



Thiago José de Almeida Pereira

Licenciado em Ciências da Engenharia Eletrotécnica
e de Computadores

eVentos 3 – Desenvolvimento de controlador difuso para navegação autónoma de veleiro

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

Orientador: Professor Doutor Luís Filipe dos Santos Gomes,
Professor Associado, Universidade Nova de Lisboa

Júri:

Presidente: Doutora Anikó Katalin Horváth da Costa

Vogais: Doutor Luís Filipe dos Santos Gomes
Doutor Luís Filipe Figueira de Brito
Palma

eVentos3 - Desenvolvimento de controlador difuso para navegação autónoma de veleiro

Copyright © Thiago José de Almeida Pereira, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

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

Dedicado a todos que contribuíram para a pessoa que sou hoje...

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu colega e amigo Miguel Santos por ter participado e acompanhado diariamente na dissertação como em todo o curso, também um grande obrigado ao Professor Doutor Luís Gomes pela sua disponibilidade, orientação e conselhos ao longo de toda a dissertação.

Em especial queria agradecer ainda à Professora Doutora Anikó Costa, ao Sr. Duarte Guerreiro e a todas as pessoas da instituição da FCT que contribuíram para este projeto.

Os trabalhos associados a esta dissertação beneficiaram de resultados obtidos no projeto Petri-Rig - Ambiente de desenvolvimento de sistemas embutidos baseado em redes de Petri, financiado pela FCT com a referência PTDC/EEI-AUT/2641/2012.

Também quero agradecer à minha família por estar sempre presente nos meus passos, por acreditarem sempre em mim, e terem sempre a máxima disponibilidade para tudo.

Um obrigado à Isabel Martins pela paciência de ler todo este documento e corrigir.

Por fim queria agradecer à Rita Abrantes por tudo o que ela faz por mim e representa na minha vida.

Resumo

Controlar a navegação de uma embarcação, neste caso de um veleiro, pode ser fácil quando controlada por um humano. Mas, quando falamos em embarcação autónoma, existem algumas dificuldades relativas à programação da rota a que a embarcação deve seguir.

Nesta dissertação foram feitas diversas análises referentes às técnicas de velejar, bem como a caracterização de um controlador utilizando lógica difusa, com a colaboração de sensores e atuadores. Para este estudo foi necessário equipar um veleiro com diversos sensores, atuadores e controladores.

Por fim, foram realizados testes e análises que possibilitaram o aperfeiçoamento das técnicas de controlo.

O objetivo é transformar um veleiro telecomandado num capaz de navegar autonomamente, por forma a ser introduzido posteriormente num de maior escala.

Palavras-chave: Veleiro, Arduino, Controlador, Lógica difusa, Bussola, GPS, Cata-vento, xFuzzy, Wi-Fi, Radio Controlo, Navegação Autónoma.

Abstract

Control the navigation mode of a vessel, in this case of a sailboat, can be easy when controlled by a human. But when we talk about autonomous sailboat, there are some difficulties in programming the route that the vessel should follow.

This dissertation analyses possible yacht sailing techniques, as well as the characterization of a fuzzy logic controller, having data acquisition from sensors and actuators. For this study, it was necessary to equip a sailboat with several instruments, performance analysis and monitoring.

Finally, tests and analyses were conducted allowing the improvement of control techniques.

Overall, the goal is to turn a remote controlled sailboat to navigate autonomously, in order to be integrated in a larger scale sailboat.

Keywords: Sailboat, Arduino, Controller, Fuzzy logic, Compass, GPS, Wind Vane, xFuzzy, Wi-Fi, Radio controlled, Autonomous Navigation

Índice

Agradecimentos.....	III
Resumo.....	V
Abstract	VII
Lista de Figuras.....	XIII
Lista de Tabelas	XV
Abreviaturas.....	XVII
1-Introdução	1
1.1 Enquadramento e Motivação	1
1.2 Âmbito e Objetivos	2
1.3 Estrutura da Dissertação.....	3
2-Conceitos de Navegação e Trabalhos Relacionados.....	5
2.1 Navegação à Vela.....	6
2.1.1 Conceitos Básicos de Navegação	6
2.1.2 Zonas e Modos de Navegação	8
2.2 FAST: FEUP “Autonomous Sailboat”	12

2.2.1	Implementação do Hardware.....	12
2.2.2	Implementação do Software	12
2.3	eVentos2 – Autonomous Sailboat Control.....	13
2.3.1	Caracterização do Hardware	13
2.3.2	Caracterização do Software	14
3	Ambientes de Desenvolvimento e Execução	15
3.1	Descrição do Hardware.....	16
3.1.1	Arquitetura do Hardware Embutido	16
3.1.2	Plataforma	17
3.1.3	Sensores e Aquisição de Dados	18
3.2	Controlo de Navegação e Ferramentas	21
3.2.1	Controladores Difusos.....	21
3.2.2	Ferramenta de Geração Xfuzzy	23
4	Controlador Desenvolvido.....	27
4.1	Implementação de Controlo Autónomo e Manual.....	28
4.2	Integração Wi-Fi – Arduino	30
4.3	Integração Rádio Controlo – Arduino.....	32
4.4	Proposta para Estrutura Hierárquica do Controlador.....	34
4.4.1	Estrutura de Três Níveis do Controlador	34
4.4.2	Arquitetura do Controlador de Navegação (nível 1) .	35
4.4.3	Pré Processamento	36
4.4.4	Proposta de Controlador	37
4.5	Modelizaros Conjuntos Difusos	45
5	Resultados Experimentais	51
5.1	Otimização dos conjuntos difusos	52
6	Conclusões e Trabalhos Futuros	61
6.1	Conclusões.....	62
6.2	Trabalhos a Desenvolver no Futuro	63

Referências	67
Anexo.....	71

Lista de Figuras

Figura 2. 1 - Componentes do veleiro	6
Figura 2. 2 - Termo de direções	7
Figura 2. 3 - Posições de navegação com direção do vento [5]	7
Figura 2. 4 - Vento aparente e real [7]	8
Figura 2. 5 – FAST: FEUP “Autonomous Sailboat” [2]	10
Figura 2. 6- Organização do Hardware do veleiro FAST [9].....	11
Figura 2. 7 – Organização do software do veleiro FAST [9].....	12
Figura 2. 8 – Veleiro do eVentos 2 [3].....	13
Figura 2. 9 - Organização do hardware do veleiro eVentos2 [3]	14
Figura 3. 1- Arquitetura do hardware embutido	16
Figura 3. 2 - Arduino mega 2560 [11]	17
Figura 3. 3 – Resumo do Arduino 2560 [11]	17
Figura 3. 4 – Catavento e sensor AS5161 [12].....	18
Figura 3. 5 – Posição do catavento em relação ao veleiro, 0º.....	19
Figura 3. 6 – GPS MT3329	19
Figura 3. 7 – Bússola CMPS10 e orientações	20
Figura 3. 8 – Implementação do controlador difuso	21
Figura 3. 9 – Tipo de funções de pertinência [21].....	22
Figura 3. 10 – Estrutura de um controlador difuso [21].....	23
Figura 3. 11 – Função de pertinência dos antecedentes.....	24
Figura 3. 12 – Função de pertinência dos consequentes.....	24
Figura 3. 13 – Ferramenta de geração automática de código Xfuzzy	25

Figura 4. 1 – Meio de comutação por rádio frequência.....	28
Figura 4. 2 - Integração Wi-fi - Arduino.....	30
Figura 4. 3 - Integração Radio controlo – Arduino	32
Figura 4. 4 - Estrutura hierárquica do controlador.....	34
Figura 4. 5 – Arquitetura do software embutido	36
Figura 4. 6 – Esquema da orientação do destino.....	36
Figura 4. 7 – Esquema do Pré processamento	37
Figura 4. 8 – Entradas e saídas do controlador	38
Figura 4. 9 – Ângulo de orientação do destino.....	39
Figura 4. 10 – Funções de pertinência dado o ângulo de orientação do destino para leme.....	39
Figura 4. 11 - Funções de pertinência dado o ângulo de orientação do vento para a vela	40
Figura 4. 12 – Funções de pertinência de atuação no leme	41
Figura 4. 13 - Funções de pertinência de atuação na vela	42
Figura 4. 14 – Controlador visto no software Xfuzzy.....	43
Figura 4. 15 – Esquema de aquisição e ajuste de parâmetros de controlo.....	45
Figura 4. 16 – Encaminhamento “unicast”	46
Figura 4. 17 – “Socket” UDP	47
Figura 4. 18 – Interface gráfica da aplicação (definir coordenadas)	48
Figura 4. 19 – Interface gráfica da aplicação (leituras de variáveis)	48
Figura 4. 20 – Interface gráfica da aplicação (configurações).....	49
Figura 5. 1 - Função de pertinência do primeiro controlador.....	52
Figura 5. 2 – Efeito de trajetória do controlador inicial.....	53
Figura 5. 3 – Atualização das funções de pertinência	54
Figura 5. 4 – Efeito do ajuste no controlador	54
Figura 5. 5 – Documento de texto com valores lidos em situação real	55
Figura 5. 6 – Resultados e análise de quatro pontos em situação de navegação	56
Figura 5. 7 – Ficheiro de valores de atuação nos servos	57
Figura 6. 1- Função de pertinência da inclinação	63
Figura 6. 2 - Sensor de velocidade do veleiro em relação ao mar[26]	64
Figura 6. 3 – Esquema de extração de água do veleiro [27].....	65
Figura A1. 1 - Esquema em blocos das ligações.....	71
Figura A1. 2 - Esquema elétrico módulo de alimentação [28].....	72
Figura A1. 3 - Indicador bateria fraca [28]	73

Lista de Tabelas

Tabela 4. 1 – Conversão de valores de atuação no leme	29
Tabela 4. 2 - Conversão de valores de atuação na vela	29
Tabela 4. 3 – Legenda das funções de pertença da orientação do veleiro	40
Tabela 4. 4 - Legenda das funções de pertença da orientação do vento	41
Tabela 4. 5 - Legenda das funções de pertença do leme	42
Tabela 4. 6 - Legenda das funções de pertença da vela	43
Tabela 4. 7 – Regras do controlador leme	44
Tabela 4. 8 - Regras do controlador leme.....	44
Tabela 5. 1 - Legenda função de pertença do controlador inicial	52
Tabela 5. 2 – Teste à resposta do controlador.....	60
Tabela A1. 1 - Tabela dos componentes para o módulo de alimentação	72

Abreviaturas

- **FPGA:** Field Programmable Gate Array
- **GPS:** Global Positioning System
- **I2C:** Inter-Integrated Circuit
- **IP:** Internet Protocol
- **NMEA:** National Marine Electronics Association
- **PID:** Proportional-Integral-Derivative
- **PWM:** Pulse-Width Modulation
- **RC:** Radio Comando
- **RX:** Receiver mode
- **TX:** Transmitter mode
- **UART:** Universal asynchronous receiver/transmitter
- **UDP:** User Datagram Protocol
- **USB:** Universal Serial Bus
- **VoIP:** Voice over Internet Protocol
- **Wi-Fi:** Wireless Fidelity



Introdução

1.1 Enquadramento e Motivação

A navegação é um dos meios de transporte mais antigos e ainda muito utilizado para o comércio e lazer. Com os avanços tecnológicos e também com a preocupação dos recursos energéticos, os meios de navegação têm vindo a evoluir de forma a melhorar e a aperfeiçoar estas áreas.

Os avanços tecnológicos permitem que as navegações tenham um maior controlo e monitorização através de controladores, atuadores e sensores. Desta forma podemos tirar maior partido das condições meteorológicas e melhorar o desempenho de navegação, conseguindo melhores consumos energéticos e eficiência entre rotas de navegação.

Têm sido desenvolvidos muitos projetos nesta área, tanto para melhorar os consumos energéticos como para tirar o melhor partido da navegação, portanto é necessário saber distinguir eficiência energética de eficiência de navegação.

Com esta situação, surge a ideia de criar um veleiro autónomo a partir de materiais de custo reduzido, com o objetivo deste ter a capacidade de se deslocar de um ponto para outro ponto sem ser necessária a intervenção do homem.

1.2 Âmbito e Objetivos

O objetivo é controlar a navegação de um veleiro através da decisão tomada por uma plataforma de controlo, utilizando tecnologia de baixo custo.

A utilização de tecnologia de baixo custo para implementar sistemas autónomos passa por analisar os diversos dispositivos necessários para a navegação e tomar decisões com base em estratégias de controlo.

Estas estratégias irão servir para atuar nos servomotores que controlam o leme e a vela de forma a garantir a navegação até à localização pretendida. Para poder atuar nos mesmos são necessários sensores que meçam a posição do veleiro, as condições do vento e a sua orientação.

Para a análise dos sensores é necessário, em primeiro lugar, haver um processamento dos dados e, por fim, chegar a um valor de atuação. Com esta dissertação, pretende-se aperfeiçoar esta inteligência através de testes e da análise comportamental do veleiro.

Finalmente, para analisar o comportamento do veleiro é necessário adquirir os dados online. Para tal, é necessário uma aplicação que comunique com o veleiro através de uma tecnologia sem fios e que nos permita ter uma interface com o mesmo, de forma a monitorizar os dados ou alterar as rotas de destino.

Esta aplicação e outros aspetos essenciais para funcionamento do veleiro, foram desenvolvidos em conjunto com um colega também a preparar a sua dissertação [1], resultando num projeto integrado com mais características, nomeadamente com capacidade de navegação em quaisquer condições de vento, favorável ou desfavorável, situação em que passará a navegar à bolina.

1.3 Estrutura da Dissertação

Esta dissertação está descrita em seis capítulos. O primeiro capítulo descreve uma fase introdutória do trabalho a desenvolver.

O segundo capítulo irá abordar conceitos de navegação de um veleiro e trabalhos desenvolvidos anteriormente com objetivos idênticos a este, por forma a contextualizar as tecnologias e implementações já abordadas.

O terceiro capítulo tem como objetivo dar a conhecer ferramentas necessárias para controlar e analisar diversas formas de navegação.

O quarto capítulo descreve o controlador difuso e mecanismos de ajuste necessários para o aperfeiçoamento do mesmo, através da análise de dados adquiridos em testes reais.

O quinto capítulo aborda todos os resultados obtidos a nível experimental e a integração de meios de comunicação entre o veleiro e o computador e entre o veleiro e o telecomando.

O sexto capítulo contém conclusões sobre todo o trabalho desenvolvido, bem como críticas positivas e construtivas e, para terminar, descrições de trabalhos a desenvolver no futuro.

Por fim consta um bloco de referências e anexos, contendo manuais de utilização da interface criada no computador, ligações do hardware e implementação e aperfeiçoamento dos controladores.



Conceitos de Navegação e Trabalhos Relacionados

A navegação é um método de condução de uma embarcação, quer isto dizer, que existem muitas formas de navegar em diversos modelos de embarcações.

Neste caso, o método abordado é a navegação à vela e o tipo de embarcação é um veleiro. Logo é preciso caracterizar o modelo do veleiro utilizado e os termos técnicos.

É importante mencionar trabalhos relacionados na área de veleiros autónomos, pois estes mostram outras formas de abordar o trabalho que será desenvolvido nesta dissertação.

Assim sendo, será abordado um projeto que já se encontra finalizado intitulado, “FAST: FEUP Autonomous Sailboat” [2], desenvolvido pela Faculdade de Engenharia da Universidade do Porto.

Seguidamente, será feita uma análise a uma dissertação de mestrado desenvolvida na Faculdade de Ciências e Tecnologias da Universidade de Lisboa, “eVentos 2 - Autonomous sailboat control” [3], que retrata o início da investigação sobre o tema inserido nesta dissertação.

2.1 Navegação à Vela

Primeiramente é necessário escolher o tipo de embarcação. Para esta dissertação foi adquirida uma réplica à escala de um veleiro utilizado em competições da “America’s cup racing yacht”, como podemos observar na figura 2.1.

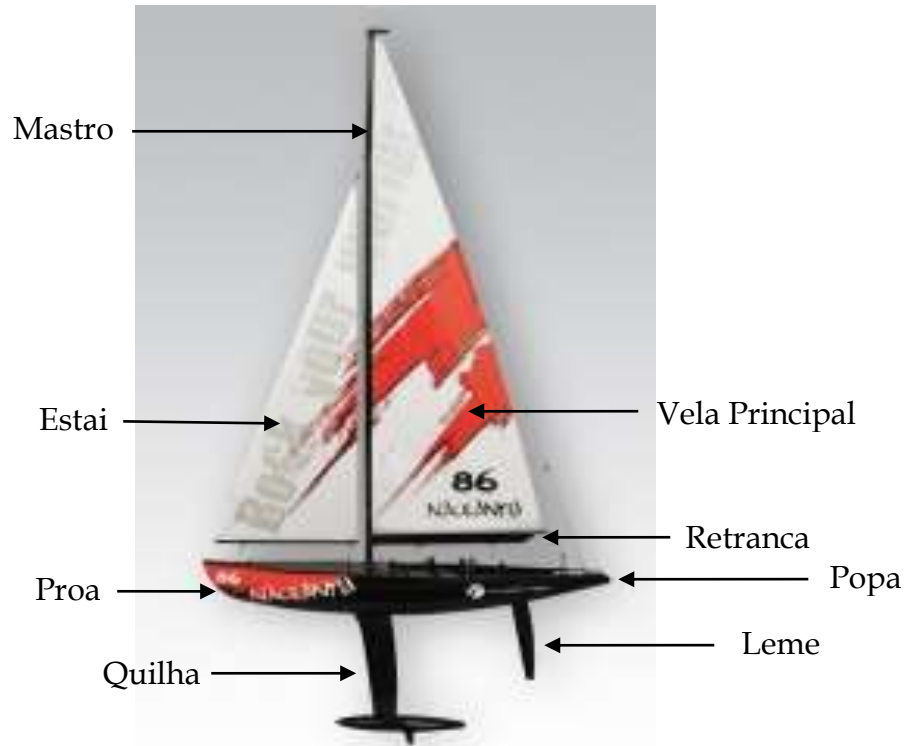


Figura 2. 1 - Componentes do veleiro.

Todas as partes de um veleiro [4] têm um nome e uma função específica e são estudadas pela ciência náutica para serem cada vez mais leves, práticas e funcionais. Existe uma infinidade de termos e de peças diferentes para cada tipo de embarcação. Na figura 2.1 apresentam-se alguns dos termos mais comuns.

2.1.1 Conceitos Básicos de Navegação

Após a análise do veleiro e dos seus principais componentes, é necessário compreender alguns termos técnicos e conceitos de navegação à vela.

Um aspeto fundamental é a utilização de termos corretos: quando uma embarcação vira para o lado direito dizemos que a embarcação vira para estibordo, quando vira à esquerda dizemos bombordo, como ilustra a figura 2.2.



