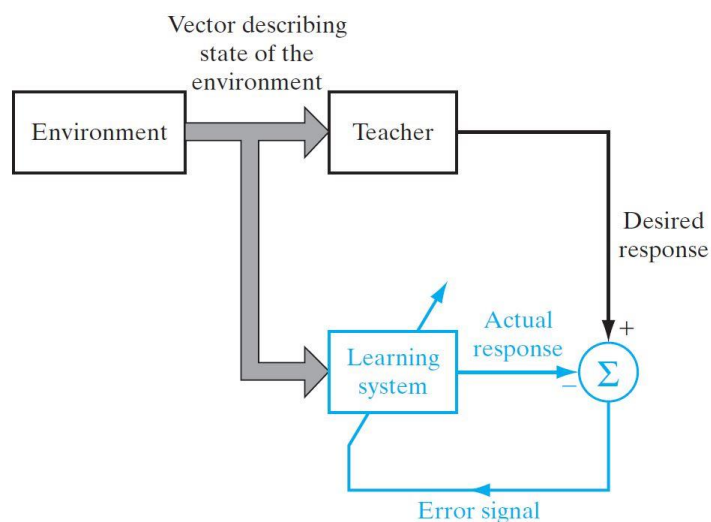


Figura 18 - Aprendizagem supervisionada [14, p. 35]

O treino de uma rede neuronal designa a operação de ajuste dos pesos sinápticos de cada neurónio atendendo ao cumprimento de determinado critério de desempenho. No caso em particular, assume-se que a rede neuronal é do tipo Perceptrão de múltiplas camadas (MLP- "*Multi-layer perceptron*"), uma rede que possibilita a resolução de problemas de classificação e de regressão. A rede do tipo MLP é caracterizada pelo elevado nível de conectividade, e existência, de pelo menos, uma camada escondida e pela utilização de funções de activação não lineares e diferenciáveis [14, p. 122]. No problema de classificação o objectivo é determinar a classe de cada observação a partir de um conjunto de características. No problema de regressão, por seu lado, procura-se um mapeamento entre conjuntos de entrada e saída, similar a um ajuste de curvas. A rede do tipo MLP é capaz de aproximar, de forma bastante satisfatória, uma qualquer função contínua apenas com uma camada oculta, desde que treinada com o algoritmo "*back-propagation*" [15, p. 238]. O treino computacionalmente eficiente de redes do tipo MLP foi conseguido com a publicação de um algoritmo em 1986 por Rumelhart et.al. [16], denominado por "*back-propagation*" [14, p. 124]. O algoritmo desenvolve-se em duas fases: fase em sentido directo ("*forward*

phase”) e fase em sentido inverso (“*backward phase*”). Na fase de sentido directo, o sinal de entrada é propagado ao longo da rede neuronal, atingindo a camada de saída da rede. Na fase de sentido inverso são calculados os erros entre o sinal de saída e o sinal de referência, propagando-se os erros pelas camadas anteriores, ajustando-se os pesos sinápticos de cada neurónio [14, p. 124].

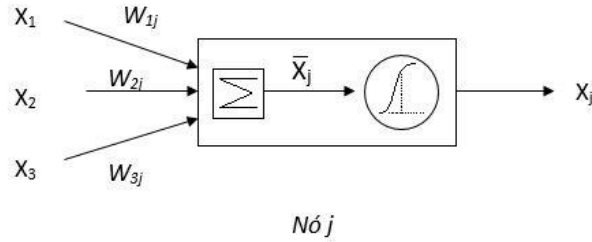


Figura 19 - Nó j de uma rede neuronal MLP. Adaptado de [15, p. 235]

A título ilustrativo, optou-se por demonstrar, de acordo com [15], o procedimento “*back-propagation*” aplicado a uma rede MLP em que se utilizam funções de activação do tipo sigmóide. Na Figura 19 encontra-se representado um modelo de neurónio de uma rede MLP, incluindo: uma camada de entrada x_i , pesos sinápticos w_{ij} , somador e função de activação. A expressão (2.9) traduz o sinal de saída do somador, pesando cada sinal de entrada, e reflectindo também a presença de um valor de “bias” w_j . A saída do nó, resultado da função de activação, é traduzida pela expressão (2.10)

$$\bar{x}_j = \sum_i w_{ij} \cdot x_i + w_j \quad (2.9)$$

$$x_j = f(\bar{x}_j) = \frac{1}{1 + \exp(-\bar{x}_j)} \quad (2.10)$$

A expressão (2.11) introduz uma métrica de avaliação do erro de extrapolação da rede. Os parâmetros d_k e x_k designam, respectivamente, o valor de referência do nó “ k ” e o valor obtido no nó “ k ” correspondentes ao par de observações “ p ”. O par “ p ” inclui dados de treino, isto é, os valores de entrada e de saída, seguindo um modelo de aprendizagem supervisionada.

$$E_p = \sum_k (d_k - x_k)^2 \quad (2.11)$$

Os modos “on-line” e “off-line” apresentam características diferentes no âmbito do treino de redes MLP segundo um modelo de aprendizagem supervisionada. O modo “on-line” caracteriza-se pela actualização dos pesos sinápticos a cada observação apresentada à rede neuronal, sendo um método especialmente utilizado em tarefas de detecção de padrões de elevada complexidade [14, p. 129]. O modo “off-line”, também designado por “batch learning”, preconiza uma actualização dos pesos sinápticos com base num conjunto de observações (“batch”), constituindo-se um “epoch” de treino [14, p. 127]. Este modo, ainda que especialmente exigente quanto aos requisitos de armazenamento, tem como vantagem permitir a paralelização do processo, agilizando a aprendizagem, e a obtenção de uma estimativa precisa do vector gradiente, o que irá contribuir para a convergência do método para um mínimo local [14]. O modo “off-line” é especialmente adequado à resolução de problemas de regressão não lineares [14, p. 128].

2.1.4 Modelos dinâmicos

Esta secção cumpre o objectivo de introduzir uma nova classe de rede: a rede neuronal dinâmica de tipo recorrente. Esta classe de rede é caracterizada pela existência de uma dependência temporal da resposta face ao sinal de estímulo, tornando-a uma solução desejável em diversas aplicações, tais como: identificação e previsão de sistemas não lineares, reconhecimento de voz e controlo de sistemas [14, p. 791].

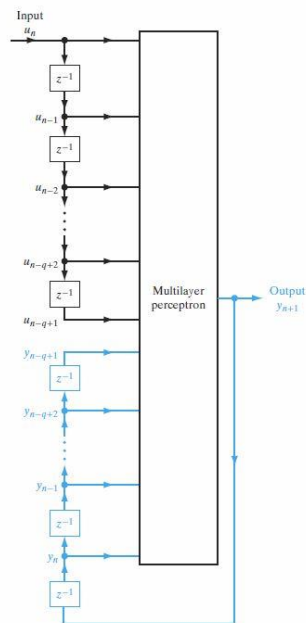


Figura 20 - Modelo NARX [14, p. 792]

A Figura 20 ilustra a arquitectura de um modelo de rede neuronal dinâmica designado por NARX (*“Nonlinear autoregressive with exogenous inputs”*). Na figura, é possível observar que o sinal de saída da rede MLP é realimentado no sinal de entrada, existindo linhas de atraso — memórias de q posições — em ambos os sinais. O comportamento dinâmico da rede é traduzido pela expressão (2.12). Na expressão, o sinal de entrada — também designado por sinal exógeno — é representado por u_n , para o valor actual, correspondendo u_{n-q+1} à última posição na memória. O sinal de saída da rede é dado por y_{n+1} .

$$y_{n+1} = F(y_n, \dots, y_{n-q+1}; u_n, \dots, u_{n-q+1}) \quad (2.12)$$

Relativamente à aprendizagem de redes neuronais recorrentes, importará referir dois algoritmos: o BPTT e RTRL. O algoritmo BPTT (*“back-propagation through time”*) designa uma extensão ao domínio temporal do algoritmo *“back-propagation”* utilizado em redes de tipo MLP [14]. O algoritmo RTRL (*“real-time recurrent learning”*) caracteriza-se por ajustar os pesos sinápticos da rede recorrente em tempo real, não se interrompendo o processamento de sinal da rede [14, p. 812]. Ao nível do tempo de computação, o algoritmo BPTT apresenta um desempenho superior ao RTRL, devendo-se, contudo, notar que os requisitos de memória aumentam significativamente à medida que o conjunto de aprendizagem — sequência constituída por pares consecutivos, com coerência temporal, de sinais de entrada e saída — se amplia [14, p. 807]. O algoritmo RTRL é especialmente adequado ao treino *“on-line”*, apresentando o algoritmo BPTT uma melhor adequação para o treino *“off-line”* [14, p. 807].

Concluindo, uma rede dinâmica recorrente, como é o modelo NARX, permite processar sinais sequenciais [15], nomeadamente, séries temporais, inferindo a relação, no passado, entre os sinais de entrada e de saída que suporte a computação de respostas futuras (*“multi-step ahead prediction”*). A capacidade preditiva da rede neuronal NARX tem sido alvo de vários estudos, destacando, a título de exemplo, os seguintes: em [17] é proposto utilizar uma rede NARX na previsão do nível do aquífero MRVA (E.U.A.), tendo um horizonte de previsão de 3 meses e um registo histórico relativo a 8 anos, tendo-se obtido resultados satisfatórios; em [18] é proposto utilizar uma rede NARX na previsão da radiação solar e velocidade de vento, tendo como objectivo integrar a informação num sistema inteligente de distribuição de electricidade, utilizando-se um horizonte de previsão de até 24 horas. Esta funcionalidade, designadamente de regressão, será demonstrada neste trabalho, ilustrando-se a operação de identificação do modelo inverso do sistema ISWEC para um conjunto alargado de ondas marítimas.

2.2 Classificadores SVM

Nesta secção será realizada uma introdução ao algoritmo SVM aplicado a problemas de classificação, descrevendo-se os seus fundamentos teóricos, aplicações práticas e implementações computacionais.

2.2.1 Introdução

Um problema de classificação, no domínio da aprendizagem automática, descreve a necessidade de identificar num conjunto de dados qual a classe a que pertence uma nova observação. A resolução deste problema através de um algoritmo treinado para um subconjunto do espaço de dados original, no qual é conhecida a relação entre as variáveis explicativas e as etiquetas de classe — a formulação de um algoritmo de aprendizagem supervisionada — permite que se infira a classe de uma observação não incluída no treino, utilizando um mecanismo de generalização do conhecimento.

O algoritmo SVM (*“support vector machine”*) [19], desenvolvido por Vapnik e Cortes (1995), designa um método de aprendizagem automática supervisionada de classificação de dados, que realiza um mapeamento não linear entre o espaço de características — espaço em que cada observação é descrita pelas várias variáveis explicativas (*“features”*) — e um espaço de elevada dimensão — espaço alargado — permitindo determinar uma superfície de decisão linear que assegure uma elevada capacidade de generalização do classificador, mesmo nos casos em que os dados de treino não se apresentem linearmente separáveis [19, p. 274]. Um conjunto de dados é considerado não linearmente separável caso não seja possível separar completamente as suas classes através de uma única fronteira linear [20].

A Figura 21 apresenta um exemplo de classes separáveis num espaço bidimensional (duas variáveis explicativas), sendo visível que uma única recta separa completamente as observações de classes distintas. Nesta figura é evidente um conceito central no desenvolvimento do classificador SVM: a existência de uma margem óptima, em que um hiperplano separa as classes com a maior distância possível. Note-se que o conceito de hiperplano traduz uma fronteira linear num espaço de elevada dimensão, da mesma forma que uma fronteira linear num espaço tridimensional seria realizada por um plano [21, p. 3]. Na mesma figura é possível observar os vectores de suporte, os quais são constituídos pelas observações localizadas exactamente sobre as margens óptimas. É importante notar que na definição do hiperplano óptimo apenas é considerada uma pequena fracção dos dados de treino, uma fracção constituída pelos vectores de suporte [19, p. 275].

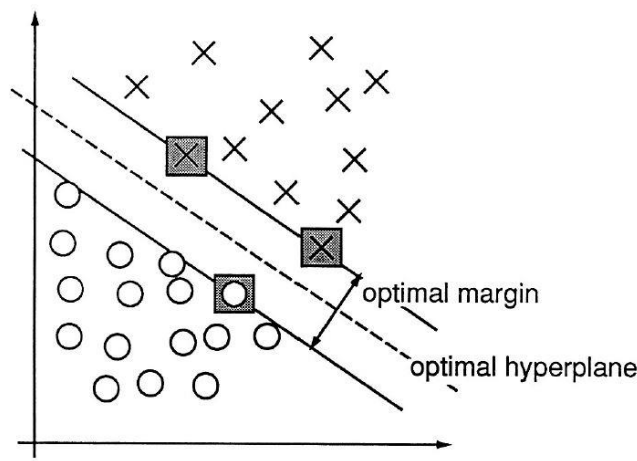


Figura 21 – Exemplo de conjunto de duas classes separáveis. Tracejado: hiperplano; Quadrado cinzento: vector de suporte [19, p. 275]

Importa ressaltar que a dimensão do conjunto de vectores de suporte é um factor que condiciona o desempenho do classificador, provando-se que um hiperplano construído com um elevado número de vectores de treino face ao número de vectores de suporte irá dotar o algoritmo de uma elevada capacidade de generalização, mesmo que o espaço alargado apresente uma dimensão infinita [19, p. 276]. De acordo com os autores de [19], o algoritmo SVM, reflectindo já a capacidade de classificar dados não linearmente separáveis — utilizando um mapeamento não linear, cujo esclarecimento se remete para a secção das funções de núcleo — apresenta-se, pelas suas qualidades de aproximação, como uma solução alternativa às redes neuronais [19, p. 276].

2.2.2 Formulação teórica

Nesta secção serão apresentadas as bases teóricas, adaptadas de [20, pp. 418-421], do algoritmo de classificação SVM genérico, introduzindo-se os conceitos de margem rígida e margem suave, pese embora que neste trabalho a sua utilização não seja explícita, na medida em que foram utilizadas soluções comerciais do algoritmo.

Assumindo que o conjunto de treino possui N vectores de entrada — as observações — que só poderão ser classificados como pertencendo a uma das duas classes disjuntas, designadamente a classe “-1” e “1”, tem-se $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, em que $x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}$.

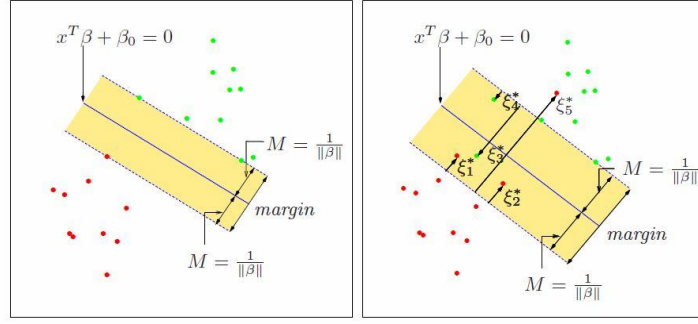


Figura 22 - Caracterização do conceito de margem em classificadores SVM. Imagem à esquerda: Dados separáveis; Imagem à direita: dados não separáveis [20, p. 418]

Um hiperplano é definido através da expressão (2.13), em que β é um vector unitário e como tal $\|\beta\| = 1$, e β_0 é o valor de enviesamento (“bias”). A expressão (2.14) designa a regra de classificação limitada às classes “-1” e “1”.

$$\{x: f(x) = x^T \cdot \beta + \beta_0 = 0\} \quad (2.13)$$

$$G(x) = \text{sign}(x^T \cdot \beta + \beta_0) \quad (2.14)$$

Considerando que as classes são separáveis (ver imagem à esquerda na Figura 22) a função $f(x) = x^T \cdot \beta + \beta_0$ com $y_i f(x_i) > 0 \forall i$, permitirá que se determine o hiperplano que apresenta a maior margem — identificada pelo parâmetro “M” — entre os pontos de treino para as classes “1” e “-1”, resultando no problema de optimização com função objectivo e conjunto de restrições dados, respectivamente, pelas expressões (2.15) e (2.16).

$$\max_{\beta, \beta_0, \|\beta\|=1} M \quad (2.15)$$

$$y_i(x_i^T \cdot \beta + \beta_0) \geq M, \quad i = 1, \dots, N \quad (2.16)$$

As expressões (2.17) e (2.18) apresentam uma versão equivalente do problema de optimização, o qual constitui um exemplo de um problema de optimização convexa. A formulação apresentada é aplicável a conjuntos de dados separáveis, utilizando o conceito de margem rígida.

$$\min_{\beta, \beta_0} \|\beta\| \quad (2.17)$$

$$y_i(x_i^T \cdot \beta + \beta_0) \geq 1, \quad i = 1, \dots, N \quad (2.18)$$

As expressões (2.19) e (2.20) formalizam o problema de optimização aplicável a conjuntos de dados não separáveis (ver imagem à direita na Figura 22), introduzindo o conceito de margem suave. A margem suave admite que algumas observações se localizem fora da região de decisão correcta e que o objectivo de maximização da margem se mantenha. Este comportamento é descrito na formulação através das variáveis de folga ξ , as quais indicam a distância de um ponto à margem correcta. Desta forma, é possível reduzir o número de classificações inválidas através da limitação de $\sum \xi$, ou seja, do somatório de desvios dos pontos localizados fora das regiões de decisão correctas.

$$y_i(x_i^T \cdot \beta + \beta_0) \geq M(1 - \xi_i) \quad (2.19)$$

$$\min \|\beta\| \quad \text{sujeito a} \quad \begin{cases} y_i(x_i^T \cdot \beta + \beta_0) \geq 1 - \xi_i \quad \forall i \\ \xi_i \geq 0, \sum \xi_i \leq \text{constante} \end{cases} \quad (2.20)$$

As expressões (2.21) e (2.22) apresentam uma formulação computacionalmente conveniente do problema de optimização descrito pela expressão (2.19). O parâmetro de custo “C”, de valor positivo, tem como função penalizar o somatório de desvios de pontos localizados fora das regiões de decisão correctas. Desta forma, um parâmetro de custo demasiado elevado tenderá a penalizar fortemente qualquer desvio, conduzindo a um sobre-ajustamento (“*overfitting*”), enquanto o seu oposto promoverá uma fronteira de decisão mais suave [20, p. 424].

$$\min_{\beta, \beta_0} \frac{1}{2} \cdot \|\beta\|^2 + C \cdot \sum_{i=1}^N \xi_i \quad (2.21)$$

$$\xi_i \geq 0, y_i(x_i^T \cdot \beta + \beta_0) \geq 1 - \xi_i, \quad \forall i \quad (2.22)$$

2.2.3 Funções de núcleo

Um classificador SVM, sob a forma de algoritmo linear, é caracterizado pela utilização de superfícies de decisão lineares, as quais são funções lineares dos vectores de entrada [22]. Considerando que os dados a classificar — também designados por espaço de características — não são sempre linearmente separáveis, tornou-se necessário dotar o algoritmo de fronteiras de decisão mais flexíveis, beneficiando o desempenho de classificação. Para tal, foi formulado um mapeamento entre o espaço original de características e um espaço de elevada dimensão, designado por espaço alargado, procurando que a separação linear neste espaço conduza a um melhor desempenho do classificador SVM, ainda que as fronteiras no espaço original não sejam lineares [20, p. 423]. A transformação do espaço de características num conjunto linearmente separável é resultado da aplicação de uma função de núcleo.

A inclusão da função de núcleo na formulação do algoritmo SVM, definido sob a forma de problema de optimização, é realizada através de um artifício designado por “*kernel trick*” em que, partindo do pressuposto de que a formulação traduz todos os vectores de entrada em produtos internos, se substituem os produtos internos por funções de núcleo [22]. A técnica de substituição do núcleo (“*kernel trick*”) pode ter igualmente aplicação em classificadores KNN (“*k-nearest neighbor*”) e classificadores de Fisher [22, p. 292].

A Figura 23 apresenta um exemplo de aplicação de funções de núcleo Gaussianas a um conjunto de dados de fronteiras não lineares, mostrando que no espaço alargado obtido se verifica uma inequívoca condição de separabilidade linear dos dados, permitindo a utilização indirecta de um classificador linear SVM no espaço de dados original.

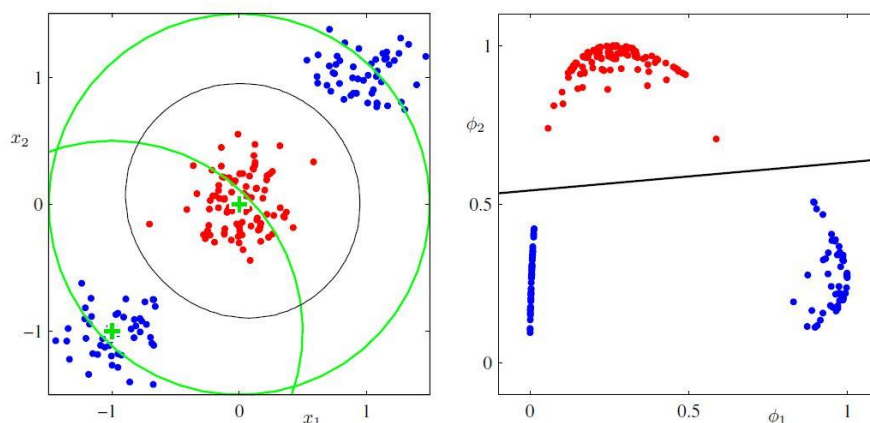


Figura 23 - Exemplo de transformação de dados linearmente não separáveis. Imagem à esquerda: espaço de dados. Imagem à direita: espaço alargado [22, p. 204]

A Tabela 1 sumariza um conjunto de funções de núcleo tipicamente utilizadas em classificadores SVM.

Tabela 1 - Funções de núcleo típicas [20]

Função de núcleo	Expressão
Polinomial (grau d)	$K(x, x')(1 + \langle x, x' \rangle)^d$
Gaussiana	$K(x, x') = \exp(-\gamma \ x - x'\ ^2), \gamma = \frac{1}{\sigma}$
Sigmóide	$K(x, x') = \tanh(k_1 \langle x, x' \rangle + k_2)$

Na função de núcleo polinomial, a flexibilidade da fronteira de decisão é definida pelo parâmetro de grau “d”, em que a rigidez de separação é tanto menor quanto maior for o grau. Nesta função, o caso de menor rigidez é observado quando o parâmetro “d” tem valor igual a 1, realizando uma separação linear dos dados. A Figura 24 ilustra 3 cenários de separação de dados através de uma função polinomial.