Figura 2. 2 - Termo de direções.

Para haver navegação à vela é necessário existir vento, dado este facto, é preciso compreender a seguinte situação descrita na figura 2.3.

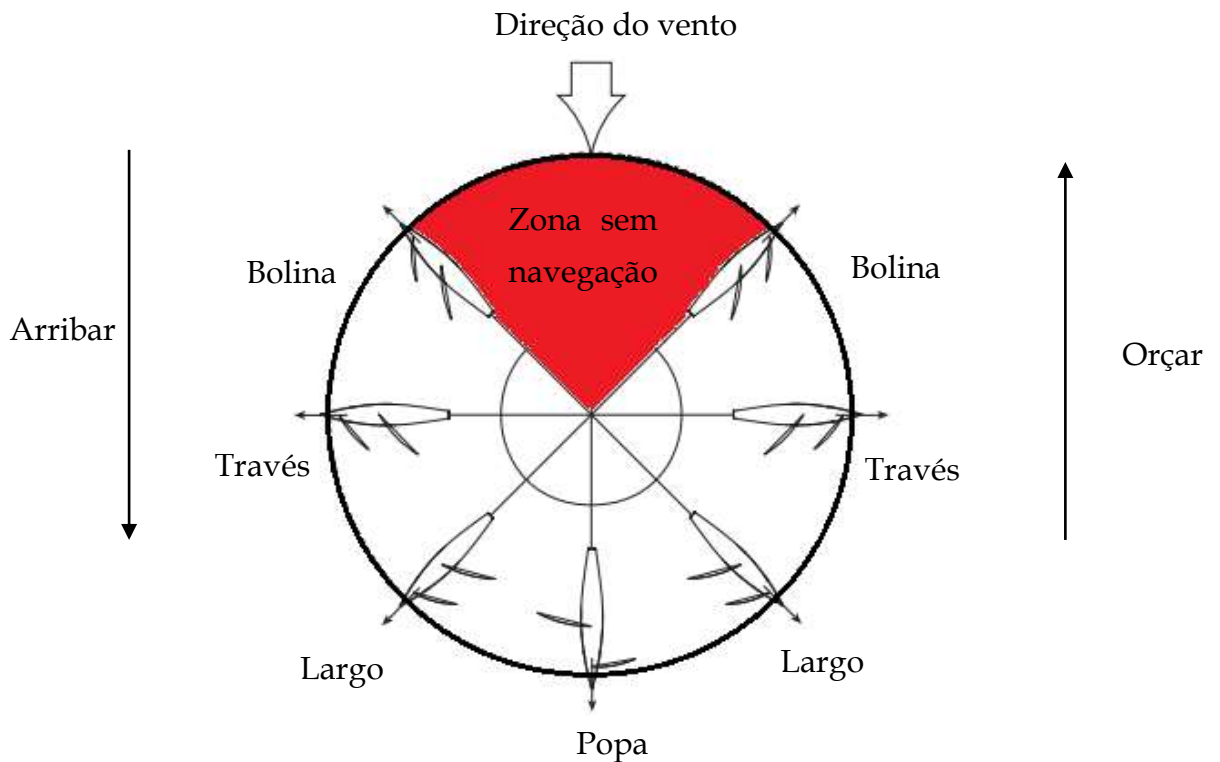


Figura 2. 3 - Posições de navegação com direção do vento [5].

A figura 2.3 indica as diferentes posições em que o veleiro se pode deparar em relação ao vento. No caso do vento estar de popa para proa dizemos que estamos em condições ideais, estamos a velejar a favor do vento. Porém, quando velejamos contra o vento [6], isto é, com vento desfavorável, não é possível navegar. Como a figura 2.3 mostra, existe um certo ângulo em que não é possí-

vel navegar, esse ângulo depende de veleiro para veleiro, ou seja, depende do tamanho das velas e da embarcação.

Nesta dissertação vamos considerar sempre vento favorável (zona branca da figura 2.3), assim diminuimos a complexidade do controlador, tendo sempre em consideração que o vento analisado pelo controlador é o vento aparente, isto é, o vento que é sentido no referencial do veleiro, vento este que é uma soma do vento real com o vento gerado pelo movimento do barco, como mostra na figura 2.4.

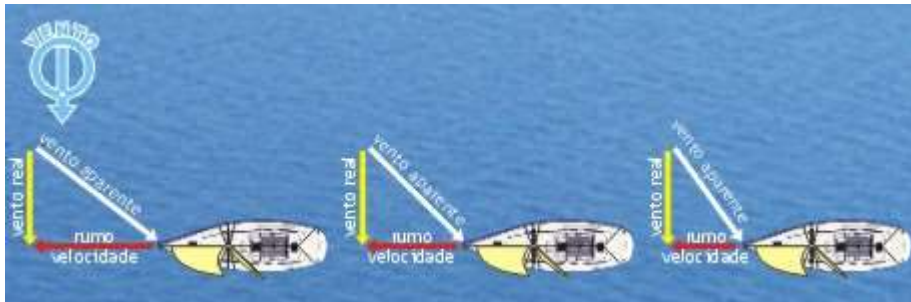


Figura 2. 4 - Vento aparente e real [7].

2.1.2 Zonas e Modos de Navegação

Como o objetivo é navegar autonomamente, é necessário perceber como se procede a navegação na zona branca da figura 2.3, isto é, como deve ser ajustada a vela e o leme nos diversos cenários.

- **Popa** - Este é o melhor cenário de navegação possível, com o vento no sentido da popa para a proa, isto é, o vetor da direção do vento está orientado na mesma direção da frente do barco. Assim, as velas devem ser folgadas ao máximo e o leme orienta o destino.
- **Largo** - Neste cenário o vetor da direção vento já não está completamente alinhado com a frente do barco, porém está muito próximo. Assim, as velas devem estar um pouco mais caçadas do que no cenário anterior.
- **Través** - Neste cenário o vento está perpendicular ao veleiro. Assim devemos ter as velas meias caçadas, isto é, entre o máximo e o mínimo de abertura e fecho das mesmas.

- **Bolina** - Este é o cenário máximo em que se consegue navegar contra o vento, isto é, temos o vento quase frontal. Assim, devemos caçar quase ao máximo possível as velas para que a quilha compense essa força, obtendo, dessa forma, movimento.

2.2 FASt: FEUP “Autonomous Sailboat”

Este projeto teve início em 2007 com um grupo de alunos do departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto, Portugal, com o âmbito de participar em competições internacionais organizadas pela Microtransat [8]. Assim, foi desenvolvido um veleiro autónomo com um comprimento total de dois metros e meio. O projeto foi bem sucedido, tendo participado em grandes eventos e competições onde foi reconhecido, tendo ganho o primeiro lugar na “World Robotic Sailing Championship” em 2012.



Figura 2. 5 - FASt: FEUP “Autonomous Sailboat” [2].

Para uma melhor compreensão do sistema implementado, será apresentada a organização do sistema eletrónico do veleiro [2] e do sistema de software embutido na plataforma utilizada.

2.2.1 Implementação do Hardware

A plataforma escolhida para implementar o sistema computacional foi uma FPGA (Field Programmable Gate Array), ligada a seis sensores e a dois servos motores, todos ligados a terminais da placa.

Os sensores são: bússola eletrônica; GPS (Global Positioning System); inclinômetro; anemômetro; sensor de posição do vento e sensor de ângulo da vela. Estes comunicam com a FPGA por diferentes tipos de protocolos de comunicação. Em seguida, e após a leitura dos valores lidos, estes serão processados resultando em valores de atuação nos dois servos motores através de um modulador PWM (Pulse-Width Modulation).

Como podemos observar na figura 2.6, o veleiro foi equipado com várias tecnologias para melhorar o comando e questões energéticas, tais como, baterias para alimentar todo o sistema, painel solar para gerar energia, uma antena RC (Radio Comando) para controlar o veleiro, uma antena Wi-Fi (Wireless Fidelity) para comunicar dados a curtas distâncias com a FPGA e uma antena IRIDIUM para comunicação por satélite a grandes distâncias. Todas essas tecnologias estão devidamente conectadas e interligadas à FPGA.

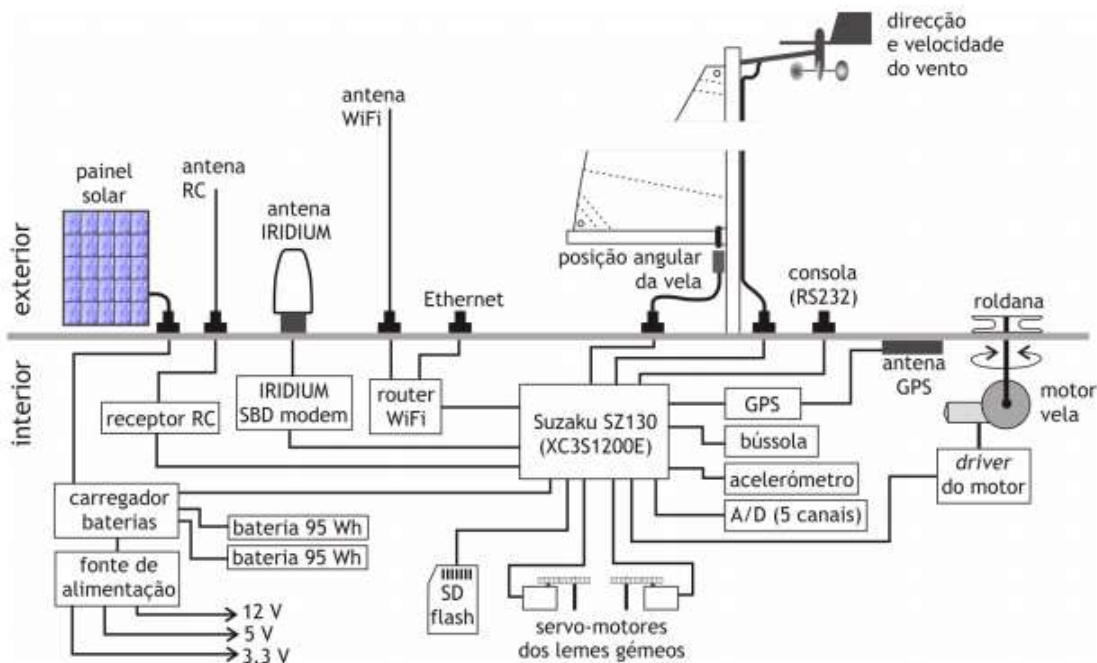


Figura 2. 6 - Organização do Hardware do veleiro FAST [9].

2.2.2 Implementação do Software

A implementação do software foi desenvolvida em C recorrendo a bibliotecas Linux. Quanto ao nível de navegação e controlo, estes estão divididos em diversos processos que comunicam mutuamente entre si através de “sockets” UDP (User Datagram Protocol), como ilustra a figura 2.7.

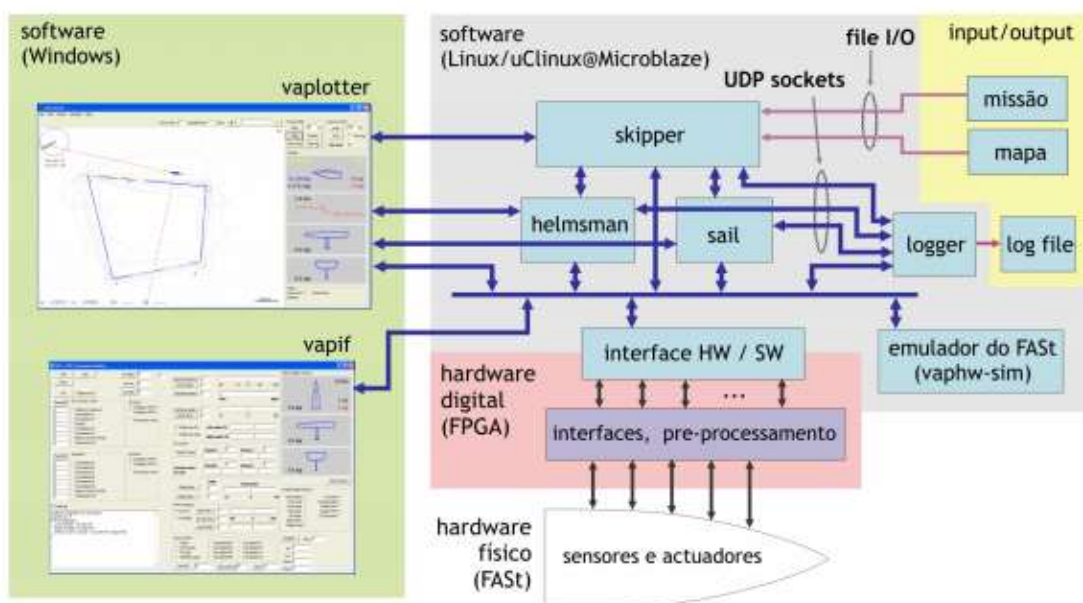


Figura 2.7 – Organização do software do veleiro FAST [9].

Para a validação do software foi desenvolvido um emulador (vapif) que em ambiente laboratorial permite ao utilizador testar diversos cenários de navegação possíveis, de forma a verificar a resposta na interface gráfica (vaplotter), permitindo uma monitorização em tempo real do veleiro. Logo, podemos monitorizar tanto navegação em tempo real como a simulada. Essa mesma aplicação permite ainda configurar rotas de navegação.

2.3 eVentos 2 - Autonomous Sailboat Control

O eVentos 2 [3] é uma dissertação de mestrado que continuou um projeto criado na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, para a criação de uma frota de veleiros autónomos.

Numa primeira fase, foi adquirido um veleiro telecomandado, como ilustra a Figura 2.8, que posteriormente seria adaptado para ser autónomo, o que não chegou a ser completamente realizado, tendo sido utilizado para testes em laboratório.



Figura 2. 8 - Veleiro do eVentos 2 [3].

2.3.1 Caracterização do Hardware

O hardware foi modelado para conter três sensores, um microprocessador e dois servos motores que atuam sobre o leme e as velas do veleiro, como ilustra a figura 2.9.

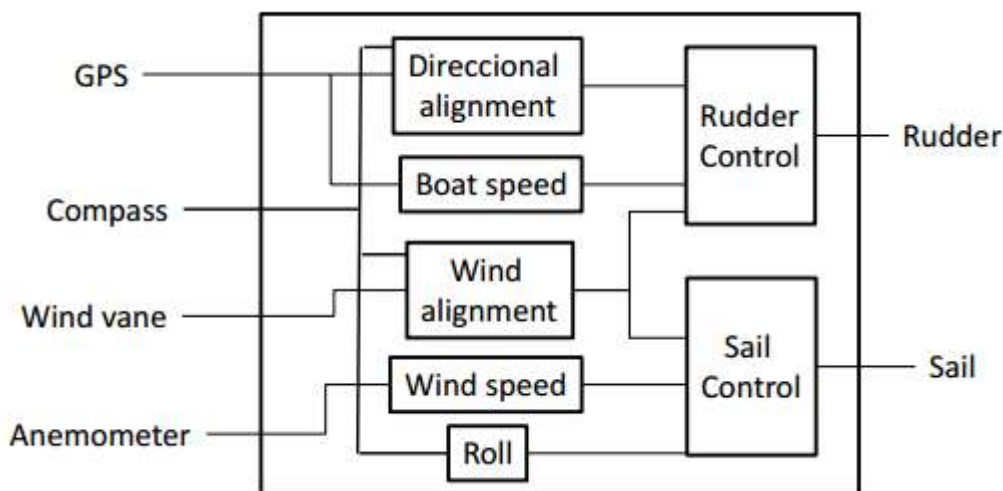


Figura 2. 9 Organização do hardware do veleiro eVentos2 [3].

O GPS, a bússola e o catavento fazem leituras de valores em tempo real e são introduzidos na plataforma escolhida. Os paramentos são convertidos em valores que são compreendidos pelos controladores e posteriormente geram valores de atuação nos dois servo motores que controlam a posição do leme e da vela.

2.3.2 Caracterização do Software

O software deste veleiro foi pensado com uma implementação de dois controladores de lógica difusa, criados através de uma plataforma de geração de código automático, o xFuzzy. Inicialmente é feita uma aquisição de valores vindos dos sensores, através de diferentes protocolos de comunicação. A partir desses valores é implementado um código em linguagem específica do Arduino, para depois o controlador tomar uma decisão com base nesses valores. Em seguida é atribuída uma saída, afetando os servo motores, como podemos verificar na imagem 2.9.

Ambientes de Desenvolvimento e Execução

Após analisar as técnicas de navegação à vela, poderão ser ajustadas estas técnicas de forma a obtermos um veleiro autónomo.

Para a navegação à vela ser autónoma, é necessário um controlador. A solução passa por obter controladores difusos e, através de uma plataforma de controlo selecionada, analisar sensores e atuar em servomotores com o auxílio do controlador, para desta forma conseguir aplicar uma técnica de navegação à vela.

3.1 Descrição do Hardware

Para o desenvolvimento do controlador é necessário uma plataforma capaz de processar toda a informação lida pelos sensores e atuar nos atuadores de forma rápida. Então, a plataforma escolhida passa por um Arduino, pois é de custo reduzido, de fácil aprendizagem e suporta diversos protocolos de comunicação, o que o torna de fácil gestão com várias funcionalidades e de, num futuro próximo, incluir mais funcionalidades no projeto.

3.1.1 Arquitetura do Hardware Embutido

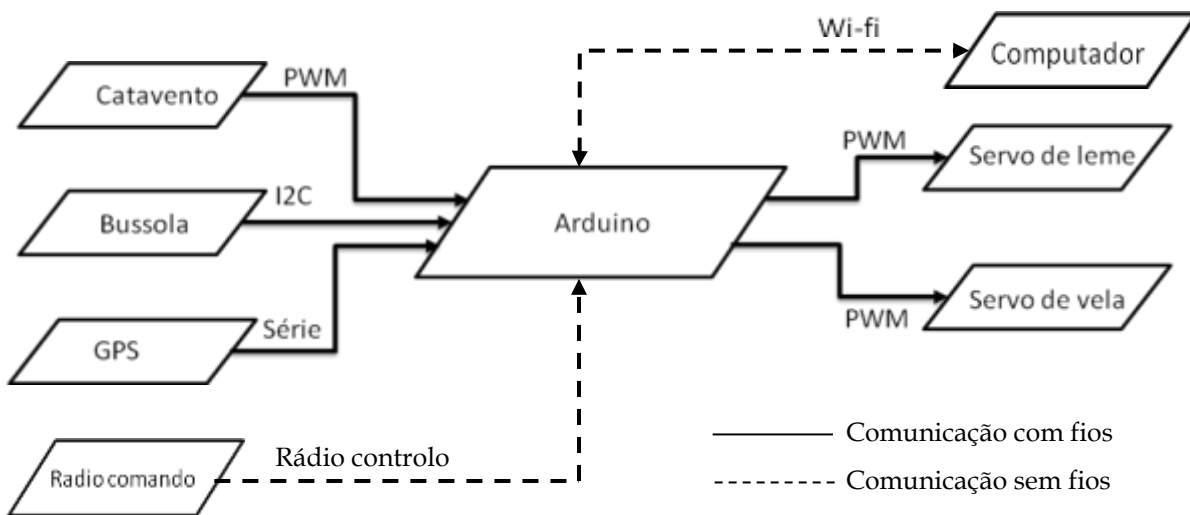


Figura 3. 1 - Arquitetura do hardware embutido.