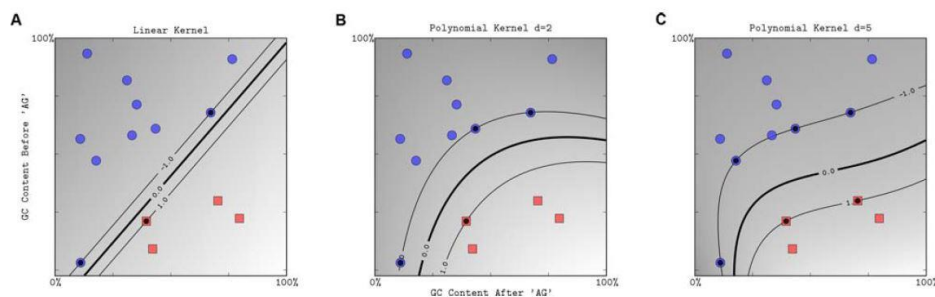


Figura 24 - Fronteira de função de núcleo polinomial. A: separação linear; B: separação d=2; C: separação d=5 [21]

Na função de núcleo Gaussiana, o parâmetro σ apresenta um comportamento inversamente proporcional ao grau “d” da função polinomial. A Figura 25 ilustra as fronteiras obtidas com diferentes funções de núcleo gaussianas. Nos três cenários ilustrados, é evidente que valores menores do parâmetro σ conduzem a superfícies de decisão de maior curvatura e especialmente localizadas, facto que poderá conduzir a um “overfitting” dos dados a classificar (vide cenário C na Figura 25) [21].

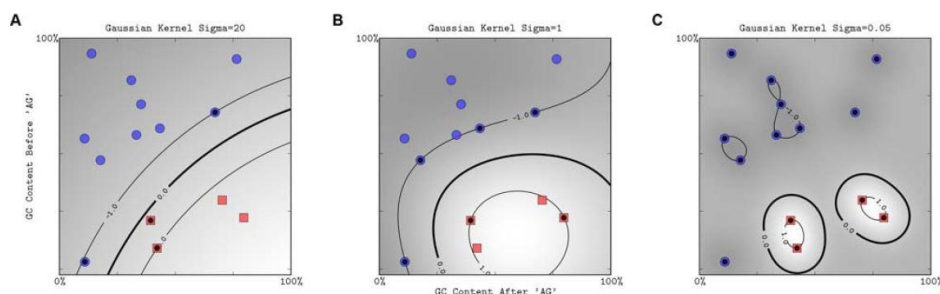


Figura 25 - Fronteira de função de núcleo gaussiana. A: $\sigma=20$; B: $\sigma=1$; C: $\sigma=0,05$ [21]

2.2.4 Sistemas de multi-classificação

Nas secções anteriores foram abordados apenas problemas de classificação simples, utilizando conjuntos de observações de duas classes disjuntas. Esta opção é justificada, essencialmente, pelo facto de o algoritmo SVM ser um classificador binário [22, p. 338]. Numa utilização corrente, é expectável que existam problemas de classificação de maior complexidade que requeiram um número superior de classes — problema de multi-classificação — sendo necessário utilizar o algoritmo SVM segundo uma heurística particular. Do conjunto de heurísticas de multi-classificação salientam-se as seguintes: “one-vs-the-rest”, “one-vs-one”.

A heurística “one-vs-the-rest” aborda o problema de classificação de K classes distintas, realizando K classificadores SVM binários. Uma das limitações da heurística prende-se com o desequilíbrio dos dados de treino, ainda que o conjunto de observações tenha igual número de observações por classe, dado que cada classificador binário irá alocar um diferente número de observações entre as classes positiva e negativa, desprezando a simetria do problema [22, p. 338]. A Figura 26 ilustra a aplicação da heurística, verificando-se que cada classificador confronta as observações de uma única classe com as observações pertencentes às restantes classes.

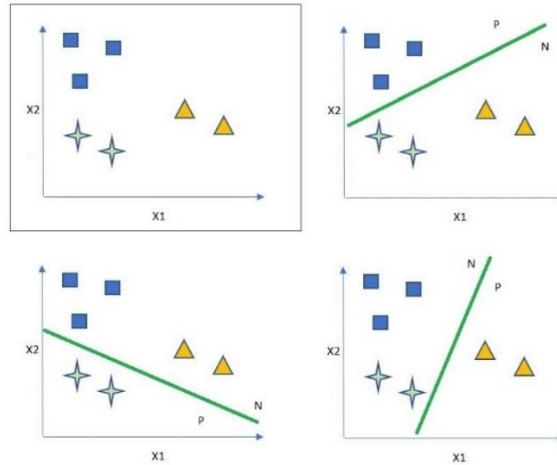


Figura 26 - Exemplo de método "one-vs-the-rest" aplicado a três classes. P: classe positiva; N: classe negativa

No caso da heurística "one-vs-one", o problema de classificação de K classes é resolvido através da realização de uma combinação de $\binom{K}{2}$ classificadores binários SVM, traduzindo-se na necessidade de treinar $K(K-1)/2$ classificadores, resultando num maior peso computacional em problemas mais extensos [22, p. 339].

Nesta heurística o resultado final é obtido através de uma contagem de votos, conceito que será generalizado no modelo de classificação ECOC ("error-correcting output codes"), obtendo-se uma maior robustez nos resultados [22, p. 339]. A Figura 27 ilustra a aplicação da heurística num conjunto de observações de três classes distintas, tendo cada classificador como dados de aprendizagem apenas observações pertencentes a duas classes.

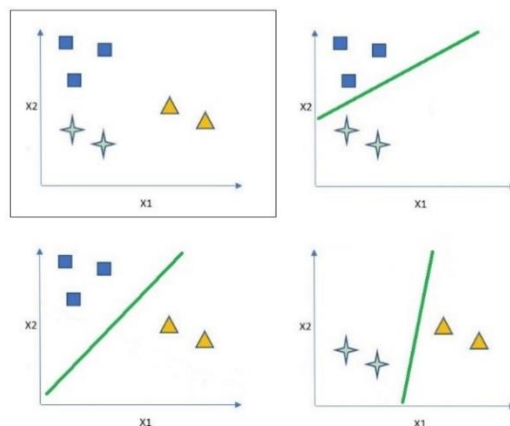


Figura 27 - Exemplo de método "one-vs-one" aplicado a três classes

2.2.5 Ferramentas computacionais do algoritmo

O crescente interesse em processos de aprendizagem automática tem motivado a criação de diversas ferramentas computacionais especializadas, sejam soluções comerciais ou de código aberto (“*open-source*”), destacando-se, numa lista não exaustiva, os seguintes exemplos: LIBSVM, Scikit-learn, WEKA e MATLAB.

A solução LIBSVM, desenvolvida por Chang e Lin (2000) [23], constitui uma biblioteca em código aberto especializada na resolução de problemas de classificação e regressão através de máquinas de suporte vectorial (SVM). Além dos problemas de classificação binária, a biblioteca permite também, resolver problemas de classificação constituídos por várias classes (multi-classificação) através do método “*one-vs-one*”, tolerando a existência de dados de treino desequilibrados — situação em que, num conjunto de dados, o número de observações por classe não é uniforme — atribuindo a cada classe um peso específico [23]. A interoperabilidade da biblioteca com várias linguagens de programação é assegurada por um vasto conjunto de interfaces e extensões compatíveis com Java, MATLAB, LabVIEW, Python, Android, C# .NET, entre outros [24].

A solução Scikit-learn, desenvolvida por Pedregosa et al. (2011) [25], é uma ferramenta em código aberto desenvolvida em Python, que agrega um conjunto de algoritmos de aprendizagem automática, nas variantes supervisionada e não supervisionada, especializados na resolução de problemas de regressão, classificação e agrupamento (“*clustering*”). O conjunto de algoritmos é constituído por vários modelos, tais como: redes neuronais, árvores de decisão, modelos K-NN (“*k-nearest neighbors*”) e modelos SVM. É importante notar que nesta solução os modelos SVM são baseados na biblioteca LIBSVM, tendo-se adicionado capacidades ao algoritmo, tais como novos métodos de multi-classificação (“*one-vs-the-rest*”) e utilização de modelos ECOC (“*error correcting output codes*”). A solução Scikit-learn permite, também, realizar de forma intuitiva a redução dimensional e o pré-processamento de dados, bem como a optimização de parâmetros e validação de modelos através de métodos de “*grid-search*” e validação cruzada, respectivamente.

A solução WEKA [26], desenvolvida na universidade de Waikato, na Nova Zelândia, é uma ferramenta computacional especializada na resolução de problemas comuns no domínio da aprendizagem automática, como classificação supervisionada e “*clustering*”, disponibilizando uma biblioteca de algoritmos acessíveis através de uma interface gráfica intuitiva. Do conjunto de algoritmos de aprendizagem supervisionada, é possível utilizar vários modelos de classificação, tais como: SVM, classificadores Bayesianos, árvores de decisão e redes neuronais. A solução WEKA, sendo um projecto em código aberto e desenvolvido em JAVA, tem como vantagem adicional a possibilidade de aplicações independentes desenvolvidas em JAVA utilizarem a sua biblioteca de algoritmos de aprendizagem automática.

A solução MATLAB® em conjunção com a ferramenta “*Statistics and Machine Learning Toolbox*” [27] oferece um extenso conjunto de algoritmos de aprendizagem, supervisionada e não supervisionada, aplicável a problemas de classificação, regressão e reconhecimento de padrões (“*clustering*”). O conjunto de algoritmos, entre outros, inclui modelos SVM de classificação binária e multi-classificação (ECOC), árvores de decisão, classificadores Bayesianos e modelos K-NN. A “*Toolbox*” oferece, à semelhança de outras soluções, uma interface gráfica que permite testar o desempenho dos vários algoritmos de classificação, realizar uma análise dimensional aos dados, bem como realizar uma selecção de atributos. Esta solução comercial tem como característica a coexistência de várias ferramentas de diferentes domínios científicos num ambiente de desenvolvimento com uma linguagem de programação particular (MATLAB) e um ambiente de simulação de sistemas dinâmicos (*Simulink*®).

2.3 Sistemas multi-agente

A natureza distribuída dos sistemas multi-agente torna a sua utilização desejável em vários ambientes, não necessariamente industriais, nos quais se pretende que os vários dispositivos existentes apresentem um comportamento cooperativo entre si, minimizando tanto quanto possível o impacto na escalabilidade da solução. No presente capítulo serão introduzidos os conceitos fundamentais na temática de sistemas multi-agente, facilitando uma posterior análise à implementação concreta no problema em estudo.

2.3.1 O Conceito de Agente

Tal como a designação sugere, um sistema multi-agente compreende um conjunto de agentes cuja interacção se crê benéfica para um determinado objectivo. O conceito de *agente* pode ser traduzido como um sistema informático, situado num determinado ambiente, capaz de agir autonomamente nesse ambiente de modo a cumprir os objectivos para os quais foi desenvolvido [28, p. 15].

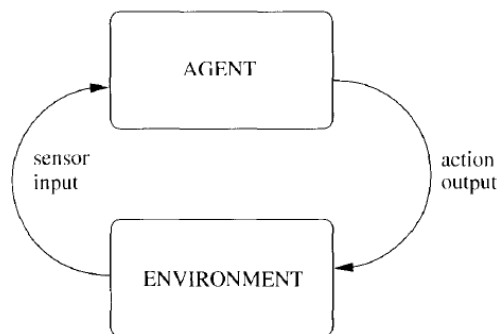


Figura 28- Diagrama agente-ambiente [28, p. 16]

A Figura 28 reproduz de forma simplificada o comportamento de um agente genérico. É possível observar que o agente possui uma representação do ambiente reforçada via sensores externos e que o mesmo é capaz de alterar esse ambiente através das suas acções. O ambiente não designa necessariamente uma envolvente física, podendo tão simplesmente ser uma abstracção de um conjunto de variáveis a observar e controlar pelo próprio agente. É também importante salientar que os ambientes podem incluir uma componente de incerteza, não reagindo de igual forma a excitações iguais realizadas pelo agente. Deste modo o agente terá de possuir a capacidade de responder a resultados inesperados por via das suas acções.

A capacidade do agente em lidar com situações inesperadas está intimamente ligada à sua habilidade social na medida em que a interacção entre agentes envolve frequentemente actos de negociação e notificação. Assim sendo, quando um agente realiza uma tarefa por influência de outro agente conclui-se que o primeiro aceitou as condições do segundo, julgando-as positivas para o seu interesse individual. A menção à palavra “influência” não é casual. A interacção entre agentes deve reger-se por pedidos e não por ordens directas, sendo cada agente livre de aceitar ou recusar os pedidos que lhe são destinados.

O conceito de inteligência de um agente, não sendo absoluto, poderá ser identificado quando o agente demonstra habilidade social, proactividade e reactividade [28, p. 23]: A proactividade num agente é identificada na capacidade do mesmo agir procurando satisfazer um objectivo individual. A reactividade, por sua vez, designa a capacidade de o agente, percepcionando qualquer alteração súbita no ambiente, agir de imediato sobre o ambiente de modo a cumprir os seus objectivos.

É interessante notar a similaridade entre o conceito de *agente* como entidade de um sistema informático e um agente económico humano, tal como descrito pela microeconomia. O agente económico é caracterizado igualmente por um comportamento genericamente racional face à escassez de recursos económicos, realizando escolhas e actos de negociação com os seus pares, tendo como objectivo a maximização da sua satisfação, ainda que as suas decisões não sejam necessariamente óptimas, um fenómeno no âmbito da economia comportamental estudado em [29].

2.3.2 Comunicação entre agentes

A multiplicidade de intervenientes num sistema multi-agente requer um mecanismo de comunicação com particularidades próprias, nomeadamente a utilização de uma linguagem comum e a existência de entidades de gestão dos agentes. Os agentes, sendo entidades autónomas, dotados de capacidade de decisão, necessitam de um tipo de linguagem que permita caracterizar as mensagens que lhes são endereçadas. A resposta a este problema baseou-se na teoria dos actos de linguagem (“*speech acts*”) permitindo separar o conteúdo das mensagens do acto comunicativo [30, p. 6]. O acto comunicativo visa descrever o tipo de mensagem, podendo apresentar, entre outros, um carácter de declaração, interrogação e negociação.

Actualmente a linguagem de comunicação entre agentes (ACL – “*agent communication language*”) mais utilizada é a FIPA-ACL, permitindo que um conjunto de agentes ao utilizar protocolos de interacção predefinidos demonstre um comportamento globalmente coerente [30, p. 6]. Esta linguagem teve a sua génese na FIPA (“*Foundation for intelligent physical agents*”), uma organização membro da IEEE (“*Institute of electrical and electronics engineers*”) cuja missão é promover a tecnologia baseada em agentes, sendo também responsável pela criação de normas que facilitem a interacção de agentes e serviços heterogéneos [31].

A estrutura de uma mensagem do tipo FIPA-ACL contém vários parâmetros necessários a uma correcta comunicação entre agentes. De acordo com a especificação FIPA [32] o único parâmetro de preenchimento obrigatório é o relativo ao acto comunicativo, esperando-se ainda assim que os parâmetros relativos ao destinatário, remetente e conteúdo estejam preenchidos.

A Tabela 2 apresenta uma lista não exaustiva dos actos comunicativos FIPA acompanhados da respectiva descrição. É possível observar através dos actos comunicativos listados que os agentes possuem de facto uma capacidade de negociação, podendo recusar uma proposta endereçada através do acto “*refuse*”. À ocorrência de falha está associado o acto comunicativo “*failure*”, informando o agente solicitador que a tarefa pedida não foi realizada. Este comportamento permite ao agente solicitador gerir a situação de falha, repetindo o pedido ao destinatário ou adoptando uma estratégia diferente.

Tabela 2 -Actos comunicativos FIPA, adaptado de [30, p. 19]

Acto comunicativo FIPA	Descrição
<i>Accept proposal</i>	Aceita proposta previamente submetida
<i>Agree</i>	Concorda realizar acção futura
<i>Call for proposal</i>	Pedido de novas propostas
<i>Confirm</i>	Informa destinatário que determinada preposição é verdadeira
<i>Failure</i>	Informa agente que execução de tarefa falhou
<i>Inform</i>	Remetente indica que preposição é verdadeira
<i>Not understood</i>	Remetente indica que não entendeu a mensagem
<i>Request</i>	Solicita a realização de uma acção por parte do destinatário
<i>Refuse</i>	Recusa execução de tarefa e indica razão

2.3.3 Gestão dos agentes

A arquitectura de um sistema multi-agente inclui uma camada cujas entidades são responsáveis pela gestão dos agentes na plataforma. Esta secção irá analisar a função de cada entidade de gestão.

De acordo com a Figura 29 a arquitectura de um sistema multi-agente é constituída por vários componentes, destacando-se os seguintes:

- *Agent Platform (AP)*
- *MTS (Message transport service)*
- *Directory Facilitator (DF)*
- *Agent Management System (AMS)*

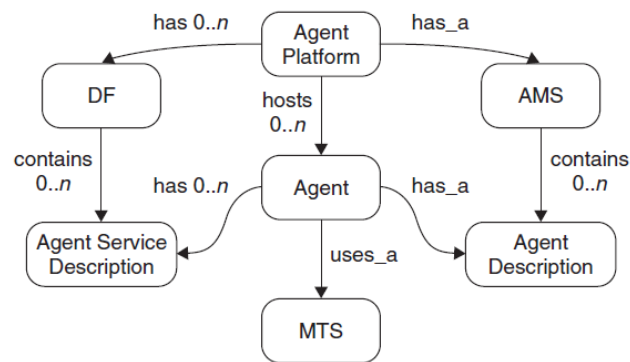


Figura 29 – Arquitetura genérica de sistema multi-agente [30, p. 15]

Agent Platform (AP): designa a infraestrutura física que suporta os agentes, nomeadamente computadores, sistemas operativos, componentes de gestão dos agentes e os agentes em si [30, p. 15].

Message transport service (MTS): designa um serviço de transporte de mensagens, permitindo que os vários agentes na plataforma troquem mensagens FIPA-ACL entre si, de acordo com os parâmetros definidos em cada mensagem [30, p. 15]

Directory Facilitator (DF): este componente, também designado por “serviço de páginas amarelas”, é responsável por manter um registo fidedigno dos serviços prestados pelos agentes subscritores [30]. O acto de registo não é obrigatório, contudo permite a qualquer agente procurar autonomamente o agente adequado à realização de uma tarefa. A existência deste serviço promove a escalabilidade do sistema multi-agente na medida em que suporta o alargamento do conjunto de agentes sem requerer uma nova configuração dos agentes.

Agent Management System (AMS): designa um sistema de gestão dos agentes. É um componente único e obrigatório na infraestrutura que suporta os agentes, sendo responsável pela criação e eliminação dos agentes [30, p. 15].

2.3.4 Modelação do comportamento racional

Um agente, como parte integrante de um sistema distribuído e cooperativo, deve adoptar um comportamento racional que lhe permita satisfazer os objectivos propostos, sejam eles individuais ou globais. Nesta secção serão introduzidos alguns conceitos acerca da modelação de um comportamento racional em agentes.

Tal como descrito anteriormente, o sistema social humano apresenta-se como uma forte inspiração aos sistemas multi-agente, designadamente na forma como indivíduos avaliam e tomam decisões racionais num ambiente inerentemente distribuído e competitivo. Este comportamento permite introduzir o conceito de raciocínio prático. O raciocínio prático, ao contrário do raciocínio teórico que actua ao nível das crenças do agente, é um processo essencialmente focado na acção, compreendendo uma fase de deliberação em que se identificam os objectivos a alcançar e uma fase de selecção dos meios que permitem satisfazer os fins a que o agente se propôs [28, p. 66]. A esta fase é dado o nome de raciocínio “meios-fins” [28].

É importante notar que a implementação de um modelo de raciocínio prático num sistema multi-agente artificial, sendo realizada através de processos computacionais, apresenta desafios ao nível da utilização dos recursos (memória, processamento), exigindo que os mesmos sejam utilizados da forma mais eficiente possível e prevendo que um processo de deliberação não se prolongue indefinidamente no tempo [28, p. 66]. Esta limitação, ainda que identificada apenas em agentes artificiais, é facilmente identificada em sistemas sociais humanos, não sendo particularmente útil que o processo de tomada de decisão seja de tal forma moroso que o propósito da decisão deixe de existir e o objectivo não seja cumprido.

O conceito de intenção, como base do modelo de raciocínio “meios-fins”, designa um estado a que o agente se compromete atingir, sendo expectável que o mesmo empreenda uma acção substancialmente relevante de forma a satisfazer o seu objectivo [28, p. 67]. O conceito de desejo, ao contrário da intenção, possui um reduzido potencial de influência na acção a empreender pelo agente [28]. A intenção possui também um carácter persistente — só sendo permitido o abandono da mesma caso exista uma justificação racional — devendo o agente garantir que a adopção de uma intenção não irá invalidar uma intenção a que se tenha comprometido anteriormente [28, p. 69]. O conceito de crença, como representação simbólica do mundo do agente, apresenta-se como suporte da intenção na medida em que o agente acredita que a acção a empreender lhe permitirá alcançar o seu objectivo, um comportamento entendido como racional [28].

O paradigma BDI (“*belief-desire-intention*”) designa um modelo de raciocínio suportado pelos conceitos introduzidos anteriormente, designadamente: crenças, desejos e intenções.

O sistema PRS (“*Procedural reasoning system*”), desenvolvido por Michael Georgeff e Amy Lansky, é um exemplo de sistema multi-agente desenvolvido de acordo com o paradigma BDI, tendo sido utilizado em diversas aplicações, desde o sistema de controlo de tráfego aéreo OASIS [33] a sistemas de simulação [28, p. 82]. Na Figura 30 é possível observar a arquitectura do sistema PRS, sendo evidente a interacção entre o ambiente (“*environment*”) e o agente através das suas crenças, representadas via sensores, e intenções que induzem uma alteração no ambiente através de uma acção.

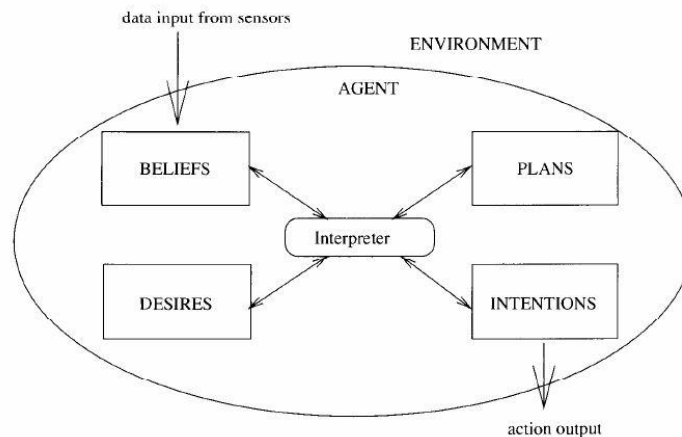


Figura 30 - Sistema PRS (Procedural reasoning system) [28, p. 83]

O planeamento no sistema PRS é realizado através de planos suportados por crenças iniciais escritas em *Prolog*, os quais são desenvolvidos manualmente por um programador que define os seguintes componentes: objectivo ou pós-condição; contexto ou pré-condição; conjunto de tarefas a executar [28]. No processo de deliberação o PRS selecciona o plano mais adequado à satisfação do objectivo que se encontra pendente numa lista —“*intention stack*” — considerando as suas crenças actuais e respeitando a prioridade de cada objectivo [28].

2.3.5 Plataformas multi-agente

O crescente interesse no estudo de sistemas multi-agente motivou o desenvolvimento de várias plataformas informáticas que explorassem o paradigma destes sistemas. Nesta secção serão resumidas as principais características do conjunto de plataformas multi-agente indicado na Tabela 3 que inclui soluções de software livre e soluções comerciais.

A plataforma *NetLogo*, desenvolvida por Uri Wilensky em 1999, é uma ferramenta gratuita de simulação de sistemas complexos interactivos, com especial enfoque na modelação de fenómenos observados nos domínios da biologia, economia, física, química e medicina [34]. Esta ferramenta é distribuída segundo uma licença de código aberto GPL², podendo ser executada em vários sistemas operativos.

² GPL, um acrónimo de “General Public License”, é uma variante de licença de software livre. Para mais detalhes aconselha-se a consulta de <https://www.gnu.org/licenses/gpl-faq.html>

Tabela 3 - Exemplo de plataformas multi-agente. (adaptado de [35])

Plataforma	Autor	Código aberto	Linguagem
NetLogo	Universidade Northwestern, E.U.A.	Sim	NetLogo, JAVA
Jason	Universidades do Rio Grande do Sul e Santa Catarina, Brasil	Sim	JAVA
Jadex	Universidade de Hamburgo, Alemanha	Sim	JAVA
MASON	Universidade George Mason, E.U.A.	Sim	JAVA
AgentScape	Universidade de Tecnologia de Delft, Países Baixos	Sim	JAVA
JACK	AOS	Não	JAVA
JADE	Telecom Itália	Sim	JAVA

A plataforma *Jason* é uma solução multi-agente de interpretação de linguagem AgentSpeak, uma linguagem de grande influência na realização de agentes cognitivos na arquitectura BDI (“*beliefs-desires-intentions*”) [36]. Esta plataforma, desenvolvida em JAVA, quando utilizada com outras soluções “*middleware*” tal como o JADE, torna possível a interoperabilidade entre agentes pertencentes a plataformas distintas, suportando-se na comunicação segundo a especificação FIPA-ACL [36].

A plataforma *Jadex* é uma solução de modelação de sistemas multi-agente baseada na arquitectura BDI, cujo foco se centra no comportamento racional de um agente individual e na satisfação dos seus objectivos, conjugando este comportamento com a observância das especificações preconizadas pela FIPA [37]. É importante notar que a plataforma *Jadex*, sendo essencialmente um componente de raciocínio do agente, deverá ser suportada por uma infra-estrutura de “*middleware*” como por exemplo o JADE [37]. Esta solução é desenvolvida em JAVA e distribuída livremente segundo a licença GPL.

A plataforma *MASON* é constituída por um conjunto de métodos desenvolvidos em Java que visam facilitar a criação de sistemas multi-agente e a visualização do seu comportamento, existindo um especial enfoque em sistemas com um largo número de agentes (“*swarm multi-agent systems*”) observados nos domínios da robótica e biologia [38]. Esta plataforma é distribuída em código aberto de forma gratuita.

A ferramenta *AgentScape* é uma solução “*middleware*” em código aberto, especializada na gestão de sistemas multi-agente de larga escala, apresentando um especial enfoque em características tais como a interoperabilidade, segurança, escalabilidade e migração de agentes [39]. A plataforma é similar à solução JADE na medida em que ambas implementam uma infra-estrutura multi-agente, distinguindo-se desta última ao não limitar a migração de agentes à própria plataforma (“*location*”) — termo que designa o espaço que acomoda objectos e agentes [40], [39]. A plataforma *AgentScape* suporta vários sistemas operativos e é distribuída gratuitamente.

A solução *Jack* é uma ferramenta comercial, multiplataforma, que implementa o paradigma de sistemas multi-agente segundo o modelo BDI, disponibilizando um ambiente de desenvolvimento intuitivo e uma linguagem orientada à modelação de agentes baseada em Java [41].

A plataforma JADE³, cujo desenvolvimento se iniciou em 1998 por iniciativa da Telecom Italia, é uma solução “*middleware*” — um termo atribuído a um componente de software que se assume como uma camada de intermediação entre o sistema operativo e as aplicações existentes [42] — que visa facilitar a realização de sistemas multi-agente fiéis às especificações FIPA, disponibilizando uma interface simples e um conjunto de abstrações que permitam a um programador desenvolver um sistema conforme as especificações sem a necessidade de as conhecer em detalhe [30, p. 29].

A plataforma JADE foi desenvolvida em JAVA, uma linguagem orientada a objectos que se caracteriza pelo recurso a uma máquina virtual de execução de código (“JVM - *JAVA virtual machine*”), permitindo que o mesmo código seja executado em diversos sistemas operativos (ex: *Windows, Mac OS X, Linux*). Esta plataforma é uma solução em código aberto, distribuída segundo uma licença LGPL⁴, prevendo-se a integração de soluções de software proprietário com a plataforma JADE sob limitações [30, p. 29]. A Figura 31 apresenta a interface gráfica da ferramenta de gestão de agentes da plataforma designada por RMA (“*remote monitoring agent*”).

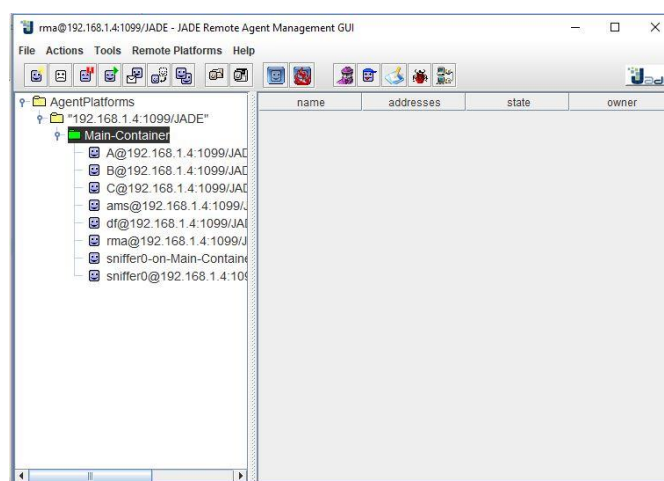


Figura 31 - Interface gráfica RMA

³ JAVA Agent DEvelopment framework. Sítio oficial: <http://jade.tilab.com/>

⁴ LGPL, um acrónimo de “Lesser General Public License”, é uma variante de licença de software livre. Para mais detalhes aconselha-se a consulta de <https://www.gnu.org/licenses/gpl-faq.html>

3 Descrição do conversor de energia ISWEC

O conversor de energia ISWEC (*Inertial Sea Wave Energy Converter*) [43] tem como princípio de funcionamento a interacção por efeito giroscópico entre um flutuador, ancorado ao fundo do mar, e um gerador de electricidade (ver Figura 32) [43, p. 11]. A Figura 33 apresenta o sistema mecânico no interior do flutuador. O sistema mecânico possui um volante de inércia em rotação (eixo " ϕ "), o qual descreve um movimento de precessão segundo o eixo " ε " sempre que o flutuador seja sujeito, por acção de uma onda marítima, a um binário segundo o eixo " δ ". A transmissão do binário à máquina eléctrica através de um movimento de precessão permite, assim, extrair energia das ondas marítimas — um recurso renovável — e convertê-la em electricidade.

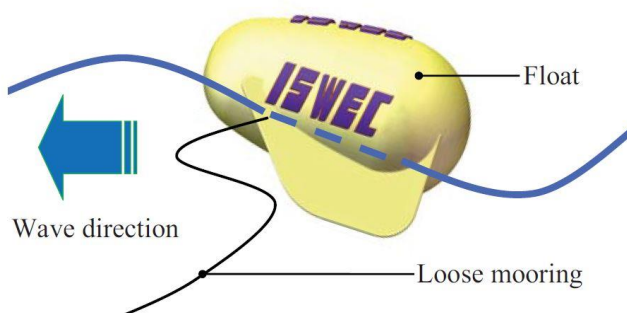


Figura 32 - Ilustração ISWEC [43]

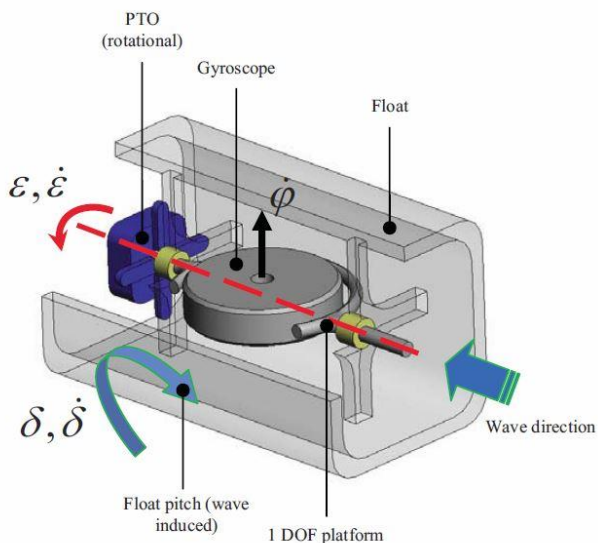


Figura 33 - Sistema mecânico ISWEC [43]

O conversor ISWEC apresenta-se como um equipamento de fácil manutenção e operação, possuindo um sistema electromecânico completamente selado no corpo do flutuador, tendo também a capacidade de se orientar automaticamente de acordo com a onda marítima [44]. O equipamento ISWEC foi lançado oficialmente em 2015, junto à costa de Pantelleria, Itália.

3.1 Modelo matemático

Nesta secção serão apresentadas, sucintamente, as bases teóricas à formulação do conversor de energia ISWEC, designadamente ao nível da hidrodinâmica do flutuador, dinâmica do sistema de conversão e modelo global do ISWEC.

3.1.1 Dinâmica do flutuador

A dinâmica do flutuador visa relacionar o comportamento do flutuador com as forças induzidas pelas ondas marítimas. Por uma questão de simplicidade, admite-se que a interacção entre o flutuador e as ondas marítimas se pode reduzir a um problema planar, isto é, num plano definido pelo vector de gravidade e vector de direcção da onda [44, p. 2]. A equação (3.1) descreve o movimento de “pitch” ou cabeceio do flutuador (indicado pelo eixo “ δ ” na Figura 33) induzido pela onda, considerando que o flutuador é um corpo rígido com uma velocidade longitudinal nula [45, p. 8]. Na equação, o parâmetro I_f designa o momento de inércia do flutuador, A_∞ representa a massa adicionada por acção do flutuador, K é a rigidez hidrostática, h_r é uma função de resposta às forças radiantes, T representa o binário induzido no flutuador por acção das ondas, δ é o ângulo de “pitch” do flutuador [45].

$$(I_f + A_\infty) \cdot \ddot{\delta} + \int_0^t h_r(t - \tau) \cdot \dot{\delta} d\tau + K \cdot \delta = T \quad (3.1)$$

A equação (3.2) descreve as várias componentes do binário induzido no flutuador, em que T_w indica o binário de “pitch” devido à onda, T_δ é o binário de controlo do movimento de “pitch” e T_m indica o binário devido às forças de amarração do flutuador, ainda que esta componente possa ser desprezada [45].

$$T = T_w - T_\delta - T_m \quad (3.2)$$

Considerando que a computação do segundo termo na equação (3.1), $\int_0^t h_r(t - \tau) \cdot \delta d\tau$ — termo representativo dos binários de radiação induzidos no flutuador — é especialmente morosa, admite-se como conveniente a aproximação deste termo pelo modelo em espaço de estados indicado de (3.3) a (3.7) [45].

$$T_{rad} = \int_0^t h_r(t - \tau) \cdot \delta d\tau \approx \begin{cases} \dot{x} = Ax + B\delta \\ T_{rad} = Cx \end{cases} \quad (3.3)$$

A expressão (3.4) representa o vector de estados.

$$x = \begin{bmatrix} rf v_1 \\ rf v_2 \\ rf v_3 \\ rf v_4 \end{bmatrix} \quad (3.4)$$

A expressão (3.5) define como vectores de entrada e saída os termos, respectivamente, o ângulo de “pitch” e binários de radiação. Os parâmetros das matrizes de estado e de saída encontram-se listados na Tabela 4.

$$u = \delta, y = T_{rad} \quad (3.5)$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.6)$$

$$C = [c_1 \quad c_2 \quad c_3 \quad c_4], \quad D = 0 \quad (3.7)$$

Tabela 4 - Parâmetros do modelo aproximado de forças de radiação [45]

Parâmetro	Valor
a₁₁	-4.52
a₁₂	-10.53
a₁₃	-10.78
a₁₄	-8.03
c₁	5.14E+06
c₂	9.29E+06
c₃	7.01E+06
c₄	0.00E+00

3.1.2 Dinâmica do sistema giroscópico

A dinâmica do sistema giroscópico procura descrever a relação entre o binário induzido no flutuador e o binário transmitido ao gerador de electricidade — também designado por PTO (“*power take-off*”). A equação (3.8) descreve o binário transmitido ao PTO, representado por T_ε [45]. O parâmetro I_g indica o momento de inércia global em torno do eixo de precessão “ ε ” e J representa o momento polar de inércia do volante de inércia em torno do seu eixo de rotação. A dinâmica do binário de “*pitch*” é descrita pela equação (3.9).

$$T_\varepsilon = I_g \cdot \ddot{\varepsilon} + (I_g - J) \cdot \dot{\delta}^2 \cdot \sin \varepsilon \cdot \cos \varepsilon - J \cdot \dot{\phi} \cdot \dot{\delta} \cdot \cos \varepsilon \quad (3.8)$$

$$T_\delta = (J \cdot (\sin \varepsilon)^2 + I_g \cdot (\cos \varepsilon)^2) \cdot \ddot{\delta} + J\ddot{\phi} \sin \varepsilon + J\dot{\phi}\dot{\varepsilon} \cos \varepsilon + 2 \cdot (J - I_g)\dot{\delta}\dot{\varepsilon} \sin \varepsilon \cos \varepsilon \quad (3.9)$$

A simplificação da equação (3.8), desprezando-se os termos de ordem superior, resulta na equação (3.10). A equação (3.11) resulta da simplificação da equação (3.9), admitindo-se as seguintes condições: a velocidade de rotação do volante de inércia é constante, as acelerações de “*pitch*” são reduzidas, os termos de ordem superior são desprezáveis [45].

$$T_\varepsilon = I_g \ddot{\varepsilon} - J\dot{\phi}\dot{\delta} \cos \varepsilon \quad (3.10)$$

$$T_\delta = J\dot{\phi}\dot{\varepsilon} \cos \varepsilon \quad (3.11)$$

3.1.3 Modelo do conversor de energia

Uma vez descritas cada uma das dinâmicas relativas ao flutuador e sistema de transmissão giroscópico, é possível obter a dinâmica global do sistema, reflectindo o acoplamento entre os dois subsistemas. A expressão (3.12) indica o momento de inércia equivalente do sistema. A equação (3.13) é resultado da substituição em (3.1) da expressão dada por (3.11), incluindo o modelo aproximado em espaço de estados dado em (3.3) [45]. Nas equações, T_w designa o binário induzido pela onda marítima, T_ε designa o binário transmitido ao PTO. O modelo do conversor é descrito por (3.13) e (3.14).

$$I_{eq} = I_f + A_\infty \quad (3.12)$$

$$T_W = I_{eq} \ddot{\delta} + T_{rad} + K\delta + J\dot{\phi} \dot{\epsilon} \cos \epsilon \quad (3.13)$$

$$T_\epsilon = I_g \ddot{\epsilon} - J\dot{\phi} \dot{\delta} \cos \epsilon \quad (3.14)$$

O modelo do conversor de energia ISWEC encontra-se representado, via diagrama de blocos, na Figura 34. As constantes do conversor de energia encontram-se listadas no anexo I.

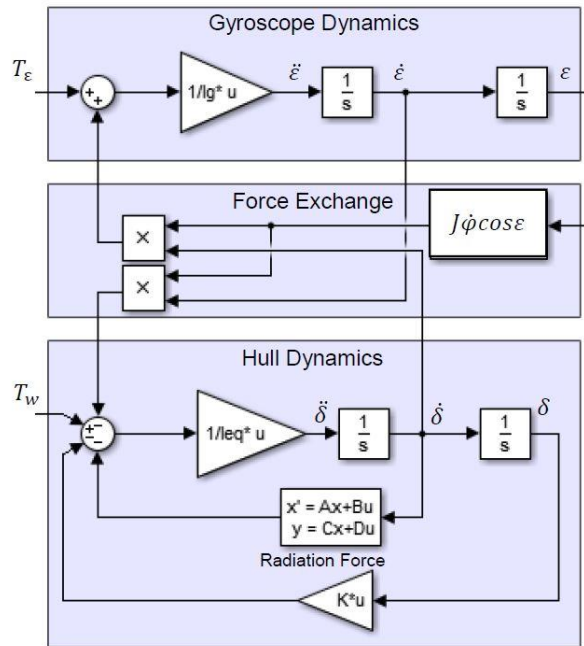


Figura 34 - Modelo do conversor ISWEC [45]

3.2 Controlador PD

O sistema de controlo original do ISWEC é do tipo proporcional-derivativo (PD), não inibindo o estudo de diferentes estratégias de controlo do dispositivo, tais como [44] e [45]. O controlador PD tem como objectivo garantir que o volante de inércia descreve um movimento oscilatório em torno da vertical, fazendo com que o PTO se comporte como um sistema mola-amortecedor. Este comportamento é realizado através da lei de controlo em (3.15), a qual reúne uma componente de rigidez k , proporcional à distância angular ε ao eixo vertical, e uma componente de amortecimento c , proporcional à velocidade segundo o eixo de precessão [44, p. 4]. A potência extraída pelo ISWEC é representada pela expressão (3.16), assumindo valores negativos para a energia que é introduzida no sistema. A potência será tanto maior quanto maior for a amplitude de “pitch” e quanto menor for o período da onda marítima [43].

$$T_\varepsilon = -k\varepsilon - c\dot{\varepsilon} \quad (3.15)$$

$$P = -T_\varepsilon \dot{\varepsilon} \quad (3.16)$$

A Figura 35 apresenta o diagrama de blocos do controlador PD aplicado ao conversor de energia ISWEC. Considerando que o controlador possui dois ganhos ajustáveis, importará conhecer o conjunto de parâmetros que maximizam a extracção de energia para cada tipo de onda marítima. O anexo I apresenta esse conjunto de ganhos, os quais foram determinados de acordo com [45]. Estas configurações do controlador são submetidas, numa base horária, de acordo a previsão do estado do mar [44, p. 4]. Nas secções seguintes, ter-se-á a oportunidade de estudar uma estratégia de ajuste dos ganhos do PD em anel fechado, não dependendo de uma previsão estática.

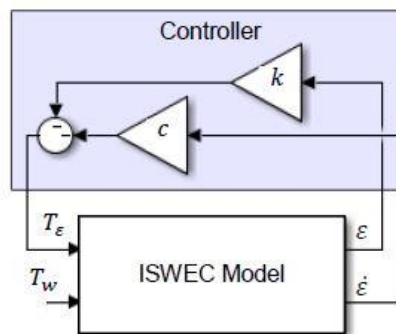


Figura 35 - Diagrama de controlador PD [45]

3.3 Modelo de ondas

Na simulação do modelo ISWEC optou-se pela utilização de ondas marítimas irregulares geradas computacionalmente, definindo-se um conjunto de parâmetros de forma a reproduzir, tão fielmente quanto possível, os diferentes estados do mar verificados em ambiente real. A análise espectral é uma ferramenta importante na caracterização estatística das ondas em águas profundas, destacando-se as formas empíricas desenvolvidas por Pierson e Moskowitz, Bretschneider, bem como o espectro JONSWAP. Na continuidade do estudo de [45], optou-se por utilizar o espectro deduzido por Bretschneider, de apenas dois parâmetros, cuja expressão é dada por [46, p. 155]:

$$S(w) = \frac{5}{16} \frac{w_m^4}{w^5} \cdot H_{\frac{1}{3}}^2 \cdot e^{-\frac{5w_m^4}{4w^4}} \quad (3.17)$$

$$T_s = \frac{2\pi}{w_m} \quad (3.18)$$

O parâmetro $H_{\frac{1}{3}}$, designa a altura significativa da onda, correspondendo à altura média das ondas no terço superior (ver Figura 36). O parâmetro w_m designa a frequência modal, correspondendo à frequência média da altura significativa da onda. Ao período significativo (T_s) corresponde o período em que se concentra o máximo de energia da onda [47, p. 17].

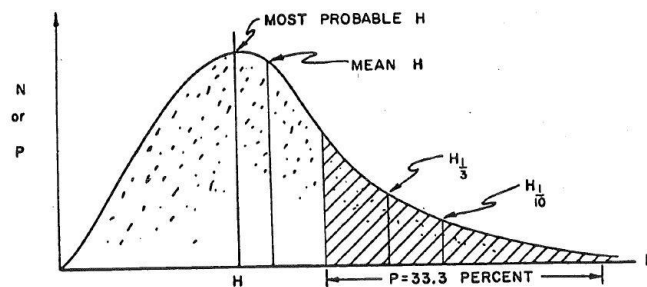


Figura 36 - Distribuição estatística da altura das ondas [47, p. 30]

Considerando a existência de distintos estados do mar, definiu-se um conjunto de oito ondas de diferentes períodos e alturas significativos (ver Tabela 5).

Tabela 5 - Características das ondas marítimas [45, p. 13]

Onda	1	2	3	4	5	6	7	8
Hs (m)	0,5	0,5	1,5	1,5	1,5	2,5	2,5	3,25
T (s)	5,5	6,5	5,5	6,5	7,5	6,5	7,5	8

O espectrograma, através da *short-time Fourier transform*, permite visualizar a densidade espectral de potência de um sinal em função do tempo. Na Figura 37 encontram-se representados os espectrogramas das oito ondas em estudo, cada uma de duração igual a 5 minutos e tempo de amostragem de 0,05 segundos, utilizando uma janela rectangular de 100 amostras e sobreposição de 50%.