A arquitetura do hardware [10] está idealizada de forma a ter todo o suporte necessário para a navegação autónoma e manual, isto é, está equipado com todos os sensores necessários para a navegação autónoma e ainda conta com o recetor de radio frequência para poder comunicar com um comando que controla o veleiro de forma manual. Conta ainda com uma ligação Wi-Fi com o computador de forma a receber e a transmitir dados para o apoio à navegação autónoma e a monitorizar o andamento do controlador e dos sensores ao longo da navegação.

3.1.2 Plataforma

O desenvolvimento do projeto foi implementado num Arduino mega 2560 R3 [11] como ilustra a figura 3.2.



Figura 3. 2 - Arduino mega 2560 [11].

O Arduino mega 2560 é um microcontrolador, com 54 pins digitais de entradas e saídas, entre as quais, saídas PWM, entradas analógicas, portas série e portos digitais. Conta, também, com diversos dispositivos que servem de apoio ao microcontrolador.

A programação é feita através de um software do Arduino e transmitida ao dispositivo via USB (Universal Serial Bus). Um resumo do dispositivo é feito na figura 3.3.

Microcontrolador	ATmega2560
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limites)	6-20V
Digital I / O pins	54 (dos quais 15 fornecer saída PWM)
Analog pinos de entrada	16
Corrente DC por I / O Pin	40 mA
Corrente DC 3.3V para Pin	50 mA
Memória Flash	256 KB, dos quais 8 KB usados pelo bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidade do relógio	16 MHz

Figura 3. 3 - Resumo do Arduino 2560 [11].

3.1.3 Sensores e Aquisição de Dados

Para a implementação do controlador é necessário adquirir dados do meio envolvente relativos à direção aparente do vento, à posição atual do veleiro e à orientação do mesmo. Assim, foram escolhidos três sensores: um catavento, um GPS e uma bússola.

- **Catavento** - A estrutura do catavento é feita a partir de um integrado AS5161 que analisa a variação do campo magnético. A estrutura é conseguida através de uma peça impressa numa impressora 3D, como podemos ver na figura 3.4.

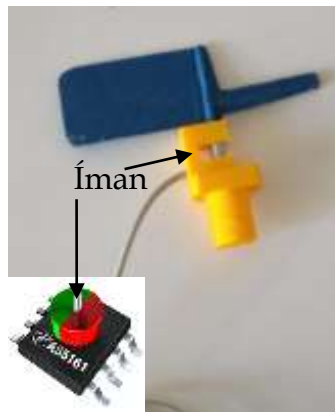


Figura 3. 4 - Catavento e sensor AS5161 [12].

Através da variação do campo magnético é possível obter diversos valores para diferentes posições do catavento, por forma a descobrir a orientação do vento. A comunicação desses valores para o Arduino é feita por PWM. Após a leitura dos dados é necessário realizar cálculos para calibrar o sensor de forma a ter valores de ângulos variáveis de 0° a 180° , como ilustra a figura 3.5.

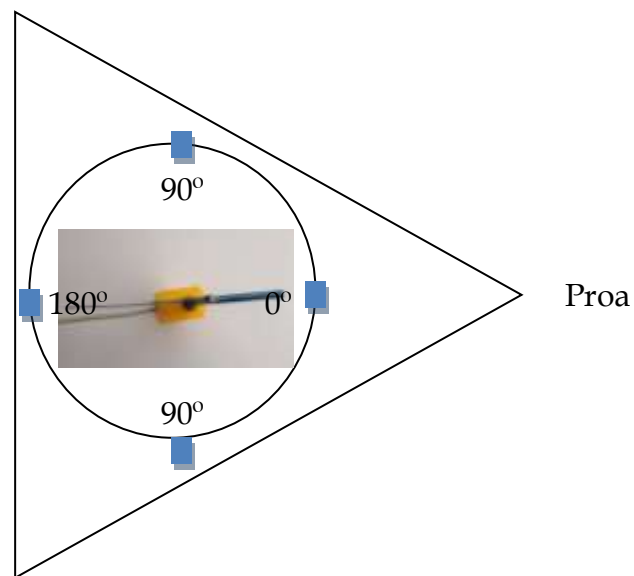


Figura 3. 5 – Posição do catavento em relação ao veleiro, 0°.

- **GPS** - Para saber a localização do veleiro foi escolhido um modelo MT3329 [13] que utiliza uma antena GPS para a captação de sinais em formato NMEA (National Marine Electronics Association) [14] de diferentes satélites e fazer uma triangulação entre eles de forma a obter uma coordenada, latitude e longitude. Para a comunicação com o Arduino é utilizado um protocolo série com uma taxa de atualização de 10 Hz. Este ainda conta com um led que sinaliza se estiver sintonizado com os satélites.



Figura 3. 6 – GPS MT3329.

- **Bússola** - Para a saber a orientação do veleiro em relação ao norte geográfico, é utilizado um sensor CMPS10 [15] usando um protocolo I2C (Inter-Integrated Circuit) de comunicação com o Arduino, afetando registros de 23 bytes, retornando os valores de “pitch”, “roll” e “yaw” [16], que estão identificados na figura 3.7.

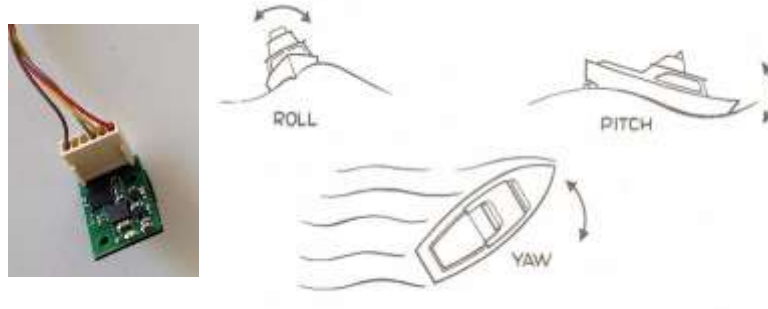


Figura 3. 7 - Bússola CMPS10 e orientações.

3.2 Controlo de Navegação e Ferramentas

Para o controlo de navegação é necessário analisar os parâmetros e gerar valores de atuação por forma a poder cumprir um objetivo pré-definido.

O tratamento da informação pode então ser realizado de inúmeras formas, desde controladores difusos, eventos discretos, proporcional, integral e derivativo (PID (Proportional-Integral-Derivative)), entre outros.

Para esta dissertação, o controlo é feito através de um controlador difuso. Desta forma é utilizada uma ferramenta de geração de código automática, que recorre a funções de pertinência para delinear os cenários previamente definidos, onde o gerador é o Xfuzzy [17].

3.2.1 Controladores Difusos

A lógica difusa [18] [19] consiste na representação da ideia em conjuntos ordinários, ou convencionais. Pode ser visto como o acerto de métodos e operações difusas que permitem a modelação de sistemas ou de controladores, sendo sempre formulado em conjuntos e regras. Assim, em vez de obter um controlador numa linguagem natural, é possível convertê-la num formato numérico de manipulação fácil, isto é, modelizar a imprecisão do raciocínio humano, tendo vários níveis de decisão, ou seja, distinguir o “alto” do “muito alto”. A lógica difusa é uma forma conveniente de mapear um espaço de entradas num espaço de saídas.

Para a implementação do controlador difuso é necessário traduzir objetivos em regras, determinar as funções de pertinência para os antecedentes e para os consequentes, substituir expressões verbais por códigos de forma a facilitar a implementação e, por fim, executar o código difuso, como ilustra a figura 3.8.

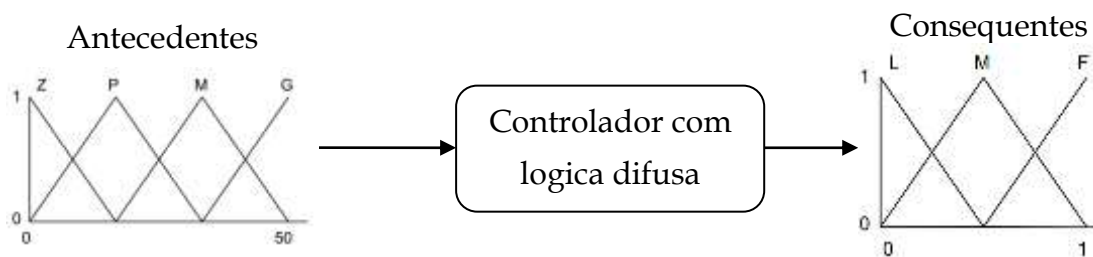


Figura 3. 8 – Implementação do controlador difuso.

Para modelizar o sistema são utilizadas funções de pertinência [20], através de curvas que definem como cada ponto de entrada é mapeado para um valor de pertinência (0 a 1). As funções de pertinência mais comuns são triangular, trapezoidal e gaussiana, descritas na figura 3.9.

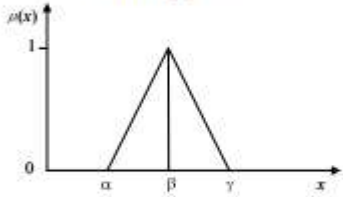
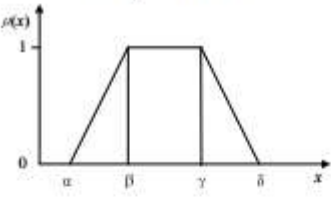
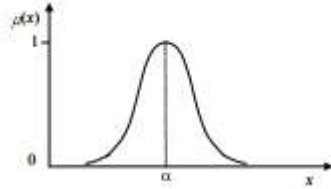
Função de pertinência	Caracterização
<p style="text-align: center;">Triangular</p> 	<p style="text-align: center;">$\Lambda : X \rightarrow [0; 1]$</p> $\Lambda(x, \alpha, \beta, \gamma) = \begin{cases} 0 & x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \alpha \leq x \leq \beta \\ \frac{\gamma-x}{\gamma-\beta}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

Figura 3. 9 - Tipo de funções de pertinência [21].

A estrutura de um controlador lógico difuso é composta por quatro módulos principais, o “fuzificador” [22], a base de conhecimento, o mecanismo de inferência e o “desfuzificador”, apresentados na figura 3.10.

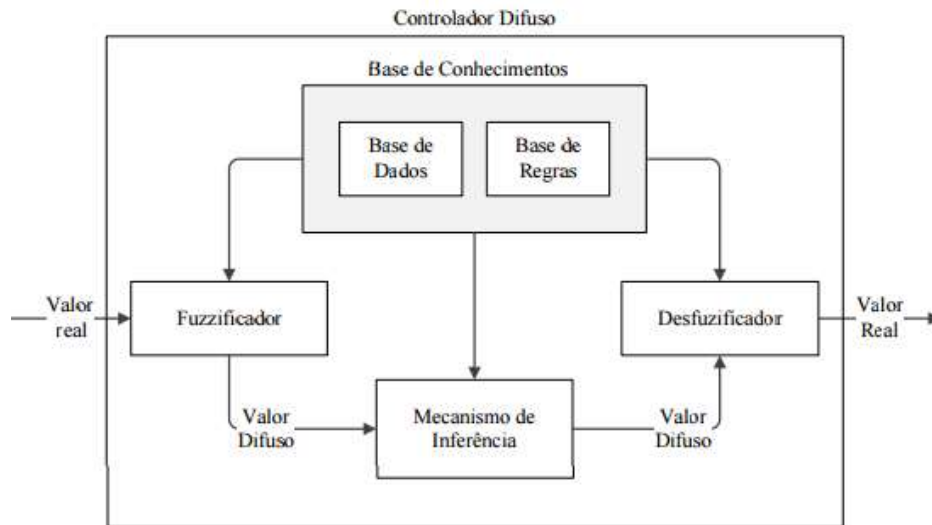


Figura 3. 10 – Estrutura de um controlador difuso [21].

Em seguida, a base de conhecimento é o elemento chave deste controlador, que deve ser melhorada por forma a que todo o controlador difuso resulte em processos mais eficientes. Para tal, é necessário uma etapa de aprendizagem, isto é, recolha de dados do sistema a controlar e assim compreender quais as alterações e melhoramentos a fazer. Quanto melhor for a aprendizagem, melhor será o desempenho deste tipo de controladores.

Quanto à parte de “fuzificação”, esta consiste em passar de valores crespos para valores “fuzificados” através de funções de pertença, sendo que a “desfuzificação” é o caminho inverso. Existem diversos mecanismos para realizar a “desfuzificação”, tais como os métodos do “centroide”, primeiro dos máximos, método da média dos máximos, entre outros.

Por fim, os mecanismos de inferência permitem o cálculo dos valores de saída do controlador tendo por base os atributos de cada regra que modelam o sistema a controlar.

3.2.2 Ferramenta de Geração Xfuzzy

A ferramenta Xfuzzy [17] integra um conjunto de ferramentas que facilitam ao utilizador conceber controladores para sistemas baseados em lógica difusa. A partir de uma interface gráfica permite introduzir as funções de perten-

ça e regras, definindo assim os antecedentes e os consequentes como ilustram as figuras 3.11 e 3.12.

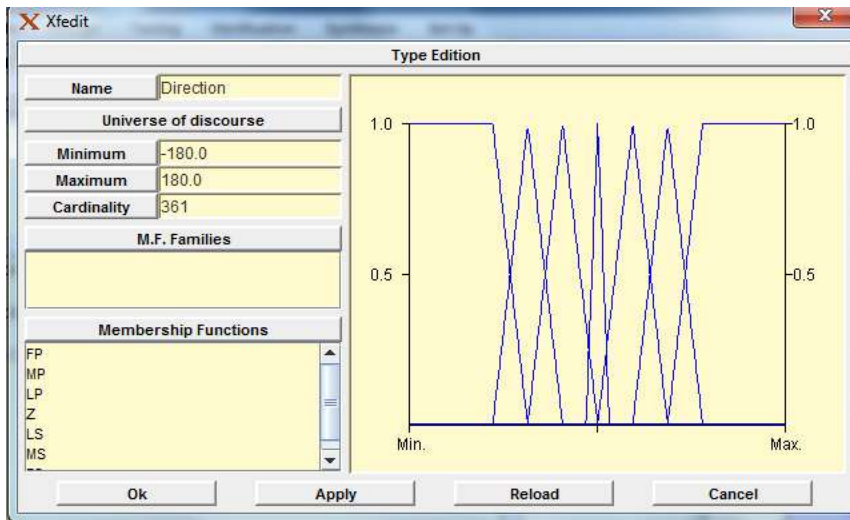


Figura 3. 11 – Função de pertença dos antecedentes.

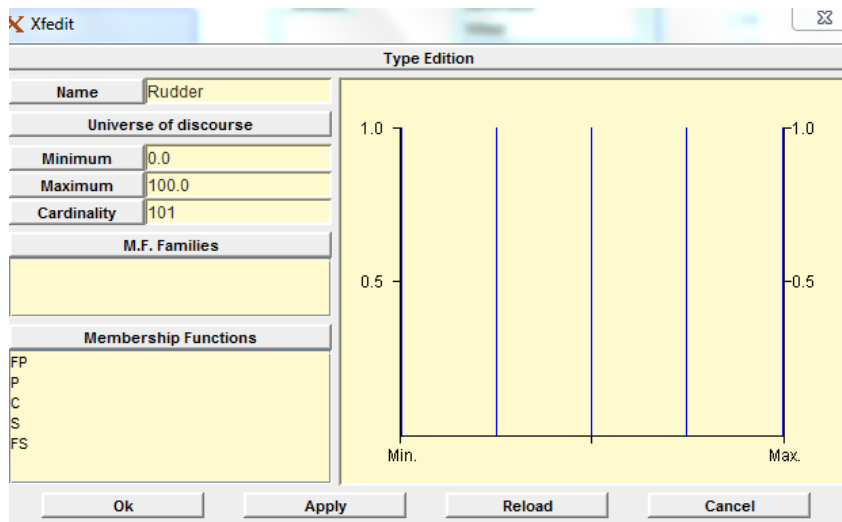


Figura 3. 12 – Função de pertença dos consequentes.

Após a definição de todas as funções de pertença é necessário criar regras para assim serem as condições que modelam o sistema e, por fim, agrupar todo o sistema, como ilustra a figura 3.13.

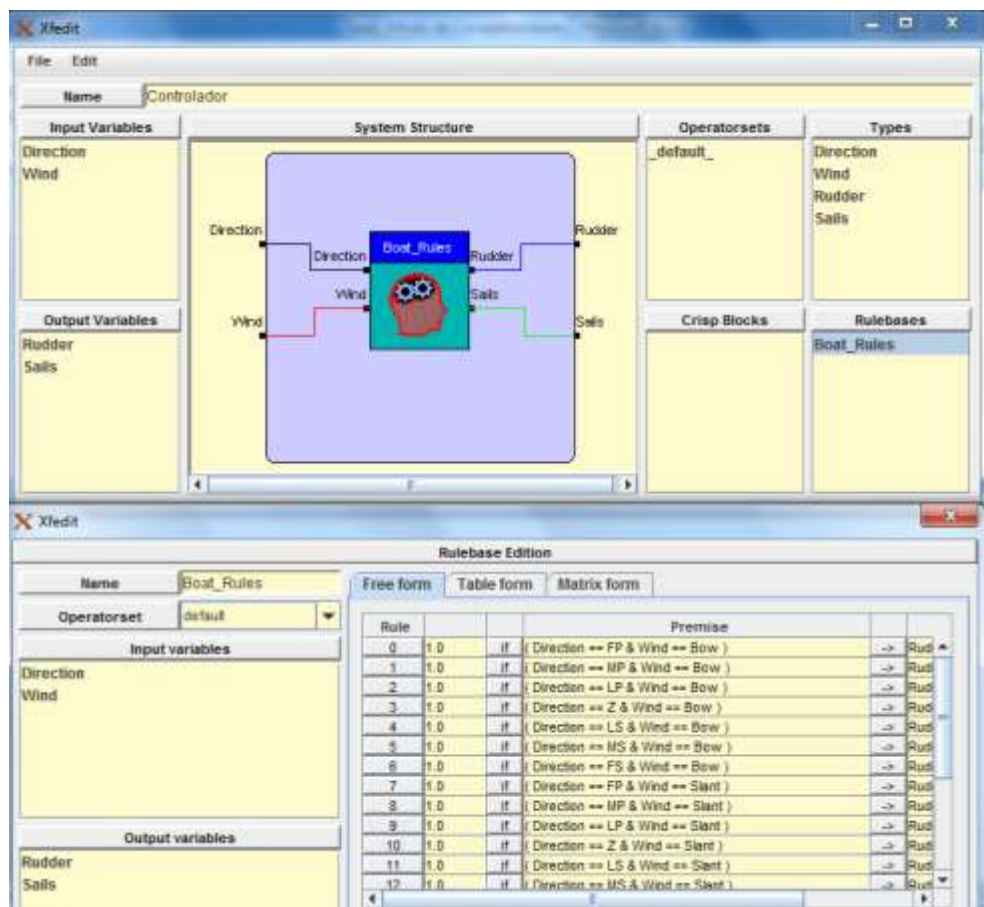


Figura 3. 13 – Ferramenta de geração automática de código Xfuzzy.

Após a implementação do controlador em lógica difusa, a ferramenta gera um código automático em diversas linguagens de programação, tais como java, c, c++ entre outras, por forma a poder ser integrado em outras plataformas.



Controlador Desenvolvido

No presente capítulo vai ser descrito o trabalho executado nesta dissertação. Em primeiro lugar, surge o conceito de controlo manual e autónomo, isto é, o utilizador pode comandar o veleiro através de um radio comando ou através de um controlador que é monitorizado por uma Aplicação de Gestão da navegação.

Para finalizar, pretende-se que o veleiro seja capaz de navegar para qualquer objetivo, quando cumpra o pré-requisito de não ter vento desfavorável.

4.1 Implementação de Controlo Autónomo e Manual

Para o controlo autónomo do veleiro é necessária a construção de um mecanismo com a capacidade de comutar entre um modo manual e autónomo, por forma a poder ter controlo do mesmo em diversas situações, tais como falhas e avarias do sistema, manobrar o mesmo em situação de teste e de vento desfavorável, zona não navegável, pois o controlador não suporta esta especificação.

O modo de comutação implementado foi através de um canal de rádio frequência, que dá a capacidade ao veleiro para variar entre modo manual e autónomo, figura 4.1. Ao nível de hardware no Arduino, o recetor de rádio frequência é incorporado através de um protocolo PWM.

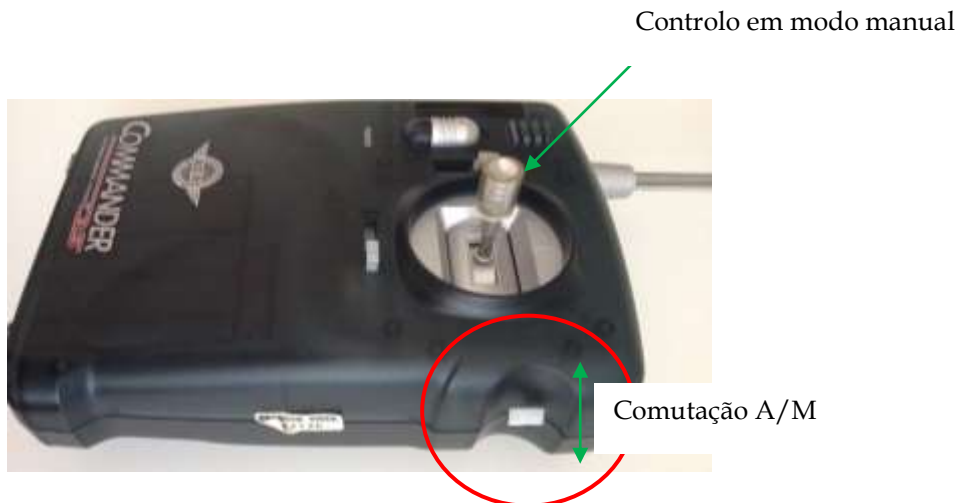


Figura 4. 1 - Meio de comutação por rádio frequência.

Ao nível de software, deve-se escutar o sinal recebido através do canal de rádio frequência de comutação (A/M), referido na figura 4.1, e desta forma selecionar como gerar os sinais de saída PWM para os servomotores, se resultantes das ações do controlador de navegação autónoma, se provenientes do comando remoto. Esta especificação tem prioridade acima de qualquer outra instrução do controlador autónomo.

Após esta decisão, se estivermos perante o modo manual, o software embutido faz a leitura dos canais vindos do rádio através do mesmo protocolo, associando ao leme e vela os valores lidos aos valores de atuação nos servos motores, como ilustram as tabelas 4.1 e 4.2.

Valores lidos RF	Decisão de atuação no leme
> 1650	Tudo a estibordo
[1400;1650]	Estibordo
[1300;1400[Centrado
[1050;1300[Bombordo
<1050	Tudo a bombordo

Tabela 4. 1 - Conversão de valores de atuação no leme.

Valores lidos RF	Decisão de atuação na vela
> 1650	Tudo folgado
[1400;1650]	Folgado
[1300;1400[Centrado
[1050;1300[Caçado
<1050	Tudo caçado

Tabela 4. 2 - Conversão de valores de atuação na vela.

Por fim, em modo autónomo, o software embutido irá recorrer a uma estrutura de controlo para a tomada de decisão sobre a atuação que deverá ocorrer mediante o cenário em que o veleiro se encontra, sendo este descrito na secção 4.2.

4.2 Integração Wi-Fi com Arduino

Em primeiro lugar, a interação entre o Arduino e o Wi-Fi é um dos aspectos mais importantes no controlo do veleiro, pois este é o meio de análise dos resultados obtidos e de interação com o utilizador, isto é, informa o mesmo das leituras dos diversos sensores e ainda possibilita o envio de novos objetivos.

Para tal foi implementado uma comunicação série UART (Universal asynchronous receiver/transmitter)) entre o Arduino e a placa Wi-Fi e um protocolo de transporte UDP entre o computador e a placa Wi-Fi.

Inicialmente, a comunicação série com a plataforma é feita através de um canal que está ligado aos portos digitais RX (Receiver mode) e TX (Transmitter mode) que se encontram ligados ao microcontrolador que é responsável pela interpretação do sinal.

Após a ligação dos meios de comunicações, figura 4.2, foi então testada a comunicação a nível de velocidade e de perdas de informação.

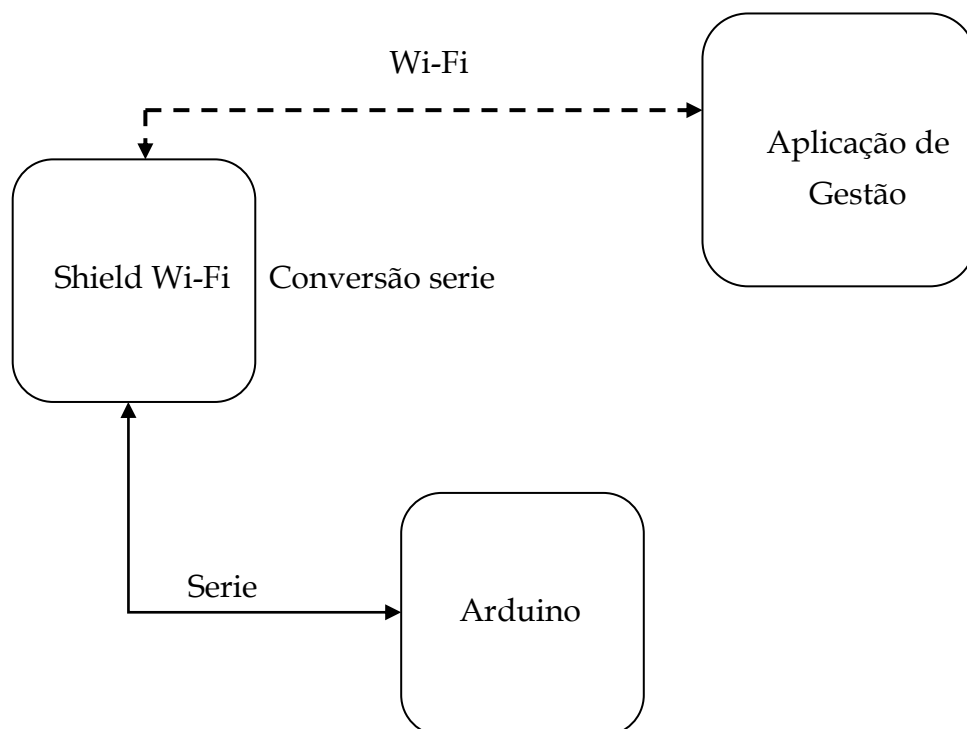


Figura 4. 2 - Integração Wi-fi - Arduino.

Os testes começaram pela troca de informações curtas entre o controlador e o computador. Com essas mensagens trocadas é possível observar uma elevada taxa de transmissão e recepção de dados, resultando numa perda nula de informações.

Seguidamente é aumentado o volume de tráfego, isto é, aumenta-se o tamanho das mensagens para cinquenta bytes. Porém, com o aumento da taxa de informação transmitida, verificou-se uma recepção não eficaz de dados e, desta forma, a plataforma apresentou um comportamento de saturação.

Esta saturação de dados deve-se ao facto de que a comunicação é convertida para série, que está limitada por uma taxa de transmissão máxima, por outras palavras, a velocidade de transmissão entre o computador e a placa Wi-Fi é maior do que a velocidade de transmissão entre a placa Wi-Fi e a plataforma. É, assim, necessário encontrar um valor de envio de dados razoável para que a comunicação se realize com sucesso para tal são necessárias várias experiências para aferir o valor mais adequado a este caso.

Logo, a limitação da ligação do computador com a plataforma no sistema impede, assim, que os dados enviados pelo computador sejam muito extensos, não sendo possível enviar dados em rajadas com bit rate muito elevado.

Em seguida, outro dos problemas detetados foi o aumento do consumo energético gerado pela introdução do Wi-Fi, com o frequente envio de informação entre o computador e a plataforma. Devido a este consumo é necessário ajustar melhor a taxa de transmissão de dados.

Para concluir, todos os testes realizados permitiram entender que deve haver um balanceamento entre a quantidade máxima permitida pelo sistema para que, deste modo, exista sucesso na transmissão e ainda permita regular a taxa de informação, enviando apenas informação essencial para o bom funcionamento e análise dos parâmetros do veleiro, resultando assim numa eficiência ao nível energético e do envio de dados.

4.3 Integração Radio Controlo - Arduino

A integração do rádio controlo com a plataforma é muito importante para conseguir seleccionar a comutação entre autónomo e manual.

Numa primeira fase é colocada a plataforma entre o recetor e os servos (figura 4.3), para ser possível embutir esta característica com o modo de funcionamento já desenvolvido, ou seja, modo autónomo.

Após a ligação, é necessário configurar o software para efetuar uma leitura do porto de controlo, e assim que este fique num modo manual passa o controlo a ser executado com a informação vinda do recetor de rádio.

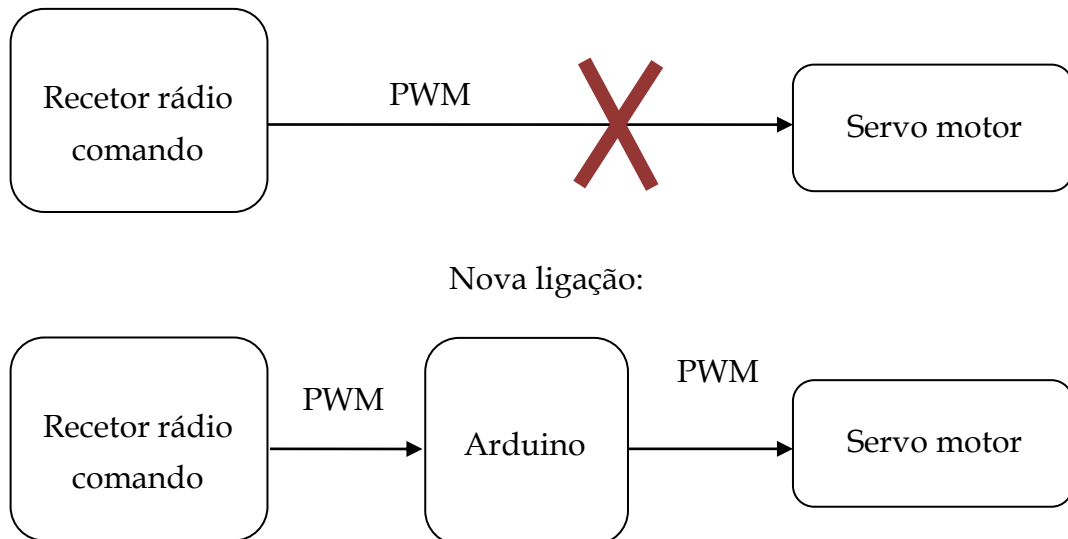


Figura 4. 3 - Integração Radio controlo - Arduino.

Posteriormente, com esta nova ligação é de notar que a resposta dos servos aos comandos manuais resultam num ligeiro atraso, porém esse é insignificante, isto porque a velocidade do veleiro é baixa, não se justificando uma resposta mais rápida. Este atraso deve-se ao facto de as instruções na plataforma serem sequenciais e esta não suportar mecanismos de multitarefa.

Em seguida, verificou-se que o funcionamento de todos os módulos em conjunto afetaram este mecanismo com um ligeiro ruído, devido à alta frequên-

cia imposta pela antena de Wi-Fi. Para atenuar este efeito é introduzido um filtro por software e por hardware.

Quanto ao primeiro, é introduzido através de intervalos de operação para, assim, um valor estar restrito a um intervalo de valores, constituído por vários valores. No segundo, foi introduzido um filtro para eliminar a componente de alta frequência ("choke") na linha de alimentação do sistema.

4.4 Proposta para Estrutura Hierárquica do Controlador

Para o controlo em modo autónomo do veleiro é, então, necessário estruturar a lógica para o processamento de dados e desta forma realizar o controlo do mesmo.

Nas secções seguintes será apresentado o modelo do controlo, especificando a estrutura hierárquica e ainda entradas e saídas do sistema, isto é, as variáveis de controlo.

4.4.1 Estrutura de Três Níveis do Controlador

A proposta de controlo [23] baseia-se numa estrutura em três níveis, considerando um desenvolvimento baseado num modelo. Esta abordagem trás vantagens para o possível melhoramento dos objetivos pretendidos, sendo abordado nesta dissertação o *nível 1 - Controlador de navegação de baixo nível* da estrutura descrita na figura 4.4.

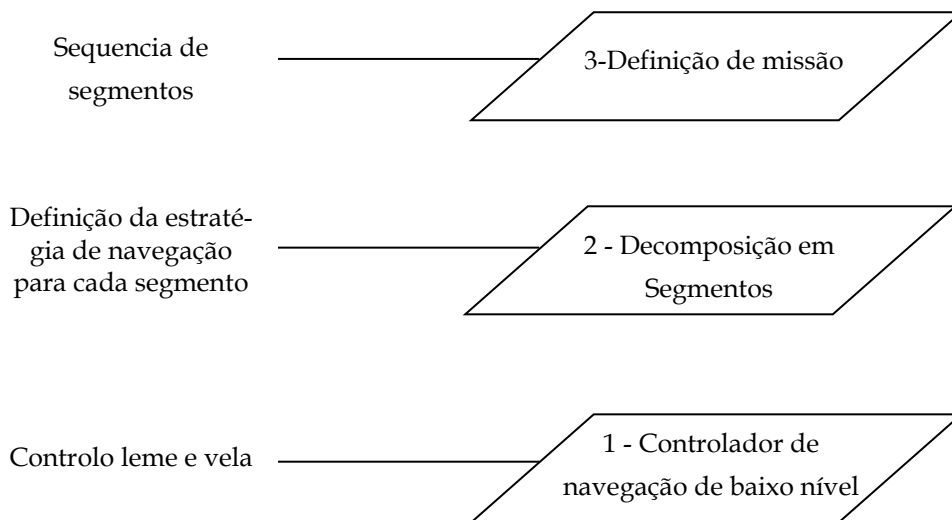


Figura 4. 4 - Estrutura hierárquica do controlador.

O nível mais acima, *nível 3 - Definição de missão* diz respeito à especificação de características do modo regata, ou seja, a capacidade de definir vários

objetivos intermédios. Em seguida o *nível 2 – Decomposição em segmentos* trata do controlo quando o vento se encontra desfavorável, isto é, quando o vento se encontra em zona sem navegação, sendo necessário criar objetivos intermédios. Estes dois níveis são retratados numa outra dissertação.

Quando ao *nível 1 – Controlador de navegação de baixo nível*, tem como objetivo o controlo do leme e vela em situação de vento favorável à navegação, sendo este descrito nas secções seguintes.

Cada nível tem objetivos específicos que interagem, porém os níveis mais baixos são independentes dos níveis mais altos, isto é, o nível 2 e o nível 3 funcionam como complemento e suporte para o nível 1 conseguir realizar tarefas mais complexas que não são abordadas nesta dissertação.

Assim, este controlador é o nível mais elementar para toda a navegação de vento favorável. Desta forma podemos recolher e analisar dados para realizar ajustes no mesmo, não influenciando os restantes níveis.

4.4.2 Arquitetura do Controlador de Navegação (nível 1)

A implementação do software embutido [10] é construído da construção de um modelo que conta com dois grandes blocos, um embutido no veleiro e uma aplicação de gestão instalada num computador para fazer a gestão de todos os dados online, sendo esta apresentada a azul nas figuras 4.5 e 4.7.

Em seguida, na figura 4.5, é apresentada a arquitetura de software embutida, que é constituída por dois módulos essenciais de um pré-processamento e de um controlo em lógica difusa.

O funcionamento inicia-se com a análise de quatro valores de entrada, que correspondem à aquisição sensorial e o estabelecimento de um objetivo. Desta forma, gera valores de saída essenciais para o controlador de lógica difusa que, por sua vez, atribui valores a cada uma das saídas, leme e vela.

Toda a análise do pré-processamento é feita através de uma aplicação que monitoriza todos os parâmetros lidos e obtidos por forma a criar uma base de dados para ser posteriormente revista e, assim, melhorar o funcionamento de todo o sistema.