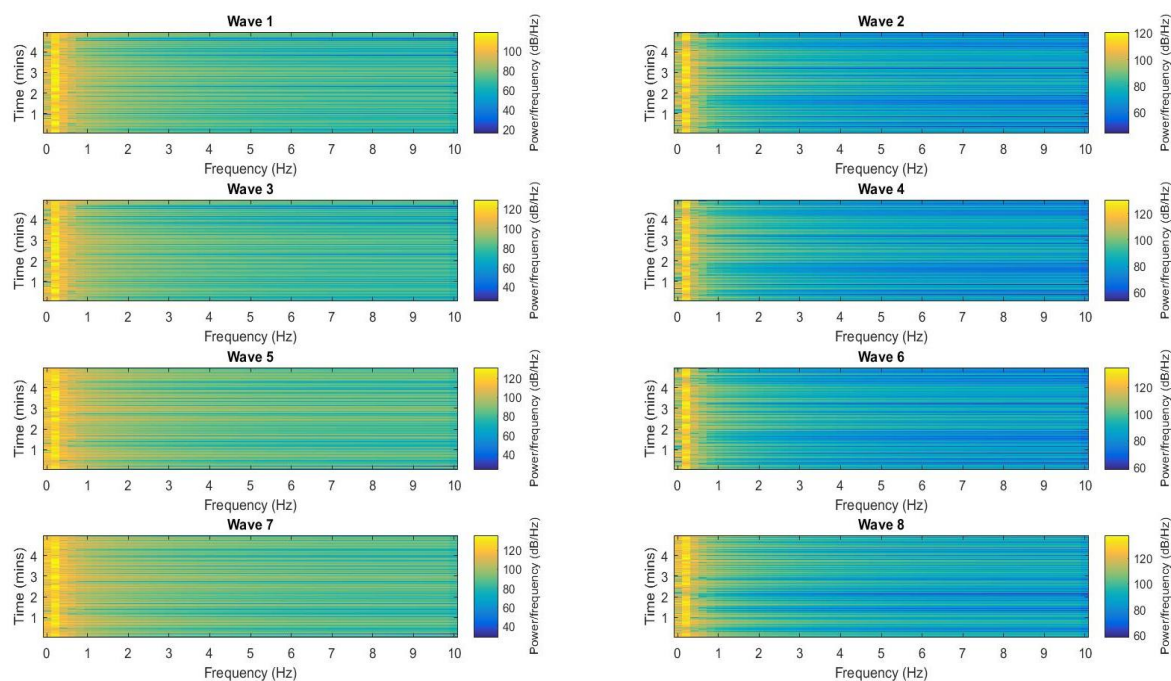


Figura 37 - Espectrogramas do conjunto de ondas

4 Controlador

O presente trabalho tem como objectivo a criação de um sistema de controlo, baseado numa arquitectura multi-agente, que se adapte continuamente às ondas marítimas que interagem com os dispositivos ISWEC, promovendo a melhoria do seu desempenho na produção de electricidade.

A configuração actual do controlador, do tipo PD, prevê que os seus ganhos são definidos periodicamente de acordo com a previsão remota do estado do mar. A principal limitação deste sistema reside no seu funcionamento em anel aberto, não permitindo a adaptação do sistema em tempo real a alterações súbitas do estado do mar. A configuração proposta visa dotar o controlador actual da capacidade de identificar o tipo de onda, aliando esta informação ao conhecimento do conjunto de ganhos que maximizam a extracção de energia para os oito tipos de onda, prevendo-se que esta configuração conduza a uma melhoria global do desempenho do ISWEC.

O controlador é constituído por três subsistemas, nomeadamente: modelo dinâmico inverso, classificador de onda e sistema multi-agente. O primeiro subsistema tem como função obter o sinal de excitação — momento induzido por onda marítima no conversor de energia — a partir de informação extraída dos sensores do ISWEC, designadamente dados de velocidade e posição, constituindo uma forma de medição indirecta do sinal de excitação. O subsistema de classificação é responsável pela identificação do tipo de onda marítima através da análise da informação gerada pelo modelo dinâmico inverso, possibilitando a selecção do par de ganhos adequado ao actual estado do mar. O último subsistema — sistema multi-agente — tem como função garantir que um conjunto de dispositivos ISWEC possui uma plataforma comum de comunicação — a utilizar na distribuição de parâmetros do controlador — e um elevado grau de autonomia e robustez face a falhas de comunicação. As próximas secções irão detalhar a implementação de cada um dos subsistemas.

4.1 Pesquisa do modelo dinâmico inverso

A característica adaptativa deste sistema assenta numa correcta identificação da onda de excitação, razão pela qual é necessário encontrar uma relação, não necessariamente linear, entre a informação fornecida pelos sensores instalados no ISWEC e o binário induzido pela onda marítima, relação que se traduz num modelo dinâmico inverso. O modelo inverso adopta essa designação por permitir determinar o sinal de perturbação do modelo dinâmico directo a partir do seu sinal de saída.

O processo de aprendizagem da rede neuronal é constituído por várias fases que se encontram representadas na Figura 38.

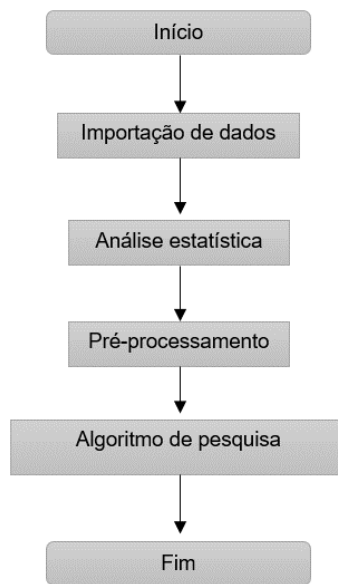


Figura 38 - Diagrama de processo

4.1.1 Importação de dados

A fase de importação de dados compreende a excitação do modelo não linear com controlador PD (proporcional derivativo), implementado em *Simulink*[®], com recurso a uma onda de referência durante um período de 300 segundos. Por uma questão de simplicidade foi escolhida a onda nº 1 do conjunto de oito ondas em análise para avaliação da arquitectura da rede neuronal. Dadas as características das ondas de excitação, as quais exibem períodos não inferiores a 5,5 segundos, optou-se por utilizar um tempo de amostragem de 0,05 segundos, obtendo-se um equilíbrio entre a representação precisa do sinal original e o volume de dados gerado — este último factor de elevada relevância no tempo de aprendizagem da rede neuronal.

Nesta fase em que é simulado o comportamento do modelo directo e se pretende extrair o conjunto de dados necessário à criação de um modelo inverso é essencial escolher um conjunto de variáveis cuja interacção faça sentido fisicamente na determinação do binário induzido pela onda. No caso em particular optou-se por extrair as seguintes variáveis:

- Velocidade angular do PTO ($\dot{\epsilon}$)
- Velocidade angular do flutuador ($\dot{\delta}$)
- Termo associado ao binário do PTO ($J \cdot \dot{\phi} \cdot \cos \epsilon$)
- Binário induzido pela onda (T_w)

A escolha das variáveis correspondentes às velocidades angulares foi motivada pelo conhecimento do efeito giroscópico, na medida em que o momento induzido pela onda marítima irá originar uma rotação no eixo que une o volante de inércia ao gerador eléctrico (PTO). O termo associado ao binário transmitido ao PTO, ainda que seja uma representação parcial, foi seleccionado tendo em conta que a sua composição define uma relação não linear entre a velocidade de rotação do volante de inércia e a rotação do eixo de transmissão do PTO.

4.1.2 Análise estatística

Uma vez concluída a importação de dados é iniciada a fase de análise estatística. A arquitectura de uma rede neuronal do tipo *NARX* requer uma criteriosa definição do número de atrasos à entrada (“*input delay*”) e número de atrasos na realimentação (“*feedback delay*”) para que se efective uma correcta aproximação do modelo dinâmico; contudo, sem uma análise preliminar aos dados a utilizar na rede neuronal, o processo de aprendizagem será um exercício de elevada incerteza, sendo possível que não se encontre o conjunto de números de atrasos adequado. A aplicação de uma análise estatística visa diminuir essa mesma incerteza, utilizando no algoritmo de busca de parâmetros da rede neuronal o conjunto de atrasos significativos.

A primeira abordagem passa por analisar a função de autocorrelação do sinal que se deseja obter via modelo inverso, ou seja o binário induzido pela onda (T_w), tentando encontrar um conjunto inicial de atrasos na realimentação que seja significativo. A função de autocorrelação, permitindo avaliar num sinal o grau de similaridade entre observações espaçadas temporalmente por “k” atrasos, poderá fornecer detalhes interessantes sobre a dinâmica do sistema, sendo definida matematicamente por [48]:

$$r_k = \frac{1}{c_0 \cdot T} \cdot \sum_{t=1}^{T-k} (y_t - \hat{y}) \cdot (y_{t+k} - \hat{y}) \quad (4.1)$$

A Figura 39 apresenta a função de autocorrelação (função *MATLAB autocorr*) do sinal relativo ao binário induzido pela onda de excitação. Numa primeira análise, verifica-se que o conjunto de atrasos significativos, excedendo o intervalo de confiança de 95% (a cor azul), é algo extenso. Contudo, é também evidente que existe uma forte correlação, de coeficiente igual ou superior a 0,8, nas observações com atraso não superior a 10, sugerindo que o atraso na realimentação se encontre nesse intervalo. Como tal este conjunto de atrasos integrará o espaço de parâmetros a testar no algoritmo de pesquisa.

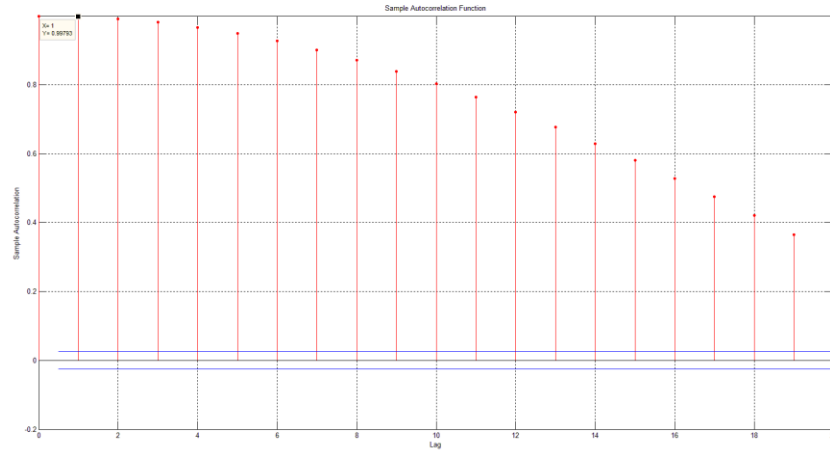


Figura 39 - Função de autocorrelação do binário induzido

Na selecção do conjunto de atrasos à entrada utilizar-se-á também uma função estatística, neste caso a função de correlação cruzada (função *MATLAB crosscorr*). Esta função mede a semelhança entre dois sinais cujas observações estão espaçadas no tempo. Considerando que o modelo dinâmico inverso possui três entradas e uma saída, o conjunto de atrasos à entrada da rede neuronal será proposto pela correlação entre cada uma das entradas do modelo e a respectiva saída.

A Figura 40 apresenta a função de correlação cruzada entre o sinal relativo à velocidade angular do flutuador ($\dot{\delta}$) e o sinal relativo ao binário induzido pela onda marítima (T_w). A função evidencia um conjunto de atrasos significativos relativamente extenso, sugerindo, ainda que seguindo um método puramente experimental, que um atraso à entrada no intervalo positivo não superior a 20 poderá ser adequado.

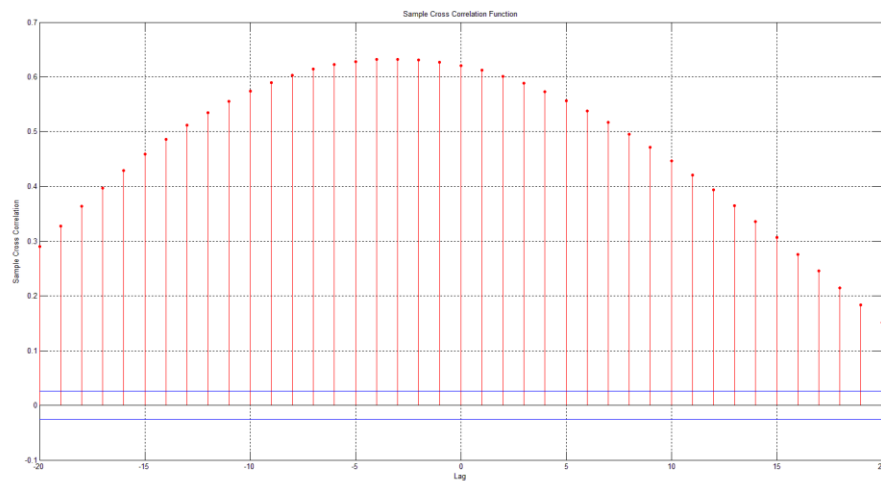


Figura 40 - Correlação cruzada entre $\dot{\delta}$ e T_w

A Figura 41 apresenta a função de correlação entre a velocidade angular do PTO e o binário induzido pela onda. As conclusões a extrair da função de correlação são análogas às da função anterior, nomeadamente no intervalo de atrasos significativos a considerar.

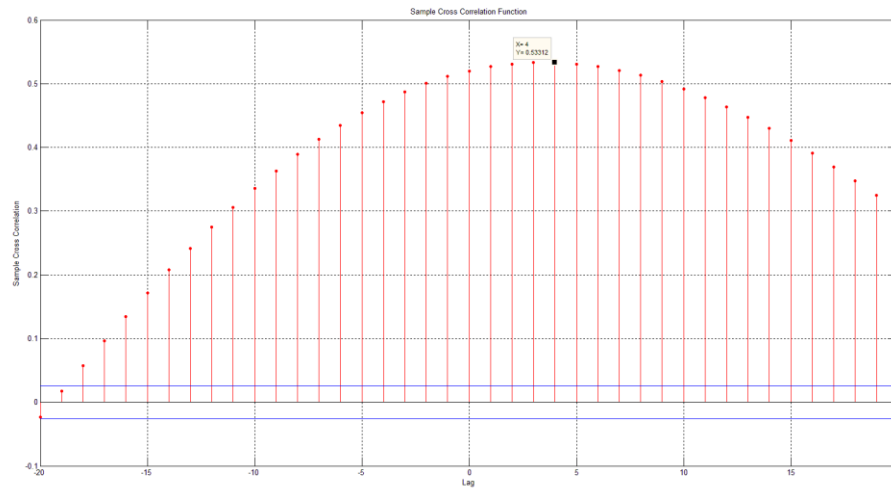


Figura 41 - Correlação cruzada entre $\dot{\epsilon}$ e T_w

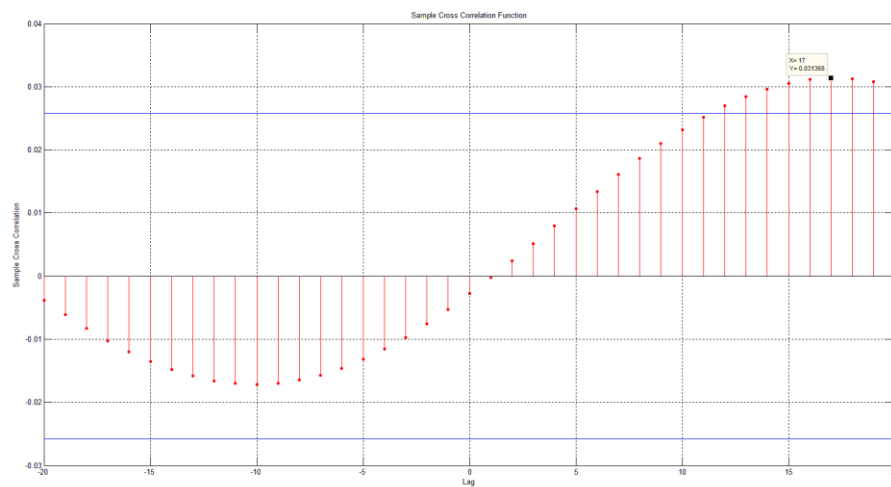


Figura 42 - Correlação cruzada entre termo do binário do PTO e T_w

A Figura 42 apresenta a função de correlação cruzada entre o termo do binário do PTO e o binário induzido pela onda. O conjunto de atrasos significativos, isto é, atrasos que excedem o intervalo de confiança a 95%, é neste caso de dimensão reduzida. Seguindo o método experimental, o conjunto de atrasos no intervalo entre doze e vinte é considerado suficiente.

Uma vez analisadas cada uma das funções de correlação cruzada e respectivos atrasos significativos, concluiu-se que o teste exploratório de atrasos à entrada no intervalo entre um e vinte seria o mais adequado, ainda que contenha a incerteza típica de uma heurística.

4.1.3 Pré-processamento de dados

A fase de pré-processamento de dados desempenha um papel de elevada relevância em sistemas inteligentes. Os dados em bruto, extraídos do modelo a extrapolar, nem sempre apresentam características ao nível de escala, amostragem ou tendência que conduzam a um treino eficiente da rede neuronal, como tal os dados de treino deverão ser sujeitos a um processo de regularização das características indesejáveis. A Tabela 6 apresenta informação estatística sobre os dados de treino não tratados. Neste caso é clara a existência de grande disparidade nas ordens de grandeza entre os sinais, nomeadamente entre o par de velocidades angulares e os sinais restantes.

Tabela 6 - Estatística de dados de treino

Sinal	Mínimo	Máximo	Intervalo de variação	Variância
Velocidade angular do flutuador	-1,67E-01	1,58E-01	3,25E-01	3,20E-03
Velocidade angular PTO	-4,35E-01	4,11E-01	8,47E-01	2,09E-02
Termo do binário do PTO	5,64E+05	6,00E+05	3,55E+04	3,39E+07
Binário da onda	-6,32E+05	6,29E+05	1,26E+06	5,08E+10

Considerando a dissemelhança de escala entre os sinais optou-se por aplicar uma técnica de normalização de dados, uma abordagem comum em sistemas inteligentes que promove, simultaneamente, um treino rápido e uma redução de qualquer tendência (“*bias*”) na rede neuronal de um sinal em detrimento de outro [49]. Do conjunto de métodos de normalização disponíveis seleccionou-se o “Z-Score”, um método que mede em cada observação a respectiva distância, em termos de desvio padrão, ao valor médio da série temporal. O coeficiente “ z_i ” da observação “ x_i ” de uma série temporal de média “ μ ” e desvio padrão “ σ ” é dado por:

$$z_i = \frac{(x_i - \mu)}{\sigma} \quad (4.2)$$

É importante notar que, uma vez treinada a rede neuronal com os dados normalizados e iniciada a sua implementação em ambiente real, a relação entrada-saída atenderá à normalização realizada no período de treino. Deste modo o sinal de entrada submetido à rede terá de ser normalizado, de acordo com a

equação acima, e o sinal de saída, correspondendo ao sinal extrapolado, terá de sofrer uma reversão da normalização segundo a média e desvio padrão verificados no treino.

4.1.4 Algoritmo de pesquisa

Uma vez realizada a normalização dos dados e identificado o conjunto de atrasos significativos, reúnem-se as condições para a execução do algoritmo de pesquisa. O algoritmo de pesquisa, cujo fluxograma se encontra ilustrado na Figura 43, tem como função testar várias combinações de atrasos de entrada e de realimentação na aprendizagem de redes neurais do tipo NARX, verificando qual o conjunto cuja rede apresenta o menor erro quadrático médio (EQM ou *MSE*) na extrapolação em anel fechado.

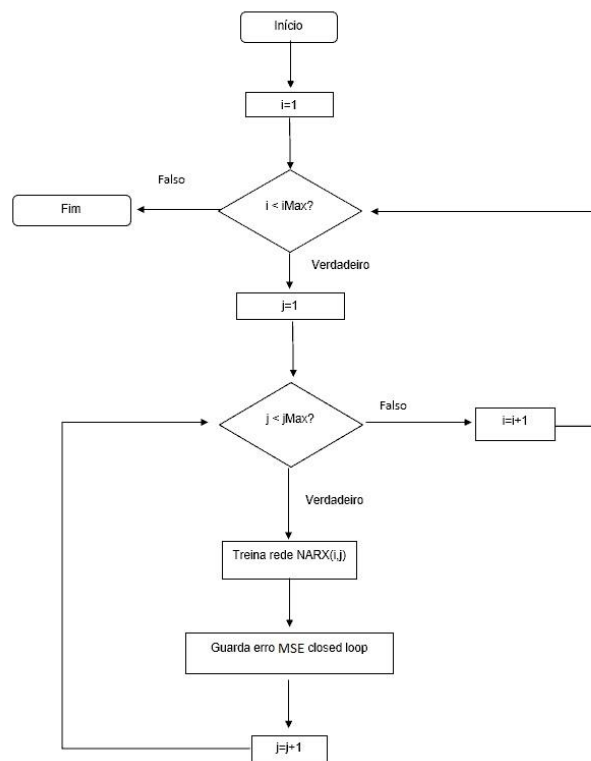


Figura 43 - Fluxograma do algoritmo de pesquisa

A primeira tarefa antes de executar o algoritmo é a definição do espaço de parâmetros de pesquisa. É importante salientar que um algoritmo de pesquisa exaustiva, caso possua um espaço de parâmetros relativamente extenso, poderá não conseguir resolver o problema em tempo útil, motivado pela morosidade no treino das sucessivas redes. O espaço de parâmetros a testar, que se indica na Tabela 7, foi definido de acordo com a análise estatística realizada anteriormente.

Tabela 7 - Parâmetros do algoritmo de pesquisa

Parâmetro	Valor mínimo	Valor máximo
Atraso à entrada (<i>input delay</i>)	1	20
Atraso na realimentação (<i>feedback delay</i>)	1	10

De acordo com o espaço de parâmetros, o algoritmo irá treinar e avaliar 200 redes neurais. A dimensão do espaço de parâmetros é, portanto, significativa e como tal irá resultar num longo tempo de execução do algoritmo, ainda que facilmente mitigável com recurso a soluções computacionais paralelas. As diferentes redes neurais treinadas pelo algoritmo apresentam uma configuração base, isto é, parâmetros que não serão alterados no decurso da pesquisa. A configuração base preconiza a utilização de uma camada oculta de dez neurónios e uma divisão de dados em que 70% se destinam ao treino, 15% à validação e outros 15% ao teste de desempenho. No treino da rede neuronal foi utilizada a função “Levenberg-Marquardt”. O processo de treino da rede neuronal foi implementado em MATLAB, recorrendo às funções “narxnet” e “train”, entre outras, incluídas no pacote de ferramentas “*Neural Network Toolbox*”.

A Figura 44 apresenta o espaço de resultados gerado pelo algoritmo de pesquisa, isto é, o erro quadrático médio (MSE) de extrapolação em anel fechado, para cada combinação de atrasos à entrada e na realimentação. Nesta figura é evidente que, para o modelo dinâmico em análise, as redes neurais com menor erro de extrapolação possuem apenas um atraso de realimentação, independentemente do atraso à entrada que tenha sido seleccionado.

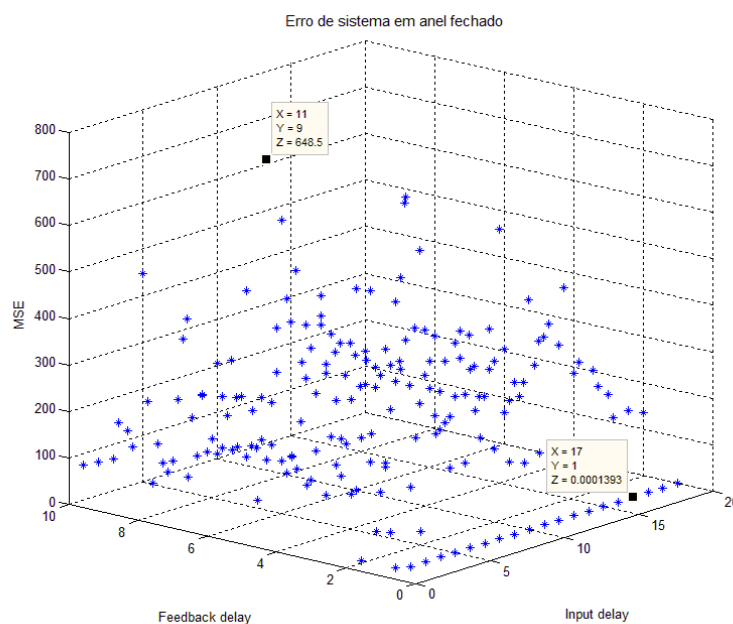


Figura 44 - Diagrama de espaço de resultados

A Tabela 8 apresenta um subconjunto do espaço de resultados, complementando assim, para fins ilustrativos, o gráfico de dispersão anterior. Neste subconjunto é evidente que o erro em anel fechado apresenta elevada disparidade, atendendo ao facto de estes valores corresponderem a dados normalizados. Verifica-se, igualmente, que o erro em anel fechado possui elevada sensibilidade ao atraso na realimentação, exibindo transições bruscas no espaço de resultados próximo da região definida pelo atraso de realimentação de valor unitário. Os exemplos na tabela a negrito reproduzem o comportamento de elevada sensibilidade descrito.

Tabela 8 - Subconjunto de espaço de resultados

Input delay (ID)	Feedback delay (FD)	MSE <i>open loop</i>	MSE <i>closed loop</i>
4	3	1.65E-08	231.2212963
5	1	9.24E-06	0.001610809
5	10	2.84E-11	447.5327308
6	1	1.08E-06	0.000318346
7	1	1.50E-06	0.000432928
7	2	8.50E-08	144.6079818
8	5	8.11E-10	149.2802059
14	4	2.01E-11	194.7094446
14	10	5.64E-11	146.9988782
15	1	1.55E-07	0.000157962
15	2	6.75E-08	105.1999462
15	3	7.69E-10	269.2489263
15	4	7.06E-11	544.6355399
17	2	4.78E-08	220.5573893
17	8	3.24E-11	166.1040625
17	9	5.88E-10	169.964698
17	10	3.87E-11	238.0209059
18	8	2.73E-12	503.4214482
20	5	4.43E-11	239.4315299
20	10	7.22E-12	131.6334026

4.1.4.1 Validação de desempenho

Uma selecção objectiva de parâmetros deve obedecer a um cumprimento de critérios claros, como tal estabeleceu-se que o erro quadrático médio de extrapolação em anel fechado não deverá exceder o valor de 0,01.

A Tabela 9 apresenta a configuração, respeitando o critério de erro, em termos de número de atrasos de cada rede e respectivos erros em anel aberto e fechado. Analisando a tabela, verifica-se facilmente que existe uma significativa homogeneidade no desempenho de aproximação ao modelo real. A configuração de rede com melhor desempenho, isto é, menor erro de aproximação, corresponde a uma rede com atraso de realimentação unitário e 17 atrasos à entrada.

Tabela 9 - Configurações seleccionadas

Input delay (ID)	Feedback delay (FD)	MSE	
		Open loop	Closed loop
4	1	5,63E-06	0,001082623
5	1	9,24E-06	0,001610809
6	1	1,08E-06	0,000318346
7	1	1,50E-06	0,000432928
8	1	2,30E-06	0,000689383
9	1	1,80E-07	0,000204564
10	1	1,95E-07	0,000191434
11	1	1,65E-07	0,000158666
12	1	3,04E-07	0,000271850
13	1	1,98E-07	0,000220996
14	1	2,00E-07	0,000193849
15	1	1,55E-07	0,000157962
16	1	1,66E-07	0,000180726
17	1	1,82E-07	0,000139291
18	1	2,06E-07	0,000155056
19	1	2,15E-07	0,000161417
20	1	3,06E-06	0,001036947

Após se ter determinado a arquitectura do modelo inverso em termos do número de atrasos, estudou-se a resposta do modelo no tempo. Ao contrário da abordagem de treino, em que apenas 15% dos dados foram utilizados na avaliação de desempenho, nesta fase optou-se por avaliar o desempenho de extrapolação da totalidade da onda.

A resposta no tempo do modelo inverso pode ser observada na Figura 45, estando o sinal de referência colorido a azul e o sinal de resposta da rede a vermelho. Uma análise visual indica que a extrapolação do modelo se realiza de modo bastante satisfatório, apresentando uma rápida convergência com o sinal de referência.

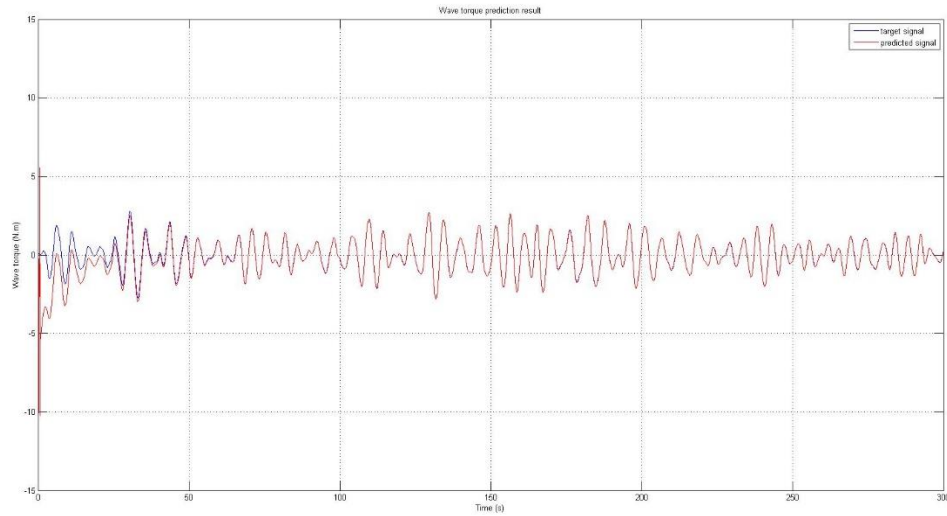


Figura 45 - Resposta no tempo do modelo inverso treinado para uma única onda de referência

O desempenho do modelo inverso, treinado para uma única onda de referência, foi avaliado através da métrica VAF (*variance accounted for*), a qual indica quão próxima é uma estimativa do sinal de referência, sendo que um valor de 100% indica uma aproximação perfeita e 0% indica o seu oposto. A expressão da VAF é dada por (4.3).

$$VAF = \left(1 - \frac{var(y - \hat{y})}{var(y)} \right) \times 100\% \quad (4.3)$$

A Tabela 10 apresenta os valores VAF do sinal estimado em comparação com o sinal de referência, em distintos períodos.

Tabela 10 - VAF do modelo inverso

Período (s)	VAF (%)
0-300	67,41
10-300	96,66

O valor VAF de 67,41 % indica uma aproximação razoável, ainda que o gráfico indique que decorridos cinquenta segundos a extrapolação é praticamente indistinguível do sinal de referência. Analisando a resposta inicial do modelo inverso, verifica-se que nos primeiros instantes é produzida uma oscilação de elevada amplitude (± 10) e duração de aproximadamente um segundo. O transiente observado, pela sua variabilidade face à média, penalizou severamente o resultado da VAF. Aplicando a mesma métrica ao

sinal expurgado do transiente, obtém-se um valor VAF de quase 97%, indicando uma aproximação de elevada qualidade. O conhecimento do comportamento dinâmico do modelo, nomeadamente em regime transiente, permite que se adoptem estratégias que promovam o sucesso das previsões, neste caso considerando um tempo de estabelecimento mínimo para validação da previsão, que neste caso seria de dez segundos.

4.1.5 Treino da rede para um sinal composto

Alcançado o sucesso no desenvolvimento de um modelo inverso do sistema ISWEC, com recurso a uma rede neuronal, a próxima fase visa alargar a capacidade de extrapolação da rede a um conjunto de oito ondas marítimas.

A ampliação da capacidade de generalização da rede, de uma única onda para um universo de oito ondas, requer um conhecimento mais profundo da resposta do modelo dinâmico. Como tal, foi construído um sinal composto por oito fracções do binário induzido por ondas marítimas distintas, com uma duração média de 124 segundos por fracção. Na aglutinação das ondas individuais, realizada por ordem crescente do índice de onda, foram respeitadas as condições de continuidade da função e da primeira derivada, procurando que o sinal composto apresentasse transições tão fluidas quanto possível. Seguidamente, o sinal composto foi utilizado na excitação do modelo não linear do conjunto ISWEC e PD, extraíndo-se os sinais relativos às velocidades angulares do flutuador e do PTO ($\dot{\delta}$ e $\dot{\varepsilon}$), bem como do termo associado ao binário transmitido ao PTO.

A Figura 46 apresenta a evolução temporal dos sinais envolvidos na simulação, sendo notória a intensidade crescente da resposta do ISWEC às sucessivas ondas, que, conforme indicado na Tabela 5, vêem modificadas as suas características dinâmicas.

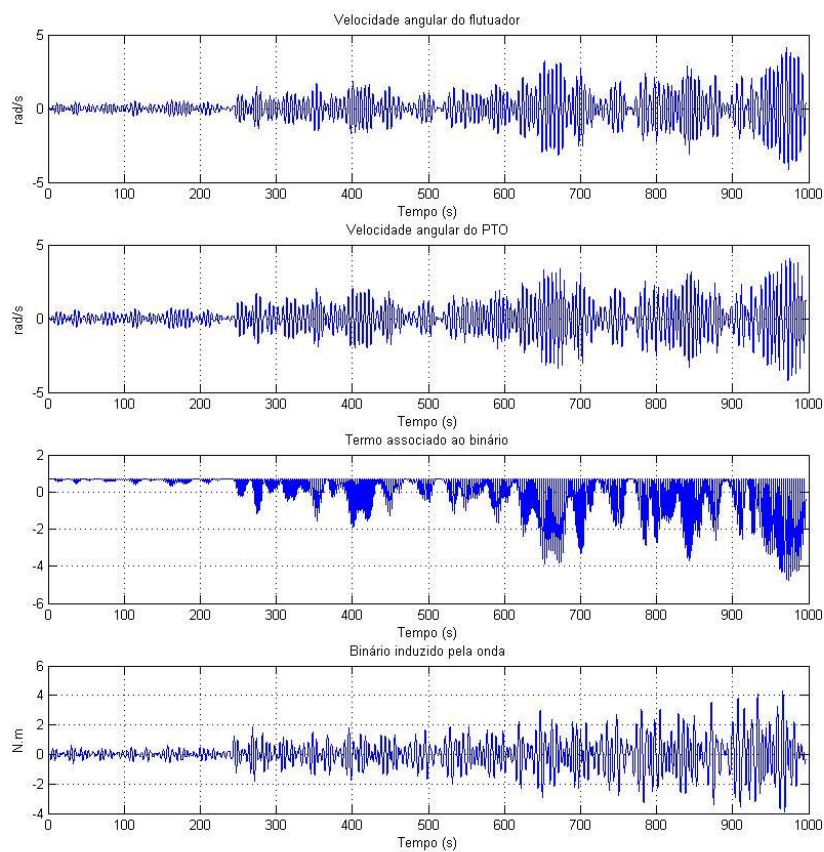


Figura 46- Sinais de saída (gráficos 1 a 3) do modelo directo. A entrada (gráfico 4) é um sinal composto por oito ondas

O treino da rede foi realizado seguindo a configuração utilizada anteriormente na extrapolação de uma única onda, mantendo-se o mesmo método de normalização e divisão de dados. A rede apresenta uma camada oculta com dez neurónios, 17 atrasos à entrada e um atraso na realimentação. A Figura 47 ilustra a configuração da rede.

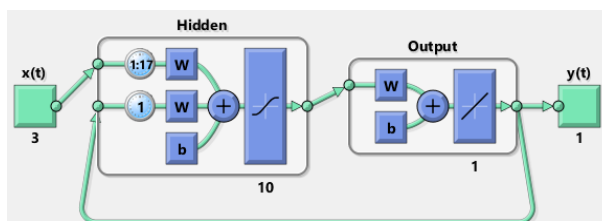


Figura 47 – Configuração da rede neuronal

A Figura 48 apresenta a resposta da rede neuronal na extrapolação do sinal composto. O gráfico superior indica visualmente quão próxima é a extrapolação do sinal verdadeiro. A natureza do sinal composto gera alguma entropia na interpretação visual, motivada pela elevada concentração de pontos coloridos que sinalizam os conjuntos de treino, validação e teste. O gráfico inferior apresenta o erro de extrapolação ao longo do sinal de 996 segundos. Neste, é possível verificar que o erro máximo é de 0,15 e que o erro médio tem fraca expressão. A localização dos erros de extrapolação segue um padrão particular, dividindo o sinal em oito fracções. De facto, os erros de extrapolação de maior expressão surgem nas transições entre fracções de ondas diferentes, evidenciando as limitações da composição do sinal. Globalmente, a rede neuronal apresenta um desempenho satisfatório (ver anexo III), exibindo um erro médio quadrático em anel fechado de apenas 0,0011 no referencial normalizado.

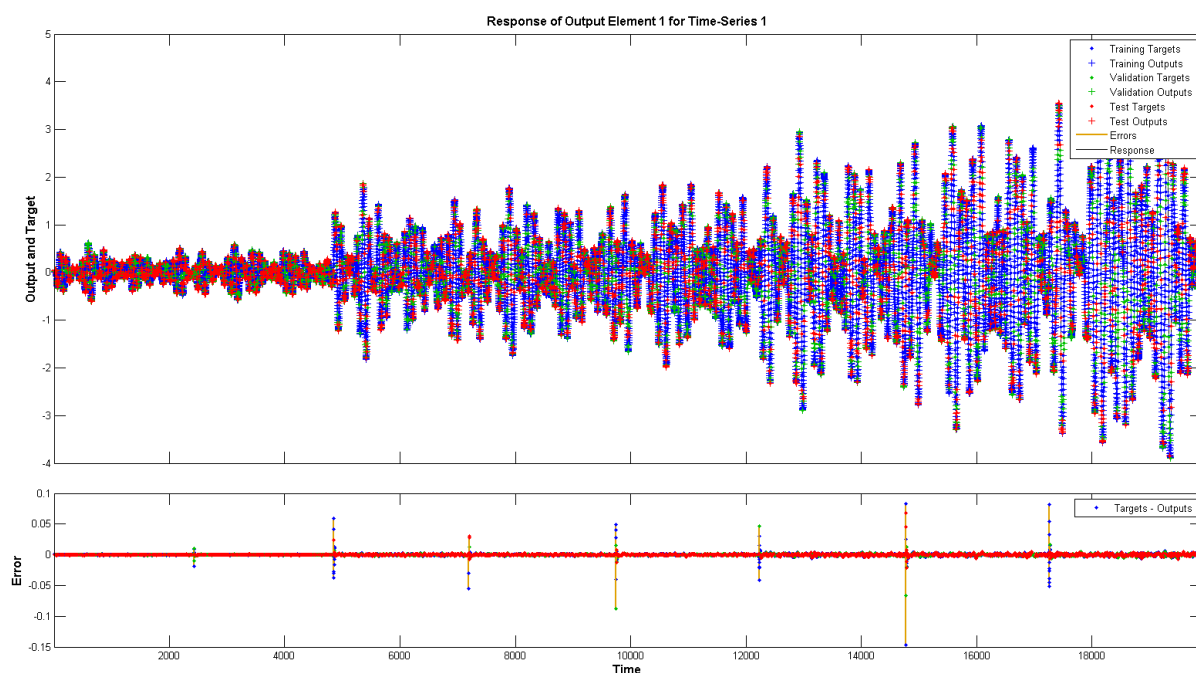


Figura 48 - Resposta da rede neuronal ao sinal composto por oito ondas

Uma vez finalizado o treino da rede com o sinal composto, a mesma deverá ser capaz de extrapolar cada uma das oito ondas separadamente. Esta secção irá avaliar o desempenho da rede neuronal para cada uma das oito ondas.

Na fase de teste da rede optou-se por utilizar um conjunto de dados com uma duração total de 200 segundos, em que cerca de 60% dos dados foram utilizados na aprendizagem da rede, correspondentes aos primeiros 120 segundos de sinal.

A Tabela 11 apresenta o valor VAF obtido na extrapolação de cada tipo de onda. Considerando que valores VAF próximos de 100% indicam uma aproximação ideal, é evidente que o desempenho é globalmente bastante satisfatório, obtendo-se um valor VAF médio de 90,65%. A tabela indica também o valor VAF correspondente aos últimos 80 segundos do sinal estimado pela rede. Esta opção teve como objectivo aferir a qualidade da aproximação a longo prazo, expurgando qualquer fenómeno transiente nos primeiros segundos de extrapolação. Os resultados obtidos neste último período sugerem, pela melhoria de desempenho, que o período inicial poderá conter um efeito transiente que penalize, ainda que ligeiramente, o valor VAF.

A análise à estimativa da onda número 1, ilustrada na Figura 49, evidencia a presença de uma oscilação, de curta duração e elevada amplitude, nos primeiros instantes (assinalada no gráfico por uma seta). Este fenómeno já fora identificado na rede treinada para uma única onda, mantendo-se as mesmas conclusões ao nível do tempo mínimo de estabelecimento, que se observa ser de aproximadamente dez segundos.

Tabela 11 - Qualidade da extrapolação das 8 ondas

Onda	VAF (%)		variação pp.
	Duração total	Últimos 80 s	
1	93.20	96.71	3.51
2	92.67	97.33	4.66
3	94.17	97.72	3.55
4	93.53	98.87	5.34
5	88.37	91.85	3.48
6	89.53	97.68	8.15
7	91.27	95.38	4.11
8	82.48	90.94	8.46

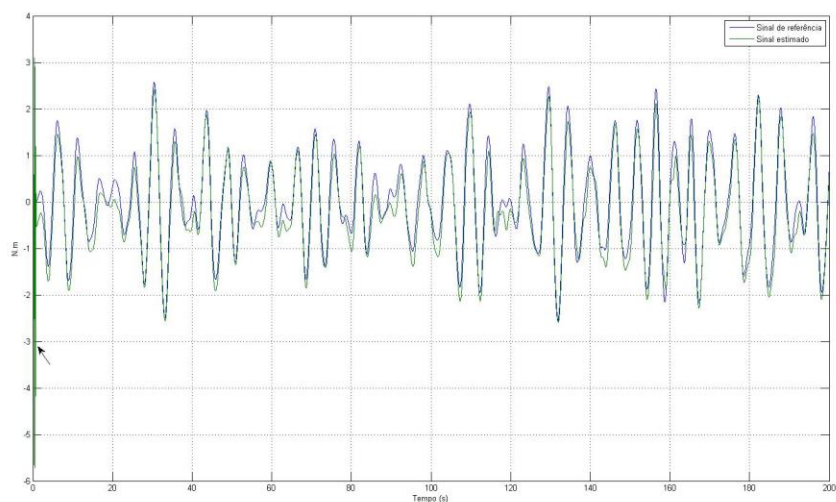


Figura 49 - Estimativa da onda 1 gerada por rede treinada por um sinal composto

4.2 Classificador de onda

De acordo com a configuração proposta, o controlador deverá ser capaz de ajustar os seus ganhos ao estado do mar actual. Para tal, será necessário implementar um algoritmo que analise o binário induzido no ISWEC, extrapolado pelo modelo dinâmico inverso, e estime com o menor erro possível qual a classe de onda marítima actual. Os algoritmos de aprendizagem automática (*“machine learning”*) são recursos valiosos na resolução deste género de problema — classificação de dados — encarregando o operador de escolher qual o conjunto relevante de características (*“features”*) a extrair dos dados e qual o algoritmo de resolução a utilizar, fases a pormenorizar nas próximas secções.

4.2.1 Extracção de características

A classificação de dados, segundo uma abordagem clássica de aprendizagem automática supervisionada, requer a extracção de características de forte relevância que permitam distinguir as diversas classes. Como tal, optou-se por extrair, simultaneamente, características do sinal no domínio do tempo e frequência, aplicando-se um método de janela deslizante, com sobreposição de 50%, extraindo em cada janela um conjunto de métricas potencialmente relevantes para o problema de classificação. Deste modo, o algoritmo de aprendizagem terá como representação de um sinal um conjunto de partições e as suas características individuais. Atendendo à natureza do sinal a classificar — uma série temporal marcada por componentes de baixa frequência — optou-se por utilizar uma janela com uma duração de 50 segundos, traduzindo um compromisso entre a captura de características relevantes e uma detecção ágil de alterações do estado do mar. A Figura 50 ilustra a aplicação do método de janela deslizante ao longo de um sinal composto pela aglutinação dos 8 tipos de onda.

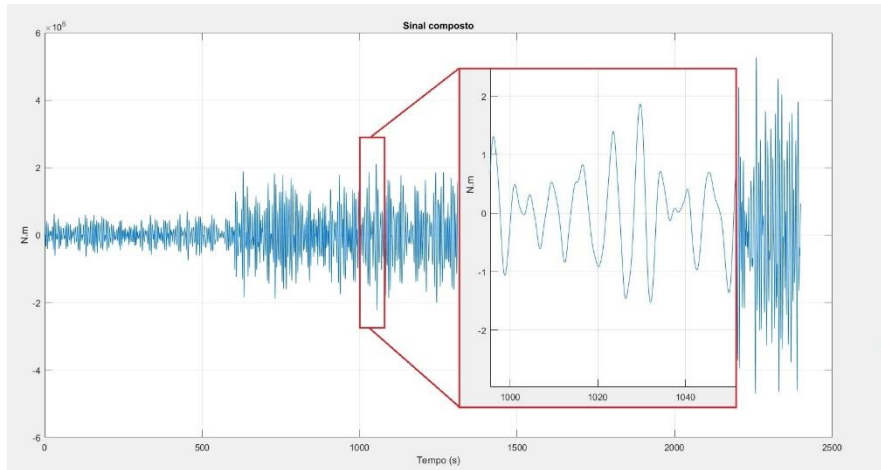


Figura 50 - Método de janela deslizante

A escassez de dados é uma característica que se manifesta em vários problemas de classificação, não sendo este caso em particular uma exceção. Os dados relativos às 8 ondas de excitação do sistema ISWEC apresentam uma duração relativamente reduzida, apenas 5 minutos por onda, o que, atendendo à natureza parcimoniosa de uma onda marítima, poderá limitar a capacidade de extrapolação do algoritmo em ambiente real. A aplicação do método de janela deslizante, segundo a configuração indicada, permitirá extrair do sinal composto várias características provenientes de 88 janelas distintas — 11 janelas por cada classe de onda — o que não sendo um conjunto de dimensão ideal, se admite ser suficientemente equilibrado e relevante para a resolução do problema em análise.

A definição de características a extrair dos dados não é um processo unívoco, atribuindo-se ao mesmo a qualidade de fase tipicamente mais complexa num projecto de aprendizagem automática [50]. Tendo-se optado por uma análise combinada no domínio do tempo e da frequência, é proposto o seguinte conjunto de características:

1. Valor RMS (*“root mean square”*)
2. Valor médio
3. Desvio padrão
4. Intervalo de variação
5. Valor máximo
6. Valor mínimo
7. Curtose (*“kurtosis”*)
8. Assimetria (*“skewness”*)
9. Frequência média
10. Largura de banda (3 dB)

O conjunto proposto de características visa descrever o sinal de excitação nas suas componentes energética e espectral, destacando também diversos atributos amostrais extraídos no domínio do tempo. No domínio da frequência, destacam-se os indicadores “Frequência média” e “Largura de banda”, indicadores que permitirão auxiliar a distinguir os diversos tipos de onda dado que as especificações espectrais são conhecidas (ver Tabela 5). É importante, contudo, notar que a proximidade entre classes de onda, no que concerne à frequência dominante, poderá introduzir alguma entropia na classificação do sinal. A introdução do indicador “Valor RMS”, traduzindo a energia do sinal, deverá reduzir a incerteza associada aos indicadores espectrais, dado que ondas de classe superior apresentam valores RMS também mais elevados. Os restantes indicadores caracterizam, sobretudo, a forma, localização e dispersão dos dados, métricas de relativa simplicidade e facilmente observáveis.