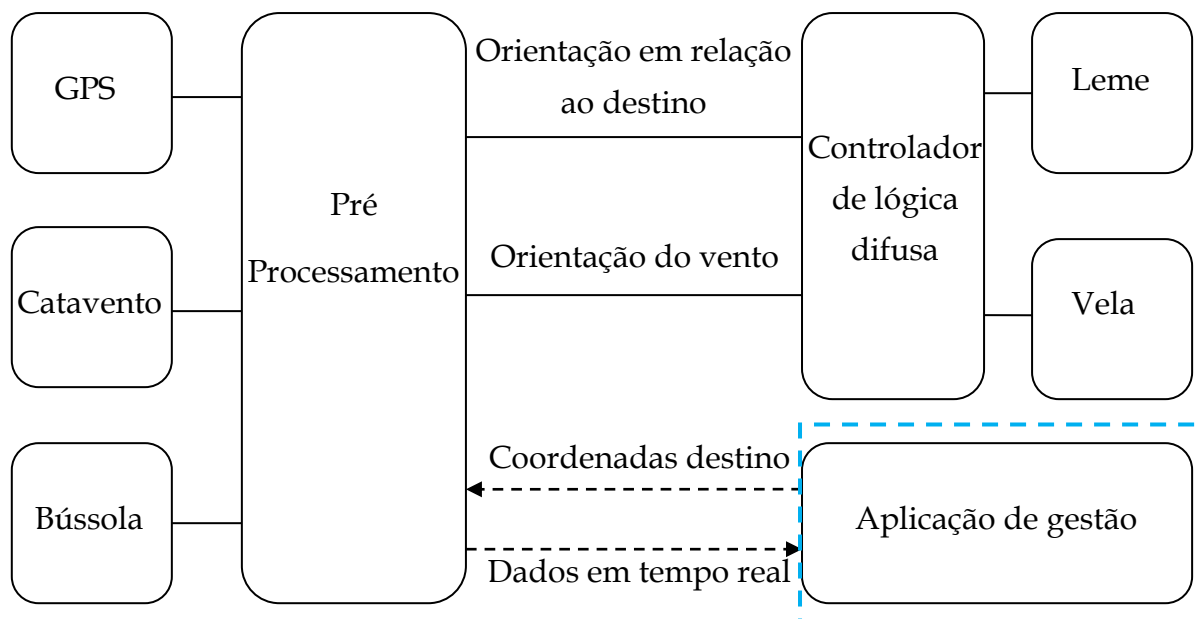


Figura 4. 5 – Arquitetura do software embutido.

4.4.3 Pré-processamento

Este módulo de pré-processamento tem como base gerar dois valores de saída para o controlador principal tomar uma decisão de atuação sobre os servos, sendo as entradas, o ângulo de orientação do destino e a orientação do vento. A orientação do destino é o ângulo que a proa faz com o vetor de direção ao destino, como pode ser visto na figura 4.6.

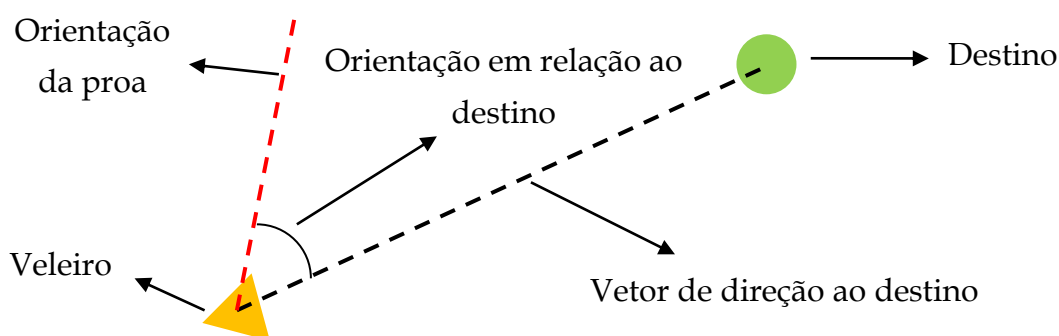


Figura 4. 6 – Esquema da orientação do destino.

A orientação do vento é simplesmente o vento aparente que é medido pelo catavento em relação à proa, como foi descrito na secção 3.1.3.

Para a geração desses dois valores é então necessário conhecer quatro variáveis de entrada, a coordenada GPS do ponto atual do veleiro, a orientação da bússola e do catavento e, por fim, a coordenada GPS do destino introduzido numa aplicação. A partir dessas variáveis são realizadas operações matemáticas para a geração das duas variáveis de saída, como está representado na figura 4.7.

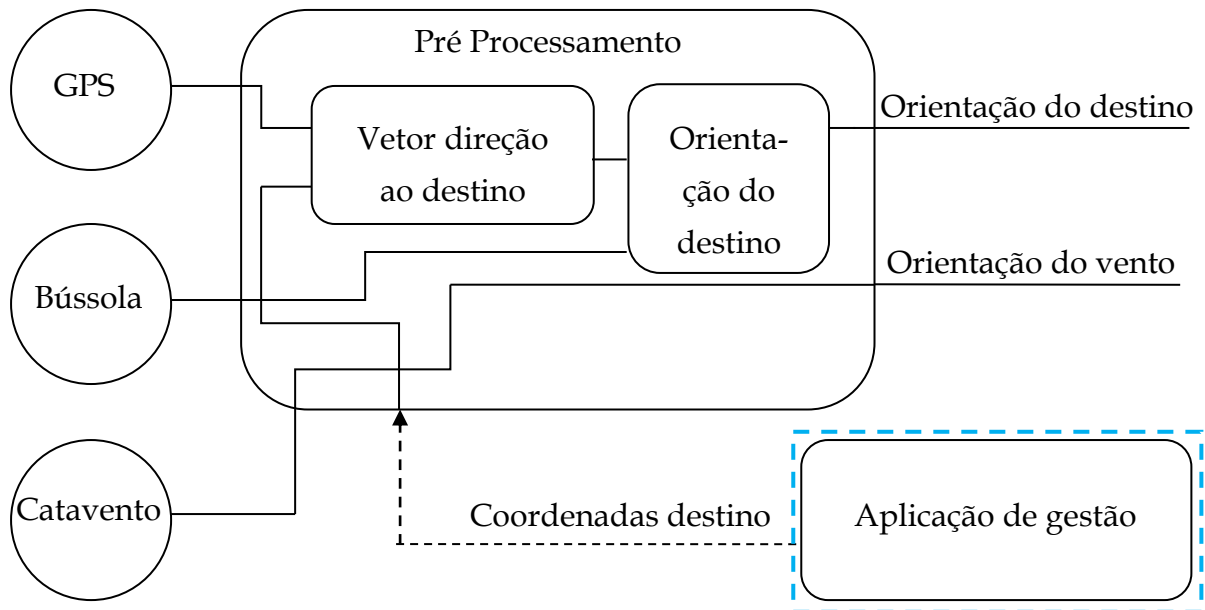


Figura 4. 7 - Esquema do Pré processamento

4.4.4 Proposta de Controlador

Para a implementação do controlador em situação de vento favorável, este foi elaborado a partir de uma lógica difusa. Para tal, torna-se necessário encontrar métodos que definam pontos intermédios, isto é, definir um “meio-termo” como descrito na secção 3.2.1. O controlador previamente caracterizado [10] foi alterado, retirando-se um dos cinco parâmetros de entrada, de forma a simplificar o mesmo.

Assim sendo, para agilizar o processo de construção e implementação dos conjuntos difusos é necessário encontrar ferramentas para este propósito, neste caso o xFuzzy para a implementação, pois este tem a capacidade de gerar um código automaticamente a partir de um modelo gráfico, como foi descrito na secção 3.2.2.

Conjuntos Difusos

Para a construção do controlador de lógica difusa é, então, necessário ter presente as entradas e saídas que definem o sistema como ilustra a figura 4.8. Com estas entradas e saídas é necessário parametrizar a partir de funções de pertinência.

Para a criação de funções e regras que definem o sistema é necessário estar de acordo com os cenários de navegação à vela para tal recorre-se aos conhecimentos de velejar obtidos pelos especialistas e conhecedores desta modalidade, obtendo-se, desta forma, um controlador difuso base.

Uma vez construído um controlador base, é necessário aumentar a sua inteligência, isto é, o controlo baseia-se não só no conhecimento da matéria, mas também na geometria e forma do veleiro. Desta forma, é necessário encontrar mecanismos de aprendizagem e otimização das variáveis e regras de controlo.

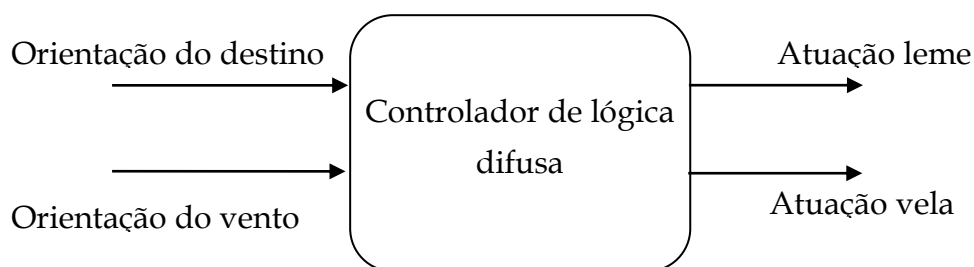


Figura 4. 8 - Entradas e saídas do controlador.

As entradas do controlador são geradas pelo pré-processamento. Neste caso trata-se do ângulo de orientação com o destino e a orientação do vento aparente (obtida pelo instrumento de medida, cata-vento).

O ângulo de orientação com o destino é representado numa gama de valores compreendidos entre -180° a $+180^\circ$. Para compreender estes ângulos (figura 4.9), nos diferentes destinos possíveis representados a amarelo, é possível verificar qual o ângulo em relação à proa, sendo definido que para virar a estibordo é atribuído uma variação de ângulos de $+180^\circ$ e para bombordo -180° .

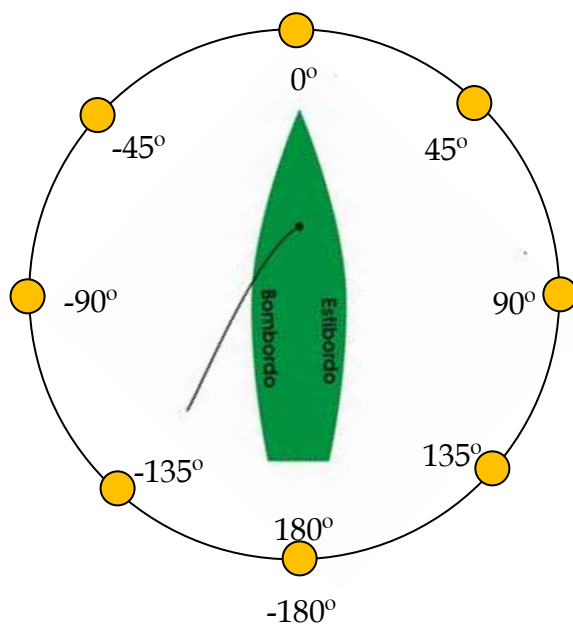


Figura 4. 9 - Ângulo de orientação do destino.

É necessário agora a parametrização dos valores possíveis do ângulo de orientação do destino recorrendo a um formalismo de funções de pertinência.

Para a caracterização destas funções foi proposto o formato triangular, descrito na seção 3.2.1, devido à maior simplicidade e representação dos diferentes cenários, sendo esta proposta uma boa caracterização dos cenários a representar (figura 4.10).

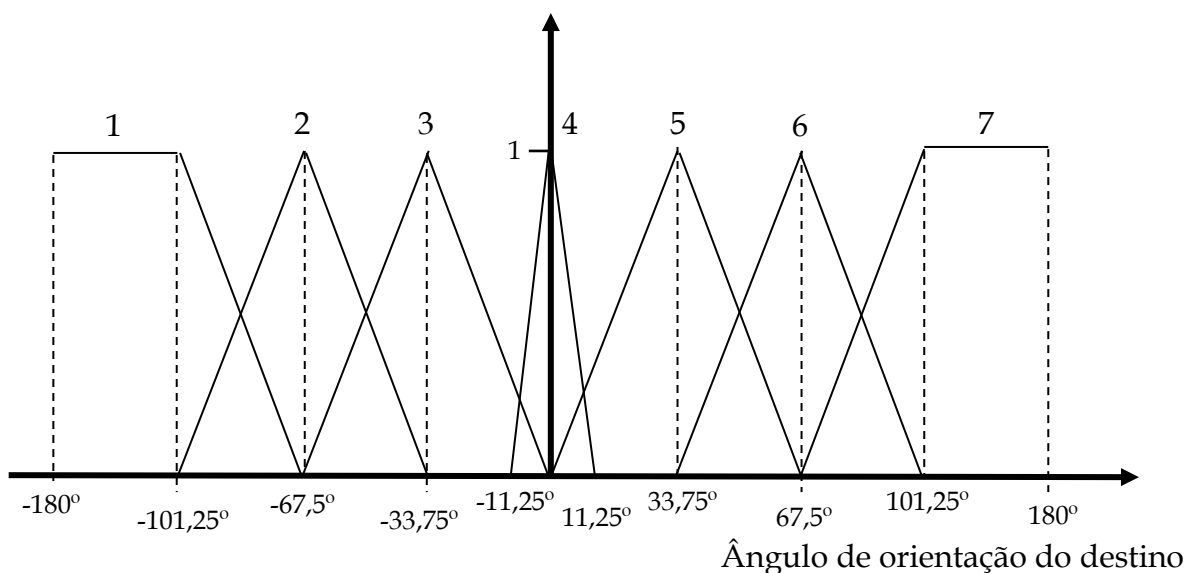


Figura 4. 10 - Funções de pertinência dado o ângulo de orientação do destino para o leme.

As associações das funções de pertinência com os cenários possíveis relativos à orientação da proa com o destino estão representadas na tabela 4.3.

Funções de pertinência	Orientação do destino em relação a proa
1	Tudo para a esquerda
2	Esquerda
3	Um pouco à esquerda
4	Centrado
5	Um pouco à direita
6	Direita
7	Tudo para a direita

Tabela 4. 3 - Legenda das funções de pertinência da orientação do veleiro.

Seguidamente, para o segundo parâmetro de entrada, ou seja, o ângulo de orientação relativo ao vento aparente, este varia numa gama de valores entre 0° e 180° , descrito na secção 3.1.3, na figura 3.5.

Tal como no caso anterior é necessário parametrizar os valores num formalismo de funções de pertinência, como ilustra a figura 4.11.

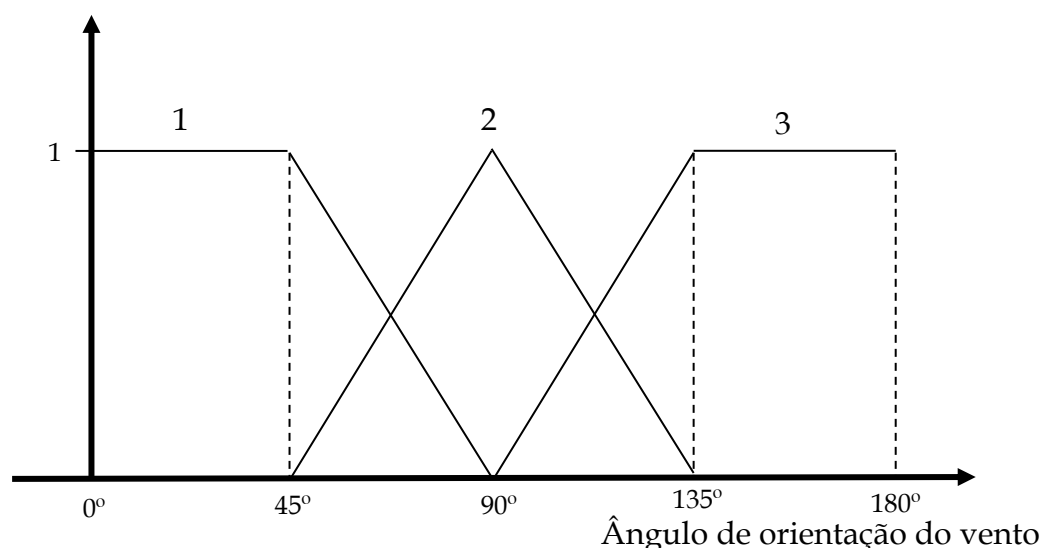


Figura 4. 11 - Funções de pertinência dado o ângulo de orientação do vento para a vela.

As associações das funções de pertinência com os cenários possíveis da orientação do vento estão representadas na tabela 4.4.

Funções de pertença	Orientação do vento
1	Vento de proa
2	Vento de través
3	Vento de popa

Tabela 4. 4 - Legenda das funções de pertença da orientação do vento.

Após a definição das entradas é necessário estabelecer as saídas. Para tal, o controlador conta com duas saídas nos dois servos de atuação, uma para o leme e outra para a vela, como ilustra a figura 4.8.

Quanto à atuação do servo leme, este tem uma gama de valores para cinco posições distintas resultando em vários graus de liberdade do sistema.

É então necessário implementar funções de pertença para cada posição, assumindo os cinco valores: 1000, 1250, 1500, 1750 e 2000, como ilustram a figura 4.12 e a tabela 4.5, codificando assim qual a posição do leme mediante o cenário atual em que se encontra o veleiro.

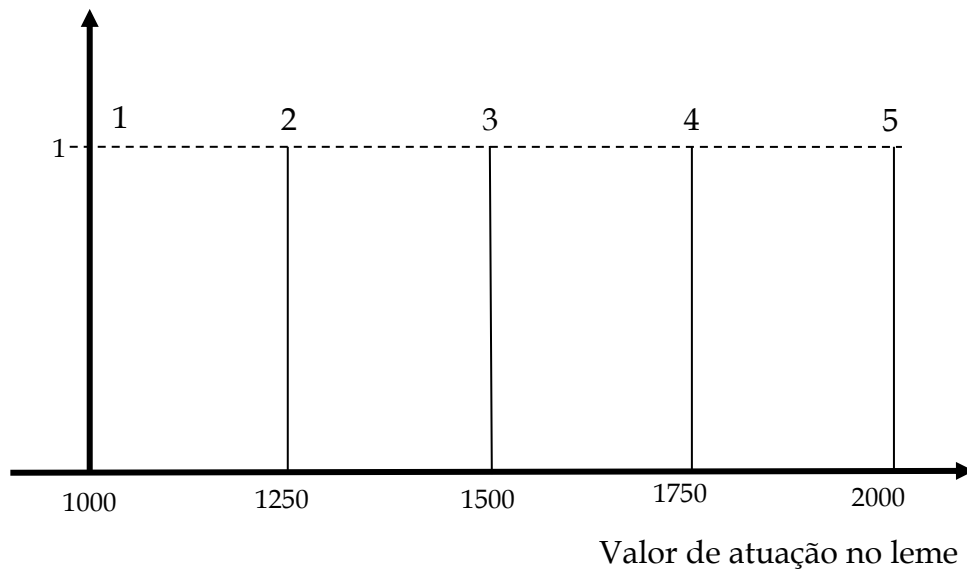


Figura 4. 12 - Funções de pertença de atuação no leme.

Função de pertença	Valor de atuação no leme	Atuação no leme
1	1000	Tudo a bombordo
2	1250	Bombordo
3	1500	Centrado
4	1750	Estibordo
5	2000	Tudo a estibordo

Tabela 4. 5 - Legenda das funções de pertença do leme.