É importante notar que a anterior definição do conjunto de características representa apenas um ponto de partida, ou seja, só numa fase posterior de treino do algoritmo se irá apurar a existência de possíveis correlações entre características bem como a efectiva relevância das mesmas no problema de classificação.

4.2.2 Algoritmo de classificação

No domínio dos sistemas de aprendizagem automática supervisionada existem vários algoritmos de classificação, como por exemplo: redes neuronais, árvores de decisão, K-NN (“*k-nearest neighbor*”), SVM (“*support vector machines*”), entre outros. No caso em particular optou-se por aplicar um algoritmo SVM com recurso ao pacote de ferramentas MATLAB® “*Statistics and Machine Learning Toolbox*”.

O espaço de dados do classificador é constituído pelo conjunto de características extraídas em cada uma das janelas deslizantes e correspondentes etiquetas de identificação das ondas marítimas, resultando numa matriz de dimensão 88 por 10. A escassez de dados, evidente no reduzido número de observações por classe, apresenta uma condicionante ao nível dos conjuntos de treino e validação, cuja definição deverá garantir que o conjunto de treino é suficientemente representativo da realidade a modelar e que, simultaneamente, o conjunto de validação permite ao classificador generalizar o seu conhecimento, ou seja, evitar ao mínimo qualquer fenómeno de “*overfitting*”. O recurso a um mecanismo de validação cruzada do tipo “K-Fold” permitirá minorar o efeito da escassez de dados na aprendizagem do algoritmo, utilizando uma proporção $(K-1)/K$ dos dados disponíveis na fase de treino e utilizando os restantes dados na fase de validação [22, p. 32]. A Figura 51 ilustra o funcionamento do método de validação cruzada em 4 partições ($K=4$) em que os blocos coloridos correspondem aos dados de validação e os restantes aos dados de treino. A cada iteração é aferido o desempenho de classificação, sendo o desempenho global

obtido através da média de resultados de cada iteração. No caso em estudo optou-se por aplicar uma validação cruzada de 5 partições (K=5).

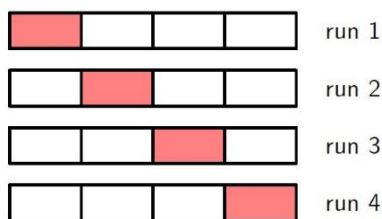


Figura 51 - Técnica de validação cruzada [22]

Na fase de treino do classificador foram implementados seis modelos SVM com diferentes configurações de função de núcleo (*kernel function*):

- Linear;
- Quadrática;
- Cúbica;
- Gaussiana fina;
- Gaussiana média;
- Gaussiana grosseira;

A aplicação de uma função de núcleo não linear a um modelo SVM tem como objectivo aumentar o desempenho de classificação, mapeando dados não linearmente separáveis num novo espaço onde a distinção entre classes seja mais evidente [20, p. 423]. As seis configurações propostas distinguem-se essencialmente ao nível do tipo de função de núcleo (Linear, polinomial, gaussiana) e dos seus parâmetros, razão pela qual a função de núcleo gaussiana possui um qualificativo adicional (fina, média, grosseira). Este qualificativo traduz a capacidade de o classificador distinguir as classes, dependendo da definição do parâmetro de escala σ . Este parâmetro é inversamente proporcional à flexibilidade do modelo, como tal, um valor elevado traduz uma separação grosseira das classes. No caso em estudo os parâmetros correspondem a: i) fina: $\sigma=0,79$; ii) média: $\sigma=3,2$; iii) grosseira: $\sigma=13$. Os seis modelos SVM partilham o mesmo valor de penalidade “C”, tendo este valor unitário.

Os modelos SVM foram implementados utilizando a função “*fitcecoc*” [27], a qual permite a resolução de problemas multi-classificação recorrendo a um conjunto de classificadores binários, modelo designado por “*error-correcting output code multiclass model (ECOC)*” [27]. Como método de multi-classificação utilizou-se o “*One-vs-One*”, o qual utiliza uma combinação de $\binom{n}{2}$ classificadores binários de “n” classes, resultando, no caso em estudo ($n=8$), em 28 classificadores distintos. Os dados de treino e validação dos modelos SVM foram normalizados.

Nesta mesma fase optou-se por realizar uma selecção do conjunto de características, avaliando o desempenho de classificação dos vários modelos para os dois conjuntos propostos. A Tabela 12 apresenta a constituição dos dois conjuntos de características.

A Tabela 13 sumariza os valores de desempenho de classificação de cada modelo SVM para os dois conjuntos de características, valores que resultam da média da sensibilidade de cada classe, ou seja, da probabilidade de se obter uma classificação verdadeira em cada uma. Comparando os dois conjuntos é evidente que o conjunto “B” apresenta um valor de desempenho com menor variabilidade e um valor médio superior. Os modelos mais simples (linear, quadrático) apresentam um desempenho razoavelmente satisfatório — sensibilidade global próxima dos 70% — não inibindo a sua utilização em ambiente de produção.

A utilização de um modelo com uma função de núcleo linear é especialmente adequada a casos em que o conjunto de observações é reduzido e o número de características é elevado, casos em que a acrescida flexibilidade de modelos com funções de núcleo não lineares (gaussianas, polinomiais) poderá conduzir a uma diminuição da capacidade de generalização (“*overfitting*”) [21]. Em ambos os conjuntos é no modelo SVM com uma função de núcleo gaussiana média que é obtido o valor máximo de desempenho, decidindo-se que o modelo treinado para o conjunto “B” é o mais adequado à extrapolação da onda marítima em virtude de apresentar o melhor desempenho, de aproximadamente 80%, e um menor número de características.

Tabela 12 - Conjuntos de características propostos (presente: X; ausente: ---)

Característica	Conjunto A	Conjunto B
1. Valor RMS	X	X
2. Valor médio	X	---
3. Desvio padrão	X	X
4. Intervalo de variação	X	X
5. Valor máximo	X	X
6. Valor mínimo	X	X
7. Curtose (“ <i>kurtosis</i> ”)	X	---
8. Assimetria (“ <i>skewness</i> ”)	X	---
9. Frequência média	X	X
10. Largura de banda (3 dB)	X	X

Tabela 13 - Resultados de classificação

Função de núcleo	Desempenho global	
	Conjunto A	Conjunto B
Linear	68.2%	71.6%
Quadrática	71.6%	75.0%
Cúbica	69.3%	70.5%
Gaussiana fina	40.9%	72.7%
Gaussiana média	72.7%	78.4%
Gaussiana grosseira	43.2%	47.7%

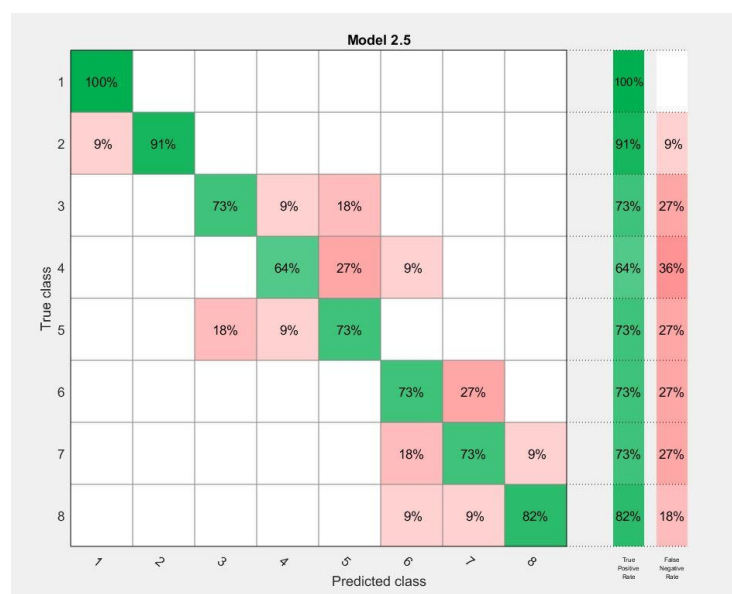


Figura 52 - Matriz de confusão SVM. Eixo vertical: Classe verdadeira; Eixo horizontal: Classe extrapolada

A avaliação de desempenho de um classificador é frequentemente auxiliada por uma matriz de confusão, a qual indica, em cada classe, o rácio de falsos negativos (*“false negative rate”*) e o rácio de verdadeiros positivos (*“true positive rate”*) — também designado por sensibilidade, ou simplesmente, a probabilidade de um conjunto de determinada classe ser detectado correctamente. A Figura 52 ilustra a matriz de confusão do modelo seleccionado, indicando, a cor verde, os rácios de verdadeiros positivos e, a cor vermelha, os rácios de falsos negativos. Observando os valores na diagonal, conclui-se que a sensibilidade do modelo é, em termos médios, superior a 70%, atingindo um desempenho particularmente elevado — valores superiores a 90%— na classificação de ondas de classes 1 e 2. A classificação da onda de classe

4 indica um menor desempenho, classificando incorrectamente 36% das observações. É importante notar que todas as observações incorrectamente classificadas — falso negativo — pertencem a uma classe adjacente ou muito próxima da classe extrapolada, sugerindo que as características escolhidas são suficientemente relevantes na separação superficial de classes, não conseguindo, em determinadas observações, realizar uma distinção mais fina de classes adjacentes.

4.3 Implementação do sistema multi-agente

Nesta secção serão apresentadas as diversas fases da implementação do sistema multi-agente através de uma plataforma computacional. Numa primeira fase serão introduzidas as principais funcionalidades da plataforma e a sua aplicabilidade no projecto em estudo. A fase seguinte é dedicada à modelação do sistema, identificando-se o conjunto de objectos que constituem o sistema multi-agente. A fase que sucede é dedicada à camada de comunicação entre o sistema multi-agente e os dispositivos ISWEC. A problemática de sincronização de processos é abordada na última fase.

4.3.1 Visão geral

No problema em estudo procura-se implementar um sistema que permita coordenar a comunicação entre vários dispositivos ISWEC dispostos num ambiente marítimo. Num cenário base, ilustrado na Figura 53, é considerado que o conjunto de dispositivos contém um agente especializado na detecção e classificação da onda marítima observada, informação que será utilizada na definição do conjunto de parâmetros do controlador PD presente em cada dispositivo que se afigure mais adequado. O conjunto de parâmetros adequado para cada tipo de onda marítima foi determinado no estudo [45]. Cada um dos dispositivos ISWEC deverá receber a informação acerca dos parâmetros através do agente classificador, tendo que empreender uma conversação negociada, ou seja, prevendo-se a possibilidade de falha (comunicação, avaria, etc...). Um sistema artificial multi-agente permite implementar esta interacção entre os vários dispositivos ISWEC.

Na implementação do sistema multi-agente optou-se por utilizar a plataforma computacional JADE como solução “middleware” entre os dispositivos ISWEC e o agente classificador. Neste caso em particular a plataforma é utilizada sobretudo em funções de comunicação e negociação, não implementando directamente nenhum mecanismo de inferência lógica e/ou inteligência artificial baseado no paradigma BDI ou similar. A plataforma JADE, sendo uma solução de grande popularidade desenvolvida em Java e assente no paradigma de uma linguagem orientada a objectos, facilita a compreensão e realização de um sistema multi-agente na medida em que a sua tradução num conjunto de objectos é clara.

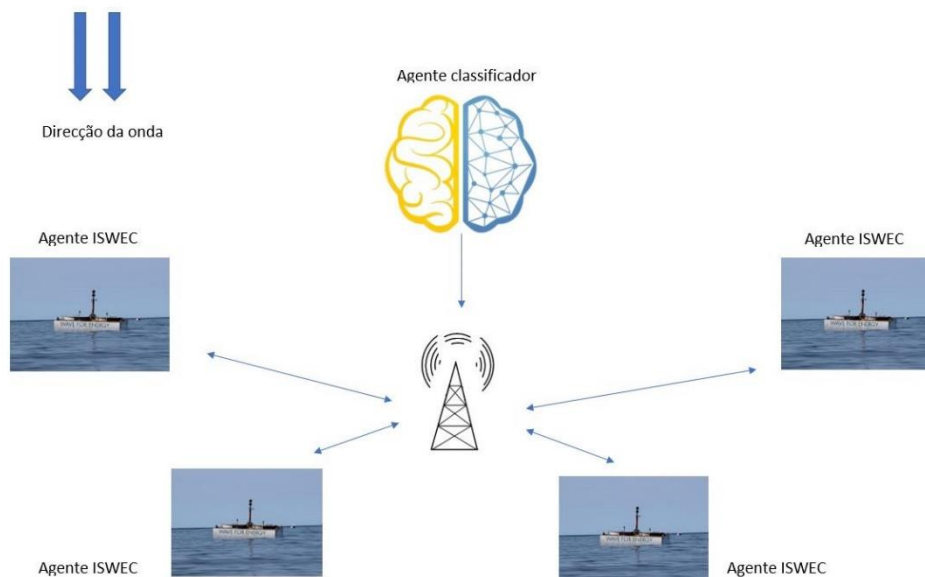


Figura 53 - Representação do sistema multi-agente real. Fonte das figuras ISWEC e classificador: [51], [52]

A plataforma JADE é disponibilizada como uma biblioteca de métodos e objectos com uma interface simples, facilmente integrável num qualquer ambiente de desenvolvimento (IDE) tal como o *NetBeans IDE* ou *Eclipse*. Na plataforma, cada agente é representado por um objecto com a sua própria “*Thread*” — um conceito computacional que prevê o desacoplamento de aplicações de “*software*” em processos autónomos, requerendo uma criteriosa gestão de concorrência e sincronismo dos mesmos — em que nenhum outro agente lhe pode induzir qualquer alteração que não seja realizada através de mensagens e aceite pelo destinatário [30, p. 30]. O sistema de mensagens no JADE é realizado de forma assíncrona, dispensando qualquer preocupação temporal, prevendo-se também que exista um serviço de subscrição em que os agentes registam a sua intenção de receber uma notificação automática após determinado evento [30]. O sistema de mensagens prevê também os casos em que as mensagens não são endereçadas explicitamente a um agente — caso em que é utilizado um identificador único — mas são remetidas a um agente anónimo que preste uma determinada tarefa, sendo essa tarefa publicada num serviço específico designado por “páginas amarelas”. A plataforma disponibiliza também um serviço de “páginas brancas”, designação dada ao registo dos agentes em execução na plataforma. Estes dois serviços são implementados com recurso a agentes distintos:

- Agente “AMS” (“*Agent Management System*”): O agente tem como responsabilidade principal a gestão da plataforma, mantendo um registo de todos os agentes que nela existam — “páginas brancas” — e atribuindo a cada um o respectivo identificador único global (“*AID-Agent identifier*”) [30, p. 34].

- Agente “DF” (“*Directory Facilitator*”): O agente é responsável pelo serviço de “páginas amarelas” — registando os serviços prestados pelos agentes — e pelo serviço de subscrição. Tomando como exemplo o caso em estudo, se cada um dos agentes ISWEC subscrever o serviço de classificação de onda marítima prestado pelo agente de detecção, este irá automaticamente notificar por mensagem cada um dos agentes sempre que existir uma actualização no tipo de onda detectada. Relativamente ao serviço de “páginas amarelas”, tomando o mesmo exemplo de aplicação, não é necessário que cada agente ISWEC conheça explicitamente a identificação do agente de detecção, bastando procurar na lista pública um agente que ofereça esse serviço. Este comportamento traz claros benefícios à adaptabilidade do sistema, exigindo-se apenas que cada agente reconheça e divulgue a sua função no sistema.

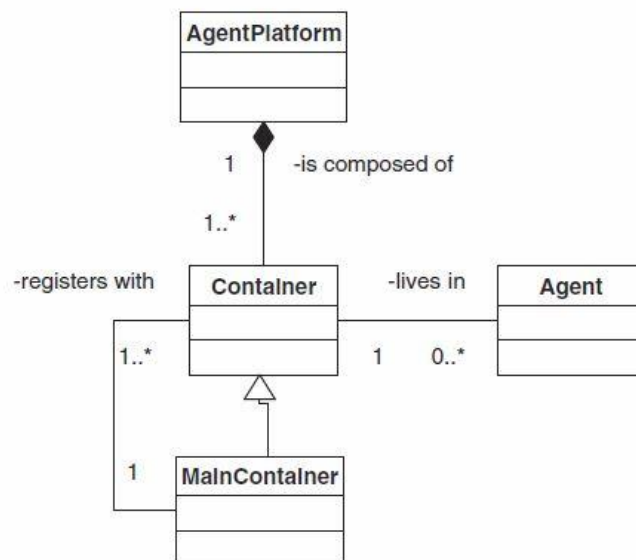


Figura 54 - Diagrama UML da arquitectura do JADE [30]

O diagrama em UML da arquitectura da plataforma JADE, ilustrado na Figura 54, permite uma melhor compreensão da interacção dos objectos da plataforma, sendo aqui representados pelas suas classes. No diagrama é possível identificar as seguintes classes: Plataforma (“*AgentPlatform*”), Repositório (“*Container*”), Repositório principal (“*MainContainer*”) e Agente (“*Agent*”). Observando o diagrama conclui-se que a plataforma de agentes obriga à constituição de pelo menos um repositório, sendo este registado num único repositório principal. De acordo com o diagrama é também evidente que a classe “Repositório” é uma classe base da classe “Repositório principal”, em que esta última herda as características da primeira. A interacção entre um agente e a plataforma de agentes é facilmente explicável pelo diagrama. Os objectos da classe “agente” existem apenas no âmbito de um objecto de classe “repositório” sendo,

contudo, possível ter um objecto “repositório” sem qualquer agente atribuído. A mesma plataforma multi-agente pode abranger várias máquinas remotas, tendo cada uma o seu repositório, mantendo-se uma camada de comunicação comum e o acesso aos agentes DF e AMS, estando estes localizados numa das máquinas.

Além das funcionalidades de modelação, a plataforma JADE disponibiliza também ferramentas, sob a forma de agentes, que através de uma interface gráfica simples permitem realizar várias tarefas de administração do sistema e testar o seu comportamento. As ferramentas a destacar são as seguintes: JADE RMA (*“Remote monitoring agent”*), *DummyAgent*, *Sniffer Agent*.

A ferramenta “JADE RMA” possui, essencialmente, funcionalidades ao nível da administração do sistema multi-agente. Entre outras tarefas, é possível verificar que agentes se encontram em execução na plataforma, terminar ou suspender a execução de um agente e migrar agentes entre repositórios. Esta ferramenta permite, também, iniciar outras ferramentas de gestão (ex: *DummyAgent*, *Sniffer Agent*).

A ferramenta *DummyAgent* designa um agente utilizado apenas para envio de mensagens FIPA-ACL a um ou mais agentes que existam na plataforma, permitindo assim validar a reacção do agente de destino [30]. As mensagens permitem uma elevada personalização, desde os actos comunicativos (ver Tabela 2) aos protocolos de conversação.

A ferramenta *Sniffer Agent* é utilizada essencialmente na análise da informação trocada entre um conjunto de agentes. A ferramenta, além de possibilitar o acesso às mensagens trocadas, permite também analisar um diagrama temporal que reflecta a interacção entre os agentes. Esta funcionalidade é especialmente útil durante o desenvolvimento de um sistema, permitindo perceber se a dinâmica observada entre os agentes corresponde ao comportamento desejado. Nas secções posteriores ter-se-á a oportunidade de observar a aplicação da ferramenta ao sistema multi-agente a implementar. No diagrama temporal, os eventos de conversação são apresentados por uma seta entre os agentes interlocutores, sendo a mensagem classificada de acordo com o seu acto comunicativo.

4.3.2 Modelação

Na modelação do sistema multi-agente optou-se por definir dois tipos de agente: um agente genérico e um agente central. O agente genérico é responsável por um único conversor ISWEC, tendo assim a capacidade de alterar os parâmetros do respectivo controlador PD. As acções realizadas por este agente são resultado da sua interacção com o agente central. O agente central tem como função fornecer, a todos os agentes que o solicitem, a previsão da onda marítima, permitindo que todos os conversores presentes no parque de ondas tenham os seus parâmetros definidos de forma a se obter o melhor desempenho na produção de electricidade. O agente central recebe a informação de previsão directamente de um sistema de aprendizagem automática, englobando uma rede neuronal NARX e um classificador SVM.

A Figura 55 apresenta um diagrama que ilustra a comunicação entre os agentes no sistema proposto. No diagrama, os eventos são representados por setas que unem os agentes em conversação. A noção de tempo é traduzida pela ordem dos eventos, sendo tão mais tardios quanto mais distantes se encontrarem do cabeçalho com o nome dos agentes. No diagrama, cada evento é identificado pelo acto performativo que realiza (consultar Tabela 2).

No início da operação do sistema multi-agente, o agente central realiza o seu registo no agente DF (*“Directory facilitator”*) ou serviço de páginas amarelas. Neste registo é publicado que o agente central, identificado univocamente pelo seu AID (*“Agent identifier”*), realiza um serviço de previsão de ondas. O agente genérico, representando qualquer conversor ISWEC no sistema, inicia a sua operação procurando, junto do agente DF, um agente que disponibilize serviços de previsão de ondas. Esta operação compreende o envio de uma mensagem de acto performativo *“Request”*, em que se inclui o tipo de serviço a procurar (*“wave forecast”*), e uma mensagem de resposta com acto performativo do tipo *“Inform”* em que é devolvido o AID de um agente que satisfaça o serviço. Uma vez obtida a identificação do agente central, o agente genérico deverá empreender com o último, em intervalos regulares, uma conversação em que solicita a previsão de onda marítima. A conversação deverá prever situações de falha do agente central, seja ao nível da interpretação do pedido ou à disponibilidade do agente, notificando o agente solicitador acerca do motivo. Deste modo, o agente genérico realiza um pedido, através de uma mensagem de acto performativo *“Request”*, em que verifica se o agente central possui dados de previsão. Em caso afirmativo, o agente central deverá enviar uma mensagem de confirmação, com acto performativo *“Agree”*, enviando imediatamente uma mensagem com os dados de previsão, tendo esta um acto performativo do tipo *“Inform”*. Em caso de indisponibilidade do agente central — situação resultante de uma falha de comunicação entre o agente e o controlador PD, por exemplo — o mesmo deverá enviar uma mensagem do tipo *“Refuse”* ao agente solicitador. Caso seja endereçada ao agente central uma mensagem fora de especificação, o mesmo deverá remeter uma mensagem com um acto performativo do tipo *“Not understood”*, indicando que a mensagem não foi entendida pelo agente. Tal como preconizado no

paradigma de sistemas multi-agente, a existência de um comportamento social de “*feedback*” permite que os agentes adotem estratégias adequadas a cada resposta.

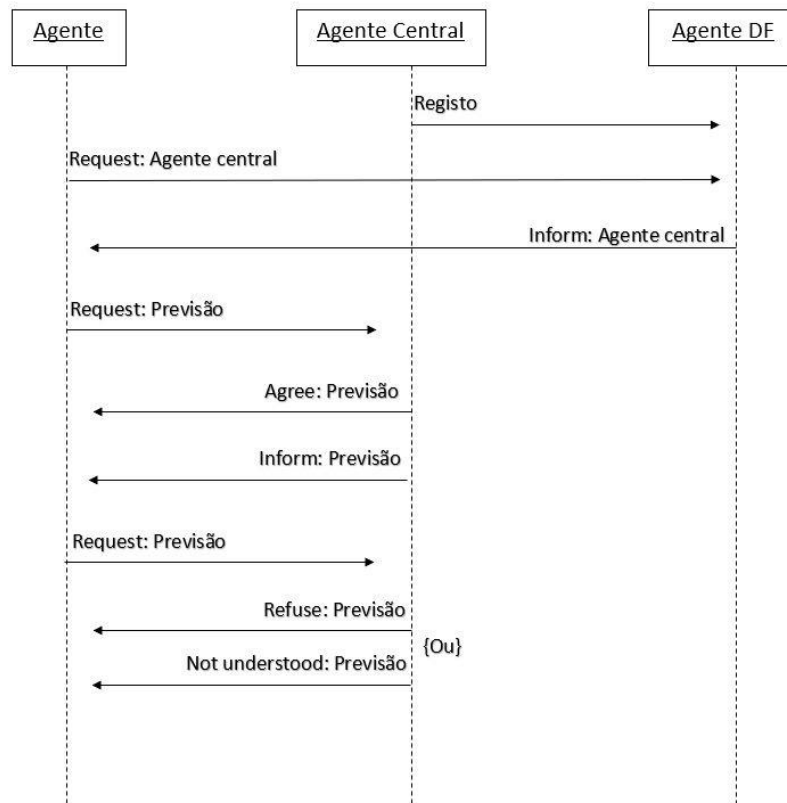


Figura 55 - Diagrama de comunicação do sistema multi-agente

Tal como descrito anteriormente, a plataforma JADE é constituída por uma biblioteca de métodos e classes baseados num paradigma de programação orientada a objectos (POO). Assim sendo, a realização do sistema multi-agente deverá seguir, igualmente, o mesmo paradigma, traduzindo o problema real em classes suportadas, se necessário, pela plataforma JADE. O diagrama de classes em UML do sistema encontra-se representado no anexo II. No desenvolvimento do sistema multi-agente optou-se por criar quatro camadas funcionais reunindo diferentes classes:

- Camada de Interface gráfica
- Camada multi-agente
- Camada de dados partilhados
- Camada de rede

A camada de interface gráfica é responsável pela comunicação com o operador, permitindo-lhe observar e modificar o estado do agente. Esta camada é constituída pelas classes: “janela” e “janelaCentral”. A classe “janela” designa uma interface gráfica de um agente genérico, permitindo apenas a observação do seu estado e a gestão da ferramenta de rede. A classe “janelaCentral” realiza uma interface gráfica dedicada ao agente central, permitindo a modificação do seu estado e o controlo da ferramenta de rede. A ferramenta de rede pertence à camada de rede. As classes em análise herdam as características da classe *JFrame*, a qual é suportada pela biblioteca Java *Swing*. A referência ao termo “herança” — intrínseca ao paradigma POO — ilustra um mecanismo que permite que uma classe adopte os mesmos atributos e métodos de outra classe, ainda que sejam classes distintas.

A camada multi-agente reúne as classes cujas características são herdadas da classe “*Agent*” da biblioteca JADE. Esta camada é responsável pelo comportamento puro do sistema multi-agente, nela se definindo a interacção entre os agentes e a lógica particular ao sistema. A camada é constituída pelas classes: “agente” e “central”. Tal como o nome indica, a classe “agente” designa um agente genérico com um padrão de conversação claramente definido: interrogar o agente central por dados de previsão actualizados. A natureza cíclica dos eventos realizados pelo agente, seja ao nível da procura de um agente central, seja ao nível do pedido de novos dados de previsão, é realizada através da classe “*TickerBehaviour*” da biblioteca JADE, sendo apenas necessário definir a periodicidade dos eventos. O padrão de conversação foi implementado através da classe “iniciador”, a qual herda características da classe estática “*AchieveREInitiator*” da biblioteca JADE. A classe “central” designa um agente que visa atender os pedidos de previsão dos vários agentes genéricos. Este agente realiza também o seu registo no agente de páginas amarelas do sistema. O padrão de conversação da classe é implementado através da classe “*AchieveREResponder*” da biblioteca JADE.

A camada de dados partilhados liga as camadas de interface gráfico, multi-agente e de rede. Os dados de previsão são acedidos e alterados através da interface gráfica e da camada multi-agente, sendo cada classe realizada por uma “*thread*” distinta. A existência de múltiplas “*threads*” com acesso a um único recurso apresenta um risco de interferência no registo dos dados entre os vários intervenientes. São aplicados mecanismos de sincronização que permitam controlar o acesso aos dados partilhados, mantendo a coerência dos dados. A camada é constituída pela classe “*forecastData*”, a qual é utilizada no registo dos dados de previsão de onda.

A camada de rede realiza a ligação entre o sistema multi-agente e um sistema exterior. No caso em estudo, pretende-se que o sistema multi-agente obtenha dados de um sistema de aprendizagem automática — dados só acessíveis ao agente central — e que cada agente genérico tenha como acção o ajuste de um controlador PD. Ambos os sistemas são exteriores à plataforma multi-agente, sendo, portanto, necessário realizar uma interface entre os dois ambientes. A camada de rede realiza a interface entre o sistema multi-agente e o ambiente exterior — no caso em particular, constituído por um ambiente de simulação MATLAB/*Simulink*® — através de um conjunto de métodos de comunicação segundo o protocolo TCP-IP.

A camada é constituída pelas classes: “daemon” e “cliente”. A definição das duas classes será explorada em maior detalhe na secção de comunicação.

4.3.3 Comunicação

O sistema multi-agente deve manter uma via de comunicação com um sistema exterior, permitindo-lhe a realização de acções e a recolha de informações. No caso em estudo, o sistema exterior representa o parque de ondas constituído por vários dispositivos ISWEC: um dispositivo classificador e dispositivos ISWEC genéricos. Assim, o sistema multi-agente deverá obter os dados de previsão e definir os parâmetros dos vários controladores. A solução encontrada compreende a criação de uma camada de rede que permite que o controlador comunique com o seu agente através do protocolo de rede TCP-IP. O agente central partilha, uma interface de comunicação de rede com o dispositivo classificador. Neste ambiente admite-se que todos os dispositivos pertencentes ao parque de ondas se encontram ligados em rede (LAN) através do protocolo TCP-IP. A plataforma JADE será suportada neste ambiente distribuído de acordo com estas premissas. A Figura 56 ilustra os componentes envolvidos na comunicação entre o sistema exterior e a plataforma JADE, assumindo um exemplo minimalista de um parque de ondas com apenas dois dispositivos ISWEC.

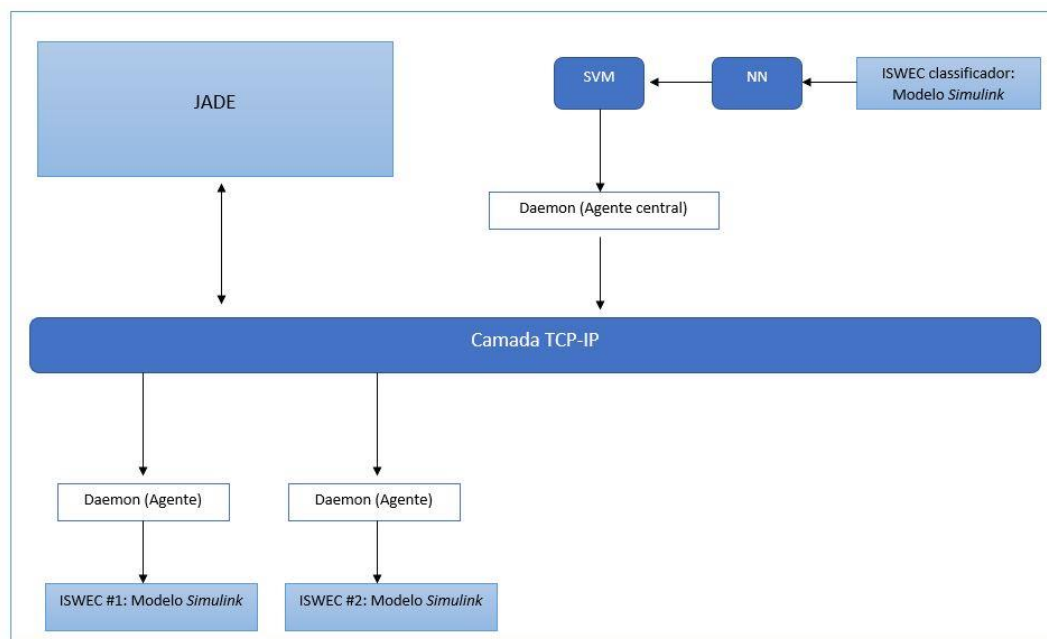


Figura 56 - Diagrama de comunicação em rede

O componente “daemon” é um processo executado em paralelo com o agente, que realiza uma interface entre o agente e a camada de comunicação comum ao sistema. O anexo II apresenta as interfaces gráficas do sistema multi-agente proposto. Uma vez iniciado o “daemon” de um agente genérico, este processo será responsável pelo envio ao controlador PD da configuração actual, a qual fora enviada pelo agente central. O “daemon” do agente central tem como função introduzir os dados de previsão provenientes do sistema de aprendizagem automática —sistema constituído por uma rede neuronal dinâmica (NN) e um classificador SVM — no sistema multi-agente da plataforma JADE. Considerando que o parque de ondas partilha uma rede de comunicação, a cada “daemon” corresponderá o endereço (IP) do próprio dispositivo, já que a comunicação se faz ao nível daquele equipamento, não se alargando aos restantes dispositivos ligados em rede.

As classes “daemon” e “client”, pertencentes à camada de rede, são realizadas por processos sob a forma de “*Threads*” cuja execução se dá em segundo plano. A interface gráfica do agente notifica o utilizador acerca da operação do componente de rede. As duas classes utilizam os objectos: *ServerSocket* e *Socket*. A interface gráfica permite que se inicie o componente de rede adoptando uma configuração padrão: endereço IP=*localhost*; Porta=8585. É possível simular o sistema multi-agente num único computador, bastando que cada agente possua um componente de rede com uma porta distinta.

4.3.4 Sincronização de processos

A Figura 57 apresenta um diagrama representativo da troca de mensagens entre os vários agentes, obtido através da ferramenta do JADE: *Sniffer Agent*. Neste sistema, os agentes genéricos são identificados pelas letras “A” e “B”, sendo o agente central identificado pela letra “C”. O agente de páginas amarelas possui a denominação padrão “DF”. De acordo com a figura, os agentes agora introduzidos pertencem ao “*Main-Container*”, uma abstracção do espaço do sistema multi-agente. Na figura, é possível observar que os agentes genéricos trocam mensagens regulares com o agente “DF”, obtendo, assim, o endereço do agente central. A troca de mensagens entre os agentes genéricos e central desenvolve-se de acordo com o comportamento definido na Figura 55. Na simulação do sistema, foi introduzida a possibilidade de falha do agente central. Este evento pode ser observado no envio de mensagens com acto performativo “*Refuse*” aos agentes “A” e “B”. A ocorrência de falha do agente central é devidamente gerida pelo agente genérico, notificando o operador através da interface gráfica (ver registo de falhas Figura 57), definindo a configuração corrente com o valor de zero.

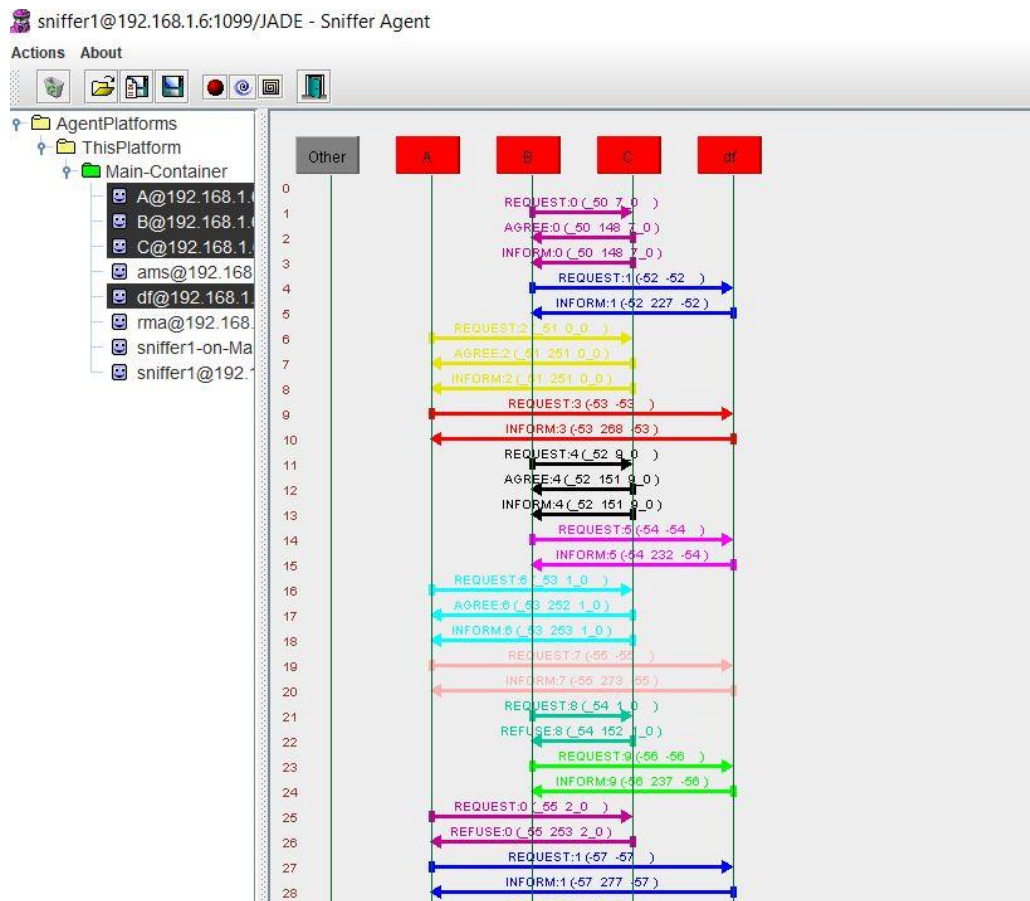


Figura 57 - Diagrama de mensagens do sistema multi-agente

4.3.5 Simulação dinâmica

Na simulação dinâmica do sistema pretende-se avaliar a interação entre um dispositivo ISWEC genérico e um dispositivo classificador. O sistema é constituído pelos seguintes componentes: a) Sistema multi-agente baseado na plataforma JADE; b) Modelo de um dispositivo genérico ISWEC; c) Modelo do dispositivo classificador.

A Figura 58 apresenta o modelo global do dispositivo classificador realizado em *Simulink*. O modelo é constituído pelo sistema dinâmico ISWEC, uma rede neuronal do tipo NARX e um sistema de classificação SVM. O sistema dinâmico ISWEC tem como entrada um sinal correspondente ao binário de um conjunto de ondas marítimas, produzindo, à sua saída, um conjunto de sinais resultante da medição de sensores internos. O sistema de rede neuronal realiza um modelo inverso do sistema ISWEC, extrapolando o sinal de excitação a partir dos dados produzidos pelos sensores. O classificador, alimentado pelo modelo inverso, requer uma análise de um conjunto de dados com uma duração de 50 segundos (1000 amostras),

tendo-se implementado um sistema de “*buffer*” de tipo FIFO (“*first-in first-out*”) — um tipo de memória em que os dados são lidos pela mesma ordem em que foram escritos, tendo larga utilização em sistemas de comunicação, permitindo que o emissor e receptor escrevam e leiam os dados com ritmos distintos [53, p. 342].

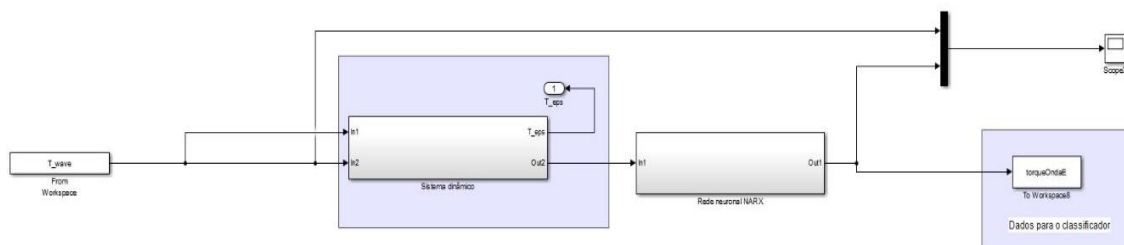


Figura 58 - Modelo do dispositivo classificador

Na excitação do sistema foi utilizado um sinal de teste — o binário induzido no dispositivo. Uma vez criado o sinal de excitação, simulou-se o funcionamento do sistema, tendo-se definido um período de 5 minutos entre eventos de ajuste dos ganhos do controlador PD. A cada evento de ajuste, o dispositivo classificador irá analisar os dados dos últimos 50 segundos relativos ao binário induzido no ISWEC, comunicando ao respectivo agente na plataforma JADE qual a classe extrapolada. O dispositivo genérico ISWEC deverá, segundo igual periodicidade, verificar junto do respectivo agente na plataforma JADE qual a classe extrapolada para o actual estado do mar, definindo os ganhos de acordo com os dados recebidos.

Tabela 14 - Desempenho do controlador sob configurações estática e dinâmica

Configuração	Potência média absorvida (W)	Desempenho relativo (referência*)
Onda 1	47938	61.04%
Onda 2	62843	80.02%
Onda 3	74643	95.05%
Onda 4	74782	95.22%
Onda 5	71366	90.88%
Onda 6	77335	98.48%
Onda 7	77597	98.81%
Onda 8	76538	97.46%
Dinâmica*	78532	100.00%

A simulação dos dois dispositivos foi realizada utilizando duas instâncias MATLAB e um sistema multi-agente no mesmo sistema operativo, garantindo-se a comunicação entre as instâncias e a plataforma JADE através de *sockets* definidos em portas distintas. A avaliação do desempenho é realizada através da métrica de potência média absorvida pelo dispositivo ISWEC.

A Tabela 14 sumariza o desempenho do controlador em distintas configurações. O sinal de teste utilizado é constituído por uma aglutinação de ondas de tipo 4 e 7, pertencendo 60% das observações ao primeiro tipo (vide Figura 59). O sinal tem uma duração de 60 minutos, não existindo ondas distintas de duração inferior a 5 minutos. A construção deste sinal tem como objectivo introduzir alguma incerteza na previsão horária, incluindo um tipo de onda com uma ocorrência mais fugaz, avaliando se o sistema é capaz de detectar a transição entre ondas e, assim, obter um melhor desempenho na extracção de energia.

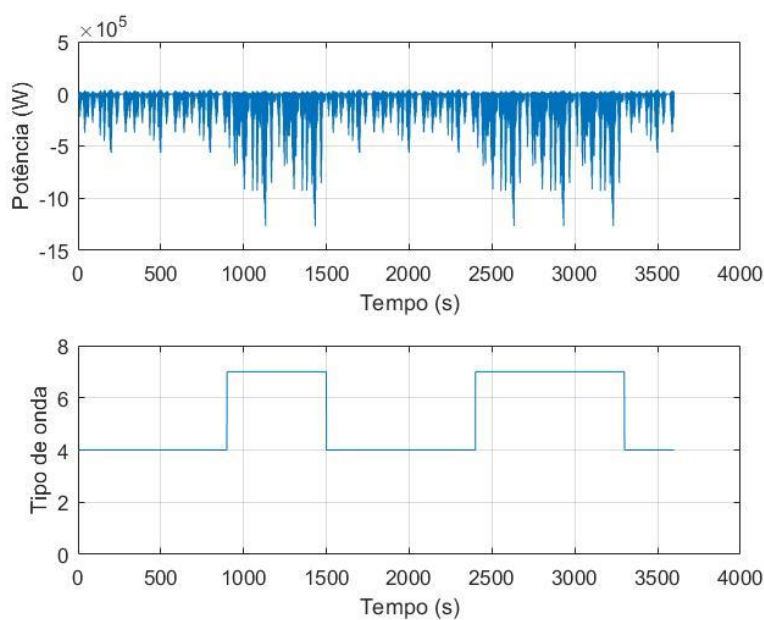


Figura 59 - Potência absorvida durante simulação de ajuste dinâmico

Conclui-se que no cenário criado, a configuração dinâmica de ganhos do sistema resulta numa potência média absorvida superior. Contudo, é importante notar que, admitindo que este sinal seria classificado, de forma probabilística, como de tipo 4, o desempenho marginal do controlador dinâmico face ao modelo de previsão convencional teria uma vantagem de aproximadamente 5%. A insuficiência de dados reais não permite uma avaliação mais profunda do sistema dinâmico de detecção, restando a realização de cenários artificiais cuja aplicabilidade se desconhece. De igual forma, o período entre eventos de ajuste, que depende do sistema classificador (buffer mínimo), é um parâmetro que requer um conhecimento probabilístico sobre o estado do mar e as suas transições.

5 Conclusão

O presente trabalho teve como objectivo avaliar um sistema multi-agente, aplicado a um parque de conversores de energia de ondas ISWEC, cuja previsão de estado mar por processos de aprendizagem automática induz um ajuste de ganhos do controlador PD. Com a aplicação desta estratégia pretende-se melhorar o desempenho do dispositivo ISWEC na produção de electricidade.

A fase de processamento através de métodos de aprendizagem automática iniciou-se com o treino de uma rede neuronal dinâmica recorrente, de tipo NARX, para um conjunto alargado de ondas marítimas, incluindo 8 tipos de onda esperados. A rede neuronal dinâmica, utilizada no modelo inverso do sistema, apresentou um desempenho de elevada qualidade, obtendo-se extrapolações do binário induzido no ISWEC com uma VAF globalmente superior a 90%. O sistema classificador tem como objectivo identificar o tipo de onda a partir dos dados processados pela rede neuronal dinâmica. Nesta fase foi utilizado um classificador SVM, tendo-se concluído que, para um conjunto de observações de 50 segundos de duração e um reduzido conjunto de variáveis explicativas, é possível obter um desempenho de classificação globalmente superior a 75%.