Em seguida, a atuação do servo vela é igualmente caracterizado com três posições de variação, estando representadas na figura 4.13 e na tabela 4.6 as funções de pertença da atuação na vela.

Em suma, os graus de liberdade atribuídos permitem uma maior robustez ao sistema, pois é possível realizar uma variação mais fina das posições dos elementos móveis do veleiro.

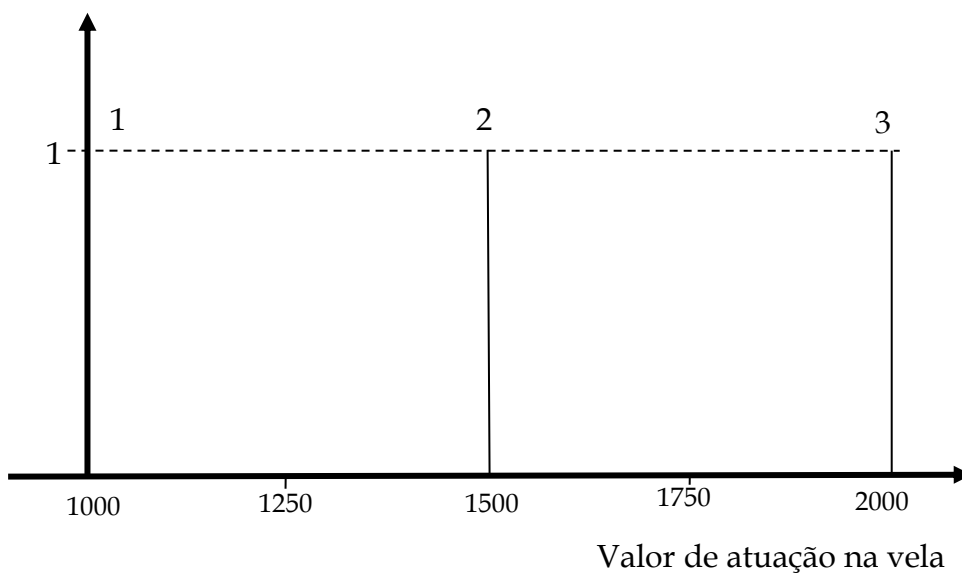


Figura 4. 13 - Funções de pertença de atuação na vela.

Função de pertença	Valor de atuação na vela	Atuação na vela
1	1000	Caçado
2	1500	Centrado
3	2000	Folgado

Tabela 4. 6 - Legenda das funções de pertença da vela.

Estrutura de Controlo

Primeiramente, à estrutura de controlo torna-se necessário relacionar as entradas com as saídas da figura 4.8. Esta relação é muito importante para determinar o comportamento do veleiro. Pode-se dizer que é o “cérebro” do controlador.

Em seguida, recorrendo ao software de geração automática de código, este necessita das entradas e saídas previamente definidas acima e ainda de um conjunto de regras responsáveis por relacionar as mesmas, como é possível verificar na figura 4.14.

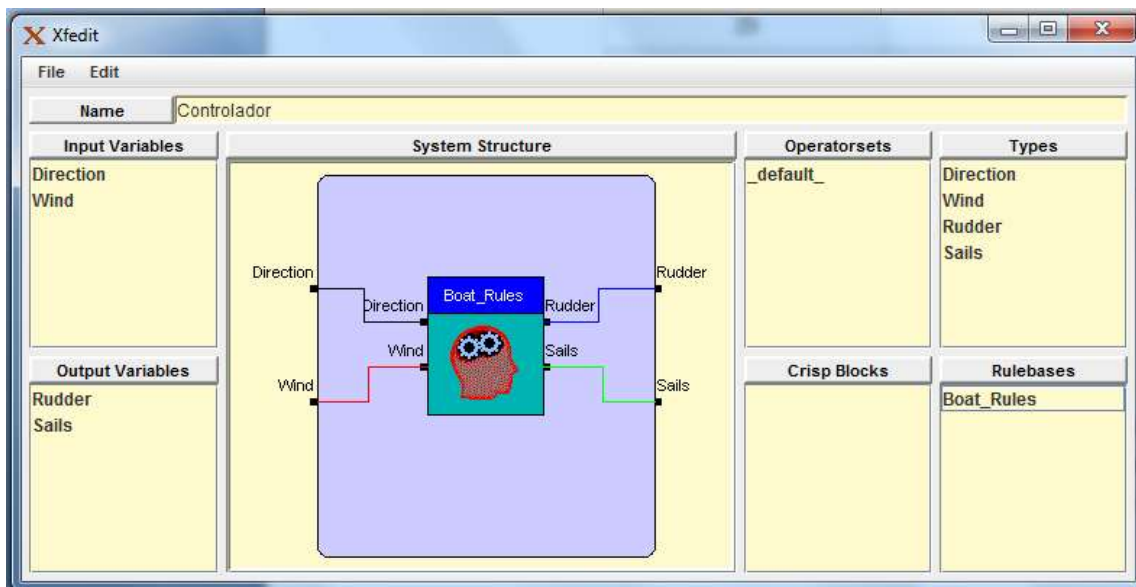


Figura 4. 14 - Controlador visto no software Xfuzzy.

É agora necessária a compreensão das regras de controlo. Com esse objetivo foram concebidas duas tabelas, tabelas 4.7 e 4.8, com todas as relações entre as entradas e saídas, isto é, mencionam-se as relações entre as funções de pertinência indicadas nas tabelas acima.

Orientação Veleiro:	Tudo para a esquerda Esquerda	Um pouco à esquerda	Centrado	Um pouco à direita	Direita	Tudo para a direita
Resposta do leme:	Tudo a bom- bordo	Bombordo	Centrado	Estibordo	Tudo a esti- bordo	

Tabela 4. 7 - Regras do controlador leme.

Direção Vento:	Vento de popa	Vento de través	Vento de proa
Resposta da vela:	Folgado	Centrado	Caçado

Tabela 4. 8 - Regras do controlador vela.

4.5 Modelizar os Conjuntos Difusos

Esta é uma abordagem que tem em vista modelizar o grau de incerteza por isso necessita de uma aprendizagem para um melhor desempenho do controlador, ou seja, é necessário realizar um “trabalho de campo” para uma aquisição sensorial recorrendo a um controlador base e, desta forma, melhorar cada vez mais a sua eficiência.

Em seguida, o ajuste dos conjuntos difusos necessita de uma realização metódica de diversos testes do controlador em diversas situações reais.

É necessário recolher dados do comportamento do veleiro, sendo que para tal este comunica através de Wi-Fi com a aplicação de gestão, reportando dados que vão sendo guardados através de uma aplicação.

Em suma, o ajuste dos conjuntos difusos necessita de uma recolha e análise de dados cuidada, pois caso sejam lidos valores incorretos aos que são verdadeiros ou interpretados de forma incorreta, o ajuste em vez de melhorar o desempenho do controlador irá realizar o efeito oposto.

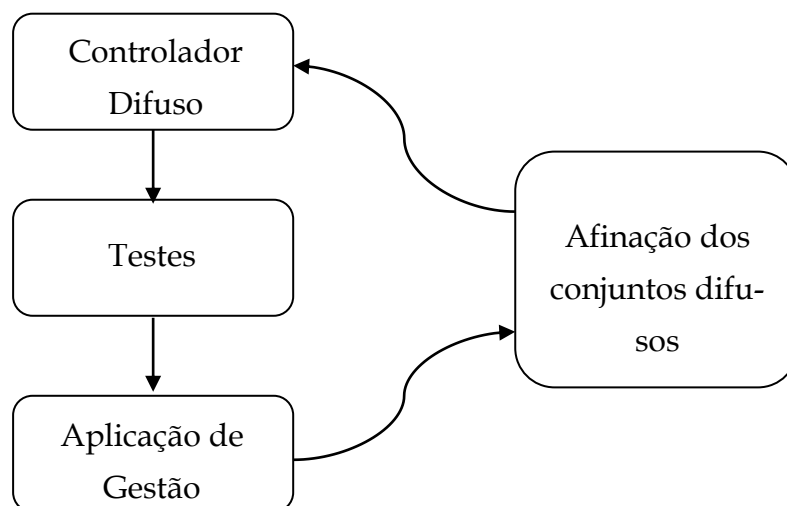


Figura 4. 15 – Esquema de aquisição e ajuste de parâmetros de controlo.

Aquisição de Dados

A aquisição de dados é essencial para o aperfeiçoamento do controlador, logo é necessário um método de aquisição eficiente e lógico, isto é, recorrendo a uma placa Wi-Fi [24] que envia dados para o computador.

A escolha do protocolo de transporte de informação é essencial para um envio eficaz de dados para o recetor. Uma das características é a comunicação em tempo real onde não existe a necessidade de uma validação dos dados à chegada, por exemplo vídeo, VoIP (Voice over Internet Protocol) [25], entre outros.

Em seguida, a comunicação é apenas realizada entre dois pontos, ou seja, apenas existe um emissor e recetor - encaminhamento "unicast" (figura 4.16). Este tipo de encaminhamento reduz possíveis erros de encaminhamento.

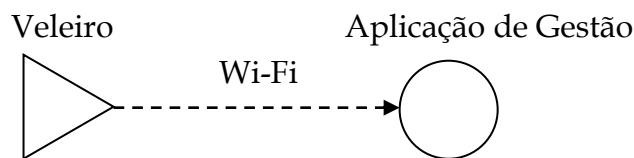


Figura 4. 16 - Encaminhamento "unicast".

Por estas razões é proposto um protocolo de transporte UDP, que aumenta a velocidade de transmissão e garante uma elevada integridade na receção correta de dados.

Assim, recorrendo ao protocolo proposto, o veleiro envia continuamente os dados importantes, definindo um "socket" UDP (endereço IP, porto de escuta) bem estruturado com a informação dos dados recolhidos, como ilustra a figura 4.17. O recetor, através de uma aplicação, escuta o endereço estabelecido e interpreta os dados, apresentando os mesmos numa interface gráfica e ainda os guarda num ficheiro de texto.

Posteriormente, dado que o veleiro não apresenta um comportamento com variações bruscas de velocidade e orientação, isto é, movimenta-se lentamente, a aquisição de dados não necessita de uma elevada taxa de armazenamento, pelo que pode ir guardando informação num período relativamente elevado, tipicamente na ordem dos cinco a dez segundos.

Porto de origem	Porto de destino
Tamanho (cabeçalho + dados)	UDP checksum
<p>Dados (24 Bytes)</p> <ul style="list-style-type: none"> • 2 bytes: Orientação do veleiro • 2 bytes: Orientação com o destino • 2 bytes: Orientação do vento • 2 bytes: Distância ao destino • 8 bytes: Coordenada atual • 8 bytes: Coordenada destino 	

Figura 4. 17 - “Socket” UDP.

Os dados do “socket” UDP foram repartidos em diversos bytes, pois cada byte suporta um número compreendido entre 0 e 255. Como por exemplo a orientação do veleiro pode ser compreendida entre 0 e 360 graus, são necessários 2 bytes para a representação do mesmo. Posteriormente, no receptor é feita a interpretação dos bytes de dados e de seguida convertido em valores reais de medição.

A interface gráfica é realizada através de um aplicação em java, que permite a interação da mesma com o veleiro, representando os dados recebidos e enviados de uma forma visual e apelativa.

Inicialmente a aplicação necessita de se conectar com o veleiro através de um hotspot previamente configurado e, em seguida, a mesma está apta a receber e enviar dados.

Quanto aos dados recebidos, estes dizem respeito às leituras dos diversos sensores embutidos e ainda possibilitam saber qual a distância até ao destino. Posteriormente a aplicação tem a capacidade de enviar coordenadas ao veleiro. Todas essas funcionalidades e configurações estão representadas nas figuras 4.18, 4.19 e 4.20.

Definição de todos os pontos GPS, definindo assim os objetivos do veleiro.

Objetivo atual a cumprir.

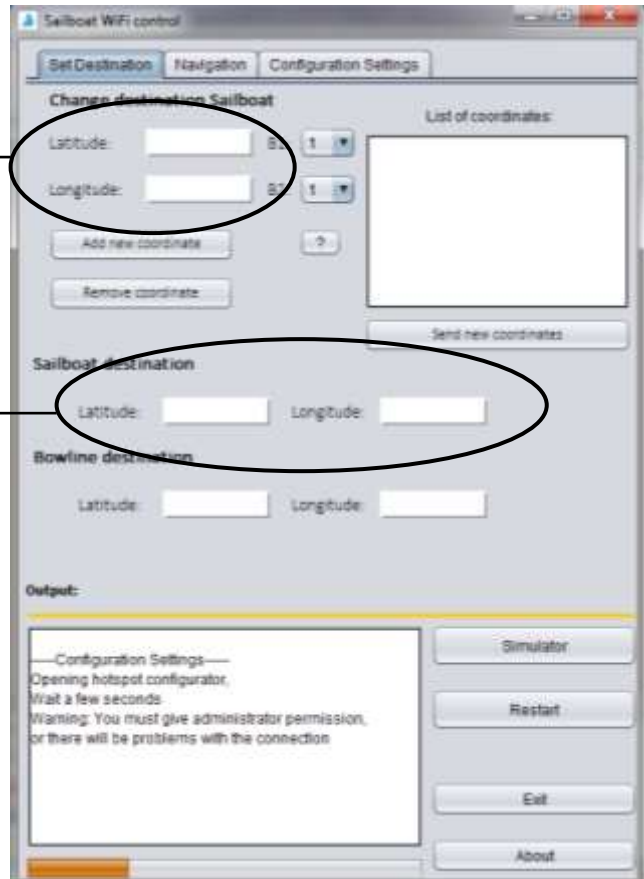


Figura 4. 18 - Interface gráfica da aplicação (definir coordenadas).



Coordenada atual onde se encontra o veleiro e possibilidade de visualização no google maps.

Leituras dos diversos sensores embutidos no veleiro e qual a distância até ao destino.

Figura 4. 19 - Interface gráfica da aplicação (leituras de variáveis).

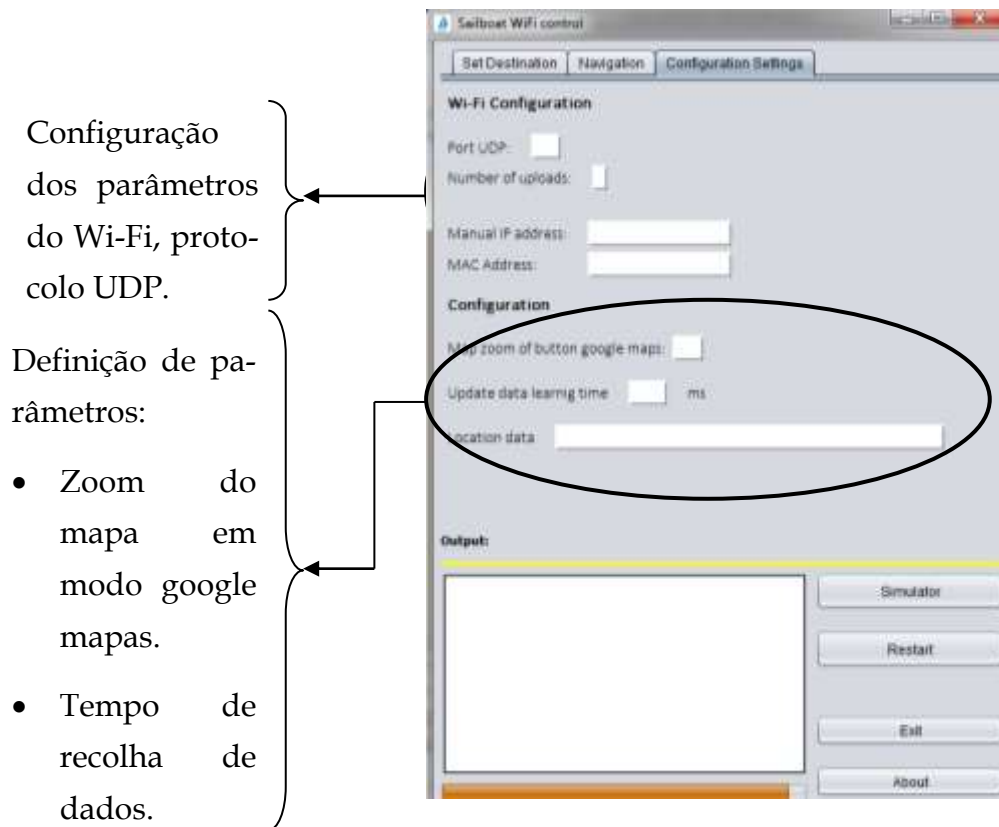


Figura 4. 20 - Interface gráfica da aplicação (configurações).



Resultados Experimentais

Os resultados experimentais permitem observar o comportamento de todo o sistema por forma a poder aperfeiçoar o sistema.

Numa primeira fase é feita uma análise da interação com os resultados e a afinação dos conjuntos difusos.

Numa última fase é analisado um resultado experimental testado em situação real e como foram realizados os ajustes de forma a obter o controlador final e uma proposta de aperfeiçoamento do mesmo.

5.1 Otimização dos conjuntos difusos

O ajuste dos conjuntos difusos passa por analisar os comportamentos do veleiro ao longo do tempo e verificar as respostas dos servos com a variação dos sensores, entradas e saídas do sistema.

Inicialmente partiu-se de um controlador no qual não estavam em vista algumas características do controlador proposto final, presente na secção 4. Este controlador inicial parte de um conjunto de funções de pertença, como ilustram a figura 5.1 e a tabela 5.1.

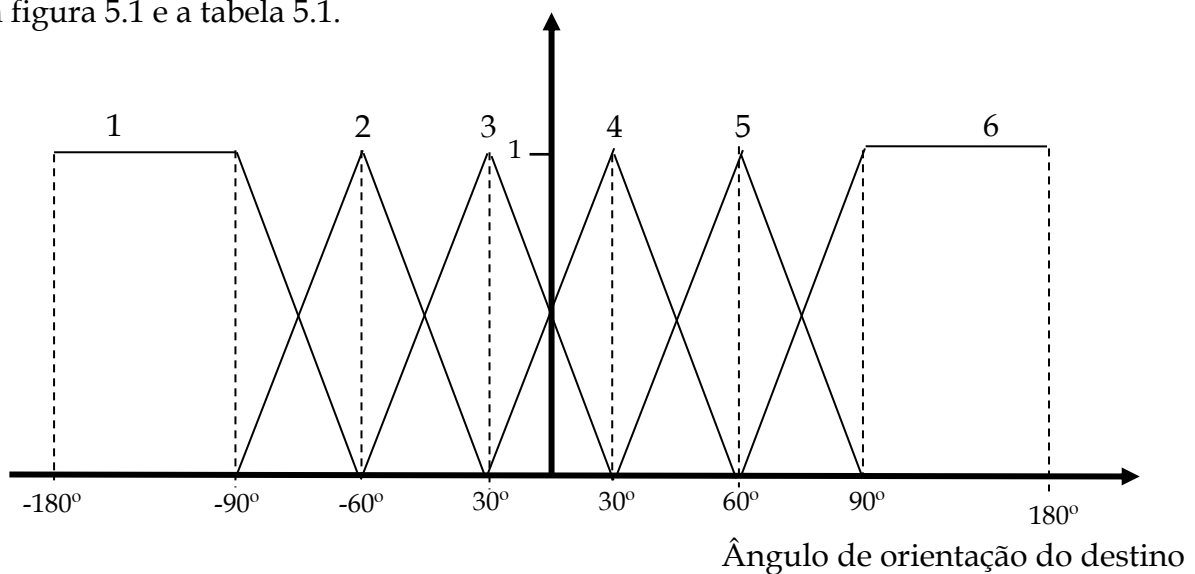


Figura 5. 1 - Função de pertinência do primeiro controlador.