No sistema multi-agente, suportado na plataforma de *software* Jade, foi possível realizar uma estratégia de comunicação entre agentes ISWEC autónomos e o agente central de classificação. Esta estratégia inclui protocolos de conversação, prevendo situações de recusa e indisponibilidade técnica, sugerindo aos interlocutores uma reacção alternativa. A comunicação entre o sistema multi-agente e os modelos de simulação, realizados em MATLAB, foi assegurada por *sockets*, obtendo-se, em termos médios, latências de reduzida expressão (cerca de 2 segundos) face à periodicidade dos ajustes automáticos (5 minutos).

A conjunção do sistema multi-agente com os métodos de aprendizagem automática revelou-se, ainda que em cenários artificiais, como uma alternativa viável ao sistema tradicional de previsão, obtendo-se um incremento na potência média absorvida pelo ISWEC. Será importante notar que esta estratégia de ajuste automático de ganhos é sensível à distribuição de ondas marítimas não consideradas pelo sistema de previsão tradicional. Dado que esta informação é desconhecida, é expectável que o incremento do desempenho do sistema de controlo multi-agente seja mais evidente quando a previsão tradicional, realizada de forma horária, não considera a presença de ondas de grande energia e de duração reduzida, utilizando ganhos ajustados para ondas de menor intensidade. No exemplo artificial, o sistema de ajuste dinâmico apresentou uma vantagem na potência extraída de 5% face ao modelo tradicional.

As principais limitações neste trabalho resumem-se à escassez de dados de aprendizagem — registo temporal de ondas marítimas. Relativamente ao trabalho futuro, seria interessante avaliar a inclusão de variáveis explicativas baseadas na transformada *Wavelet*, melhorando o desempenho do classificador.

6 Referências

- [1] S. Huntington, “The clash of civilizations and the remaking of world order,” 1996, p. 68.
- [2] Office of the Historian, “Oil Embargo, 1973–1974,” Department of state - United States of America, [Online]. Available: <https://history.state.gov/milestones/1969-1976/oil-embargo>. [Acedido em 27 Setembro 2017].
- [3] BP, “BP Statistical Review of World Energy - June 2017,” BP, 2017.
- [4] J. Ramage, Energy - A guidebook, Oxford University Press, 1997.
- [5] EIA, “International Energy Outlook 2017,” U.S. Energy Information Administration, 2017.
- [6] T. Thorpe, “A Brief Review of Wave Energy,” 1999.
- [7] A. F. O. Falcão, “Modelling of Wave Energy Conversion,” Instituto Superior Técnico, Universidade Técnica de Lisboa, 2014.
- [8] ScienceDirect, “ScienceDirect,” [Online]. Available: <http://www.sciencedirect.com/search?q=wec%20energy>. [Acedido em 15 Setembro 2017].
- [9] A. F. O. Falcão, “Energia das ondas,” em *Seminário sobre a física e a energia*, Lisboa, 2005.
- [10] WavEC - Offshore Renewables, “Comunicado de Imprensa sobre a Central Ondas do Pico pelo WavEC,” [Online]. Available: http://www.wavec.org/content/files/Comunicado_de_Imprensa_Central_do_Pico_19_abril_2018.pdf. [Acedido em 25 2018].
- [11] J. Blackledge, E. Coyle, D. Kearney, R. McGuirk e B. Norton, “Estimation of Wave Energy from Wind Velocity,” *Engineering Letters*, Novembro 2013.
- [12] A. F. O. Falcão, “Wave energy utilization: A review of the technologies,” 2009.
- [13] Wave Dragon ApS, “Principles,” [Online]. Available: <http://www.wavedragon.net>. [Acedido em 29 Setembro 2017].
- [14] S. Haykin, Neural networks and learning machines, Pearson, 2009.
- [15] J. Jang, C. Sun e E. Mizutani, Neuro-Fuzzy & Soft Computing, Pearson, 2015.
- [16] D. Rumelhart, G. Hinton e R. Williams, “Learning representations by back-propagating errors,” 1986.
- [17] S. Guzman, J. Paz e M. Tagert, “The use of NARX neural networks to forecast daily groundwater levels,” Springer, 2017.

- [18] A. Di Piazza, M. Di Piazza e G. Vitale, "Solar and wind forecasting by NARX neural networks," *EDP Sciences*, 2016.
- [19] C. Cortes e V. Vapnik, "Support-vector networks," *Machine Learning*, nº 20, pp. 273-297, 1995.
- [20] T. Hastie, R. Tibshirani e J. Friedman, *The Elements of Statistical Learning*, Springer.
- [21] A. Ben-Hur, C. Ong, S. Sonnenburg, B. Scholkopf e G. Ratsch, "Support Vector Machines and Kernels for Computational Biology," *PLoS - Computational biology*, vol. 4, nº 10, 2008.
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [23] C.-C. Chang e C.-J. Lin, "LIBSVM: A library for support vector machines," National Taiwan University, Taiwan, 2001.
- [24] C.-C. Chang e C.-J. Lin, "LIBSVM -- A Library for Support Vector Machines\Interfaces and extensions to LIBSVM," [Online]. Available: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>. [Acedido em 10 2 2018].
- [25] P. e. al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [26] E. Frank, M. Hall e I. Witten, "The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"," Morgan Kaufmann, 2016.
- [27] Mathworks, *Statistics and Machine Learning Toolbox*, R2016, 2016.
- [28] M. Wooldridge, "Intelligent Agents," em *An Introduction to MultiAgent Systems*, John Wiley & Sons, Ltd, 2002, p. 15.
- [29] R. Thaler, "Toward a positive theory of consumer choice," *Journal of economic behaviour and organization*, vol. 1, pp. 39-60, 1980.
- [30] F. Bellifemine, G. Caire e D. Greenwood, *developing multi-agent systems with JADE*, John Wiley & Sons, Ltd, 2007.
- [31] FIPA, "Welcome to FIPA!," [Online]. Available: www.fipa.org. [Acedido em 1 June 2017].
- [32] FIPA, "FIPA ACL Message Structure Specification - SC00061G," 2002.
- [33] A. Rao e M. Georgeff, "BDI Agents: From theory to practice," em *Proceedings of the First international conference on Multiagent systems*, 1995.
- [34] U. Wilensky, "NetLogo," Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL., [Online]. Available: <http://ccl.northwestern.edu/netlogo/>. [Acedido em 14 1 18].
- [35] K. Kravari e N. Bassiliades, "A survey of agent platforms," *Journal of artificial societies and social simulation*, vol. 18, nº 11, 2015.

- [36] J. Hubner e R. Bordini, "Jason - A Java-based interpreter for and extended version of AgentSpeak," [Online]. Available: <http://jason.sourceforge.net/wp/>. [Acedido em 5 2 18].
- [37] A. Pokahr, L. Braubach e W. Lamersdorf, "Jadex: A BDI reasoning engine," em *Jadex: A BDI reasoning engine*, Springer, 2005, pp. 149-174.
- [38] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan e G. Balan, "MASON: A Multi-Agent Simulation Environment," George Mason University, VA, 2005.
- [39] D-CIS LAB, "AgentScape - Distributed Agent Middleware\AgentScape 2.0 Documentation," [Online]. Available: <http://www.agentscape.org/>. [Acedido em 27 03 2018].
- [40] J. Dongarra, K. Madsen e J. Wasniewski, "Applied Parallel Computing: State of the Art," em *Applied Parallel Computing: State of the Art*, Springer, 2006, p. 678.
- [41] AOS - Autonomous Decision-making software, "JACK," AOS, 15 2 2018. [Online]. Available: <http://www.agent-software.com.au/products/jack/>.
- [42] Microsoft, "What is middleware?," [Online]. Available: <https://azure.microsoft.com/en-in/overview/what-is-middleware/>. [Acedido em 16 2 2018].
- [43] G. Bracco, "ISWEC: a gyroscopic wave energy converter," Politecnico di Torino, 2010.
- [44] G. Vissio, D. Valério, G. Mattiazzo, G. Bracco, P. Beirão e N. Pozzi, "ISWEC - Linear Quadratic Regulator oscillating control," 2016.
- [45] R. Laranjeira, "Control of the wave energy converter ISWEC in simulation," IST, 2016.
- [46] J. F. Wilson, Dynamics of offshore structures, John Wiley & Sons, Inc., 2003, p. 155.
- [47] C. L. Bretschneider, "Generation of waves by wind - State of the art," em *International summer course*, Lunteren, 1964.
- [48] G. E. P. G. M. J. Box, Time Series Analysis: Forecasting and Control, Prentice Hall, 1994.
- [49] P. E. K. Kevin L. Priddy, Artificial Neural Networks: An Introduction, SPIE Press, 2005.
- [50] P. Domingos, "A Few Useful Things to Know about Machine Learning," Department of Computer Science and Engineering, University of Washington, Seattle.
- [51] TidalEnergy Today, "Wave for Energy: ISWEC device ready for production," TidalEnergy Today, [Online]. Available: <https://tidalenergytoday.com/2015/08/10/wave-for-energy-iswec-device-ready-for-production/>. [Acedido em 20 3 2018].
- [52] Machine Learning at Berkeley (University of California), "Machine Learning at Berkeley," 2018. [Online]. Available: <https://ml.berkeley.edu/home>. [Acedido em 4 3 2018].

- [53] A. Oliveira, J. Monteiro e G. Arroz, *Arquitectura de computadores: dos sistemas digitais aos microprocessadores*, IST Press, 2009.

Anexo I

Este anexo inclui informações adicionais sobre o dispositivo ISWEC.

Figura 60 - Parâmetros do sistema ISWEC [45]

Símbolo	Parâmetro	Valor
$\dot{\varphi}$	Velocidade angular do volante de inércia	40 rad/s
K	Rigidez hidrostática	1.9×10^7 N.m/rad
I_{eq}	Momento de inércia equivalente do flutuador	1.2×10^7 kg.m ²
J	Momento polar de inércia do volante	15000 kg.m ²
I_g	Momento de inércia do volante segundo o eixo ε	45000 kg.m ²

Figura 61 - Conjunto de ganhos óptimos para cada tipo de onda [45]

Classe de onda	$k / 10^4$ N.m	$c / 10^4$ N.m.s	Potência / k.W
1	11.17	3.83	5.99
2	10.49	6.42	5.74
3	6.61	7.14	47.80
4	6.91	7.51	47.90
5	6.49	5.95	52.93
6	5.97	10.75	115.25
7	5.82	9.84	121.66
8	5.30	11.24	150.21

Anexo II

Este anexo inclui informação adicional sobre o sistema multi-agente desenvolvido em Java, nomeadamente o diagrama de classes e interfaces gráficas.

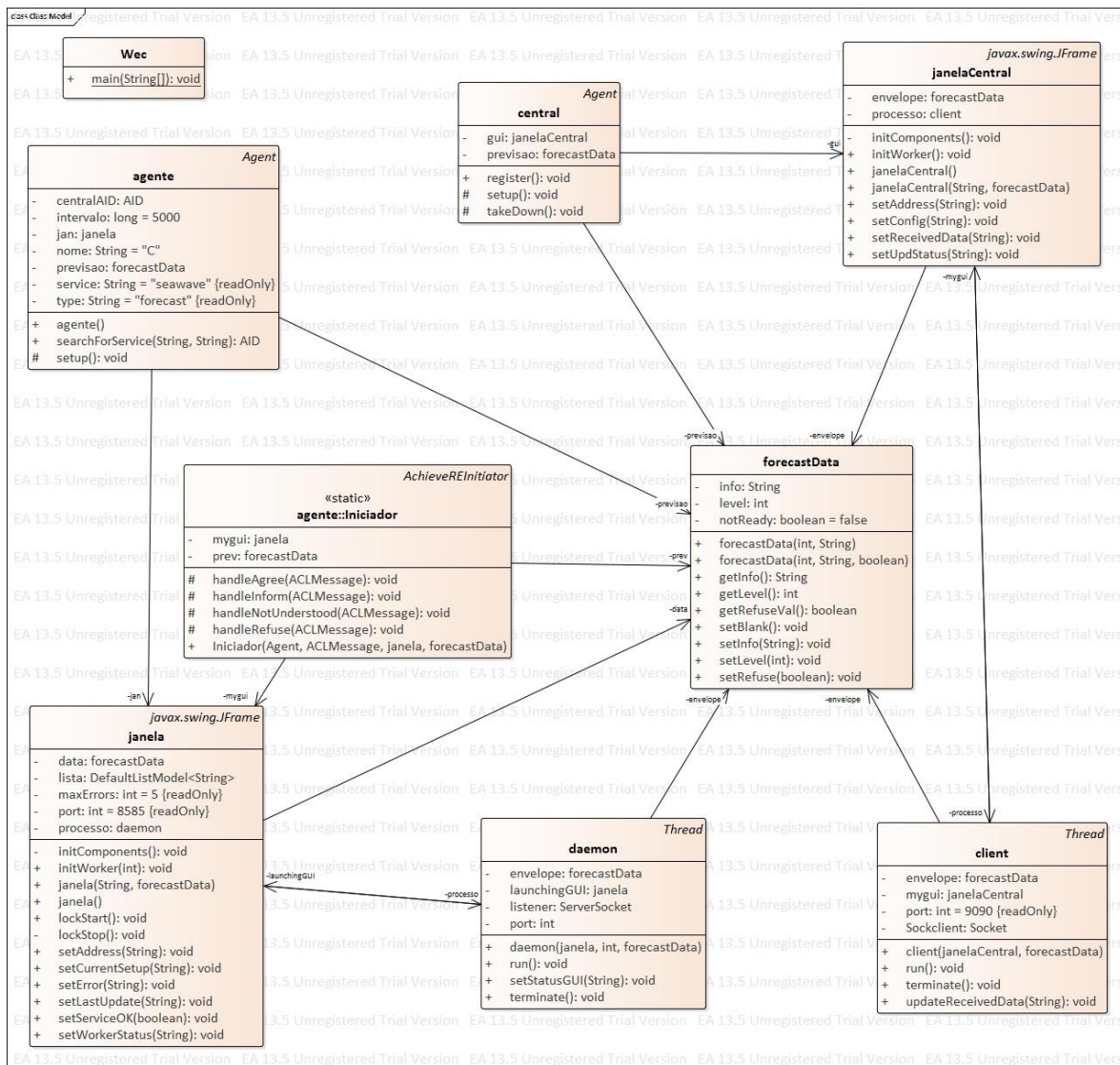


Figura 62 - Diagrama UML do sistema multi-agente

Anexo II

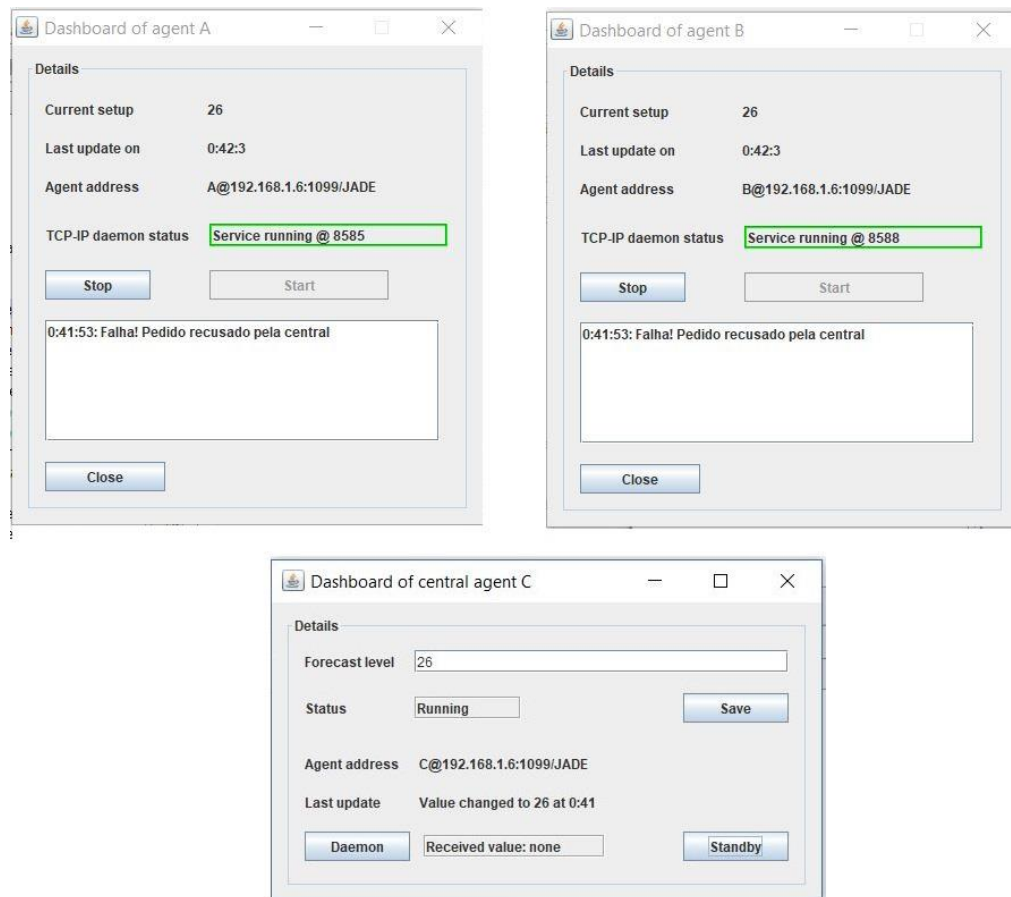


Figura 63 - Interfaces gráficas do sistema multi-agente

Anexo III

Este anexo inclui respostas da rede NARX.

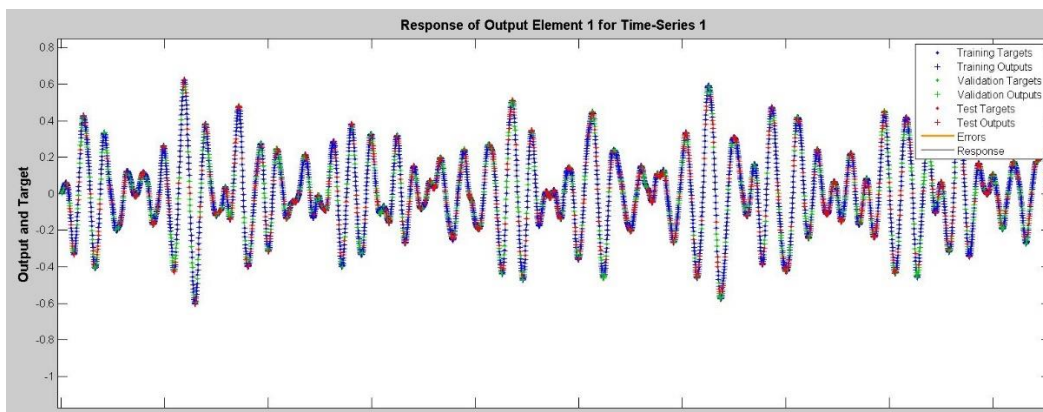


Figura 64 - Resposta a ondas 1 e 2

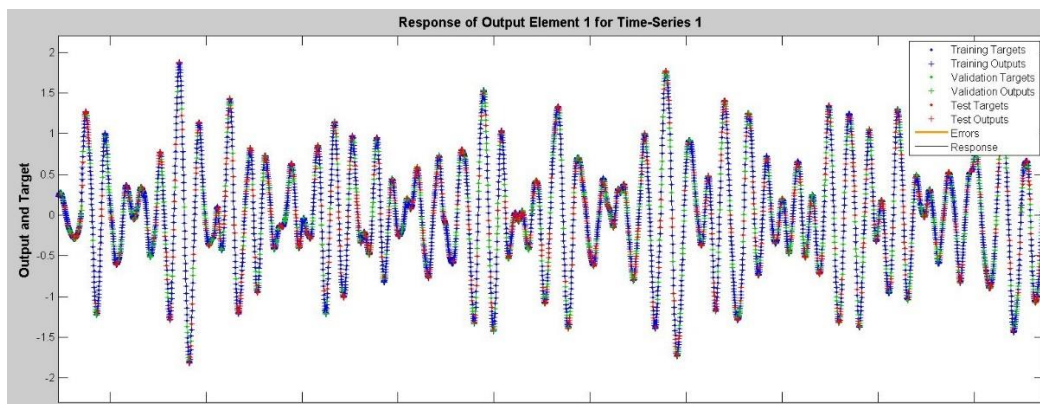


Figura 65 - Resposta a ondas 3 e 4

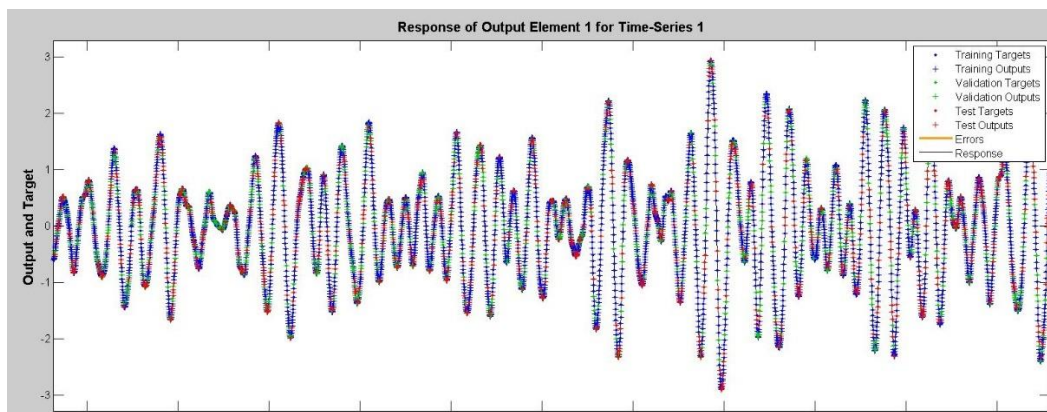


Figura 66 - Resposta a ondas 5 e 6

Anexo III

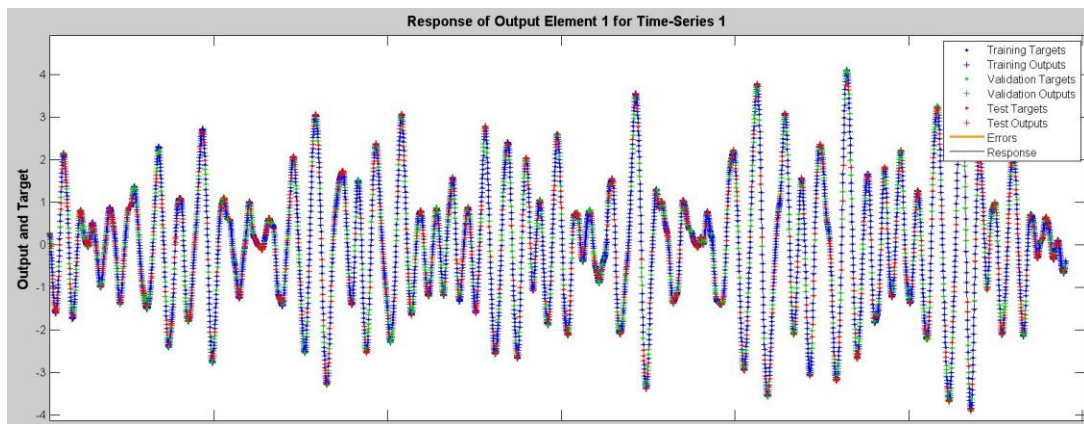


Figura 67 - Resposta a ondas 7 e 8