Funções de pertinência	Orientação do destino em relação a proa
1	Tudo para a esquerda
2	Esquerda
3	Um pouco à esquerda
4	Um pouco à direita
5	Direita
6	Tudo para a direita

Tabela 5. 1 - Legenda função de pertinência do controlador inicial.

Existem, simplificada, duas formas de analisar o comportamento do controlador inicialmente proposto.

Quanto à primeira, esta é realizada com base numa análise visual do comportamento do veleiro em testes realizados no lago do Parque da Paz, em Almada. Uma vez feita a análise, foi possível com o auxílio de um especialista entender se o comportamento do veleiro foi o correto, ou o mais aconselhado.

Com base nesta análise é possível identificar um problema de desempenho. O controlador é eficaz na medida que alcança o destino, porém não é eficiente na decisão da melhor rota a navegar, como pode ser visto na figura 5.3.



Figura 5. 2 - Efeito de trajetória do controlador inicial.

Como podemos visualizar na figura 5.2, o teste ao controlador mostra que o veleiro se deslocou pela linha a branco. Esta linha consiste numa interpolação de diversas leituras GPS dos pontos que o veleiro atingiu.

Em seguida, era de esperar que a variação do vento variasse com a curvatura da trajetória, porém as leituras do cata-vento mostram que durante as diversas leituras o vento se manteve constante na direção assinalada.

Podemos concluir que a trajetória ideal seria uma linha reta, linha preta a tracejado na figura 5.2. Após diversos testes, é fácil de entender o porquê deste comportamento, o controlador não possui graus de liberdade para que o efeito produzido pelo leme resulte numa posição centrada.

Torna-se então necessário introduzir no controlador uma função de pertinência que defina o modo de centro do leme e, assim, mediante um conjunto de

valores obtidos sensorialmente, o veleiro é capaz de seguir o percurso em linha reta.

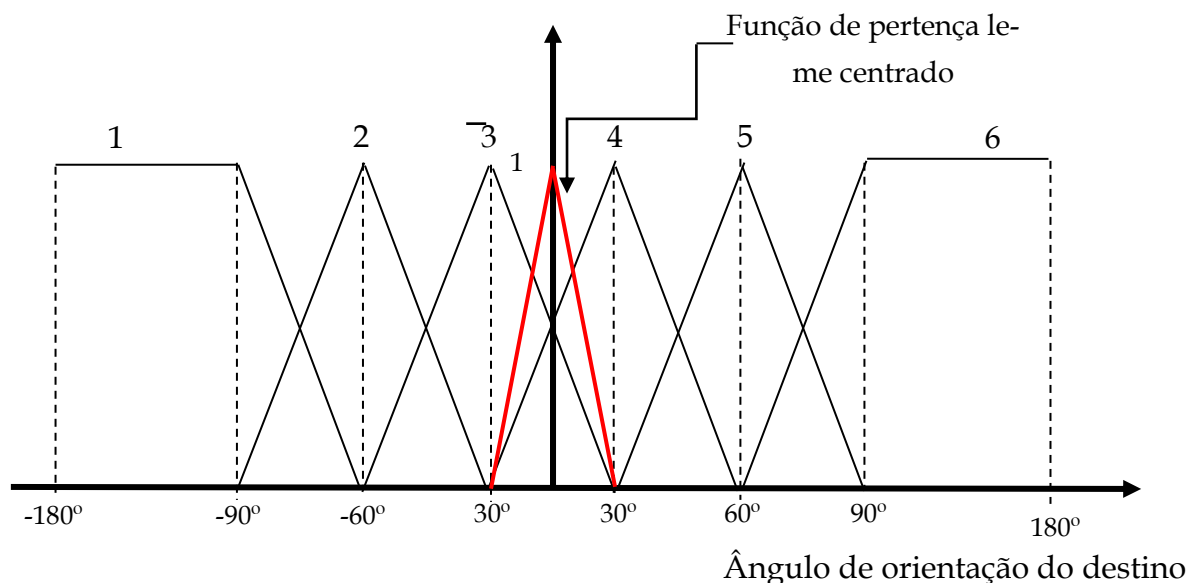


Figura 5.3 – Atualização das funções de pertinência.

Com esta atualização foi repetido o mesmo teste em situação laboratorial controlada para poder simular as mesmas condições de vento ao longo do tempo, por forma a verificar se o comportamento do veleiro era o idealizado, isto é, um movimento aproximado ao da linha a tracejado. Porém esse efeito não foi obtido exatamente como o pretendido, contudo obteve-se uma melhor aproximação, como verificado na figura 5.4.



Figura 5.4 – Efeito do ajuste no controlador.

Por inspeção à introdução de mais uma função de pertinência, a resposta do controlador aproximou-se da resposta ótima.

Uma vez feita esta alteração, é necessário realizar uma análise dos valores obtidos em situação real de navegação. Para tal, e recorrendo à aplicação de gestão, secção 4.5.1, é possível obter um ficheiro em modo de texto que documenta o percurso do veleiro ao longo do tempo (figura 5.5).

```
-----  
Destination: latitude: 38.686859- Longitude: -9.145654  
Actual Point: latitude: 38.687038- Longitude: -9.146057  
Boat Guidance: 111  
Direction to destination: 8  
Distance to destination: 40  
Wind alignment: 85  
Bowline Off  
-----  
Destination: latitude: 38.686859- Longitude: -9.145654  
Actual Point: latitude: 38.687038- Longitude: -9.146057  
Boat Guidance: 110  
Direction to destination: 9  
Distance to destination: 40  
Wind alignment: 100  
Bowline Off  
-----  
Destination: latitude: 38.686859- Longitude: -9.145654  
Actual Point: latitude: 38.687038- Longitude: -9.146057  
Boat Guidance: 144  
Direction to destination: -24  
Distance to destination: 40  
Wind alignment: 126  
Bowline Off  
-----
```

Figura 5. 5 – Documento de texto com valores lidos em situação real.

Esta aplicação, como é visível, guarda todas as informações necessárias, sendo que as variáveis de interesse para o estudo do controlador são “Direction to destination” e “Wind alignment”, que correspondem às variáveis de entrada do sistema, orientação do destino e orientação do vento respetivamente.

Em seguida, as saídas são obtidas aplicando as entradas às funções de pertença, secção 4.4.4. Uma vez obtido o valor de saída, ou seja, o valor de atuação no leme e na vela, é então necessário verificar a existência de correlação entre o ponto atual e seguinte lidos pela aplicação de gestão.

Para melhor compreensão da análise efetuada é descrito a observação de quatro pontos e posteriores respostas, como ilustra a figura 5.6.

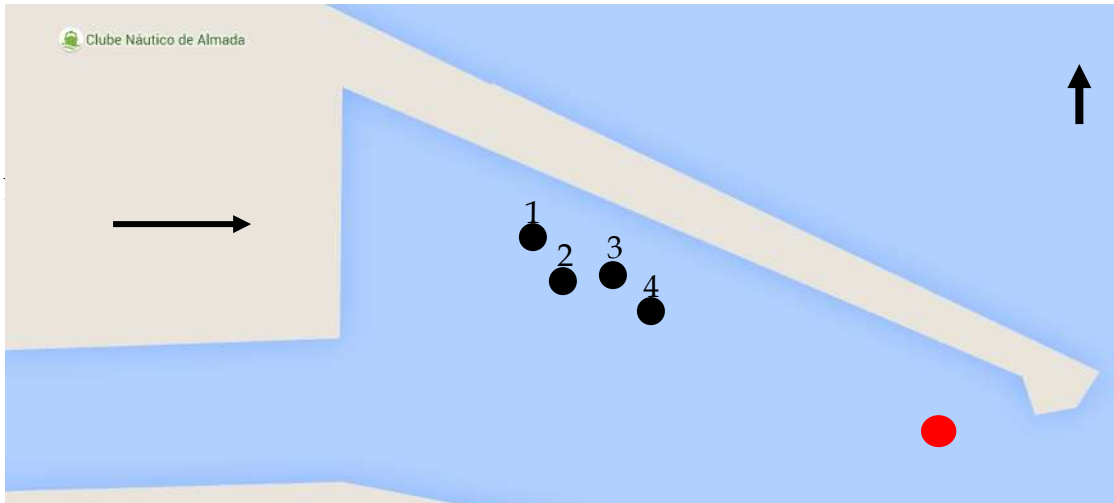


Figura 5. 6 - Resultados e análise de quatro pontos em situação de navegação.

Ponto 1 -

Destino: Latitude: 38.686859 - Longitude: -9.145654

Ponto atual: Latitude: 38.687001 - Longitude: -9.146079

Orientação do barco em relação ao norte: 106 °

Orientação do destino: -9 °

Distância ao destino: 31 metros

Orientação do vento: 179 °

Ponto 2 -

Destino: Latitude: 38.686859- Longitude: -9.145654

Ponto atual: Latitude: 38.686973- Longitude: -9.146055

Orientação do barco em relação ao norte: 109°

Orientação do destino: 12 °

Distância ao destino: 27 metros

Orientação do vento: 167°

Ponto 3 -

Destino: Latitude: 38.686859 - Longitude: -9.145654

Ponto atual: Latitude: 38.686980- Longitude: -9.146004

Orientação do barco em relação ao norte: 107 °

Orientação do destino: -5 °

Distância ao destino: 26 metros

Orientação do vento: 173 °

Ponto 4 –

Destino: Latitude: 38.686859 - Longitude: -9.145654

Ponto atual: Latitude: 38.686954- Longitude: -9.145977

Orientação do barco em relação ao norte: 111 °

Orientação do destino: -8 °

Distância ao destino: 22 metros

Orientação do vento: 174 °

Após a análise das situações dos quatro pontos é possível notar o andamento do veleiro. As duas variáveis de entrada do controlador, orientação do destino e orientação do vento mostram que a resposta do controlador a estes valores era manter o leme na posição centrada e as velas todas folgadas o que foi visto na prática. Porém, é possível verificar que os pontos não se encontram alinhados. Com a ajuda de um especialista na área da vela, foi possível concluir após o teste realizado no Rio Tejo, junto ao clube náutico em Cacilhas, que as correntes marítimas e a inclinação do veleiro são aspetos chave para o aperfeiçoamento do controlador.

De modo a melhorar estes dois aspetos, é apresentado no capítulo 6 algumas propostas de trabalhos futuros para solucionar os problemas apresentados.

Em seguida na tabela 5.2 é apresentado um conjunto completo de testes realizados ao controlador durante um teste realizado na rua Comandante António Norte, Amadora. Com a relação das entradas e saídas do sistema, obtendo assim o valor para a posição do leme e vela apresentados em percentagem. Posteriormente esses valores são convertidos em valores para atuar nos servos.

Todos os valores foram obtidos através de um ficheiro de texto gerado pela aplicação de gestão, como mostra a figura 5.7.

```
-----  
Orientacao com destino: -52<----->Orientacao do vento: 95  
Leme: 11.48<----->Vela: 58.52  
-----  
Orientacao com destino: -100<----->Orientacao do vento: 88  
Leme: 0.00<----->Vela: 47.78  
-----  
Orientacao com destino: -37<----->Orientacao do vento: 83  
Leme: 22.44<----->Vela: 42.22  
-----
```

Figura 5.7 - Ficheiro de valores de atuação nos servos.

Latitude atual	Longitude atual	Direção do vento (°)	Orientação com o destino (°)	Atuação no Leme (%)	Atuação na Vela (%)	Relação entrada-saída
38.7624999	-9.246241	161	-32	25	100	Ok
38.7624999	-9.246241	64	89	100	21,11	Ok
38.7624999	-9.246241	70	9	64,29	25	Ok
38.7624999	-9.246241	3	54	90	0	Ok
38.7624999	-9.246241	5	48	85,56	0	Ok
38.76231	-9.24627	95	-52	11,48	58,52	Ok
38.76231	-9.24627	88	-100	0	47,78	Ok
38.76231	-9.246265	83	-37	22,44	42,42	Ok
38.762294	-9.246239	36	95	100	0	Ok
38.762294	-9.246239	82	90	100	39,47	Ok
38.762294	-9.246239	7	-10	31,82	0	Ok
38.762294	-9.246239	2	-17	25	0	Ok
38.762294	-9.246239	0	65	98,15	0	Ok
38.762439	-9.246136	17	76	100	0	Ok
38.762439	-9.246136	19	77	100	0	Ok
38.762439	-9.246136	2	-23	25	0	Ok
38.762439	-9.246136	1	-39	21,11	0	Ok
38.762439	-9.246136	7	53	89,26	0	Ok
38.762462	-9.245889	22	28	75	0	Ok

38.762462	-9.245818	8	68	100	0	Ok
38.762462	-9.245818	5	69	100	0	Ok
38.76247	-9.245716	40	-4	46,12	0	Ok
38.762474	-9.245700	33	-40	20,37	0	Ok
38.762451	-9.24561	31	-4	46,12	0	Ok
38.762443	-9.24561	44	-17	25	0	Ok
38.762443	-9.24561	6	65	98,15	0	Ok
38.762443	-9.24561	7	72	100	0	Ok
38.762443	-9.24561	22	31	75	0	Ok
38.762522	-9.24550	24	16	75	0	Ok
38.762541	-9.24543	15	70	100	0	Ok
38.762541	-9.245438	18	70	100	0	Ok
38.762541	-9.245438	16	66	98,89	0	Ok
38.762545	-9.245196	44	26	75	0	Ok
38.762545	-9.245118	33	105	100	0	Ok
38.762561	-9.245091	27	67	99,63	0	Ok
38.762576	-9.244990	3	27	75	0	Ok
38.762527	-9.244944	74	26	75	32,12	Ok
38.762489	-9.244883	17	-20	25	0	Ok
38.762493	-9.244848	33	78	100	0	Ok
38.762493	-9.244848	26	52	88,52	0	Ok
38.762503	-9.244779	46	75	100	1,39	Ok
38.762503	-9.244779	47	76	100	2,80	Ok
38.762503	-9.244779	25	35	75,93	0	Ok
38.762443	-9.244852	17	-30	25	0	Ok

38.762434	-9.244852	15	28	75	0	Ok
38.762434	-9.244852	41	92	100	0	Ok
38.762434	-9.244852	22	33	75	0	Ok
38.762451	-9.244738	48	58	92,96	4,25	Ok
38.762451	-9.244738	63	72	100	20	Ok
38.762451	-9.244738	105	171	100	66,67	Ok
38.762381	-9.245439	96	-10	31,82	65,52	Ok
38.762381	-9.245439	47	99	100	2,27	Ok
38.762381	-9.245439	3	-16	25	0	Ok
38.762320	-9.24532	69	-17	25	25,95	Ok
38.762320	-9.24532	10	-2	48,32	0	Ok

Tabela 5. 2 - Teste à resposta do controlador

Para concluir, a análise dos resultados experimentais foram essenciais para a afinação do controlador, contudo também mostraram que existe uma necessidade de um melhoramento do mesmo e dos sensores para conhecer melhor o cenário de navegação, por forma a contornar situações e cenários de navegação menos favoráveis.



Conclusões e Trabalhos Futuros

Para concluir, é feita uma análise aos aspetos mais e menos positivos da realização desta dissertação. Com a análise desses aspetos serão propostos trabalhos futuros e possíveis melhorias ao sistema.

6.1 Conclusões

Inicialmente, a arquitetura previamente idealizada foi ao longo do tempo reestruturada, mostrando que a idealização de um modelo feita de início não foi a mais adequada, logo é importante mencionar que este é um aspeto muito importante em qualquer projeto. Contudo, a arquitetura final, baseada num modelo hierárquico, mostrou ser uma forma mais organizada de estruturar um modelo.

Em seguida, os sensores escolhidos mostraram uma boa resposta aos testes reais, tendo o GPS mostrado algumas dificuldades de conexão, porém este depende muito do local e das condições climáticas. Os servomotores utilizados estiveram de acordo com as expectativas, mas verificou-se a necessidade de ter alguns cuidados para evitar esforços mecânicos ou problemas elétricos, de forma a evitar sobreaquecimentos.

O Arduíno mostrou ser uma boa plataforma de controlo, isto é, a interligação de dispositivos com o mesmo são de fácil aprendizagem e há muita informação de auxílio nas comunidades da plataforma.

O controlador implementado mostrou bons resultados, porém o método de afinação do mesmo não foi o melhor, porque a necessidade constante de um especialista para a análise de resultados dificulta este processo. Ainda assim mostrou uma boa eficácia a cumprir os destinos pretendidos.

Para a monitorização de todo o processo foi implementada uma aplicação de gestão que foi a peça fundamental para a construção e desenvolvimento deste projeto. Com esta aplicação foi possível observar os dados em tempo real e as decisões tomadas pelo veleiro, sendo possível verificar o seu comportamento.

Posteriormente, com a monitorização verificou-se que as correntes marítimas são uma peça fundamental para a navegação, pois na presença das mesmas o comportamento do veleiro não era o ideal, segundo especialistas.

Este projeto contou com a participação na prova “Robotics EXercise 2015” organizado pela Marinha Portuguesa através do centro de investigação localizado no Canal do Arsenal do Alfeite, Almada, tendo sido um exercício de aprendizagem e aperfeiçoamento dos controladores do veleiro.

Para concluir, no geral, a navegação autónoma em condições de vento favorável foi conseguida e observada por diversas vezes porém é notória a necessidade de aperfeiçoar o controlador e os sensores de análise das condições envolventes. Importa destacar ainda que a aplicação construída é muito importante e obteve bons resultados como um auxiliar do controlo e análise do veleiro.

6.2 Trabalhos a Desenvolver no Futuro

Em consideração aos testes realizados e análise comportamental do veleiro surge a necessidade de apresentar trabalhos futuros a serem realizados.

- **Atualização do controlador com a inclinação**

Um dos problemas detetados foi a inclinação do veleiro, com a força do vento o veleiro adorna para os lados e, ao se folgar as velas no momento da inclinação, faz com que as mesmas batam na água. Para melhorar este problema é necessária a introdução de mais uma função de pertinência no controlador que indique se o veleiro está direito ou inclinado, como ilustra a figura 6.1. Esta inclinação é medida pela variável de inclinação (roll) da bússola.

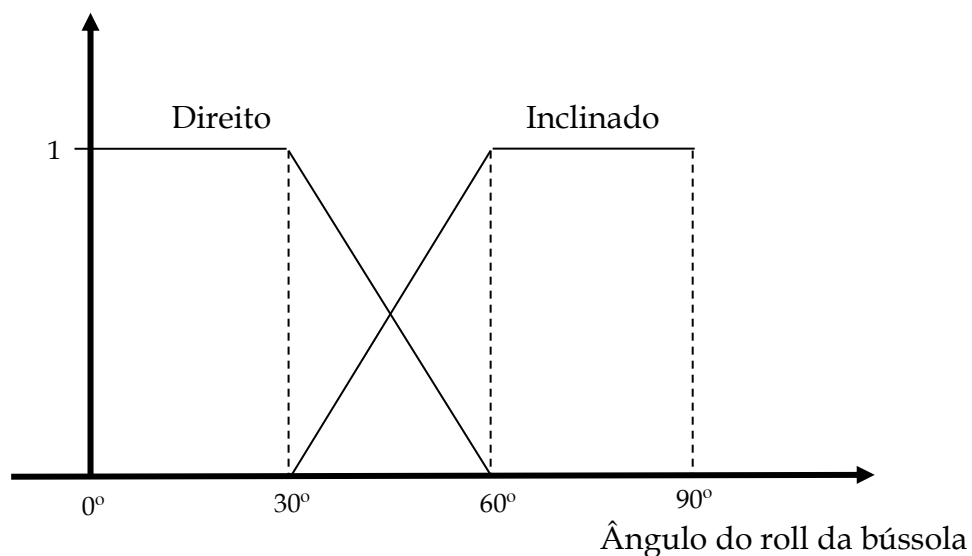


Figura 6. 1- Função de pertinência da inclinação.

Posteriormente à criação da função de pertença de inclinação é necessário incluir nas regras de controlo da vela, tendo uma situação para quando está inclinado e outra para quando o veleiro está direito.

- **Análise das correntes marítimas**

Um dos grandes problemas deparados foi a existência de correntes marítimas, para tal a solução passa por analisar as mesmas através de uma ventoinha submersa na parte traseira do veleiro, como ilustra a figura 6.2.



Figura 6. 2 - Sensor de velocidade do veleiro em relação ao mar [26].

Para poder analisar a corrente marítima basta comparar a velocidade do veleiro em relação à terra, sendo esta velocidade medida através do GPS e comparar com a velocidade do veleiro em relação à água. Com isto pode ser averiguado se o veleiro está a velejar contra a corrente e tomar uma decisão de alterar a sua rota, por forma a tirar maior partido das correntes.

- **Bomba de extração de água**

Um aspeto fundamental observado é a necessidade de extrair a água que entra no casco do veleiro já que em todos os testes realizados foi encontrada água dentro do casco. Para tal surge a necessidade de introduzir uma bomba de extração de água no veleiro (como ilustra a figura 6.3), uma vez que com dois

motores pequenos isolados e um sensor de medidor da altura da água dentro do casco é possível implementar o sistema de extração de água.

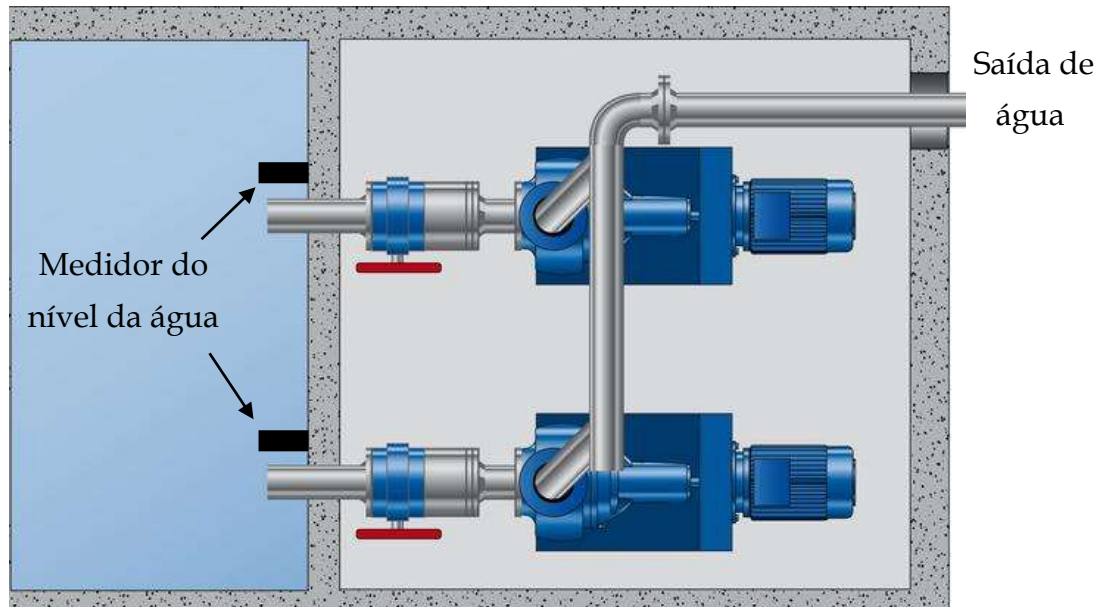


Figura 6. 3 - Esquema de extração de água do veleiro [27].

- **Autonomia energética do veleiro**

Um aspeto a ser revisto é a autonomia energética. Neste momento o veleiro necessita de carregar as pilhas, pois não tem a capacidade de autocarregar as mesmas, para tal surge a ideia de instalar mecanismos que carreguem as baterias de forma autónoma, isto é, através de energias renováveis. Por exemplo, tal poderá ser conseguido com a instalação de painéis fotovoltaicos no veleiro de pequenas dimensões, os quais podem ser rígidos ou flexíveis, sendo possível a instalação em diversos locais do veleiro. De notar que será necessário alterar a configuração elétrica de todo o sistema.

Referências

- [1] M. F. P. Santos, *eVentos 4- Controlador para navegação autónoma de veleiro em modo de regata*, Almada: Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2015.
- [2] P. FEUP, “FASt - Autonomous sailing boat,” Department of Electrical and Computer Engineering of the University of Porto, Portugal, [Online]. Available: <http://paginas.fe.up.pt/~jca/fast/>. [Acedido em 12 junho 2015].
- [3] F. M. G. Martins, *eVentos 2 - Autonomous sailboat control*, Almada: Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2013.
- [4] T. Lochhaas, “Basic Sailboat Terms - Essential Words You Need to Know,” [Online]. Available: <http://sailing.about.com/od/introductiontosailing/a/Basic-Sailboat-Terms.htm>. [Acedido em 14 Julho 2015].
- [5] *User manual ThunderTiger Naulantia*, 2015.
- [6] ange, “NAVEGAÇÃO À VELA,” [Online]. Available: <http://www.ange.pt/arquivo/avela.htm>. [Acedido em 15 Julho 2015].

- [7] Learn to sail, "Points of sailing names," [Online]. Available: <http://www.learn-sailing.com/tracking.html>. [Acedido em 14 junho 2015].
- [8] Microtransat, "The Microtransat Challenge," [Online]. Available: <http://www.microtransat.org/>. [Acedido em 13 junho 2015].
- [9] J. C. Alves e N. A. Cruz, "Um sistema computacional reconfiguravel embarcado num veleiro autónomo," em *Jornadas Sobre Sistemas Reconfiguráveis - REC*, Monte da Caparica, Portugal, 2009.
- [10] L. Gomes, A. Costa, F. Moutinho e R. Mota, "Attracting students to Engineering through," em *IEEE ICIT 2015*, Sevilha, Espanha, 2015.
- [11] Arduino, "Arduino Mega," [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMega>. [Acedido em 20 Maio 2015].
- [12] AS5161, "Datasheet," [Online]. Available: http://ams.com/eng/content/download/326783/1143637/file/AS5161_Datasheet_EN_v1.pdf/320895. [Acedido em 06 Maio 2015].
- [13] L. T. Inc, *Datasheet of GPS smart antenna module*, Taiwan, 2006.
- [14] NMEA, *The NMEA 0183 Protocol*, New Bern, USA, 2000.
- [15] R. Electronics, "CMPS10 - Tilt Compensated Compass Module," [Online]. Available: <http://www.robot-electronics.co.uk/htm/cmsps10doc.htm>. [Acedido em 11 Maio 2015].
- [16] Atmarine, "Roll, Pitch and Yaw," [Online]. Available: <http://www.atmarine.fi/ckfinder/userfiles/images/roll%20pitch%20pic.bmp>. [Acedido em 4 Maio 2015].

- [17] F. Velo, I. Baturone, S. Solano e A. Barriga, "Rapid design of fuzzy," em *The 12th IEEE International Conference on Fuzzy Systems*, 2003.
- [18] C. Negoita, *Expert systems and fuzzy systems*, Pequim: Benjamin/Cummings Pub. Co, 1986.
- [19] S. K. Alonso, "Structure of a fuzzy inference system," [Online]. Available: http://www.dma.fi.upm.es/java/fuzzy/fuzzyinf/introfis_en.htm. [Acedido em 12 Maio 2015].
- [20] Yaldex, "Fuzzification," [Online]. Available: http://www.yaldex.com/game-development/1592730043_ch33lev1sec2.html. [Acedido em 15 Maio 2015].
- [21] C. Lucena, *Aplicação de técnicas de controlo óptimo*, Almada: Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, 2012.
- [22] M. M. Elkhatab e J. J. Soraghan, *Fuzzy Stabilization of Fuzzy Control Systems*, University of Strathclyde, United Kingdom, 1998.
- [23] L. Gomes, M. Santos, T. Pereira e A. Costa, "Model-Based Development of an Autonomous Sailing Yacht Controller," em *Autonomous Robot Systems and Competitions (ICARSC)*, Vila Real, Portugal, 2015.
- [24] Seeedstudio, "Wi-Fi Shield," [Online]. Available: <http://www.seeedstudio.com/depot/Wifi-Shield-p-1220.html>. [Acedido em 14 Maio 2015].
- [25] B. Goode, *Voice over Internet protocol (VoIP)*, USA: IEEE, 2002.
- [26] nautiplanet, "nautiplanet," [Online]. Available: <http://www.nautiplanet.com/sensores-humminbird-sensor-de-velocidade-sw-p-842.html?osCsid=j7gggc9c0e6alrgcellac4cjs3>. [Acedido em 12 Agosto 2015].

- [27] KSB, "KSB," [Online]. Available: http://www.ksb.com/ksb-br-pt/Produtos_e_servicos/aguas_residuais/transporte_municipal_aguas_residuais/estacoes_bombeamento/. [Acedido em 14 Agosto 2015].
- [28] Circuitlab, "circuitlab editor," [Online]. Available: <https://www.circuitlab.com/editor/#?id=7pq5wm>. [Acedido em 16 Agosto 2015].

Circuito Elétrico Veleiro.

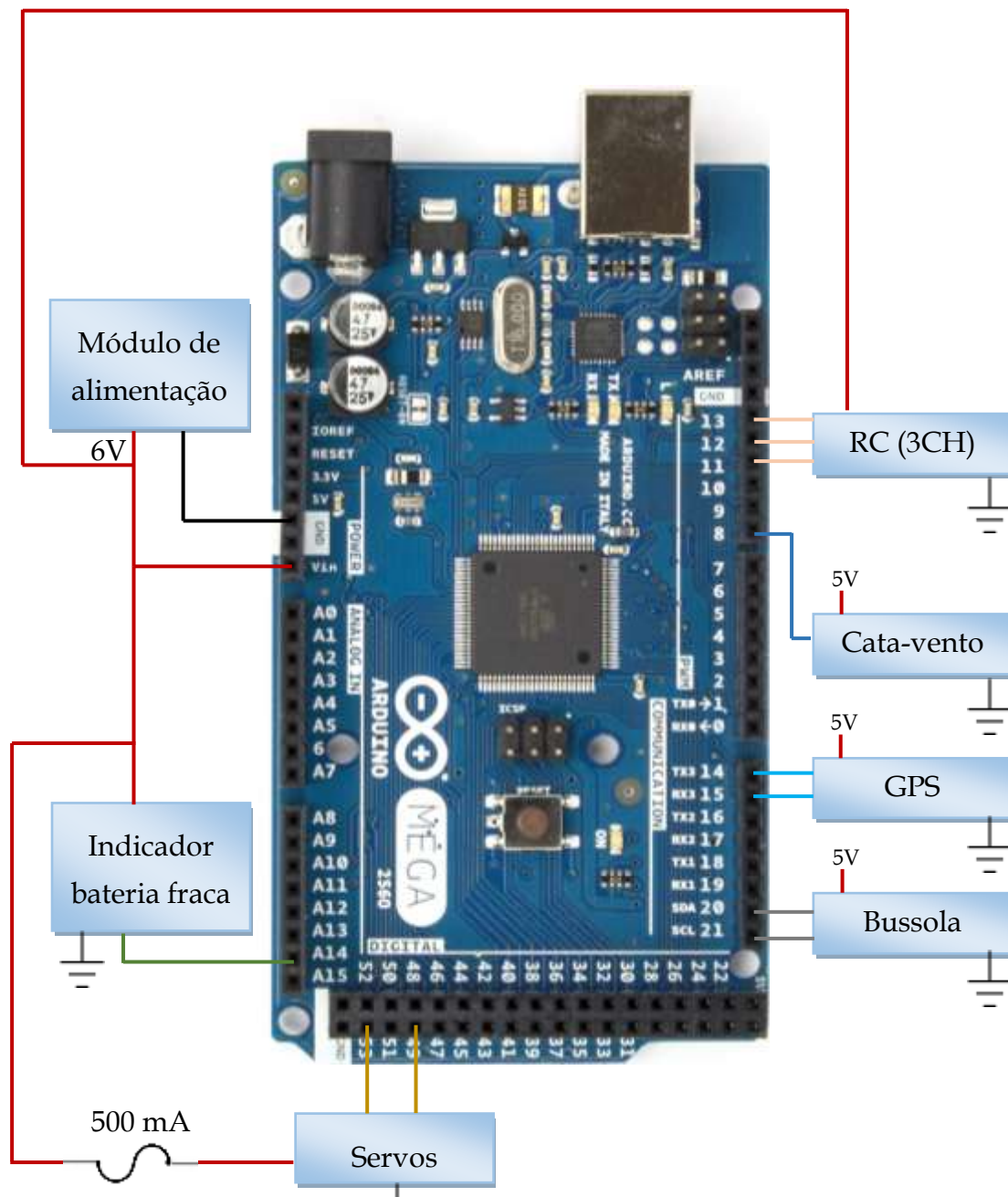


Figura A1.1 - Esquema em blocos das ligações.

Primeiramente, é descrita o esquema de ligações do veleiro, mencionando os módulos que descreve o hardware.

Seguidamente, como é visível na figura A1.1, o fusível na alimentação dos servos protege os mesmos de picos de corrente, que poderá ocorrer quando existe algum problema mecânico.

Posteriormente, é descrito o módulo de alimentação que permite implementar um interruptor magnético (SW1), para ligar e desligar o hardware, evitando a necessidade de um interruptor à prova de água, ou necessidade de abertura da embarcação.

Recorre-se à montagem seguinte, figura A1.2, pois o SW1 não suporta as correntes impostas pelo circuito, funcionando apenas como elemento polarizador do transístor, podendo este suportar a corrente alimentando todo o hardware.

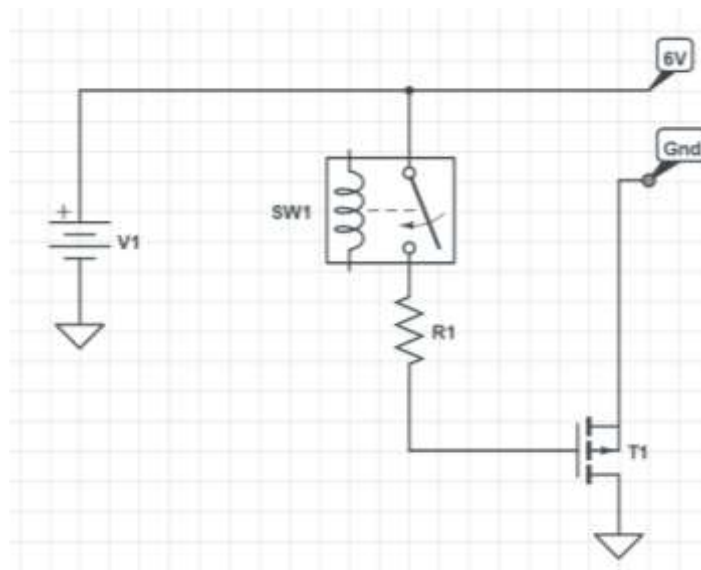


Figura A1. 2 - Esquema elétrico módulo de alimentação [28].

Componente	Referência
V1 - Bateria de alimentação	NiMH 6V 2200mAh
SW1 - Interruptor, Relé reed	V23100 V4005-A000
R1 - Resistência de polarização	10 kΩ
T1 - Transístor	IRF840

Tabela A1. 1 - Tabela dos componentes para o módulo de alimentação.

Por fim, o indicador de bateria fraca permite ao utilizador saber quando a mesma tem de ser substituída ou carregada, dando assim graus de liberdade de programação a possíveis rotinas. Este módulo consiste num divisor de tensão que protege o porto analógico da plataforma de controlo de ultrapassar o seu valor máximo permitido, isto é 5V.

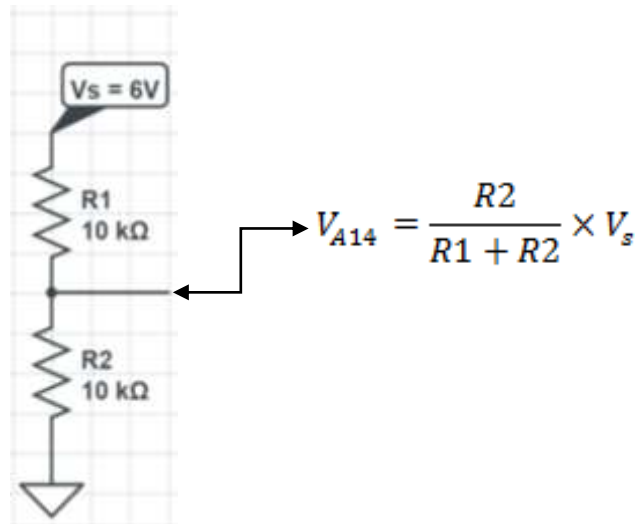


Figura A1. 3 - Indicador bateria fraca [28].